# PIC16C7X

## 8-Bit CMOS Microcontrollers with A/D Converter

### Devices included in this data sheet:

- PIC16C72
- PIC16C73
- PIC16C73A
- PIC16C74
- PIC16C74A
- PIC16C76
- PIC16C77

### PIC16C7X Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K x 14 words of Program Memory, up to 368 x 8 bytes of Data Memory (RAM)
- Interrupt capability
- Eight level deep hardware stack
- Direct, indirect, and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS EPROM technology
- Fully static design
- Wide operating voltage range: 2.5V to 6.0V
- High Sink/Source Current 25/25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - < 2 mA @ 5V, 4 MHz
  - 15 μA typical @ 3V, 32 kHz
  - < 1 μA typical standby current

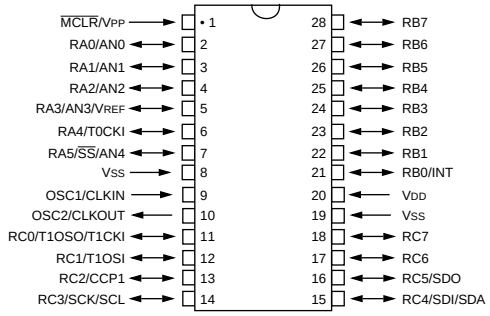
### PIC16C7X Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Capture, Compare, PWM module(s)
- Capture is 16-bit, max. resolution is 12.5 ns, Compare is 16-bit, max. resolution is 200 ns, PWM max. resolution is 10-bit
- 8-bit multichannel analog-to-digital converter
- Synchronous Serial Port (SSP) with SPI™ and I²C™
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI)
- Parallel Slave Port (PSP) 8-bits wide, with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls
- Brown-out detection circuitry for Brown-out Reset (BOR)

| PIC16C7X Features | 72 | 73 | 73A | 74 | 74A | 76 | 77 |
|---|---|---|---|---|---|---|---|
| Program Memory (EPROM) x 14 | 2K | 4K | 4K | 4K | 4K | 8K | 8K |
| Data Memory (Bytes) x 8 | 128 | 192 | 192 | 192 | 192 | 368 | 368 |
| I/O Pins | 22 | 22 | 22 | 33 | 33 | 22 | 33 |
| Parallel Slave Port | — | — | — | Yes | Yes | — | Yes |
| Capture/Compare/PWM Modules | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| Timer Modules | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| A/D Channels | 5 | 5 | 5 | 8 | 8 | 5 | 8 |
| Serial Communication | SPI/I²C | SPI/I²C, USART | SPI/I²C, USART | SPI/I²C, USART | SPI/I²C, USART | SPI/I²C, USART | SPI/I²C, USART |
| In-Circuit Serial Programming | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Brown-out Reset | Yes | — | Yes | — | Yes | Yes | Yes |
| Interrupt Sources | 8 | 11 | 11 | 12 | 12 | 11 | 12 |

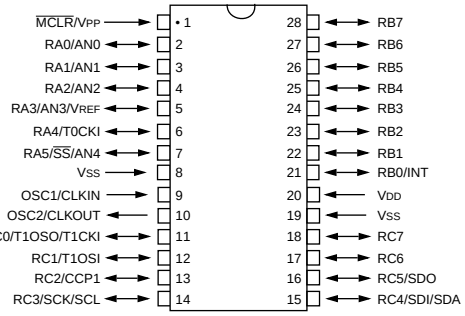
## Pin Diagrams

### SDIP, SOIC, Windowed Side Brazed Ceramic



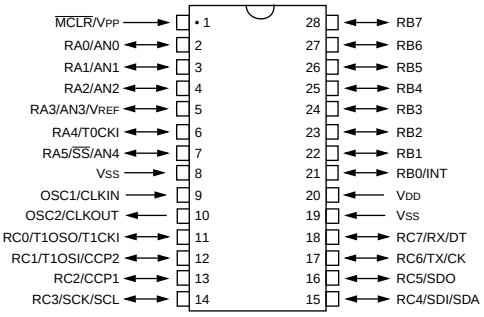

**PIC16C72**

### SSOP



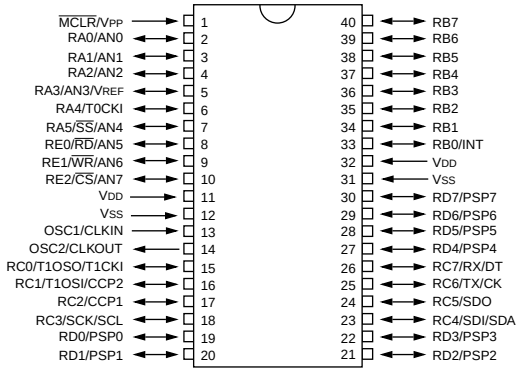

**PIC16C72**

### SDIP, SOIC, Windowed Side Brazed Ceramic




**PIC16C73**
**PIC16C73A**
**PIC16C76**

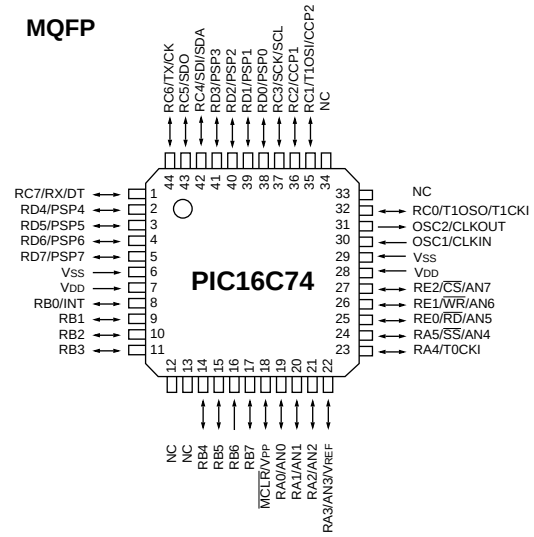
### PDIP, Windowed CERDIP



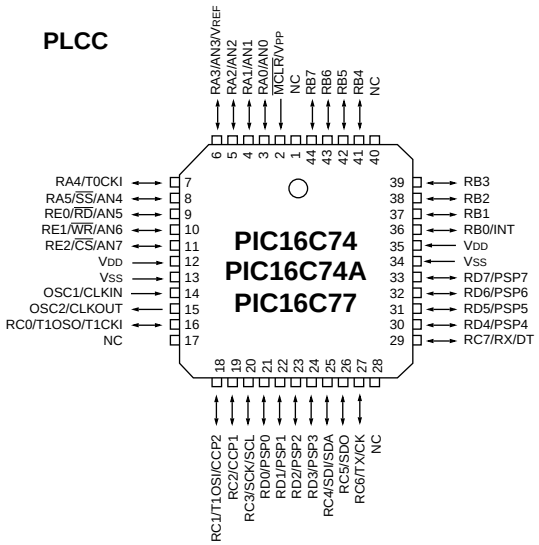

**PIC16C74**
**PIC16C74A**
**PIC16C77**

**Pin Diagrams (Cont.'d)**

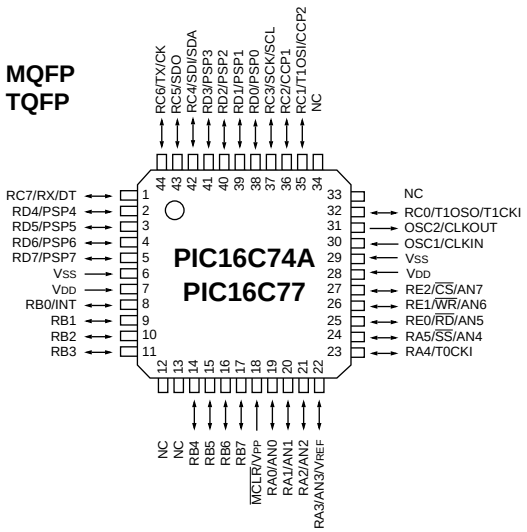
**MQFP**




**PLCC**




**MQFP**
**TQFP**



## Table of Contents



For register and module descriptions in this data sheet, device legends show which devices apply to those sections. As an example, the legend below would mean that the following section applies only to the PIC16C72, PIC16C73A and PIC16C74A devices.

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

## To Our Valued Customers

We constantly strive to improve the quality of all our products and documentation. We have spent an exceptional amount of time to ensure that these documents are correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please use the reader response form in the back of this data sheet to inform us. We appreciate your assistance in making this a better document.

## 1.0 GENERAL DESCRIPTION

The PIC16C7X is a family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers with integrated analog-to-digital (A/D) converters, in the PIC16CXX mid-range family.

All PIC16/17 microcontrollers employ an advanced RISC architecture. The PIC16CXX microcontroller family has enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with the separate 8-bit wide data. The two stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches which require two cycles. A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16CXX microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

The **PIC16C72** has 128 bytes of RAM and 22 I/O pins. In addition several peripheral features are available including: three timer/counters, one Capture/Compare/PWM module and one serial port. The Synchronous Serial Port can be configured as either a 3-wire Serial Peripheral Interface (SPI) or the two-wire Inter-Integrated Circuit ($I^2C$) bus. Also a 5-channel high-speed 8-bit A/D is provided. The 8-bit resolution is ideally suited for applications requiring low-cost analog interface, e.g. thermostat control, pressure sensing, etc.

The **PIC16C73/73A** devices have 192 bytes of RAM, while the **PIC16C76** has 368 byes of RAM. Each device has 22 I/O pins. In addition, several peripheral features are available including: three timer/counters, two Capture/Compare/PWM modules and two serial ports. The Synchronous Serial Port can be configured as either a 3-wire Serial Peripheral Interface (SPI) or the two-wire Inter-Integrated Circuit ($I^2C$) bus. The Universal Synchronous Asynchronous Receiver Transmitter (USART) is also known as the Serial Communications Interface or SCI. Also a 5-channel high-speed 8-bit A/D is provided.The 8-bit resolution is ideally suited for applications requiring low-cost analog interface, e.g. thermostat control, pressure sensing, etc.

The **PIC16C74/74A** devices have 192 bytes of RAM, while the **PIC16C77** has 368 bytes of RAM. Each device has 33 I/O pins. In addition several peripheral features are available including: three timer/counters, two Capture/Compare/PWM modules and two serial ports. The Synchronous Serial Port can be configured as either a 3-wire Serial Peripheral Interface (SPI) or the two-wire Inter-Integrated Circuit ($I^2C$) bus. The Universal Synchronous Asynchronous Receiver Transmitter (USART) is also known as the Serial Communications Interface or SCI. An 8-bit Parallel Slave Port is provided. Also an 8-channel high-speed 8-bit A/D is provided. The 8-bit resolution is ideally suited for applications requiring low-cost analog interface, e.g. thermostat control, pressure sensing, etc.

The PIC16C7X family has special features to reduce external components, thus reducing cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals. The SLEEP (power-down) feature provides a power saving mode. The user can wake up the chip from SLEEP through several external and internal interrupts and resets.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock-up.

A UV erasable CERDIP packaged version is ideal for code development while the cost-effective One-Time-Programmable (OTP) version is suitable for production in any volume.

The PIC16C7X family fits perfectly in applications ranging from security and remote sensors to appliance control and automotive. The EPROM technology makes customization of application programs (transmitter codes, motor speeds, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low cost, low power, high performance, ease of use and I/O flexibility make the PIC16C7X very versatile even in areas where no microcontroller use has been considered before (e.g. timer functions, serial communication, capture and compare, PWM functions and coprocessor applications).

### 1.1 Family and Upward Compatibility

Users familiar with the PIC16C5X microcontroller family will realize that this is an enhanced version of the PIC16C5X architecture. Please refer to Appendix A for a detailed list of enhancements. Code written for the PIC16C5X can be easily ported to the PIC16CXX family of devices (Appendix B).

### 1.2 Development Support

PIC16C7X devices are supported by the complete line of Microchip Development tools.

Please refer to Section 16.0 for more details about Microchip's development tools.

**TABLE 1-1: PIC16C7XX FAMILY OF DEVCES**

| | | PIC16C710 | PIC16C71 | PIC16C711 | PIC16C715 | PIC16C72 | PIC16CR72(1) |
|---|---|---|---|---|---|---|---|
| Clock | Maximum Frequency of Operation (MHz) | 20 | 20 | 20 | 20 | 20 | 20 |
| Memory | EPROM Program Memory (x14 words) | 512 | 1K | 1K | 2K | 2K | — |
| | ROM Program Memory (14K words) | — | — | — | — | — | 2K |
| | Data Memory (bytes) | 36 | 36 | 68 | 128 | 128 | 128 |
| Peripherals | Timer Module(s) | TMR0 | TMR0 | TMR0 | TMR0 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 |
| | Capture/Compare/ PWM Module(s) | — | — | — | — | 1 | 1 |
| | Serial Port(s) (SPI/$I^2C$, USART) | — | — | — | — | SPI/$I^2C$ | SPI/$I^2C$ |
| | Parallel Slave Port | — | — | — | — | — | — |
| | A/D Converter (8-bit) Channels | 4 | 4 | 4 | 4 | 5 | 5 |
| Features | Interrupt Sources | 4 | 4 | 4 | 4 | 8 | 8 |
| | I/O Pins | 13 | 13 | 13 | 13 | 22 | 22 |
| | Voltage Range (Volts) | 3.0-6.0 | 3.0-6.0 | 3.0-6.0 | 3.0-5.5 | 2.5-6.0 | 3.0-5.5 |
| | In-Circuit Serial Programming | Yes | Yes | Yes | Yes | Yes | Yes |
| | Brown-out Reset | Yes | — | Yes | Yes | Yes | Yes |
| | Packages | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP | 28-pin SDIP, SOIC, SSOP | 28-pin SDIP, SOIC, SSOP |

| | | PIC16C73A | PIC16C74A | PIC16C76 | PIC16C77 |
|---|---|---|---|---|---|
| Clock | Maximum Frequency of Operation (MHz) | 20 | 20 | 20 | 20 |
| Memory | EPROM Program Memory (x14 words) | 4K | 4K | 8K | 8K |
| | Data Memory (bytes) | 192 | 192 | 368 | 368 |
| Peripherals | Timer Module(s) | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 |
| | Capture/Compare/PWM Module(s) | 2 | 2 | 2 | 2 |
| | Serial Port(s) (SPI/$I^2C$, USART) | SPI/$I^2C$, USART | SPI/$I^2C$, USART | SPI/$I^2C$, USART | SPI/$I^2C$, USART |
| | Parallel Slave Port | — | Yes | — | Yes |
| | A/D Converter (8-bit) Channels | 5 | 8 | 5 | 8 |
| Features | Interrupt Sources | 11 | 12 | 11 | 12 |
| | I/O Pins | 22 | 33 | 22 | 33 |
| | Voltage Range (Volts) | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 |
| | In-Circuit Serial Programming | Yes | Yes | Yes | Yes |
| | Brown-out Reset | Yes | Yes | Yes | Yes |
| | Packages | 28-pin SDIP, SOIC | 40-pin DIP; 44-pin PLCC, MQFP, TQFP | 28-pin SDIP, SOIC | 40-pin DIP; 44-pin PLCC, MQFP, TQFP |

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C7XX Family devices use serial programming with clock pin RB6 and data pin RB7.

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

## 2.0 PIC16C7X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in the PIC16C7X Product Identification System section at the end of this data sheet. When placing orders, please use that page of the data sheet to specify the correct part number.

For the PIC16C7X family, there are two device “types” as indicated in the device number:

1. **C**, as in PIC16**C**74. These devices have EPROM type memory and operate over the standard voltage range.
2. **LC**, as in PIC16**LC**74. These devices have EPROM type memory and operate over an extended voltage range.

### 2.1 UV Erasable Devices

The UV erasable version, offered in CERDIP package is optimal for prototype development and pilot programs. This version can be erased and reprogrammed to any of the oscillator modes.

Microchip's PICSTART® Plus and PRO MATE® II programmers both support programming of the PIC16C7X.

### 2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers who need the flexibility for frequent code updates and small volume applications.

The OTP devices, packaged in plastic packages, permit the user to program them once. In addition to the program memory, the configuration bits must also be programmed.

### 2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

### 2.4 Serialized Quick-Turnaround Production (SQTP℠) Devices

Microchip offers a unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random, or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password, or ID number.

**NOTES:**

## 3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16CXX family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16CXX uses a Harvard architecture, in which, program and data are accessed from separate memories using separate buses. This improves bandwidth over traditional von Neumann architecture in which program and data are fetched from the same memory using the same bus. Separating program and data buses further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 14-bits wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions (Example 3-1). Consequently, all instructions (35) execute in a single cycle (200 ns @ 20 MHz) except for program branches.

The table below lists program memory (EPROM) and data memory (RAM) for each PIC16C7X device.

| Device | Program Memory | Data Memory |
|---|---|---|
| PIC16C72 | 2K x 14 | 128 x 8 |
| PIC16C73 | 4K x 14 | 192 x 8 |
| PIC16C73A | 4K x 14 | 192 x 8 |
| PIC16C74 | 4K x 14 | 192 x 8 |
| PIC16C74A | 4K x 14 | 192 x 8 |
| PIC16C76 | 8K x 14 | 368 x 8 |
| PIC16C77 | 8K x 14 | 386 x 8 |

The PIC16CXX can directly or indirectly address its register files or data memory. All special function registers, including the program counter, are mapped in the data memory. The PIC16CXX has an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of ‘special optimal situations’ make programming with the PIC16CXX simple yet efficient. In addition, the learning curve is reduced significantly.

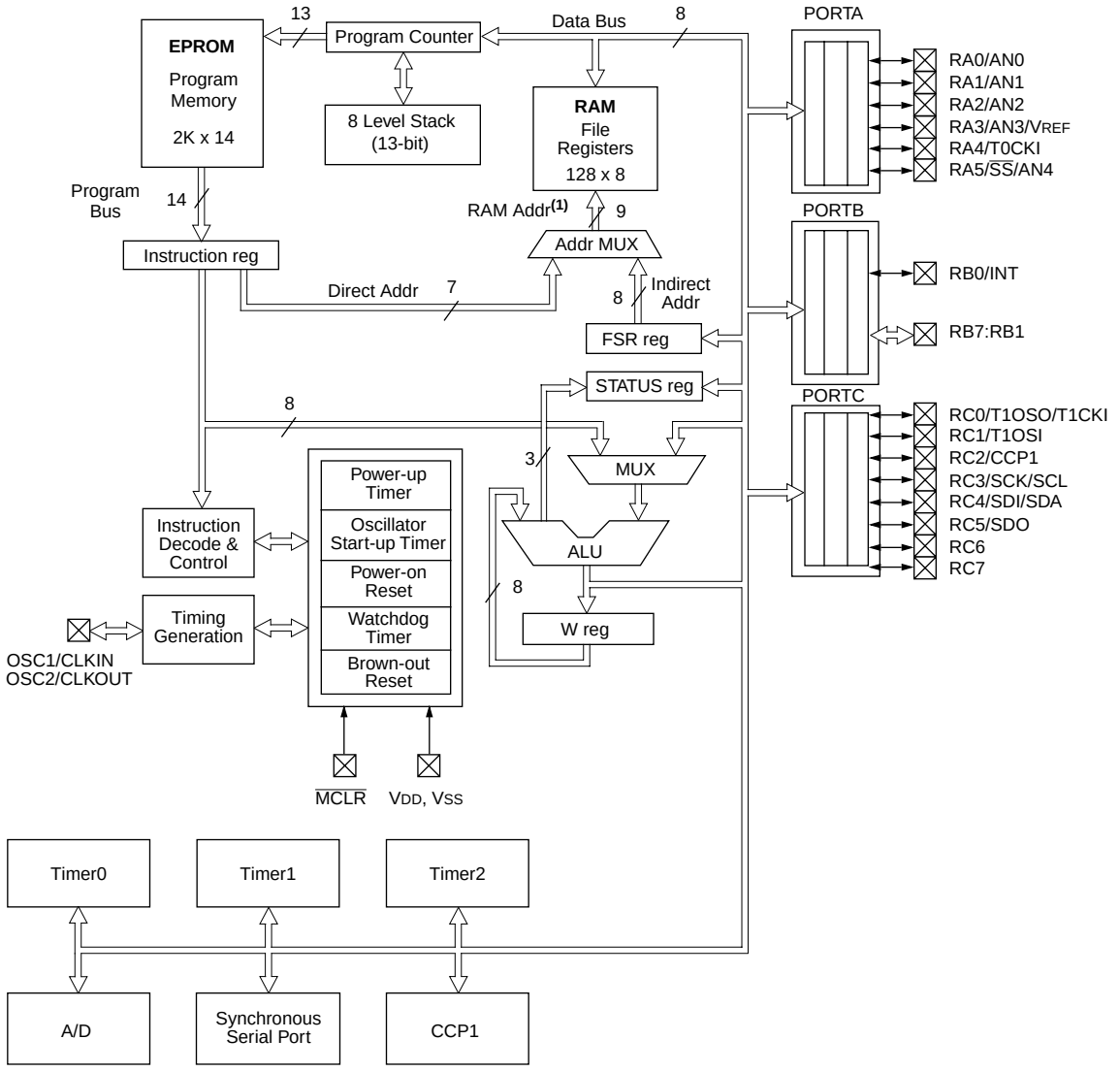
PIC16CXX devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between the data in the working register and any register file.

The ALU is 8-bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ bit and a $\overline{\text{digit}}$ $\overline{\text{borrow}}$ out bit, respectively, in subtraction. See the SUBLW and SUBWF instructions for examples.

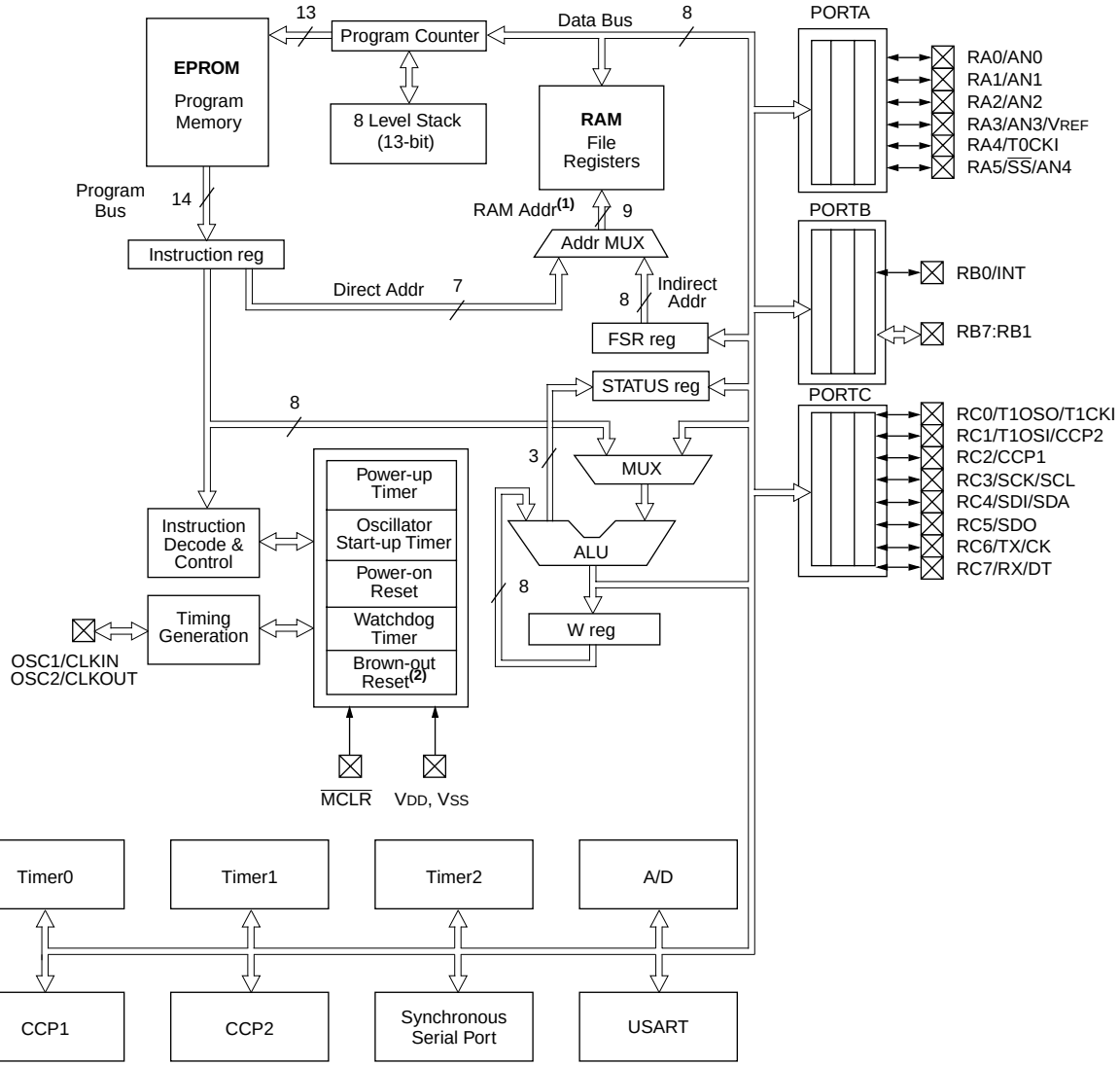
**FIGURE 3-1: PIC16C72 BLOCK DIAGRAM**




Note 1: Higher order bits are from the STATUS register.

**FIGURE 3-2: PIC16C73/73A/76 BLOCK DIAGRAM**

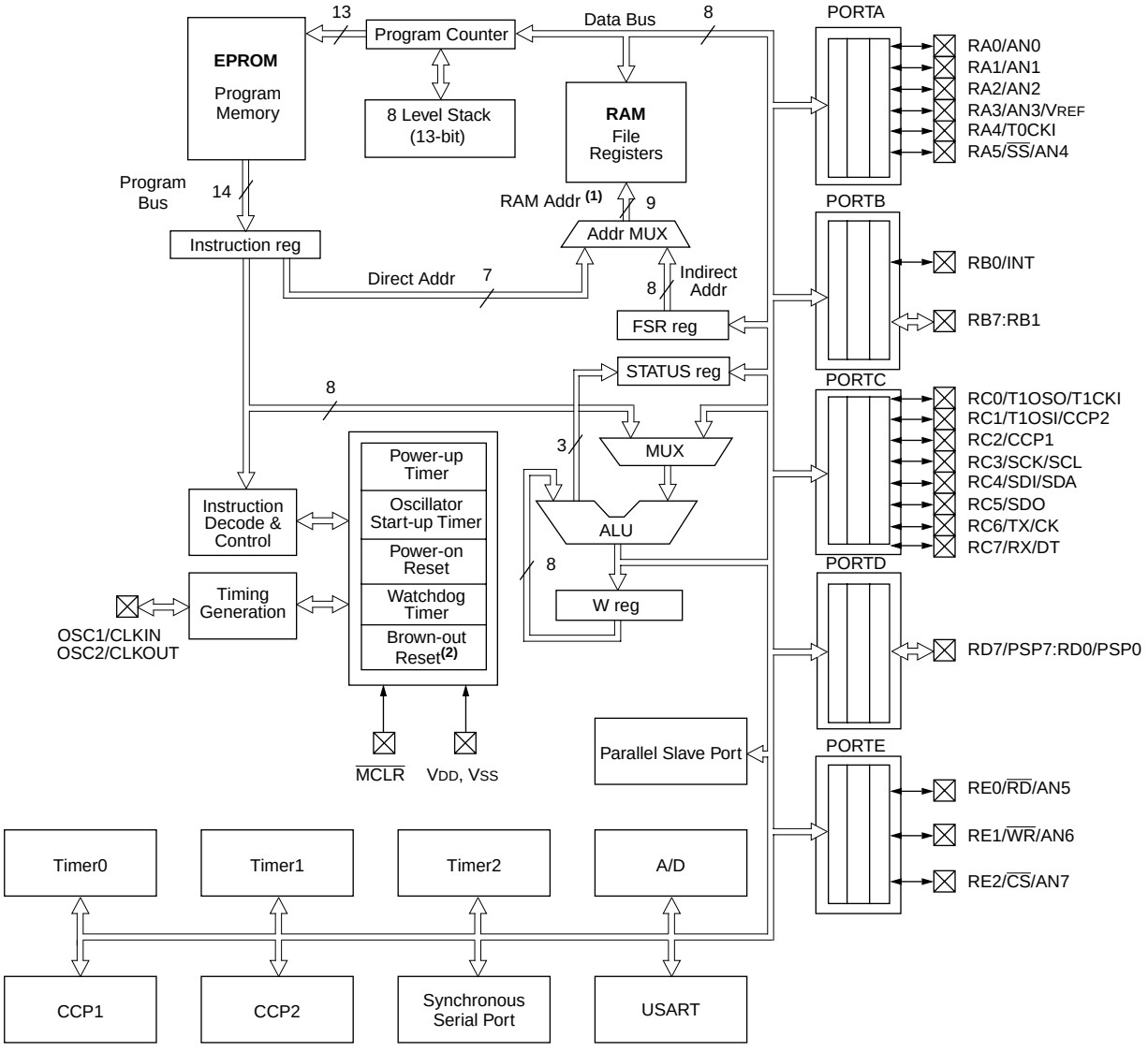
| Device | Program Memory | Data Memory (RAM) |
|---|---|---|
| PIC16C73 | 4K x 14 | 192 x 8 |
| PIC16C73A | 4K x 14 | 192 x 8 |
| PIC16C76 | 8K x 14 | 368 x 8 |




Note 1: Higher order bits are from the STATUS register.
2: Brown-out Reset is not available on the PIC16C73.

**FIGURE 3-3: PIC16C74/74A/77 BLOCK DIAGRAM**

| Device | Program Memory | Data Memory (RAM) |
|---|---|---|
| PIC16C74 | 4K x 14 | 192 x 8 |
| PIC16C74A | 4K x 14 | 192 x 8 |
| PIC16C77 | 8K x 14 | 368 x 8 |




Note 1: Higher order bits are from the STATUS register.
2: Brown-out Reset is not available on the PIC16C74.

**TABLE 3-1: PIC16C72 PINOUT DESCRIPTION**

| Pin Name | DIP Pin# | SSOP Pin# | SOIC Pin# | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|---|
| OSC1/CLKIN | 9 | 9 | 9 | I | ST/CMOS(3) | Oscillator crystal input/external clock source input. |
| OSC2/CLKOUT | 10 | 10 | 10 | O | — | Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate. |
| $\overline{MCLR}$/VPP | 1 | 1 | 1 | 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 | 2 | 2 | I/O | TTL | RA0 can also be analog input0 |
| RA1/AN1 | 3 | 3 | 3 | I/O | TTL | RA1 can also be analog input1 |
| RA2/AN2 | 4 | 4 | 4 | I/O | TTL | RA2 can also be analog input2 |
| RA3/AN3/VREF | 5 | 5 | 5 | I/O | TTL | RA3 can also be analog input3 or analog reference voltage |
| RA4/T0CKI | 6 | 6 | 6 | I/O | ST | RA4 can also be the clock input to the Timer0 module. Output is open drain type. |
| RA5/$\overline{SS}$/AN4 | 7 | 7 | 7 | 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 | 21 | 21 | 21 | I/O | TTL/ST(1) | RB0 can also be the external interrupt pin. |
| RB1 | 22 | 22 | 22 | I/O | TTL | |
| RB2 | 23 | 23 | 23 | I/O | TTL | |
| RB3 | 24 | 24 | 24 | I/O | TTL | |
| RB4 | 25 | 25 | 25 | I/O | TTL | Interrupt on change pin. |
| RB5 | 26 | 26 | 26 | I/O | TTL | Interrupt on change pin. |
| RB6 | 27 | 27 | 27 | I/O | TTL/ST(2) | Interrupt on change pin. Serial programming clock. |
| RB7 | 28 | 28 | 28 | I/O | TTL/ST(2) | Interrupt on change pin. Serial programming data. |
| | | | | | | PORTC is a bi-directional I/O port. |
| RC0/T1OSO/T1CKI | 11 | 11 | 11 | I/O | ST | RC0 can also be the Timer1 oscillator output or Timer1 clock input. |
| RC1/T1OSI | 12 | 12 | 12 | I/O | ST | RC1 can also be the Timer1 oscillator input. |
| RC2/CCP1 | 13 | 13 | 13 | I/O | ST | RC2 can also be the Capture1 input/Compare1 output/ PWM1 output. |
| RC3/SCK/SCL | 14 | 14 | 14 | I/O | ST | RC3 can also be the synchronous serial clock input/output for both SPI and $I^2C$ modes. |
| RC4/SDI/SDA | 15 | 15 | 15 | I/O | ST | RC4 can also be the SPI Data In (SPI mode) or data I/O ($I^2C$ mode). |
| RC5/SDO | 16 | 16 | 16 | I/O | ST | RC5 can also be the SPI Data Out (SPI mode). |
| RC6 | 17 | 17 | 17 | I/O | ST | |
| RC7 | 18 | 18 | 18 | I/O | ST | |
| VSS | 8, 19 | 8, 19 | 8, 19 | P | — | Ground reference for logic and I/O pins. |
| VDD | 20 | 20 | 20 | 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.

**TABLE 3-2: PIC16C73/73A/76 PINOUT DESCRIPTION**

| Pin Name | DIP Pin# | SOIC Pin# | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|
| OSC1/CLKIN | 9 | 9 | I | ST/CMOS(3) | Oscillator crystal input/external clock source input. |
| OSC2/CLKOUT | 10 | 10 | O | — | Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate. |
| $\overline{MCLR}$/VPP | 1 | 1 | 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 | 2 | I/O | TTL | RA0 can also be analog input0 |
| RA1/AN1 | 3 | 3 | I/O | TTL | RA1 can also be analog input1 |
| RA2/AN2 | 4 | 4 | I/O | TTL | RA2 can also be analog input2 |
| RA3/AN3/VREF | 5 | 5 | I/O | TTL | RA3 can also be analog input3 or analog reference voltage |
| RA4/T0CKI | 6 | 6 | I/O | ST | RA4 can also be the clock input to the Timer0 module. Output is open drain type. |
| RA5/$\overline{SS}$/AN4 | 7 | 7 | 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 | 21 | 21 | I/O | TTL/ST(1) | RB0 can also be the external interrupt pin. |
| RB1 | 22 | 22 | I/O | TTL | |
| RB2 | 23 | 23 | I/O | TTL | |
| RB3 | 24 | 24 | I/O | TTL | |
| RB4 | 25 | 25 | I/O | TTL | Interrupt on change pin. |
| RB5 | 26 | 26 | I/O | TTL | Interrupt on change pin. |
| RB6 | 27 | 27 | I/O | TTL/ST(2) | Interrupt on change pin. Serial programming clock. |
| RB7 | 28 | 28 | I/O | TTL/ST(2) | Interrupt on change pin. Serial programming data. |
| | | | | | PORTC is a bi-directional I/O port. |
| RC0/T1OSO/T1CKI | 11 | 11 | I/O | ST | RC0 can also be the Timer1 oscillator output or Timer1 clock input. |
| RC1/T1OSI/CCP2 | 12 | 12 | I/O | ST | RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. |
| RC2/CCP1 | 13 | 13 | I/O | ST | RC2 can also be the Capture1 input/Compare1 output/ PWM1 output. |
| RC3/SCK/SCL | 14 | 14 | I/O | ST | RC3 can also be the synchronous serial clock input/output for both SPI and $I^2C$ modes. |
| RC4/SDI/SDA | 15 | 15 | I/O | ST | RC4 can also be the SPI Data In (SPI mode) or data I/O ($I^2C$ mode). |
| RC5/SDO | 16 | 16 | I/O | ST | RC5 can also be the SPI Data Out (SPI mode). |
| RC6/TX/CK | 17 | 17 | I/O | ST | RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. |
| RC7/RX/DT | 18 | 18 | I/O | ST | RC7 can also be the USART Asynchronous Receive or Synchronous Data. |
| VSS | 8, 19 | 8, 19 | P | — | Ground reference for logic and I/O pins. |
| VDD | 20 | 20 | 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.

**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.

**TABLE 3-3: PIC16C74/74A/77 PINOUT DESCRIPTION (Cont.'d)**

| Pin Name | DIP Pin# | PLCC Pin# | QFP Pin# | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|---|
| | | | | | | PORTC is a bi-directional I/O port. |
| RC0/T1OSO/T1CKI | 15 | 16 | 32 | I/O | ST | RC0 can also be the Timer1 oscillator output or a Timer1 clock input. |
| RC1/T1OSI/CCP2 | 16 | 18 | 35 | I/O | ST | RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. |
| RC2/CCP1 | 17 | 19 | 36 | I/O | ST | RC2 can also be the Capture1 input/Compare1 output/ PWM1 output. |
| RC3/SCK/SCL | 18 | 20 | 37 | I/O | ST | RC3 can also be the synchronous serial clock input/ output for both SPI and $I^2C$ modes. |
| RC4/SDI/SDA | 23 | 25 | 42 | I/O | ST | RC4 can also be the SPI Data In (SPI mode) or data I/O ($I^2C$ mode). |
| RC5/SDO | 24 | 26 | 43 | I/O | ST | RC5 can also be the SPI Data Out (SPI mode). |
| RC6/TX/CK | 25 | 27 | 44 | I/O | ST | RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. |
| RC7/RX/DT | 26 | 29 | 1 | I/O | ST | RC7 can also be the USART Asynchronous Receive or Synchronous Data. |
| | | | | | | PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus. |
| RD0/PSP0 | 19 | 21 | 38 | I/O | ST/TTL(3) | |
| RD1/PSP1 | 20 | 22 | 39 | I/O | ST/TTL(3) | |
| RD2/PSP2 | 21 | 23 | 40 | I/O | ST/TTL(3) | |
| RD3/PSP3 | 22 | 24 | 41 | I/O | ST/TTL(3) | |
| RD4/PSP4 | 27 | 30 | 2 | I/O | ST/TTL(3) | |
| RD5/PSP5 | 28 | 31 | 3 | I/O | ST/TTL(3) | |
| RD6/PSP6 | 29 | 32 | 4 | I/O | ST/TTL(3) | |
| RD7/PSP7 | 30 | 33 | 5 | I/O | ST/TTL(3) | |
| | | | | | | PORTE is a bi-directional I/O port. |
| RE0/$\overline{RD}$/AN5 | 8 | 9 | 25 | I/O | ST/TTL(3) | RE0 can also be read control for the parallel slave port, or analog input5. |
| RE1/$\overline{WR}$/AN6 | 9 | 10 | 26 | I/O | ST/TTL(3) | RE1 can also be write control for the parallel slave port, or analog input6. |
| RE2/$\overline{CS}$/AN7 | 10 | 11 | 27 | I/O | ST/TTL(3) | RE2 can also be select control for the parallel slave port, or analog input7. |
| VSS | 12,31 | 13,34 | 6,29 | P | — | Ground reference for logic and I/O pins. |
| VDD | 11,32 | 12,35 | 7,28 | P | — | Positive supply for logic and I/O pins. |
| NC | — | 1,17,28, 40 | 12,13, 33,34 | | — | These pins are not internally connected. These pins should be left unconnected. |

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.

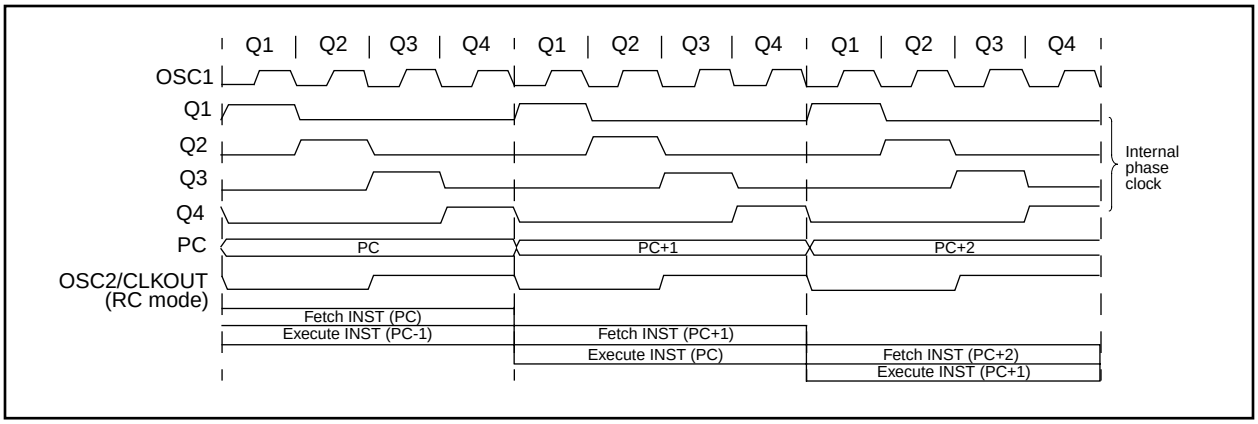
### 3.2 Instruction Flow/Pipelining

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.

**NOTES:**

## 4.0 MEMORY ORGANIZATION

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

### 4.1 Program Memory Organization

The PIC16C7X family has a 13-bit program counter capable of addressing an 8K x 14 program memory space. The amount of program memory available to each device is listed below:

| Device | Program Memory | Address Range |
|---|---|---|
| PIC16C72 | 2K x 14 | 0000h-07FFh |
| PIC16C73 | 4K x 14 | 0000h-0FFFh |
| PIC16C73A | 4K x 14 | 0000h-0FFFh |
| PIC16C74 | 4K x 14 | 0000h-0FFFh |
| PIC16C74A | 4K x 14 | 0000h-0FFFh |
| PIC16C76 | 8K x 14 | 0000h-1FFFh |
| PIC16C77 | 8K x 14 | 0000h-1FFFh |

For those devices with less than 8K program memory, accessing a location above the physically implemented address will cause a wraparound.

The reset vector is at 0000h and the interrupt vector is at 0004h.

**FIGURE 4-1: PIC16C72 PROGRAM MEMORY MAP AND STACK**

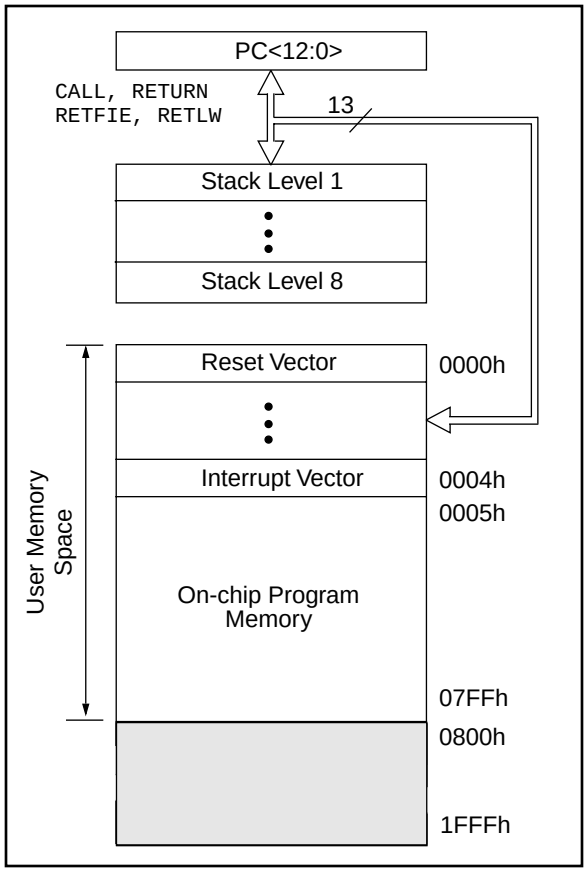



**FIGURE 4-2: PIC16C73/73A/74/74A PROGRAM MEMORY MAP AND STACK**

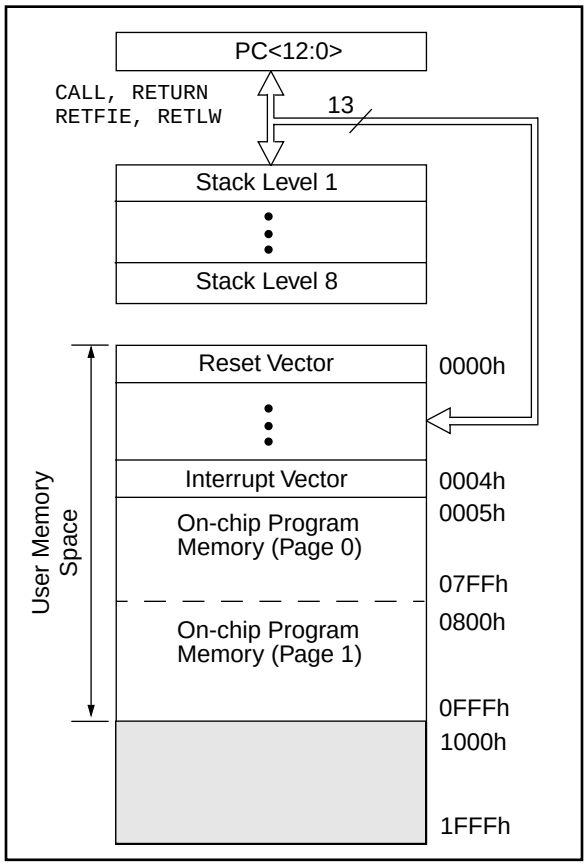
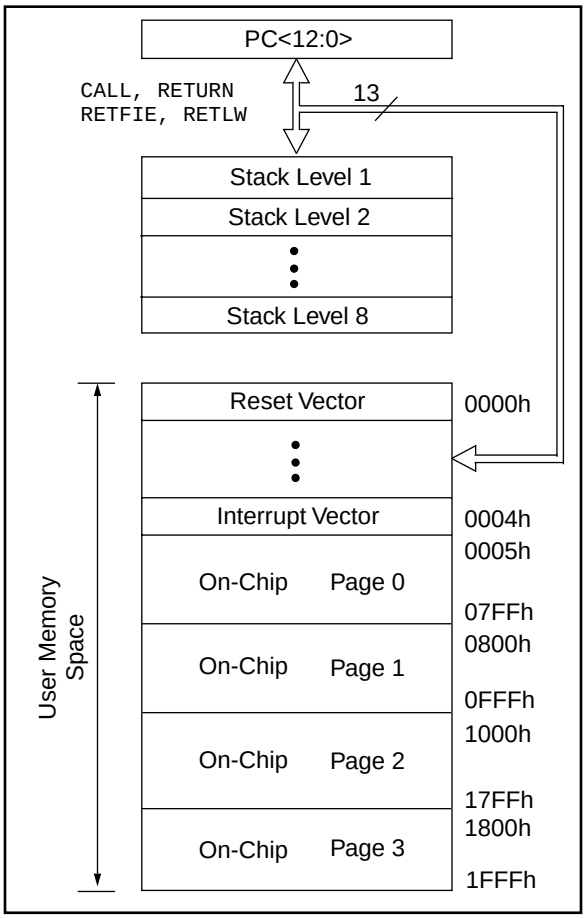


**FIGURE 4-3: PIC16C76/77 PROGRAM MEMORY MAP AND STACK**




## 4.2 Data Memory Organization

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

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 and RP0 are the bank select bits.

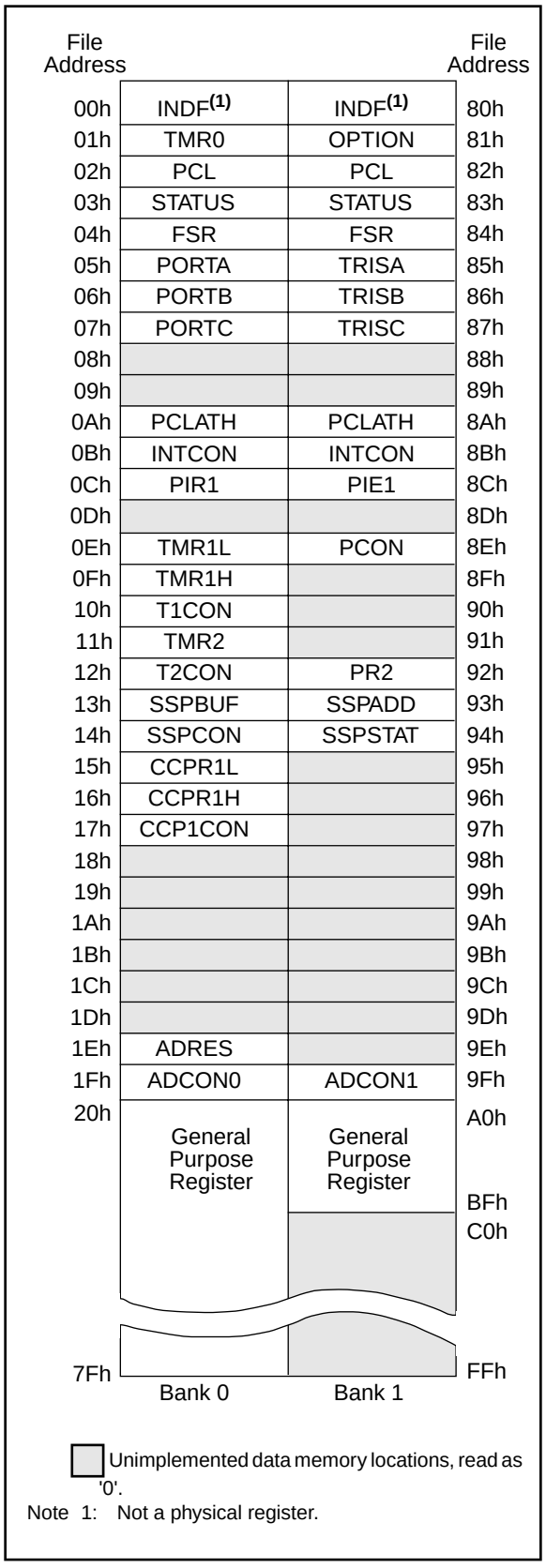
RP1:RP0 (STATUS<6:5>)
= 00 → Bank0
= 01 → Bank1
= 10 → Bank2
= 11 → Bank3

Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for the Special Function Registers. Above the Special Function Registers are General Purpose Registers, implemented as static RAM. All implemented banks contain special function registers. Some “high use” special function registers from one bank may be mirrored in another bank for code reduction and quicker access.

### 4.2.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly, or indirectly through the File Select Register FSR (Section 4.5).

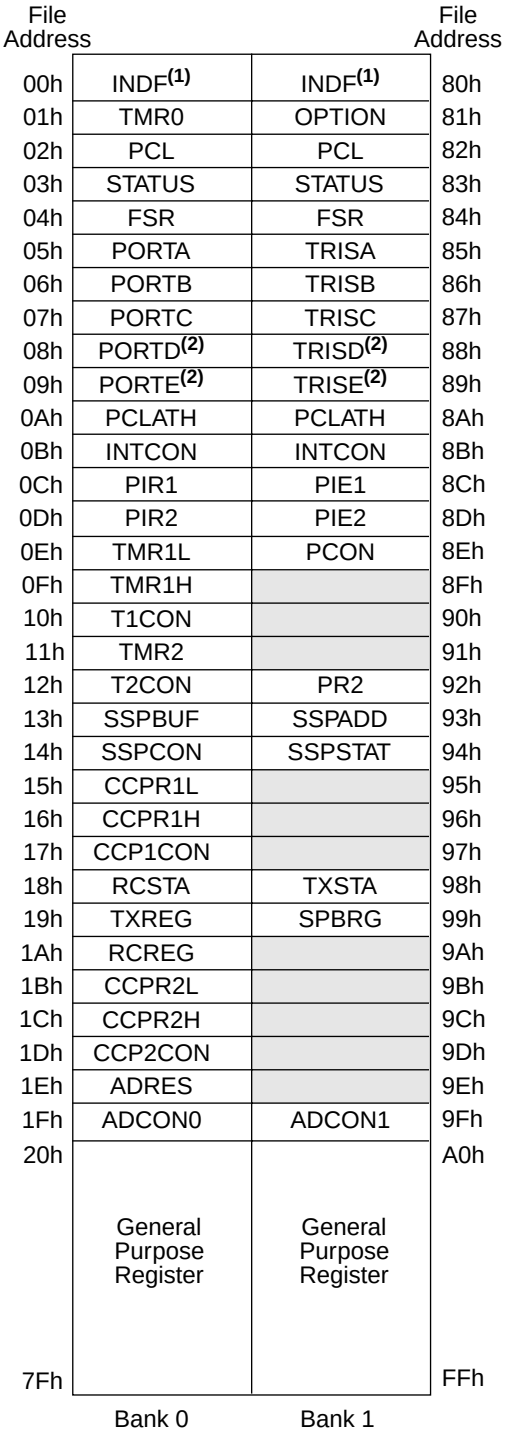
**FIGURE 4-4: PIC16C72 REGISTER FILE MAP**




Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.

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

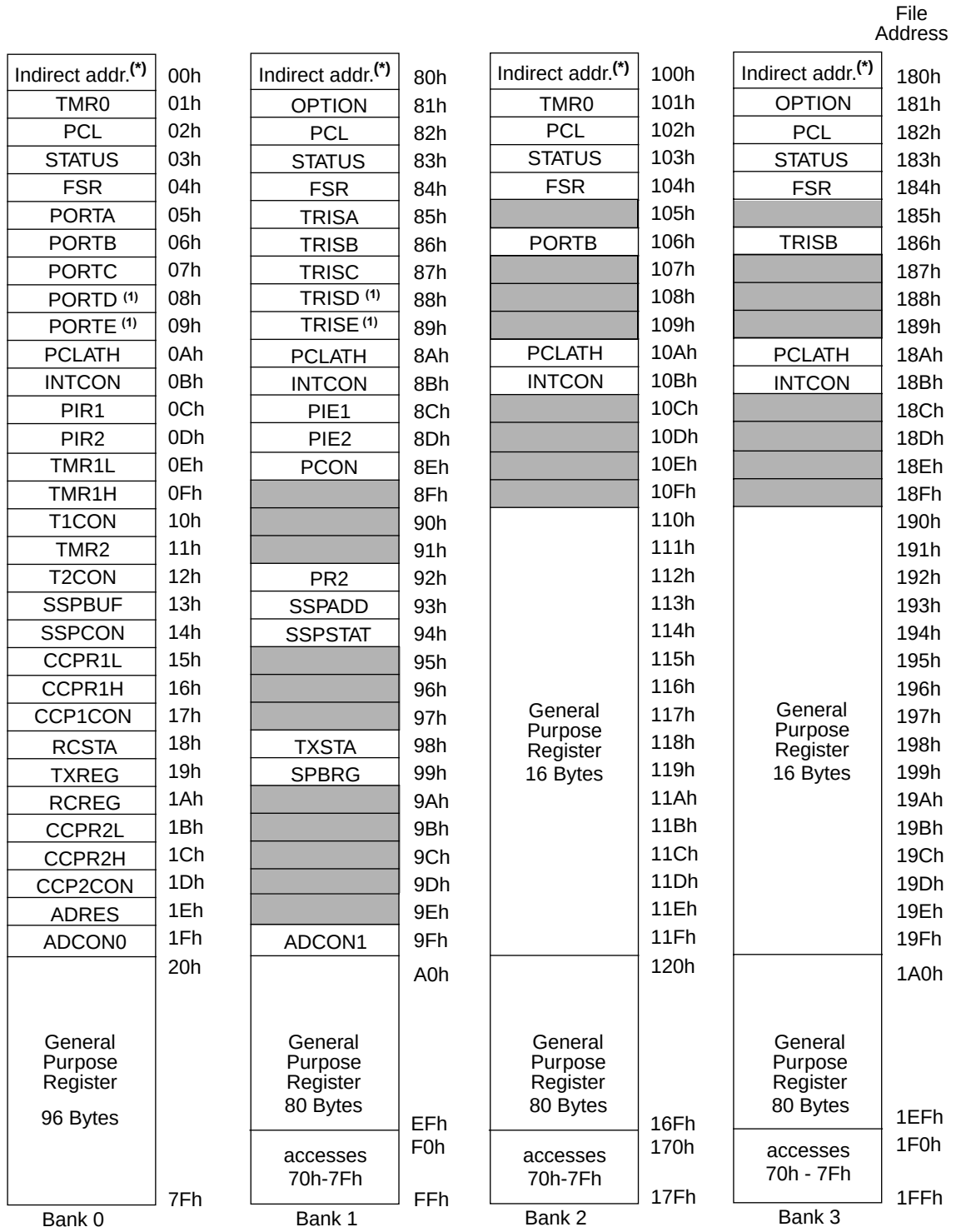



Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.

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

**FIGURE 4-6: PIC16C76/77 REGISTER FILE MAP**




Note 1: PORTD, PORTE, TRISD, and TRISE are unimplemented on the PIC16C76, read as '0'.

| **Note:** | The upper 16 bytes of data memory in banks 1, 2, and 3 are mapped in Bank 0. This may require relocation of data memory usage in the user application code if upgrading to the PIC16C76/77. |
|---|---|

### 4.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and Peripheral Modules for controlling the desired operation of the device. These registers are implemented as static RAM.

The special function registers can be classified into two sets (core and peripheral). Those registers associated with the “core” functions are described in this section, and those related to the operation of the peripheral features are described in the section of that peripheral feature.

**TABLE 4-1: PIC16C72 SPECIAL FUNCTION REGISTER SUMMARY**

| 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 (3) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 0** | | | | | | | | | | | |
| 00h(1) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 0000 0000 |
| 01h | TMR0 | Timer0 module’s register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 02h(1) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 03h(1) | STATUS | IRP(4) | RP1(4) | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 04h(1) | FSR | Indirect data memory address pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 05h | PORTA | — | — | PORTA Data Latch when written: PORTA pins when read | | | | | | --0x 0000 | --0u 0000 |
| 06h | PORTB | PORTB Data Latch when written: PORTB pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 07h | PORTC | PORTC Data Latch when written: PORTC pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 08h | — | Unimplemented | | | | | | | | — | — |
| 09h | — | Unimplemented | | | | | | | | — | — |
| 0Ah(1,2) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 0Bh(1) | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | — | ADIF | — | — | SSPIF | CCP1IF | TMR2IF | TMR1IF | -0-- 0000 | -0-- 0000 |
| 0Dh | — | Unimplemented | | | | | | | | — | — |
| 0Eh | TMR1L | Holding register for the Least Significant Byte of the 16-bit TMR1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding register for the Most Significant Byte of the 16-bit TMR1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{T1SYNC}$ | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |
| 11h | TMR2 | Timer2 module’s register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 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 |
| 15h | CCPR1L | Capture/Compare/PWM Register (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM Register (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | — | — | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |
| 18h | — | Unimplemented | | | | | | | | — | — |
| 19h | — | Unimplemented | | | | | | | | — | — |
| 1Ah | — | Unimplemented | | | | | | | | — | — |
| 1Bh | — | Unimplemented | | | | | | | | — | — |
| 1Ch | — | Unimplemented | | | | | | | | — | — |
| 1Dh | — | Unimplemented | | | | | | | | — | — |
| 1Eh | ADRES | A/D Result Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/$\overline{DONE}$ | — | ADON | 0000 00-0 | 0000 00-0 |

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'.
Shaded locations are unimplemented, read as ‘0’.

Note 1: These registers can be addressed from either bank.
2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
3: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.
4: The IRP and RP1 bits are reserved on the PIC16C72, always maintain these bits clear.

