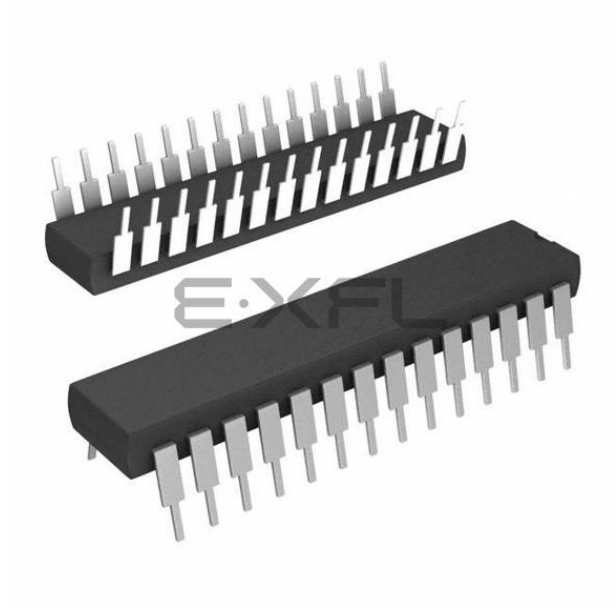


**Welcome to E-XFL.COM**

## 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 | 20MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 22 |
| Program Memory Size | 7KB (4K x 14) |
| Program Memory Type | OTP |
| EEPROM Size | - |
| RAM Size | 192 x 8 |
| Voltage - Supply (Vcc/Vdd) | 4V ~ 6V |
| Data Converters | A/D 5x8b |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 28-DIP (0.300", 7.62mm) |
| Supplier Device Package | 28-SPDIP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16c73a-20i-sp |

**TABLE 3-3: PIC16C74/74A/77 PINOUT DESCRIPTION**

| Pin Name | DIP Pin# | PLCC Pin# | QFP Pin# | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|---|
| OSC1/CLKIN | 13 | 14 | 30 | I | ST/CMOS(4) | Oscillator crystal input/external clock source input. |
| OSC2/CLKOUT | 14 | 15 | 31 | 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 | 1 | 2 | 18 | 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 | 2 | 3 | 19 | I/O | TTL | RA0 can also be analog input0 |
| RA1/AN1 | 3 | 4 | 20 | I/O | TTL | RA1 can also be analog input1 |
| RA2/AN2 | 4 | 5 | 21 | I/O | TTL | RA2 can also be analog input2 |
| RA3/AN3/VREF | 5 | 6 | 22 | I/O | TTL | RA3 can also be analog input3 or analog reference voltage |
| RA4/T0CKI | 6 | 7 | 23 | I/O | ST | RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type. |
| RA5/$\overline{SS}$/AN4 | 7 | 8 | 24 | I/O | TTL | RA5 can also be analog input4 or the slave select for the synchronous serial port. |
| | | | | | | PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. |
| RB0/INT | 33 | 36 | 8 | I/O | TTL/ST(1) | RB0 can also be the external interrupt pin. |
| RB1 | 34 | 37 | 9 | I/O | TTL | |
| RB2 | 35 | 38 | 10 | I/O | TTL | |
| RB3 | 36 | 39 | 11 | I/O | TTL | |
| RB4 | 37 | 41 | 14 | I/O | TTL | Interrupt on change pin. |
| RB5 | 38 | 42 | 15 | I/O | TTL | Interrupt on change pin. |
| RB6 | 39 | 43 | 16 | I/O | TTL/ST(2) | Interrupt on change pin. Serial programming clock. |
| RB7 | 40 | 44 | 17 | I/O | TTL/ST(2) | Interrupt on change pin. Serial programming data. |

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 an 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 as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

### 3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-4.

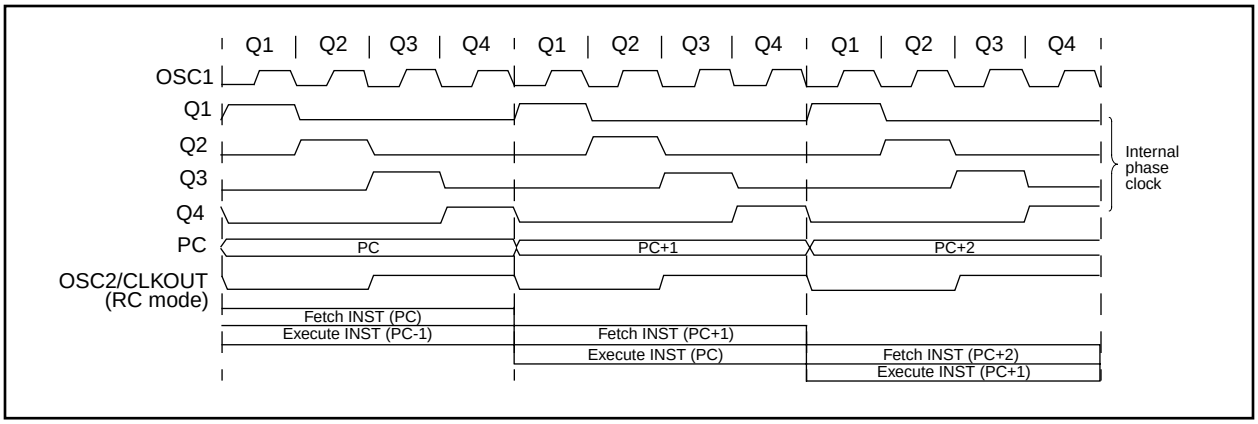
### 3.2 Instruction Flow/Pipelining

An “Instruction Cycle” consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. GOTO) then two cycles are required to complete the instruction (Example 3-1).