**TABLE 4-1: PIC16C72 SPECIAL FUNCTION REGISTER SUMMARY (Cont.'d)**

| 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 (3) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 1** | | | | | | | | | | | |
| 80h(1) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 0000 0000 |
| 81h | OPTION | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 82h(1) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 83h(1) | STATUS | IRP(4) | RP1(4) | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 84h(1) | FSR | Indirect data memory address pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | --11 1111 | --11 1111 |
| 86h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 88h | — | Unimplemented | | | | | | | | — | — |
| 89h | — | Unimplemented | | | | | | | | — | — |
| 8Ah(1,2) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the PC | | | | | ---0 0000 | ---0 0000 |
| 8Bh(1) | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 8Ch | PIE1 | — | ADIE | — | — | SSPIE | CCP1IE | TMR2IE | TMR1IE | -0-- 0000 | -0-- 0000 |
| 8Dh | — | Unimplemented | | | | | | | | — | — |
| 8Eh | PCON | — | — | — | — | — | — | $\overline{POR}$ | $\overline{BOR}$ | ---- --qq | ---- --uu |
| 8Fh | — | Unimplemented | | | | | | | | — | — |
| 90h | — | Unimplemented | | | | | | | | — | — |
| 91h | — | Unimplemented | | | | | | | | — | — |
| 92h | PR2 | Timer2 Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 93h | SSPADD | Synchronous Serial Port ($I^2C$ mode) Address Register | | | | | | | | 0000 0000 | 0000 0000 |
| 94h | SSPSTAT | — | — | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | --00 0000 | --00 0000 |
| 95h | — | Unimplemented | | | | | | | | — | — |
| 96h | — | Unimplemented | | | | | | | | — | — |
| 97h | — | Unimplemented | | | | | | | | — | — |
| 98h | — | Unimplemented | | | | | | | | — | — |
| 99h | — | Unimplemented | | | | | | | | — | — |
| 9Ah | — | Unimplemented | | | | | | | | — | — |
| 9Bh | — | Unimplemented | | | | | | | | — | — |
| 9Ch | — | Unimplemented | | | | | | | | — | — |
| 9Dh | — | Unimplemented | | | | | | | | — | — |
| 9Eh | — | Unimplemented | | | | | | | | — | — |
| 9Fh | ADCON1 | — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 | ---- -000 | ---- -000 |

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'.
Shaded locations are unimplemented, read as '0'.

Note 1: These registers can be addressed from either bank.
2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
3: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.
4: The IRP and RP1 bits are reserved on the PIC16C72, always maintain these bits clear.

**TABLE 4-2: PIC16C73/73A/74/74A SPECIAL FUNCTION REGISTER SUMMARY**

| 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 (2) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 0** | | | | | | | | | | | |
| 00h(4) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 0000 0000 |
| 01h | TMR0 | Timer0 module's register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 02h(4) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 03h(4) | STATUS | IRP(7) | RP1(7) | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 04h(4) | FSR | Indirect data memory address pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 05h | PORTA | — | — | PORTA Data Latch when written: PORTA pins when read | | | | | | --0x 0000 | --0u 0000 |
| 06h | PORTB | PORTB Data Latch when written: PORTB pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 07h | PORTC | PORTC Data Latch when written: PORTC pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 08h(5) | PORTD | PORTD Data Latch when written: PORTD pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 09h(5) | PORTE | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |
| 0Ah(1,4) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 0Bh(4) | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(3) | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 0Dh | PIR2 | — | — | — | – | — | — | — | CCP2IF | ---- ---0 | ---- ---0 |
| 0Eh | TMR1L | Holding register for the Least Significant Byte of the 16-bit TMR1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding register for the Most Significant Byte of the 16-bit TMR1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{T1SYNC}$ | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |
| 11h | TMR2 | Timer2 module's register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 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 |
| 15h | CCPR1L | Capture/Compare/PWM Register1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM Register1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | — | — | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 19h | TXREG | USART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | USART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Bh | CCPR2L | Capture/Compare/PWM Register2 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Ch | CCPR2H | Capture/Compare/PWM Register2 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Dh | CCP2CON | — | — | CCP2X | CCP2Y | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | --00 0000 | --00 0000 |
| 1Eh | ADRES | A/D Result Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/$\overline{DONE}$ | — | ADON | 0000 00-0 | 0000 00-0 |

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.

3: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A, always maintain these bits clear.

4: These registers can be addressed from either bank.

5: PORTD and PORTE are not physically implemented on the PIC16C73/73A, read as '0'.

6: Brown-out Reset is not implemented on the PIC16C73 or the PIC16C74, read as '0'.

7: The IRP and RP1 bits are reserved on the PIC16C73/73A/74/74A, always maintain these bits clear.

**TABLE 4-2: PIC16C73/73A/74/74A SPECIAL FUNCTION REGISTER SUMMARY (Cont.'d)**

| 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 (2) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 1** | | | | | | | | | | | |
| 80h(4) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 0000 0000 |
| 81h | OPTION | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 82h(4) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 83h(4) | STATUS | IRP(7) | RP1(7) | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 84h(4) | FSR | Indirect data memory address pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | --11 1111 | --11 1111 |
| 86h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 88h(5) | TRISD | PORTD Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 89h(5) | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction Bits | | | 0000 -111 | 0000 -111 |
| 8Ah(1,4) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 8Bh(4) | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 8Ch | PIE1 | PSPIE(3) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 8Dh | PIE2 | — | — | — | — | — | — | — | CCP2IE | ---- ---0 | ---- ---0 |
| 8Eh | PCON | — | — | — | — | — | — | $\overline{POR}$ | $\overline{BOR}$(6) | ---- --qq | ---- --uu |
| 8Fh | — | Unimplemented | | | | | | | | — | — |
| 90h | — | Unimplemented | | | | | | | | — | — |
| 91h | — | Unimplemented | | | | | | | | — | — |
| 92h | PR2 | Timer2 Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 93h | SSPADD | Synchronous Serial Port ($I^2C$ mode) Address Register | | | | | | | | 0000 0000 | 0000 0000 |
| 94h | SSPSTAT | — | — | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | --00 0000 | --00 0000 |
| 95h | — | Unimplemented | | | | | | | | — | — |
| 96h | — | Unimplemented | | | | | | | | — | — |
| 97h | — | Unimplemented | | | | | | | | — | — |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |
| 9Ah | — | Unimplemented | | | | | | | | — | — |
| 9Bh | — | Unimplemented | | | | | | | | — | — |
| 9Ch | — | Unimplemented | | | | | | | | — | — |
| 9Dh | — | Unimplemented | | | | | | | | — | — |
| 9Eh | — | Unimplemented | | | | | | | | — | — |
| 9Fh | ADCON1 | — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 | ---- -000 | ---- -000 |

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'.
Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
2: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.
3: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A, always maintain these bits clear.
4: These registers can be addressed from either bank.
5: PORTD and PORTE are not physically implemented on the PIC16C73/73A, read as '0'.
6: Brown-out Reset is not implemented on the PIC16C73 or the PIC16C74, read as '0'.
7: The IRP and RP1 bits are reserved on the PIC16C73/73A/74/74A, always maintain these bits clear.

**TABLE 4-3: PIC16C76/77 SPECIAL FUNCTION REGISTER SUMMARY**

| 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 (2) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 0** | | | | | | | | | | | |
| 00h(4) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 0000 0000 |
| 01h | TMR0 | Timer0 module's register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 02h(4) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 03h(4) | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 04h(4) | FSR | Indirect data memory address pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 05h | PORTA | — | — | PORTA Data Latch when written: PORTA pins when read | | | | | | --0x 0000 | --0u 0000 |
| 06h | PORTB | PORTB Data Latch when written: PORTB pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 07h | PORTC | PORTC Data Latch when written: PORTC pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 08h(5) | PORTD | PORTD Data Latch when written: PORTD pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 09h(5) | PORTE | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |
| 0Ah(1,4) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 0Bh(4) | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(3) | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 0Dh | PIR2 | — | — | — | – | — | — | — | CCP2IF | ---- ---0 | ---- ---0 |
| 0Eh | TMR1L | Holding register for the Least Significant Byte of the 16-bit TMR1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding register for the Most Significant Byte of the 16-bit TMR1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{T1SYNC}$ | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |
| 11h | TMR2 | Timer2 module's register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 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 |
| 15h | CCPR1L | Capture/Compare/PWM Register1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM Register1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | — | — | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 19h | TXREG | USART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | USART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Bh | CCPR2L | Capture/Compare/PWM Register2 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Ch | CCPR2H | Capture/Compare/PWM Register2 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Dh | CCP2CON | — | — | CCP2X | CCP2Y | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | --00 0000 | --00 0000 |
| 1Eh | ADRES | A/D Result Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/$\overline{DONE}$ | — | ADON | 0000 00-0 | 0000 00-0 |

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'.
Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.

3: Bits PSPIE and PSPIF are reserved on the PIC16C76, always maintain these bits clear.

4: These registers can be addressed from any bank.

5: PORTD and PORTE are not physically implemented on the PIC16C76, read as '0'.

**TABLE 4-3: PIC16C76/77 SPECIAL FUNCTION REGISTER SUMMARY (Cont.'d)**

| 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 (2) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 1** | | | | | | | | | | | |
| 80h(4) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 0000 0000 |
| 81h | OPTION | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 82h(4) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 83h(4) | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 84h(4) | FSR | Indirect data memory address pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | --11 1111 | --11 1111 |
| 86h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 88h(5) | TRISD | PORTD Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 89h(5) | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction Bits | | | 0000 -111 | 0000 -111 |
| 8Ah(1,4) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 8Bh(4) | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 8Ch | PIE1 | PSPIE(3) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 8Dh | PIE2 | — | — | — | — | — | — | — | CCP2IE | ---- ---0 | ---- ---0 |
| 8Eh | PCON | — | — | — | — | — | — | $\overline{POR}$ | $\overline{BOR}$ | ---- --qq | ---- --uu |
| 8Fh | — | Unimplemented | | | | | | | | — | — |
| 90h | — | Unimplemented | | | | | | | | — | — |
| 91h | — | Unimplemented | | | | | | | | — | — |
| 92h | PR2 | Timer2 Period Register | | | | | | | | 1111 1111 | 1111 1111 |
| 93h | SSPADD | Synchronous Serial Port ($I^2C$ mode) Address Register | | | | | | | | 0000 0000 | 0000 0000 |
| 94h | SSPSTAT | SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 0000 0000 | 0000 0000 |
| 95h | — | Unimplemented | | | | | | | | — | — |
| 96h | — | Unimplemented | | | | | | | | — | — |
| 97h | — | Unimplemented | | | | | | | | — | — |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |
| 9Ah | — | Unimplemented | | | | | | | | — | — |
| 9Bh | — | Unimplemented | | | | | | | | — | — |
| 9Ch | — | Unimplemented | | | | | | | | — | — |
| 9Dh | — | Unimplemented | | | | | | | | — | — |
| 9Eh | — | Unimplemented | | | | | | | | — | — |
| 9Fh | ADCON1 | — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 | ---- -000 | ---- -000 |

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'.
Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.

3: Bits PSPIE and PSPIF are reserved on the PIC16C76, always maintain these bits clear.

4: These registers can be addressed from any bank.

5: PORTD and PORTE are not physically implemented on the PIC16C76, read as '0'.

**TABLE 4-3: PIC16C76/77 SPECIAL FUNCTION REGISTER SUMMARY (Cont.'d)**

| 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 (2) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 2** | | | | | | | | | | | |
| 100h(4) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 0000 0000 |
| 101h | TMR0 | Timer0 module's register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 102h(4) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 103h(4) | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 104h(4) | FSR | Indirect data memory address pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 105h | — | Unimplemented | | | | | | | | — | — |
| 106h | PORTB | PORTB Data Latch when written: PORTB pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 107h | — | Unimplemented | | | | | | | | — | — |
| 108h | — | Unimplemented | | | | | | | | — | — |
| 109h | — | Unimplemented | | | | | | | | — | — |
| 10Ah(1,4) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 10Bh(4) | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 10Ch-10Fh | — | Unimplemented | | | | | | | | — | — |
| **Bank 3** | | | | | | | | | | | |
| 180h(4) | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 0000 0000 |
| 181h | OPTION | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 182h(4) | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 183h(4) | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 184h(4) | FSR | Indirect data memory address pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 185h | — | Unimplemented | | | | | | | | — | — |
| 186h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 187h | — | Unimplemented | | | | | | | | — | — |
| 188h | — | Unimplemented | | | | | | | | — | — |
| 189h | — | Unimplemented | | | | | | | | — | — |
| 18Ah(1,4) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 18Bh(4) | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 18Ch-18Fh | — | Unimplemented | | | | | | | | — | — |

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented read as '0'.
Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
2: Other (non power-up) resets include external reset through $\overline{MCLR}$ and Watchdog Timer Reset.
3: Bits PSPIE and PSPIF are reserved on the PIC16C76, always maintain these bits clear.
4: These registers can be addressed from any bank.
5: PORTD and PORTE are not physically implemented on the PIC16C76, read as '0'.

4.2.2.1 STATUS REGISTER

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

The STATUS register, shown in Figure 4-7, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper-three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register because these instructions do not affect the Z, C or DC bits from the STATUS register. For other instructions, not affecting any status bits, see the "Instruction Set Summary."

**Note 1:** For those devices that do not use bits IRP and RP1 (STATUS<7:6>), maintain these bits clear to ensure upward compatibility with future products.

**Note 2:** The C and DC bits operate as a $\overline{borrow}$ and $\overline{digit\ borrow}$ bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

**FIGURE 4-7: STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)**

| R/W-0 | R/W-0 | R/W-0 | R-1 | R-1 | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C |
| bit7 | | | | | | | bit0 |

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

bit 7: **IRP**: Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h - 1FFh)
0 = Bank 0, 1 (00h - FFh)

bit 6-5: **RP1:RP0**: Register Bank Select bits (used for direct addressing)
`11` = Bank 3 (180h - 1FFh)
`10` = Bank 2 (100h - 17Fh)
`01` = Bank 1 (80h - FFh)
`00` = Bank 0 (00h - 7Fh)
Each bank is 128 bytes

bit 4: **$\overline{TO}$**: Time-out bit
1 = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
0 = A WDT time-out occurred

bit 3: **$\overline{PD}$**: Power-down bit
1 = After power-up or by the `CLRWDT` instruction
0 = By execution of the `SLEEP` instruction

bit 2: **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1: **DC**: Digit carry/$\overline{borrow}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions) (for $\overline{borrow}$ the polarity is reversed)
1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result

bit 0: **C**: Carry/$\overline{borrow}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
1 = A carry-out from the most significant bit of the result occurred
0 = No carry-out from the most significant bit of the result occurred
Note: For $\overline{borrow}$ the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

4.2.2.2 OPTION REGISTER

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

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the External INT Interrupt, TMR0, and the weak pull-ups on PORTB.

**Note:** To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.

**FIGURE 4-8: OPTION REGISTER (ADDRESS 81h, 181h)**

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| bit7 | | | | | | | bit0 |

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

bit 7: **$\overline{\text{RBPU}}$**: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6: **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5: **T0CS**: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4: **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3: **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0: **PS2:PS0**: Prescaler Rate Select bits

| Bit Value | TMR0 Rate | WDT Rate |
|---|---|---|
| 000 | 1 : 2 | 1 : 1 |
| 001 | 1 : 4 | 1 : 2 |
| 010 | 1 : 8 | 1 : 4 |
| 011 | 1 : 16 | 1 : 8 |
| 100 | 1 : 32 | 1 : 16 |
| 101 | 1 : 64 | 1 : 32 |
| 110 | 1 : 128 | 1 : 64 |
| 111 | 1 : 256 | 1 : 128 |

4.2.2.3 INTCON REGISTER

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

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, 10Bh, 18Bh)**

| 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 | PEIE | 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: **PEIE**: Peripheral Interrupt Enable bit
1 = Enables all un-masked peripheral interrupts
0 = Disables all peripheral interrupts

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 PIC16C73 and PIC16C74, 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 14.5 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.

4.2.2.4 PIE1 REGISTER

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

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 PIC16C72 (ADDRESS 8Ch)**

| U-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | ADIE | — | — | SSPIE | CCP1IE | TMR2IE | TMR1IE |
| 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 5-4: **Unimplemented**: Read as '0'

bit 3: **SSPIE**: Synchronous Serial Port Interrupt Enable bit
1 = Enables the SSP interrupt
0 = Disables the SSP interrupt

bit 2: **CCP1IE**: CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1: **TMR2IE**: TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0: **TMR1IE**: TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

**FIGURE 4-11: PIE1 REGISTER PIC16C73/73A/74/74A/76/77 (ADDRESS 8Ch)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| PSPIE(1) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE |
| bit7 | | | | | | | bit0 |

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

bit 7: **PSPIE(1):** Parallel Slave Port Read/Write Interrupt Enable bit
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt

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

bit 5: **RCIE**: USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt

bit 4: **TXIE**: USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt

bit 3: **SSPIE**: Synchronous Serial Port Interrupt Enable bit
1 = Enables the SSP interrupt
0 = Disables the SSP interrupt

bit 2: **CCP1IE**: CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1: **TMR2IE**: TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0: **TMR1IE**: TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

Note 1: PIC16C73/73A/76 devices do not have a Parallel Slave Port implemented, this bit location is reserved on these devices, always maintain this bit clear.

4.2.2.5 PIR1 REGISTER

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

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 4-12: PIR1 REGISTER PIC16C72 (ADDRESS 0Ch)**

| U-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | ADIF | — | — | SSPIF | CCP1IF | TMR2IF | TMR1IF |
| 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 (must be cleared in software)
0 = The A/D conversion is not complete

bit 5-4: **Unimplemented**: Read as '0'

bit 3: **SSPIF**: Synchronous Serial Port Interrupt Flag bit
1 = The transmission/reception is complete (must be cleared in software)
0 = Waiting to transmit/receive

bit 2: **CCP1IF**: CCP1 Interrupt Flag bit
Capture Mode
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare Mode
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM Mode
Unused in this mode

bit 1: **TMR2IF**: TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0: **TMR1IF**: TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

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 4-13: PIR1 REGISTER PIC16C73/73A/74/74A/76/77 (ADDRESS 0Ch)**

| R/W-0 | R/W-0 | R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| PSPIF(1) | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF |
| bit7 | | | | | | | bit0 |

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

bit 7: **PSPIF(1):** Parallel Slave Port Read/Write Interrupt Flag bit
1 = A read or a write operation has taken place (must be cleared in software)
0 = No read or write has occurred

bit 6: **ADIF**: A/D Converter Interrupt Flag bit
1 = An A/D conversion completed (must be cleared in software)
0 = The A/D conversion is not complete

bit 5: **RCIF**: USART Receive Interrupt Flag bit
1 = The USART receive buffer is full (cleared by reading RCREG)
0 = The USART receive buffer is empty

bit 4: **TXIF**: USART Transmit Interrupt Flag bit
1 = The USART transmit buffer is empty (cleared by writing to TXREG)
0 = The USART transmit buffer is full

bit 3: **SSPIF**: Synchronous Serial Port Interrupt Flag bit
1 = The transmission/reception is complete (must be cleared in software)
0 = Waiting to transmit/receive

bit 2: **CCP1IF**: CCP1 Interrupt Flag bit
Capture Mode
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare Mode
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM Mode
Unused in this mode

bit 1: **TMR2IF**: TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0: **TMR1IF**: TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

Note 1: PIC16C73/73A/76 devices do not have a Parallel Slave Port implemented, this bit location is reserved on these devices, always maintain this bit clear.

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.

4.2.2.6 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

4.2.2.7 PIR2 REGISTER

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

This register contains the CCP2 interrupt flag bit.

| **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 4-15: PIR2 REGISTER (ADDRESS 0Dh)**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | CCP2IF |
| 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: **CCP2IF**: CCP2 Interrupt Flag bit

Capture Mode
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred

Compare Mode
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred

PWM Mode
Unused

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.

4.2.2.8 PCON REGISTER

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

The Power Control (PCON) register contains a flag bit to allow differentiation between a Power-on Reset (POR) to an external $\overline{MCLR}$ Reset or WDT Reset. Those devices with brown-out detection circuitry contain an additional bit to differentiate a Brown-out Reset condition from a Power-on Reset condition.

**Note:** $\overline{BOR}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent resets to see if $\overline{BOR}$ is clear, indicating a brown-out has occurred. The $\overline{BOR}$ 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 4-16: PCON REGISTER (ADDRESS 8Eh)**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-q |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | $\overline{POR}$ | $\overline{BOR}$(1) |
| 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: **$\overline{POR}$**: 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: **$\overline{BOR}$(1)**: 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)

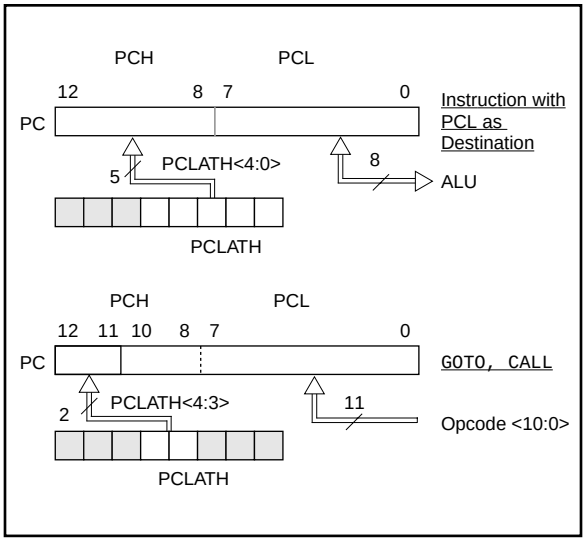
Note 1: Brown-out Reset is not implemented on the PIC16C73/74.

## 4.3 PCL and PCLATH

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

The program counter (PC) is 13-bits wide. The low byte comes from the PCL register, which is a readable and writable register. The upper bits (PC<12:8>) are not readable, but are indirectly writable through the PCLATH register. On any reset, the upper bits of the PC will be cleared. Figure 4-17 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in the figure shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

**FIGURE 4-17: LOADING OF PC IN DIFFERENT SITUATIONS**




### 4.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note *"Implementing a Table Read"* (AN556).

### 4.3.2 STACK

The PIC16CXX family has an 8 level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

**Note 1:** There are no status bits to indicate stack overflow or stack underflow conditions.

**Note 2:** There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW, and RETFIE instructions, or the vectoring to an interrupt address.

## 4.4 Program Memory Paging

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

PIC16C7X devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction the upper 2 bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is pushed onto the stack. Therefore, manipulation of the PCLATH<4:3> bits are not required for the return instructions (which POPs the address from the stack).

**Note:** PIC16C7X devices with 4K or less of program memory ignore paging bit PCLATH<4>. The use of PCLATH<4> as a general purpose read/write bit is not recommended since this may affect upward compatibility with future products.

Example 4-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that PCLATH is saved and restored by the interrupt service routine (if interrupts are used).

**EXAMPLE 4-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0**

```
ORG 0x500
BSF     PCLATH,3  ;Select page 1 (800h-FFFh)
BCF     PCLATH,4  ;Only on >4K devices
CALL    SUB1_P1   ;Call subroutine in
        :         ;page 1 (800h-FFFh)
        :
        :
ORG 0x900
SUB1_P1:          ;called subroutine
        :         ;page 1 (800h-FFFh)
        :
RETURN            ;return to Call subroutine
                  ;in page 0 (000h-7FFh)
```

## 4.5 Indirect Addressing, INDF and FSR Registers

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

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

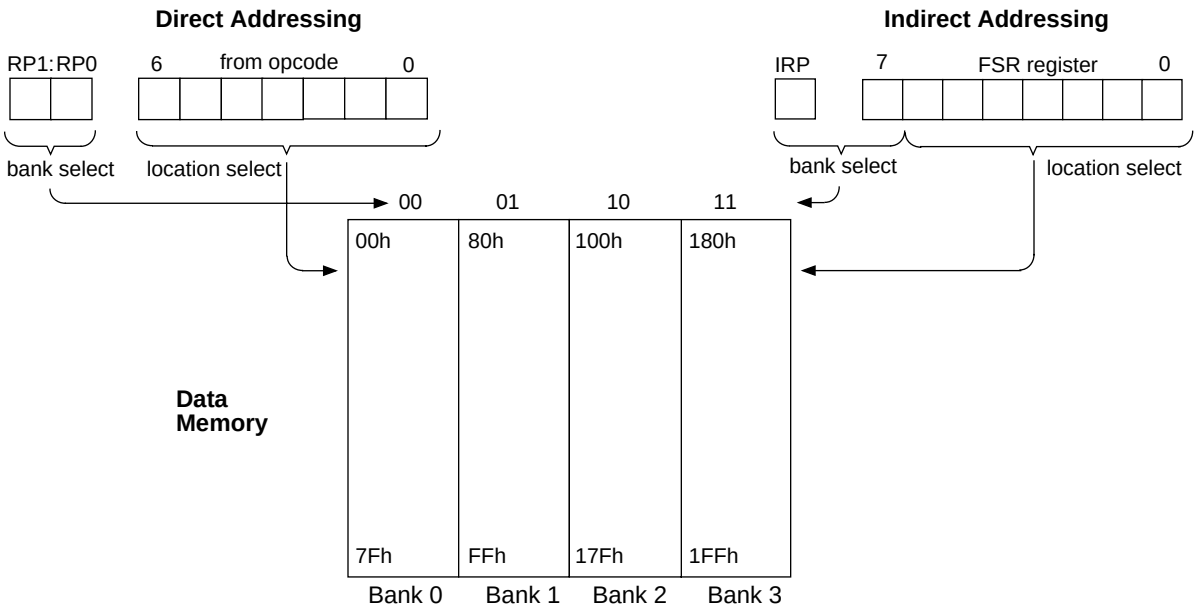
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself indirectly (FSR = '0') will read 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-18.

A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 4-2.

**EXAMPLE 4-2: INDIRECT ADDRESSING**

```
        movlw  0x20   ;initialize pointer
        movwf  FSR    ;to RAM
NEXT    clrf   INDF   ;clear INDF register
        incf   FSR,F  ;inc pointer
        btfss  FSR,4  ;all done?
        goto   NEXT   ;no clear next
CONTINUE
        :              ;yes continue
```

**FIGURE 4-18: DIRECT/INDIRECT ADDRESSING**




For register file map detail see Figure 4-4, and Figure 4-5.

**NOTES:**

## 5.0 I/O PORTS

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

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

### 5.1 PORTA and TRISA Registers

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

PORTA is a 6-bit latch.

The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers) which can configure these pins as output or input.

Setting a TRISA register bit puts the corresponding output driver in a hi-impedance mode. Clearing a bit in the TRISA register puts the contents of the output latch on the selected pin(s).

Reading the PORTA register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore a write to a port implies that the port pins are read, this value is modified, and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin.

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

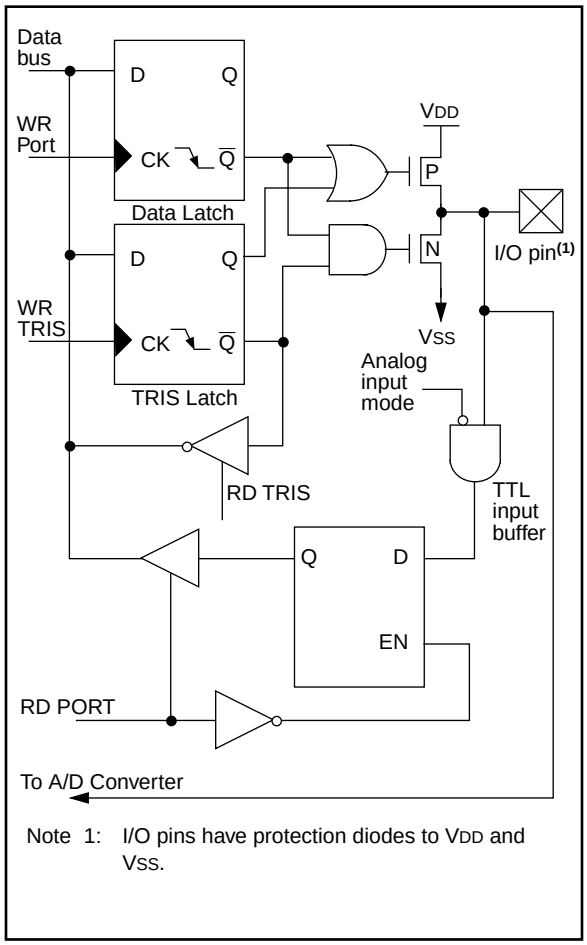
| **Note:** | On a Power-on Reset, these pins are configured as analog inputs and read as '0'. |
|---|---|

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

**EXAMPLE 5-1: INITIALIZING PORTA**

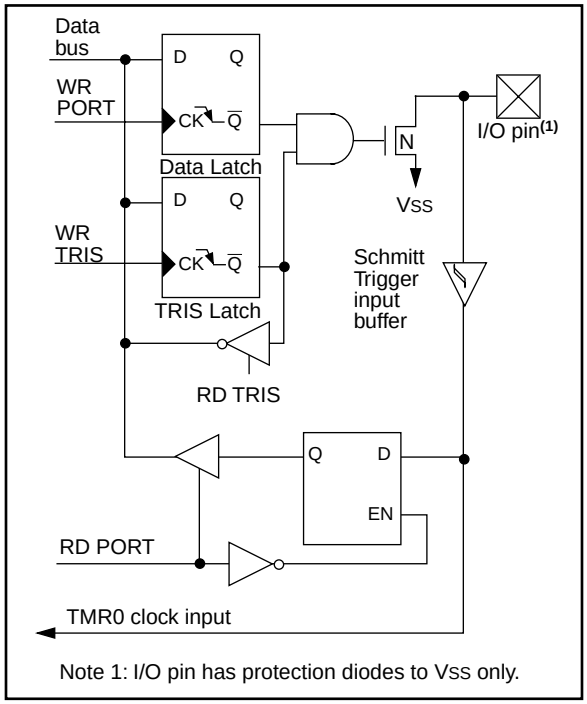
```
BCF    STATUS, RP0  ;
BCF    STATUS, RP1  ; PIC16C76/77 only
CLRF   PORTA        ; Initialize PORTA by
                    ; clearing output
                    ; data latches
BSF    STATUS, RP0  ; Select Bank 1
MOVLW  0xCF         ; Value used to
                    ; initialize data
                    ; direction
MOVWF  TRISA        ; Set RA<3:0> as inputs
                    ; RA<5:4> as outputs
                    ; TRISA<7:6> are always
                    ; read as '0'.
```

**FIGURE 5-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS**




Note 1: I/O pins have protection diodes to VDD and VSS.

**FIGURE 5-2: BLOCK DIAGRAM OF RA4/ T0CKI PIN**




Note 1: I/O pin has protection diodes to VSS only.

**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.

## 5.2 PORTB and TRISB Registers

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

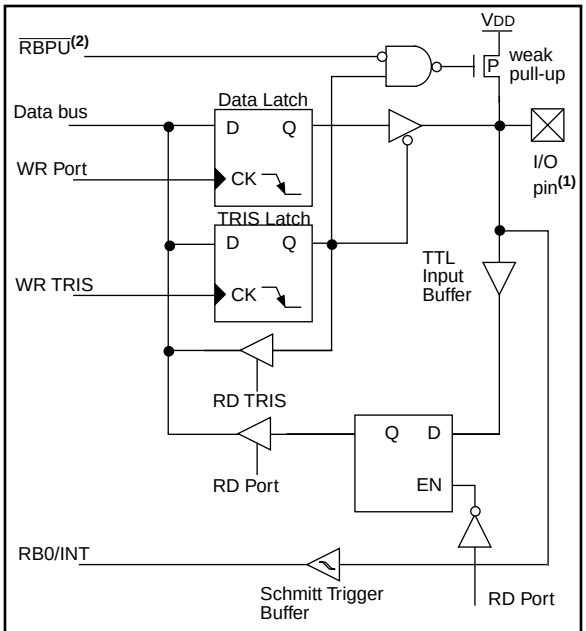
PORTB is an 8-bit wide bi-directional port. The corresponding data direction register is TRISB. Setting a bit in the TRISB register puts the corresponding output driver in a hi-impedance input mode. Clearing a bit in the TRISB register puts the contents of the output latch on the selected pin(s).

**EXAMPLE 5-2: INITIALIZING PORTB**

```
BCF    STATUS, RP0  ;
CLRF   PORTB        ; Initialize PORTB by
                    ; clearing output
                    ; data latches
BSF    STATUS, RP0  ; Select Bank 1
MOVLW  0xCF         ; Value used to
                    ; initialize data
                    ; direction
MOVWF  TRISB        ; Set RB<3:0> as inputs
                    ; RB<5:4> as outputs
                    ; RB<7:6> as inputs
```

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit $\overline{\text{RBPU}}$ (OPTION<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

**FIGURE 5-3: BLOCK DIAGRAM OF RB3:RB0 PINS**




Note 1: I/O pins have diode protection to VDD and VSS.
2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the $\overline{\text{RBPU}}$ bit (OPTION<7>).

Four of PORTB's pins, RB7:RB4, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e. any RB7:RB4 pin configured as an output is excluded from the interrupt on change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'ed together to generate the RB Port Change Interrupt with flag bit RBIF (INTCON<0>).

This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in the following manner:

a) Any read or write of PORTB. This will end the mismatch condition.
b) Clear flag bit RBIF.

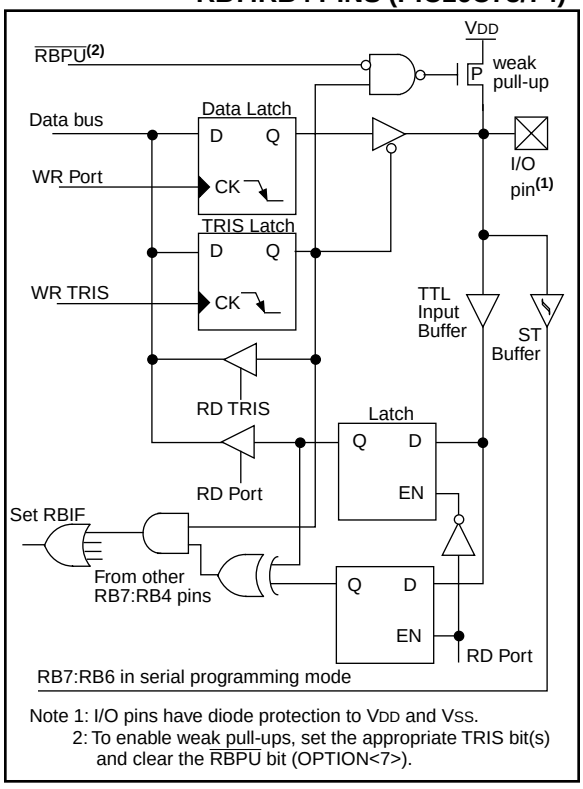
A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition, and allow flag bit RBIF to be cleared.

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a keypad and make it possible for wake-up on key-depression. Refer to the Embedded Control Handbook, *"Implementing Wake-Up on Key Stroke"* (AN552).

| **Note:** | For the PIC16C73/74, if a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then interrupt flag bit RBIF may not get set. |
|---|---|

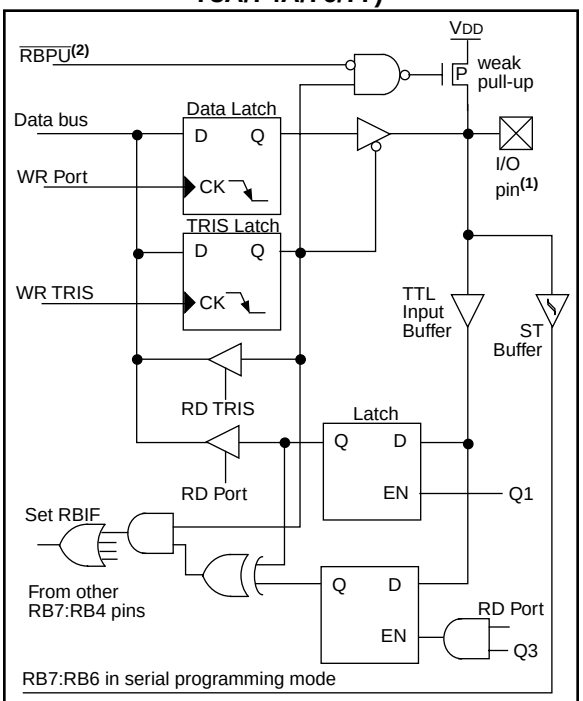
The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

**FIGURE 5-4: BLOCK DIAGRAM OF RB7:RB4 PINS (PIC16C73/74)**




Note 1: I/O pins have diode protection to VDD and VSS.
2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION<7>).

**FIGURE 5-5: BLOCK DIAGRAM OF RB7:RB4 PINS (PIC16C72/ 73A/74A/76/77)**




Note 1: I/O pins have diode protection to VDD and VSS.
2: To enable weak pull-ups, set the appropriate TRIS bit(s) and clear the RBPU bit (OPTION<7>).

**TABLE 5-3: PORTB FUNCTIONS**

| Name | Bit# | Buffer | Function |
|---|---|---|---|
| RB0/INT | bit0 | TTL/ST(1) | Input/output pin or external interrupt input. Internal software programmable weak pull-up. |
| RB1 | bit1 | TTL | Input/output pin. Internal software programmable weak pull-up. |
| RB2 | bit2 | TTL | Input/output pin. Internal software programmable weak pull-up. |
| RB3 | bit3 | TTL | Input/output pin. Internal software programmable weak pull-up. |
| RB4 | bit4 | TTL | Input/output pin (with interrupt on change). Internal software programmable weak pull-up. |
| RB5 | bit5 | TTL | Input/output pin (with interrupt on change). Internal software programmable weak pull-up. |
| RB6 | bit6 | TTL/ST(2) | Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming clock. |
| RB7 | bit7 | TTL/ST(2) | Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming data. |

Legend: 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.

**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.

## 5.3 PORTC and TRISC Registers

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

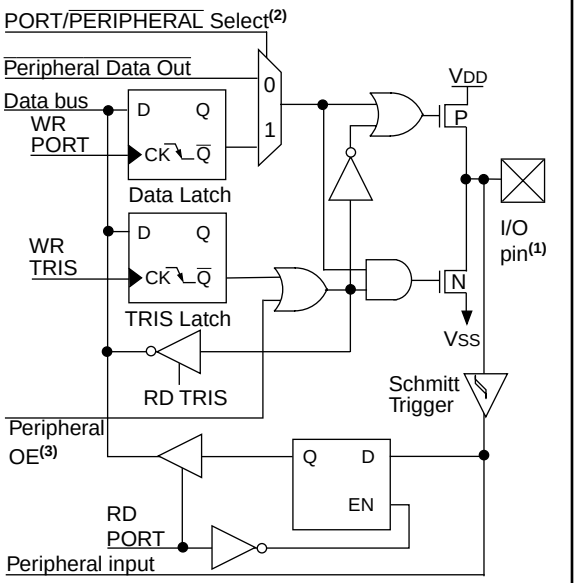
PORTC is an 8-bit bi-directional port. Each pin is individually configurable as an input or output through the TRISC register. PORTC is multiplexed with several peripheral functions (Table 5-5). PORTC pins have Schmitt Trigger input buffers.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (BSF, BCF, XORWF) with TRISC as destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

**EXAMPLE 5-3: INITIALIZING PORTC**

```
BCF     STATUS, RP0  ; Select Bank 0
BCF     STATUS, RP1  ; PIC16C76/77 only
CLRF    PORTC        ; Initialize PORTC by
                     ; clearing output
                     ; data latches
BSF     STATUS, RP0  ; Select Bank 1
MOVLW   0xCF         ; Value used to
                     ; initialize data
                     ; direction
MOVWF   TRISC        ; Set RC<3:0> as inputs
                     ; RC<5:4> as outputs
                     ; RC<7:6> as inputs
```

**FIGURE 5-6: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE)**




Note 1: I/O pins have diode protection to VDD and VSS.
2: Port/Peripheral select signal selects between port data and peripheral output.
3: Peripheral OE (output enable) is only activated if peripheral select is active.

**TABLE 5-5: PORTC FUNCTIONS**

| Name | Bit# | Buffer Type | Function |
|---|---|---|---|
| RC0/T1OSO/T1CKI | bit0 | ST | Input/output port pin or Timer1 oscillator output/Timer1 clock input |
| RC1/T1OSI/CCP2(1) | bit1 | ST | Input/output port pin or Timer1 oscillator input or Capture2 input/ Compare2 output/PWM2 output |
| RC2/CCP1 | bit2 | ST | Input/output port pin or Capture1 input/Compare1 output/PWM1 output |
| RC3/SCK/SCL | bit3 | ST | RC3 can also be the synchronous serial clock for both SPI and $I^2C$ modes. |
| RC4/SDI/SDA | bit4 | ST | RC4 can also be the SPI Data In (SPI mode) or data I/O ($I^2C$ mode). |
| RC5/SDO | bit5 | ST | Input/output port pin or Synchronous Serial Port data output |
| RC6/TX/CK(2) | bit6 | ST | Input/output port pin or USART Asynchronous Transmit, or USART Synchronous Clock |
| RC7/RX/DT(2) | bit7 | ST | Input/output port pin or USART Asynchronous Receive, or USART Synchronous Data |

Legend: ST = Schmitt Trigger input
Note 1: The CCP2 multiplexed function is not enabled on the PIC16C72.
2: The TX/CK and RX/DT multiplexed functions are not enabled on the PIC16C72.

**TABLE 5-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 07h | PORTC | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | uuuu uuuu |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |

Legend: x = unknown, u = unchanged.

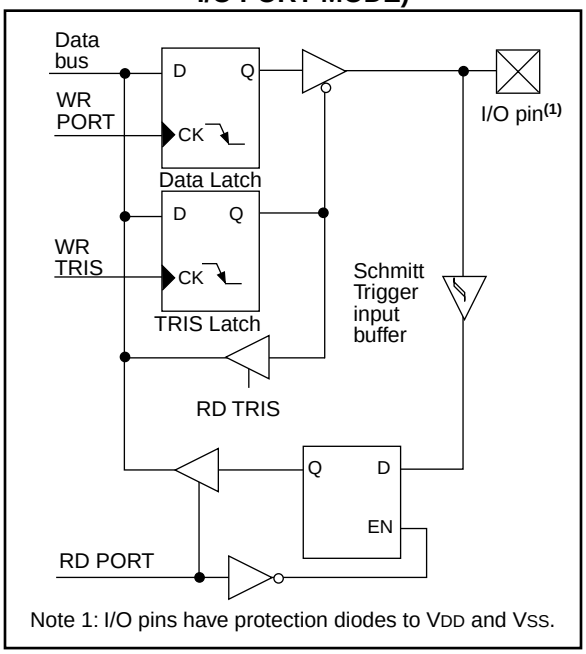
## 5.4 PORTD and TRISD Registers

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

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

PORTD can be configured as an 8-bit wide microprocessor port (parallel slave port) by setting control bit PSPMODE (TRISE<4>). In this mode, the input buffers are TTL.

**FIGURE 5-7: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)**




Note 1: I/O pins have protection diodes to VDD and VSS.

**TABLE 5-7: PORTD FUNCTIONS**

| Name | Bit# | Buffer Type | Function |
|---|---|---|---|
| RD0/PSP0 | bit0 | ST/TTL(1) | Input/output port pin or parallel slave port bit0 |
| RD1/PSP1 | bit1 | ST/TTL(1) | Input/output port pin or parallel slave port bit1 |
| RD2/PSP2 | bit2 | ST/TTL(1) | Input/output port pin or parallel slave port bit2 |
| RD3/PSP3 | bit3 | ST/TTL(1) | Input/output port pin or parallel slave port bit3 |
| RD4/PSP4 | bit4 | ST/TTL(1) | Input/output port pin or parallel slave port bit4 |
| RD5/PSP5 | bit5 | ST/TTL(1) | Input/output port pin or parallel slave port bit5 |
| RD6/PSP6 | bit6 | ST/TTL(1) | Input/output port pin or parallel slave port bit6 |
| RD7/PSP7 | bit7 | ST/TTL(1) | Input/output port pin or parallel slave port bit7 |

Legend: ST = Schmitt Trigger input TTL = TTL input
Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffer when in Parallel Slave Port Mode.

**TABLE 5-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 08h | PORTD | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 | xxxx xxxx | uuuu uuuu |
| 88h | TRISD | PORTD Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 89h | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction Bits | | | 0000 -111 | 0000 -111 |

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

## 5.5 PORTE and TRISE Register

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

PORTE has three pins RE0/$\overline{RD}$/AN5, RE1/$\overline{WR}$/AN6 and RE2/$\overline{CS}$/AN7, which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers.

I/O PORTE becomes control inputs for the microprocessor port when bit PSPMODE (TRISE<4>) is set. In this mode, the user must make sure that the TRISE<2:0> bits are set (pins are configured as digital inputs) and that register ADCON1 is configured for digital I/O. In this mode the input buffers are TTL.

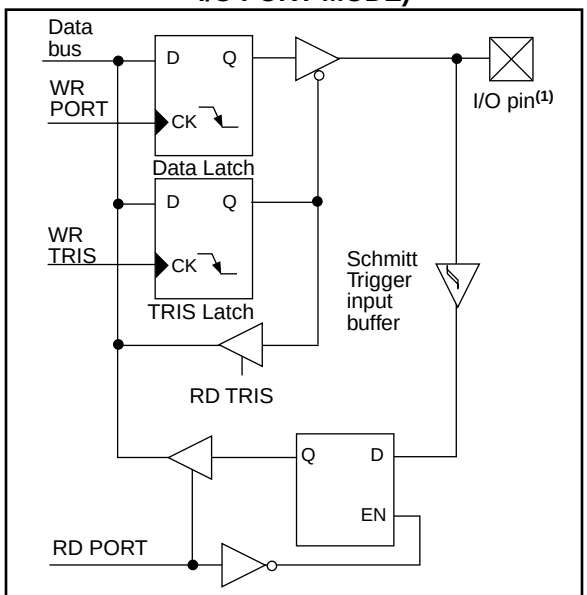
Figure 5-9 shows the TRISE register, which also controls the parallel slave port operation.

PORTE pins are multiplexed with analog inputs. The operation of these pins is selected by control bits in the ADCON1 register. When selected as an analog input, these pins will read as '0's.

TRISE controls the direction of the RE pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs.

**Note:** On a Power-on Reset these pins are configured as analog inputs.

**FIGURE 5-8: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)**




Note 1: I/O pins have protection diodes to VDD and VSS.

**FIGURE 5-9: TRISE REGISTER (ADDRESS 89h)**

| R-0 | R-0 | R/W-0 | R/W-0 | U-0 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| IBF | OBF | IBOV | PSPMODE | — | bit2 | bit1 | bit0 |
| bit7 | | | | | | | bit0 |

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

bit 7 : **IBF:** Input Buffer Full Status bit
1 = A word has been received and is waiting to be read by the CPU
0 = No word has been received

bit 6: **OBF**: Output Buffer Full Status bit
1 = The output buffer still holds a previously written word
0 = The output buffer has been read

bit 5: **IBOV**: Input Buffer Overflow Detect bit (in microprocessor mode)
1 = A write occurred when a previously input word has not been read (must be cleared in software)
0 = No overflow occurred

bit 4: **PSPMODE**: Parallel Slave Port Mode Select bit
1 = Parallel slave port mode
0 = General purpose I/O mode

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

**PORTE Data Direction Bits**

bit 2: **Bit2**: Direction Control bit for pin RE2/$\overline{CS}$/AN7
1 = Input
0 = Output

bit 1: **Bit1**: Direction Control bit for pin RE1/$\overline{WR}$/AN6
1 = Input
0 = Output

bit 0: **Bit0**: Direction Control bit for pin RE0/$\overline{RD}$/AN5
1 = Input
0 = Output

**TABLE 5-9: PORTE FUNCTIONS**

| Name | Bit# | Buffer Type | Function |
|---|---|---|---|
| RE0/$\overline{RD}$/AN5 | bit0 | ST/TTL(1) | Input/output port pin or read control input in parallel slave port mode or analog input:<br>$\overline{RD}$<br>1 = Not a read operation<br>0 = Read operation. Reads PORTD register (if chip selected) |
| RE1/$\overline{WR}$/AN6 | bit1 | ST/TTL(1) | Input/output port pin or write control input in parallel slave port mode or analog input:<br>$\overline{WR}$<br>1 = Not a write operation<br>0 = Write operation. Writes PORTD register (if chip selected) |
| RE2/$\overline{CS}$/AN7 | bit2 | ST/TTL(1) | Input/output port pin or chip select control input in parallel slave port mode or analog input:<br>$\overline{CS}$<br>1 = Device is not selected<br>0 = Device is selected |

Legend: ST = Schmitt Trigger input TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port Mode.

**TABLE 5-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 09h | PORTE | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |
| 89h | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction Bits | | | 0000 -111 | 0000 -111 |
| 9Fh | ADCON1 | — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 | ---- -000 | ---- -000 |

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

## 5.6 I/O Programming Considerations

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

### 5.6.1 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 5-4 shows the effect of two sequential read-modify-write instructions on an I/O port.

**EXAMPLE 5-4: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT**

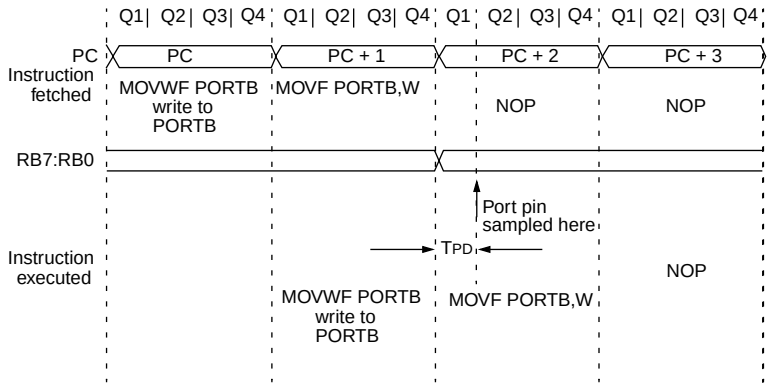
```
;Initial PORT settings: PORTB<7:4> Inputs
;                       PORTB<3:0> Outputs
;PORTB<7:6> have external pull-ups and are
;not connected to other circuitry
;
;                    PORT latch  PORT pins
;                    ----------  ---------
  BCF PORTB, 7      ; 01pp pppp    11pp pppp
  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.

### 5.6.2 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 5-10). 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 5-10: 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.

## 5.7 Parallel Slave Port

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

PORTD operates as an 8-bit wide Parallel Slave Port, or microprocessor port when control bit PSPMODE (TRISE<4>) is set. In slave mode it is asynchronously readable and writable by the external world through $\overline{RD}$ control input pin RE0/$\overline{RD}$/AN5 and $\overline{WR}$ control input pin RE1/$\overline{WR}$/AN6.

It can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PSPMODE enables port pin RE0/$\overline{RD}$/AN5 to be the $\overline{RD}$ input, RE1/$\overline{WR}$/AN6 to be the $\overline{WR}$ input and RE2/$\overline{CS}$/AN7 to be the $\overline{CS}$ (chip select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set) and the A/D port configuration bits PCFG2:PCFG0 (ADCON1<2:0>) must be set, which will configure pins RE2:RE0 as digital I/O.

There are actually two 8-bit latches, one for data-out (from the PIC16/17) and one for data input. The user writes 8-bit data to PORTD data latch and reads data from the port pin latch (note that they have the same address). In this mode, the TRISD register is ignored, since the microprocessor is controlling the direction of data flow.

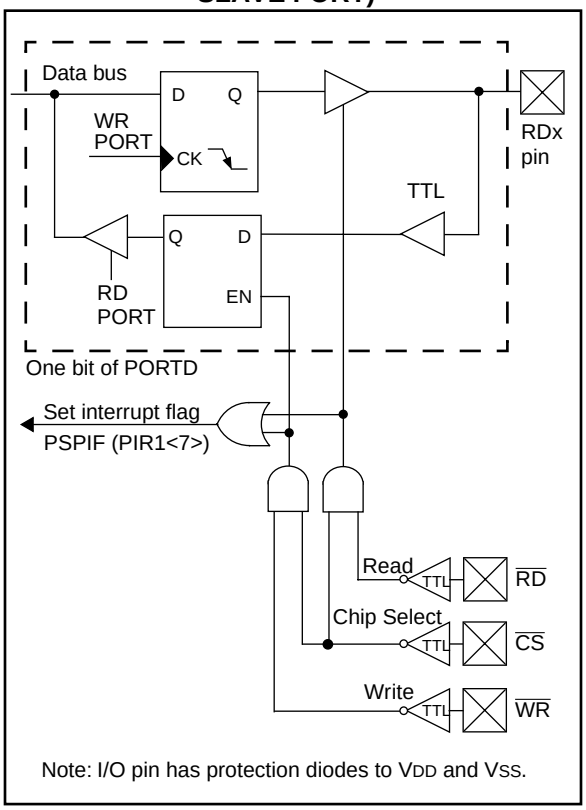
A write to the PSP occurs when both the $\overline{CS}$ and $\overline{WR}$ lines are first detected low. When either the $\overline{CS}$ or $\overline{WR}$ lines become high (level triggered), then the Input Buffer Full status flag bit IBF (TRISE<7>) is set on the Q4 clock cycle, following the next Q2 cycle, to signal the write is complete (Figure 5-12). The interrupt flag bit PSPIF (PIR1<7>) is also set on the same Q4 clock cycle. IBF can only be cleared by reading the PORTD input latch. The input Buffer Overflow status flag bit IBOV (TRISE<5>) is set if a second write to the Parallel Slave Port is attempted when the previous byte has not been read out of the buffer.

A read from the PSP occurs when both the $\overline{CS}$ and $\overline{RD}$ lines are first detected low. The Output Buffer Full status flag bit OBF (TRISE<6>) is cleared immediately (Figure 5-13) indicating that the PORTD latch is waiting to be read by the external bus. When either the $\overline{CS}$ or $\overline{RD}$ pin becomes high (level triggered), the interrupt flag bit PSPIF is set on the Q4 clock cycle, following the next Q2 cycle, indicating that the read is complete. OBF remains low until data is written to PORTD by the user firmware.

When not in Parallel Slave Port mode, the IBF and OBF bits are held clear. However, if flag bit IBOV was previously set, it must be cleared in firmware.

An interrupt is generated and latched into flag bit PSPIF when a read or write operation is completed. PSPIF must be cleared by the user in firmware and the interrupt can be disabled by clearing the interrupt enable bit PSPIE (PIE1<7>).

**FIGURE 5-11: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)**




Note: I/O pin has protection diodes to $V_{DD}$ and $V_{SS}$.

**FIGURE 5-12: PARALLEL SLAVE PORT WRITE WAVEFORMS**

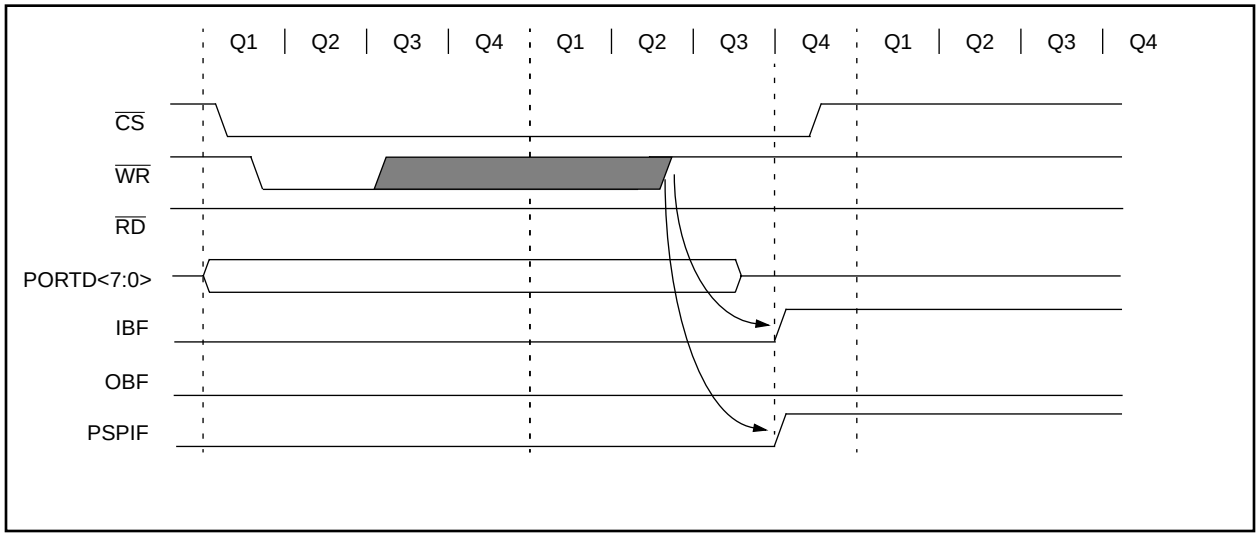



**FIGURE 5-13: PARALLEL SLAVE PORT READ WAVEFORMS**

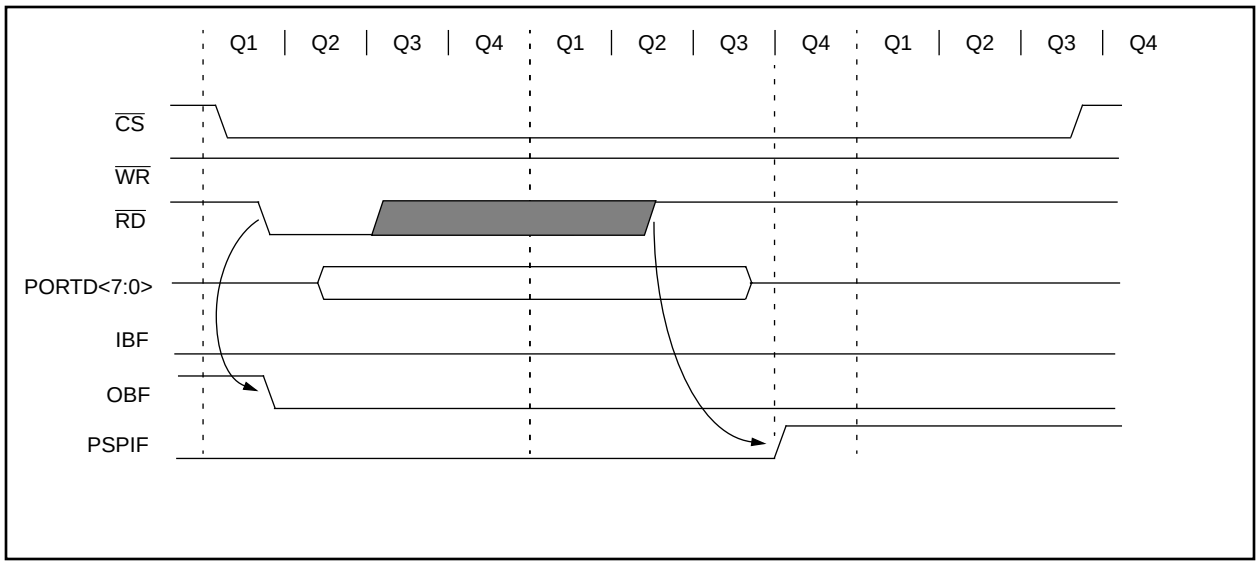



**TABLE 5-11: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 08h | PORTD | Port data latch when written: Port pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 09h | PORTE | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |
| 89h | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction Bits | | | 0000 -111 | 0000 -111 |
| 0Ch | PIR1 | PSPIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 9Fh | ADCON1 | — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 | ---- -000 | ---- -000 |

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Parallel Slave Port.

**NOTES:**

## 6.0 OVERVIEW OF TIMER MODULES

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

The PIC16C72, PIC16C73/73A, PIC16C74/74A, PIC16C76/77 each have three timer modules.

Each module can generate an interrupt to indicate that an event has occurred (i.e. timer overflow). Each of these modules is explained in full detail in the following sections. The timer modules are:

- Timer0 Module (Section 7.0)
- Timer1 Module (Section 8.0)
- Timer2 Module (Section 9.0)

### 6.1 Timer0 Overview

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

The Timer0 module is a simple 8-bit overflow counter. The clock source can be either the internal system clock (Fosc/4) or an external clock. When the clock source is an external clock, the Timer0 module can be selected to increment on either the rising or falling edge.

The Timer0 module also has a programmable prescaler option. This prescaler can be assigned to either the Timer0 module or the Watchdog Timer. Bit PSA (OPTION<3>) assigns the prescaler, and bits PS2:PS0 (OPTION<2:0>) determine the prescaler value. Timer0 can increment at the following rates: 1:1 (when prescaler assigned to Watchdog timer), 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, and 1:256 (Timer0 only).

Synchronization of the external clock occurs after the prescaler. When the prescaler is used, the external clock frequency may be higher then the device's frequency. The maximum frequency is 50 MHz, given the high and low time requirements of the clock.

### 6.2 Timer1 Overview

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

Timer1 is a 16-bit timer/counter. The clock source can be either the internal system clock (Fosc/4), an external clock, or an external crystal. Timer1 can operate as either a timer or a counter. When operating as a counter (external clock source), the counter can either operate synchronized to the device or asynchronously to the device. Asynchronous operation allows Timer1 to operate during sleep, which is useful for applications that require a real-time clock as well as the power savings of SLEEP mode.

Timer1 also has a prescaler option which allows Timer1 to increment at the following rates: 1:1, 1:2, 1:4, and 1:8. Timer1 can be used in conjunction with the Capture/Compare/PWM module. When used with a CCP module, Timer1 is the time-base for 16-bit Capture or the 16-bit Compare and must be synchronized to the device.

### 6.3 Timer2 Overview

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

Timer2 is an 8-bit timer with a programmable prescaler and postscaler, as well as an 8-bit period register (PR2). Timer2 can be used with the CCP1 module (in PWM mode) as well as the Baud Rate Generator for the Synchronous Serial Port (SSP). The prescaler option allows Timer2 to increment at the following rates: 1:1, 1:4, 1:16.

The postscaler allows the TMR2 register to match the period register (PR2) a programmable number of times before generating an interrupt. The postscaler can be programmed from 1:1 to 1:16 (inclusive).

### 6.4 CCP Overview

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

The CCP module(s) can operate in one of these three modes: 16-bit capture, 16-bit compare, or up to 10-bit Pulse Width Modulation (PWM).

Capture mode captures the 16-bit value of TMR1 into the CCPRxH:CCPRxL register pair. The capture event can be programmed for either the falling edge, rising edge, fourth rising edge, or the sixteenth rising edge of the CCPx pin.

Compare mode compares the TMR1H:TMR1L register pair to the CCPRxH:CCPRxL register pair. When a match occurs an interrupt can be generated, and the output pin CCPx can be forced to given state (High or Low), TMR1 can be reset (CCP1), or TMR1 reset and start A/D conversion (CCP2). This depends on the control bits CCPxM3:CCPxM0.

PWM mode compares the TMR2 register to a 10-bit duty cycle register (CCPRxH:CCPRxL<5:4>) as well as to an 8-bit period register (PR2). When the TMR2 register = Duty Cycle register, the CCPx pin will be forced low. When TMR2 = PR2, TMR2 is cleared to 00h, an interrupt can be generated, and the CCPx pin (if an output) will be forced high.

**NOTES:**

## 7.0 TIMER0 MODULE

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

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 7-1 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 7-2 and Figure 7-3). 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 7.2.

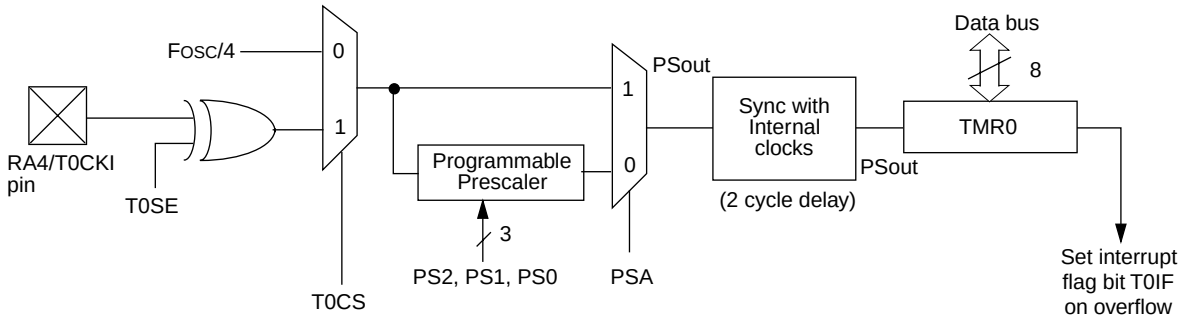
The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler assignment is controlled in software by control bit PSA (OPTION<3>). Clearing bit PSA will assign the prescaler to the Timer0 module. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable. Section 7.3 details the operation of the prescaler.

### 7.1 Timer0 Interrupt

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

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 7-4 for Timer0 interrupt timing.

**FIGURE 7-1: TIMER0 BLOCK DIAGRAM**




Note 1: T0CS, T0SE, PSA, PS2:PS0 (OPTION<5:0>).
2: The prescaler is shared with Watchdog Timer (refer to Figure 7-6 for detailed block diagram).

**FIGURE 7-2: TIMER0 TIMING: INTERNAL CLOCK/NO PRESCALE**

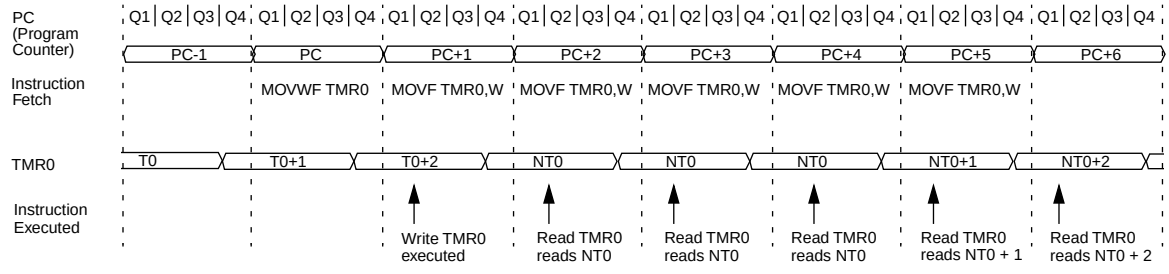


**FIGURE 7-3: TIMER0 TIMING: INTERNAL CLOCK/PRESCALE 1:2**

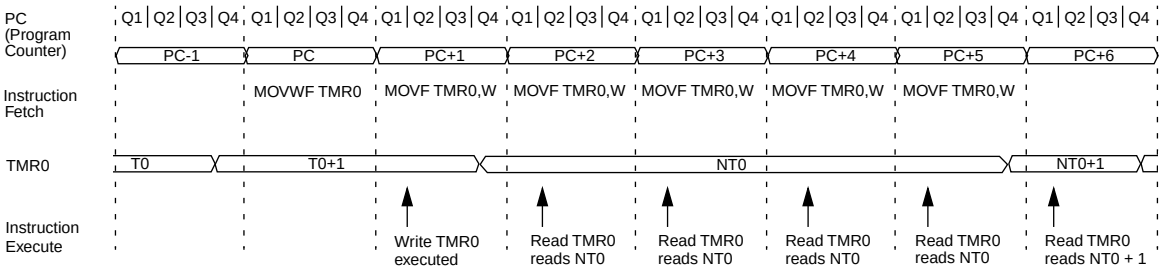
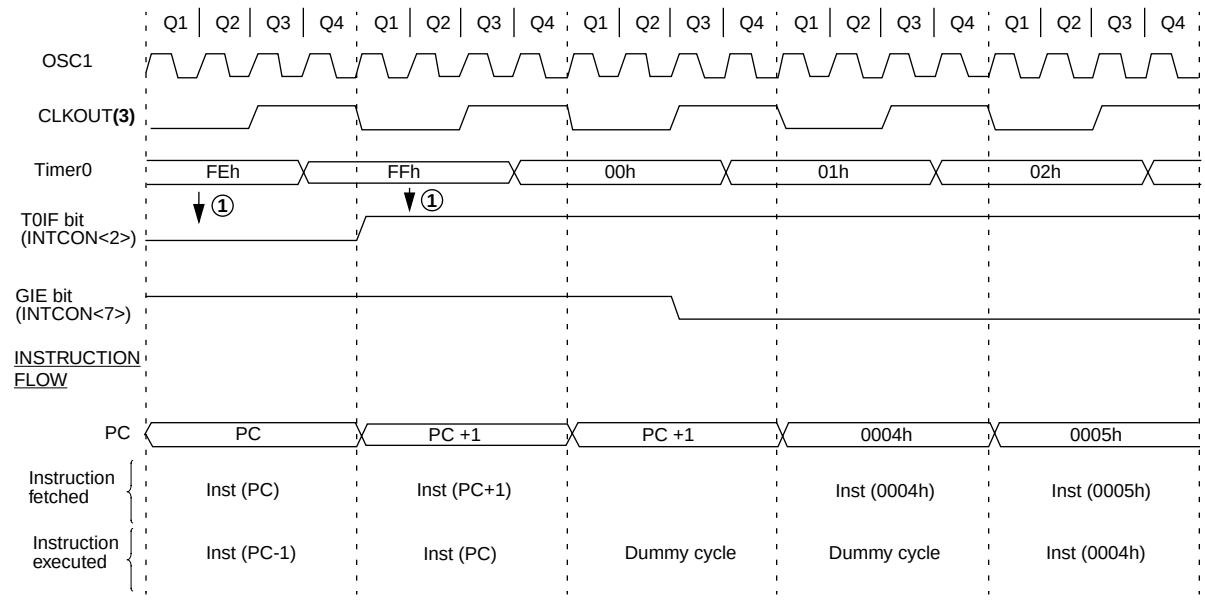



**FIGURE 7-4: TIMER0 INTERRUPT TIMING**




Note 1: Interrupt flag bit T0IF is sampled here (every Q1).
2: Interrupt latency = 4Tcy where Tcy = instruction cycle time.
3: CLKOUT is available only in RC oscillator mode.

## 7.2 Using Timer0 with an External Clock

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

When an external clock input is used for Timer0, it must meet certain requirements. The requirements ensure the external clock can be synchronized with the internal phase clock (TOSC). Also, there is a delay in the actual incrementing of Timer0 after synchronization.

### 7.2.1 EXTERNAL CLOCK SYNCHRONIZATION

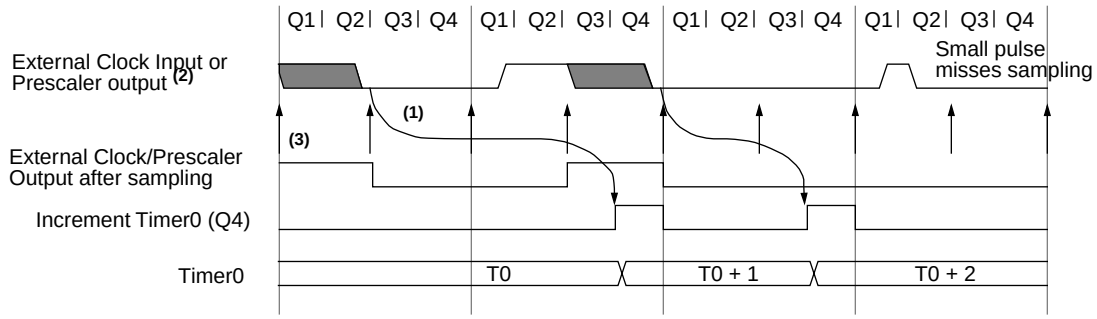
When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 7-5). Therefore, it is necessary for T0CKI to be high for at least 2Tosc (and a small RC delay of 20 ns) and low for at least 2Tosc (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple-counter type prescaler so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple-counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4Tosc (and a small RC delay of 40 ns) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

### 7.2.2 TMR0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the Timer0 module is actually incremented. Figure 7-5 shows the delay from the external clock edge to the timer incrementing.

**FIGURE 7-5: TIMER0 TIMING WITH EXTERNAL CLOCK**




Note 1: Delay from clock input change to Timer0 increment is 3Tosc to 7Tosc. (Duration of Q = Tosc). Therefore, the error in measuring the interval between two edges on Timer0 input = ±4Tosc max.
2: External clock if no prescaler selected, Prescaler output otherwise.
3: The arrows indicate the points in time where sampling occurs.

## 7.3 Prescaler

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

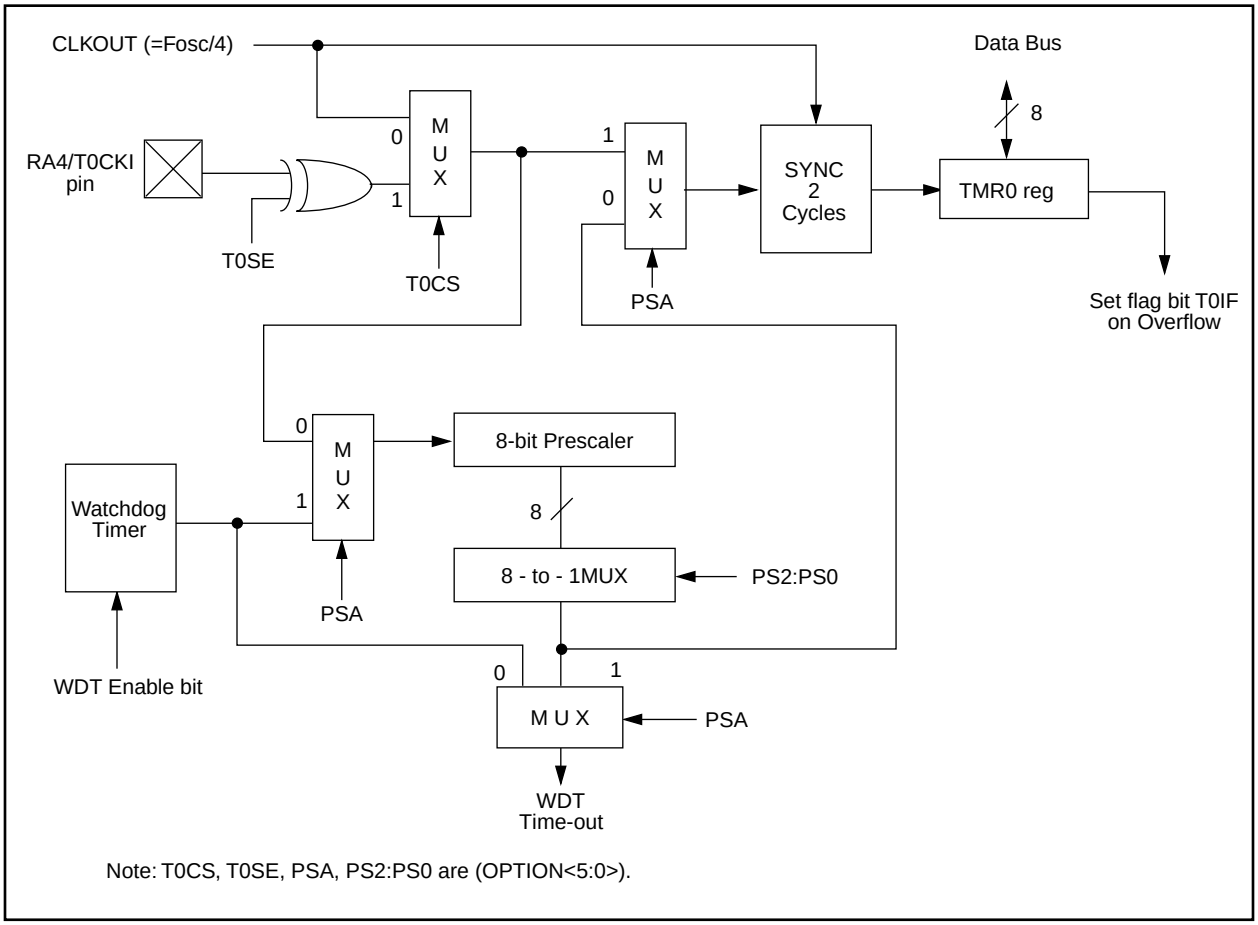
An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer, respectively (Figure 7-6). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet. Note that there is only one prescaler available which is mutually exclusively shared between the Timer0 module and the Watchdog Timer. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer, and vice-versa.

The PSA and PS2:PS0 bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g. `CLRF 1, MOVWF 1, BSF 1,x`....etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

| **Note:** | Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count, but will not change the prescaler assignment. |
|---|---|

**FIGURE 7-6: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**




Note: T0CS, T0SE, PSA, PS2:PS0 are (OPTION<5:0>).

### 7.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control, i.e., it can be changed "on the fly" during program execution.

| **Note:** | To avoid an unintended device RESET, the following instruction sequence (shown in Example 7-1) must be executed when changing the prescaler assignment from Timer0 to the WDT. This sequence must be followed even if the WDT is disabled. |
|---|---|

**EXAMPLE 7-1: CHANGING PRESCALER (TIMER0→WDT)**

Lines 2 and 3 do NOT have to be included if the final desired prescale value is other than 1:1. If 1:1 is final desired value, then a temporary prescale value is set in lines 2 and 3 and the final prescale value will be set in lines 10 and 11.

```
1)  BSF    STATUS, RP0   ;Bank 1
2)  MOVLW  b'xx0x0xxx'   ;Select clock source and prescale value of
3)  MOVWF  OPTION_REG    ;other than 1:1
4)  BCF    STATUS, RP0   ;Bank 0
5)  CLRF   TMR0          ;Clear TMR0 and prescaler
6)  BSF    STATUS, RP1   ;Bank 1
7)  MOVLW  b'xxxx1xxx'   ;Select WDT, do not change prescale value
8)  MOVWF  OPTION_REG    ;
9)  CLRWDT               ;Clears WDT and prescaler
10) MOVLW  b'xxxx1xxx'   ;Select new prescale value and WDT
11) MOVWF  OPTION_REG    ;
12) BCF    STATUS, RP0   ;Bank 0
```

To change prescaler from the WDT to the Timer0 module use the sequence shown in Example 7-2.

**EXAMPLE 7-2: CHANGING PRESCALER (WDT→TIMER0)**

```
CLRWDT              ;Clear WDT and prescaler
BSF     STATUS, RP0 ;Bank 1
MOVLW   b'xxxx0xxx' ;Select TMR0, new prescale value and
MOVWF   OPTION_REG  ;clock source
BCF     STATUS, RP0 ;Bank 0
```

**TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER0**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 01h,101h | TMR0 | Timer0 module's register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Bh,8Bh, 10Bh,18Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 81h,181h | OPTION | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | --11 1111 | --11 1111 |

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

**NOTES:**

## 8.0 TIMER1 MODULE

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

The Timer1 module is a 16-bit timer/counter consisting of two 8-bit registers (TMR1H and TMR1L) which are readable and writable. The TMR1 Register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 Interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit TMR1IF (PIR1<0>). This interrupt can be enabled/disabled by setting/clearing TMR1 interrupt enable bit TMR1IE (PIE1<0>).

Timer1 can operate in one of two modes:

- As a timer
- As a counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>).

In timer mode, Timer1 increments every instruction cycle. In counter mode, it increments on every rising edge of the external clock input.

Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON<0>).

Timer1 also has an internal “reset input”. This reset can be generated by either of the two CCP modules (Section 10.0). Figure 8-1 shows the Timer1 control register.

For the PIC16C72/73A/74A/76/77, when the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI/CCP2 and RC0/T1OSO/T1CKI pins become inputs. That is, the TRISC<1:0> value is ignored.

For the PIC16C73/74, when the Timer1 oscillator is enabled (T1OSCEN is set), RC1/T1OSI/CCP2 pin becomes an input, however the RC0/T1OSO/T1CKI pin will have to be configured as an input by setting the TRISC<0> bit.

**FIGURE 8-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON |
| bit7 | | | | | | | bit0 |

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

bit 7-6: **Unimplemented:** Read as '0'

bit 5-4: **T1CKPS1:T1CKPS0**: Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3: **T1OSCEN**: Timer1 Oscillator Enable Control bit
1 = Oscillator is enabled
0 = Oscillator is shut off
Note: The oscillator inverter and feedback resistor are turned off to eliminate power drain

bit 2: **$\overline{\text{T1SYNC}}$**: Timer1 External Clock Input Synchronization Control bit

TMR1CS = 1
1 = Do not synchronize external clock input
0 = Synchronize external clock input

TMR1CS = 0
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1: **TMR1CS**: Timer1 Clock Source Select bit
1 = External clock from pin RC0/T1OSO/T1CKI (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0: **TMR1ON**: Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

## 8.1 Timer1 Operation in Timer Mode

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

Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is FOSC/4. The synchronize control bit $\overline{\text{T1SYNC}}$ (T1CON<2>) has no effect since the internal clock is always in sync.

## 8.2 Timer1 Operation in Synchronized Counter Mode

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

Counter mode is selected by setting bit TMR1CS. In this mode the timer increments on every rising edge of clock input on pin RC1/T1OSI/CCP2 when bit T1OSCEN is set or pin RC0/T1OSO/T1CKI when bit T1OSCEN is cleared.

If $\overline{\text{T1SYNC}}$ is cleared, then the external clock input is synchronized with internal phase clocks. The synchronization is done after the prescaler stage. The prescaler stage is an asynchronous ripple-counter.

In this configuration, during SLEEP mode, Timer1 will not increment even if the external clock is present, since the synchronization circuit is shut off. The prescaler however will continue to increment.

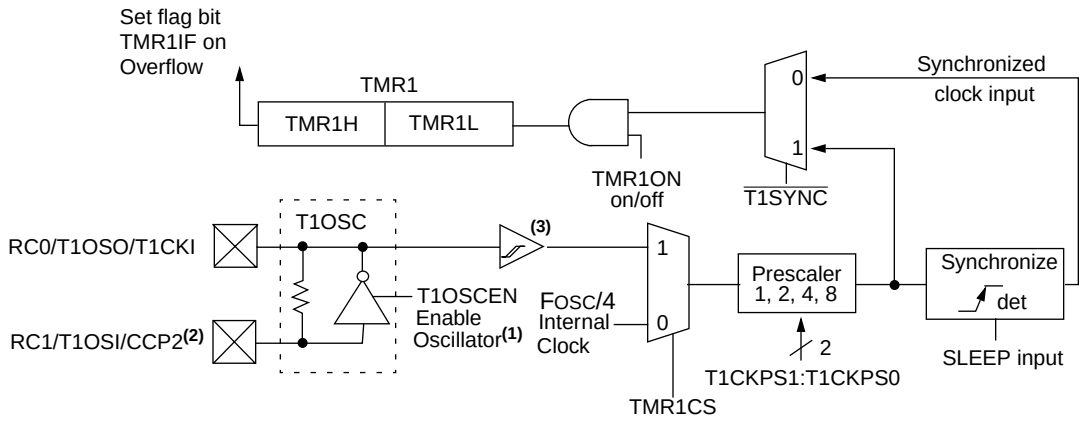
### 8.2.1 EXTERNAL CLOCK INPUT TIMING FOR SYNCHRONIZED COUNTER MODE

When an external clock input is used for Timer1 in synchronized counter mode, it must meet certain requirements. The external clock requirement is due to internal phase clock (Tosc) synchronization. Also, there is a delay in the actual incrementing of TMR1 after synchronization.

When the prescaler is 1:1, the external clock input is the same as the prescaler output. The synchronization of T1CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, it is necessary for T1CKI to be high for at least 2Tosc (and a small RC delay of 20 ns) and low for at least 2Tosc (and a small RC delay of 20 ns). Refer to the appropriate electrical specifications, parameters 45, 46, and 47.

When a prescaler other than 1:1 is used, the external clock input is divided by the asynchronous ripple-counter type prescaler so that the prescaler output is symmetrical. In order for the external clock to meet the sampling requirement, the ripple-counter must be taken into account. Therefore, it is necessary for T1CKI to have a period of at least 4Tosc (and a small RC delay of 40 ns) divided by the prescaler value. The only requirement on T1CKI high and low time is that they do not violate the minimum pulse width requirements of 10 ns). Refer to the appropriate electrical specifications, parameters 40, 42, 45, 46, and 47.

**FIGURE 8-2: TIMER1 BLOCK DIAGRAM**




Note 1: When the T1OSCEN bit is cleared, the inverter and feedback resistor are turned off. This eliminates power drain.
2: The CCP2 module is not implemented in the PIC16C72.
3: For the PIC16C73 and PIC16C74, the Schmitt Trigger is not implemented in external clock mode.

## 8.3 Timer1 Operation in Asynchronous Counter Mode

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

If control bit $\overline{\text{T1SYNC}}$ (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. The timer will continue to run during SLEEP and can generate an interrupt on overflow which will wake-up the processor. However, special precautions in software are needed to read/write the timer (Section 8.3.2).

In asynchronous counter mode, Timer1 can not be used as a time-base for capture or compare operations.

### 8.3.1 EXTERNAL CLOCK INPUT TIMING WITH UNSYNCHRONIZED CLOCK

If control bit $\overline{\text{T1SYNC}}$ is set, the timer will increment completely asynchronously. The input clock must meet certain minimum high time and low time requirements. Refer to the appropriate Electrical Specifications Section, timing parameters 45, 46, and 47.

### 8.3.2 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running, from an external asynchronous clock, will guarantee a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself poses certain problems since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers while the register is incrementing. This may produce an unpredictable value in the timer register.

Reading the 16-bit value requires some care. Example 8-1 is an example routine to read the 16-bit timer value. This is useful if the timer cannot be stopped.

**EXAMPLE 8-1: READING A 16-BIT FREE-RUNNING TIMER**

```
; All interrupts are disabled
   MOVF   TMR1H, W  ;Read high byte
   MOVWF  TMPH      ;
   MOVF   TMR1L, W  ;Read low byte
   MOVWF  TMPL      ;
   MOVF   TMR1H, W  ;Read high byte
   SUBWF  TMPH,  W  ;Sub 1st read
                    ; with 2nd read
   BTFSC  STATUS,Z  ;Is result = 0
   GOTO   CONTINUE  ;Good 16-bit read
;
; TMR1L may have rolled over between the read
; of the high and low bytes. Reading the high
; and low bytes now will read a good value.
;
   MOVF   TMR1H, W  ;Read high byte
   MOVWF  TMPH      ;
   MOVF   TMR1L, W  ;Read low byte
   MOVWF  TMPL      ;
; Re-enable the Interrupt (if required)
CONTINUE            ;Continue with your code
```

## 8.4 Timer1 Oscillator

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

A crystal oscillator circuit is built in between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting control bit T1OSCEN (T1CON<3>). The oscillator is a low power oscillator rated up to 200 kHz. It will continue to run during SLEEP. It is primarily intended for a 32 kHz crystal. Table 8-1 shows the capacitor selection for the Timer1 oscillator.

The Timer1 oscillator is identical to the LP oscillator. The user must provide a software time delay to ensure proper oscillator start-up.

**TABLE 8-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR**

| Osc Type | Freq | C1 | C2 |
|---|---|---|---|
| LP | 32 kHz | 33 pF | 33 pF |
| | 100 kHz | 15 pF | 15 pF |
| | 200 kHz | 15 pF | 15 pF |
| **These values are for design guidance only.** | | | |
| **Crystals Tested:** | | | |
| 32.768 kHz | Epson C-001R32.768K-A | | ± 20 PPM |
| 100 kHz | Epson C-2 100.00 KC-P | | ± 20 PPM |
| 200 kHz | STD XTL 200.000 kHz | | ± 20 PPM |

Note 1: Higher capacitance increases the stability of oscillator but also increases the start-up time.
2: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

### 8.5 Resetting Timer1 using a CCP Trigger Output

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

The CCP2 module is not implemented on the PIC16C72 device.

If the CCP1 or CCP2 module is configured in compare mode to generate a “special event trigger" (CCP1M3:CCP1M0 = 1011), this signal will reset Timer1.

**Note:** The special event triggers from the CCP1 and CCP2 modules will not set interrupt flag bit TMR1IF (PIR1<0>).

Timer1 must be configured for either timer or synchronized counter mode to take advantage of this feature. If Timer1 is running in asynchronous counter mode, this reset operation may not work.

In the event that a write to Timer1 coincides with a special event trigger from CCP1 or CCP2, the write will take precedence.

In this mode of operation, the CCPRxH:CCPRxL registers pair effectively becomes the period register for Timer1.

### 8.6 Resetting of Timer1 Register Pair (TMR1H, TMR1L)

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

TMR1H and TMR1L registers are not reset to 00h on a POR or any other reset except by the CCP1 and CCP2 special event triggers.

T1CON register is reset to 00h on a Power-on Reset or a Brown-out Reset, which shuts off the timer and leaves a 1:1 prescale. In all other resets, the register is unaffected.

### 8.7 Timer1 Prescaler

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

The prescaler counter is cleared on writes to the TMR1H or TMR1L registers.

**TABLE 8-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER**

| 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(1,2) | 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 |
| 0Eh | TMR1L | Holding register for the Least Significant Byte of the 16-bit TMR1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding register for the Most Significant Byte of the 16-bit TMR1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Timer1 module.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.
2: The PIC16C72 does not have a Parallel Slave Port or a USART, these bits are unimplemented, read as '0'.

## 9.0 TIMER2 MODULE

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

Timer2 is an 8-bit timer with a prescaler and a postscaler. It can be used as the PWM time-base for PWM mode of the CCP module(s). The TMR2 register is readable and writable, and is cleared on any device reset.

The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T2CKPS1:T2CKPS0 (T2CON<1:0>).

The Timer2 module has an 8-bit period register PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon reset.

The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit TMR2IF, (PIR1<1>)).

Timer2 can be shut off by clearing control bit TMR2ON (T2CON<2>) to minimize power consumption.

Figure 9-2 shows the Timer2 control register.

### 9.1 Timer2 Prescaler and Postscaler

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

The prescaler and postscaler counters are cleared when any of the following occurs:

- a write to the TMR2 register
- a write to the T2CON register
- any device reset (Power-on Reset, $\overline{MCLR}$ reset, Watchdog Timer reset, or Brown-out Reset)

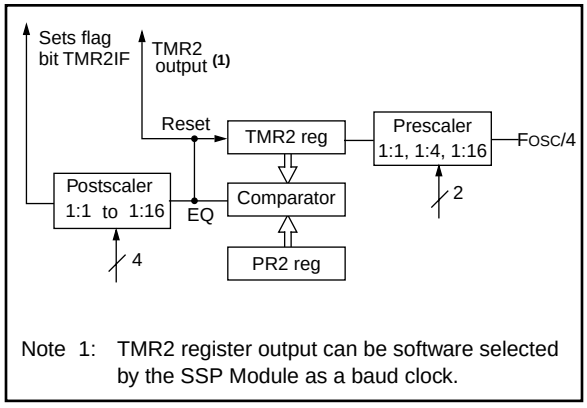
TMR2 is not cleared when T2CON is written.

### 9.2 Output of TMR2

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

The output of TMR2 (before the postscaler) is fed to the Synchronous Serial Port module which optionally uses it to generate shift clock.

**FIGURE 9-1: TIMER2 BLOCK DIAGRAM**




Note 1: TMR2 register output can be software selected by the SSP Module as a baud clock.

**FIGURE 9-2: T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)**

| U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 |
| 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 6-3: **TOUTPS3:TOUTPS0**: Timer2 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2: **TMR2ON**: Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0: **T2CKPS1:T2CKPS0**: Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

**TABLE 9-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER**

| 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(1,2) | 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 |
| 11h | TMR2 | Timer2 module's register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 92h | PR2 | Timer2 Period Register | | | | | | | | 1111 1111 | 1111 1111 |

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by the Timer2 module.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.
2: The PIC16C72 does not have a Parallel Slave Port or a USART, these bits are unimplemented, read as '0'.

## 10.0 CAPTURE/COMPARE/PWM MODULE(s)

| Applicable Devices | | | | | | | |
|---|---|---|---|---|---|---|---|
| 72 | 73 | 73A | 74 | 74A | 76 | 77 | **CCP1** |
| 72 | 73 | 73A | 74 | 74A | 76 | 77 | **CCP2** |

Each CCP (Capture/Compare/PWM) module contains a 16-bit register which can operate as a 16-bit capture register, as a 16-bit compare register or as a PWM master/slave Duty Cycle register. Both the CCP1 and CCP2 modules are identical in operation, with the exception of the operation of the special event trigger. Table 10-1 and Table 10-2 show the resources and interactions of the CCP module(s). In the following sections, the operation of a CCP module is described with respect to CCP1. CCP2 operates the same as CCP1, except where noted.

**CCP1 module:**

Capture/Compare/PWM Register1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. All are readable and writable.

**CCP2 module:**

Capture/Compare/PWM Register2 (CCPR2) is comprised of two 8-bit registers: CCPR2L (low byte) and CCPR2H (high byte). The CCP2CON register controls the operation of CCP2. All are readable and writable.

For use of the CCP modules, refer to the Embedded Control Handbook, *"Using the CCP Modules"* (AN594).

**TABLE 10-1: CCP MODE - TIMER RESOURCE**

| CCP Mode | Timer Resource |
|---|---|
| Capture | Timer1 |
| Compare | Timer1 |
| PWM | Timer2 |

**TABLE 10-2: INTERACTION OF TWO CCP MODULES**

| CCPx Mode | CCPy Mode | Interaction |
|---|---|---|
| Capture | Capture | Same TMR1 time-base. |
| Capture | Compare | The compare should be configured for the special event trigger, which clears TMR1. |
| Compare | Compare | The compare(s) should be configured for the special event trigger, which clears TMR1. |
| PWM | PWM | The PWMs will have the same frequency, and update rate (TMR2 interrupt). |
| PWM | Capture | None |
| PWM | Compare | None |

**FIGURE 10-1: CCP1CON REGISTER (ADDRESS 17h)/CCP2CON REGISTER (ADDRESS 1Dh)**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | CCPxX | CCPxY | CCPxM3 | CCPxM2 | CCPxM1 | CCPxM0 |
| bit7 | | | | | | | bit0 |

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

bit 7-6: **Unimplemented:** Read as '0'

bit 5-4: **CCPxX:CCPxY**: PWM Least Significant bits
Capture Mode: Unused
Compare Mode: Unused
PWM Mode: These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPRxL.

bit 3-0: **CCPxM3:CCPxM0**: CCPx Mode Select bits
0000 = Capture/Compare/PWM off (resets CCPx module)
0100 = Capture mode, every falling edge
0101 = Capture mode, every rising edge
0110 = Capture mode, every 4th rising edge
0111 = Capture mode, every 16th rising edge
1000 = Compare mode, set output on match (CCPxIF bit is set)
1001 = Compare mode, clear output on match (CCPxIF bit is set)
1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)
1011 = Compare mode, trigger special event (CCPxIF bit is set; CCP1 resets TMR1; CCP2 resets TMR1 and starts an A/D conversion (if A/D module is enabled))
11xx = PWM mode

## 10.1 Capture Mode

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

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin RC2/CCP1. An event is defined as:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

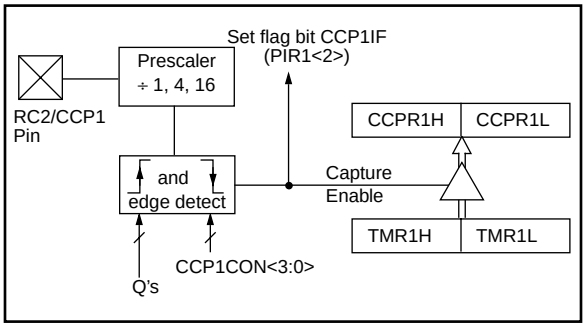
An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit CCP1IF (PIR1<2>) is set. It must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value will be lost.

### 10.1.1 CCP PIN CONFIGURATION

In Capture mode, the RC2/CCP1 pin should be configured as an input by setting the TRISC<2> bit.

| **Note:** | If the RC2/CCP1 is configured as an output, a write to the port can cause a capture condition. |
|---|---|

**FIGURE 10-2: CAPTURE MODE OPERATION BLOCK DIAGRAM**




### 10.1.2 TIMER1 MODE SELECTION

Timer1 must be running in timer mode or synchronized counter mode for the CCP module to use the capture feature. In asynchronous counter mode, the capture operation may not work.

### 10.1.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep bit CCP1IE (PIE1<2>) clear to avoid false interrupts and should clear the flag bit CCP1IF following any such change in operating mode.

### 10.1.4 CCP PRESCALER

There are four prescaler settings, specified by bits CCP1M3:CCP1M0. Whenever the CCP module is turned off, or the CCP module is not in capture mode, the prescaler counter is cleared. This means that any reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared, therefore the first capture may be from a non-zero prescaler. Example 10-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

**EXAMPLE 10-1: CHANGING BETWEEN CAPTURE PRESCALERS**

```
CLRF   CCP1CON      ;Turn CCP module off
MOVLW  NEW_CAPT_PS  ;Load the W reg with
                    ; the new prescaler
                    ; mode value and CCP ON
MOVWF  CCP1CON      ;Load CCP1CON with this
                    ; value
```

## 10.2 Compare Mode

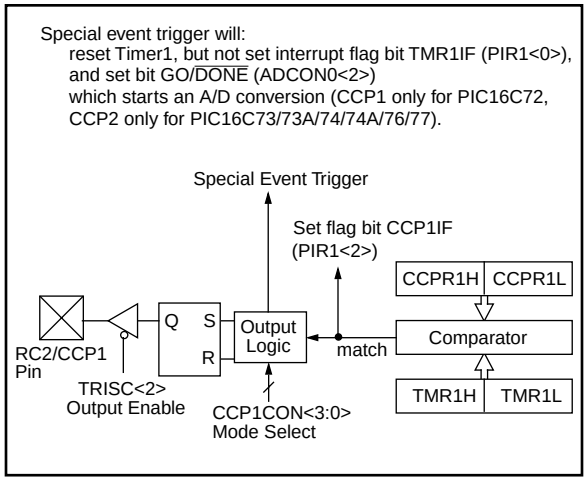
| Applicable Devices | | | | | | |
|---|---|---|---|---|---|---|
| 72 | 73 | 73A | 74 | 74A | 76 | 77 |

In Compare mode, the 16-bit CCPR1 register value is constantly compared against the TMR1 register pair value. When a match occurs, the RC2/CCP1 pin is:

- Driven High
- Driven Low
- Remains Unchanged

The action on the pin is based on the value of control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). At the same time, interrupt flag bit CCP1IF is set.

**FIGURE 10-3: COMPARE MODE OPERATION BLOCK DIAGRAM**




### 10.2.1 CCP PIN CONFIGURATION

The user must configure the RC2/CCP1 pin as an output by clearing the TRISC<2> bit.

| **Note:** | Clearing the CCP1CON register will force the RC2/CCP1 compare output latch to the default low level. This is not the data latch. |
|---|---|

### 10.2.2 TIMER1 MODE SELECTION

Timer1 must be running in Timer mode or Synchronized Counter mode if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

### 10.2.3 SOFTWARE INTERRUPT MODE

When generate software interrupt is chosen the CCP1 pin is not affected. Only a CCP interrupt is generated (if enabled).

### 10.2.4 SPECIAL EVENT TRIGGER

In this mode, an internal hardware trigger is generated which may be used to initiate an action.

The special event trigger output of CCP1 resets the TMR1 register pair. This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

The special trigger output of CCP2 resets the TMR1 register pair, and starts an A/D conversion (if the A/D module is enabled).

For the PIC16C72 only, the special event trigger output of CCP1 resets the TMR1 register pair, and starts an A/D conversion (if the A/D module is enabled).

| **Note:** | The special event trigger from the CCP1and CCP2 modules will not set interrupt flag bit TMR1IF (PIR1<0>). |
|---|---|

## 10.3 PWM Mode

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

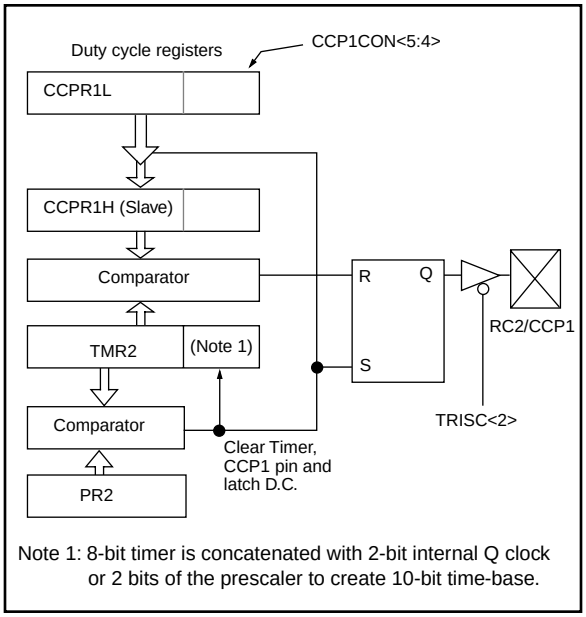
In Pulse Width Modulation (PWM) mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

| **Note:** | Clearing the CCP1CON register will force the CCP1 PWM output latch to the default low level. This is not the PORTC I/O data latch. |
|---|---|

Figure 10-4 shows a simplified block diagram of the CCP module in PWM mode.

For a step by step procedure on how to set up the CCP module for PWM operation, see Section 10.3.3.

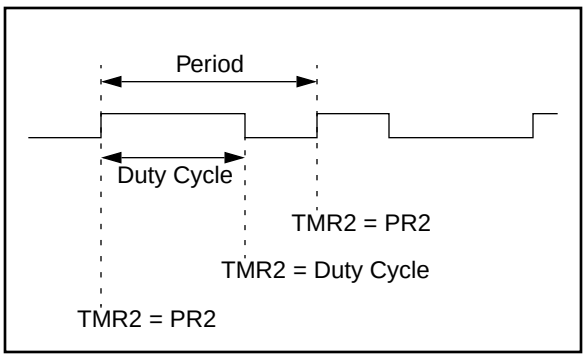
**FIGURE 10-4: SIMPLIFIED PWM BLOCK DIAGRAM**




Note 1: 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time-base.

A PWM output (Figure 10-5) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

**FIGURE 10-5: PWM OUTPUT**




### 10.3.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

$$\text{PWM period} = [(PR2) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 prescale value})$$

PWM frequency is defined as 1 / [PWM period].

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP1 pin is set (exception: if PWM duty cycle = 0%, the CCP1 pin will not be set)
- The PWM duty cycle is latched from CCPR1L into CCPR1H

| **Note:** | The Timer2 postscaler (see Section 9.1) is not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output. |
|---|---|

### 10.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available: the CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

$$\text{PWM duty cycle} = (CCPR1L{:}CCP1CON\langle 5{:}4\rangle) \cdot T_{osc} \cdot (\text{TMR2 prescale value})$$

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.

The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

When the CCPR1H and 2-bit latch match TMR2 concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

Maximum PWM resolution (bits) for a given PWM frequency:

$$= \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)} \text{ bits}$$

| **Note:** | If the PWM duty cycle value is longer than the PWM period the CCP1 pin will not be cleared. |
|---|---|

**EXAMPLE 10-2: PWM PERIOD AND DUTY CYCLE CALCULATION**

Desired PWM frequency is 78.125 kHz,
Fosc = 20 MHz
TMR2 prescale = 1

$1/78.125\ kHz = [(PR2) + 1] \cdot 4 \cdot 1/20\ MHz \cdot 1$

$12.8\ \mu s = [(PR2) + 1] \cdot 4 \cdot 50\ ns \cdot 1$

$PR2 = 63$

Find the maximum resolution of the duty cycle that can be used with a 78.125 kHz frequency and 20 MHz oscillator:

$1/78.125\ kHz = 2^{PWM\ RESOLUTION} \cdot 1/20\ MHz \cdot 1$

$12.8\ \mu s = 2^{PWM\ RESOLUTION} \cdot 50\ ns \cdot 1$

$256 = 2^{PWM\ RESOLUTION}$

$\log(256) = (PWM\ Resolution) \cdot \log(2)$

$8.0 = PWM\ Resolution$

At most, an 8-bit resolution duty cycle can be obtained from a 78.125 kHz frequency and a 20 MHz oscillator, i.e., 0 ≤ CCPR1L:CCP1CON<5:4> ≤ 255. Any value greater than 255 will result in a 100% duty cycle.

In order to achieve higher resolution, the PWM frequency must be decreased. In order to achieve higher PWM frequency, the resolution must be decreased.

Table 10-3 lists example PWM frequencies and resolutions for Fosc = 20 MHz. The TMR2 prescaler and PR2 values are also shown.

### 10.3.3 SET-UP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISC<2> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

**TABLE 10-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz**

| PWM Frequency | 1.22 kHz | 4.88 kHz | 19.53 kHz | 78.12 kHz | 156.3 kHz | 208.3 kHz |
|---|---|---|---|---|---|---|
| Timer Prescaler (1, 4, 16) | 16 | 4 | 1 | 1 | 1 | 1 |
| PR2 Value | 0xFF | 0xFF | 0xFF | 0x3F | 0x1F | 0x17 |
| Maximum Resolution (bits) | 10 | 10 | 10 | 8 | 7 | 5.5 |

**TABLE 10-4: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, AND TIMER1**

| 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(1,2) | ADIF | RCIF(2) | TXIF(2) | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 0Dh(2) | PIR2 | — | — | — | — | — | — | — | CCP2IF | ---- ---0 | ---- ---0 |
| 8Ch | PIE1 | PSPIE(1,2) | ADIE | RCIE(2) | TXIE(2) | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 8Dh(2) | PIE2 | — | — | — | — | — | — | — | CCP2IE | ---- ---0 | ---- ---0 |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 0Eh | TMR1L | Holding register for the Least Significant Byte of the 16-bit TMR1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding register for the Most Significant Byte of the 16-bit TMR1register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{T1SYNC}$ | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |
| 15h | CCPR1L | Capture/Compare/PWM register1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM register1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | — | — | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |
| 1Bh(2) | CCPR2L | Capture/Compare/PWM register2 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Ch(2) | CCPR2H | Capture/Compare/PWM register2 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Dh(2) | CCP2CON | — | — | CCP2X | CCP2Y | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | --00 0000 | --00 0000 |

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by Capture and Timer1.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.
2: The PIC16C72 does not have a Parallel Slave Port, USART or CCP2 module, these bits are unimplemented, read as '0'.

**TABLE 10-5: REGISTERS ASSOCIATED WITH PWM AND TIMER2**

| 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(1,2) | ADIF | RCIF(2) | TXIF(2) | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 0Dh(2) | PIR2 | — | — | — | — | — | — | — | CCP2IF | ---- ---0 | ---- ---0 |
| 8Ch | PIE1 | PSPIE(1,2) | ADIE | RCIE(2) | TXIE(2) | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 8Dh(2) | PIE2 | — | — | — | — | — | — | — | CCP2IE | ---- ---0 | ---- ---0 |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 11h | TMR2 | Timer2 module's register | | | | | | | | 0000 0000 | 0000 0000 |
| 92h | PR2 | Timer2 module's period register | | | | | | | | 1111 1111 | 1111 1111 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 15h | CCPR1L | Capture/Compare/PWM register1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM register1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | — | — | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |
| 1Bh(2) | CCPR2L | Capture/Compare/PWM register2 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Ch(2) | CCPR2H | Capture/Compare/PWM register2 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Dh(2) | CCP2CON | — | — | CCP2X | CCP2Y | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | --00 0000 | --00 0000 |

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by PWM and Timer2.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

2: The PIC16C72 does not have a Parallel Slave Port, USART or CCP2 module, these bits are unimplemented, read as '0'.

# 11.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

## 11.1 SSP Module Overview

The Synchronous Serial Port (SSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be Serial EEPROMs, shift registers, display drivers, A/D converters, etc. The SSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit ($I^2C$)

The SSP module in $I^2C$ mode works the same in all PIC16C7X devices that have an SSP module. However the SSP Module in SPI mode has differences between the PIC16C76/77 and the other PIC16C7X devices.

The register definitions and operational description of SPI mode has been split into two sections because of the differences between the PIC16C76/77 and the other PIC16C7X devices. The default reset values of both the SPI modules is the same regardless of the device:



Refer to Application Note AN578, *"Use of the SSP Module in the $I^2C$ Multi-Master Environment."*

## 11.2 SPI Mode for PIC16C72/73/73A/74/74A

This section contains register definitions and operational characteristics of the SPI module for the PIC16C72, PIC16C73, PIC16C73A, PIC16C74, PIC16C74A.

**FIGURE 11-1: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS 94h)**

| U-0 | U-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| — | — | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF |
| bit7 | | | | | | | bit0 |

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

bit 7-6: **Unimplemented:** Read as '0'

bit 5: **D/$\overline{A}$**: Data/Address bit ($I^2C$ mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4: **P**: Stop bit ($I^2C$ mode only. This bit is cleared when the SSP module is disabled, SSPEN is cleared)
1 = Indicates that a stop bit has been detected last (this bit is '0' on RESET)
0 = Stop bit was not detected last

bit 3: **S**: Start bit ($I^2C$ mode only. This bit is cleared when the SSP module is disabled, SSPEN is cleared)
1 = Indicates that a start bit has been detected last (this bit is '0' on RESET)
0 = Start bit was not detected last

bit 2: **R/$\overline{W}$**: Read/Write bit information ($I^2C$ mode only)
This bit holds the R/W bit information following the last address match. This bit is valid from the address match to the next start bit, stop bit, or $\overline{ACK}$ bit.
1 = Read
0 = Write

bit 1: **UA**: Update Address (10-bit $I^2C$ mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0: **BF**: Buffer Full Status bit

Receive (SPI and $I^2C$ modes)
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty

Transmit ($I^2C$ mode only)
1 = Transmit in progress, SSPBUF is full
0 = Transmit complete, SSPBUF is empty

**FIGURE 11-2: SSPCON: SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit7 | | | | | | | bit0 |

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

bit 7: **WCOL**: Write Collision Detect bit
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6: **SSPOV**: Receive Overflow Detect bit

In SPI mode
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR register is lost. Overflow can only occur in slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In master mode the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
0 = No overflow

In $I^2C$ mode
1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in transmit mode. SSPOV must be cleared in software in either mode.
0 = No overflow

bit 5: **SSPEN**: Synchronous Serial Port Enable bit

In SPI mode
1 = Enables serial port and configures SCK, SDO, and SDI as serial port pins
0 = Disables serial port and configures these pins as I/O port pins

In $I^2C$ mode
1 = Enables the serial port and configures the SDA and SCL pins as serial port pins
0 = Disables serial port and configures these pins as I/O port pins
In both modes, when enabled, these pins must be properly configured as input or output.

bit 4: **CKP**: Clock Polarity Select bit

In SPI mode
1 = Idle state for clock is a high level. Transmit happens on falling edge, receive on rising edge.
0 = Idle state for clock is a low level. Transmit happens on rising edge, receive on falling edge.

In $I^2C$ mode
SCK release control
1 = Enable clock
0 = Holds clock low (clock stretch) (Used to ensure data setup time)

bit 3-0: **SSPM3:SSPM0**: Synchronous Serial Port Mode Select bits
0000 = SPI master mode, clock = Fosc/4
0001 = SPI master mode, clock = Fosc/16
0010 = SPI master mode, clock = Fosc/64
0011 = SPI master mode, clock = TMR2 output/2
0100 = SPI slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.
0101 = SPI slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin.
0110 = $I^2C$ slave mode, 7-bit address
0111 = $I^2C$ slave mode, 10-bit address
1011 = $I^2C$ firmware controlled Master Mode (slave idle)
1110 = $I^2C$ slave mode, 7-bit address with start and stop bit interrupts enabled
1111 = $I^2C$ slave mode, 10-bit address with start and stop bit interrupts enabled

### 11.2.1 OPERATION OF SSP MODULE IN SPI MODE

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

The SPI mode allows 8-bits of data to be synchronously transmitted and received simultaneously. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO)
- Serial Data In (SDI)
- Serial Clock (SCK)

Additionally a fourth pin may be used when in a slave mode of operation:

- Slave Select ($\overline{SS}$)

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits in the SSPCON register (SSPCON<5:0>). These control bits allow the following to be specified:

- Master Mode (SCK is the clock output)
- Slave Mode (SCK is the clock input)
- Clock Polarity (Output/Input data on the Rising/Falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select Mode (Slave mode only)

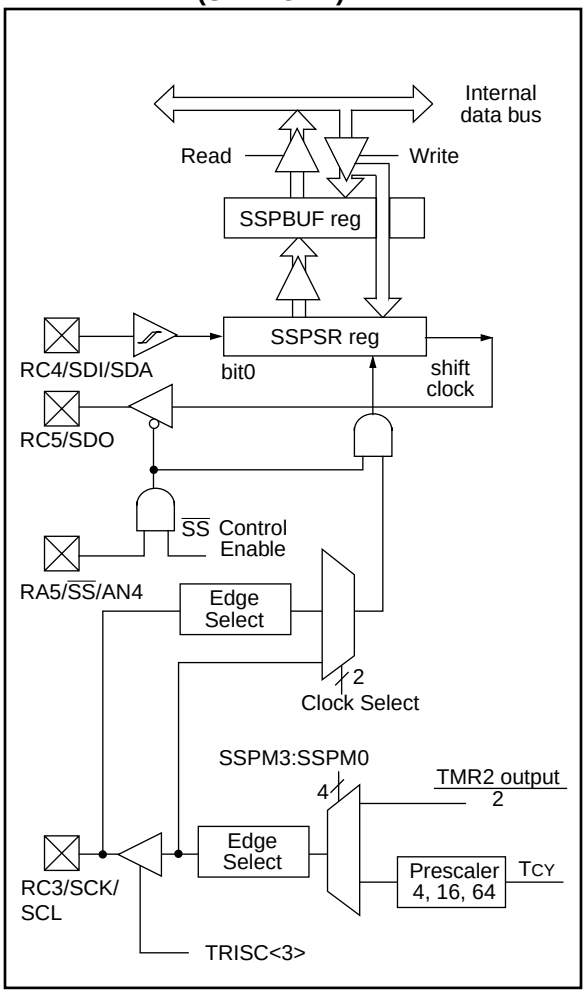
The SSP consists of a transmit/receive Shift Register (SSPSR) and a Buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR, until the received data is ready. Once the 8-bits of data have been received, that byte is moved to the SSPBUF register. Then the Buffer Full bit, BF (SSPSTAT<0>) and flag bit SSPIF are set. This double buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored, and the write collision detect bit, WCOL (SSPCON<7>) will be set. User software must clear bit WCOL so that it can be determined if the following write(s) to the SSPBUF completed successfully. When the application software is expecting to receive valid data, the SSPBUF register should be read before the next byte of data to transfer is written to the SSPBUF register. The Buffer Full bit BF (SSPSTAT<0>) indicates when the SSPBUF register has been loaded with the received data (transmission is complete). When the SSPBUF is read, bit BF is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally the SSP Interrupt is used to determine when the transmission/reception has completed. The SSPBUF register must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 11-1 shows the loading of the SSPBUF (SSPSR) register for data transmission. The shaded instruction is only required if the received data is meaningful.

**EXAMPLE 11-1: LOADING THE SSPBUF (SSPSR) REGISTER**