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

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

**FIGURE 3-4: CLOCK/INSTRUCTION CYCLE**

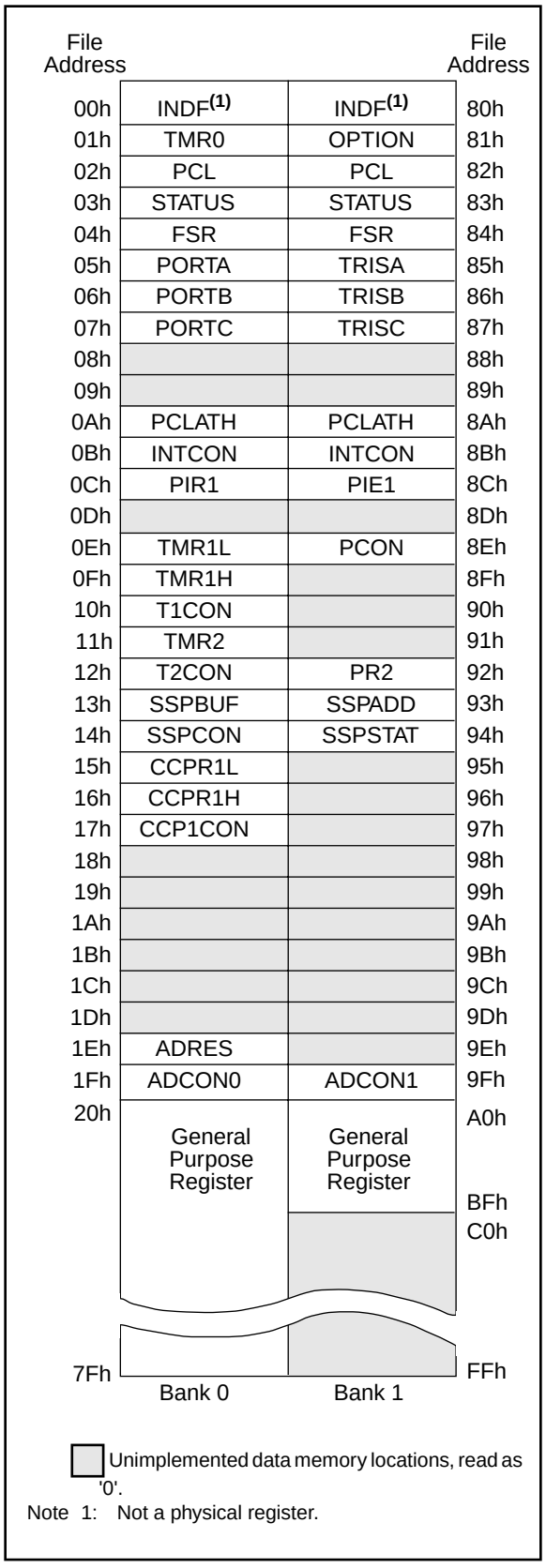



**EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW**

| | Tcy0 | Tcy1 | Tcy2 | Tcy3 | Tcy4 | Tcy5 |
|---|---|---|---|---|---|---|
| 1. MOVLW 55h | Fetch 1 | Execute 1 | | | | |
| 2. MOVWF PORTB | | Fetch 2 | Execute 2 | | | |
| 3. CALL SUB_1 | | | Fetch 3 | Execute 3 | | |
| 4. BSF PORTA, BIT3 (Forced NOP) | | | | Fetch 4 | Flush | |
| 5. Instruction @ address SUB_1 | | | | | Fetch SUB_1 | Execute SUB_1 |

All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is “flushed” from the pipeline while the new instruction is being fetched and then executed.

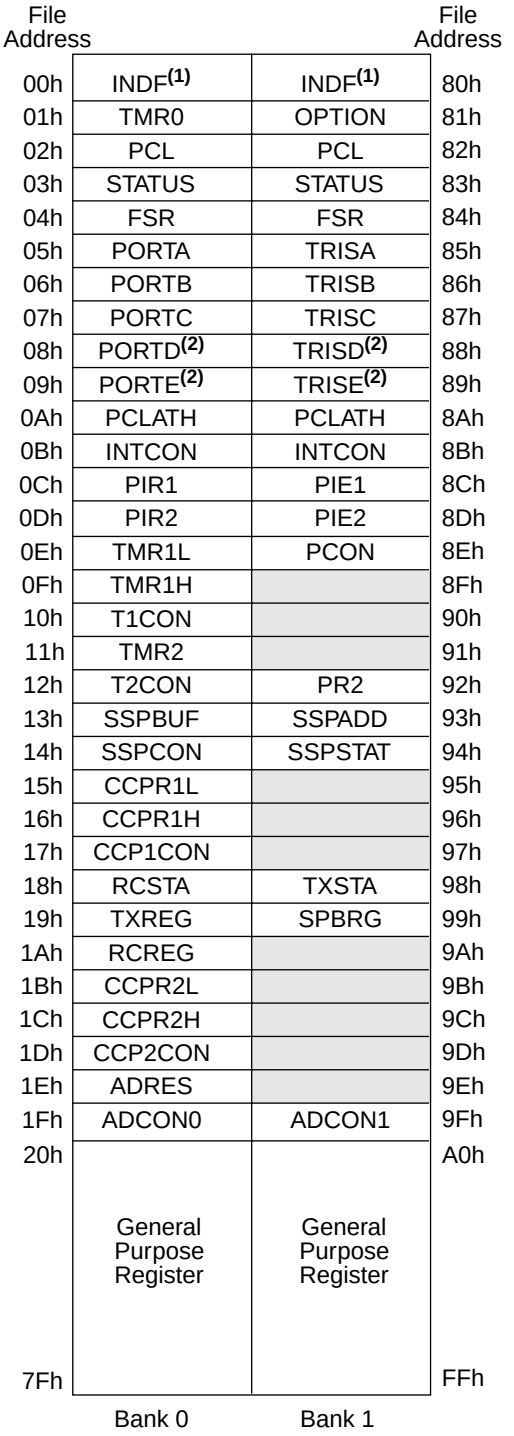
**FIGURE 4-4: PIC16C72 REGISTER FILE MAP**




Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.

**FIGURE 4-5: PIC16C73/73A/74/74A REGISTER FILE MAP**




Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.

2: These registers are not physically implemented on the PIC16C73/73A, read as '0'.

4.2.2.6 PIE2 REGISTER

| Applicable Devices | | | | | | |
|---|---|---|---|---|---|---|
| 72 | 73 | 73A | 74 | 74A | 76 | 77 |

This register contains the individual enable bit for the CCP2 peripheral interrupt.

**FIGURE 4-14: PIE2 REGISTER (ADDRESS 8Dh)**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | CCP2IE |
| bit7 | | | | | | | bit0 |

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

bit 7-1: **Unimplemented:** Read as '0'

bit 0: **CCP2IE**: CCP2 Interrupt Enable bit
1 = Enables the CCP2 interrupt
0 = Disables the CCP2 interrupt

**TABLE 5-1: PORTA FUNCTIONS**

| Name | Bit# | Buffer | Function |
|---|---|---|---|
| RA0/AN0 | bit0 | TTL | Input/output or analog input |
| RA1/AN1 | bit1 | TTL | Input/output or analog input |
| RA2/AN2 | bit2 | TTL | Input/output or analog input |
| RA3/AN3/VREF | bit3 | TTL | Input/output or analog input or VREF |
| RA4/T0CKI | bit4 | ST | Input/output or external clock input for Timer0<br>Output is open drain type |
| RA5/$\overline{SS}$/AN4 | bit5 | TTL | Input/output or slave select input for synchronous serial port or analog input |

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: POR, BOR | Value on all other resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 05h | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | --0x 0000 | --0u 0000 |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | --11 1111 | --11 1111 |
| 9Fh | ADCON1 | — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 | ---- -000 | ---- -000 |

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTA.

The $\overline{SS}$ pin allows a synchronous slave mode. The SPI must be in slave mode (SSPCON<3:0> = 04h) and the TRISA<5> bit must be set the for synchronous slave mode to be enabled. When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte, and becomes a floating output. If the $\overline{SS}$ pin is taken low without resetting SPI mode, the transmission will continue from the point at which it was taken high. External pull-up/pull-down resistors may be desirable, depending on the application.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

**FIGURE 11-5: SPI MODE TIMING, MASTER MODE OR SLAVE MODE W/O $\overline{SS}$ CONTROL**

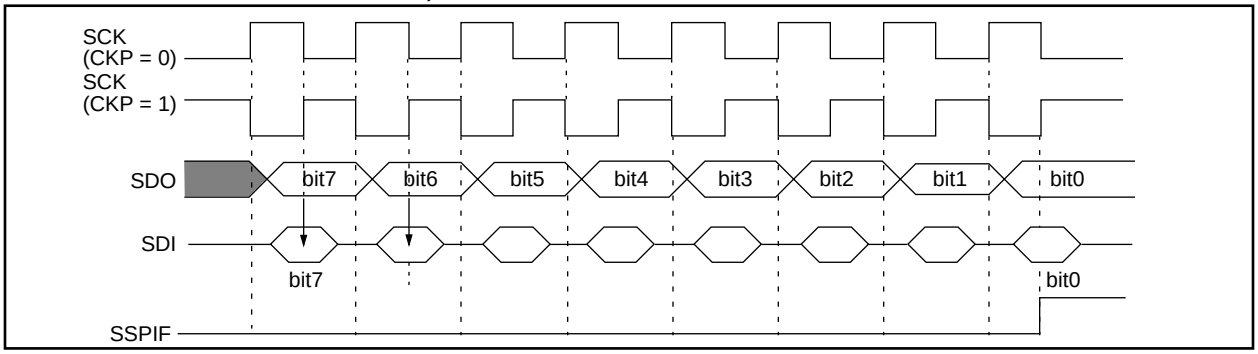



**FIGURE 11-6: SPI MODE TIMING, SLAVE MODE WITH $\overline{SS}$ CONTROL**

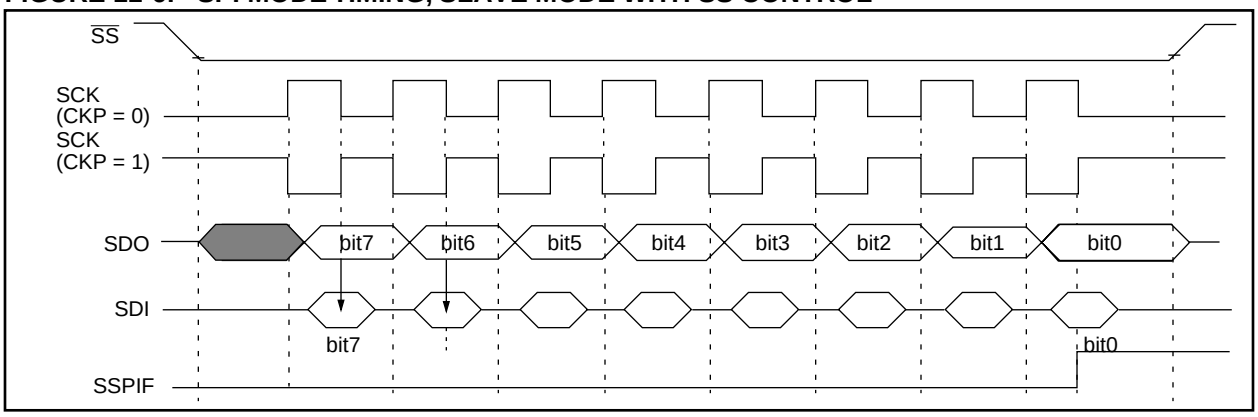



**TABLE 11-1: REGISTERS ASSOCIATED WITH SPI OPERATION**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1,2) | ADIF | RCIF(2) | TXIF(2) | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1,2) | ADIE | RCIE(2) | TXIE(2) | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | --11 1111 | --11 1111 |
| 94h | SSPSTAT | — | — | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | --00 0000 | --00 0000 |

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the SSP in SPI mode.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A, always maintain these bits clear.
2: The PIC16C72 does not have a Parallel Slave Port or USART, these bits are unimplemented, read as '0'.

**FIGURE 11-13: SPI MODE TIMING (SLAVE MODE WITH CKE = 1) (PIC16C76/77)**

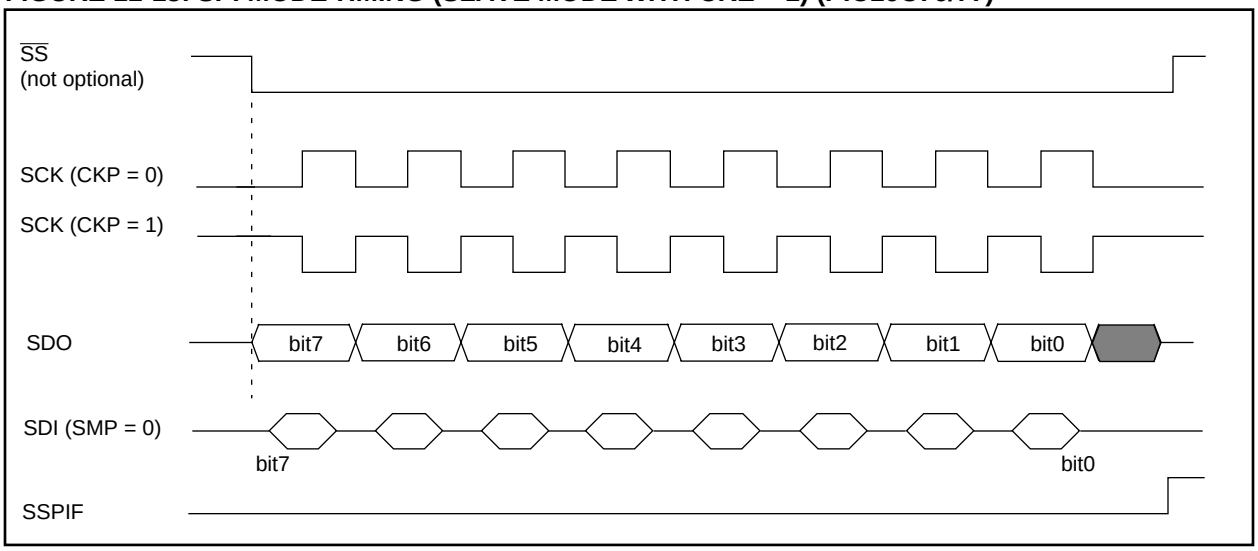



**TABLE 11-2: REGISTERS ASSOCIATED WITH SPI OPERATION (PIC16C76/77)**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh. 10Bh,18Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | --11 1111 | --11 1111 |
| 94h | SSPSTAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 0000 0000 | 0000 0000 |

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the SSP in SPI mode.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C76, always maintain these bits clear.