```
     BSF   STATUS, RP0    ;Specify Bank 1
LOOP BTFSS SSPSTAT, BF    ;Has data been
                          ;received
                          ;(transmit
                          ;complete)?
     GOTO  LOOP           ;No
     BCF   STATUS, RP0    ;Specify Bank 0
     MOVF  SSPBUF, W      ;W reg = contents
                          ;of SSPBUF
     MOVWF RXDATA         ;Save in user RAM
     MOVF  TXDATA, W      ;W reg = contents
                          ; of TXDATA
     MOVWF SSPBUF         ;New data to xmit
```

The block diagram of the SSP module, when in SPI mode (Figure 11-3), shows that the SSPSR register is not directly readable or writable, and can only be accessed from addressing the SSPBUF register. Additionally, the SSP status register (SSPSTAT) indicates the various status conditions.

**FIGURE 11-3: SSP BLOCK DIAGRAM (SPI MODE)**



To enable the serial port, SSP enable bit SSPEN (SSPCON<5>) must be set. To reset or reconfigure SPI mode, clear enable bit SSPEN, re-initialize SSPCON register, and then set enable bit SSPEN. This configures the SDI, SDO, SCK, and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, they must have their data direction bits (in the TRIS register) appropriately programmed. That is:

- SDI must have TRISC<4> set
- SDO must have TRISC<5> cleared
- SCK (Master mode) must have TRISC<3> cleared
- SCK (Slave mode) must have TRISC<3> set
- $\overline{SS}$ must have TRISA<5> set (if implemented)

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value. An example would be in master mode where you are only sending data (to a display driver), then both SDI and $\overline{SS}$ could be used as general purpose outputs by clearing their corresponding TRIS register bits.

Figure 11-4 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge, and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data — Slave sends dummy data
- Master sends data — Slave sends data
- Master sends dummy data — Slave sends data

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2) is to broadcast data by the software protocol.

In master mode the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SCK output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "line activity monitor" mode.

In slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched interrupt flag bit SSPIF (PIR1<3>) is set.

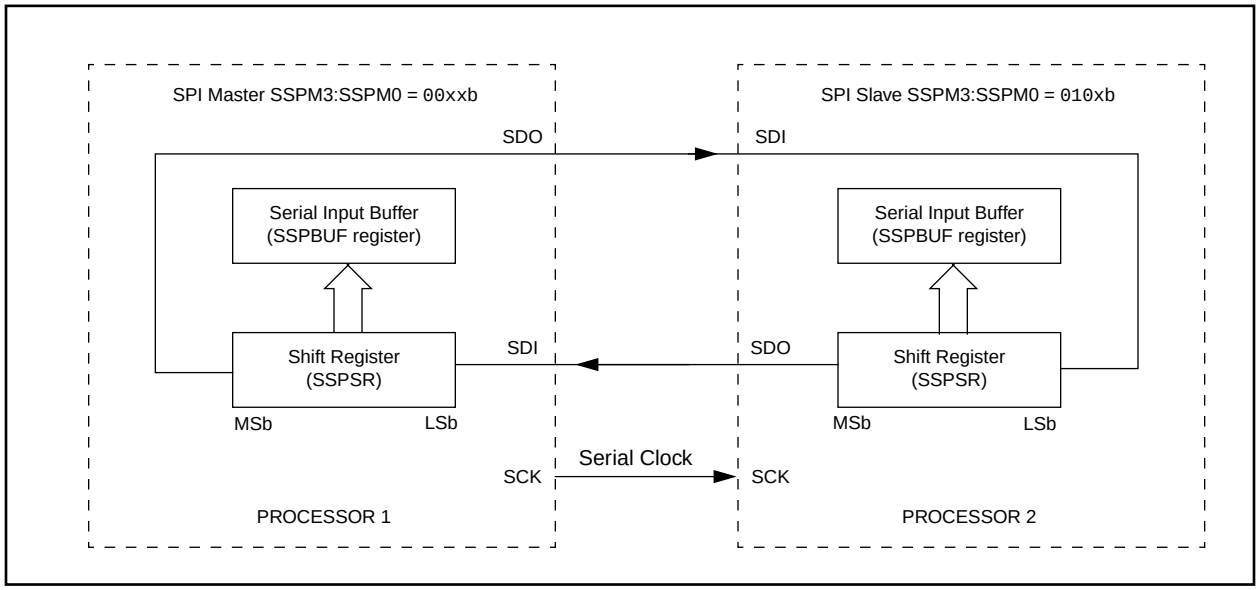
The clock polarity is selected by appropriately programming bit CKP (SSPCON<4>). This then would give waveforms for SPI communication as shown in Figure 11-5 and Figure 11-6 where the MSB is transmitted first. In master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- Fosc/4 (or $T_{CY}$)
- Fosc/16 (or 4 • $T_{CY}$)
- Fosc/64 (or 16 • $T_{CY}$)
- Timer2 output/2

This allows a maximum bit clock frequency (at 20 MHz) of 5 MHz. When in slave mode the external clock must meet the minimum high and low times.

In sleep mode, the slave can transmit and receive data and wake the device from sleep.

**FIGURE 11-4: SPI MASTER/SLAVE CONNECTION**



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**

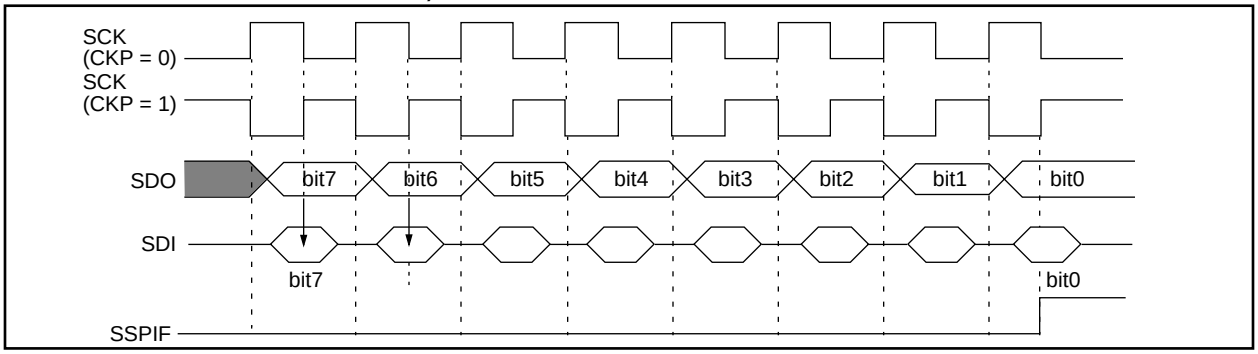



**FIGURE 11-6: SPI MODE TIMING, SLAVE MODE WITH $\overline{SS}$ CONTROL**

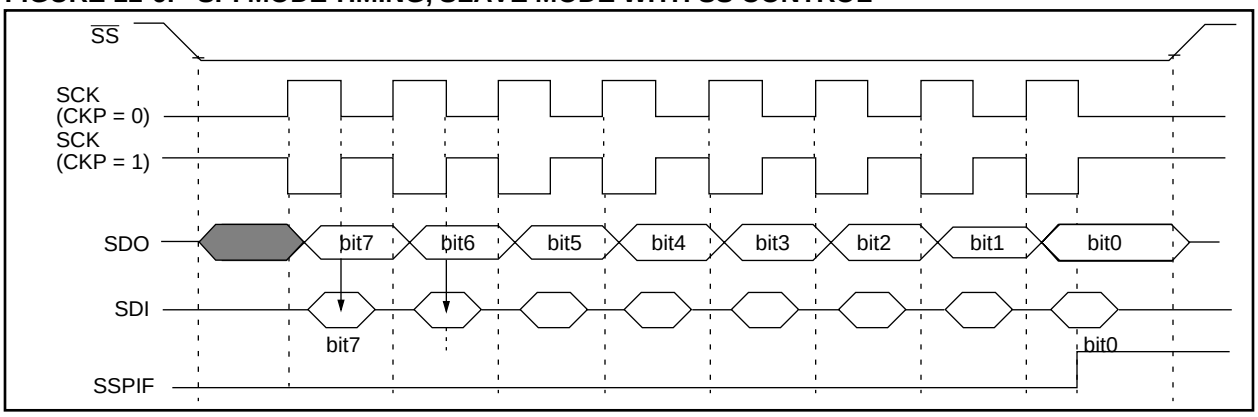



**TABLE 11-1: REGISTERS ASSOCIATED WITH SPI OPERATION**

| 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'.

## 11.3 SPI Mode for PIC16C76/77

This section contains register definitions and operational characteristics of the SPI module on the PIC16C76 and PIC16C77 only.

**FIGURE 11-7: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS 94h)(PIC16C76/77)**

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| SMP | CKE | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF |
| bit7 | | | | | | | bit0 |

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

bit 7: **SMP:** SPI data input sample phase
SPI Master Mode
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time
SPI Slave Mode
SMP must be cleared when SPI is used in slave mode

bit 6: **CKE**: SPI Clock Edge Select (Figure 11-11, Figure 11-12, and Figure 11-13)
CKP = 0
1 = Data transmitted on rising edge of SCK
0 = Data transmitted on falling edge of SCK
CKP = 1
1 = Data transmitted on falling edge of SCK
0 = Data transmitted on rising edge of SCK

bit 5: **D/$\overline{A}$**: Data/$\overline{Address}$ bit ($I^2C$ mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4: **P**: Stop bit ($I^2C$ mode only. This bit is cleared when the SSP module is disabled, or when the Start bit is detected last, SSPEN is cleared)
1 = Indicates that a stop bit has been detected last (this bit is '0' on RESET)
0 = Stop bit was not detected last

bit 3: **S**: Start bit ($I^2C$ mode only. This bit is cleared when the SSP module is disabled, or when the Stop bit is detected last, SSPEN is cleared)
1 = Indicates that a start bit has been detected last (this bit is '0' on RESET)
0 = Start bit was not detected last

bit 2: **R/$\overline{W}$**: Read/Write bit information ($I^2C$ mode only)
This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next start bit, stop bit, or $\overline{ACK}$ bit.
1 = Read
0 = Write

bit 1: **UA**: Update Address (10-bit $I^2C$ mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0: **BF**: Buffer Full Status bit

Receive (SPI and $I^2C$ modes)
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty

Transmit ($I^2C$ mode only)
1 = Transmit in progress, SSPBUF is full
0 = Transmit complete, SSPBUF is empty

**FIGURE 11-8: SSPCON: SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)(PIC16C76/77)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit7 | | | | | | | bit0 |

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

bit 7: **WCOL**: Write Collision Detect bit
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6: **SSPOV**: Receive Overflow Indicator bit

In SPI mode
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In master mode the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
0 = No overflow

In $I^2C$ mode
1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in transmit mode. SSPOV must be cleared in software in either mode.
0 = No overflow

bit 5: **SSPEN**: Synchronous Serial Port Enable bit

In SPI mode
1 = Enables serial port and configures SCK, SDO, and SDI as serial port pins
0 = Disables serial port and configures these pins as I/O port pins

In $I^2C$ mode
1 = Enables the serial port and configures the SDA and SCL pins as serial port pins
0 = Disables serial port and configures these pins as I/O port pins
In both modes, when enabled, these pins must be properly configured as input or output.

bit 4: **CKP**: Clock Polarity Select bit
In SPI mode
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level
In $I^2C$ mode
SCK release control
1 = Enable clock
0 = Holds clock low (clock stretch) (Used to ensure data setup time)

bit 3-0: **SSPM3:SSPM0**: Synchronous Serial Port Mode Select bits
0000 = SPI master mode, clock = FOSC/4
0001 = SPI master mode, clock = FOSC/16
0010 = SPI master mode, clock = FOSC/64
0011 = SPI master mode, clock = TMR2 output/2
0100 = SPI slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.
0101 = SPI slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin
0110 = $I^2C$ slave mode, 7-bit address
0111 = $I^2C$ slave mode, 10-bit address
1011 = $I^2C$ firmware controlled master mode (slave idle)
1110 = $I^2C$ slave mode, 7-bit address with start and stop bit interrupts enabled
1111 = $I^2C$ slave mode, 10-bit address with start and stop bit interrupts enabled

### 11.3.1 SPI MODE FOR PIC16C76/77

The SPI mode allows 8-bits of data to be synchronously transmitted and received simultaneously. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) RC5/SDO
- Serial Data In (SDI) RC4/SDI/SDA
- Serial Clock (SCK) RC3/SCK/SCL

Additionally a fourth pin may be used when in a slave mode of operation:

- Slave Select ($\overline{SS}$) RA5/$\overline{SS}$/AN4

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits in the SSPCON register (SSPCON<5:0>) and SSPSTAT<7:6>. These control bits allow the following to be specified:

- Master Mode (SCK is the clock output)
- Slave Mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Clock edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select Mode (Slave mode only)

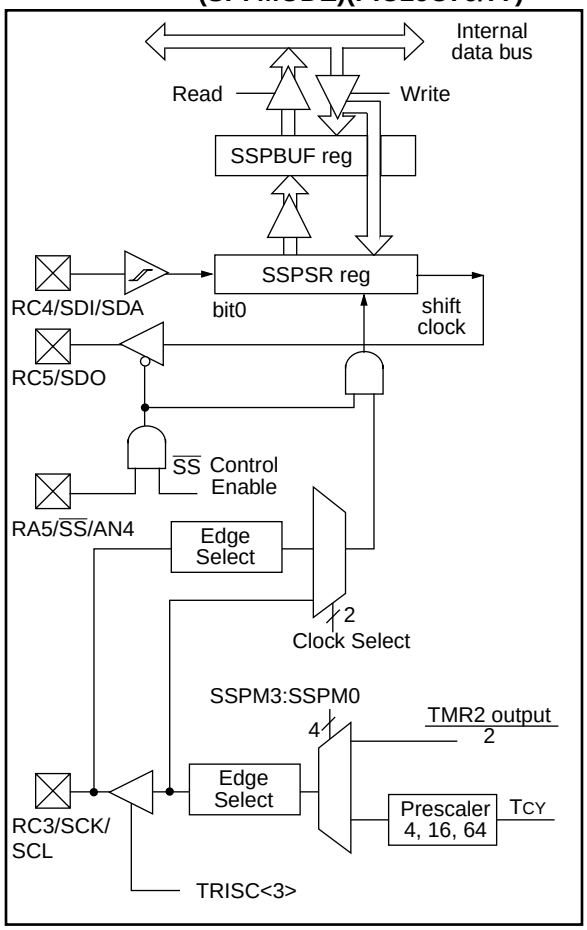
The SSP consists of a transmit/receive Shift Register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8-bits of data have been received, that byte is moved to the SSPBUF register. Then the buffer full detect bit BF (SSPSTAT<0>) and interrupt flag bit SSPIF (PIR1<3>) are set. This double buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored, and the write collision detect bit WCOL (SSPCON<7>) will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully. When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer full bit BF (SSPSTAT<0>) indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, bit BF is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally the SSP Interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 11-2 shows the loading of the SSPBUF (SSPSR) for data transmission. The shaded instruction is only required if the received data is meaningful.

**EXAMPLE 11-2: LOADING THE SSPBUF (SSPSR) REGISTER (PIC16C76/77)**

```
     BCF   STATUS, RP1    ;Specify Bank 1
     BSF   STATUS, RP0    ;
LOOP BTFSS SSPSTAT, BF    ;Has data been
                          ;received
                          ;(transmit
                          ;complete)?
     GOTO  LOOP           ;No
     BCF   STATUS, RP0    ;Specify Bank 0
     MOVF  SSPBUF, W      ;W reg = contents
                          ; of SSPBUF
     MOVWF RXDATA         ;Save in user RAM
     MOVF  TXDATA, W      ;W reg = contents
                          ; of TXDATA
     MOVWF SSPBUF         ;New data to xmit
```

The block diagram of the SSP module, when in SPI mode (Figure 11-9), shows that the SSPSR is not directly readable or writable, and can only be accessed from addressing the SSPBUF register. Additionally, the SSP status register (SSPSTAT) indicates the various status conditions.

**FIGURE 11-9: SSP BLOCK DIAGRAM (SPI MODE)(PIC16C76/77)**



To enable the serial port, SSP Enable bit, SSPEN (SSPCON<5>) must be set. To reset or reconfigure SPI mode, clear bit SSPEN, re-initialize the SSPCON register, and then set bit SSPEN. This configures the SDI, SDO, SCK, and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, they must have their data direction bits (in the TRISC register) appropriately programmed. That is:

- SDI must have TRISC<4> set
- SDO must have TRISC<5> cleared
- SCK (Master mode) must have TRISC<3> cleared
- SCK (Slave mode) must have TRISC<3> set
- $\overline{SS}$ must have TRISA<5> set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value. An example would be in master mode where you are only sending data (to a display driver), then both SDI and $\overline{SS}$ could be used as general purpose outputs by clearing their corresponding TRIS register bits.

Figure 11-10 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge, and latched on the opposite edge of the clock. Both processors should be programmed to same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application firmware. This leads to three scenarios for data transmission:

- Master sends data — Slave sends dummy data
- Master sends data — Slave sends data
- Master sends dummy data — Slave sends data

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2) is to broadcast data by the firmware protocol.

In master mode the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SCK output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "line activity monitor" mode.

In slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched the interrupt flag bit SSPIF (PIR1<3>) is set.

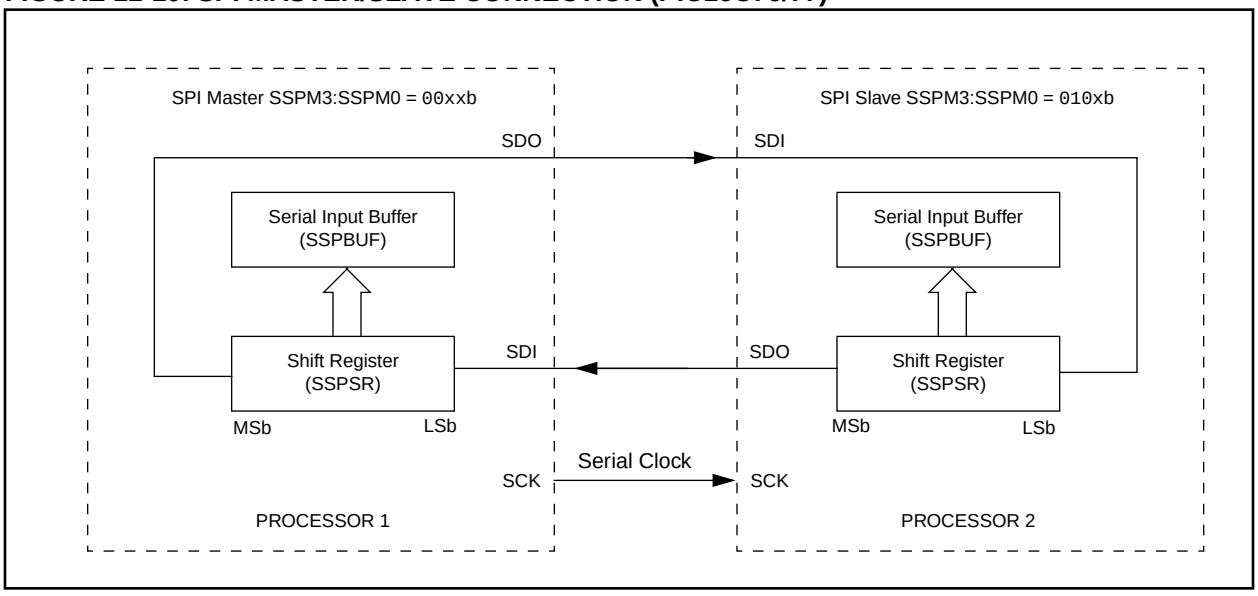
The clock polarity is selected by appropriately programming bit CKP (SSPCON<4>). This then would give waveforms for SPI communication as shown in Figure 11-11, Figure 11-12, and Figure 11-13 where the MSB is transmitted first. In master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 • TCY)
- FOSC/64 (or 16 • TCY)
- Timer2 output/2

This allows a maximum bit clock frequency (at 20 MHz) of 5 MHz. When in slave mode the external clock must meet the minimum high and low times.

In sleep mode, the slave can transmit and receive data and wake the device from sleep.

**FIGURE 11-10: SPI MASTER/SLAVE CONNECTION (PIC16C76/77)**



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 for the 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.

.

| | |
|---|---|
| **Note:** | When the SPI is in Slave Mode with $\overline{SS}$ pin control enabled, (SSPCON<3:0> = 0100) the SPI module will reset if the $\overline{SS}$ pin is set to $V_{DD}$. |
| **Note:** | If the SPI is used in Slave Mode with CKE = '1', then the $\overline{SS}$ pin control must be enabled. |

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 11-11: SPI MODE TIMING, MASTER MODE (PIC16C76/77)**

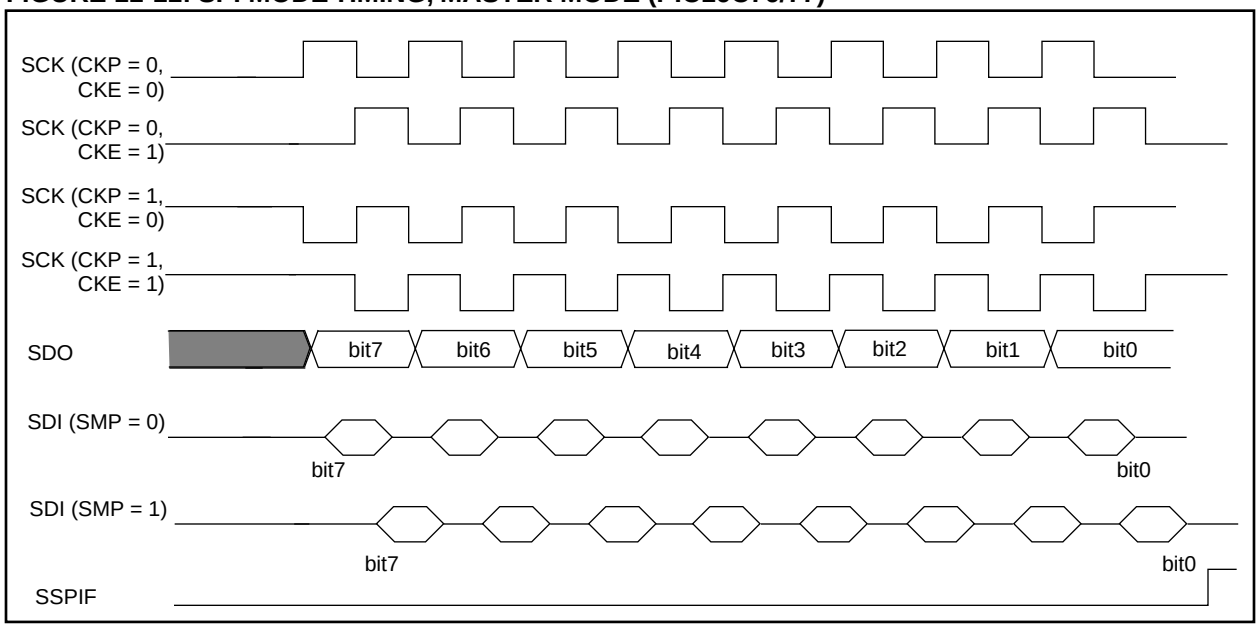



**FIGURE 11-12: SPI MODE TIMING (SLAVE MODE WITH CKE = 0) (PIC16C76/77)**

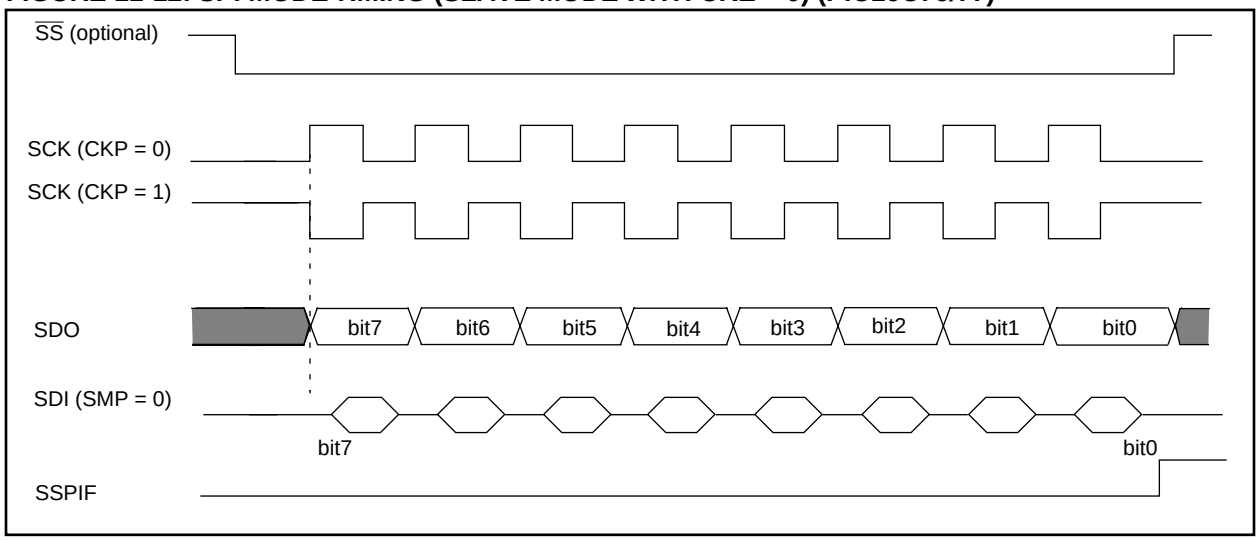


**FIGURE 11-13: SPI MODE TIMING (SLAVE MODE WITH CKE = 1) (PIC16C76/77)**

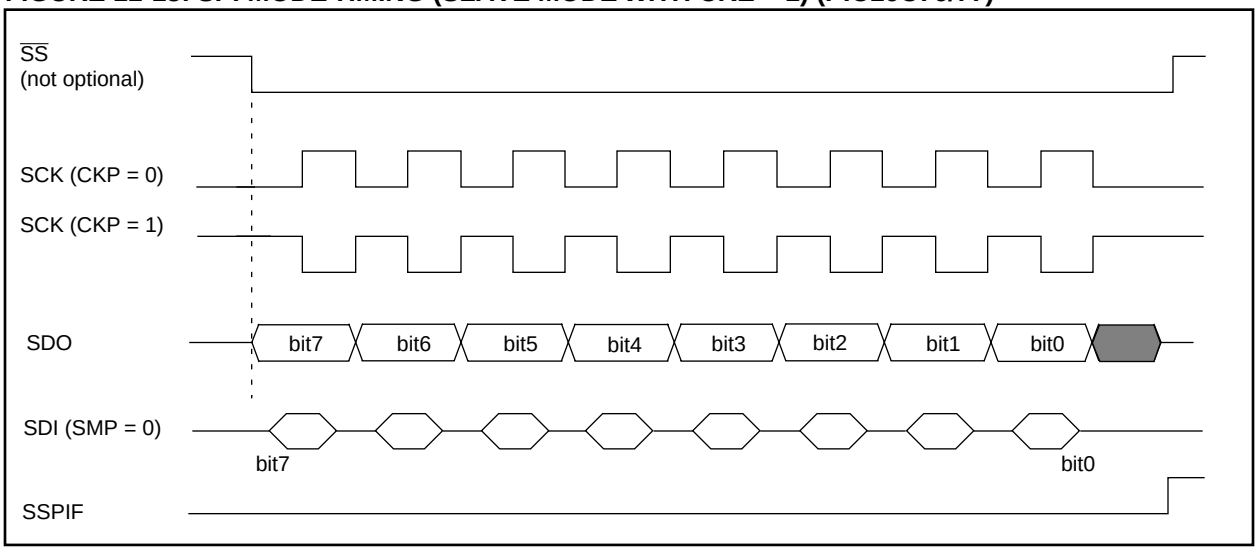



**TABLE 11-2: REGISTERS ASSOCIATED WITH SPI OPERATION (PIC16C76/77)**

| 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.

## 11.4 I²C™ Overview

This section provides an overview of the Inter-Integrated Circuit ($I^2C$) bus, with Section 11.5 discussing the operation of the SSP module in $I^2C$ mode.

The $I^2C$ bus is a two-wire serial interface developed by the Philips Corporation. The original specification, or standard mode, was for data transfers of up to 100 Kbps. The enhanced specification (fast mode) is also supported. This device will communicate with both standard and fast mode devices if attached to the same bus. The clock will determine the data rate.

The $I^2C$ interface employs a comprehensive protocol to ensure reliable transmission and reception of data. When transmitting data, one device is the "master" which initiates transfer on the bus and generates the clock signals to permit that transfer, while the other device(s) acts as the "slave." All portions of the slave protocol are implemented in the SSP module's hardware, except general call support, while portions of the master protocol need to be addressed in the PIC16CXX software. Table 11-3 defines some of the $I^2C$ bus terminology. For additional information on the $I^2C$ interface specification, refer to the Philips document "*The $I^2C$ bus and how to use it.*"#939839340011, which can be obtained from the Philips Corporation.

In the $I^2C$ interface protocol each device has an address. When a master wishes to initiate a data transfer, it first transmits the address of the device that it wishes to "talk" to. All devices "listen" to see if this is their address. Within this address, a bit specifies if the master wishes to read-from/write-to the slave device. The master and slave are always in opposite modes (transmitter/receiver) of operation during a data transfer. That is they can be thought of as operating in either of these two relations:

- Master-transmitter and Slave-receiver
- Slave-transmitter and Master-receiver

In both cases the master generates the clock signal.

The output stages of the clock (SCL) and data (SDA) lines must have an open-drain or open-collector in order to perform the wired-AND function of the bus. External pull-up resistors are used to ensure a high level when no device is pulling the line down. The number of devices that may be attached to the $I^2C$ bus is limited only by the maximum bus loading specification of 400 pF.

### 11.4.1 INITIATING AND TERMINATING DATA TRANSFER

During times of no data transfer (idle time), both the clock line (SCL) and the data line (SDA) are pulled high through the external pull-up resistors. The START and STOP conditions determine the start and stop of data transmission. The START condition is defined as a high to low transition of the SDA when the SCL is high. The STOP condition is defined as a low to high transition of the SDA when the SCL is high. Figure 11-14 shows the START and STOP conditions. The master generates these conditions for starting and terminating data transfer. Due to the definition of the START and STOP conditions, when data is being transmitted, the SDA line can only change state when the SCL line is low.

**FIGURE 11-14: START AND STOP CONDITIONS**

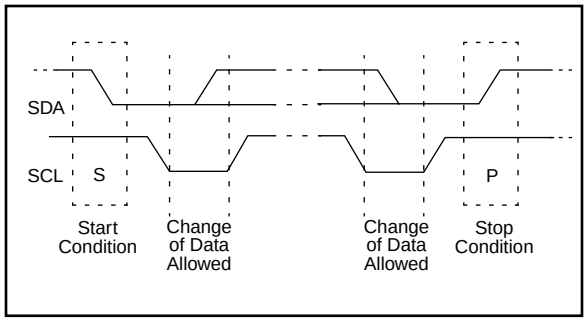



**TABLE 11-3: I²C BUS TERMINOLOGY**

| Term | Description |
|---|---|
| Transmitter | The device that sends the data to the bus. |
| Receiver | The device that receives the data from the bus. |
| Master | The device which initiates the transfer, generates the clock and terminates the transfer. |
| Slave | The device addressed by a master. |
| Multi-master | More than one master device in a system. These masters can attempt to control the bus at the same time without corrupting the message. |
| Arbitration | Procedure that ensures that only one of the master devices will control the bus. This ensure that the transfer data does not get corrupted. |
| Synchronization | Procedure where the clock signals of two or more devices are synchronized. |

### 11.4.2 ADDRESSING $I^2C$ DEVICES

There are two address formats. The simplest is the 7-bit address format with a R/$\overline{W}$ bit (Figure 11-15). The more complex is the 10-bit address with a R/$\overline{W}$ bit (Figure 11-16). For 10-bit address format, two bytes must be transmitted with the first five bits specifying this to be a 10-bit address.

**FIGURE 11-15: 7-BIT ADDRESS FORMAT**

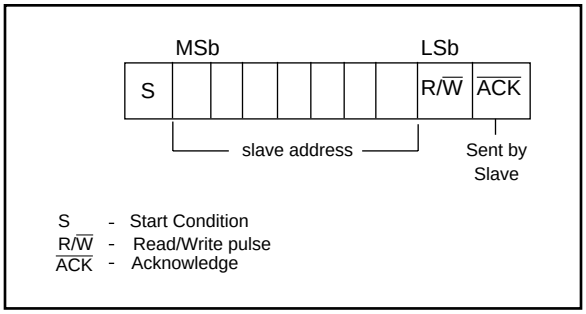
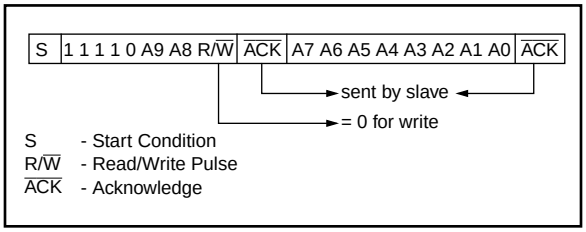


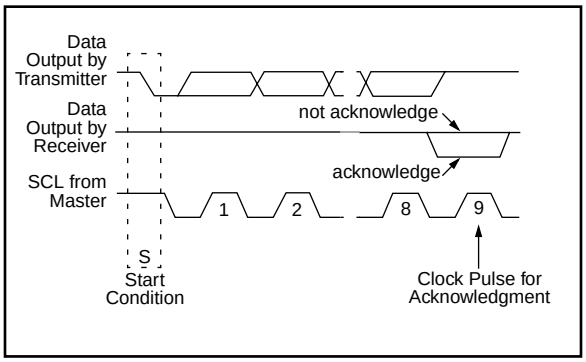

**FIGURE 11-16: $I^2C$ 10-BIT ADDRESS FORMAT**




### 11.4.3 TRANSFER ACKNOWLEDGE

All data must be transmitted per byte, with no limit to the number of bytes transmitted per data transfer. After each byte, the slave-receiver generates an acknowledge bit ($\overline{ACK}$) (Figure 11-17). When a slave-receiver doesn't acknowledge the slave address or received data, the master must abort the transfer. The slave must leave SDA high so that the master can generate the STOP condition (Figure 11-14).

**FIGURE 11-17: SLAVE-RECEIVER ACKNOWLEDGE**




If the master is receiving the data (master-receiver), it generates an acknowledge signal for each received byte of data, except for the last byte. To signal the end of data to the slave-transmitter, the master does not generate an acknowledge (not acknowledge). The slave then releases the SDA line so the master can generate the STOP condition. The master can also generate the STOP condition during the acknowledge pulse for valid termination of data transfer.

If the slave needs to delay the transmission of the next byte, holding the SCL line low will force the master into a wait state. Data transfer continues when the slave releases the SCL line. This allows the slave to move the received data or fetch the data it needs to transfer before allowing the clock to start. This wait state technique can also be implemented at the bit level, Figure 11-18. The slave will inherently stretch the clock, when it is a transmitter, but will not when it is a receiver. The slave will have to clear the SSPCON<4> bit to enable clock stretching when it is a receiver.

**FIGURE 11-18: DATA TRANSFER WAIT STATE**

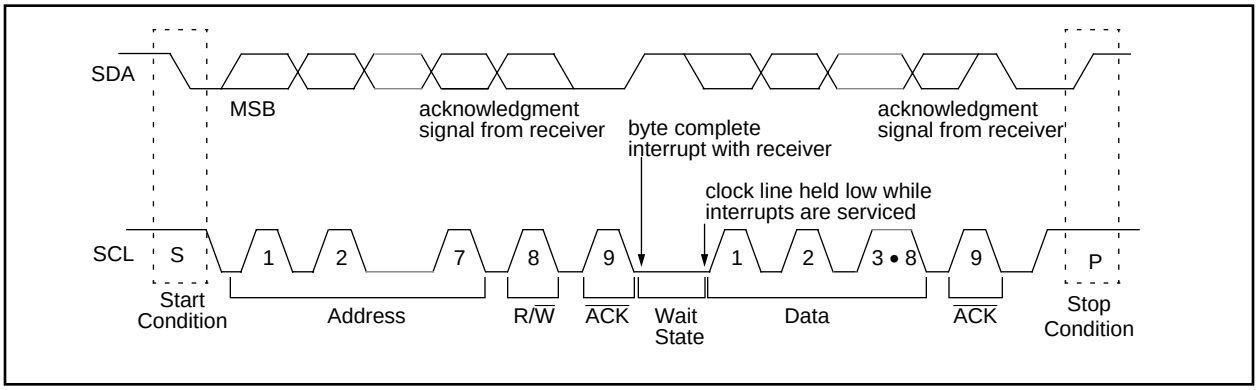


Figure 11-19 and Figure 11-20 show Master-transmitter and Master-receiver data transfer sequences.

When a master does not wish to relinquish the bus (by generating a STOP condition), a repeated START condition (Sr) must be generated. This condition is identical to the start condition (SDA goes high-to-low while SCL is high), but occurs after a data transfer acknowledge pulse (not the bus-free state). This allows a master to send "commands" to the slave and then receive the requested information or to address a different slave device. This sequence is shown in Figure 11-21.

**FIGURE 11-19: MASTER-TRANSMITTER SEQUENCE**

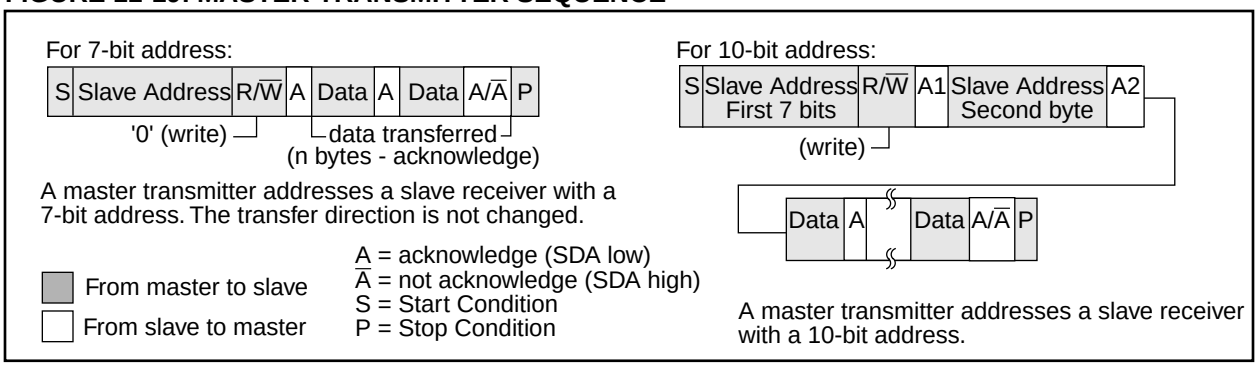



**FIGURE 11-20: MASTER-RECEIVER SEQUENCE**

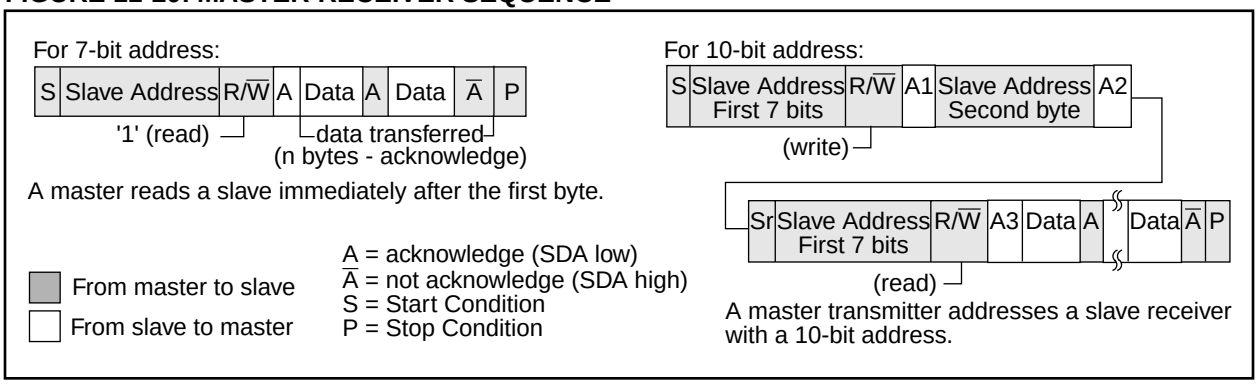
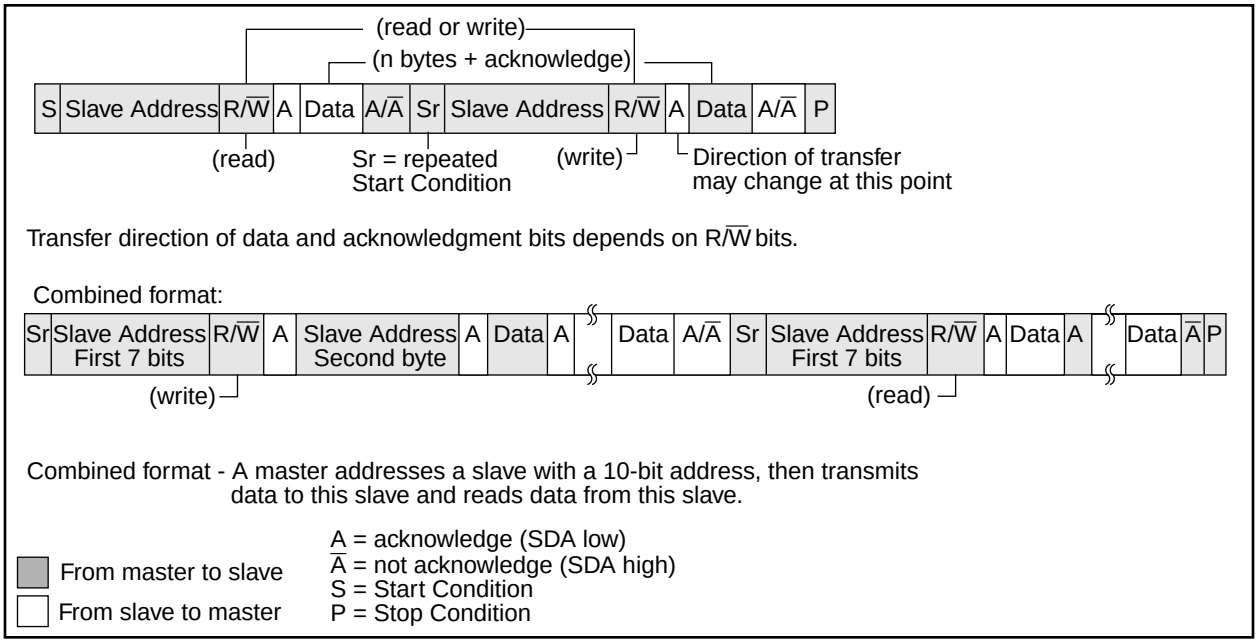



**FIGURE 11-21: COMBINED FORMAT**



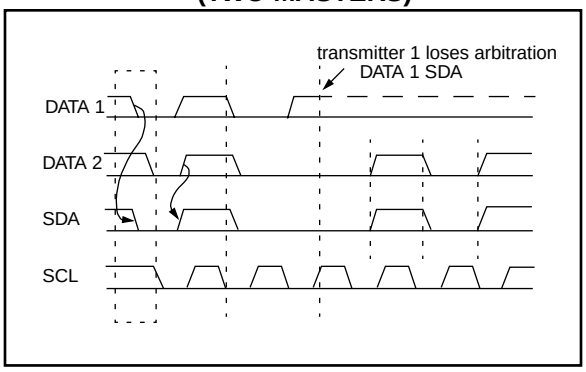
11.4.4 MULTI-MASTER

The $I^2C$ protocol allows a system to have more than one master. This is called multi-master. When two or more masters try to transfer data at the same time, arbitration and synchronization occur.

11.4.4.1 ARBITRATION

Arbitration takes place on the SDA line, while the SCL line is high. The master which transmits a high when the other master transmits a low loses arbitration (Figure 11-22), and turns off its data output stage. A master which lost arbitration can generate clock pulses until the end of the data byte where it lost arbitration. When the master devices are addressing the same device, arbitration continues into the data.

**FIGURE 11-22: MULTI-MASTER ARBITRATION (TWO MASTERS)**




Masters that also incorporate the slave function, and have lost arbitration must immediately switch over to slave-receiver mode. This is because the winning master-transmitter may be addressing it.

Arbitration is not allowed between:

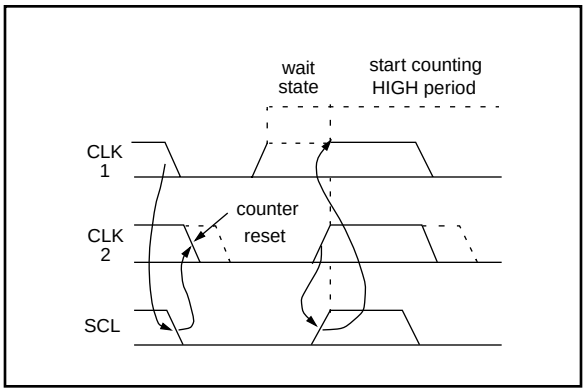
- A repeated START condition
- A STOP condition and a data bit
- A repeated START condition and a STOP condition

Care needs to be taken to ensure that these conditions do not occur.

11.2.4.2 Clock Synchronization

Clock synchronization occurs after the devices have started arbitration. This is performed using a wired-AND connection to the SCL line. A high to low transition on the SCL line causes the concerned devices to start counting off their low period. Once a device clock has gone low, it will hold the SCL line low until its SCL high state is reached. The low to high transition of this clock may not change the state of the SCL line, if another device clock is still within its low period. The SCL line is held low by the device with the longest low period. Devices with shorter low periods enter a high wait-state, until the SCL line comes high. When the SCL line comes high, all devices start counting off their high periods. The first device to complete its high period will pull the SCL line low. The SCL line high time is determined by the device with the shortest high period, Figure 11-23.

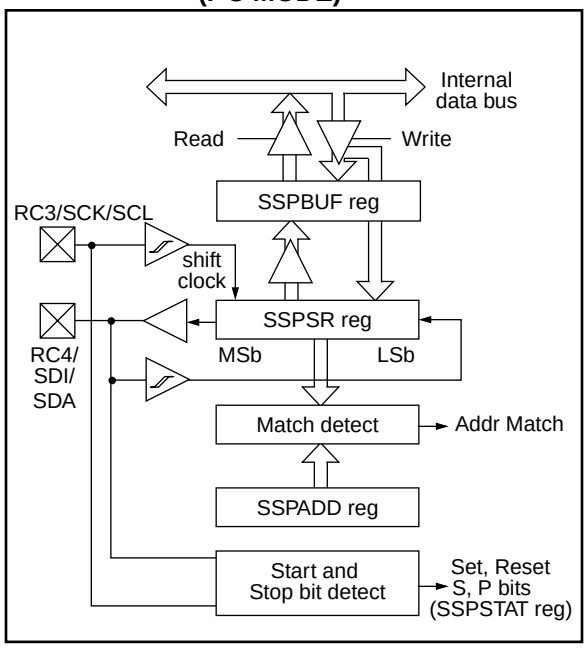
**FIGURE 11-23: CLOCK SYNCHRONIZATION**



## 11.5 SSP I²C Operation

The SSP module in I²C mode fully implements all slave functions, except general call support, and provides interrupts on start and stop bits in hardware to facilitate firmware implementations of the master functions. The SSP module implements the standard mode specifications as well as 7-bit and 10-bit addressing. Two pins are used for data transfer. These are the RC3/SCK/SCL pin, which is the clock (SCL), and the RC4/SDI/SDA pin, which is the data (SDA). The user must configure these pins as inputs or outputs through the TRISC<4:3> bits. The SSP module functions are enabled by setting SSP Enable bit SSPEN (SSPCON<5>).

**FIGURE 11-24: SSP BLOCK DIAGRAM (I²C MODE)**




The SSP module has five registers for I²C operation. These are the:

- SSP Control Register (SSPCON)
- SSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer (SSPBUF)
- SSP Shift Register (SSPSR) - Not directly accessible
- SSP Address Register (SSPADD)

The SSPCON register allows control of the I²C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I²C modes to be selected:

- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- I²C Slave mode (7-bit address), with start and stop bit interrupts enabled
- I²C Slave mode (10-bit address), with start and stop bit interrupts enabled
- I²C Firmware controlled Master Mode, slave is idle

Selection of any I²C mode, with the SSPEN bit set, forces the SCL and SDA pins to be open drain, provided these pins are programmed to inputs by setting the appropriate TRISC bits.

The SSPSTAT register gives the status of the data transfer. This information includes detection of a START or STOP bit, specifies if the received byte was data or address if the next byte is the completion of 10-bit address, and if this will be a read or write data transfer. The SSPSTAT register is read only.

The SSPBUF is the register to which transfer data is written to or read from. The SSPSR register shifts the data in or out of the device. In receive operations, the SSPBUF and SSPSR create a doubled buffered receiver. This allows reception of the next byte to begin before reading the last byte of received data. When the complete byte is received, it is transferred to the SSPBUF register and flag bit SSPIF is set. If another complete byte is received before the SSPBUF register is read, a receiver overflow has occurred and bit SSPOV (SSPCON<6>) is set and the byte in the SSPSR is lost.

The SSPADD register holds the slave address. In 10-bit mode, the user first needs to write the high byte of the address (`1111 0 A9 A8 0`). Following the high byte address match, the low byte of the address needs to be loaded (A7:A0).

### 11.5.1 SLAVE MODE

In slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The SSP module will override the input state with the output data when required (slave-transmitter).

When an address is matched or the data transfer after an address match is received, the hardware automatically will generate the acknowledge ($\overline{ACK}$) pulse, and then load the SSPBUF register with the received value currently in the SSPSR register.

There are certain conditions that will cause the SSP module not to give this $\overline{ACK}$ pulse. These are if either (or both):

a) The buffer full bit BF (SSPSTAT<0>) was set before the transfer was received.

b) The overflow bit SSPOV (SSPCON<6>) was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF (PIR1<3>) is set. Table 11-4 shows what happens when a data transfer byte is received, given the status of bits BF and SSPOV. The shaded cells show the condition where user software did not properly clear the overflow condition. Flag bit BF is cleared by reading the SSPBUF register while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the $I^2C$ specification as well as the requirement of the SSP module is shown in timing parameter #100 and parameter #101.

#### 11.5.1.1 ADDRESSING

Once the SSP module has been enabled, it waits for a START condition to occur. Following the START condition, the 8-bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the BF and SSPOV bits are clear, the following events occur:

a) The SSPSR register value is loaded into the SSPBUF register.

b) The buffer full bit, BF is set.

c) An $\overline{ACK}$ pulse is generated.

d) SSP interrupt flag bit, SSPIF (PIR1<3>) is set (interrupt is generated if enabled) - on the falling edge of the ninth SCL pulse.

In 10-bit address mode, two address bytes need to be received by the slave (Figure 11-16). The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit R/$\overline{W}$ (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address the first byte would equal '1111 0 A9 A8 0', where A9 and A8 are the two MSbs of the address. The sequence of events for 10-bit address is as follows, with steps 7- 9 for slave-transmitter:

1. Receive first (high) byte of Address (bits SSPIF, BF, and bit UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with second (low) byte of Address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
4. Receive second (low) byte of Address (bits SSPIF, BF, and UA are set).
5. Update the SSPADD register with the first (high) byte of Address, if match releases SCL line, this will clear bit UA.
6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
7. Receive repeated START condition.
8. Receive first (high) byte of Address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

**TABLE 11-4: DATA TRANSFER RECEIVED BYTE ACTIONS**

| Status Bits as Data Transfer is Received | | | | |
|---|---|---|---|---|
| **BF** | **SSPOV** | **SSPSR → SSPBUF** | **Generate $\overline{ACK}$ Pulse** | **Set bit SSPIF (SSP Interrupt occurs if enabled)** |
| 0 | 0 | Yes | Yes | Yes |
| 1 | 0 | No | No | Yes |
| 1 | 1 | No | No | Yes |
| 0 | 1 | No | No | Yes |

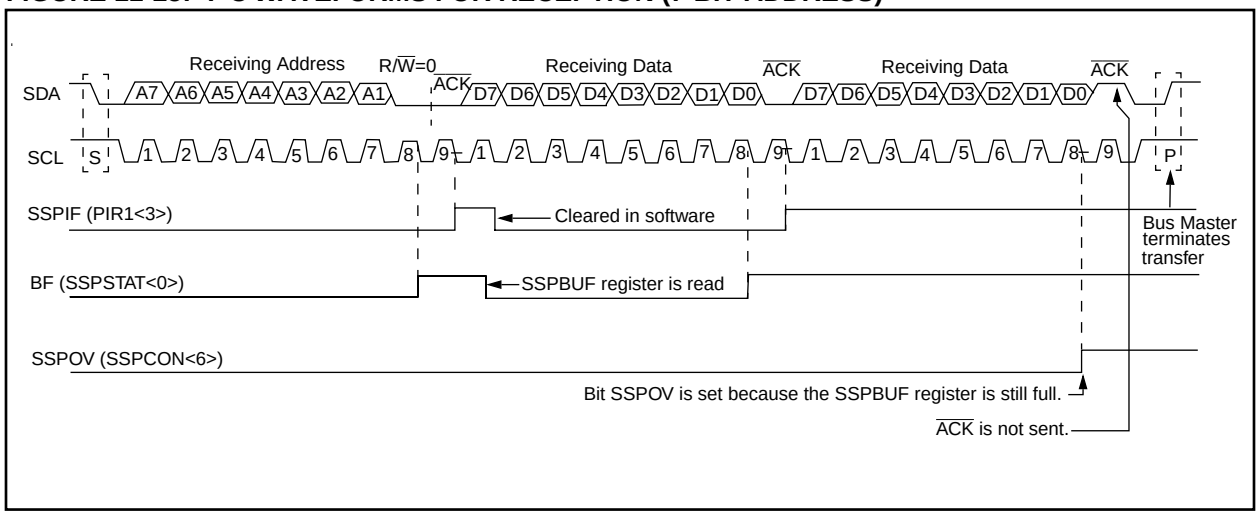
11.5.1.2 RECEPTION

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)**



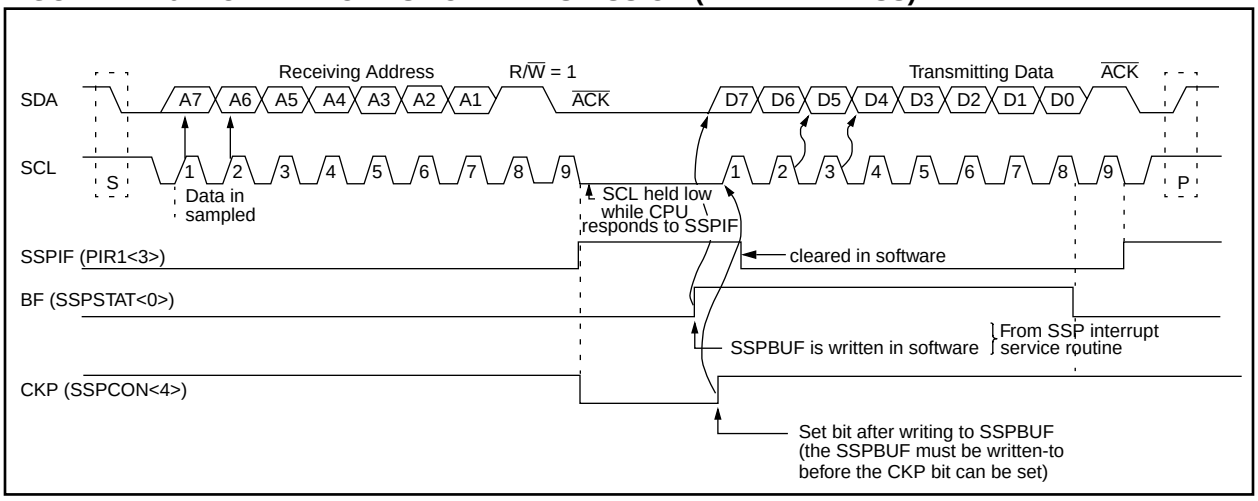
11.5.1.3 TRANSMISSION

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit, and pin RC3/SCK/SCL is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RC3/SCK/SCL should be enabled by setting bit CKP (SSPCON<4>). The master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 11-26).

An SSP interrupt is generated for each data transfer byte. Flag bit SSPIF must be cleared in software, and the SSPSTAT register is used to determine the status of the byte. Flag bit SSPIF is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not $\overline{ACK}$), then the data transfer is complete. When the $\overline{ACK}$ is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave then monitors for another occurrence of the START bit. If the SDA line was low ($\overline{ACK}$), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RC3/SCK/SCL should be enabled by setting bit CKP.

**FIGURE 11-26: I$^2$C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**



### 11.5.2 MASTER MODE

Master mode of operation is supported in firmware using interrupt generation on the detection of the START and STOP conditions. The STOP (P) and START (S) bits are cleared from a reset or when the SSP module is disabled. The STOP (P) and START (S) bits will toggle based on the START and STOP conditions. Control of the $I^2C$ bus may be taken when the P bit is set, or the bus is idle and both the S and P bits are clear.

In master mode the SCL and SDA lines are manipulated by clearing the corresponding TRISC<4:3> bit(s). The output level is always low, irrespective of the value(s) in PORTC<4:3>. So when transmitting data, a '1' data bit must have the TRISC<4> bit set (input) and a '0' data bit must have the TRISC<4> bit cleared (output). The same scenario is true for the SCL line with the TRISC<3> bit.

The following events will cause SSP Interrupt Flag bit, SSPIF, to be set (SSP Interrupt if enabled):

- START condition
- STOP condition
- Data transfer byte transmitted/received

Master mode of operation can be done with either the slave mode idle (SSPM3:SSPM0 = 1011) or with the slave active. When both master and slave modes are enabled, the software needs to differentiate the source(s) of the interrupt.

### 11.5.3 MULTI-MASTER MODE

In multi-master mode, the interrupt generation on the detection of the START and STOP conditions allows the determination of when the bus is free. The STOP (P) and START (S) bits are cleared from a reset or when the SSP module is disabled. The STOP (P) and START (S) bits will toggle based on the START and STOP conditions. Control of the $I^2C$ bus may be taken when bit P (SSPSTAT<4>) is set, or the bus is idle and both the S and P bits clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the STOP condition occurs.

In multi-master operation, the SDA line must be monitored to see if the signal level is the expected output level. This check only needs to be done when a high level is output. If a high level is expected and a low level is present, the device needs to release the SDA and SCL lines (set TRISC<4:3>). There are two stages where this arbitration can be lost, these are:

- Address Transfer
- Data Transfer

When the slave logic is enabled, the slave continues to receive. If arbitration was lost during the address transfer stage, communication to the device may be in progress. If addressed an $\overline{ACK}$ pulse will be generated. If arbitration was lost during the data transfer stage, the device will need to re-transfer the data at a later time.

**TABLE 11-5: REGISTERS ASSOCIATED WITH $I^2C$ 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, 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 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 93h | SSPADD | Synchronous Serial Port ($I^2C$ mode) Address Register | | | | | | | | 0000 0000 | 0000 0000 |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 94h | SSPSTAT | SMP(2) | CKE(2) | D/$\overline{A}$ | P | S | R/$\overline{W}$ | UA | BF | 0000 0000 | 0000 0000 |
| 87h | TRISC | PORTC Data Direction register | | | | | | | | 1111 1111 | 1111 1111 |

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.
Shaded cells are not used by SSP module in SPI mode.

Note 1: PSPIF and PSPIE are reserved on the PIC16C73/73A/76, always maintain these bits clear.
2: The SMP and CKE bits are implemented on the PIC16C76/77 only. All other PIC16C7X devices have these two bits unimplemented, read as '0'.

**FIGURE 11-27: OPERATION OF THE I²C MODULE IN IDLE_MODE, RCV_MODE OR XMIT_MODE**