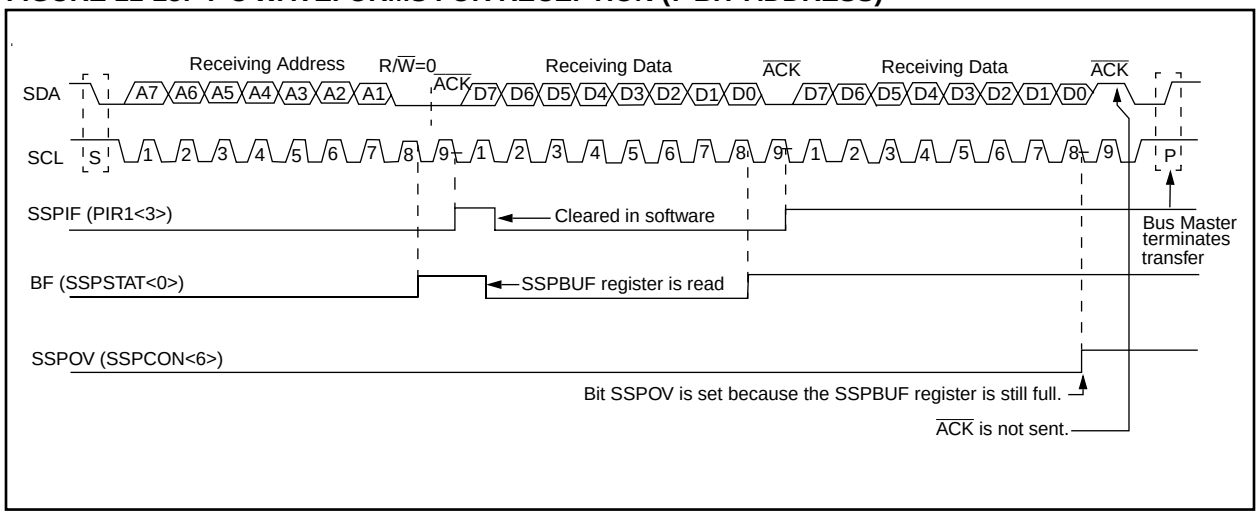
11.5.1.2 RECEPTION

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register.

When the address byte overflow condition exists, then no acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set or bit SSPOV (SSPCON<6>) is set.

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF (PIR1<3>) must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

**FIGURE 11-25: $I^2C$ WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**



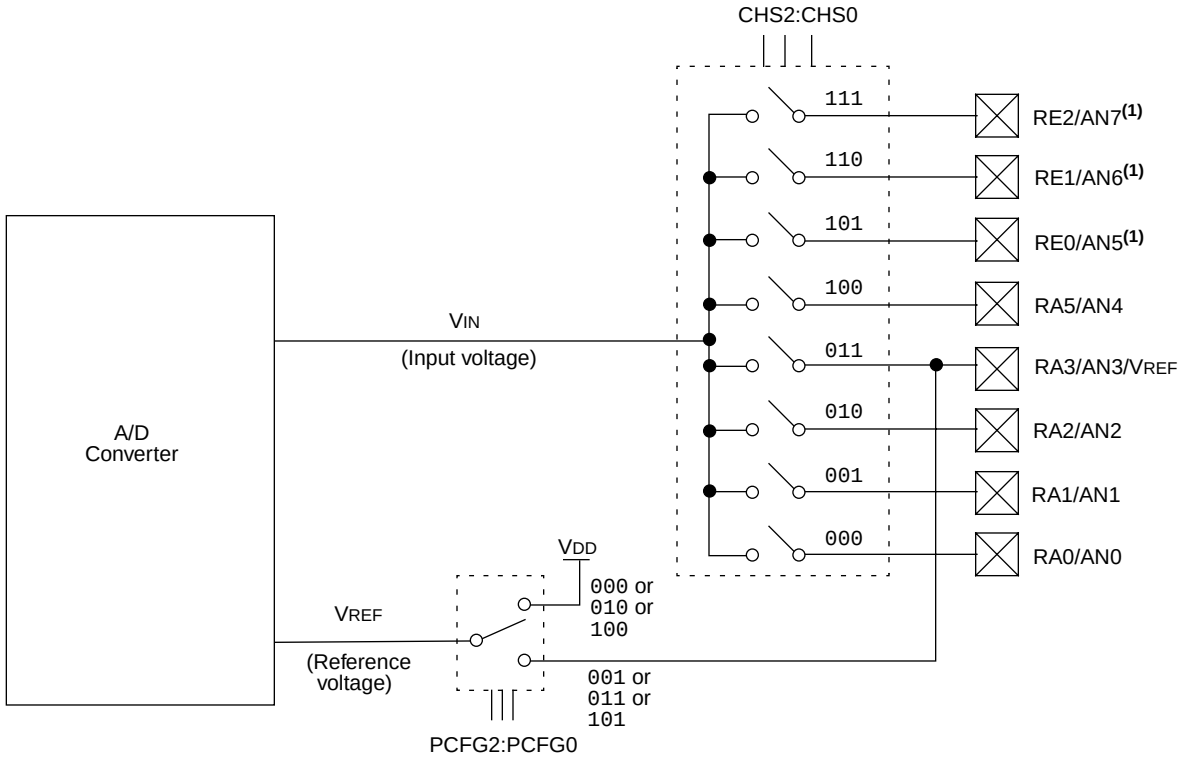
The ADRES register contains the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRES register, the GO/$\overline{\text{DONE}}$ bit (ADCON0<2>) is cleared, and A/D interrupt flag bit ADIF is set. The block diagrams of the A/D module are shown in Figure 13-3.

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 13.1. 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 TAD. A minimum wait of 2TAD is required before next acquisition starts.

**FIGURE 13-3: A/D BLOCK DIAGRAM**




Note 1: Not available on PIC16C72/73/73A/76.

**FIGURE 14-10: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO $V_{DD}$): CASE 1**

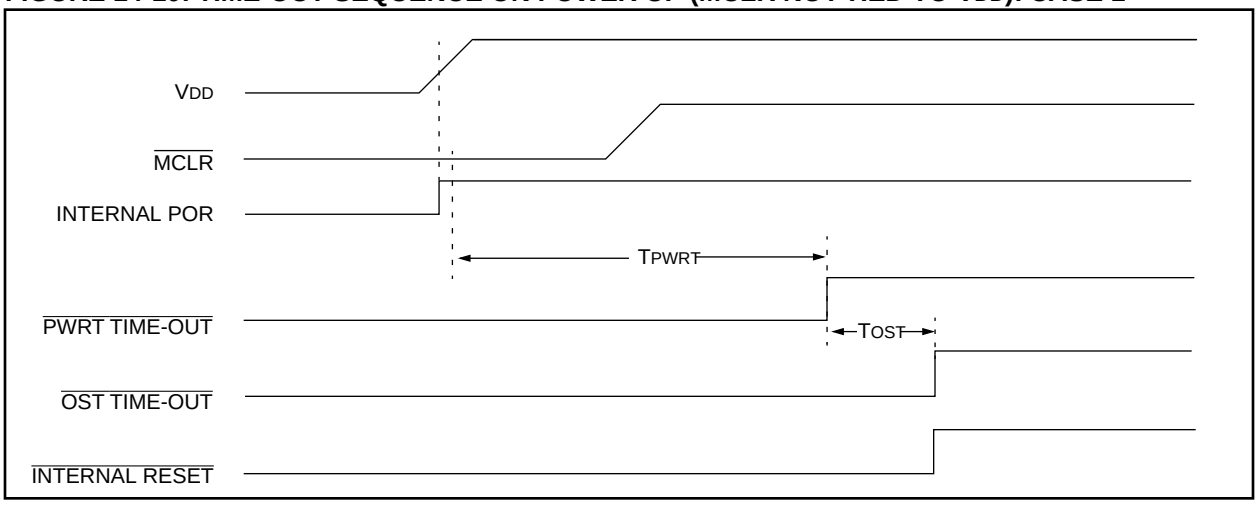



**FIGURE 14-11: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO $V_{DD}$): CASE 2**

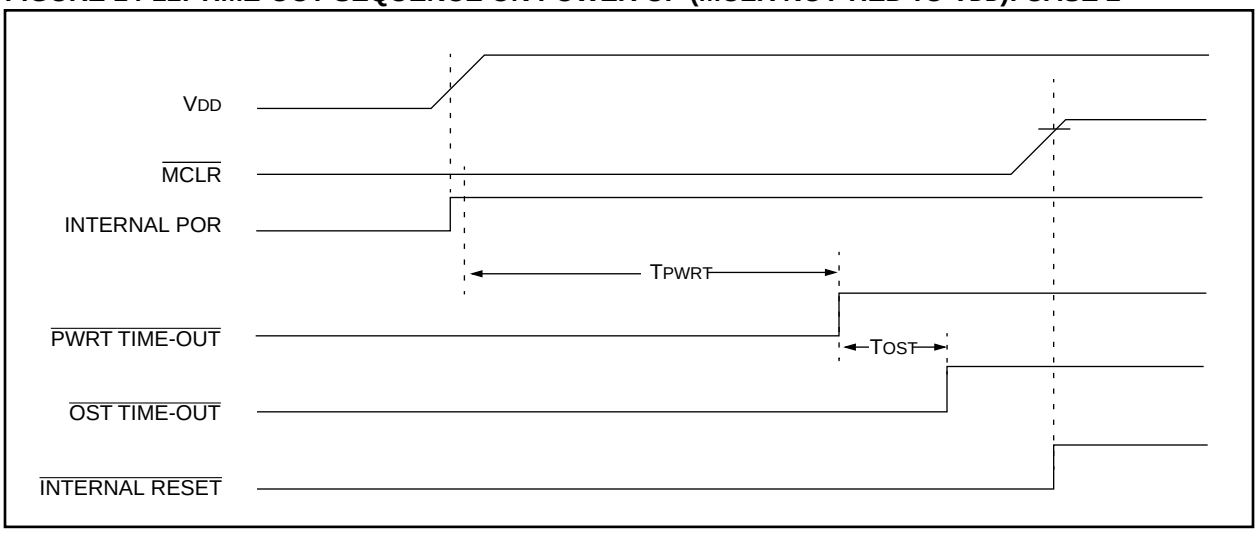



**FIGURE 14-12: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO $V_{DD}$)**

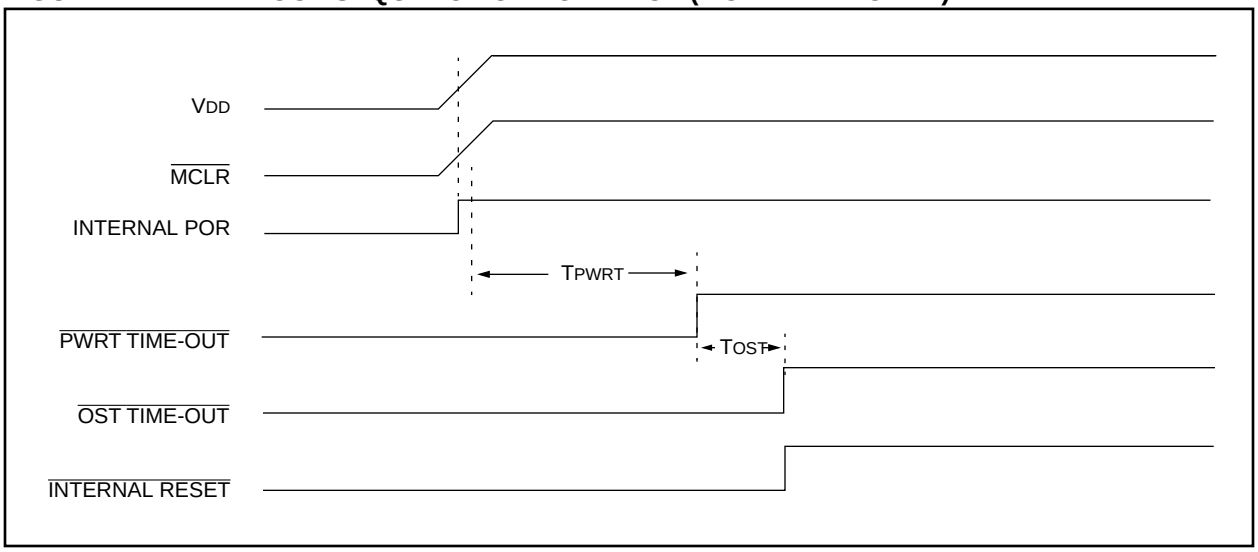
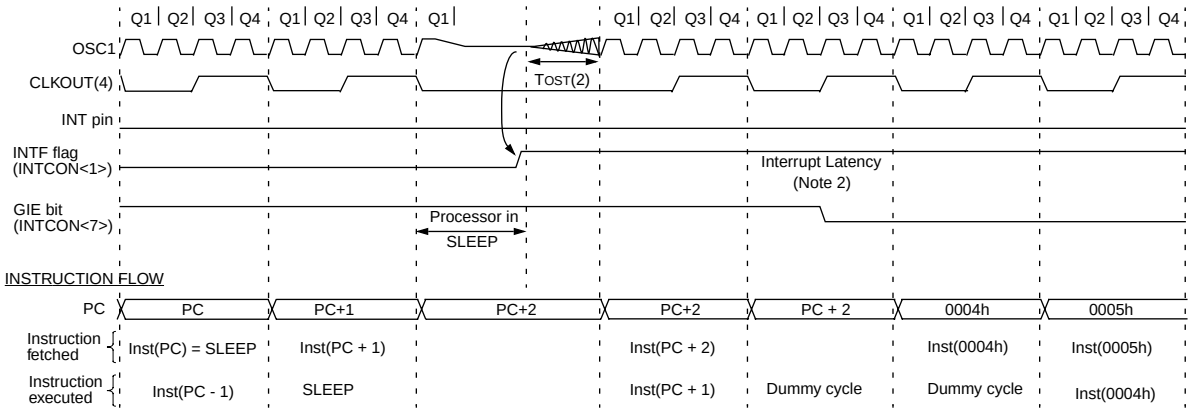


**FIGURE 14-20: WAKE-UP FROM SLEEP THROUGH INTERRUPT**




Note 1: XT, HS or LP oscillator mode assumed.
2: TOST = 1024TOSC (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.

## 14.9 Program Verification/Code Protection

| Applicable Devices | | | | | | |
|---|---|---|---|---|---|---|
| 72 | 73 | 73A | 74 | 74A | 76 | 77 |

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. |
|---|---|

## 14.10 ID Locations

| Applicable Devices | | | | | | |
|---|---|---|---|---|---|---|
| 72 | 73 | 73A | 74 | 74A | 76 | 77 |

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.