```
IDLE_MODE (7-bit):
if (Addr_match)                 {       Set interrupt;
                                        if (R/W = 1)    {       Send ACK = 0;
                                                                set XMIT_MODE;
                                                        }
                                        else if (R/W = 0) set RCV_MODE;
                                }
RCV_MODE:
if ((SSPBUF=Full) OR (SSPOV = 1))
        {       Set SSPOV;
                Do not acknowledge;
        }
else    {        transfer SSPSR → SSPBUF;
                send ACK = 0;
        }
Receive 8-bits in SSPSR;
Set interrupt;
XMIT_MODE:
While ((SSPBUF = Empty) AND (CKP=0)) Hold SCL Low;
Send byte;
Set interrupt;
if ( ACK Received = 1)          {       End of transmission;
                                        Go back to IDLE_MODE;
                                }
else if ( ACK Received = 0)   Go back to XMIT_MODE;
IDLE_MODE (10-Bit):
If (High_byte_addr_match AND (R/W = 0))
        {        PRIOR_ADDR_MATCH = FALSE;
                Set interrupt;
                if ((SSPBUF = Full) OR ((SSPOV = 1))
                        {        Set SSPOV;
                                Do not acknowledge;
                        }
                else    {       Set UA = 1;
                                Send ACK = 0;
                                While (SSPADD not updated) Hold SCL low;
                                Clear UA = 0;
                                Receive Low_addr_byte;
                                Set interrupt;
                                Set UA = 1;
                                If (Low_byte_addr_match)
                                        {       PRIOR_ADDR_MATCH = TRUE;
                                                Send ACK = 0;
                                                while (SSPADD not updated) Hold SCL low;
                                                Clear UA = 0;
                                                Set RCV_MODE;
                                        }
                        }
        }
else if (High_byte_addr_match AND (R/W = 1)
        {       if (PRIOR_ADDR_MATCH)
                        {       send ACK = 0;
                                set XMIT_MODE;
                        }
        else PRIOR_ADDR_MATCH = FALSE;
        }
```

## 12.0 UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

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

The Universal Synchronous Asynchronous Receiver Transmitter (USART) module is one of the two serial I/O modules. (USART is also known as a Serial Communications Interface or SCI). The USART can be configured as a full duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, Serial EEPROMs etc.

The USART can be configured in the following modes:

- Asynchronous (full duplex)
- Synchronous - Master (half duplex)
- Synchronous - Slave (half duplex)

Bit SPEN (RCSTA<7>), and bits TRISC<7:6>, have to be set in order to configure pins RC6/TX/CK and RC7/RX/DT as the Universal Synchronous Asynchronous Receiver Transmitter.

**FIGURE 12-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R-1 | R/W-0 |
|---|---|---|---|---|---|---|---|
| CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D |
| bit7 | | | | | | | bit0 |

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

bit 7: **CSRC**: Clock Source Select bit

Asynchronous mode
Don't care

Synchronous mode
1 = Master mode (Clock generated internally from BRG)
0 = Slave mode (Clock from external source)

bit 6: **TX9**: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5: **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled
Note: SREN/CREN overrides TXEN in SYNC mode.

bit 4: **SYNC**: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

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

bit 2: **BRGH**: High Baud Rate Select bit

Asynchronous mode
1 = High speed

> **Note:** For the PIC16C73/73A/74/74A, the asynchronous high speed mode (BRGH = 1) may experience a high rate of receive errors. It is recommended that BRGH = 0. If you desire a higher baud rate than BRGH = 0 can support, refer to the device errata for additional information, or use the PIC16C76/77.

0 = Low speed

Synchronous mode
Unused in this mode

bit 1: **TRMT**: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0: **TX9D**: 9th bit of transmit data. Can be parity bit.

**FIGURE 12-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R-0 | R-0 | R-x |
|---|---|---|---|---|---|---|---|
| SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D |
| bit7 | | | | | | | bit0 |

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

bit 7: **SPEN**: Serial Port Enable bit
1 = Serial port enabled (Configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)
0 = Serial port disabled

bit 6: **RX9**: 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5: **SREN**: Single Receive Enable bit

Asynchronous mode
Don't care

Synchronous mode - master
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.

Synchronous mode - slave
Unused in this mode

bit 4: **CREN**: Continuous Receive Enable bit

Asynchronous mode
1 = Enables continuous receive
0 = Disables continuous receive

Synchronous mode
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

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

bit 2: **FERR**: Framing Error bit
1 = Framing error (Can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1: **OERR**: Overrun Error bit
1 = Overrun error (Can be cleared by clearing bit CREN)
0 = No overrun error

bit 0: **RX9D**: 9th bit of received data (Can be parity bit)

## 12.1 USART Baud Rate Generator (BRG)

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

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free running 8-bit timer. In asynchronous mode bit BRGH (TXSTA<2>) also controls the baud rate. In synchronous mode bit BRGH is ignored. Table 12-1 shows the formula for computation of the baud rate for different USART modes which only apply in master mode (internal clock).

Given the desired baud rate and Fosc, the nearest integer value for the SPBRG register can be calculated using the formula in Table 12-1. From this, the error in baud rate can be determined.

Example 12-1 shows the calculation of the baud rate error for the following conditions:

FOSC = 16 MHz
Desired Baud Rate = 9600
BRGH = 0
SYNC = 0

### EXAMPLE 12-1: CALCULATING BAUD RATE ERROR

Desired Baud rate = Fosc / (64 (X + 1))

9600 = 16000000 /(64 (X + 1))

X = ⌊25.042⌋ = 25

Calculated Baud Rate=16000000 / (64 (25 + 1))

= 9615

$$\text{Error} = \frac{\text{(Calculated Baud Rate - Desired Baud Rate)}}{\text{Desired Baud Rate}}$$

= (9615 - 9600) / 9600

= 0.16%

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the FOSC/(16(X + 1)) equation can reduce the baud rate error in some cases.

**Note:** For the PIC16C73/73A/74/74A, the asynchronous high speed mode (BRGH = 1) may experience a high rate of receive errors. It is recommended that BRGH = 0. If you desire a higher baud rate than BRGH = 0 can support, refer to the device errata for additional information, or use the PIC16C76/77.

Writing a new value to the SPBRG register, causes the BRG timer to be reset (or cleared), this ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

### TABLE 12-1: BAUD RATE FORMULA

| SYNC | BRGH = 0 (Low Speed) | BRGH = 1 (High Speed) |
|---|---|---|
| 0 | (Asynchronous) Baud Rate = FOSC/(64(X+1)) | Baud Rate= FOSC/(16(X+1)) |
| 1 | (Synchronous) Baud Rate = FOSC/(4(X+1)) | NA |

X = value in SPBRG (0 to 255)

### TABLE 12-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used by the BRG.

### TABLE 12-3: BAUD RATES FOR SYNCHRONOUS MODE

| BAUD RATE (K) | FOSC = 20 MHz | | | 16 MHz | | | 10 MHz | | | 7.15909 MHz | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 0.3 | NA | - | - | NA | - | - | NA | - | - | NA | - | - |
| 1.2 | NA | - | - | NA | - | - | NA | - | - | NA | - | - |
| 2.4 | NA | - | - | NA | - | - | NA | - | - | NA | - | - |
| 9.6 | NA | - | - | NA | - | - | 9.766 | +1.73 | 255 | 9.622 | +0.23 | 185 |
| 19.2 | 19.53 | +1.73 | 255 | 19.23 | +0.16 | 207 | 19.23 | +0.16 | 129 | 19.24 | +0.23 | 92 |
| 76.8 | 76.92 | +0.16 | 64 | 76.92 | +0.16 | 51 | 75.76 | -1.36 | 32 | 77.82 | +1.32 | 22 |
| 96 | 96.15 | +0.16 | 51 | 95.24 | -0.79 | 41 | 96.15 | +0.16 | 25 | 94.20 | -1.88 | 18 |
| 300 | 294.1 | -1.96 | 16 | 307.69 | +2.56 | 12 | 312.5 | +4.17 | 7 | 298.3 | -0.57 | 5 |
| 500 | 500 | 0 | 9 | 500 | 0 | 7 | 500 | 0 | 4 | NA | - | - |
| HIGH | 5000 | - | 0 | 4000 | - | 0 | 2500 | - | 0 | 1789.8 | - | 0 |
| LOW | 19.53 | - | 255 | 15.625 | - | 255 | 9.766 | - | 255 | 6.991 | - | 255 |

| BAUD RATE (K) | FOSC = 5.0688 MHz | | | 4 MHz | | | 3.579545 MHz | | | 1 MHz | | | 32.768 kHz | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 0.3 | NA | - | - | NA | - | - | NA | - | - | NA | - | - | 0.303 | +1.14 | 26 |
| 1.2 | NA | - | - | NA | - | - | NA | - | - | 1.202 | +0.16 | 207 | 1.170 | -2.48 | 6 |
| 2.4 | NA | - | - | NA | - | - | NA | - | - | 2.404 | +0.16 | 103 | NA | - | - |
| 9.6 | 9.6 | 0 | 131 | 9.615 | +0.16 | 103 | 9.622 | +0.23 | 92 | 9.615 | +0.16 | 25 | NA | - | - |
| 19.2 | 19.2 | 0 | 65 | 19.231 | +0.16 | 51 | 19.04 | -0.83 | 46 | 19.24 | +0.16 | 12 | NA | - | - |
| 76.8 | 79.2 | +3.13 | 15 | 76.923 | +0.16 | 12 | 74.57 | -2.90 | 11 | 83.34 | +8.51 | 2 | NA | - | - |
| 96 | 97.48 | +1.54 | 12 | 1000 | +4.17 | 9 | 99.43 | +3.57 | 8 | NA | - | - | NA | - | - |
| 300 | 316.8 | +5.60 | 3 | NA | - | - | 298.3 | -0.57 | 2 | NA | - | - | NA | - | - |
| 500 | NA | - | - | NA | - | - | NA | - | - | NA | - | - | NA | - | - |
| HIGH | 1267 | - | 0 | 100 | - | 0 | 894.9 | - | 0 | 250 | - | 0 | 8.192 | - | 0 |
| LOW | 4.950 | - | 255 | 3.906 | - | 255 | 3.496 | - | 255 | 0.9766 | - | 255 | 0.032 | - | 255 |

### TABLE 12-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

| BAUD RATE (K) | FOSC = 20 MHz | | | 16 MHz | | | 10 MHz | | | 7.15909 MHz | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 0.3 | NA | - | - | NA | - | - | NA | - | - | NA | - | - |
| 1.2 | 1.221 | +1.73 | 255 | 1.202 | +0.16 | 207 | 1.202 | +0.16 | 129 | 1.203 | +0.23 | 92 |
| 2.4 | 2.404 | +0.16 | 129 | 2.404 | +0.16 | 103 | 2.404 | +0.16 | 64 | 2.380 | -0.83 | 46 |
| 9.6 | 9.469 | -1.36 | 32 | 9.615 | +0.16 | 25 | 9.766 | +1.73 | 15 | 9.322 | -2.90 | 11 |
| 19.2 | 19.53 | +1.73 | 15 | 19.23 | +0.16 | 12 | 19.53 | +1.73 | 7 | 18.64 | -2.90 | 5 |
| 76.8 | 78.13 | +1.73 | 3 | 83.33 | +8.51 | 2 | 78.13 | +1.73 | 1 | NA | - | - |
| 96 | 104.2 | +8.51 | 2 | NA | - | - | NA | - | - | NA | - | - |
| 300 | 312.5 | +4.17 | 0 | NA | - | - | NA | - | - | NA | - | - |
| 500 | NA | - | - | NA | - | - | NA | - | - | NA | - | - |
| HIGH | 312.5 | - | 0 | 250 | - | 0 | 156.3 | - | 0 | 111.9 | - | 0 |
| LOW | 1.221 | - | 255 | 0.977 | - | 255 | 0.6104 | - | 255 | 0.437 | - | 255 |

| BAUD RATE (K) | FOSC = 5.0688 MHz | | | 4 MHz | | | 3.579545 MHz | | | 1 MHz | | | 32.768 kHz | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 0.3 | 0.31 | +3.13 | 255 | 0.3005 | -0.17 | 207 | 0.301 | +0.23 | 185 | 0.300 | +0.16 | 51 | 0.256 | -14.67 | 1 |
| 1.2 | 1.2 | 0 | 65 | 1.202 | +1.67 | 51 | 1.190 | -0.83 | 46 | 1.202 | +0.16 | 12 | NA | - | - |
| 2.4 | 2.4 | 0 | 32 | 2.404 | +1.67 | 25 | 2.432 | +1.32 | 22 | 2.232 | -6.99 | 6 | NA | - | - |
| 9.6 | 9.9 | +3.13 | 7 | NA | - | - | 9.322 | -2.90 | 5 | NA | - | - | NA | - | - |
| 19.2 | 19.8 | +3.13 | 3 | NA | - | - | 18.64 | -2.90 | 2 | NA | - | - | NA | - | - |
| 76.8 | 79.2 | +3.13 | 0 | NA | - | - | NA | - | - | NA | - | - | NA | - | - |
| 96 | NA | - | - | NA | - | - | NA | - | - | NA | - | - | NA | - | - |
| 300 | NA | - | - | NA | - | - | NA | - | - | NA | - | - | NA | - | - |
| 500 | NA | - | - | NA | - | - | NA | - | - | NA | - | - | NA | - | - |
| HIGH | 79.2 | - | 0 | 62.500 | - | 0 | 55.93 | - | 0 | 15.63 | - | 0 | 0.512 | - | 0 |
| LOW | 0.3094 | - | 255 | 3.906 | - | 255 | 0.2185 | - | 255 | 0.0610 | - | 255 | 0.0020 | - | 255 |

**TABLE 12-5: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)**

| BAUD RATE (K) | FOSC = 20 MHz | | | 16 MHz | | | 10 MHz | | | 7.16 MHz | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 9.6 | 9.615 | +0.16 | 129 | 9.615 | +0.16 | 103 | 9.615 | +0.16 | 64 | 9.520 | -0.83 | 46 |
| 19.2 | 19.230 | +0.16 | 64 | 19.230 | +0.16 | 51 | 18.939 | -1.36 | 32 | 19.454 | +1.32 | 22 |
| 38.4 | 37.878 | -1.36 | 32 | 38.461 | +0.16 | 25 | 39.062 | +1.7 | 15 | 37.286 | -2.90 | 11 |
| 57.6 | 56.818 | -1.36 | 21 | 58.823 | +2.12 | 16 | 56.818 | -1.36 | 10 | 55.930 | -2.90 | 7 |
| 115.2 | 113.636 | -1.36 | 10 | 111.111 | -3.55 | 8 | 125 | +8.51 | 4 | 111.860 | -2.90 | 3 |
| 250 | 250 | 0 | 4 | 250 | 0 | 3 | NA | - | - | NA | - | - |
| 625 | 625 | 0 | 1 | NA | - | - | 625 | 0 | 0 | NA | - | - |
| 1250 | 1250 | 0 | 0 | NA | - | - | NA | - | - | NA | - | - |

| BAUD RATE (K) | FOSC = 5.068 MHz | | | 4 MHz | | | 3.579 MHz | | | 1 MHz | | | 32.768 kHz | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 9.6 | 9.6 | 0 | 32 | NA | - | - | 9.727 | +1.32 | 22 | 8.928 | -6.99 | 6 | NA | - | - |
| 19.2 | 18.645 | -2.94 | 16 | 1.202 | +0.17 | 207 | 18.643 | -2.90 | 11 | 20.833 | +8.51 | 2 | NA | - | - |
| 38.4 | 39.6 | +3.12 | 7 | 2.403 | +0.13 | 103 | 37.286 | -2.90 | 5 | 31.25 | -18.61 | 1 | NA | - | - |
| 57.6 | 52.8 | -8.33 | 5 | 9.615 | +0.16 | 25 | 55.930 | -2.90 | 3 | 62.5 | +8.51 | 0 | NA | - | - |
| 115.2 | 105.6 | -8.33 | 2 | 19.231 | +0.16 | 12 | 111.860 | -2.90 | 1 | NA | - | - | NA | - | - |
| 250 | NA | - | - | NA | - | - | 223.721 | -10.51 | 0 | NA | - | - | NA | - | - |
| 625 | NA | - | - | NA | - | - | NA | - | - | NA | - | - | NA | - | - |
| 1250 | NA | - | - | NA | - | - | NA | - | - | NA | - | - | NA | - | - |

**Note:** For the PIC16C73/73A/74/74A, the asynchronous high speed mode (BRGH = 1) may experience a high rate of receive errors. It is recommended that BRGH = 0. If you desire a higher baud rate than BRGH = 0 can support, refer to the device errata for additional information, or use the PIC16C76/77.

12.1.1 SAMPLING

The data on the RC7/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin. If bit BRGH (TXSTA<2>) is clear (i.e., at the low baud rates), the sampling is done on the seventh, eighth and ninth falling edges of a x16 clock (Figure 12-3). If bit BRGH is set (i.e., at the high baud rates), the sampling is done on the 3 clock edges preceding the second rising edge after the first falling edge of a x4 clock (Figure 12-4 and Figure 12-5).

**FIGURE 12-3: RX PIN SAMPLING SCHEME. BRGH = 0 (PIC16C73/73A/74/74A)**

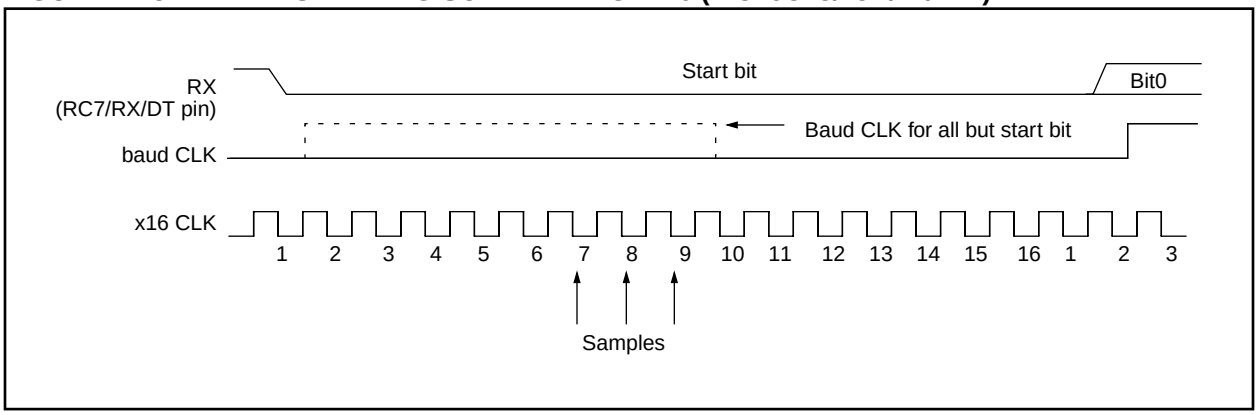



**FIGURE 12-4: RX PIN SAMPLING SCHEME, BRGH = 1 (PIC16C73/73A/74/74A)**

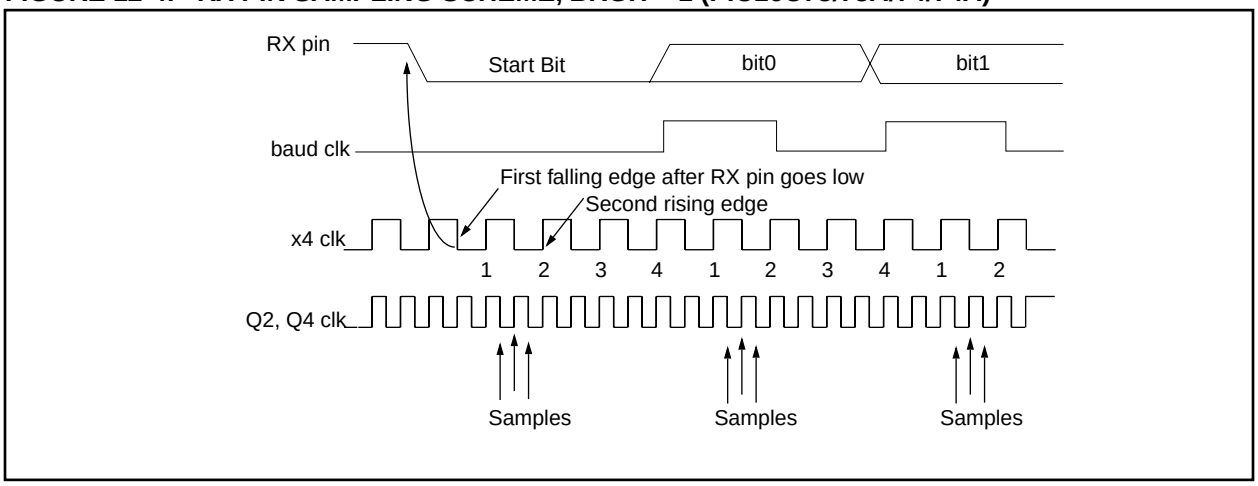



**FIGURE 12-5: RX PIN SAMPLING SCHEME, BRGH = 1 (PIC16C73/73A/74/74A)**

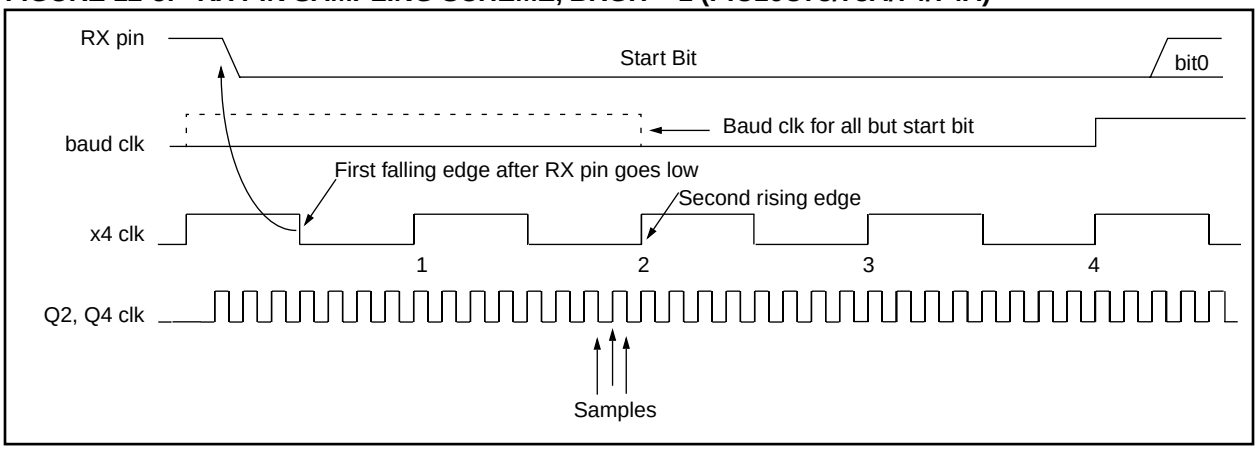
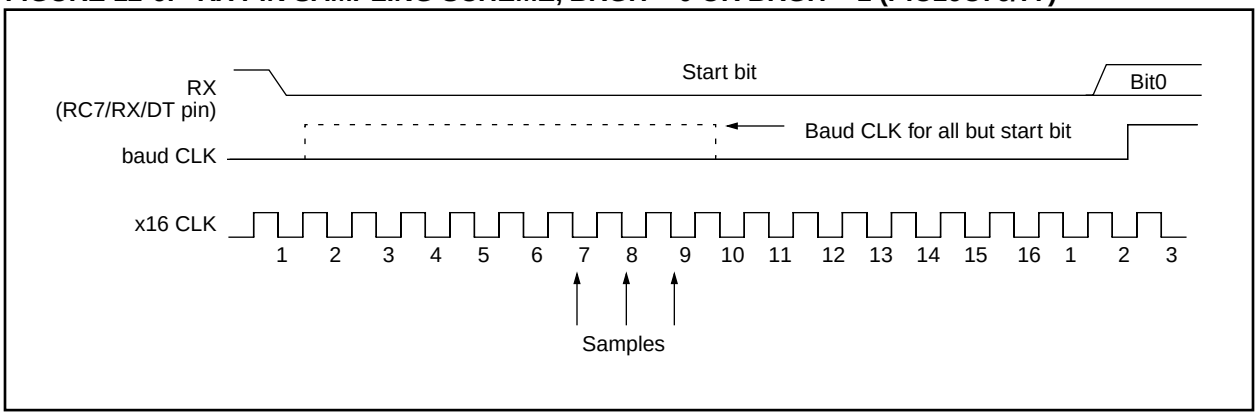


**FIGURE 12-6: RX PIN SAMPLING SCHEME, BRGH = 0 OR BRGH = 1 (PIC16C76/77)**



## 12.2 USART Asynchronous Mode

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

In this mode, the USART uses standard nonreturn-to-zero (NRZ) format (one start bit, eight or nine data bits and one stop bit). The most common data format is 8-bits. An on-chip dedicated 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The USART transmits and receives the LSb first. The USART's transmitter and receiver are functionally independent but use the same data format and baud rate. The baud rate generator produces a clock either x16 or x64 of the bit shift rate, depending on bit BRGH (TXSTA<2>). Parity is not supported by the hardware, but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during SLEEP.

Asynchronous mode is selected by clearing bit SYNC (TXSTA<4>).

The USART Asynchronous module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

### 12.2.1 USART ASYNCHRONOUS TRANSMITTER

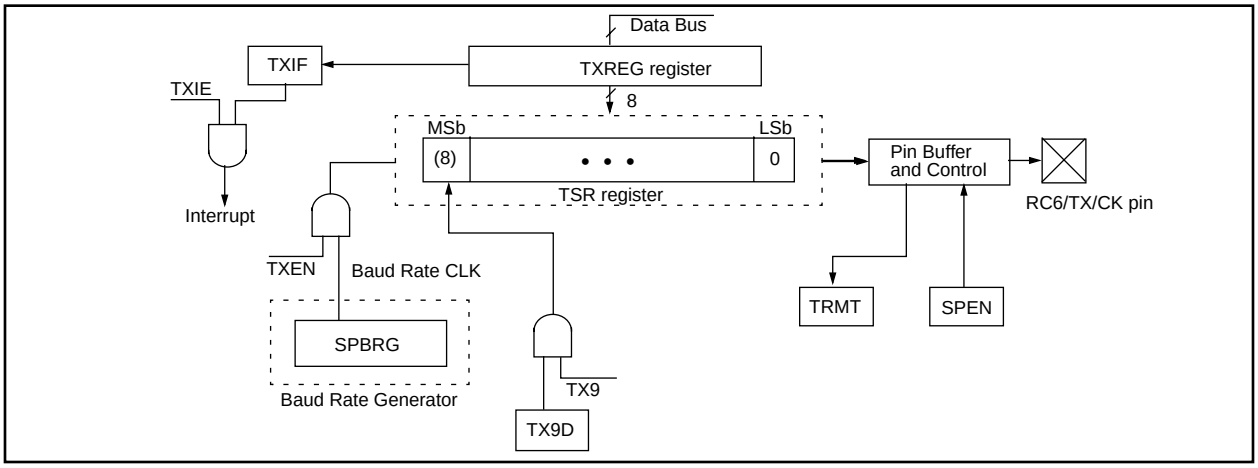
The USART transmitter block diagram is shown in Figure 12-7. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the STOP bit has been transmitted from the previous load. As soon as the STOP bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one TCY), the TXREG register is empty and flag bit TXIF (PIR1<4>) is set. This interrupt can be enabled/disabled by setting/clearing enable bit TXIE ( PIE1<4>). Flag bit TXIF will be set regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicated the status of the TXREG register, another bit TRMT (TXSTA<1>) shows the status of the TSR register. Status bit TRMT is a read only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty.

| | |
|---|---|
| **Note 1:** | The TSR register is not mapped in data memory so it is not available to the user. |
| **Note 2:** | Flag bit TXIF is set when enable bit TXEN is set. |

Transmission is enabled by setting enable bit TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data and the baud rate generator (BRG) has produced a shift clock (Figure 12-7). The transmission can also be started by first loading the TXREG register and then setting enable bit TXEN. Normally when transmission is first started, the TSR register is empty, so a transfer to the TXREG register will result in an immediate transfer to TSR resulting in an empty TXREG. A back-to-back transfer is thus possible (Figure 12-9). Clearing enable bit TXEN during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result the RC6/TX/CK pin will revert to hi-impedance.

In order to select 9-bit transmission, transmit bit TX9 (TXSTA<6>) should be set and the ninth bit should be written to TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG register can result in an immediate transfer of the data to the TSR register (if the TSR is empty). In such a case, an incorrect ninth data bit maybe loaded in the TSR register.

**FIGURE 12-7: USART TRANSMIT BLOCK DIAGRAM**



Steps to follow when setting up an Asynchronous Transmission:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (Section 12.1)
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).

**FIGURE 12-8: ASYNCHRONOUS MASTER TRANSMISSION**

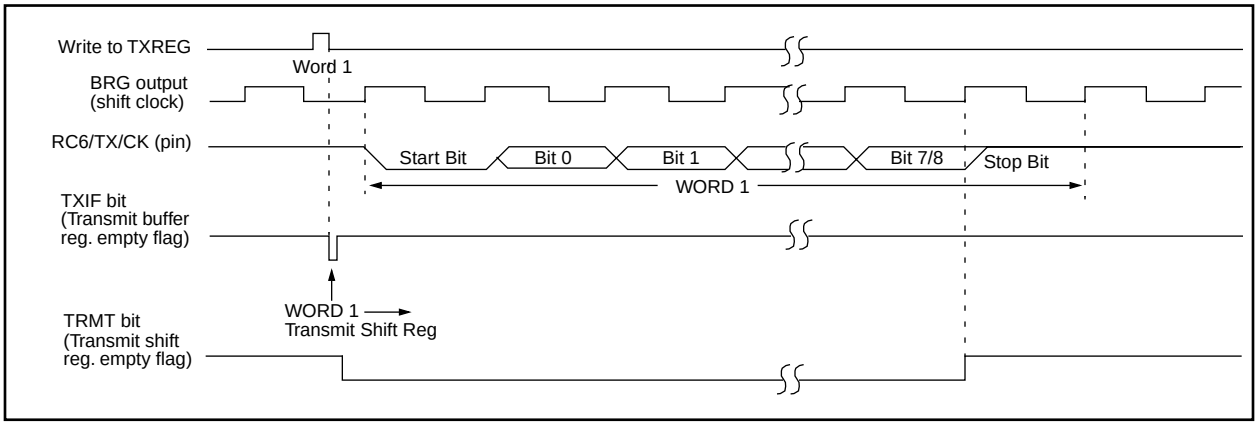
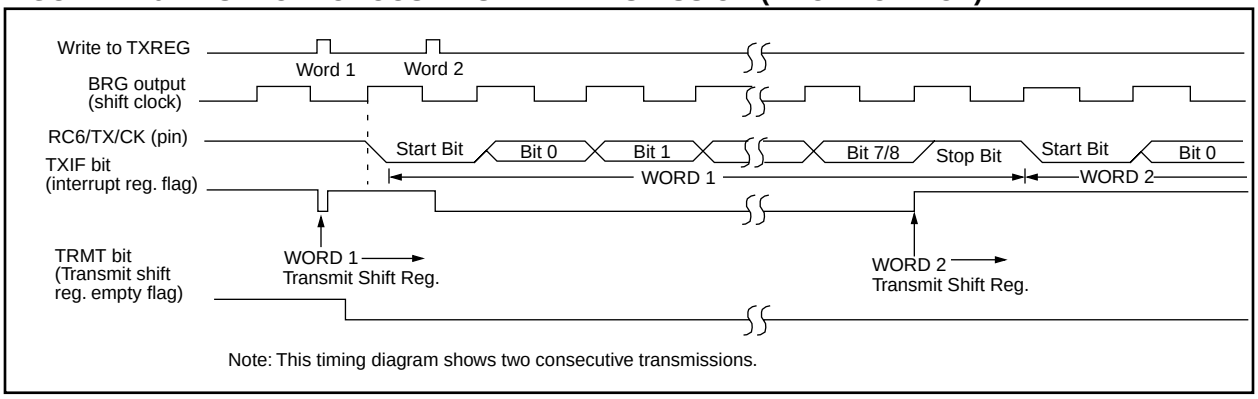



**FIGURE 12-9: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)**




Note: This timing diagram shows two consecutive transmissions.

**TABLE 12-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | PSPIF(1) | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 19h | TXREG | USART Transmit Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Transmission.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

### 12.2.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 12-10. The data is received on the RC7/RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC.

Once Asynchronous mode is selected, reception is enabled by setting bit CREN (RCSTA<4>).

The heart of the receiver is the receive (serial) shift register (RSR). After sampling the STOP bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>). Flag bit RCIF is a read only bit which is cleared by the hardware. It is cleared when the RCREG register has been read and is empty. The RCREG is a double buffered register, i.e. it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte begin shifting to the RSR register. On the detection of the STOP bit of the third byte, if the RCREG register is still full then overrun error bit OERR (RCSTA<1>) will be set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Overrun bit OERR has to be cleared in software. This is done by resetting the receive logic (CREN is cleared and then set). If bit OERR is set, transfers from the RSR register to the RCREG register are inhibited, so it is essential to clear error bit OERR if it is set. Framing error bit FERR (RCSTA<2>) is set if a stop bit is detected as clear. Bit FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG, will load bits RX9D and FERR with new values, therefore it is essential for the user to read the RCSTA register before reading RCREG register in order not to lose the old FERR and RX9D information.

**FIGURE 12-10: USART RECEIVE BLOCK DIAGRAM**

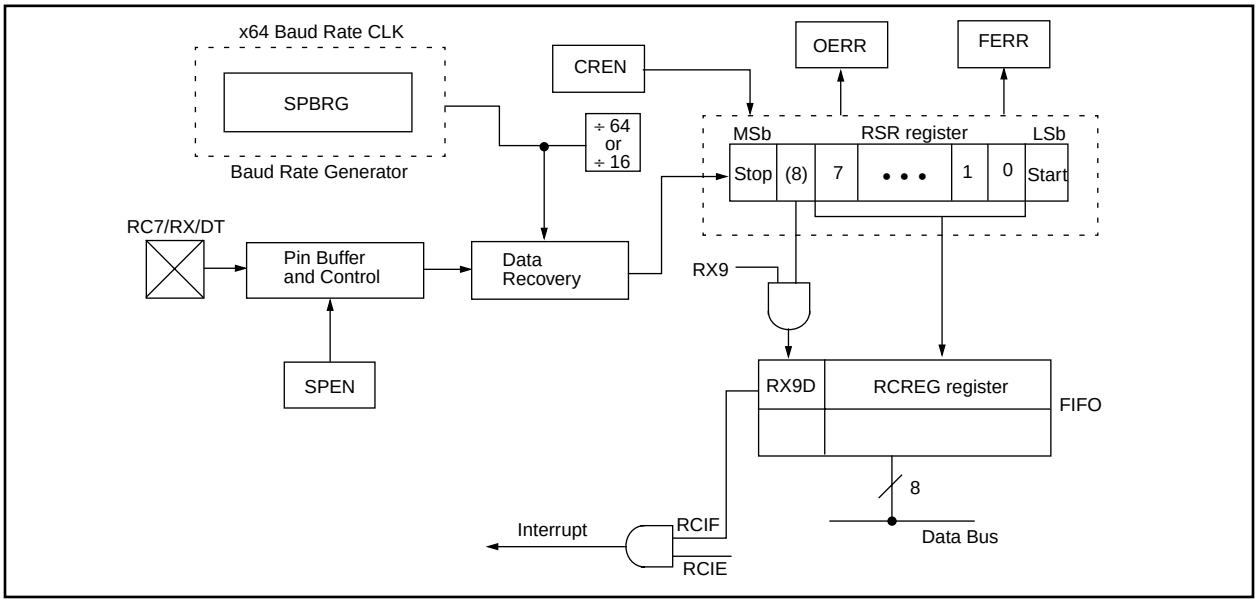
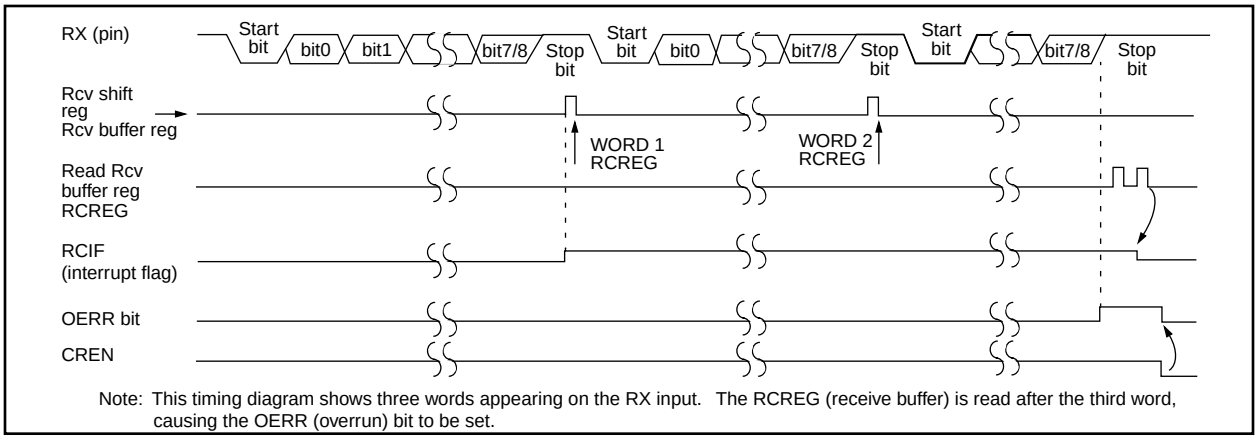



**FIGURE 12-11: ASYNCHRONOUS RECEPTION**




Note: This timing diagram shows three words appearing on the RX input. The RCREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

Steps to follow when setting up an Asynchronous Reception:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH. (Section 12.1).
2. Enable the asynchronous serial port by clearing bit SYNC, and setting bit SPEN.
3. If interrupts are desired, then set enable bit RCIE.
4. If 9-bit reception is desired, then set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
7. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit CREN.

**TABLE 12-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | PSPIF(1) | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 1Ah | RCREG | USART Receive Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Reception.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

## 12.3 USART Synchronous Master Mode

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

In Synchronous Master mode, the data is transmitted in a half-duplex manner i.e. transmission and reception do not occur at the same time. When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit SYNC (TXSTA<4>). In addition enable bit SPEN (RCSTA<7>) is set in order to configure the RC6/TX/CK and RC7/RX/DT I/O pins to CK (clock) and DT (data) lines respectively. The Master mode indicates that the processor transmits the master clock on the CK line. The Master mode is entered by setting bit CSRC (TXSTA<7>).

### 12.3.1 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 12-7. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer register TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available). Once the TXREG register transfers the data to the TSR register (occurs in one Tcycle), the TXREG is empty and interrupt bit, TXIF (PIR1<4>) is set. The interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit TRMT (TXSTA<1>) shows the status of the TSR register. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

Transmission is enabled by setting enable bit TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on the CK line. Data out is stable around the falling edge of the synchronous clock (Figure 12-12). The transmission can also be started by first loading the TXREG register and then setting bit TXEN (Figure 12-13). This is advantageous when slow baud rates are selected, since the BRG is kept in reset when bits TXEN, CREN, and SREN are clear. Setting enable bit TXEN will start the BRG, creating a shift clock immediately. Normally when transmission is first started, the TSR register is empty, so a transfer to the TXREG register will result in an immediate transfer to TSR resulting in an empty TXREG. Back-to-back transfers are possible.

Clearing enable bit TXEN, during a transmission, will cause the transmission to be aborted and will reset the transmitter. The DT and CK pins will revert to hi-impedance. If either bit CREN or bit SREN is set, during a transmission, the transmission is aborted and the DT pin reverts to a hi-impedance state (for a reception). The CK pin will remain an output if bit CSRC is set (internal clock). The transmitter logic however is not reset although it is disconnected from the pins. In order to reset the transmitter, the user has to clear bit TXEN. If bit SREN is set (to interrupt an on-going transmission and receive a single word), then after the single word is received, bit SREN will be cleared and the serial port will revert back to transmitting since bit TXEN is still set. The DT line will immediately switch from hi-impedance receive mode to transmit and start driving. To avoid this, bit TXEN should be cleared.

In order to select 9-bit transmission, the TX9 (TXSTA<6>) bit should be set and the ninth bit should be written to bit TX9D (TXSTA<0>). The ninth bit must be written before writing the 8-bit data to the TXREG register. This is because a data write to the TXREG can result in an immediate transfer of the data to the TSR register (if the TSR is empty). If the TSR was empty and the TXREG was written before writing the “new” TX9D, the “present” value of bit TX9D is loaded.

Steps to follow when setting up a Synchronous Master Transmission:

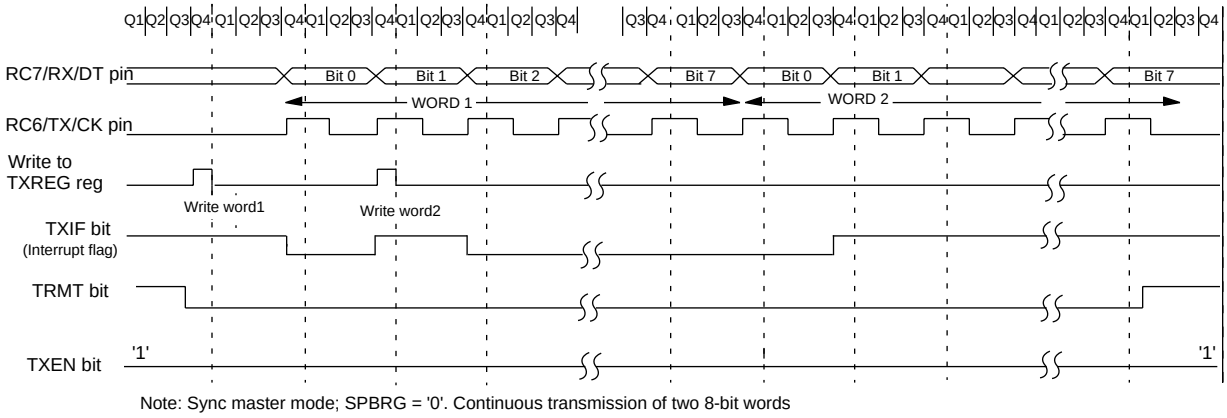
1. Initialize the SPBRG register for the appropriate baud rate (Section 12.1).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN, and CSRC.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set bit TX9.
5. Enable the transmission by setting bit TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG register.

**TABLE 12-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | PSPIF(1) | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 19h | TXREG | USART Transmit Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

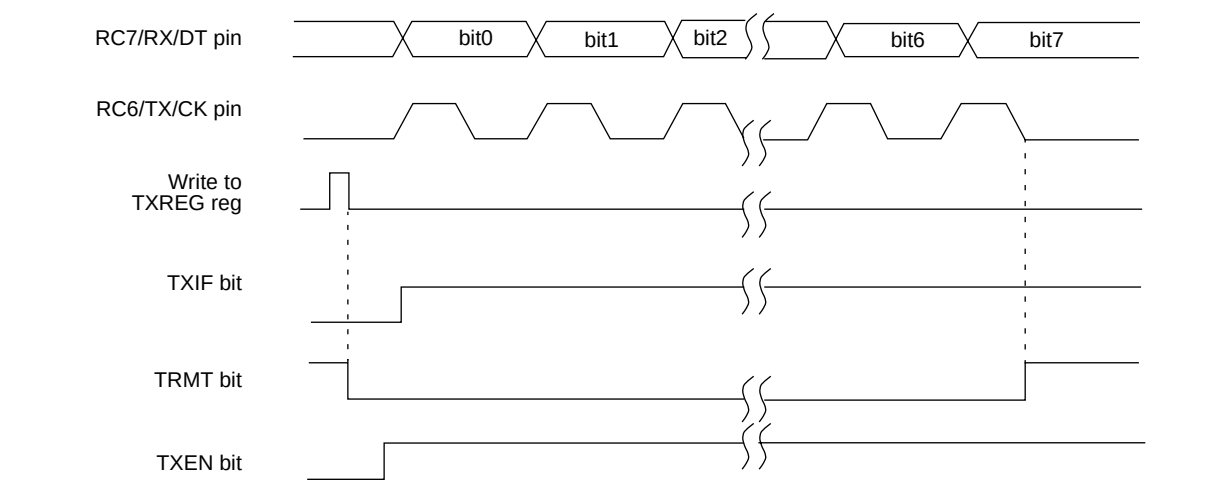
Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for Synchronous Master Transmission.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

**FIGURE 12-12: SYNCHRONOUS TRANSMISSION**




Note: Sync master mode; SPBRG = '0'. Continuous transmission of two 8-bit words

**FIGURE 12-13: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**



### 12.3.2 USART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either enable bit SREN (RCSTA<5>) or enable bit CREN (RCSTA<4>). Data is sampled on the RC7/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set then CREN takes precedence. After clocking the last bit, the received data in the Receive Shift Register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, interrupt flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>). Flag bit RCIF is a read only bit which is reset by the hardware. In this case it is reset when the RCREG register has been read and is empty. The RCREG is a double buffered register, i.e. it is a two deep FIFO. It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full then overrun error bit OERR (RCSTA<1>) is set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Bit OERR has to be cleared in software (by clearing bit CREN). If bit OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear bit OERR if it is set. The 9th receive bit is buffered the same way as the receive data. Reading the RCREG register, will load bit RX9D with a new value, therefore it is essential for the user to read the RCSTA register before reading RCREG in order not to lose the old RX9D information.

Steps to follow when setting up a Synchronous Master Reception:

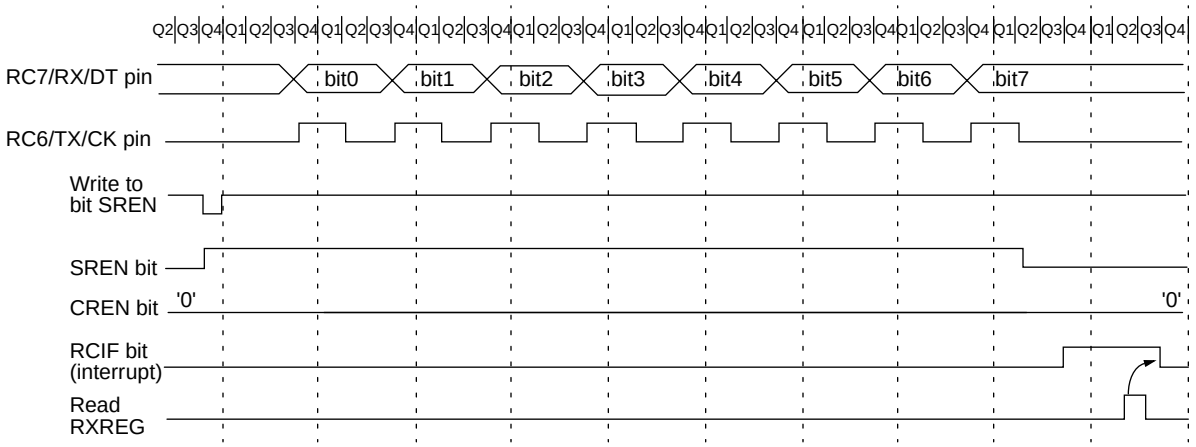
1. Initialize the SPBRG register for the appropriate baud rate. (Section 12.1)
2. Enable the synchronous master serial port by setting bits SYNC, SPEN, and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, then set enable bit RCIE.
5. If 9-bit reception is desired, then set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception set bit CREN.
7. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.

**TABLE 12-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | PSPIF(1) | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 1Ah | RCREG | USART Receive Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Master Reception.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

**FIGURE 12-14: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**




Note: Timing diagram demonstrates SYNC master mode with bit SREN = '1' and bit BRG = '0'.

## 12.4 USART Synchronous Slave Mode

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

Synchronous slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the RC6/TX/CK pin (instead of being supplied internally in master mode). This allows the device to transfer or receive data while in SLEEP mode. Slave mode is entered by clearing bit CSRC (TXSTA<7>).

### 12.4.1 USART SYNCHRONOUS SLAVE TRANSMIT

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

If two words are written to the TXREG and then the SLEEP instruction is executed, the following will occur:

a) The first word will immediately transfer to the TSR register and transmit.
b) The second word will remain in TXREG register.
c) Flag bit TXIF will not be set.
d) When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.
e) If enable bit TXIE is set, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Transmission:

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

### 12.4.2 USART SYNCHRONOUS SLAVE RECEPTION

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

If receive is enabled, by setting bit CREN, prior to the SLEEP instruction, then a word may be received during SLEEP. On completely receiving the word, the RSR register will transfer the data to the RCREG register and if enable bit RCIE bit is set, the interrupt generated will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Steps to follow when setting up a Synchronous Slave Reception:

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

**TABLE 12-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | PSPIF(1) | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 19h | TXREG | USART Transmit Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Transmission.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

**TABLE 12-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | PSPIF(1) | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 1Ah | RCREG | USART Receive Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Reception.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16C73/73A/76, always maintain these bits clear.

**NOTES:**

## 13.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

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

The analog-to-digital (A/D) converter module has five inputs for the PIC16C72/73/73A/76, and eight for the PIC16C74/74A/77.

The A/D allows conversion of an analog input signal to a corresponding 8-bit digital number (refer to Application Note AN546 for use of A/D Converter). The output of the sample and hold is the input into the converter, which generates the result via successive approximation. The analog reference voltage is software selectable to either the device's positive supply voltage ($V_{DD}$) or the voltage level on the RA3/AN3/$V_{REF}$ pin.

The A/D converter has a unique feature of being able to operate while the device is in SLEEP mode. To operate in sleep, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

The A/D module has three registers. These registers are:

- A/D Result Register (ADRES)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)

The ADCON0 register, shown in Figure 13-1, controls the operation of the A/D module. The ADCON1 register, shown in Figure 13-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be a voltage reference) or as digital I/O.

**FIGURE 13-1: ADCON0 REGISTER (ADDRESS 1Fh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | — | ADON |
| bit7 | | | | | | | bit0 |

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

bit 7-6: **ADCS1:ADCS0:** A/D Conversion Clock Select bits
00 = $F_{OSC}$/2
01 = $F_{OSC}$/8
10 = $F_{OSC}$/32
11 = $F_{RC}$ (clock derived from an internal RC oscillator)

bit 5-3: **CHS2:CHS0**: Analog Channel Select bits
000 = channel 0, (RA0/AN0)
001 = channel 1, (RA1/AN1)
010 = channel 2, (RA2/AN2)
011 = channel 3, (RA3/AN3)
100 = channel 4, (RA5/AN4)
101 = channel 5, (RE0/AN5)**(1)**
110 = channel 6, (RE1/AN6)**(1)**
111 = channel 7, (RE2/AN7)**(1)**

bit 2: **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit

If ADON = 1
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (This bit is automatically cleared by hardware when the A/D conversion is complete)

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

bit 0: **ADON**: A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shutoff and consumes no operating current

**Note 1:** A/D channels 5, 6, and 7 are implemented on the PIC16C74/74A/77 only.

**FIGURE 13-2: ADCON1 REGISTER (ADDRESS 9Fh)**

| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 |
| bit7 | | | | | | | bit0 |

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

bit 7-3: **Unimplemented:** Read as '0'

bit 2-0: **PCFG2:PCFG0**: A/D Port Configuration Control bits

| PCFG2:PCFG0 | RA0 | RA1 | RA2 | RA5 | RA3 | RE0(1) | RE1(1) | RE2(1) | VREF |
|---|---|---|---|---|---|---|---|---|---|
| 000 | A | A | A | A | A | A | A | A | VDD |
| 001 | A | A | A | A | VREF | A | A | A | RA3 |
| 010 | A | A | A | A | A | D | D | D | VDD |
| 011 | A | A | A | A | VREF | D | D | D | RA3 |
| 100 | A | A | D | D | A | D | D | D | VDD |
| 101 | A | A | D | D | VREF | D | D | D | RA3 |
| 11x | D | D | D | D | D | D | D | D | — |

A = Analog input

D = Digital I/O

**Note 1:** RE0, RE1, and RE2 are implemented on the PIC16C74/74A/77 only.

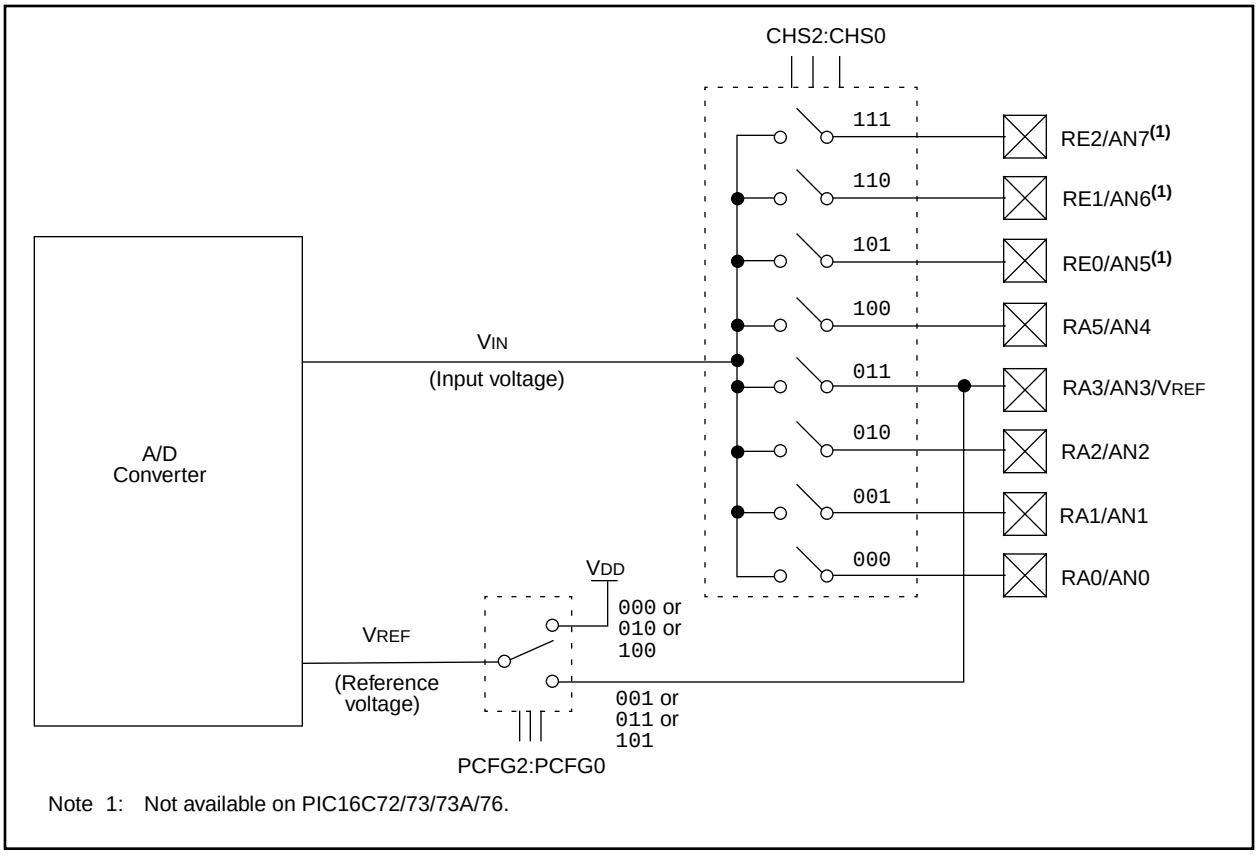
The ADRES register contains the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRES register, the GO/$\overline{\text{DONE}}$ bit (ADCON0<2>) is cleared, and A/D interrupt flag bit ADIF is set. The block diagrams of the A/D module are shown in Figure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

## 13.1 A/D Acquisition Requirements

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

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 13-4. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), Figure 13-4. The source impedance affects the offset voltage at the analog input (due to pin leakage current). **The maximum recommended impedance for analog sources is 10 kΩ.** After the analog input channel is selected (changed) this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 13-1 may be used. This equation calculates the acquisition time to within 1/2 LSb error is used (512 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified accuracy.

**EQUATION 13-1: A/D MINIMUM CHARGING TIME**

$$V_{HOLD} = (V_{REF} - (V_{REF}/512)) \cdot (1 - e^{(-T_{CAP}/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$$

Given: $V_{HOLD} = (V_{REF}/512)$, for 1/2 LSb resolution

The above equation reduces to:

$$T_{CAP} = -(51.2\ pF)(1\ k\Omega + R_{SS} + R_S) \ln(1/511)$$

Example 13-1 shows the calculation of the minimum required acquisition time TACQ. This calculation is based on the following system assumptions.

CHOLD = 51.2 pF

Rs = 10 kΩ

1/2 LSb error

VDD = 5V → Rss = 7 kΩ

Temp (application system max.) = 50°C

VHOLD = 0 @ t = 0

**Note 1:** The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

**Note 2:** The charge holding capacitor (CHOLD) is not discharged after each conversion.

**Note 3:** The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

**Note 4:** After a conversion has completed, a 2.0TAD delay must complete before acquisition can begin again. During this time the holding capacitor is not connected to the selected A/D input channel.

**EXAMPLE 13-1: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME**

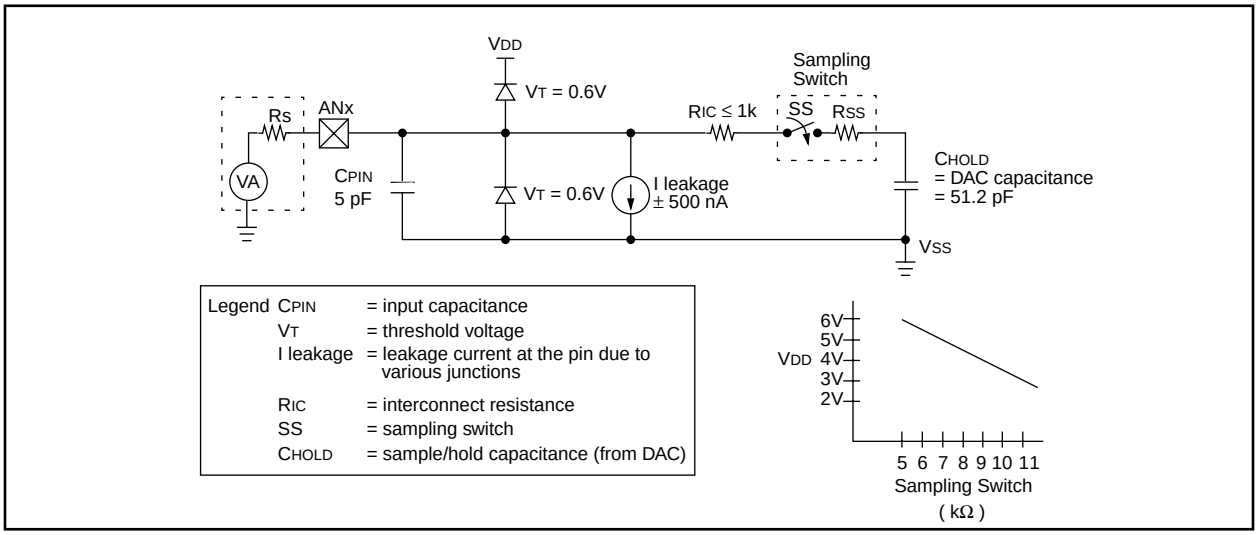
TACQ = Amplifier Settling Time +
Holding Capacitor Charging Time +
Temperature Coefficient

TACQ = 5 µs + TCAP + [(Temp - 25°C)(0.05 µs/°C)]

TCAP = -CHOLD (RIC + RSS + RS) ln(1/511)
-51.2 pF (1 kΩ + 7 kΩ + 10 kΩ) ln(0.0020)
-51.2 pF (18 kΩ) ln(0.0020)
-0.921 µs (-6.2364)
5.747 µs

TACQ = 5 µs + 5.747 µs + [(50°C - 25°C)(0.05 µs/°C)]
10.747 µs + 1.25 µs
11.997 µs

**FIGURE 13-4: ANALOG INPUT MODEL**



### 13.2 Selecting the A/D Conversion Clock

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

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 9.5TAD per 8-bit conversion. The source of the A/D conversion clock is software selectable. The four possible options for TAD are:

- 2TOSC
- 8TOSC
- 32TOSC
- Internal RC oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 µs.

Table 13-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

### 13.3 Configuring Analog Port Pins

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

The ADCON1, TRISA, and TRISE registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS2:CHS0 bits and the TRIS bits.

**Note 1:** When reading the port register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs, will convert an analog input. Analog levels on a digitally configured input will not affect the conversion accuracy.

**Note 2:** Analog levels on any pin that is defined as a digital input (including the AN7:AN0 pins), may cause the input buffer to consume current that is out of the devices specification.

**TABLE 13-1: TAD vs. DEVICE OPERATING FREQUENCIES**

| AD Clock Source (TAD) | | Device Frequency | | | |
|---|---|---|---|---|---|
| **Operation** | **ADCS1:ADCS0** | **20 MHz** | **5 MHz** | **1.25 MHz** | **333.33 kHz** |
| 2TOSC | 00 | 100 ns(2) | 400 ns(2) | 1.6 µs | 6 µs |
| 8TOSC | 01 | 400 ns(2) | 1.6 µs | 6.4 µs | 24 µs(3) |
| 32TOSC | 10 | 1.6 µs | 6.4 µs | 25.6 µs(3) | 96 µs(3) |
| RC(5) | 11 | 2 - 6 µs(1,4) | 2 - 6 µs(1,4) | 2 - 6 µs(1,4) | 2 - 6 µs(1) |

Legend: Shaded cells are outside of recommended range.

Note 1: The RC source has a typical TAD time of 4 µs.

2: These values violate the minimum required TAD time.

3: For faster conversion times, the selection of another clock source is recommended.

4: When device frequency is greater than 1 MHz, the RC A/D conversion clock source is recommended for sleep operation only.

5: For extended voltage devices (LC), please refer to Electrical Specifications section.

## 13.4 A/D Conversions

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

Example 13-2 shows how to perform an A/D conversion. The RA pins are configured as analog inputs. The analog reference ($V_{REF}$) is the device $V_{DD}$. The A/D interrupt is enabled, and the A/D conversion clock is FRC. The conversion is performed on the RA0 pin (channel 0).

**Note:** The GO/$\overline{\text{DONE}}$ bit should **NOT** be set in the same instruction that turns on the A/D.

Clearing the GO/$\overline{\text{DONE}}$ bit during a conversion will abort the current conversion. The ADRES register will NOT be updated with the partially completed A/D conversion sample. That is, the ADRES register will continue to contain the value of the last completed conversion (or the last value written to the ADRES register). After the A/D conversion is aborted, a $2T_{AD}$ wait is required before the next acquisition is started. After this $2T_{AD}$ wait, an acquisition is automatically started on the selected channel.

**EXAMPLE 13-2: A/D CONVERSION**