## 14.11 In-Circuit Serial Programming

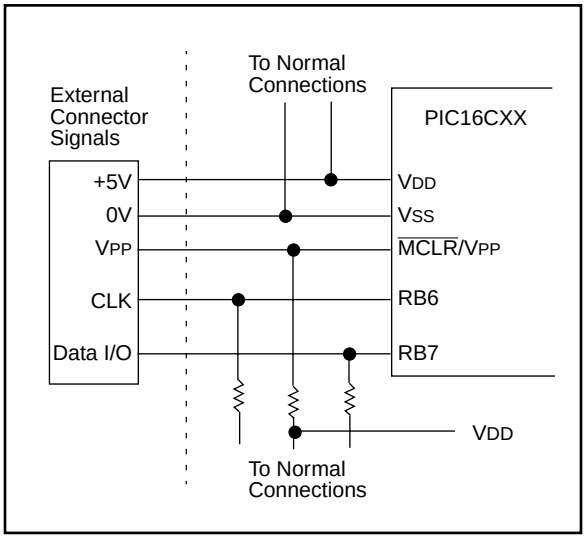
| Applicable Devices | | | | | | |
|---|---|---|---|---|---|---|
| 72 | 73 | 73A | 74 | 74A | 76 | 77 |

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 $\overline{MCLR}$ ($V_{PP}$) 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) 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 14-21: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**



**BTFSS** **Bit Test f, Skip if Set**

Syntax: [*label*] BTFSS f,b

Operands: 0 ≤ f ≤ 127
0 ≤ 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 | No-Operation |

If Skip: (2nd Cycle)

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

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 ≤ k ≤ 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 | No-Operation | No-Operation | No-Operation | No-Operation |

Example

```
HERE    CALL    THERE
```

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

**IORWF** **Inclusive OR W with f**

Syntax: [ *label* ] IORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .OR. (f) → (destination)

Status Affected: Z

Encoding:

| 00 | 0100 | dfff | ffff |
|---|---|---|---|

Description: Inclusive OR the W register with register 'f'. 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
```
IORWF   RESULT, 0
```
Before Instruction
RESULT = 0x13
W = 0x91
After Instruction
RESULT = 0x13
W = 0x93
Z = 1

**MOVF** **Move f**

Syntax: [ *label* ] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (destination)

Status Affected: Z

Encoding:

| 00 | 1000 | dfff | ffff |
|---|---|---|---|

Description: The contents of register f is moved to a destination dependant upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example
```
MOVF    FSR, 0
```
After Instruction
W = value in FSR register
Z = 1

**MOVLW** **Move Literal to W**

Syntax: [ *label* ] MOVLW k

Operands: $0 \le k \le 255$

Operation: k → (W)

Status Affected: None

Encoding:

| 11 | 00xx | kkkk | kkkk |
|---|---|---|---|

Description: The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example
```
MOVLW   0x5A
```
After Instruction
W = 0x5A

**MOVWF** **Move W to f**

Syntax: [ *label* ] MOVWF f

Operands: $0 \le f \le 127$

Operation: (W) → (f)

Status Affected: None

Encoding:

| 00 | 0000 | 1fff | ffff |
|---|---|---|---|

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example
```
MOVWF   OPTION_REG
```
Before Instruction
OPTION = 0xFF
W = 0x4F
After Instruction
OPTION = 0x4F
W = 0x4F

**TABLE 16-1: DEVELOPMENT TOOLS FROM MICROCHIP**

| | | PIC12C5XX | PIC14000 | PIC16C5X | PIC16CXXX | PIC16C6X | PIC16C7XX | PIC16C8X | PIC16C9XX | PIC17C4X | PIC17C75X | 24CXX 25CXX 93CXX | HCS200 HCS300 HCS301 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Emulator Products | PICMASTER®/ PICMASTER-CE In-Circuit Emulator | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | Available 3Q97 | | |
| | ICEPIC Low-Cost In-Circuit Emulator | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | |
| Software Tools | MPLAB™ Integrated Development Environment | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| | MPLAB™ C Compiler | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| | *fuzzy*TECH®-MP Explorer/Edition Fuzzy Logic Dev. Tool | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| | MP-DriveWay™ Applications Code Generator | | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | | | |
| | Total Endurance™ Software Model | | | | | | | | | | | ✓ | |
| Programmers | PICSTART® Lite Ultra Low-Cost Dev. Kit | | | ✓ | | ✓ | ✓ | ✓ | | | | | |
| | PICSTART® Plus Low-Cost Universal Dev. Kit | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| | PRO MATE® II Universal Programmer | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KEELOQ® Programmer | | | | | | | | | | | | ✓ |
| Demo Boards | SEEVAL® Designers Kit | | | | | | | | | | | ✓ | |
| | PICDEM-1 | | | ✓ | ✓ | | | ✓ | | ✓ | | | |
| | PICDEM-2 | | | | | ✓ | ✓ | | | | | | |
| | PICDEM-3 | | | | | | | | ✓ | | | | |
| | KEELOQ® Evaluation Kit | | | | | | | | | | | | ✓ |

| Applicable Devices | 72 | 73 | 73A | 74 | 74A | 76 | 77 |
|---|---|---|---|---|---|---|---|

### 18.4 Timing Parameter Symbology

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS
3. TCC:ST ($I^2C$ specifications only)
4. Ts ($I^2C$ specifications only)

| | | | |
|---|---|---|---|
| **T** | | | |
| F | Frequency | T | Time |

Lowercase letters (pp) and their meanings:

| | | | |
|---|---|---|---|
| **pp** | | | |
| cc | CCP1 | osc | OSC1 |
| ck | CLKOUT | rd | $\overline{RD}$ |
| cs | $\overline{CS}$ | rw | $\overline{RD}$ or $\overline{WR}$ |
| di | SDI | sc | SCK |
| do | SDO | ss | $\overline{SS}$ |
| dt | Data in | t0 | T0CKI |
| io | I/O port | t1 | T1CKI |
| mc | $\overline{MCLR}$ | wr | $\overline{WR}$ |

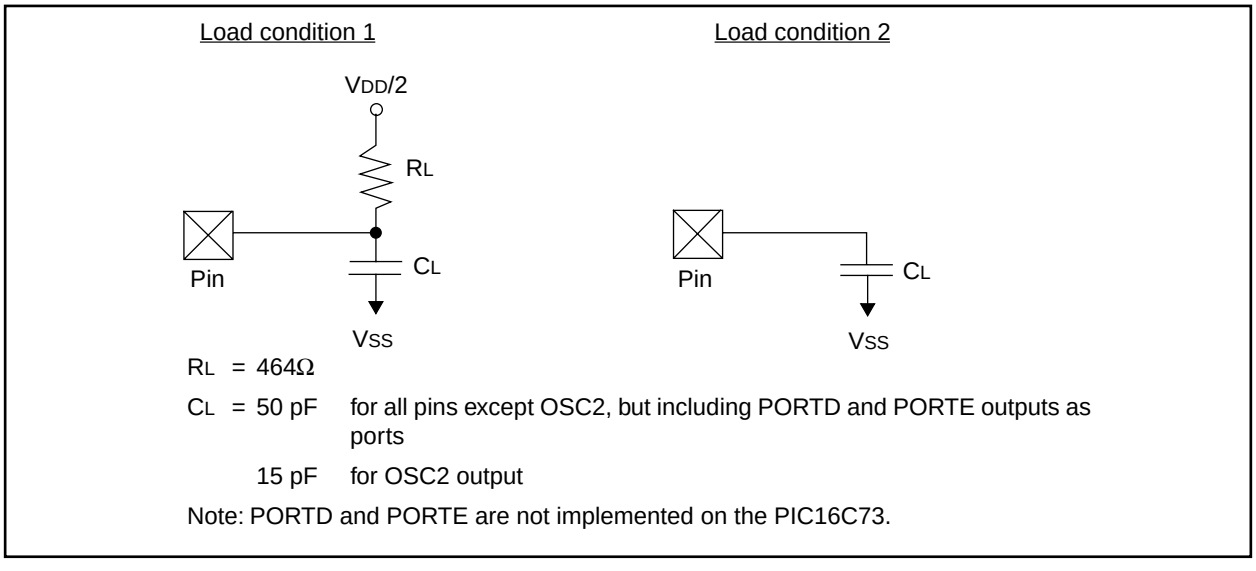
Uppercase letters and their meanings:

| | | | |
|---|---|---|---|
| **S** | | | |
| F | Fall | P | Period |
| H | High | R | Rise |
| I | Invalid (Hi-impedance) | V | Valid |
| L | Low | Z | Hi-impedance |
| **$I^2C$ only** | | | |
| AA | output access | High | High |
| BUF | Bus free | Low | Low |

TCC:ST ($I^2C$ specifications only)

| | | | |
|---|---|---|---|
| **CC** | | | |
| HD | Hold | SU | Setup |
| **ST** | | | |
| DAT | DATA input hold | STO | STOP condition |
| STA | START condition | | |

**FIGURE 18-1: LOAD CONDITIONS**




$R_L$ = 464Ω

$C_L$ = 50 pF for all pins except OSC2, but including PORTD and PORTE outputs as ports

15 pF for OSC2 output

Note: PORTD and PORTE are not implemented on the PIC16C73.

| Applicable Devices | 72 | 73 | 73A | 74 | 74A | 76 | 77 |
|---|---|---|---|---|---|---|---|

### 19.1 DC Characteristics: PIC16C73A/74A-04 (Commercial, Industrial, Extended) PIC16C73A/74A-10 (Commercial, Industrial, Extended) PIC16C73A/74A-20 (Commercial, Industrial, Extended)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C for extended,<br>-40°C ≤ TA ≤ +85°C for industrial and<br>0°C ≤ TA ≤ +70°C for commercial | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Characteristic** | **Sym** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| D001<br>D001A | Supply Voltage | VDD | 4.0<br>4.5 | -<br>- | 6.0<br>5.5 | V<br>V | XT, RC and LP osc configuration<br>HS osc configuration |
| D002* | RAM Data Retention Voltage (Note 1) | VDR | - | 1.5 | - | V | |
| 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<br>3.7 | 4.0<br>4.0 | 4.3<br>4.4 | V<br>V | BODEN bit in configuration word enabled<br>Extended Range Only |
| D010<br><br><br>D013<br><br>D015* | Supply Current (Note 2,5)<br><br><br>Brown-out Reset Current (Note 6) | IDD<br><br><br><br>ΔIBOR | -<br><br><br>-<br><br>- | 2.7<br><br><br>10<br><br>350 | 5<br><br><br>20<br><br>425 | mA<br><br><br>mA<br><br>µA | XT, RC osc configuration<br>FOSC = 4 MHz, VDD = 5.5V (Note 4)<br><br>HS osc configuration<br>FOSC = 20 MHz, VDD = 5.5V<br><br>BOR enabled VDD = 5.0V |
| D020<br>D021<br>D021A<br>D021B<br><br>D023* | Power-down Current (Note 3,5)<br><br><br><br><br>Brown-out Reset Current (Note 6) | IPD<br><br><br><br><br>ΔIBOR | -<br>-<br>-<br>-<br><br>- | 10.5<br>1.5<br>1.5<br>2.5<br><br>350 | 42<br>16<br>19<br>19<br><br>425 | µA<br>µA<br>µA<br>µA<br><br>µA | VDD = 4.0V, WDT enabled, -40°C to +85°C<br>VDD = 4.0V, WDT disabled, -0°C to +70°C<br>VDD = 4.0V, WDT disabled, -40°C to +85°C<br>VDD = 4.0V, WDT disabled, -40°C to +125°C<br><br>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 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
$\overline{MCLR}$ = 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: Timer1 oscillator (when enabled) adds approximately 20 µA to the specification. This value is from characterization and is for design guidance only. This is not tested.

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

| Applicable Devices | 72 | 73 | 73A | 74 | 74A | 76 | 77 |
|---|---|---|---|---|---|---|---|

**FIGURE 21-3: TYPICAL IPD vs. VDD @ 25°C (WDT ENABLED, RC MODE)**

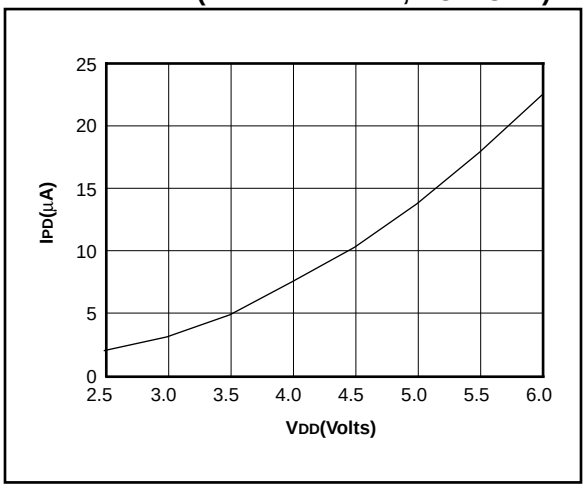



**FIGURE 21-4: MAXIMUM IPD vs. VDD (WDT ENABLED, RC MODE)**

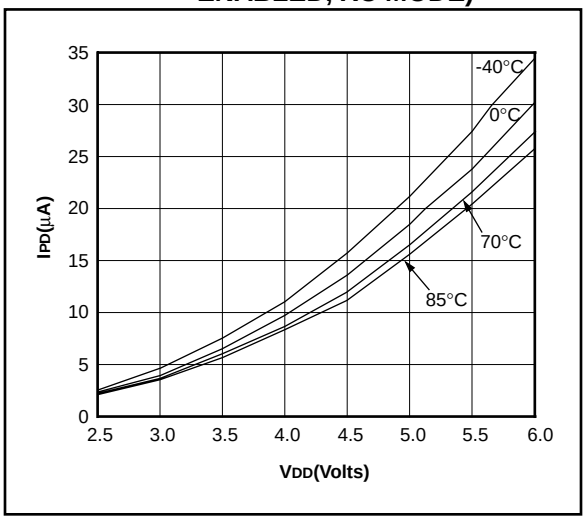
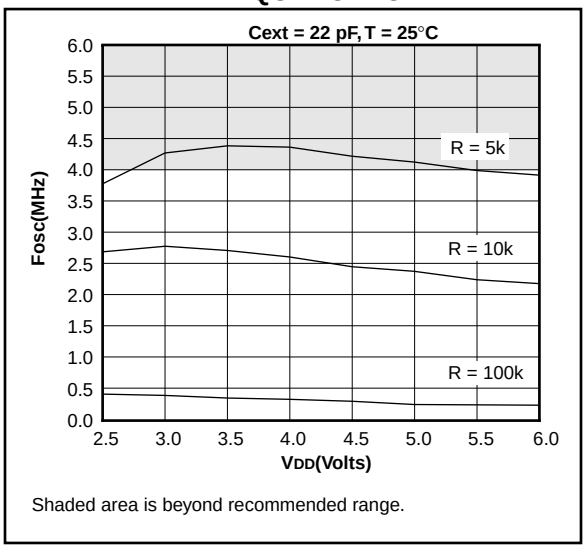



**FIGURE 21-5: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD**




Shaded area is beyond recommended range.