```
     BSF     STATUS, RP0         ; Select Bank 1
     BCF     STATUS, RP1         ; PIC16C76/77 only
     CLRF    ADCON1              ; Configure A/D inputs
     BSF     PIE1,   ADIE        ; Enable A/D interrupts
     BCF     STATUS, RP0         ; Select Bank 0
     MOVLW   0xC1                ; RC Clock, A/D is on, Channel 0 is selected
     MOVWF   ADCON0              ;
     BCF     PIR1,   ADIF        ; Clear A/D interrupt flag bit
     BSF     INTCON, PEIE        ; Enable peripheral interrupts
     BSF     INTCON, GIE         ; Enable all interrupts
;
;  Ensure that the required sampling time for the selected input channel has elapsed.
;  Then the conversion may be started.
;
     BSF     ADCON0, GO          ; Start A/D Conversion
       :                         ; The ADIF bit will be set and the GO/DONE bit
       :                         ;  is cleared upon completion of the A/D Conversion.
```

### 13.4.1 FASTER CONVERSION - LOWER RESOLUTION TRADE-OFF

Not all applications require a result with 8-bits of resolution, but may instead require a faster conversion time. The A/D module allows users to make the trade-off of conversion speed to resolution. Regardless of the resolution required, the acquisition time is the same. To speed up the conversion, the clock source of the A/D module may be switched so that the TAD time violates the minimum specified time (see the applicable electrical specification). Once the TAD time violates the minimum specified time, all the following A/D result bits are not valid (see A/D Conversion Timing in the Electrical Specifications section.) The clock sources may only be switched between the three oscillator versions (cannot be switched from/to RC). The equation to determine the time before the oscillator can be switched is as follows:

Conversion time = $2T_{AD} + N \bullet T_{AD} + (8 - N)(2T_{OSC})$
Where: N = number of bits of resolution required.

Since the TAD is based from the device oscillator, the user must use some method (a timer, software loop, etc.) to determine when the A/D oscillator may be changed. Example 13-3 shows a comparison of time required for a conversion with 4-bits of resolution, versus the 8-bit resolution conversion. The example is for devices operating at 20 MHz and 16 MHz (The A/D clock is programmed for $32T_{OSC}$), and assumes that immediately after $6T_{AD}$, the A/D clock is programmed for $2T_{OSC}$.

The $2T_{OSC}$ violates the minimum TAD time since the last 4-bits will not be converted to correct values.

**EXAMPLE 13-3: 4-BIT vs. 8-BIT CONVERSION TIMES**

| | Freq. (MHz)(1) | Resolution | |
|---|---|---|---|
| | | 4-bit | 8-bit |
| TAD | 20 | 1.6 µs | 1.6 µs |
| | 16 | 2.0 µs | 2.0 µs |
| TOSC | 20 | 50 ns | 50 ns |
| | 16 | 62.5 ns | 62.5 ns |
| $2T_{AD} + N \bullet T_{AD} + (8 - N)(2T_{OSC})$ | 20 | 10 µs | 16 µs |
| | 16 | 12.5 µs | 20 µs |

Note 1: PIC16C7X devices have a minimum TAD time of 1.6 µs.

### 13.5 A/D Operation During Sleep

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

The A/D module can operate during SLEEP mode. This requires that the A/D clock source be set to RC (ADCS1:ADCS0 = 11). When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. This allows the SLEEP instruction to be executed, which eliminates all digital switching noise from the conversion. When the conversion is completed the GO/$\overline{\text{DONE}}$ bit will be cleared, and the result loaded into the ADRES register. If the A/D interrupt is enabled, the device will wake-up from SLEEP. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

When the A/D clock source is another clock option (not RC), a SLEEP instruction will cause the present conversion to be aborted and the A/D module to be turned off, though the ADON bit will remain set.

Turning off the A/D places the A/D module in its lowest current consumption state.

| **Note:** | For the A/D module to operate in SLEEP, the A/D clock source must be set to RC (ADCS1:ADCS0 = 11). To perform an A/D conversion in SLEEP, ensure the SLEEP instruction immediately follows the instruction that sets the GO/$\overline{\text{DONE}}$ bit. |
|---|---|

### 13.6 A/D Accuracy/Error

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

The absolute accuracy specified for the A/D converter includes the sum of all contributions for quantization error, integral error, differential error, full scale error, offset error, and monotonicity. It is defined as the maximum deviation from an actual transition versus an ideal transition for any code. The absolute error of the A/D converter is specified at < ±1 LSb for $V_{DD} = V_{REF}$ (over the device's specified operating range). However, the accuracy of the A/D converter will degrade as $V_{DD}$ diverges from $V_{REF}$.

For a given range of analog inputs, the output digital code will be the same. This is due to the quantization of the analog input to a digital code. Quantization error is typically ± 1/2 LSb and is inherent in the analog to digital conversion process. The only way to reduce quantization error is to increase the resolution of the A/D converter.

Offset error measures the first actual transition of a code versus the first ideal transition of a code. Offset error shifts the entire transfer function. Offset error can be calibrated out of a system or introduced into a system through the interaction of the total leakage current and source impedance at the analog input.

Gain error measures the maximum deviation of the last actual transition and the last ideal transition adjusted for offset error. This error appears as a change in slope of the transfer function. The difference in gain error to full scale error is that full scale does not take offset error into account. Gain error can be calibrated out in software.

Linearity error refers to the uniformity of the code changes. Linearity errors cannot be calibrated out of the system. Integral non-linearity error measures the actual code transition versus the ideal code transition adjusted by the gain error for each code.

Differential non-linearity measures the maximum actual code width versus the ideal code width. This measure is unadjusted.

The maximum pin leakage current is ± 1 µA.

In systems where the device frequency is low, use of the A/D RC clock is preferred. At moderate to high frequencies, $T_{AD}$ should be derived from the device oscillator. $T_{AD}$ must not violate the minimum and should be ≤ 8 µs for preferred operation. This is because $T_{AD}$, when derived from $T_{OSC}$, is kept away from on-chip phase clock transitions. This reduces, to a large extent, the effects of digital switching noise. This is not possible with the RC derived clock. The loss of accuracy due to digital switching noise can be significant if many I/O pins are active.

In systems where the device will enter SLEEP mode after the start of the A/D conversion, the RC clock source selection is required. In this mode, the digital noise from the modules in SLEEP are stopped. This method gives high accuracy.

### 13.7 Effects of a RESET

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

A device reset forces all registers to their reset state. This forces the A/D module to be turned off, and any conversion is aborted.

The value that is in the ADRES register is not modified for a Power-on Reset. The ADRES register will contain unknown data after a Power-on Reset.

### 13.8 Use of the CCP Trigger

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

**Note:** In the PIC16C72, the "special event trigger" is implemented in the CCP1 module.

An A/D conversion can be started by the “special event trigger” of the CCP2 module (CCP1 on the PIC16C72 only). This requires that the CCP2M3:CCP2M0 bits (CCP2CON<3:0>) be programmed as 1011 and that the A/D module is enabled (ADON bit is set). When the trigger occurs, the GO/$\overline{\text{DONE}}$ bit will be set, starting the A/D conversion, and the Timer1 counter will be reset to zero. Timer1 is reset to automatically repeat the A/D acquisition period with minimal software overhead (moving the ADRES to the desired location). The appropriate analog input channel must be selected and the minimum acquisition done before the “special event trigger” sets the GO/$\overline{\text{DONE}}$ bit (starts a conversion).

If the A/D module is not enabled (ADON is cleared), then the “special event trigger” will be ignored by the A/D module, but will still reset the Timer1 counter.

### 13.9 Connection Considerations

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

If the input voltage exceeds the rail values ($V_{SS}$ or $V_{DD}$) by greater than 0.2V, then the accuracy of the conversion is out of specification.

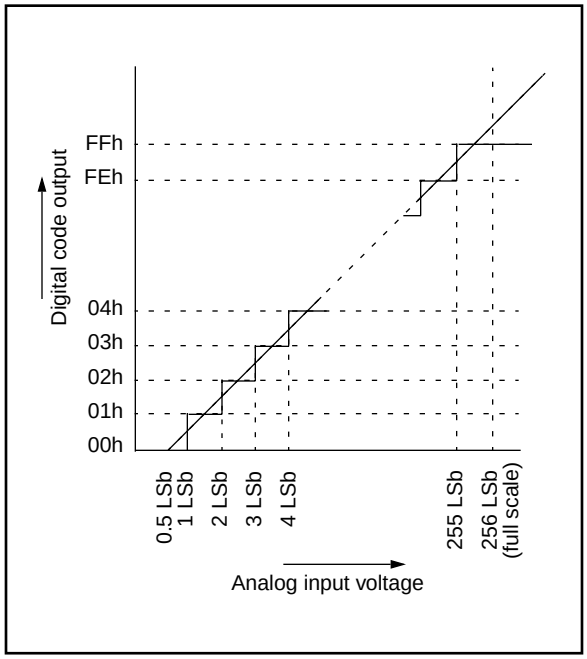
An external RC filter is sometimes added for anti-aliasing of the input signal. The R component should be selected to ensure that the total source impedance is kept under the 10 kΩ recommended specification. Any external components connected (via hi-impedance) to an analog input pin (capacitor, zener diode, etc.) should have very little leakage current at the pin.

### 13.10 Transfer Function

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

The ideal transfer function of the A/D converter is as follows: the first transition occurs when the analog input voltage ($V_{AIN}$) is Analog $V_{REF}$/256 (Figure 13-5).

**FIGURE 13-5: A/D TRANSFER FUNCTION**




### 13.11 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 13-6: FLOWCHART OF A/D OPERATION**

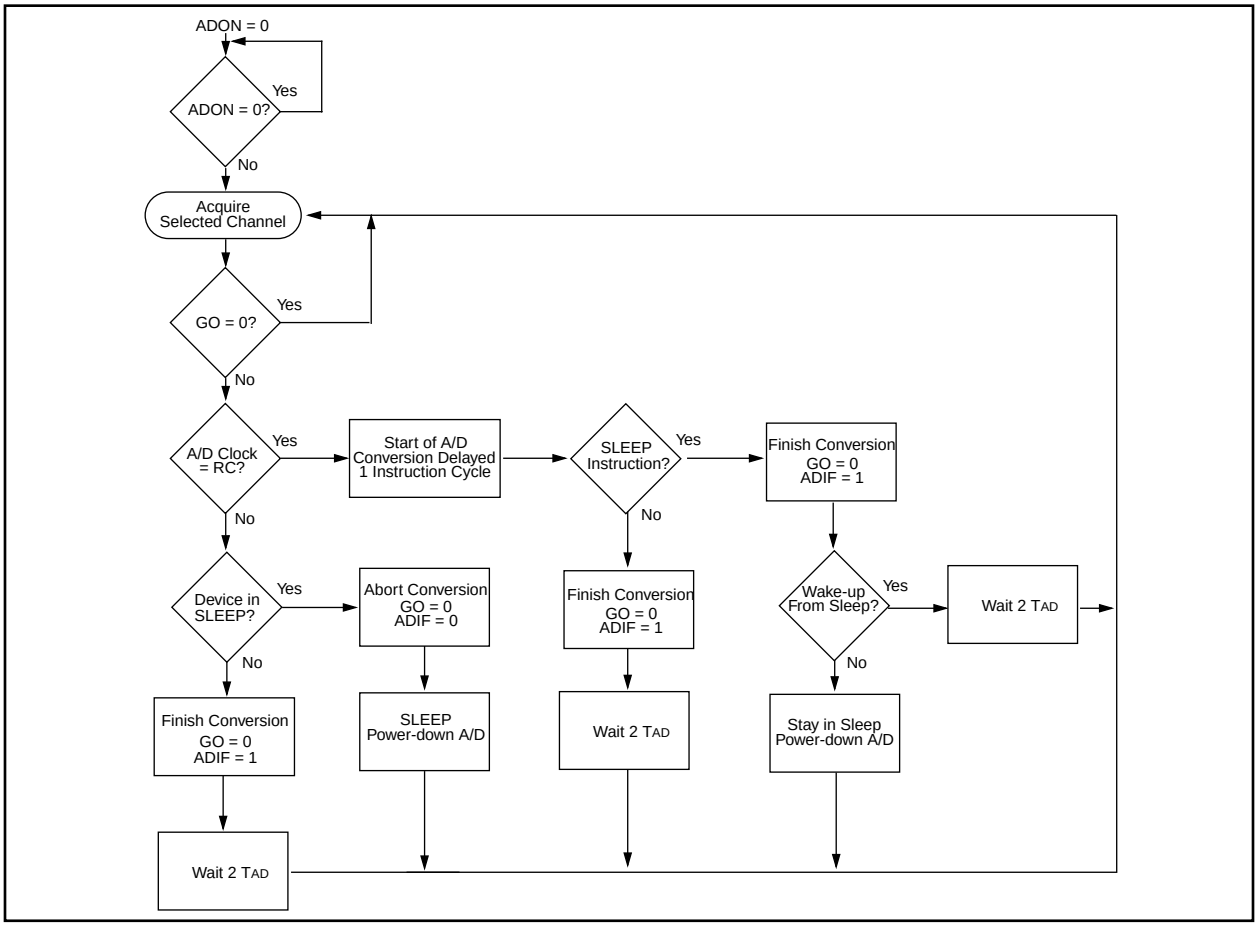



**TABLE 13-2: REGISTERS/BITS ASSOCIATED WITH A/D, PIC16C72**

| 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 | — | ADIF | — | — | SSPIF | CCP1IF | TMR2IF | TMR1IF | -0-- 0000 | -0-- 0000 |
| 8Ch | PIE1 | — | ADIE | — | — | SSPIE | CCP1IE | TMR2IE | TMR1IE | -0-- 0000 | -0-- 0000 |
| 1Eh | ADRES | A/D Result Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | — | ADON | 0000 00-0 | 0000 00-0 |
| 9Fh | ADCON1 | — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 | ---- -000 | ---- -000 |
| 05h | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | --0x 0000 | --0u 0000 |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | --11 1111 | --11 1111 |

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for A/D conversion.

**TABLE 13-3: SUMMARY OF A/D REGISTERS, PIC16C73/73A/74/74A/76/77**

| 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 |
| 0Dh | PIR2 | — | — | — | — | — | — | — | CCP2IF | ---- ---0 | ---- ---0 |
| 8Dh | PIE2 | — | — | — | — | — | — | — | CCP2IE | ---- ---0 | ---- ---0 |
| 1Eh | ADRES | A/D Result Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | — | ADON | 0000 00-0 | 0000 00-0 |
| 9Fh | ADCON1 | — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 | ---- -000 | ---- -000 |
| 05h | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | --0x 0000 | --0u 0000 |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | --11 1111 | --11 1111 |
| 09h | PORTE | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |
| 89h | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction Bits | | | 0000 -111 | 0000 -111 |

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for A/D conversion.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC6C73/73A/76, always maintain these bits clear.

**NOTES:**

## 14.0 SPECIAL FEATURES OF THE CPU

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

What sets a microcontroller apart from other processors are special circuits to deal with the needs of real-time applications. The PIC16CXX family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection. These are:

- Oscillator selection
- Reset
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- SLEEP
- Code protection
- ID locations
- In-circuit serial programming

The PIC16CXX has a Watchdog Timer which can be shut off only through configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only, designed to keep the part in reset while the power supply stabilizes. With these two timers on-chip, most applications need no external reset circuitry.

SLEEP mode is designed to offer a very low current power-down mode. The user can wake-up from SLEEP through external reset, Watchdog Timer Wake-up, or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits are used to select various options.

### 14.1 Configuration Bits

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

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h - 3FFFh), which can be accessed only during programming.

**FIGURE 14-1: CONFIGURATION WORD FOR PIC16C73/74**

| bit13 | | | | | | | | | | | | | | bit0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | — | CP1 | CP0 | PWRTE | WDTE | FOSC1 | FOSC0 | |

Register: CONFIG
Address 2007h

bit 13-5: **Unimplemented**: Read as '1'

bit 4: **CP1:CP0**: Code protection bits
11 = Code protection off
10 = Upper half of program memory code protected
01 = Upper 3/4th of program memory code protected
00 = All memory is code protected

bit 3: **PWRTE**: Power-up Timer Enable bit
1 = Power-up Timer enabled
0 = Power-up Timer disabled

bit 2: **WDTE**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 1-0: **FOSC1:FOSC0**: Oscillator Selection bits
11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

**FIGURE 14-2: CONFIGURATION WORD FOR PIC16C72/73A/74A/76/77**

| CP1 | CP0 | CP1 | CP0 | CP1 | CP0 | — | BODEN | CP1 | CP0 | $\overline{\text{PWRTE}}$ | WDTE | FOSC1 | FOSC0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bit13 | | | | | | | | | | | | | bit0 |

Register: CONFIG
Address 2007h

bit 13-8 **CP1:CP0**: Code Protection bits **(2)**
5-4: 11 = Code protection off
10 = Upper half of program memory code protected
01 = Upper 3/4th of program memory code protected
00 = All memory is code protected

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

bit 6: **BODEN**: Brown-out Reset Enable bit **(1)**
1 = BOR enabled
0 = BOR disabled

bit 3: **$\overline{\text{PWRTE}}$**: Power-up Timer Enable bit **(1)**
1 = PWRT disabled
0 = PWRT enabled

bit 2: **WDTE**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 1-0: **FOSC1:FOSC0**: Oscillator Selection bits
11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

Note 1: Enabling Brown-out Reset automatically enables Power-up Timer (PWRT) regardless of the value of bit $\overline{\text{PWRTE}}$. Ensure the Power-up Timer is enabled anytime Brown-out Reset is enabled.
2: All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed.

## 14.2 Oscillator Configurations

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

### 14.2.1 OSCILLATOR TYPES

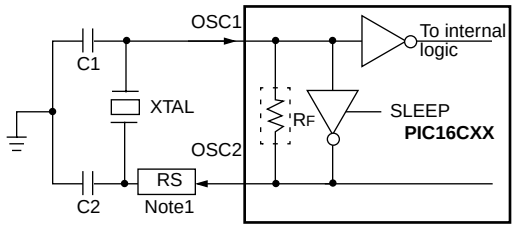
The PIC16CXX can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

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

### 14.2.2 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 14-3). 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 14-4).

**FIGURE 14-3: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)**




See Table 14-1 and Table 14-2 for recommended values of C1 and C2.

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

**FIGURE 14-4: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)**

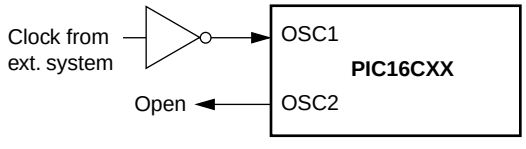



**TABLE 14-1: CERAMIC RESONATORS**

| Ranges Tested: | | | |
|---|---|---|---|
| **Mode** | **Freq** | **OSC1** | **OSC2** |
| XT | 455 kHz<br>2.0 MHz<br>4.0 MHz | 68 - 100 pF<br>15 - 68 pF<br>15 - 68 pF | 68 - 100 pF<br>15 - 68 pF<br>15 - 68 pF |
| HS | 8.0 MHz<br>16.0 MHz | 10 - 68 pF<br>10 - 22 pF | 10 - 68 pF<br>10 - 22 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 14-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR**

| Osc Type | Crystal Freq | Cap. Range C1 | Cap. Range C2 |
|---|---|---|---|
| LP | 32 kHz | 33 pF | 33 pF |
| | 200 kHz | 15 pF | 15 pF |
| XT | 200 kHz | 47-68 pF | 47-68 pF |
| | 1 MHz | 15 pF | 15 pF |
| | 4 MHz | 15 pF | 15 pF |
| HS | 4 MHz | 15 pF | 15 pF |
| | 8 MHz | 15-33 pF | 15-33 pF |
| | 20 MHz | 15-33 pF | 15-33 pF |
| **These values are for design guidance only.** See notes at bottom of page. | | | |
| **Crystals Used** | | | |
| 32 kHz | Epson C-001R32.768K-A | ± 20 PPM | |
| 200 kHz | STD XTL 200.000KHz | ± 20 PPM | |
| 1 MHz | ECS ECS-10-13-1 | ± 50 PPM | |
| 4 MHz | ECS ECS-40-20-1 | ± 50 PPM | |
| 8 MHz | EPSON CA-301 8.000M-C | ± 30 PPM | |
| 20 MHz | EPSON CA-301 20.000M-C | ± 30 PPM | |

Note 1: Recommended values of C1 and C2 are identical to the ranges tested (Table 14-1).
2: Higher capacitance increases the stability of oscillator but also increases the start-up time.
3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
4: Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification.

### 14.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance, or one with parallel resonance.

Figure 14-5 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

**FIGURE 14-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT**

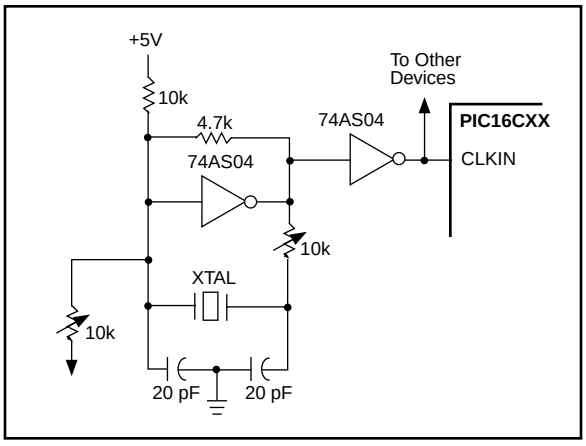
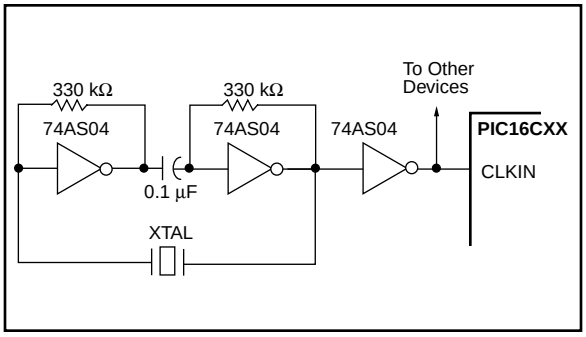



Figure 14-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

**FIGURE 14-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT**




### 14.2.4 RC OSCILLATOR

For timing insensitive applications the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (Rext) and capacitor (Cext) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low Cext values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 14-7 shows how the R/C combination is connected to the PIC16CXX. For Rext values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high Rext values (e.g. 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep Rext between 3 kΩ and 100 kΩ.

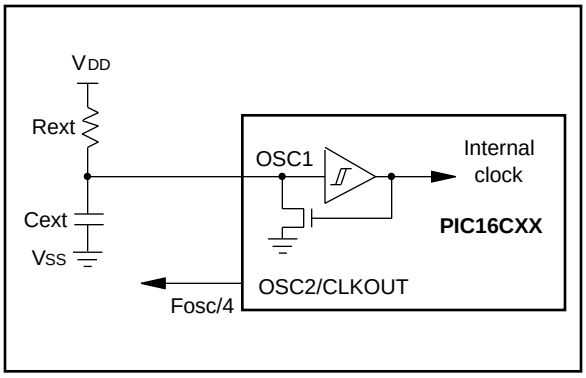
Although the oscillator will operate with no external capacitor (Cext = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance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

See characterization data for desired device for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See characterization data for desired device for variation of oscillator frequency due to $V_{DD}$ for given Rext/Cext values as well as frequency variation due to operating temperature for given R, C, and $V_{DD}$ values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin, and can be used for test purposes or to synchronize other logic (see Figure 3-4 for waveform).

**FIGURE 14-7: RC OSCILLATOR MODE**



## 14.3 Reset

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

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) (PIC16C72/73A/74A/76/77)

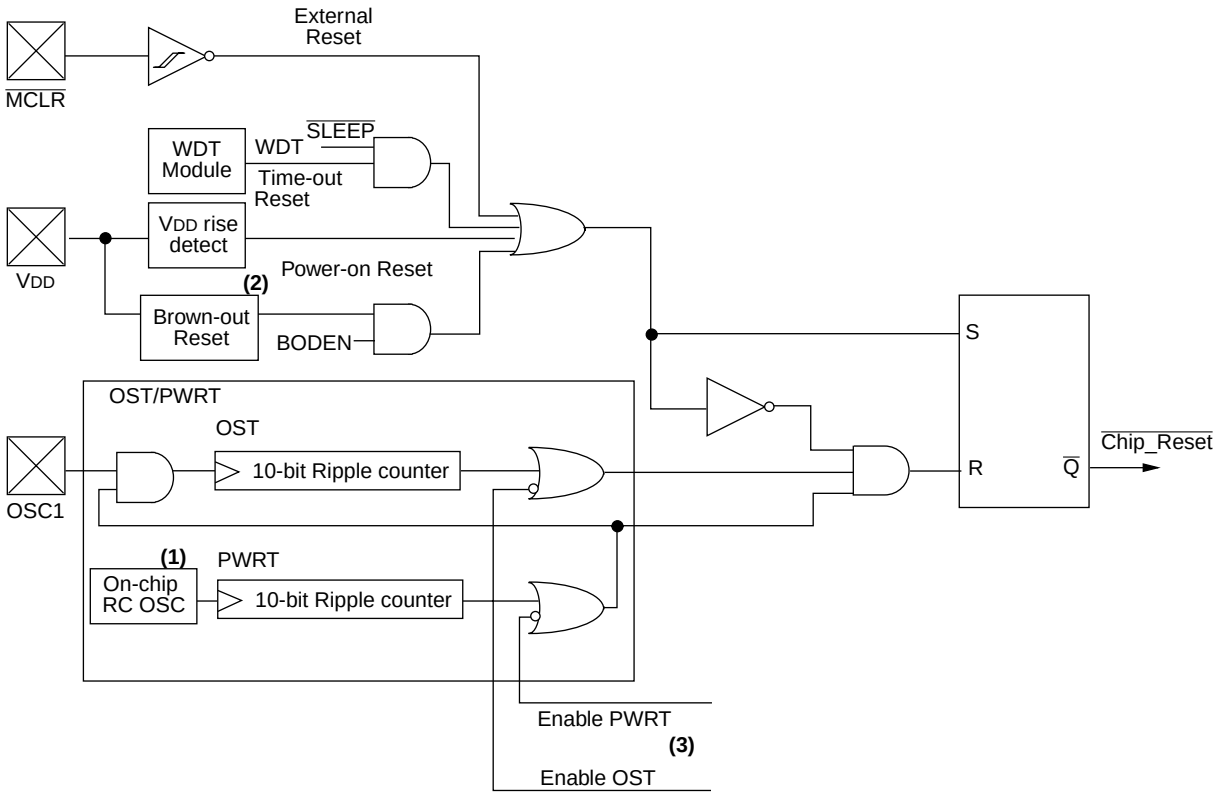
Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a "reset state" on Power-on Reset (POR), on the $\overline{MCLR}$ and WDT Reset, on $\overline{MCLR}$ reset during SLEEP, and Brown-out Reset (BOR). 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 14-5 and Table 14-6. These bits are used in software to determine the nature of the reset. See Table 14-8 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 14-8.

The PIC16C72/73A/74A/76/77 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 14-8: 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 PIC16C72/73A/74A/76/77.
3: See Table 14-3 and Table 14-4 for time-out situations.

## 14.4 Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST), and Brown-out Reset (BOR)

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

### 14.4.1 POWER-ON RESET (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.5V - 2.1V). To take advantage of the POR, just tie the $\overline{MCLR}$ pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for VDD is specified. See Electrical Specifications for details.

When the device starts normal operation (exits the reset condition), device operating parameters (voltage, frequency, temperature, ...) must be met to ensure operation. If these conditions are not met, the device must be held in reset until the operating conditions are met. Brown-out Reset may be used to meet the startup conditions.

For additional information, refer to Application Note AN607, "*Power-up Trouble Shooting*."

### 14.4.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 72 ms nominal time-out on power-up only, from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in reset as long as the PWRT is active. The PWRT's time delay allows VDD to rise to an acceptable level. A configuration bit is provided to enable/disable the PWRT.

The power-up time delay will vary from chip to chip due to VDD, temperature, and process variation. See DC parameters for details.

### 14.4.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This ensures that the crystal oscillator or resonator has started and stabilized.

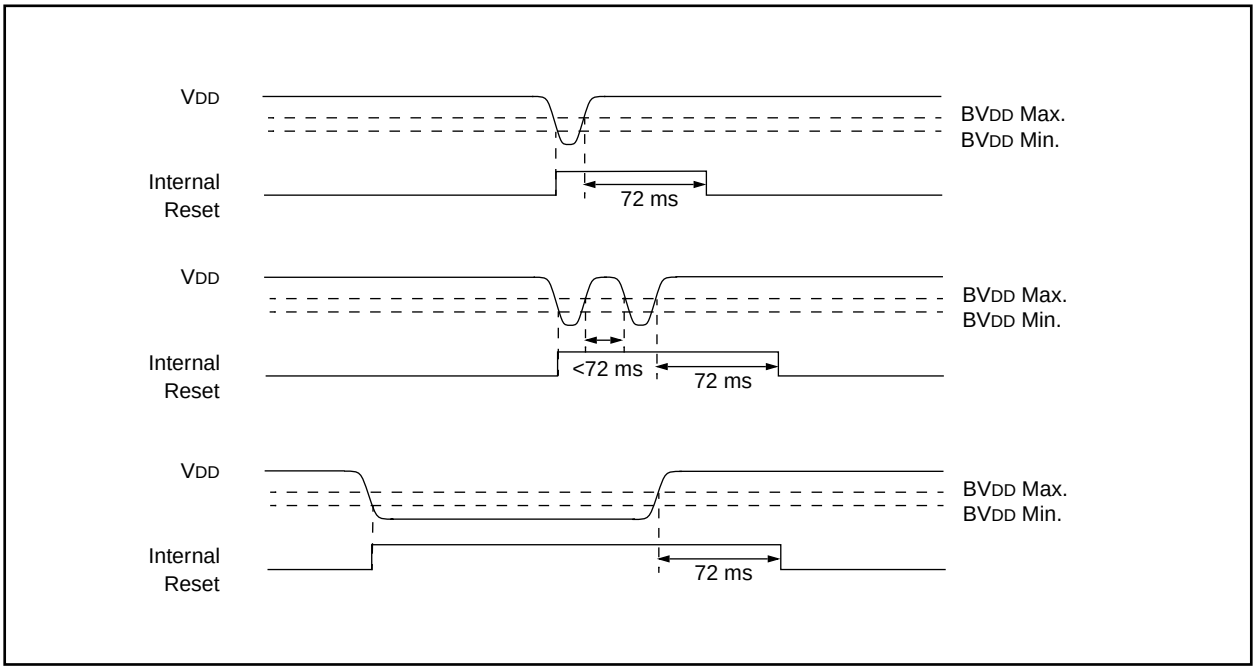
The OST time-out is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from SLEEP.

### 14.4.4 BROWN-OUT RESET (BOR)

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

A configuration bit, BODEN, can disable (if clear/programmed) or enable (if set) the Brown-out Reset circuitry. If VDD falls below 4.0V (3.8V - 4.2V range) for greater than parameter #35, the brown-out situation will reset the chip. A reset may not occur if VDD falls below 4.0V for less than parameter #35. The chip will remain in Brown-out Reset until VDD rises above BVDD. The Power-up Timer will now be invoked and will keep the chip in RESET an additional 72 ms. If VDD drops below BVDD while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once VDD rises above BVDD, the Power-up Timer will execute a 72 ms time delay. The Power-up Timer should always be enabled when Brown-out Reset is enabled. Figure 14-9 shows typical brown-out situations.

**FIGURE 14-9: BROWN-OUT SITUATIONS**



### 14.4.5 TIME-OUT SEQUENCE

On power-up the time-out sequence is as follows: First PWRT time-out is invoked after the POR time delay has expired. Then OST is activated. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. Figure 14-10, Figure 14-11, and Figure 14-12 depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, if $\overline{MCLR}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{MCLR}$ high will begin execution immediately (Figure 14-11). This is useful for testing purposes or to synchronize more than one PIC16CXX device operating in parallel.

Table 14-7 shows the reset conditions for some special function registers, while Table 14-8 shows the reset conditions for all the registers.

### 14.4.6 POWER CONTROL/STATUS REGISTER (PCON)

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

The Power Control/Status Register, PCON has up to two bits, depending upon the device. Bit0 is not implemented on the PIC16C73 or PIC16C74.

Bit0 is Brown-out Reset Status bit, $\overline{BOR}$. Bit $\overline{BOR}$ is unknown on a Power-on Reset. It must then be set by the user and checked on subsequent resets to see if bit $\overline{BOR}$ cleared, indicating a BOR occurred. The $\overline{BOR}$ bit is a "Don't Care" bit and is not necessarily predictable if the Brown-out Reset circuitry is disabled (by clearing bit BODEN in the Configuration Word).

Bit1 is $\overline{POR}$ (Power-on Reset Status bit). It is cleared on a Power-on Reset and unaffected otherwise. The user must set this bit following a Power-on Reset.

**TABLE 14-3: TIME-OUT IN VARIOUS SITUATIONS, PIC16C73/74**

| Oscillator Configuration | Power-up | | Wake-up from SLEEP |
|---|---|---|---|
| | PWRTE = 1 | PWRTE = 0 | |
| XT, HS, LP | 72 ms + 1024TOSC | 1024TOSC | 1024 TOSC |
| RC | 72 ms | — | — |

**TABLE 14-4: TIME-OUT IN VARIOUS SITUATIONS, PIC16C72/73A/74A/76/77**

| Oscillator Configuration | Power-up | | Brown-out | Wake-up from SLEEP |
|---|---|---|---|---|
| | $\overline{PWRTE}$ = 0 | $\overline{PWRTE}$ = 1 | | |
| XT, HS, LP | 72 ms + 1024TOSC | 1024TOSC | 72 ms + 1024TOSC | 1024TOSC |
| RC | 72 ms | — | 72 ms | — |

**TABLE 14-5: STATUS BITS AND THEIR SIGNIFICANCE, PIC16C73/74**

| $\overline{POR}$ | $\overline{TO}$ | $\overline{PD}$ | |
|---|---|---|---|
| 0 | 1 | 1 | Power-on Reset |
| 0 | 0 | x | Illegal, $\overline{TO}$ is set on $\overline{POR}$ |
| 0 | x | 0 | Illegal, $\overline{PD}$ is set on $\overline{POR}$ |
| 1 | 0 | 1 | WDT Reset |
| 1 | 0 | 0 | WDT Wake-up |
| 1 | u | u | $\overline{MCLR}$ Reset during normal operation |
| 1 | 1 | 0 | $\overline{MCLR}$ Reset during SLEEP or interrupt wake-up from SLEEP |

Legend: u = unchanged, x = unknown

**TABLE 14-6: STATUS BITS AND THEIR SIGNIFICANCE, PIC16C72/73A/74A/76/77**

| $\overline{POR}$ | $\overline{BOR}$ | $\overline{TO}$ | $\overline{PD}$ | |
|---|---|---|---|---|
| 0 | x | 1 | 1 | Power-on Reset |
| 0 | x | 0 | x | Illegal, $\overline{TO}$ is set on $\overline{POR}$ |
| 0 | x | x | 0 | Illegal, $\overline{PD}$ is set on $\overline{POR}$ |
| 1 | 0 | x | x | Brown-out Reset |
| 1 | 1 | 0 | 1 | WDT Reset |
| 1 | 1 | 0 | 0 | WDT Wake-up |
| 1 | 1 | u | u | $\overline{MCLR}$ Reset during normal operation |
| 1 | 1 | 1 | 0 | $\overline{MCLR}$ Reset during SLEEP or interrupt wake-up from SLEEP |

**TABLE 14-7: RESET CONDITION FOR SPECIAL REGISTERS**

| Condition | Program Counter | STATUS Register | PCON Register PIC16C73/74 | PCON Register PIC16C72/73A/74A/76/77 |
|---|---|---|---|---|
| Power-on Reset | 000h | 0001 1xxx | ---- --0- | ---- --0x |
| $\overline{MCLR}$ Reset during normal operation | 000h | 000u uuuu | ---- --u- | ---- --uu |
| $\overline{MCLR}$ Reset during SLEEP | 000h | 0001 0uuu | ---- --u- | ---- --uu |
| WDT Reset | 000h | 0000 1uuu | ---- --u- | ---- --uu |
| WDT Wake-up | PC + 1 | uuu0 0uuu | ---- --u- | ---- --uu |
| Brown-out Reset | 000h | 0001 1uuu | N/A | ---- --u0 |
| Interrupt wake-up from SLEEP | PC + 1(1) | uuu1 0uuu | ---- --u- | ---- --uu |

Legend: u = unchanged, x = unknown, - = unimplemented bit read as '0'.
Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**TABLE 14-8: INITIALIZATION CONDITIONS FOR ALL REGISTERS**

| Register | Applicable Devices | | | | | | | Power-on Reset, Brown-out Reset | $\overline{MCLR}$ Resets WDT Reset | Wake-up via WDT or Interrupt |
|---|---|---|---|---|---|---|---|---|---|---|
| W | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| INDF | 72 | 73 | 73A | 74 | 74A | 76 | 77 | N/A | N/A | N/A |
| TMR0 | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PCL | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000h | 0000h | PC + 1(2) |
| STATUS | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0001 1xxx | 000q quuu(3) | uuuq quuu(3) |
| FSR | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTA | 72 | 73 | 73A | 74 | 74A | 76 | 77 | --0x 0000 | --0u 0000 | --uu uuuu |
| PORTB | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTC | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTD | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTE | 72 | 73 | 73A | 74 | 74A | 76 | 77 | ---- -xxx | ---- -uuu | ---- -uuu |
| PCLATH | 72 | 73 | 73A | 74 | 74A | 76 | 77 | ---0 0000 | ---0 0000 | ---u uuuu |

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition
Note 1: One or more bits in INTCON, PIR1 and/or PIR2 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 14-7 for reset value for specific condition.

**TABLE 14-8: INITIALIZATION CONDITIONS FOR ALL REGISTERS (Cont.'d)**

| Register | Applicable Devices | | | | | | | Power-on Reset, Brown-out Reset | $\overline{MCLR}$ Resets WDT Reset | Wake-up via WDT or Interrupt |
|---|---|---|---|---|---|---|---|---|---|---|
| INTCON | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 000x | 0000 000u | uuuu uuuu(1) |
| PIR1 | 72 | 73 | 73A | 74 | 74A | 76 | 77 | -0-- 0000 | -0-- 0000 | -u-- uuuu(1) |
| | 72 | 73 | 73A | 74 | 74A | 76 | 77 | -000 0000 | -000 0000 | -uuu uuuu(1) |
| | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 0000 | 0000 0000 | uuuu uuuu(1) |
| PIR2 | 72 | 73 | 73A | 74 | 74A | 76 | 77 | ---- ---0 | ---- ---0 | ---- ---u(1) |
| TMR1L | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| TMR1H | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| T1CON | 72 | 73 | 73A | 74 | 74A | 76 | 77 | --00 0000 | --uu uuuu | --uu uuuu |
| TMR2 | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 0000 | 0000 0000 | uuuu uuuu |
| T2CON | 72 | 73 | 73A | 74 | 74A | 76 | 77 | -000 0000 | -000 0000 | -uuu uuuu |
| SSPBUF | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| SSPCON | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 0000 | 0000 0000 | uuuu uuuu |
| CCPR1L | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCPR1H | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCP1CON | 72 | 73 | 73A | 74 | 74A | 76 | 77 | --00 0000 | --00 0000 | --uu uuuu |
| RCSTA | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 -00x | 0000 -00x | uuuu -uuu |
| TXREG | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 0000 | 0000 0000 | uuuu uuuu |
| RCREG | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 0000 | 0000 0000 | uuuu uuuu |
| CCPR2L | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCPR2H | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCP2CON | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 0000 | 0000 0000 | uuuu uuuu |
| ADRES | 72 | 73 | 73A | 74 | 74A | 76 | 77 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADCON0 | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 00-0 | 0000 00-0 | uuuu uu-u |
| OPTION | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISA | 72 | 73 | 73A | 74 | 74A | 76 | 77 | --11 1111 | --11 1111 | --uu uuuu |
| TRISB | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISC | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISD | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISE | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 -111 | 0000 -111 | uuuu -uuu |
| PIE1 | 72 | 73 | 73A | 74 | 74A | 76 | 77 | -0-- 0000 | -0-- 0000 | -u-- uuuu |
| | 72 | 73 | 73A | 74 | 74A | 76 | 77 | -000 0000 | -000 0000 | -uuu uuuu |
| | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PIE2 | 72 | 73 | 73A | 74 | 74A | 76 | 77 | ---- ---0 | ---- ---0 | ---- ---u |
| PCON | 72 | 73 | 73A | 74 | 74A | 76 | 77 | ---- --0- | ---- --u- | ---- --u- |
| | 72 | 73 | 73A | 74 | 74A | 76 | 77 | ---- --0u | ---- --uu | ---- --uu |
| PR2 | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 1111 1111 | 1111 1111 | 1111 1111 |

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition

Note 1: One or more bits in INTCON, PIR1 and/or PIR2 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 14-7 for reset value for specific condition.

**TABLE 14-8: INITIALIZATION CONDITIONS FOR ALL REGISTERS (Cont.'d)**

| Register | Applicable Devices | | | | | | | Power-on Reset, Brown-out Reset | $\overline{MCLR}$ Resets WDT Reset | Wake-up via WDT or Interrupt |
|---|---|---|---|---|---|---|---|---|---|---|
| SSPADD | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 0000 | 0000 0000 | uuuu uuuu |
| SSPSTAT | 72 | 73 | 73A | 74 | 74A | 76 | 77 | --00 0000 | --00 0000 | --uu uuuu |
| TXSTA | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 -010 | 0000 -010 | uuuu -uuu |
| SPBRG | 72 | 73 | 73A | 74 | 74A | 76 | 77 | 0000 0000 | 0000 0000 | uuuu uuuu |
| ADCON1 | 72 | 73 | 73A | 74 | 74A | 76 | 77 | ---- -000 | ---- -000 | ---- -uuu |

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition

Note 1: One or more bits in INTCON, PIR1 and/or PIR2 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 14-7 for reset value for specific condition.

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

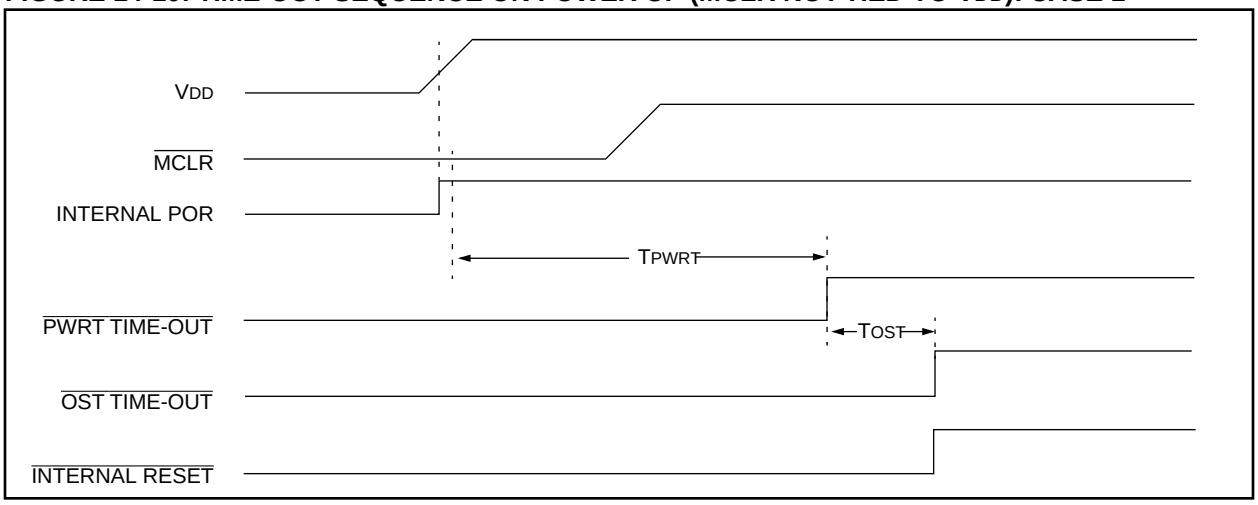



**FIGURE 14-11: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO $V_{DD}$): CASE 2**

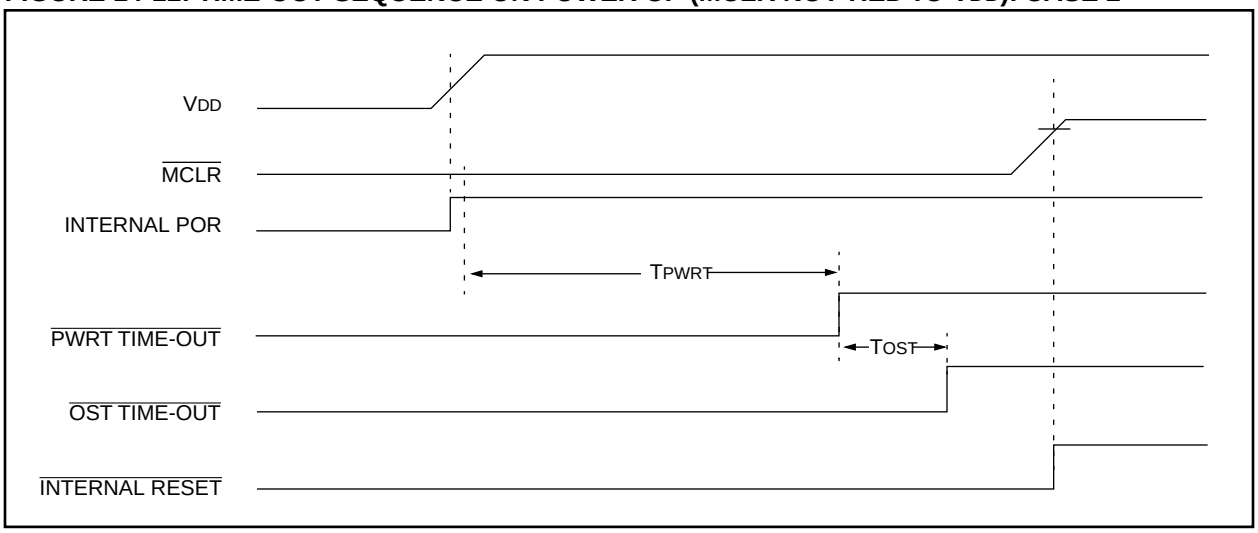



**FIGURE 14-12: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO $V_{DD}$)**

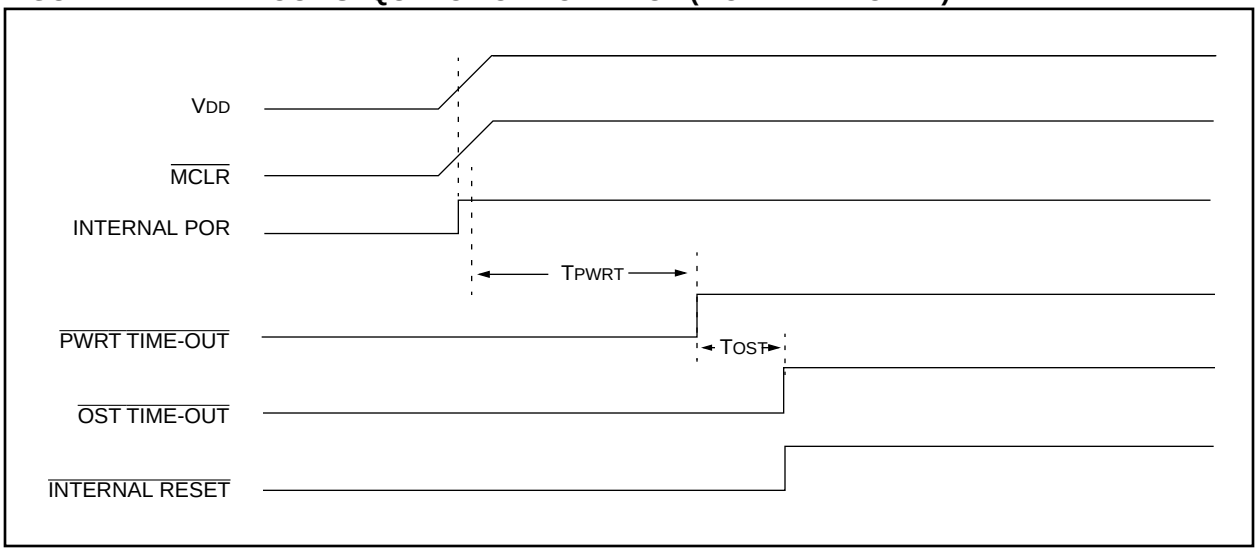
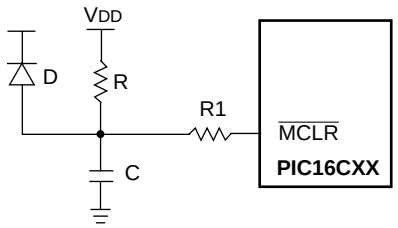


**FIGURE 14-13: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)**

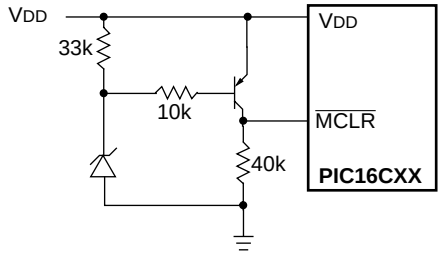



Note 1: External Power-on Reset circuit is required only if VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to make sure that voltage drop across R does not violate the device's electrical specification.

3: R1 = 100Ω to 1 kΩ will limit any current flowing into MCLR from external capacitor C in the event of MCLR/VPP pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

**FIGURE 14-14: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 1**

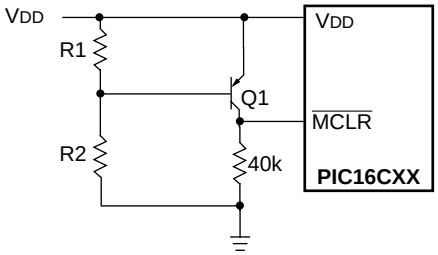



Note 1: This circuit will activate reset when VDD goes below (Vz + 0.7V) where Vz = Zener voltage.

2: Internal brown-out detection on the PIC16C72/73A/74A/76/77 should be disabled when using this circuit.

3: Resistors should be adjusted for the characteristics of the transistor.

**FIGURE 14-15: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 2**




Note 1: This brown-out circuit is less expensive, albeit less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \cdot \frac{R1}{R1 + R2} = 0.7V$$

2: Internal brown-out detection on the PIC16C72/73A/74A/76/77 should be disabled when using this circuit.

3: Resistors should be adjusted for the characteristics of the transistor.

## 14.5 Interrupts

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

The PIC16C7X family has up to 12 sources of 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 14-17). 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.

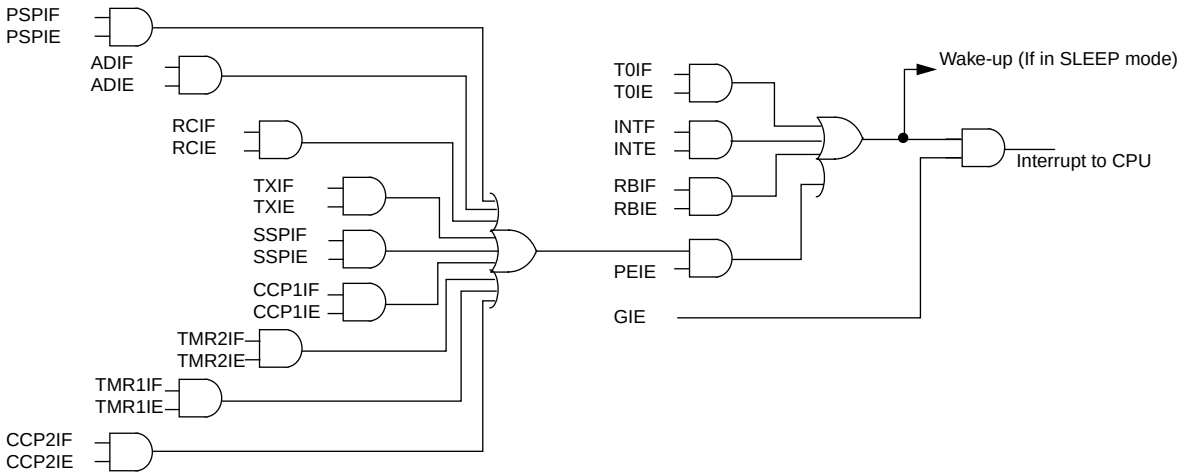
**Note:** For the PIC16C73/74, if an interrupt occurs while the Global Interrupt Enable (GIE) bit is being cleared, the GIE bit may unintentionally be re-enabled by the user's Interrupt Service Routine (the RETFIE instruction). The events that would cause this to occur are:

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
```

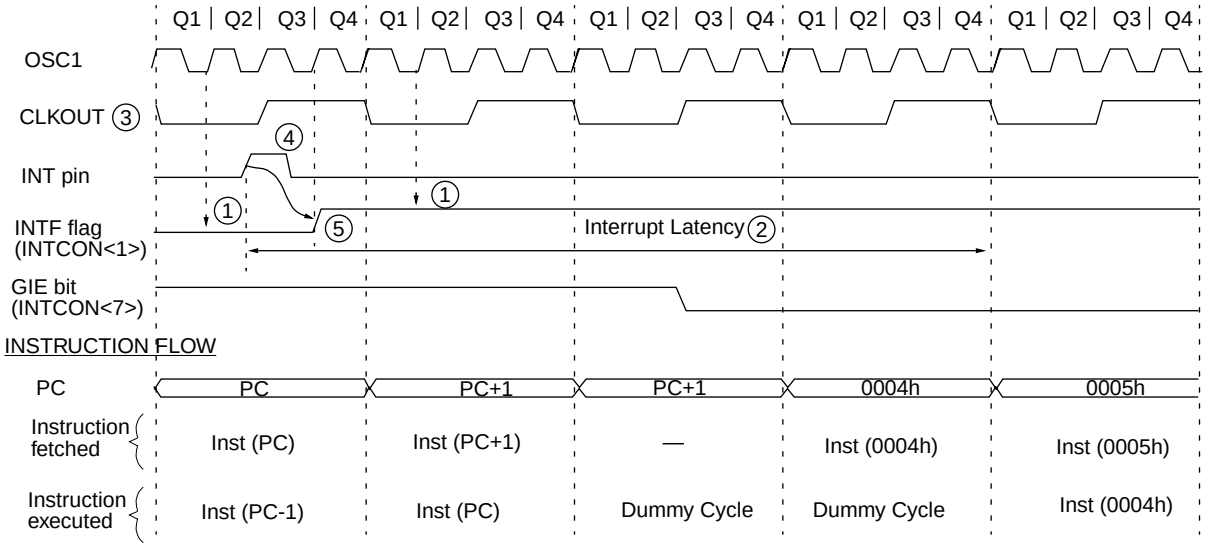
**FIGURE 14-16: INTERRUPT LOGIC**




The following table shows which devices have which interrupts.

| Device | T0IF | INTF | RBIF | PSPIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | CCP2IF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIC16C72 | Yes | Yes | Yes | - | Yes | - | - | Yes | Yes | Yes | Yes | - |
| PIC16C73 | Yes | Yes | Yes | - | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| PIC16C73A | Yes | Yes | Yes | - | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| PIC16C74 | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| PIC16C74A | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| PIC16C76 | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| PIC16C77 | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |

**FIGURE 14-17: INT PIN INTERRUPT TIMING**




Note 1: INTF flag is sampled here (every Q1).
2: Interrupt latency = 3-4 Tcy where Tcy = instruction cycle time.
Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
3: CLKOUT is available only in RC oscillator mode.
4: For minimum width of INT pulse, refer to AC specs.
5: INTF is enabled to be set anytime during the Q4-Q1 cycles.

#### 14.5.1 INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered: either rising if bit INTEDG (OPTION<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit INTE (INTCON<4>). Flag bit INTF must be cleared in software in the interrupt service routine before re-enabling this interrupt. The INT interrupt can wake-up the processor from SLEEP, if bit INTE was set prior to going into SLEEP. The status of global interrupt enable bit GIE decides whether or not the processor branches to the interrupt vector following wake-up. See Section 14.8 for details on SLEEP mode.

#### 14.5.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set flag bit T0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit T0IE (INTCON<5>). (Section 7.0)

#### 14.5.3 PORTB INTCON CHANGE

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<4>). (Section 5.2)

| **Note:** | For the PIC16C73/74, if a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not get set. |
|---|---|

### 14.6 Context Saving During Interrupts

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

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt i.e., W register and STATUS register. This will have to be implemented in software.

Example 14-1 stores and restores the STATUS, W, and PCLATH registers. The register, W_TEMP, must be defined in each bank and must be defined at the same offset from the bank base address (i.e., if W_TEMP is defined at 0x20 in bank 0, it must also be defined at 0xA0 in bank 1).

The example:

a) Stores the W register.
b) Stores the STATUS register in bank 0.
c) Stores the PCLATH register.
d) Executes the ISR code.
e) Restores the STATUS register (and bank select bit).
f) Restores the W and PCLATH registers.

**EXAMPLE 14-1: SAVING STATUS, W, AND PCLATH REGISTERS IN RAM**

```
MOVWF    W_TEMP           ;Copy W to TEMP register, could be bank one or zero
SWAPF    STATUS,W         ;Swap status to be saved into W
CLRF     STATUS           ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF    STATUS_TEMP      ;Save status to bank zero STATUS_TEMP register
MOVF     PCLATH, W        ;Only required if using pages 1, 2 and/or 3
MOVWF    PCLATH_TEMP      ;Save PCLATH into W
CLRF     PCLATH           ;Page zero, regardless of current page
BCF      STATUS, IRP      ;Return to Bank 0
MOVF     FSR, W           ;Copy FSR to W
MOVWF    FSR_TEMP         ;Copy FSR from W to FSR_TEMP
:
:(ISR)
:
MOVF     PCLATH_TEMP, W   ;Restore PCLATH
MOVWF    PCLATH           ;Move W into PCLATH
SWAPF    STATUS_TEMP,W    ;Swap STATUS_TEMP register into W
                          ;(sets bank to original state)
MOVWF    STATUS           ;Move W into STATUS register
SWAPF    W_TEMP,F         ;Swap W_TEMP
SWAPF    W_TEMP,W         ;Swap W_TEMP into W
```

## 14.7 Watchdog Timer (WDT)

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

The Watchdog Timer is as a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run, even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins of the device has been stopped, for example, by execution of a SLEEP instruction. During normal operation, a WDT time-out generates a device RESET (Watchdog Timer Reset). If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer Wake-up). The WDT can be permanently disabled by clearing configuration bit WDTE (Section 14.1).

### 14.7.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET condition.

The $\overline{TO}$ bit in the STATUS register will be cleared upon a Watchdog Timer time-out.

### 14.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken into account that under worst case conditions (VDD = Min., Temperature = Max., and max. WDT prescaler) it may take several seconds before a WDT time-out occurs.

| **Note:** | When a CLRWDT instruction is executed and the prescaler is assigned to the WDT, the prescaler count will be cleared, but the prescaler assignment is not changed. |
|---|---|

**FIGURE 14-18: WATCHDOG TIMER BLOCK DIAGRAM**

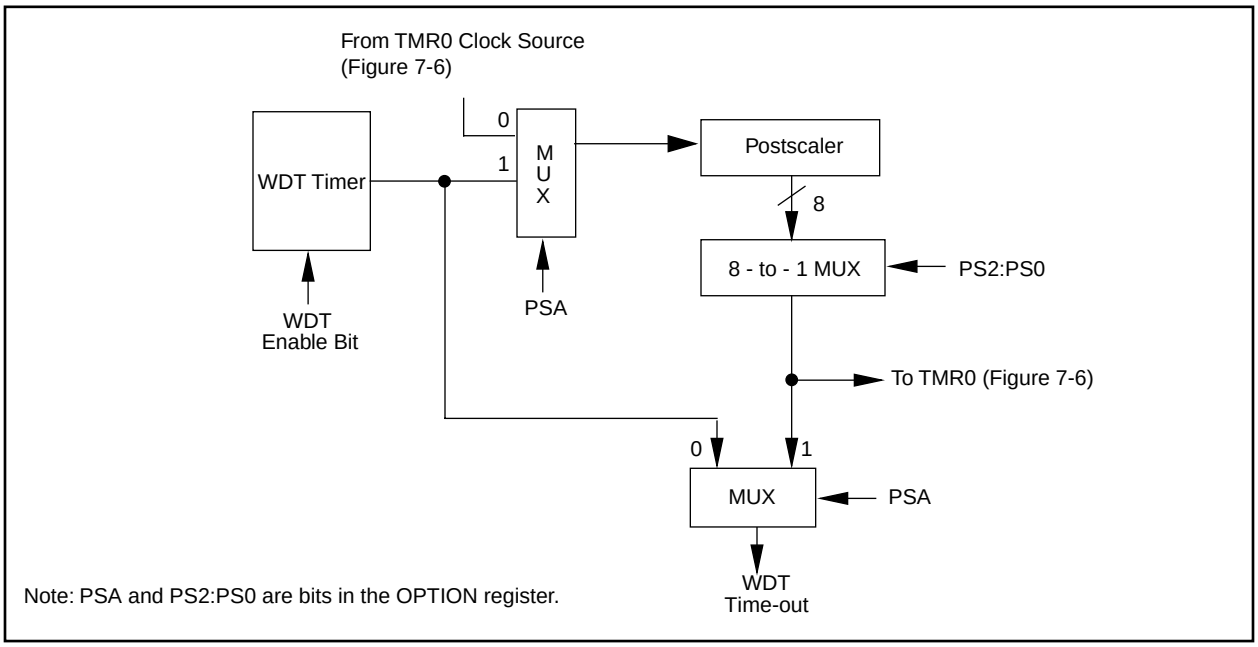



**FIGURE 14-19: SUMMARY OF WATCHDOG TIMER REGISTERS**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 2007h | Config. bits | **(1)** | BODEN(1) | CP1 | CP0 | PWRTE(1) | WDTE | FOSC1 | FOSC0 |
| 81h,181h | OPTION | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |

Legend: Shaded cells are not used by the Watchdog Timer.
Note 1: See Figure 14-1, and Figure 14-2 for operation of these bits.

## 14.8 Power-down Mode (SLEEP)

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

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

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{PD}$ bit (STATUS<3>) is cleared, the $\overline{TO}$ (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 $\overline{MCLR}$ pin must be at a logic high level ($V_{IHMC}$).

### 14.8.1 WAKE-UP FROM SLEEP

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

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

External $\overline{MCLR}$ Reset will cause a device reset. All other events are considered a continuation of program execution and cause a "wake-up". The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register can be used to determine the cause of device reset. The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The $\overline{TO}$ 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. SSP (Start/Stop) bit detect interrupt.
3. SSP transmit or receive in slave mode (SPI/$I^2C$).
4. CCP capture mode interrupt.
5. Parallel Slave Port read or write.
6. A/D conversion (when A/D clock source is RC).
7. Special event trigger (Timer1 in asynchronous mode using an external clock).
8. USART TX or RX (synchronous slave mode).

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.

### 14.8.2 WAKE-UP USING INTERRUPTS

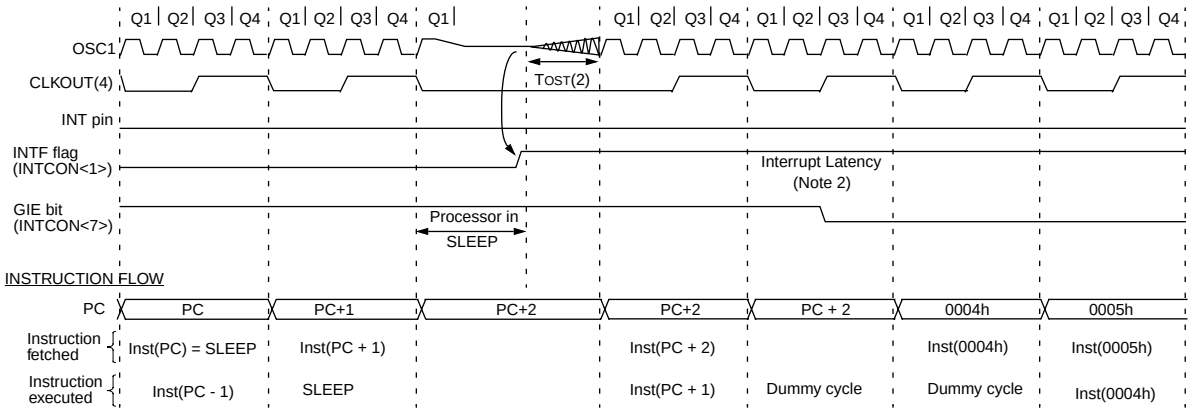
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT postscaler will not be cleared, the $\overline{TO}$ bit will not be set and $\overline{PD}$ bits will not be cleared.
- If the interrupt occurs **during or after** the execution of a SLEEP instruction, the device will immediately wake up from sleep. The SLEEP instruction will be completely executed before the wake-up. Therefore, the WDT and WDT postscaler will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

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

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

**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

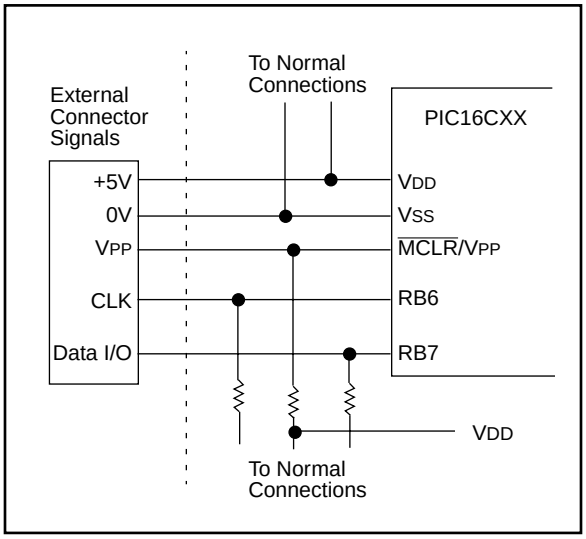
| Applicable Devices | | | | | | |
|---|---|---|---|---|---|---|
| 72 | 73 | 73A | 74 | 74A | 76 | 77 |

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**



## 15.0 INSTRUCTION SET SUMMARY

Each PIC16CXX instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CXX instruction set summary in Table 15-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 15-1 shows the opcode field descriptions.

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

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

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

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

**TABLE 15-1: OPCODE FIELD DESCRIPTIONS**

| Field | Description |
|---|---|
| f | Register file address (0x00 to 0x7F) |
| W | Working register (accumulator) |
| b | Bit address within an 8-bit file register |
| k | Literal field, constant data or label |
| x | Don't care location (= 0 or 1)<br>The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools. |
| d | Destination select; d = 0: store result in W,<br>d = 1: store result in file register f.<br>Default is d = 1 |
| label | Label name |
| TOS | Top of Stack |
| PC | Program Counter |
| PCLATH | Program Counter High Latch |
| GIE | Global Interrupt Enable bit |
| WDT | Watchdog Timer/Counter |
| $\overline{TO}$ | Time-out bit |
| $\overline{PD}$ | Power-down bit |
| dest | Destination either the W register or the specified register file location |
| [ ] | Options |
| ( ) | Contents |
| $\rightarrow$ | Assigned to |
| < > | Register bit field |
| $\in$ | In the set of |
| *italics* | User defined term (font is courier) |

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 µs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 µs.

Table 15-2 lists the instructions recognized by the MPASM assembler.

Figure 15-1 shows the general formats that the instructions can have.

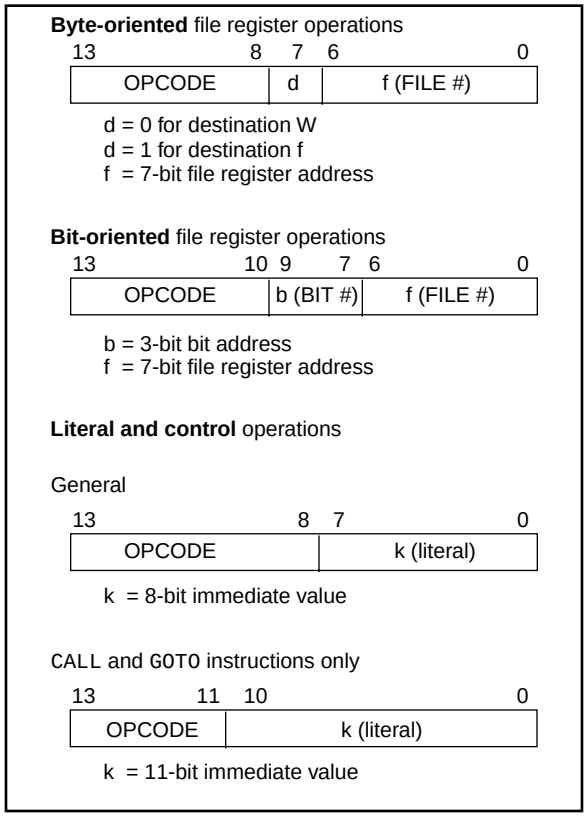
**Note:** To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

**FIGURE 15-1: GENERAL FORMAT FOR INSTRUCTIONS**



### TABLE 15-2: 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.

## 15.1 Instruction Descriptions

**ADDLW** **Add Literal and W**

Syntax: [*label*] ADDLW k

Operands: $0 \le k \le 255$

Operation: (W) + k → (W)

Status Affected: C, DC, Z

Encoding:

| | | | |
|---|---|---|---|
| 11 | 111x | kkkk | kkkk |

Description: The contents of the W register are added to the eight bit literal 'k' and 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:
```
ADDLW   0x15
```
Before Instruction
W = 0x10
After Instruction
W = 0x25

**ANDLW** **AND Literal with W**

Syntax: [*label*] ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. (k) → (W)

Status Affected: Z

Encoding:

| | | | |
|---|---|---|---|
| 11 | 1001 | kkkk | kkkk |

Description: The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example
```
ANDLW   0x5F
```
Before Instruction
W = 0xA3
After Instruction
W = 0x03

**ADDWF** **Add W and f**

Syntax: [*label*] ADDWF f,d

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

Operation: (W) + (f) → (destination)

Status Affected: C, DC, Z

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0111 | dfff | ffff |

Description: Add the contents of the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example
```
ADDWF   FSR, 0
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0xD9
FSR = 0xC2

**ANDWF** **AND W with f**

Syntax: [*label*] ANDWF f,d

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

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

Status Affected: Z

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0101 | dfff | ffff |

Description: AND the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example
```
ANDWF   FSR, 1
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0x17
FSR = 0x02

**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 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 TRUE
if FLAG<1>=1,
PC = address FALSE

**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

**CLRF** **Clear f**

Syntax: [*label*] CLRF f

Operands: $0 \le f \le 127$

Operation: 00h → (f)
1 → Z

Status Affected: Z

Encoding:

| 00 | 0001 | 1fff | ffff |
|---|---|---|---|

Description: The contents of register 'f' are cleared and the Z bit is set.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example
```
CLRF    FLAG_REG
```
Before Instruction
FLAG_REG = 0x5A
After Instruction
FLAG_REG = 0x00
Z = 1

**CLRW** **Clear W**

Syntax: [ *label* ] CLRW

Operands: None

Operation: 00h → (W)
1 → Z

Status Affected: Z

Encoding:

| 00 | 0001 | 0xxx | xxxx |
|---|---|---|---|

Description: W register is cleared. Zero bit (Z) is set.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | No-Operation | Process data | Write to W |

Example
```
CLRW
```
Before Instruction
W = 0x5A
After Instruction
W = 0x00
Z = 1

**CLRWDT** **Clear Watchdog Timer**

Syntax: [ *label* ] CLRWDT

Operands: None

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

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

Encoding:

| 00 | 0000 | 0110 | 0100 |
|---|---|---|---|

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

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | No-Operation | Process data | Clear WDT Counter |

Example
```
CLRWDT
```
Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescaler = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

**COMF** **Complement f**

Syntax: [ *label* ] COMF f,d

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

Operation: $(\bar{f}) \rightarrow$ (destination)

Status Affected: Z

Encoding:

| 00 | 1001 | dfff | ffff |
|---|---|---|---|

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

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example
```
COMF    REG1,0
```
Before Instruction
REG1 = 0x13
After Instruction
REG1 = 0x13
W = 0xEC

**DECF** **Decrement f**

Syntax: [*label*] DECF f,d

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

Operation: (f) - 1 → (destination)

Status Affected: Z

Encoding:

| 00 | 0011 | dfff | ffff |
|---|---|---|---|

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

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example
```
DECF    CNT, 1
```
Before Instruction
CNT = 0x01
Z = 0
After Instruction
CNT = 0x00
Z = 1

**DECFSZ** **Decrement f, Skip if 0**

Syntax: [ *label* ] DECFSZ f,d

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

Operation: (f) - 1 → (destination); skip if result = 0

Status Affected: None

Encoding:

| 00 | 1011 | dfff | ffff |
|---|---|---|---|

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

Words: 1

Cycles: 1(2)

Q Cycle Activity:

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

If Skip: (2nd Cycle)

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

Example
```
HERE    DECFSZ  CNT, 1
        GOTO    LOOP
CONTINUE •
         •
         •
```
Before Instruction
PC = address HERE
After Instruction
CNT = CNT - 1
if CNT = 0,
PC = address CONTINUE
if CNT ≠ 0,
PC = address HERE+1

**GOTO** **Unconditional Branch**

Syntax: [ *label* ] GOTO k

Operands: $0 \le k \le 2047$

Operation: $k \rightarrow PC<10:0>$
$PCLATH<4:3> \rightarrow PC<12:11>$

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 | No-Operation | No-Operation | No-Operation | No-Operation |

Example

```
GOTO THERE
```

After Instruction
PC = Address THERE

**INCF** **Increment f**

Syntax: [ *label* ] INCF f,d

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

Operation: (f) + 1 → (destination)

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 destination |

Example

```
INCF    CNT, 1
```

Before Instruction
CNT = 0xFF
Z = 0
After Instruction
CNT = 0x00
Z = 1

**INCFSZ** **Increment f, Skip if 0**

Syntax: [ *label* ] INCFSZ f,d

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

Operation: (f) + 1 → (destination), skip if result = 0

Status Affected: None

Encoding:

| 00 | 1111 | dfff | ffff |
|---|---|---|---|

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

Words: 1

Cycles: 1(2)

Q Cycle Activity:

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

If Skip: (2nd Cycle)

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

Example

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

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

**IORLW** **Inclusive OR Literal with W**

Syntax: [ *label* ] IORLW k

Operands: $0 \le k \le 255$

Operation: (W) .OR. k → (W)

Status Affected: Z

Encoding:

| 11 | 1000 | kkkk | kkkk |
|---|---|---|---|

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

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example

```
IORLW   0x35
```

Before Instruction
W = 0x9A
After Instruction
W = 0xBF
Z = 1

**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

**NOP** **No Operation**

Syntax: [ *label* ] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0000 | 0xx0 | 0000 |

Description: No operation.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example

```
NOP
```

**RETFIE** **Return from Interrupt**

Syntax: [ *label* ] RETFIE

Operands: None

Operation: TOS → PC,
1 → GIE

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0000 | 0000 | 1001 |

Description: Return from Interrupt. Stack is POPed and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

| | Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|---|
| 1st Cycle | Decode | No-Operation | Set the GIE bit | Pop from the Stack |
| 2nd Cycle | No-Operation | No-Operation | No-Operation | No-Operation |

Example

```
RETFIE
```

After Interrupt
PC = TOS
GIE = 1

**OPTION** **Load Option Register**

Syntax: [ *label* ] OPTION

Operands: None

Operation: (W) → OPTION

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0000 | 0110 | 0010 |

Description: The contents of the W register are loaded in the OPTION register. This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.

Words: 1

Cycles: 1

Example

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

**RETLW** **Return with Literal in W**

Syntax: [ *label* ] RETLW k

Operands: $0 \le k \le 255$

Operation: k → (W);
TOS → 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' | No-Operation | Write to W, Pop from the Stack |
| 2nd Cycle | No-Operation | No-Operation | No-Operation | No-Operation |

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 → 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 | No-Operation | No-Operation | Pop from the Stack |
| 2nd Cycle | No-Operation | No-Operation | No-Operation | No-Operation |

Example

```
RETURN
```

After Interrupt
PC = TOS

**RLF** **Rotate Left f through Carry**

Syntax: [ *label* ] RLF f,d

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

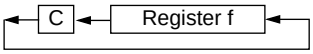
Operation: See description below

Status Affected: C

Encoding:

| 00 | 1101 | dfff | ffff |
|---|---|---|---|

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




Words: 1

Cycles: 1

Q Cycle Activity:

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

Example

```
RLF     REG1,0
```

Before Instruction
```
REG1  =  1110 0110
C     =  0
```
After Instruction
```
REG1  =  1110 0110
W     =  1100 1100
C     =  1
```

**RRF** **Rotate Right f through Carry**

Syntax: [ *label* ] RRF f,d

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

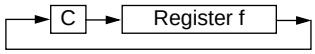
Operation: See description below

Status Affected: C

Encoding:

| 00 | 1100 | dfff | ffff |
|---|---|---|---|

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




Words: 1

Cycles: 1

Q Cycle Activity:

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

Example

```
RRF     REG1,0
```

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

**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 14.8 for more details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | No-Operation | No-Operation | Go to Sleep |

Example: SLEEP

**SUBLW Subtract W from Literal**

Syntax: [ *label* ] SUBLW k

Operands: 0 ≤ k ≤ 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

**SUBWF** **Subtract W from f**

Syntax: [ *label* ] SUBWF f,d

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

Operation: (f) - (W) → (destination)

Status Affected: C, DC, Z

Encoding:

| 00 | 0010 | dfff | ffff |
|---|---|---|---|

Description: Subtract (2's complement method) W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example 1: `SUBWF REG1,1`

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

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

Example 2: Before Instruction
REG1 = 2
W = 2
C = ?
Z = ?

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

Example 3: Before Instruction
REG1 = 1
W = 2
C = ?
Z = ?

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

**SWAPF** **Swap Nibbles in f**

Syntax: [ *label* ] SWAPF f,d

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

Operation: (f<3:0>) → (destination<7:4>),
(f<7:4>) → (destination<3:0>)

Status Affected: None

Encoding:

| 00 | 1110 | dfff | ffff |
|---|---|---|---|

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example
```
SWAPF REG, 0
```
Before Instruction
REG1 = 0xA5

After Instruction
REG1 = 0xA5
W = 0x5A

**TRIS** **Load TRIS Register**

Syntax: [*label*] TRIS f

Operands: $5 \le f \le 7$

Operation: (W) → TRIS register f;

Status Affected: None

Encoding:

| 00 | 0000 | 0110 | 0fff |
|---|---|---|---|

Description: The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.

Words: 1

Cycles: 1

Example

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

**XORLW** **Exclusive OR Literal with W**

Syntax: [*label*] XORLW k

Operands: $0 \le k \le 255$

Operation: (W) .XOR. k → (W)

Status Affected: Z

Encoding:

| | | | |
|---|---|---|---|
| 11 | 1010 | kkkk | kkkk |

Description: The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example:

```
XORLW   0xAF
```

Before Instruction

W = 0xB5

After Instruction

W = 0x1A

**XORWF** **Exclusive OR W with f**

Syntax: [*label*] XORWF f,d

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

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

Status Affected: Z

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0110 | dfff | ffff |

Description: Exclusive OR the contents of the W register with register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example

```
XORWF   REG  1
```

Before Instruction

REG = 0xAF
W = 0xB5

After Instruction

REG = 0x1A
W = 0xB5

# 16.0 DEVELOPMENT SUPPORT

## 16.1 Development Tools

The PIC16/17 microcontrollers are supported with a full range of hardware and software development tools:

- PICMASTER/PICMASTER CE Real-Time In-Circuit Emulator
- ICEPIC Low-Cost PIC16C5X and PIC16CXXX In-Circuit Emulator
- PRO MATE® II Universal Programmer
- PICSTART® Plus Entry-Level Prototype Programmer
- PICDEM-1 Low-Cost Demonstration Board
- PICDEM-2 Low-Cost Demonstration Board
- PICDEM-3 Low-Cost Demonstration Board
- MPASM Assembler
- MPLAB-SIM Software Simulator
- MPLAB-C (C Compiler)
- Fuzzy logic development system (*fuzzy*TECH®–MP)

## 16.2 PICMASTER: High Performance Universal In-Circuit Emulator with MPLAB IDE

The PICMASTER Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for all microcontrollers in the PIC12C5XX, PIC14C000, PIC16C5X, PIC16CXXX and PIC17CXX families. PICMASTER is supplied with the MPLAB™ Integrated Development Environment (IDE), which allows editing, “make” and download, and source debugging from a single environment.

Interchangeable target probes allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the PICMASTER allows expansion to support all new Microchip microcontrollers.

The PICMASTER Emulator System has been designed as a real-time emulation system with advanced features that are generally found on more expensive development tools. The PC compatible 386 (and higher) machine platform and Microsoft Windows® 3.x environment were chosen to best make these features available to you, the end user.

A CE compliant version of PICMASTER is available for European Union (EU) countries.

## 16.3 ICEPIC: Low-cost PIC16CXXX In-Circuit Emulator

ICEPIC is a low-cost in-circuit emulator solution for the Microchip PIC16C5X and PIC16CXXX families of 8-bit OTP microcontrollers.

ICEPIC is designed to operate on PC-compatible machines ranging from 286-AT® through Pentium™ based machines under Windows 3.x environment. ICEPIC features real time, non-intrusive emulation.

## 16.4 PRO MATE II: Universal Programmer

The PRO MATE II Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode.

The PRO MATE II has programmable $V_{DD}$ and $V_{PP}$ supplies which allows it to verify programmed memory at $V_{DD}$ min and $V_{DD}$ max for maximum reliability. It has an LCD display for displaying error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE II can read, verify or program PIC16C5X, PIC16CXXX, PIC17CXX and PIC14000 devices. It can also set configuration and code-protect bits in this mode.

## 16.5 PICSTART Plus Entry Level Development System

The PICSTART programmer is an easy-to-use, low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. PICSTART Plus is not recommended for production programming.

PICSTART Plus supports all PIC12C5XX, PIC14000, PIC16C5X, PIC16CXXX and PIC17CXX devices with up to 40 pins. Larger pin count devices such as the PIC16C923 and PIC16C924 may be supported with an adapter socket.

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

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

### 16.7 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

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

### 16.8 PICDEM-3 Low-Cost PIC16CXXX Demonstration Board

The PICDEM-3 is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with a LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-3 board, on a PRO MATE II programmer or PICSTART Plus with an adapter socket, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-3 board to test firmware. Additional prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include an RS-232 interface, push-button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM-3 board is an LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM-3 provides an additional RS-232 interface and Windows 3.1 software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals.

### 16.9 MPLAB Integrated Development Environment Software

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

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

MPLAB allows you to:

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

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

### 16.10 Assembler (MPASM)

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

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

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

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.

### 16.11 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.

### 16.12 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 (PICMASTER emulator software versions 1.13 and later).

### 16.13 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.

### 16.14 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.

### 16.15 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.

### 16.16 TrueGauge® Intelligent Battery Management

The TrueGauge development tool supports system development with the MTA11200B TrueGauge Intelligent Battery Management IC. System design verification can be accomplished before hardware prototypes are built. User interface is graphically-oriented and measured data can be saved in a file for exporting to Microsoft Excel.

### 16.17 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.

**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 |
|---|---|---|---|---|---|---|---|

## 17.0 ELECTRICAL CHARACTERISTICS FOR PIC16C72

**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) ..... 1.0W

Maximum current out of $V_{SS}$ pin ..... 300 mA

Maximum current into $V_{DD}$ pin ..... 250 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 ..... 25 mA

Maximum current sunk by PORTA and PORTB (combined) ..... 200 mA

Maximum current sourced by PORTA and PORTB (combined) ..... 200 mA

Maximum current sunk by PORTC ..... 200 mA

Maximum current sourced by PORTC ..... 200 mA

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

**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 17-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)**

| OSC | PIC16C72-04 | PIC16C72-10 | PIC16C72-20 | PIC16LC72-04 | JW Devices |
|---|---|---|---|---|---|
| RC | $V_{DD}$: 4.0V to 6.0V<br>$I_{DD}$: 5 mA max. at 5.5V<br>$I_{PD}$: 16 µA max. at 4V<br>Freq: 4 MHz max. | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 2.7 mA typ. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 2.7 mA typ. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | $V_{DD}$: 2.5V to 6.0V<br>$I_{DD}$: 3.8 mA max. at 3.0V<br>$I_{PD}$: 5.0 µA max. at 3V<br>Freq: 4 MHz max. | $V_{DD}$: 4.0V to 6.0V<br>$I_{DD}$: 5 mA max. at 5.5V<br>$I_{PD}$: 16 µA max. at 4V<br>Freq: 4 MHz max. |
| XT | $V_{DD}$: 4.0V to 6.0V<br>$I_{DD}$: 5 mA max. at 5.5V<br>$I_{PD}$: 16 µA max. at 4V<br>Freq: 4 MHz max. | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 2.7 mA typ. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 2.7 mA typ. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | $V_{DD}$: 2.5V to 6.0V<br>$I_{DD}$: 3.8 mA max. at 3.0V<br>$I_{PD}$: 5.0 µA max. at 3V<br>Freq: 4 MHz max. | $V_{DD}$: 4.0V to 6.0V<br>$I_{DD}$: 5 mA max. at 5.5V<br>$I_{PD}$: 16 µA max. at 4V<br>Freq: 4 MHz max. |
| HS | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 13.5 mA typ. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4.5V<br>Freq: 4 MHz max. | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 10 mA max. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4.5V<br>Freq: 10 MHz max. | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 20 mA max. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4.5V<br>Freq: 20 MHz max. | Not recommended for use in HS mode | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 20 mA max. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4.5V<br>Freq: 20 MHz max. |
| LP | $V_{DD}$: 4.0V to 6.0V<br>$I_{DD}$: 52.5 µA typ. at 32 kHz, 4.0V<br>$I_{PD}$: 0.9 µA typ. at 4.0V<br>Freq: 200 kHz max. | Not recommended for use in LP mode | Not recommended for use in LP mode | $V_{DD}$: 2.5V to 6.0V<br>$I_{DD}$: 48 µA max. at 32 kHz, 3.0V<br>$I_{PD}$: 5.0 µA max. at 3.0V<br>Freq: 200 kHz max. | $V_{DD}$: 2.5V to 6.0V<br>$I_{DD}$: 48 µA max. at 32 kHz, 3.0V<br>$I_{PD}$: 5.0 µA max. at 3.0V<br>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 | 72 | 73 | 73A | 74 | 74A | 76 | 77 |
|---|---|---|---|---|---|---|---|

### 17.1 DC Characteristics: PIC16C72-04 (Commercial, Industrial, Extended) PIC16C72-10 (Commercial, Industrial, Extended) PIC16C72-20 (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 | Supply Voltage | VDD | 4.0 | - | 6.0 | V | XT, RC and LP osc configuration |
| D001A | | | 4.5 | - | 5.5 | V | 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 | 4.0 | 4.3 | V | BODEN bit in configuration word enabled |
| | | | 3.7 | 4.0 | 4.4 | V | Extended Only |
| D010 | Supply Current (Note 2,5) | IDD | - | 2.7 | 5.0 | mA | XT, RC osc configuration FOSC = 4 MHz, VDD = 5.5V (Note 4) |
| D013 | | | - | 10 | 20 | mA | HS osc configuration FOSC = 20 MHz, VDD = 5.5V |
| D015 | Brown-out Reset Current (Note 6) | ΔIBOR | - | 350 | 425 | µA | BOR enabled VDD = 5.0V |
| D020 | Power-down Current (Note 3,5) | IPD | - | 10.5 | 42 | µA | VDD = 4.0V, WDT enabled, -40°C to +85°C |
| D021 | | | - | 1.5 | 16 | µA | VDD = 4.0V, WDT disabled, -0°C to +70°C |
| D021A | | | - | 1.5 | 19 | µA | VDD = 4.0V, WDT disabled, -40°C to +85°C |
| D021B | | | - | 2.5 | 19 | µA | VDD = 4.0V, WDT disabled, -40°C to +125°C |
| D023 | Brown-out Reset Current (Note 6) | ΔIBOR | - | 350 | 425 | 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 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 |
|---|---|---|---|---|---|---|---|

## 17.2 DC Characteristics: PIC16LC72-04 (Commercial, Industrial)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -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 | Supply Voltage | VDD | 2.5 | - | 6.0 | V | LP, XT, RC osc configuration (DC - 4 MHz) |
| 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 | 4.0 | 4.3 | V | BODEN bit in configuration word enabled |
| D010 | Supply Current (Note 2,5) | IDD | - | 2.0 | 3.8 | mA | XT, RC osc configuration<br>FOSC = 4 MHz, VDD = 3.0V (Note 4) |
| D010A | | | - | 22.5 | 48 | µA | LP osc configuration<br>FOSC = 32 kHz, VDD = 3.0V, WDT disabled |
| D015* | Brown-out Reset Current (Note 6) | ΔIBOR | - | 350 | 425 | µA | BOR enabled VDD = 5.0V |
| D020 | Power-down Current (Note 3,5) | IPD | - | 7.5 | 30 | µA | VDD = 3.0V, WDT enabled, -40°C to +85°C |
| D021 | | | - | 0.9 | 5 | µA | VDD = 3.0V, WDT disabled, 0°C to +70°C |
| D021A | | | - | 0.9 | 5 | µA | VDD = 3.0V, WDT disabled, -40°C to +85°C |
| D023* | Brown-out Reset Current (Note 6) | ΔIBOR | - | 350 | 425 | 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 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 |
|---|---|---|---|---|---|---|---|

### 17.3 DC Characteristics: PIC16C72-04 (Commercial, Industrial, Extended)
### PIC16C72-10 (Commercial, Industrial, Extended)
### PIC16C72-20 (Commercial, Industrial, Extended)
### PIC16LC72-04 (Commercial, Industrial)

| 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<br>Operating voltage VDD range as described in DC spec Section 17.1 and Section 17.2. | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Characteristic** | **Sym** | **Min** | **Typ †** | **Max** | **Units** | **Conditions** |
| | **Input Low Voltage** | | | | | | |
| | I/O ports | VIL | | | | | |
| D030 | with TTL buffer | | VSS | - | 0.15VDD | V | For entire VDD range |
| D030A | | | VSS | - | 0.8V | V | 4.5 ≤ VDD ≤ 5.5V |
| D031 | with Schmitt Trigger buffer | | VSS | - | 0.2VDD | V | |
| D032 | $\overline{MCLR}$, OSC1 (in RC mode) | | VSS | - | 0.2VDD | V | |
| D033 | OSC1 (in XT, HS and LP) | | VSS | - | 0.3VDD | V | Note1 |
| | **Input High Voltage** | | | | | | |
| | I/O ports | VIH | | - | | | |
| D040 | with TTL buffer | | 2.0 | - | VDD | V | 4.5 ≤ VDD ≤ 5.5V |
| D040A | | | 0.25VDD + 0.8V | - | VDD | V | For entire VDD range |
| D041 | with Schmitt Trigger buffer | | 0.8VDD | - | VDD | V | For entire VDD range |
| D042 | $\overline{MCLR}$ | | 0.8VDD | - | VDD | V | |
| D042A | OSC1 (XT, HS and LP) | | 0.7VDD | - | VDD | V | Note1 |
| D043 | OSC1 (in RC mode) | | 0.9VDD | - | VDD | V | |
| D070 | PORTB weak pull-up current | IPURB | 50 | 250 | †400 | µA | VDD = 5V, VPIN = VSS |
| | **Input Leakage Current** (Notes 2, 3) | | | | | | |
| D060 | I/O ports | IIL | - | - | ±1 | µA | Vss ≤ VPIN ≤ VDD, Pin at hi-impedance |
| D061 | $\overline{MCLR}$, RA4/T0CKI | | - | - | ±5 | µA | Vss ≤ VPIN ≤ VDD |
| D063 | OSC1 | | - | - | ±5 | µA | Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration |
| | **Output Low Voltage** | | | | | | |
| D080 | I/O ports | VOL | - | - | 0.6 | V | IOL = 8.5 mA, VDD = 4.5V, -40°C to +85°C |
| D080A | | | - | - | 0.6 | V | IOL = 7.0 mA, VDD = 4.5V, -40°C to +125°C |
| D083 | OSC2/CLKOUT (RC osc config) | | - | - | 0.6 | V | IOL = 1.6 mA, VDD = 4.5V, -40°C to +85°C |
| D083A | | | - | - | 0.6 | V | IOL = 1.2 mA, VDD = 4.5V, -40°C to +125°C |

* 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 $\overline{MCLR}$/VPP 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.

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

**DC CHARACTERISTICS**

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +125°C for extended,
-40°C ≤ TA ≤ +85°C for industrial and
0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD range as described in DC spec Section 17.1 and Section 17.2.

| 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 |
| D150* | **Open-Drain High Voltage** | VOD | - | - | 14 | V | RA4 pin |
| | **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 | |
| D102 | SCL, SDA in $I^2C$ mode | CB | - | - | 400 | 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 $\overline{MCLR}$/VPP 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.

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

### 17.4 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 | $\overline{WR}$ |

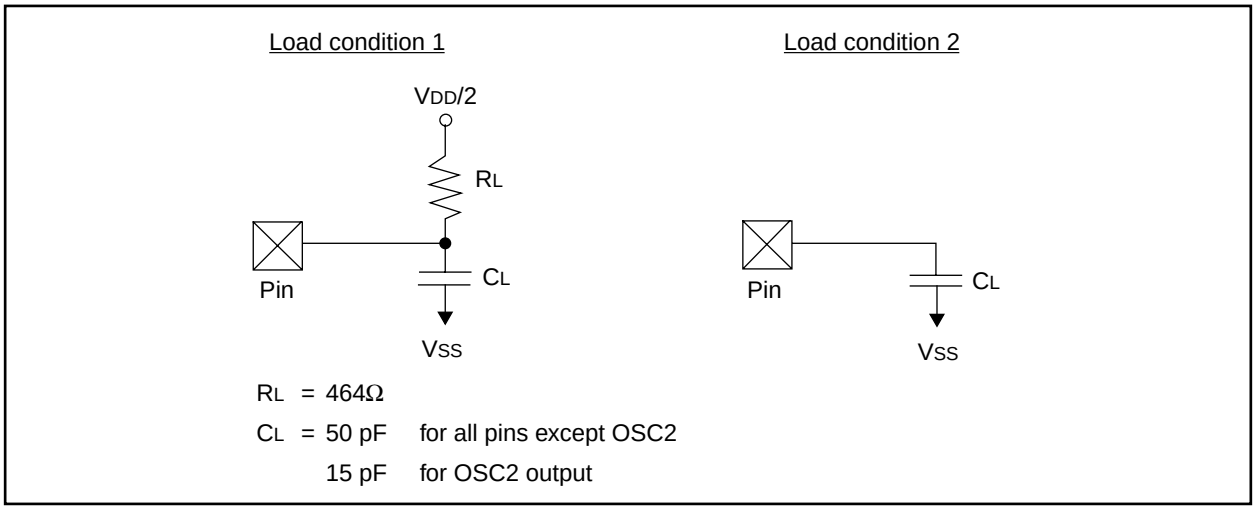
Uppercase letters and their meanings:

| | | | |
|---|---|---|---|
| **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 17-1: LOAD CONDITIONS**




$R_L$ = 464Ω

$C_L$ = 50 pF for all pins except OSC2

15 pF for OSC2 output

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

## 17.5 Timing Diagrams and Specifications

**FIGURE 17-2: EXTERNAL CLOCK TIMING**

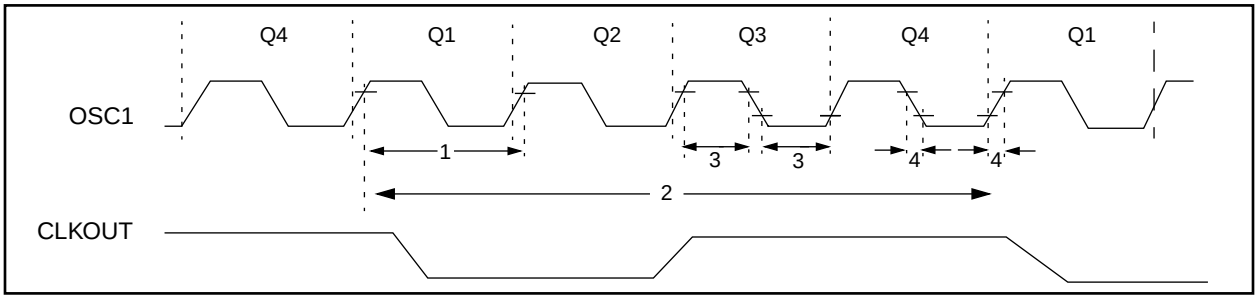



**TABLE 17-2: EXTERNAL CLOCK TIMING REQUIREMENTS**

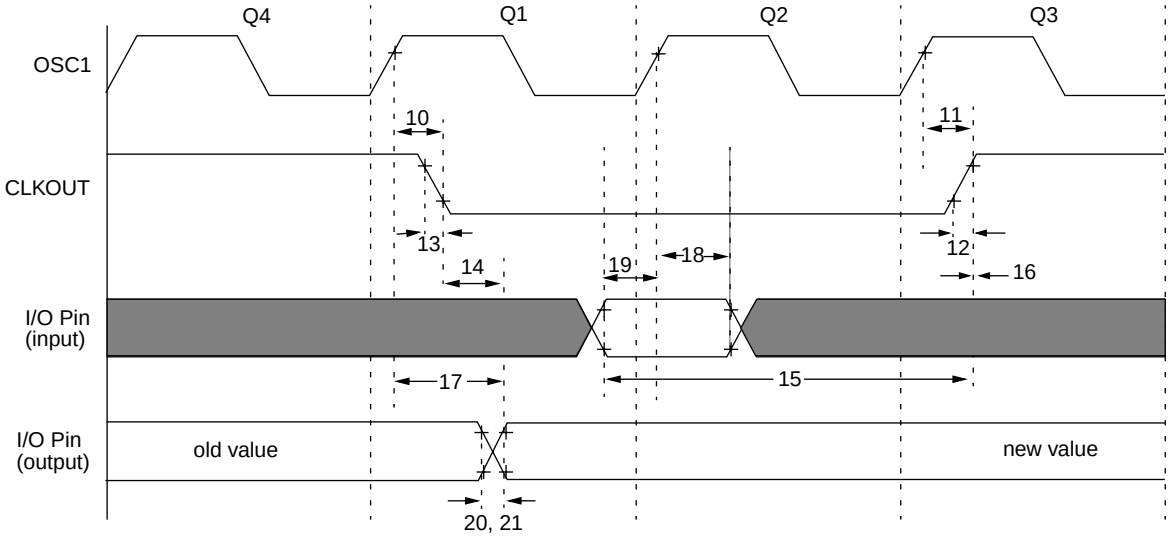
| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | Fosc | **External CLKIN Frequency (Note 1)** | DC | — | 4 | MHz | XT and RC osc mode |
| | | | DC | — | 4 | MHz | HS osc mode (-04) |
| | | | DC | — | 10 | MHz | HS osc mode (-10) |
| | | | DC | — | 20 | MHz | HS osc mode (-20) |
| | | | DC | — | 200 | kHz | LP osc mode |
| | | **Oscillator Frequency (Note 1)** | DC | — | 4 | MHz | RC osc mode |
| | | | 0.1 | — | 4 | MHz | XT osc mode |
| | | | 4 | — | 20 | MHz | HS osc mode |
| | | | 5 | — | 200 | kHz | LP osc mode |
| 1 | Tosc | **External CLKIN Period (Note 1)** | 250 | — | — | ns | XT and RC osc mode |
| | | | 250 | — | — | ns | HS osc mode (-04) |
| | | | 100 | — | — | ns | HS osc mode (-10) |
| | | | 50 | — | — | ns | HS osc mode (-20) |
| | | | 5 | — | — | µs | LP osc mode |
| | | **Oscillator Period (Note 1)** | 250 | — | — | ns | RC osc mode |
| | | | 250 | — | 10,000 | ns | XT osc mode |
| | | | 250 | — | 250 | ns | HS osc mode (-04) |
| | | | 100 | — | 250 | ns | HS osc mode (-10) |
| | | | 50 | — | 250 | ns | HS osc mode (-20) |
| | | | 5 | — | — | µs | LP osc mode |
| 2 | TCY | **Instruction Cycle Time (Note 1)** | 200 | — | DC | ns | TCY = 4/FOSC |
| 3 | TosL, TosH | **External Clock in (OSC1) High or Low Time** | 100 | — | — | ns | XT oscillator |
| | | | 2.5 | — | — | µs | LP oscillator |
| | | | 15 | — | — | ns | HS oscillator |
| 4 | TosR, TosF | **External Clock in (OSC1) Rise or Fall Time** | — | — | 25 | ns | XT oscillator |
| | | | — | — | 50 | ns | LP oscillator |
| | | | — | — | 15 | ns | HS oscillator |

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

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

**FIGURE 17-3: CLKOUT AND I/O TIMING**




Note: Refer to Figure 17-1 for load conditions.

**TABLE 17-3: CLKOUT AND I/O TIMING REQUIREMENTS**

| Parameter No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 10* | TosH2ckL | OSC1↑ to CLKOUT↓ | | — | 75 | 200 | ns | Note 1 |
| 11* | TosH2ckH | OSC1↑ to CLKOUT↑ | | — | 75 | 200 | ns | Note 1 |
| 12* | TckR | CLKOUT rise time | | — | 35 | 100 | ns | Note 1 |
| 13* | TckF | CLKOUT fall time | | — | 35 | 100 | ns | Note 1 |
| 14* | TckL2ioV | CLKOUT ↓ to Port out valid | | — | — | 0.5TCY + 20 | ns | Note 1 |
| 15* | TioV2ckH | Port in valid before CLKOUT ↑ | | TOSC + 200 | — | — | ns | Note 1 |
| 16* | TckH2ioI | Port in hold after CLKOUT ↑ | | 0 | — | — | ns | Note 1 |
| 17* | TosH2ioV | OSC1↑ (Q1 cycle) to Port out valid | | — | 50 | 150 | ns | |
| 18* | TosH2ioI | OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time) | PIC16**C**72 | 100 | — | — | ns | |
| | | | PIC16**LC**72 | 200 | — | — | ns | |
| 19* | TioV2osH | Port input valid to OSC1↑ (I/O in setup time) | | 0 | — | — | ns | |
| 20* | TioR | Port output rise time | PIC16**C**72 | — | 10 | 40 | ns | |
| | | | PIC16**LC**72 | — | — | 80 | ns | |
| 21* | TioF | Port output fall time | PIC16**C**72 | — | 10 | 40 | ns | |
| | | | PIC16**LC**72 | — | — | 80 | ns | |
| 22††* | Tinp | INT pin high or low time | | TCY | — | — | ns | |
| 23††* | Trbp | RB7:RB4 change INT high or low time | | TCY | — | — | ns | |

* These parameters are characterized but not tested.

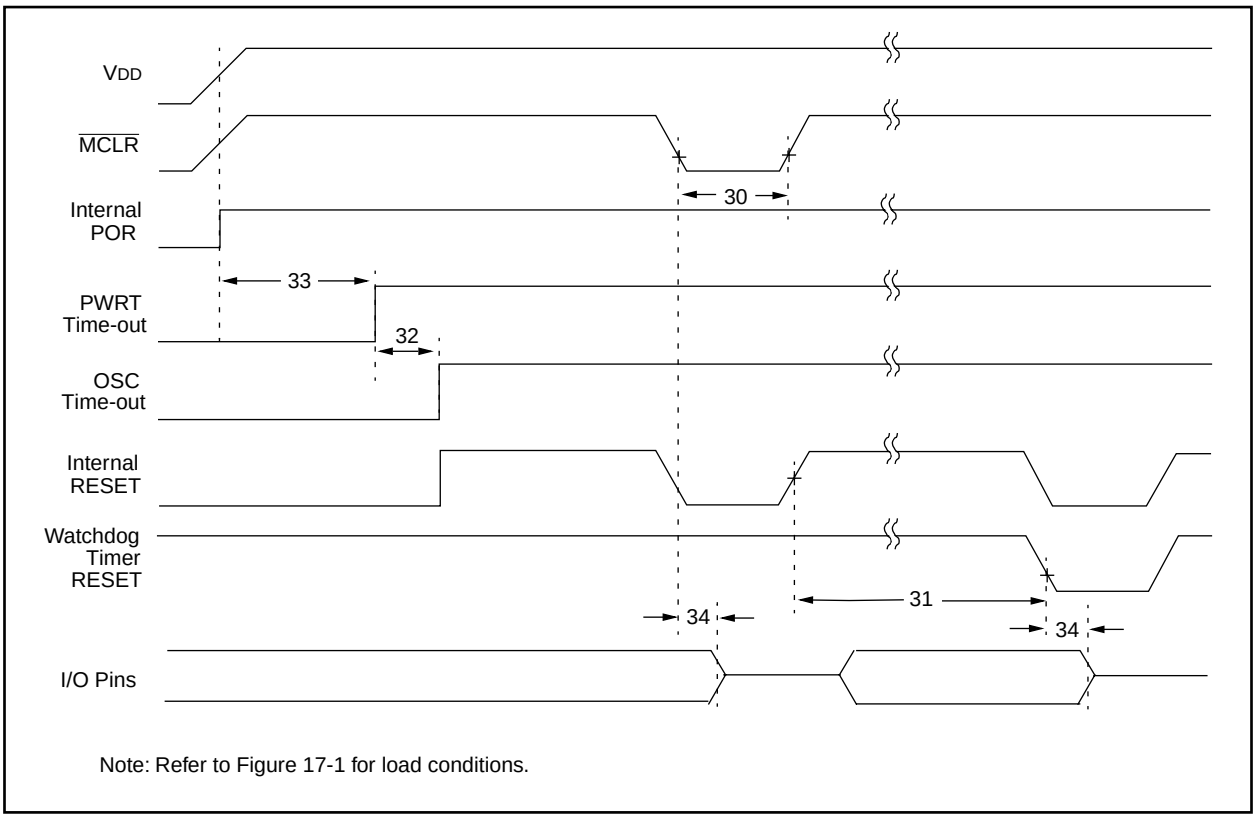
†Data in "Typ" column is at 5V, 25˚C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† 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 TOSC.

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

**FIGURE 17-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**




Note: Refer to Figure 17-1 for load conditions.

**FIGURE 17-5: BROWN-OUT RESET TIMING**

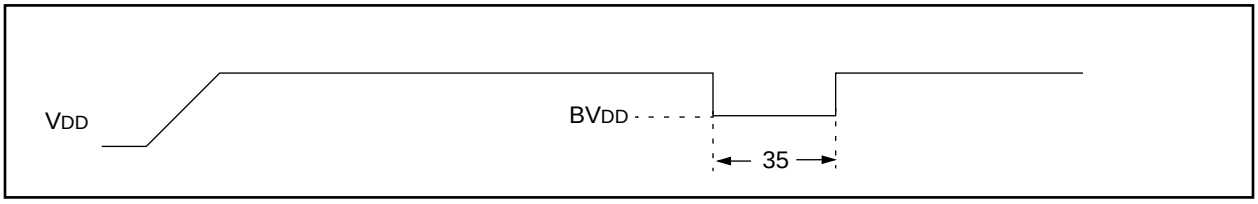



**TABLE 17-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS**

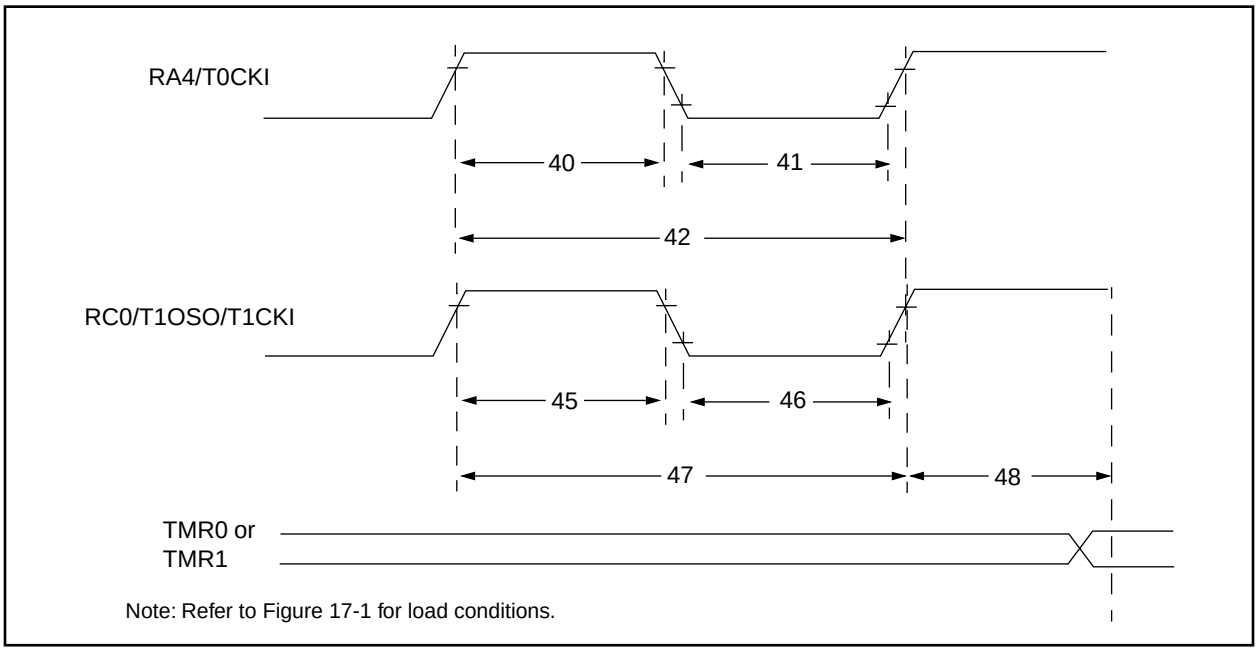
| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 30 | TmcL | $\overline{MCLR}$ Pulse Width (low) | 2 | — | — | µs | VDD = 5V, -40˚C to +125˚C |
| 31* | Twdt | Watchdog Timer Time-out Period (No Prescaler) | 7 | 18 | 33 | ms | VDD = 5V, -40˚C to +125˚C |
| 32 | Tost | Oscillation Start-up Timer Period | — | 1024TOSC | — | — | TOSC = OSC1 period |
| 33* | Tpwrt | Power-up Timer Period | 28 | 72 | 132 | ms | VDD = 5V, -40˚C to +125˚C |
| 34 | TIOZ | I/O Hi-impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset | — | — | 2.1 | µs | |
| 35 | TBOR | Brown-out Reset pulse width | 100 | — | — | µs | VDD ≤ BVDD (D005) |

* 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.

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

**FIGURE 17-6: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**




Note: Refer to Figure 17-1 for load conditions.

**TABLE 17-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

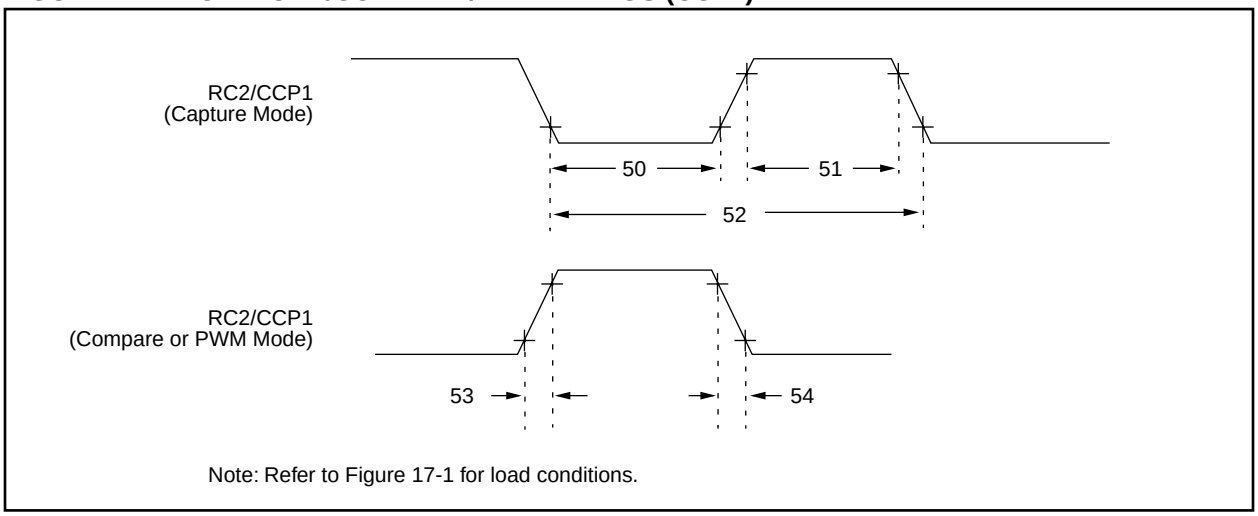
| Param No. | Sym | Characteristic | | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 40* | Tt0H | T0CKI High Pulse Width | | No Prescaler | 0.5TCY + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With Prescaler | 10 | — | — | ns | |
| 41* | Tt0L | T0CKI Low Pulse Width | | No Prescaler | 0.5TCY + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With Prescaler | 10 | — | — | ns | |
| 42* | Tt0P | T0CKI Period | | No Prescaler | TCY + 40 | — | — | ns | |
| | | | | With Prescaler | Greater of: 20 or TCY + 40 / N | — | — | ns | N = prescale value (2, 4, ..., 256) |
| 45* | Tt1H | T1CKI High Time | Synchronous, Prescaler = 1 | | 0.5TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | PIC16**C**7X | 15 | — | — | ns | |
| | | | | PIC16**LC**7X | 25 | — | — | ns | |
| | | | Asynchronous | PIC16**C**7X | 30 | — | — | ns | |
| | | | | PIC16**LC**7X | 50 | — | — | ns | |
| 46* | Tt1L | T1CKI Low Time | Synchronous, Prescaler = 1 | | 0.5TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | PIC16**C**7X | 15 | — | — | ns | |
| | | | | PIC16**LC**7X | 25 | — | — | ns | |
| | | | Asynchronous | PIC16**C**7X | 30 | — | — | ns | |
| | | | | PIC16**LC**7X | 50 | — | — | ns | |
| 47* | Tt1P | T1CKI input period | Synchronous | PIC16**C**7X | Greater of: 30 OR TCY + 40 / N | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | | PIC16**LC**7X | Greater of: 50 OR TCY + 40 / N | | | | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | PIC16**C**7X | 60 | — | — | ns | |
| | | | | PIC16**LC**7X | 100 | — | — | ns | |
| | Ft1 | Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN) | | | DC | — | 200 | kHz | |
| 48 | TCKEZtmr1 | Delay from external clock edge to timer increment | | | 2Tosc | — | 7Tosc | — | |

* 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.

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

**FIGURE 17-7: CAPTURE/COMPARE/PWM TIMINGS (CCP1)**




Note: Refer to Figure 17-1 for load conditions.

**TABLE 17-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1)**

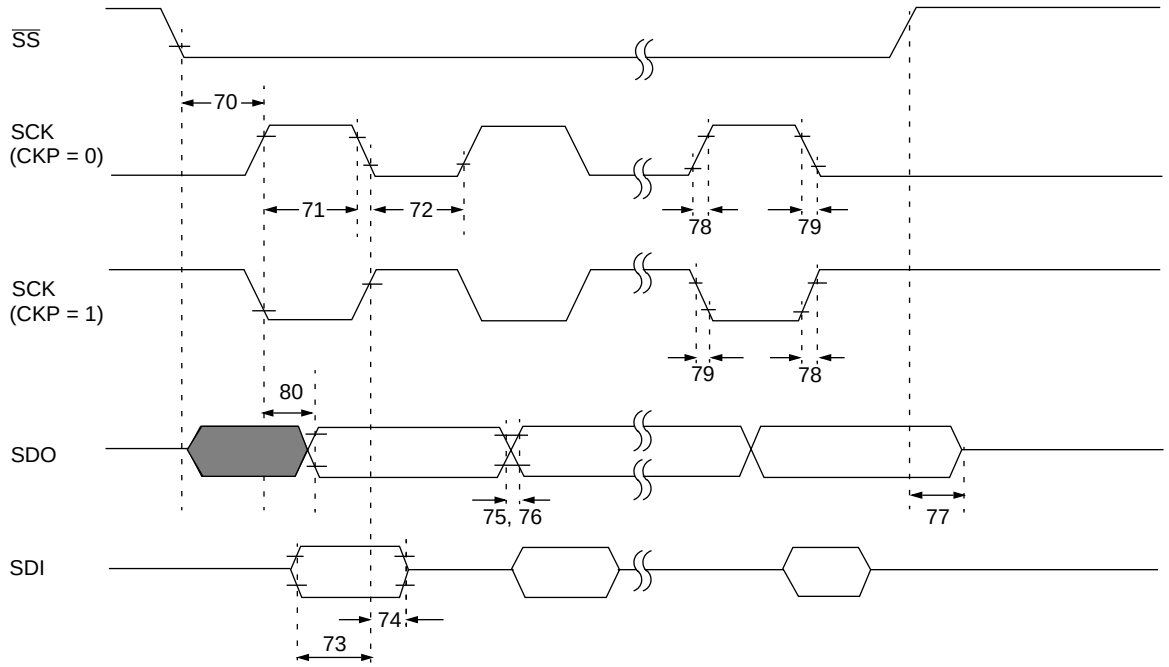
| Param No. | Sym | Characteristic | | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 50* | TccL | CCP1 input low time | No Prescaler | | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | PIC16**C**72 | 10 | — | — | ns | |
| | | | | PIC16**LC**72 | 20 | — | — | ns | |
| 51* | TccH | CCP1 input high time | No Prescaler | | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | PIC16**C**72 | 10 | — | — | ns | |
| | | | | PIC16**LC**72 | 20 | — | — | ns | |
| 52* | TccP | CCP1 input period | | | $\frac{3T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1,4 or 16) |
| 53* | TccR | CCP1 output rise time | | PIC16**C**72 | — | 10 | 25 | ns | |
| | | | | PIC16**LC**72 | — | 25 | 45 | ns | |
| 54* | TccF | CCP1 output fall time | | PIC16**C**72 | — | 10 | 25 | ns | |
| | | | | PIC16**LC**72 | — | 25 | 45 | 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.