**FIGURE 21-6: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD**

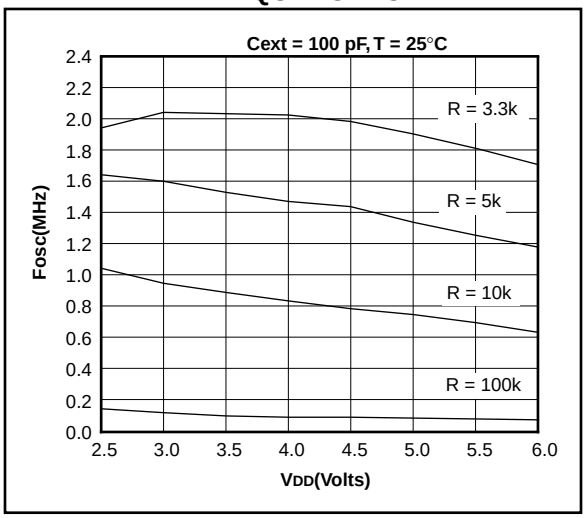
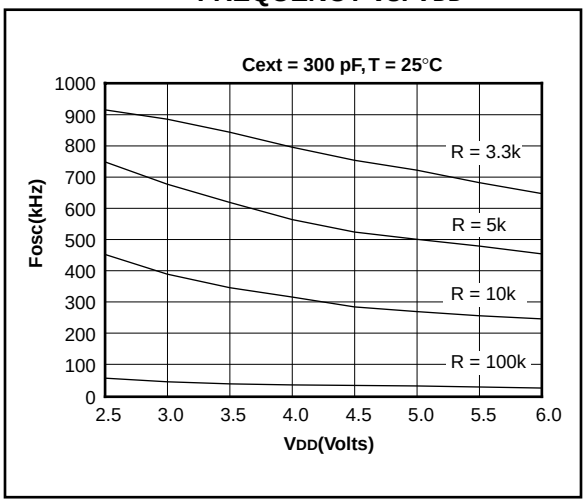



**FIGURE 21-7: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD**




Data based on matrix samples. See first page of this section for details.

## E.10 PIC17CXXX Family of Devices

| | | PIC17C42A | PIC17CR42 | PIC17C43 | PIC17CR43 | PIC17C44 |
|---|---|---|---|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 33 | 33 | 33 | 33 | 33 |
| **Memory** | EPROM Program Memory (words) | 2K | — | 4K | — | 8K |
| | ROM Program Memory (words) | — | 2K | — | 4K | — |
| | RAM Data Memory (bytes) | 232 | 232 | 454 | 454 | 454 |
| **Peripherals** | Timer Module(s) | TMR0, TMR1, TMR2, TMR3 | TMR0, TMR1, TMR2, TMR3 | TMR0, TMR1, TMR2, TMR3 | TMR0, TMR1, TMR2, TMR3 | TMR0, TMR1, TMR2, TMR3 |
| | Captures/PWM Module(s) | 2 | 2 | 2 | 2 | 2 |
| | Serial Port(s) (USART) | Yes | Yes | Yes | Yes | Yes |
| **Features** | Hardware Multiply | Yes | Yes | Yes | Yes | Yes |
| | External Interrupts | Yes | Yes | Yes | Yes | Yes |
| | Interrupt Sources | 11 | 11 | 11 | 11 | 11 |
| | I/O Pins | 33 | 33 | 33 | 33 | 33 |
| | Voltage Range (Volts) | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 |
| | Number of Instructions | 58 | 58 | 58 | 58 | 58 |
| | Packages | 40-pin DIP; 44-pin PLCC, MQFP, TQFP | 40-pin DIP; 44-pin PLCC, MQFP, TQFP | 40-pin DIP; 44-pin PLCC, MQFP, TQFP | 40-pin DIP; 44-pin PLCC, MQFP, TQFP | 40-pin DIP; 44-pin PLCC, MQFP, TQFP |

| | | PIC17C752 | PIC17C756 |
|---|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 33 | 33 |
| **Memory** | EPROM Program Memory (words) | 8K | 16K |
| | ROM Program Memory (words) | — | — |
| | RAM Data Memory (bytes) | 454 | 902 |
| **Peripherals** | Timer Module(s) | TMR0, TMR1, TMR2, TMR3 | TMR0, TMR1, TMR2, TMR3 |
| | Captures/PWM Module(s) | 4/3 | 4/3 |
| | Serial Port(s) (USART) | 2 | 2 |
| **Features** | Hardware Multiply | Yes | Yes |
| | External Interrupts | Yes | Yes |
| | Interrupt Sources | 18 | 18 |
| | I/O Pins | 50 | 50 |
| | Voltage Range (Volts) | 3.0-6.0 | 3.0-6.0 |
| | Number of Instructions | 58 | 58 |
| | Packages | 64-pin DIP; 68-pin LCC, 68-pin TQFP | 64-pin DIP; 68-pin LCC, 68-pin TQFP |

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

## PIN COMPATIBILITY

Devices that have the same package type and VDD, VSS and $\overline{MCLR}$ pin locations are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only requires minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

**TABLE E-1: PIN COMPATIBLE DEVICES**

| Pin Compatible Devices | Package |
|---|---|
| PIC12C508, PIC12C509, PIC12C671, PIC12C672 | 8-pin |
| PIC16C154, PIC16CR154, PIC16C156, PIC16CR156, PIC16C158, PIC16CR158, PIC16C52, PIC16C54, PIC16C54A, PIC16CR54A, PIC16C56, PIC16C58A, PIC16CR58A, PIC16C61, PIC16C554, PIC16C556, PIC16C558 PIC16C620, PIC16C621, PIC16C622 PIC16C641, PIC16C642, PIC16C661, PIC16C662 PIC16C710, PIC16C71, PIC16C711, PIC16C715 PIC16F83, PIC16CR83, PIC16F84A, PIC16CR84 | 18-pin, 20-pin |
| PIC16C55, PIC16C57, PIC16CR57B | 28-pin |
| PIC16CR62, PIC16C62A, PIC16C63, PIC16CR63, PIC16C66, PIC16C72, PIC16C73A, PIC16C76 | 28-pin |
| PIC16CR64, PIC16C64A, PIC16C65A, PIC16CR65, PIC16C67, PIC16C74A, PIC16C77 | 40-pin |
| PIC17CR42, PIC17C42A, PIC17C43, PIC17CR43, PIC17C44 | 40-pin |
| PIC16C923, PIC16C924 | 64/68-pin |
| PIC17C756, PIC17C752 | 64/68-pin |