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

**FIGURE 17-8: SPI MODE TIMING**




Note: Refer to Figure 17-1 for load conditions

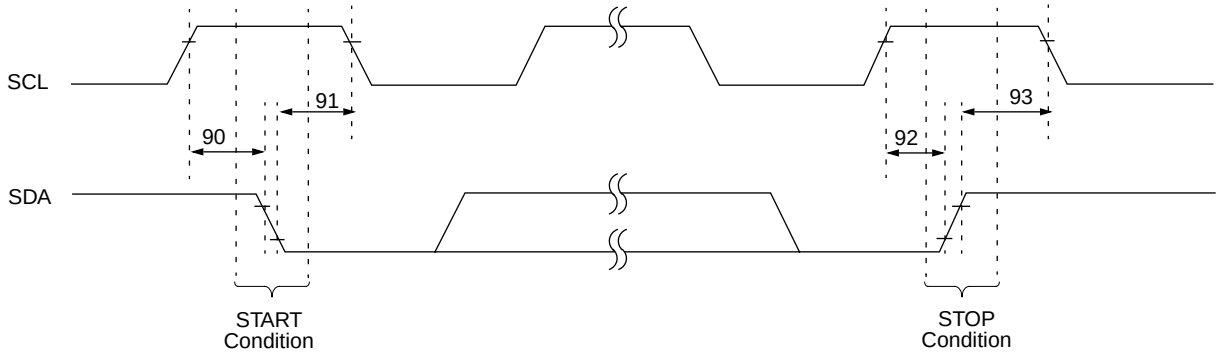
**TABLE 17-7: SPI MODE REQUIREMENTS**

| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 70 | TssL2scH, TssL2scL | $\overline{SS}\downarrow$ to SCK$\downarrow$ or SCK$\uparrow$ input | TCY | — | — | ns | |
| 71 | TscH | SCK input high time (slave mode) | TCY + 20 | — | — | ns | |
| 72 | TscL | SCK input low time (slave mode) | TCY + 20 | — | — | ns | |
| 73 | TdiV2scH, TdiV2scL | Setup time of SDI data input to SCK edge | 50 | — | — | ns | |
| 74 | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | 50 | — | — | ns | |
| 75 | TdoR | SDO data output rise time | — | 10 | 25 | ns | |
| 76 | TdoF | SDO data output fall time | — | 10 | 25 | ns | |
| 77 | TssH2doZ | $\overline{SS}\uparrow$ to SDO output hi-impedance | 10 | — | 50 | ns | |
| 78 | TscR | SCK output rise time (master mode) | — | 10 | 25 | ns | |
| 79 | TscF | SCK output fall time (master mode) | — | 10 | 25 | ns | |
| 80 | TscH2doV, TscL2doV | SDO data output valid after SCK edge | — | — | 50 | ns | |

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

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

**FIGURE 17-9: $I^2C$ BUS START/STOP BITS TIMING**



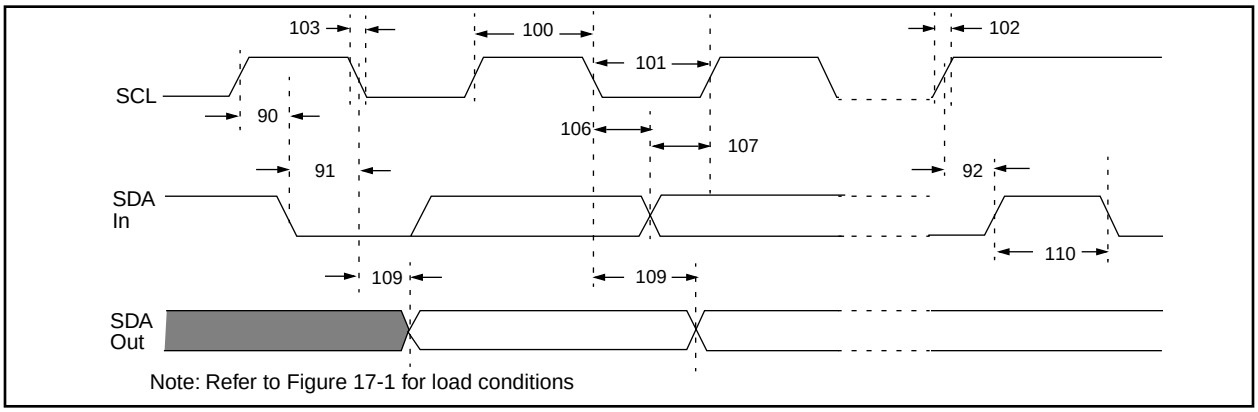

Note: Refer to Figure 17-1 for load conditions

**TABLE 17-8: $I^2C$ BUS START/STOP BITS REQUIREMENTS**

| Parameter No. | Sym | Characteristic | | Min | Typ | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 90 | TSU:STA | START condition Setup time | 100 kHz mode | 4700 | — | — | ns | Only relevant for repeated START condition |
| | | | 400 kHz mode | 600 | — | — | | |
| 91 | THD:STA | START condition Hold time | 100 kHz mode | 4000 | — | — | ns | After this period the first clock pulse is generated |
| | | | 400 kHz mode | 600 | — | — | | |
| 92 | TSU:STO | STOP condition Setup time | 100 kHz mode | 4700 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |
| 93 | THD:STO | STOP condition Hold time | 100 kHz mode | 4000 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |

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

**FIGURE 17-10: $I^2C$ BUS DATA TIMING**




Note: Refer to Figure 17-1 for load conditions

**TABLE 17-9: $I^2C$ BUS DATA REQUIREMENTS**

| Parameter No. | Sym | Characteristic | | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 100 | THIGH | Clock high time | 100 kHz mode | 4.0 | — | µs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 0.6 | — | µs | Device must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5TCY | — | | |
| 101 | TLOW | Clock low time | 100 kHz mode | 4.7 | — | µs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 1.3 | — | µs | Device must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5TCY | — | | |
| 102 | TR | SDA and SCL rise time | 100 kHz mode | — | 1000 | ns | |
| | | | 400 kHz mode | 20 + 0.1Cb | 300 | ns | Cb is specified to be from 10 to 400 pF |
| 103 | TF | SDA and SCL fall time | 100 kHz mode | — | 300 | ns | |
| | | | 400 kHz mode | 20 + 0.1Cb | 300 | ns | Cb is specified to be from 10 to 400 pF |
| 90 | TSU:STA | START condition setup time | 100 kHz mode | 4.7 | — | µs | Only relevant for repeated START condition |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 91 | THD:STA | START condition hold time | 100 kHz mode | 4.0 | — | µs | After this period the first clock pulse is generated |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 106 | THD:DAT | Data input hold time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | µs | |
| 107 | TSU:DAT | Data input setup time | 100 kHz mode | 250 | — | ns | Note 2 |
| | | | 400 kHz mode | 100 | — | ns | |
| 92 | TSU:STO | STOP condition setup time | 100 kHz mode | 4.7 | — | µs | |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 109 | TAA | Output valid from clock | 100 kHz mode | — | 3500 | ns | Note 1 |
| | | | 400 kHz mode | — | — | ns | |
| 110 | TBUF | Bus free time | 100 kHz mode | 4.7 | — | µs | Time the bus must be free before a new transmission can start |
| | | | 400 kHz mode | 1.3 | — | µs | |
| | Cb | Bus capacitive loading | | — | 400 | pF | |

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A fast-mode (400 kHz) $I^2C$-bus device can be used in a standard-mode (100 kHz)S $I^2C$-bus system, but the requirement tsu;DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line TR max.+tsu;DAT = 1000 + 250 = 1250 ns (according to the standard-mode $I^2C$ bus specification) before the SCL line is released.

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

**TABLE 17-10: A/D CONVERTER CHARACTERISTICS:**
**PIC16C72-04 (Commercial, Industrial, Extended)**
**PIC16C72-10 (Commercial, Industrial, Extended)**
**PIC16C72-20 (Commercial, Industrial, Extended)**
**PIC16LC72-04 (Commercial, Industrial)**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| A01 | NR | Resolution | | — | — | 8-bits | bit | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A02 | EABS | Total Absolute error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A03 | EIL | Integral linearity error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A04 | EDL | Differential linearity error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A05 | EFS | Full scale error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A06 | EOFF | Offset error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A10 | — | Monotonicity | | — | guaranteed | — | — | VSS ≤ VAIN ≤ VREF |
| A20 | VREF | Reference voltage | | 3.0V | — | VDD + 0.3 | V | |
| A25 | VAIN | Analog input voltage | | VSS - 0.3 | — | VREF + 0.3 | V | |
| A30 | ZAIN | Recommended impedance of analog voltage source | | — | — | 10.0 | kΩ | |
| A40 | IAD | A/D conversion current (VDD) | PIC16**C**72 | — | 180 | — | μA | Average current consumption when A/D is on. (Note 1) |
| | | | PIC16**LC**72 | — | 90 | — | μA | |
| A50 | IREF | VREF input current (Note 2) | | 10 | — | 1000 | μA | During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see Section 13.1. |
| | | | | — | — | 10 | μA | During A/D Conversion cycle |

* These parameters are characterized but not tested.

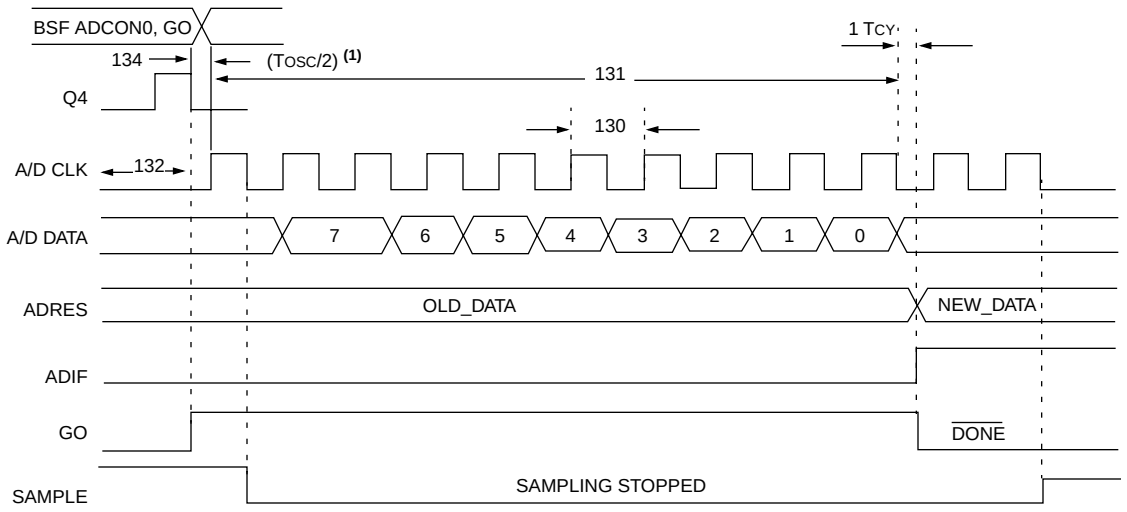
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

2: VREF current is from RA3 pin or VDD pin, whichever is selected as reference input.

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

**FIGURE 17-11: 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 17-11: A/D CONVERSION REQUIREMENTS**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 130 | TAD | A/D clock period | PIC16**C**72 | 1.6 | — | — | µs | TOSC based, VREF ≥ 3.0V |
| | | | PIC16**LC**72 | 2.0 | — | — | µs | TOSC based, VREF full range |
| | | | PIC16**C**72 | 2.0 | 4.0 | 6.0 | µs | A/D RC Mode |
| | | | PIC16**LC**72 | 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., 20.0 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.

§ This specification ensured by design.

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

2: See Section 13.1 for min conditions.

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

## 18.0 ELECTRICAL CHARACTERISTICS FOR PIC16C73/74

**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 VSS (except VDD, $\overline{MCLR}$. and RA4) ..... -0.3V to (VDD + 0.3V)

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

Voltage on $\overline{MCLR}$ with respect to VSS (Note 2) ..... 0 to +14V

Voltage on RA4 with respect to Vss ..... 0 to +14V

Total power dissipation (Note 1) ..... 1.0W

Maximum current out of VSS pin ..... 300 mA

Maximum current into VDD pin ..... 250 mA

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

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

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

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

Maximum current sunk by PORTA, PORTB, and PORTE (combined) (Note 3) ..... 200 mA

Maximum current sourced by PORTA, PORTB, and PORTE (combined) (Note 3) ..... 200 mA

Maximum current sunk by PORTC and PORTD (combined) (Note 3) ..... 200 mA

Maximum current sourced by PORTC and PORTD (combined) (Note 3) ..... 200 mA

**Note 1:** Power dissipation is calculated as follows: Pdis = VDD x {IDD - ∑ IOH} + ∑ {(VDD - VOH) x IOH} + ∑(Vol x IOL)

**Note 2:** Voltage spikes below VSS at the $\overline{MCLR}$ 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 VSS.

**Note 3:** PORTD and PORTE are not implemented on the PIC16C73.

† 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 18-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)**

| OSC | PIC16C73-04<br>PIC16C74-04 | PIC16C73-10<br>PIC16C74-10 | PIC16C73-20<br>PIC16C74-20 | PIC16LC73-04<br>PIC16LC74-04 | JW Devices |
|---|---|---|---|---|---|
| RC | VDD: 4.0V to 6.0V<br>IDD: 5 mA max. at 5.5V<br>IPD: 21 µA max. at 4V<br>Freq: 4 MHz max. | VDD: 4.5V to 5.5V<br>IDD: 2.7 mA typ. at 5.5V<br>IPD: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | VDD: 4.5V to 5.5V<br>IDD: 2.7 mA typ. at 5.5V<br>IPD: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | VDD: 3.0V to 6.0V<br>IDD: 3.8 mA max. at 3.0V<br>IPD: 13.5 µA max. at 3V<br>Freq: 4 MHz max. | VDD: 4.0V to 6.0V<br>IDD: 5 mA max. at 5.5V<br>IPD: 21 µA max. at 4V<br>Freq: 4 MHz max. |
| XT | VDD: 4.0V to 6.0V<br>IDD: 5 mA max. at 5.5V<br>IPD: 21 µA max. at 4V<br>Freq: 4 MHz max. | VDD: 4.5V to 5.5V<br>IDD: 2.7 mA typ. at 5.5V<br>IPD: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | VDD: 4.5V to 5.5V<br>IDD: 2.7 mA typ. at 5.5V<br>IPD: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | VDD: 3.0V to 6.0V<br>IDD: 3.8 mA max. at 3.0V<br>IPD: 13.5 µA max. at 3V<br>Freq: 4 MHz max. | VDD: 4.0V to 6.0V<br>IDD: 5 mA max. at 5.5V<br>IPD: 21 µA max. at 4V<br>Freq: 4 MHz max. |
| HS | VDD: 4.5V to 5.5V<br>IDD: 13.5 mA typ. at 5.5V<br>IPD: 1.5 µA typ. at 4.5V<br>Freq: 4 MHz max. | VDD: 4.5V to 5.5V<br>IDD: 15 mA max. at 5.5V<br>IPD: 1.5 µA typ. at 4.5V<br>Freq: 10 MHz max. | VDD: 4.5V to 5.5V<br>IDD: 30 mA max. at 5.5V<br>IPD: 1.5 µA typ. at 4.5V<br>Freq: 20 MHz max. | Not recommended for use in HS mode | VDD: 4.5V to 5.5V<br>IDD: 30 mA max. at 5.5V<br>IPD: 1.5 µA typ. at 4.5V<br>Freq: 20 MHz max. |
| LP | VDD: 4.0V to 6.0V<br>IDD: 52.5 µA typ. at 32 kHz, 4.0V<br>IPD: 0.9 µA typ. at 4.0V<br>Freq: 200 kHz max. | Not recommended for use in LP mode | Not recommended for use in LP mode | VDD: 3.0V to 6.0V<br>IDD: 48 µA max. at 32 kHz, 3.0V<br>IPD: 13.5 µA max. at 3.0V<br>Freq: 200 kHz max. | VDD: 3.0V to 6.0V<br>IDD: 48 µA max. at 32 kHz, 3.0V<br>IPD: 13.5 µA max. at 3.0V<br>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 | 72 | 73 | 73A | 74 | 74A | 76 | 77 |
|---|---|---|---|---|---|---|---|

### 18.1 DC Characteristics: PIC16C73/74-04 (Commercial, Industrial) PIC16C73/74-10 (Commercial, Industrial) PIC16C73/74-20 (Commercial, Industrial)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -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 | Supply Voltage | VDD | 4.0 | - | 6.0 | V | XT, RC and LP osc configuration |
| D001A | | | 4.5 | - | 5.5 | V | 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 |
| D010 | Supply Current (Note 2,5) | IDD | - | 2.7 | 5 | mA | XT, RC osc configuration<br>FOSC = 4 MHz, VDD = 5.5V (Note 4) |
| D013 | | | - | 13.5 | 30 | mA | HS osc configuration<br>FOSC = 20 MHz, VDD = 5.5V |
| D020 | Power-down Current (Note 3,5) | IPD | - | 10.5 | 42 | µA | VDD = 4.0V, WDT enabled, -40°C to +85°C |
| D021 | | | - | 1.5 | 21 | µA | VDD = 4.0V, WDT disabled, -0°C to +70°C |
| D021A | | | - | 1.5 | 24 | µA | VDD = 4.0V, WDT disabled, -40°C to +85°C |

* 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.

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

## 18.2 DC Characteristics: PIC16LC73/74-04 (Commercial, Industrial)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -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 | Supply Voltage | VDD | 3.0 | - | 6.0 | V | LP, XT, RC osc configuration (DC - 4 MHz) |
| 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 |
| D010 | Supply Current (Note 2,5) | IDD | - | 2.0 | 3.8 | mA | XT, RC osc configuration<br>FOSC = 4 MHz, VDD = 3.0V (Note 4) |
| D010A | | | - | 22.5 | 48 | µA | LP osc configuration<br>FOSC = 32 kHz, VDD = 3.0V, WDT disabled |
| D020 | Power-down Current (Note 3,5) | IPD | - | 7.5 | 30 | µA | VDD = 3.0V, WDT enabled, -40°C to +85°C |
| D021 | | | - | 0.9 | 13.5 | µA | VDD = 3.0V, WDT disabled, 0°C to +70°C |
| D021A | | | - | 0.9 | 18 | µA | VDD = 3.0V, WDT disabled, -40°C to +85°C |

* 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.

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

### 18.3 DC Characteristics: PIC16C73/74-04 (Commercial, Industrial)
### PIC16C73/74-10 (Commercial, Industrial)
### PIC16C73/74-20 (Commercial, Industrial)
### PIC16LC73/74-04 (Commercial, Industrial)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial and<br>0°C ≤ TA ≤ +70°C for commercial<br>Operating voltage VDD range as described in DC spec Section 18.1 and Section 18.2. | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Characteristic** | **Sym** | **Min** | **Typ †** | **Max** | **Units** | **Conditions** |
| | **Input Low Voltage** | | | | | | |
| | I/O ports | VIL | | | | | |
| D030 | with TTL buffer | | VSS | - | 0.15VDD | V | For entire VDD range |
| D030A | | | VSS | - | 0.8V | V | 4.5V ≤ VDD ≤ 5.5V |
| D031 | with Schmitt Trigger buffer | | VSS | - | 0.2VDD | V | |
| D032 | MCLR, OSC1 (in RC mode) | | VSS | - | 0.2VDD | V | |
| D033 | OSC1 (in XT, HS and LP) | | VSS | - | 0.3VDD | V | Note1 |
| | **Input High Voltage** | | | | | | |
| | I/O ports | VIH | | - | | | |
| D040 | with TTL buffer | | 2.0 | - | VDD | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | 0.25VDD + 0.8V | - | VDD | V | For entire VDD range |
| D041 | with Schmitt Trigger buffer | | 0.8VDD | - | VDD | V | For entire VDD range |
| D042 | MCLR | | 0.8VDD | - | VDD | V | |
| D042A | OSC1 (XT, HS and LP) | | 0.7VDD | - | VDD | V | Note1 |
| D043 | OSC1 (in RC mode) | | 0.9VDD | - | VDD | V | |
| D070 | PORTB weak pull-up current | IPURB | 50 | 250 | 400 | µA | VDD = 5V, VPIN = VSS |
| | **Input Leakage Current** (Notes 2, 3) | | | | | | |
| D060 | I/O ports | IIL | - | - | ±1 | µA | Vss ≤ VPIN ≤ VDD, Pin at hi-impedance |
| D061 | MCLR, RA4/T0CKI | | - | - | ±5 | µA | Vss ≤ VPIN ≤ VDD |
| D063 | OSC1 | | - | - | ±5 | µA | Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration |
| | **Output Low Voltage** | | | | | | |
| D080 | I/O ports | VOL | - | - | 0.6 | V | IOL = 8.5 mA, VDD = 4.5V, -40°C to +85°C |
| D083 | OSC2/CLKOUT (RC osc config) | | - | - | 0.6 | V | IOL = 1.6 mA, VDD = 4.5V, -40°C to +85°C |
| | **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 |
| D092 | OSC2/CLKOUT (RC osc config) | | VDD - 0.7 | - | - | V | IOH = -1.3 mA, VDD = 4.5V, -40°C to +85°C |
| D150* | **Open-Drain High Voltage** | VOD | - | - | 14 | V | RA4 pin |

* 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 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.

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

**DC CHARACTERISTICS**

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +85°C for industrial and
0°C ≤ TA ≤ +70°C for commercial
Operating voltage VDD range as described in DC spec Section 18.1 and Section 18.2.

| Param No. | Characteristic | Sym | Min | Typ † | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | **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 | |
| D102 | SCL, SDA in $I^2C$ mode | CB | - | - | 400 | 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.

| 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 | $\overline{WR}$ |

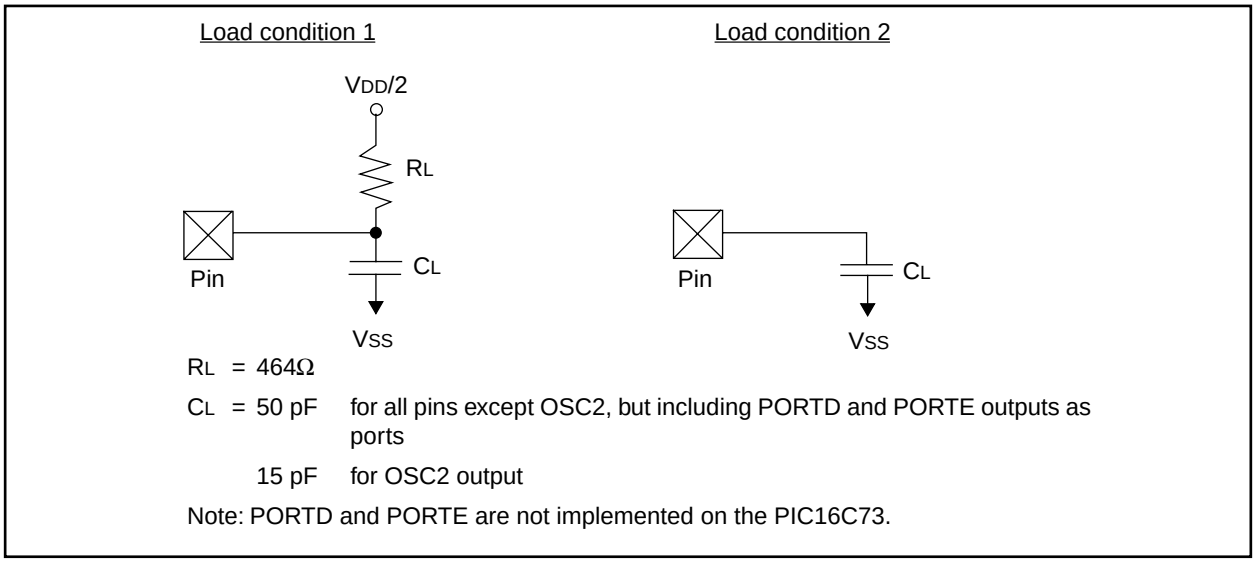
Uppercase letters and their meanings:

| | | | |
|---|---|---|---|
| **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 |
|---|---|---|---|---|---|---|---|

## 18.5 Timing Diagrams and Specifications

**FIGURE 18-2: EXTERNAL CLOCK TIMING**

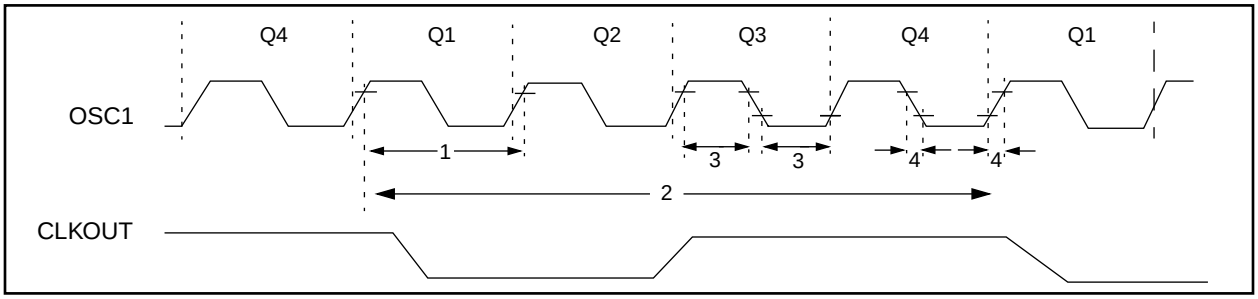



**TABLE 18-2: EXTERNAL CLOCK TIMING REQUIREMENTS**

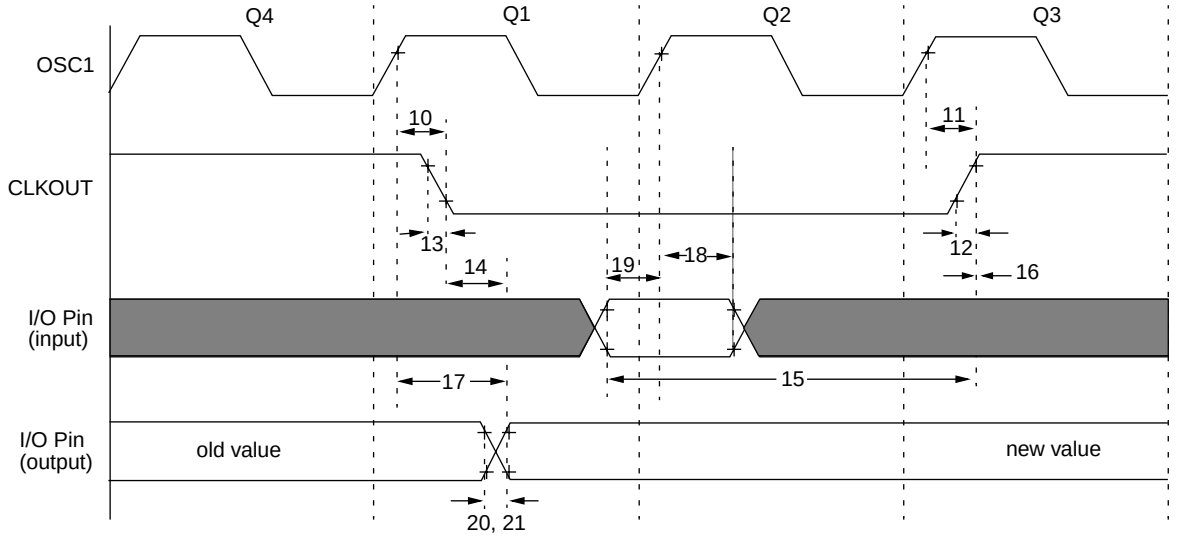
| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | Fosc | **External CLKIN Frequency (Note 1)** | DC | — | 4 | MHz | XT and RC osc mode |
| | | | DC | — | 4 | MHz | HS osc mode (-04) |
| | | | DC | — | 10 | MHz | HS osc mode (-10) |
| | | | DC | — | 20 | MHz | HS osc mode (-20) |
| | | | DC | — | 200 | kHz | LP osc mode |
| | | **Oscillator Frequency (Note 1)** | DC | — | 4 | MHz | RC osc mode |
| | | | 0.1 | — | 4 | MHz | XT osc mode |
| | | | 4 | — | 20 | MHz | HS osc mode |
| | | | 5 | — | 200 | kHz | LP osc mode |
| 1 | Tosc | **External CLKIN Period (Note 1)** | 250 | — | — | ns | XT and RC osc mode |
| | | | 250 | — | — | ns | HS osc mode (-04) |
| | | | 100 | — | — | ns | HS osc mode (-10) |
| | | | 50 | — | — | ns | HS osc mode (-20) |
| | | | 5 | — | — | µs | LP osc mode |
| | | **Oscillator Period (Note 1)** | 250 | — | — | ns | RC osc mode |
| | | | 250 | — | 10,000 | ns | XT osc mode |
| | | | 250 | — | 250 | ns | HS osc mode (-04) |
| | | | 100 | — | 250 | ns | HS osc mode (-10) |
| | | | 50 | — | 250 | ns | HS osc mode (-20) |
| | | | 5 | — | — | µs | LP osc mode |
| 2 | TCY | **Instruction Cycle Time (Note 1)** | 200 | — | DC | ns | TCY = 4/FOSC |
| 3 | TosL, TosH | **External Clock in (OSC1) High or Low Time** | 50 | — | — | ns | XT oscillator |
| | | | 2.5 | — | — | µs | LP oscillator |
| | | | 15 | — | — | ns | HS oscillator |
| 4 | TosR, TosF | **External Clock in (OSC1) Rise or Fall Time** | — | — | 25 | ns | XT oscillator |
| | | | — | — | 50 | ns | LP oscillator |
| | | | — | — | 15 | ns | HS oscillator |

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

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

**FIGURE 18-3: CLKOUT AND I/O TIMING**




Note: Refer to Figure 18-1 for load conditions.

**TABLE 18-3: CLKOUT AND I/O TIMING REQUIREMENTS**

| Parameter No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 10* | TosH2ckL | OSC1↑ to CLKOUT↓ | | — | 75 | 200 | ns | Note 1 |
| 11* | TosH2ckH | OSC1↑ to CLKOUT↑ | | — | 75 | 200 | ns | Note 1 |
| 12* | TckR | CLKOUT rise time | | — | 35 | 100 | ns | Note 1 |
| 13* | TckF | CLKOUT fall time | | — | 35 | 100 | ns | Note 1 |
| 14* | TckL2ioV | CLKOUT ↓ to Port out valid | | — | — | 0.5TCY + 20 | ns | Note 1 |
| 15* | TioV2ckH | Port in valid before CLKOUT ↑ | | 0.25TCY + 25 | — | — | ns | Note 1 |
| 16* | TckH2ioI | Port in hold after CLKOUT ↑ | | 0 | — | — | ns | Note 1 |
| 17* | TosH2ioV | OSC1↑ (Q1 cycle) to Port out valid | | — | 50 | 150 | ns | |
| 18* | TosH2ioI | OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time) | PIC16**C**73/74 | 100 | — | — | ns | |
| | | | PIC16**LC**73/74 | 200 | — | — | ns | |
| 19* | TioV2osH | Port input valid to OSC1↑ (I/O in setup time) | | 0 | — | — | ns | |
| 20* | TioR | Port output rise time | PIC16**C**73/74 | — | 10 | 25 | ns | |
| | | | PIC16**LC**73/74 | — | — | 60 | ns | |
| 21* | TioF | Port output fall time | PIC16**C**73/74 | — | 10 | 25 | ns | |
| | | | PIC16**LC**73/74 | — | — | 60 | ns | |
| 22††* | Tinp | INT pin high or low time | | TCY | — | — | ns | |
| 23††* | Trbp | RB7:RB4 change INT high or low time | | TCY | — | — | ns | |

* These parameters are characterized but not tested.

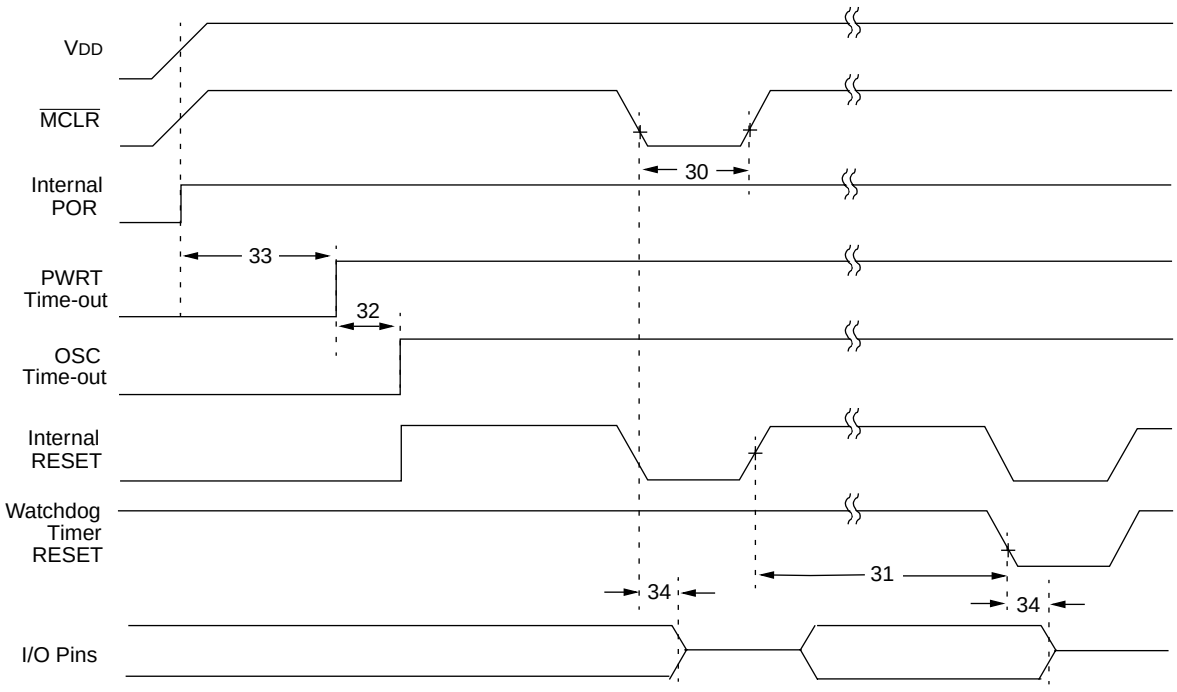
†Data in "Typ" column is at 5V, 25˚C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† 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 TOSC.

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

**FIGURE 18-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**




Note: Refer to Figure 18-1 for load conditions.

**TABLE 18-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS**

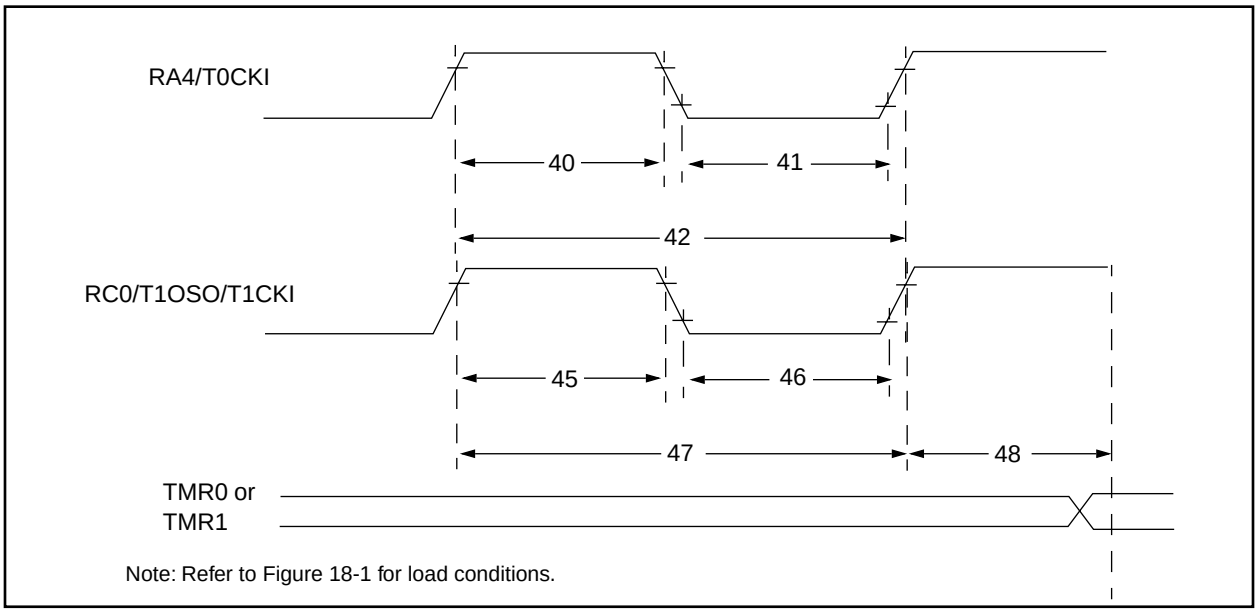
| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 30 | TmcL | $\overline{MCLR}$ Pulse Width (low) | 100 | — | — | ns | VDD = 5V, -40˚C to +85˚C |
| 31* | Twdt | Watchdog Timer Time-out Period (No Prescaler) | 7 | 18 | 33 | ms | VDD = 5V, -40˚C to +85˚C |
| 32 | Tost | Oscillation Start-up Timer Period | — | 1024TOSC | — | — | TOSC = OSC1 period |
| 33* | Tpwrt | Power up Timer Period | 28 | 72 | 132 | ms | VDD = 5V, -40˚C to +85˚C |
| 34 | TIOZ | I/O Hi-impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset | — | — | 100 | 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.

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

**FIGURE 18-5: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**




Note: Refer to Figure 18-1 for load conditions.

**TABLE 18-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

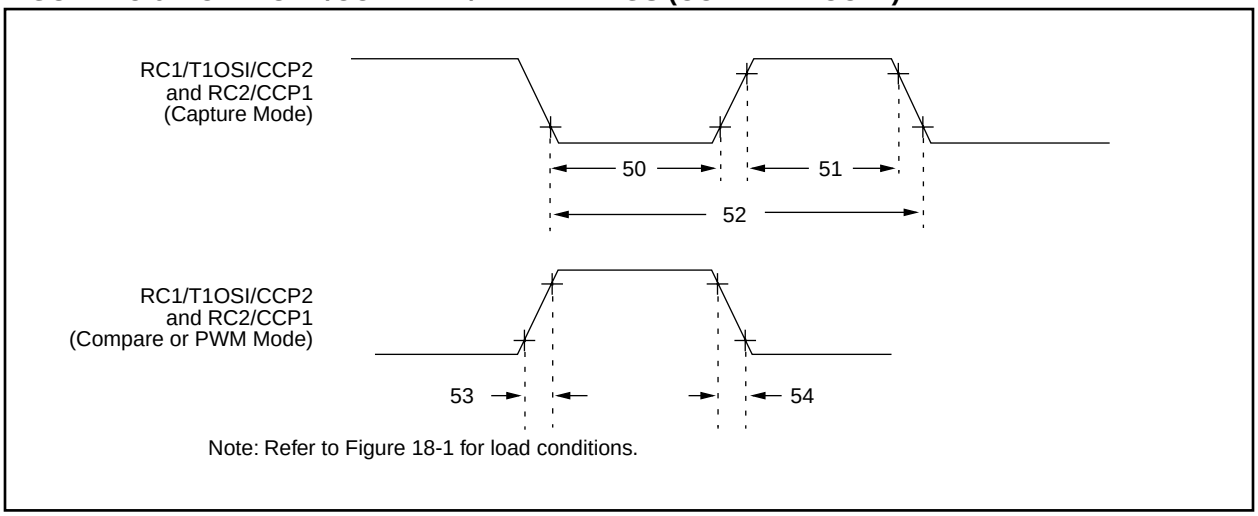
| Param No. | Sym | Characteristic | | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 40* | Tt0H | T0CKI High Pulse Width | | No Prescaler | 0.5TCY + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With Prescaler | 10 | — | — | ns | |
| 41* | Tt0L | T0CKI Low Pulse Width | | No Prescaler | 0.5TCY + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With Prescaler | 10 | — | — | ns | |
| 42* | Tt0P | T0CKI Period | | No Prescaler | TCY + 40 | — | — | ns | |
| | | | | With Prescaler | Greater of: 20 or TCY + 40 / N | — | — | ns | N = prescale value (2, 4, ..., 256) |
| 45* | Tt1H | T1CKI High Time | Synchronous, Prescaler = 1 | | 0.5TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | PIC16**C**7X | 15 | — | — | ns | |
| | | | | PIC16**LC**7X | 25 | — | — | ns | |
| | | | Asynchronous | PIC16**C**7X | 30 | — | — | ns | |
| | | | | PIC16**LC**7X | 50 | — | — | ns | |
| 46* | Tt1L | T1CKI Low Time | Synchronous, Prescaler = 1 | | 0.5TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | PIC16**C**7X | 15 | — | — | ns | |
| | | | | PIC16**LC**7X | 25 | — | — | ns | |
| | | | Asynchronous | PIC16**C**7X | 30 | — | — | ns | |
| | | | | PIC16**LC**7X | 50 | — | — | ns | |
| 47* | Tt1P | T1CKI input period | Synchronous | PIC16**C**7X | Greater of: 30 OR TCY + 40 / N | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | | PIC16**LC**7X | Greater of: 50 OR TCY + 40 / N | | | | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | PIC16**C**7X | 60 | — | — | ns | |
| | | | | PIC16**LC**7X | 100 | — | — | ns | |
| | Ft1 | Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN) | | | DC | — | 200 | kHz | |
| 48 | TCKEZtmr1 | Delay from external clock edge to timer increment | | | 2Tosc | — | 7Tosc | — | |

* 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.

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

**FIGURE 18-6: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)**




Note: Refer to Figure 18-1 for load conditions.

**TABLE 18-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)**

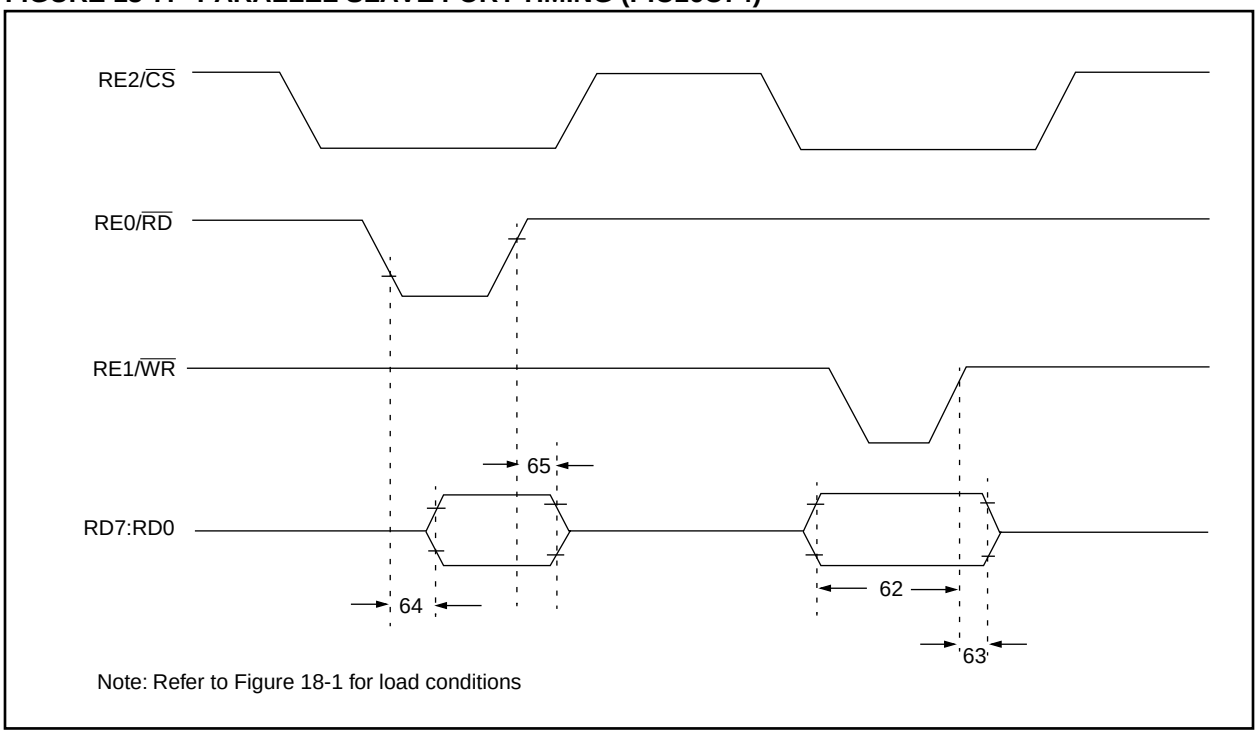
| Parameter No. | Sym | Characteristic | | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 50* | TccL | CCP1 and CCP2 input low time | No Prescaler | | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | PIC16**C**73/74 | 10 | — | — | ns | |
| | | | | PIC16**LC**73/74 | 20 | — | — | ns | |
| 51* | TccH | CCP1 and CCP2 input high time | No Prescaler | | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | PIC16**C**73/74 | 10 | — | — | ns | |
| | | | | PIC16**LC**73/74 | 20 | — | — | ns | |
| 52* | TccP | CCP1 and CCP2 input period | | | $\frac{3T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1,4 or 16) |
| 53* | TccR | CCP1 and CCP2 output fall time | | PIC16**C**73/74 | — | 10 | 25 | ns | |
| | | | | PIC16**LC**73/74 | — | 25 | 45 | ns | |
| 54* | TccF | CCP1 and CCP2 output fall time | | PIC16**C**73/74 | — | 10 | 25 | ns | |
| | | | | PIC16**LC**73/74 | — | 25 | 45 | 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.

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

**FIGURE 18-7: PARALLEL SLAVE PORT TIMING (PIC16C74)**




Note: Refer to Figure 18-1 for load conditions

**TABLE 18-7: PARALLEL SLAVE PORT REQUIREMENTS (PIC16C74)**

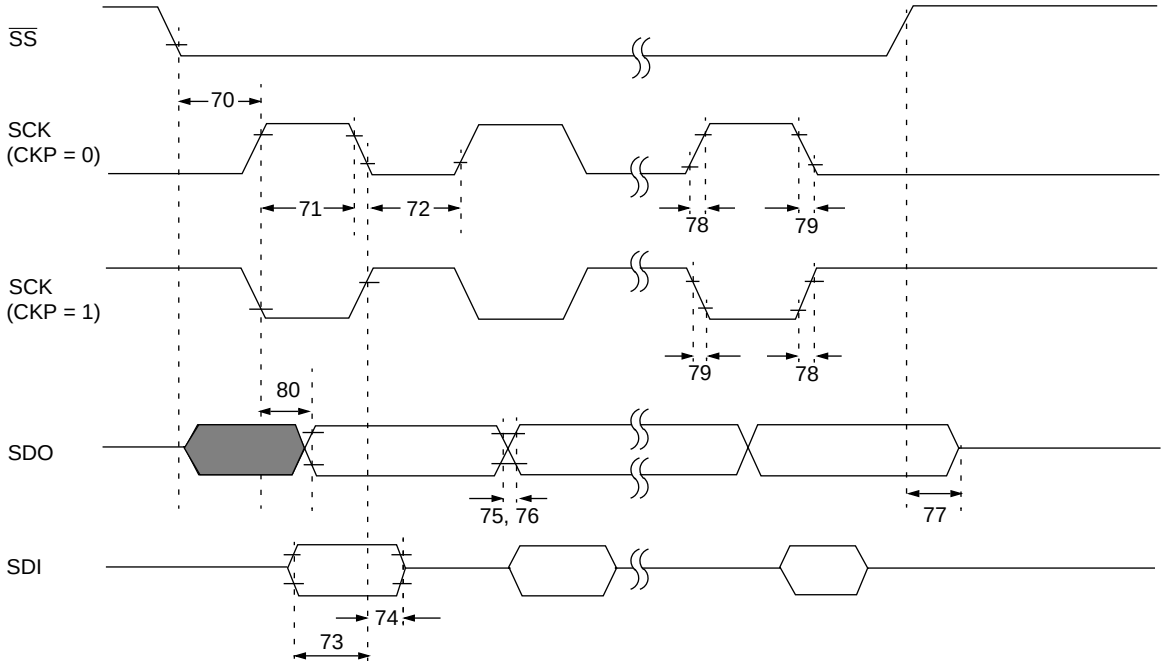
| Parameter No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 62 | TdtV2wrH | Data in valid before $\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ (setup time) | | 20 | — | — | ns | |
| 63* | TwrH2dtI | $\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ to data–in invalid (hold time) | PIC16**C**74 | 20 | — | — | ns | |
| | | | PIC16**LC**74 | 35 | — | — | ns | |
| 64 | TrdL2dtV | $\overline{RD}\downarrow$ and $\overline{CS}\downarrow$ to data–out valid | | — | — | 80 | ns | |
| 65 | TrdH2dtI | $\overline{RD}\uparrow$ or $\overline{CS}\downarrow$ to data–out invalid | | 10 | — | 30 | 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.

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

**FIGURE 18-8: SPI MODE TIMING**




Note: Refer to Figure 18-1 for load conditions

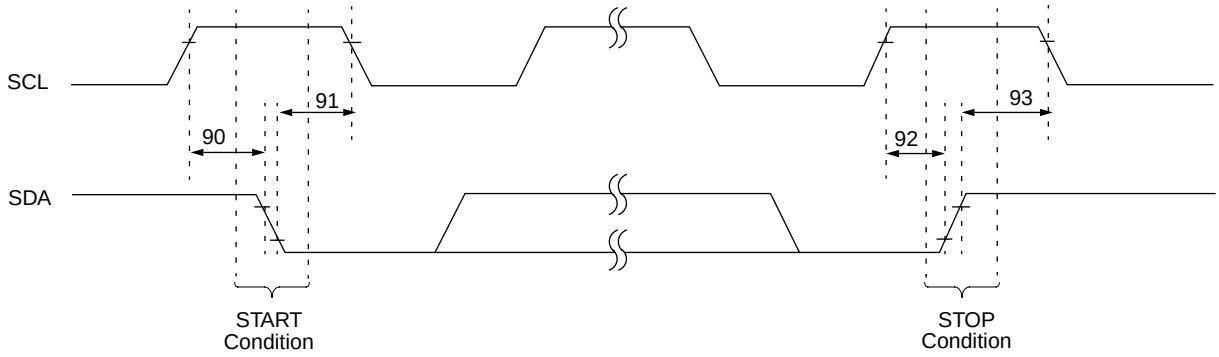
**TABLE 18-8: SPI MODE REQUIREMENTS**

| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 70 | TssL2scH, TssL2scL | $\overline{SS}\downarrow$ to SCK$\downarrow$ or SCK$\uparrow$ input | TCY | — | — | ns | |
| 71 | TscH | SCK input high time (slave mode) | TCY + 20 | — | — | ns | |
| 72 | TscL | SCK input low time (slave mode) | TCY + 20 | — | — | ns | |
| 73 | TdiV2scH, TdiV2scL | Setup time of SDI data input to SCK edge | 50 | — | — | ns | |
| 74 | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | 50 | — | — | ns | |
| 75 | TdoR | SDO data output rise time | — | 10 | 25 | ns | |
| 76 | TdoF | SDO data output fall time | — | 10 | 25 | ns | |
| 77 | TssH2doZ | $\overline{SS}\uparrow$ to SDO output hi-impedance | 10 | — | 50 | ns | |
| 78 | TscR | SCK output rise time (master mode) | — | 10 | 25 | ns | |
| 79 | TscF | SCK output fall time (master mode) | — | 10 | 25 | ns | |
| 80 | TscH2doV, TscL2doV | SDO data output valid after SCK edge | — | — | 50 | ns | |

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

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

**FIGURE 18-9: $I^2C$ BUS START/STOP BITS TIMING**



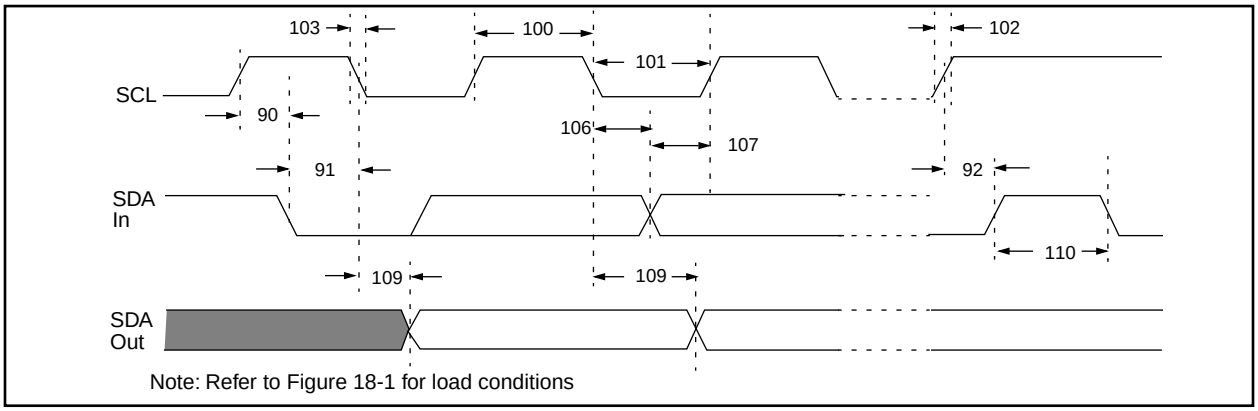

Note: Refer to Figure 18-1 for load conditions

**TABLE 18-9: $I^2C$ BUS START/STOP BITS REQUIREMENTS**

| Parameter No. | Sym | Characteristic | | Min | Typ | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 90 | TSU:STA | START condition Setup time | 100 kHz mode | 4700 | — | — | ns | Only relevant for repeated START condition |
| | | | 400 kHz mode | 600 | — | — | | |
| 91 | THD:STA | START condition Hold time | 100 kHz mode | 4000 | — | — | ns | After this period the first clock pulse is generated |
| | | | 400 kHz mode | 600 | — | — | | |
| 92 | TSU:STO | STOP condition Setup time | 100 kHz mode | 4700 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |
| 93 | THD:STO | STOP condition Hold time | 100 kHz mode | 4000 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |

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

**FIGURE 18-10: I²C BUS DATA TIMING**




Note: Refer to Figure 18-1 for load conditions

**TABLE 18-10: I²C BUS DATA REQUIREMENTS**

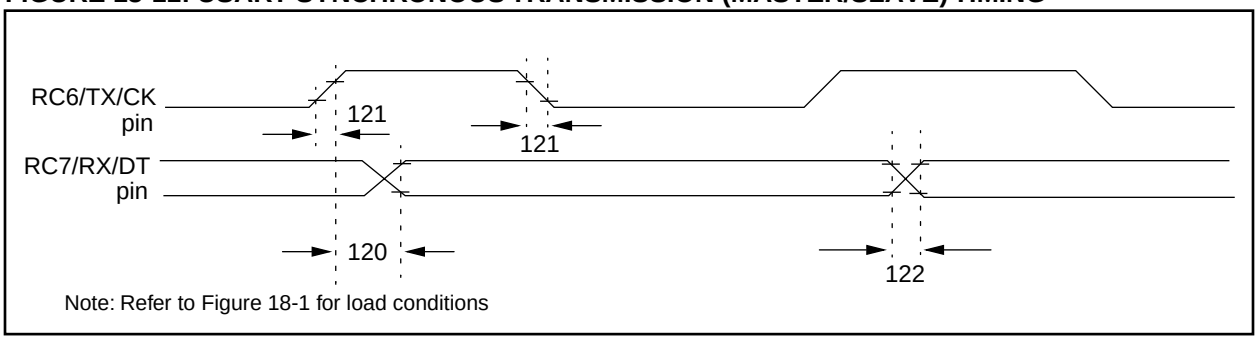
| Parameter No. | Sym | Characteristic | | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 100 | THIGH | Clock high time | 100 kHz mode | 4.0 | — | µs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 0.6 | — | µs | Device must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5TCY | — | | |
| 101 | TLOW | Clock low time | 100 kHz mode | 4.7 | — | µs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 1.3 | — | µs | Device must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5TCY | — | | |
| 102 | TR | SDA and SCL rise time | 100 kHz mode | — | 1000 | ns | |
| | | | 400 kHz mode | 20 + 0.1Cb | 300 | ns | Cb is specified to be from 10 to 400 pF |
| 103 | TF | SDA and SCL fall time | 100 kHz mode | — | 300 | ns | |
| | | | 400 kHz mode | 20 + 0.1Cb | 300 | ns | Cb is specified to be from 10 to 400 pF |
| 90 | TSU:STA | START condition setup time | 100 kHz mode | 4.7 | — | µs | Only relevant for repeated START condition |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 91 | THD:STA | START condition hold time | 100 kHz mode | 4.0 | — | µs | After this period the first clock pulse is generated |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 106 | THD:DAT | Data input hold time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | µs | |
| 107 | TSU:DAT | Data input setup time | 100 kHz mode | 250 | — | ns | Note 2 |
| | | | 400 kHz mode | 100 | — | ns | |
| 92 | TSU:STO | STOP condition setup time | 100 kHz mode | 4.7 | — | µs | |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 109 | TAA | Output valid from clock | 100 kHz mode | — | 3500 | ns | Note 1 |
| | | | 400 kHz mode | — | — | ns | |
| 110 | TBUF | Bus free time | 100 kHz mode | 4.7 | — | µs | Time the bus must be free before a new transmission can start |
| | | | 400 kHz mode | 1.3 | — | µs | |
| | Cb | Bus capacitive loading | | — | 400 | pF | |

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A fast-mode (400 kHz) I²C-bus device can be used in a standard-mode (100 kHz) I²C-bus system, but the requirement tsu;DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line TR max.+tsu;DAT = 1000 + 250 = 1250 ns (according to the standard-mode I²C bus specification) before the SCL line is released.

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

**FIGURE 18-11: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**



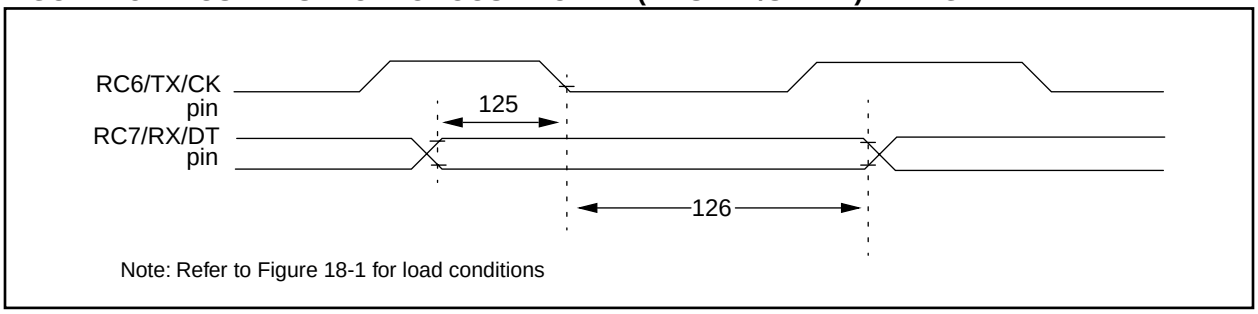

Note: Refer to Figure 18-1 for load conditions

**TABLE 18-11: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

| Parameter No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 120 | TckH2dtV | SYNC XMIT (MASTER & SLAVE) Clock high to data out valid | PIC16**C**73/74 | — | — | 80 | ns | |
| | | | PIC16**LC**73/74 | — | — | 100 | ns | |
| 121 | Tckrf | Clock out rise time and fall time (Master Mode) | PIC16**C**73/74 | — | — | 45 | ns | |
| | | | PIC16**LC**73/74 | — | — | 50 | ns | |
| 122 | Tdtrf | Data out rise time and fall time | PIC16**C**73/74 | — | — | 45 | ns | |
| | | | PIC16**LC**73/74 | — | — | 50 | ns | |

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

**FIGURE 18-12: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**




Note: Refer to Figure 18-1 for load conditions

**TABLE 18-12: USART SYNCHRONOUS RECEIVE REQUIREMENTS**

| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 125 | TdtV2ckL | SYNC RCV (MASTER & SLAVE) Data setup before CK ↓ (DT setup time) | 15 | — | — | ns | |
| 126 | TckL2dtl | Data hold after CK ↓ (DT hold time) | 15 | — | — | ns | |

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

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

**TABLE 18-13: A/D CONVERTER CHARACTERISTICS:**

**PIC16C73/74-04 (Commercial, Industrial)**
**PIC16C73/74-10 (Commercial, Industrial)**
**PIC16C73/74-20 (Commercial, Industrial)**
**PIC16LC73/74-04 (Commercial, Industrial)**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| A01 | NR | Resolution | | — | — | 8-bits | bit | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A02 | EABS | Total Absolute error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A03 | EIL | Integral linearity error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A04 | EDL | Differential linearity error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A05 | EFS | Full scale error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A06 | EOFF | Offset error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A10 | — | Monotonicity | | — | guaranteed | — | — | VSS ≤ VAIN ≤ VREF |
| A20 | VREF | Reference voltage | | 3.0V | — | VDD + 0.3 | V | |
| A25 | VAIN | Analog input voltage | | VSS - 0.3 | — | VREF + 0.3 | V | |
| A30 | ZAIN | Recommended impedance of analog voltage source | | — | — | 10.0 | kΩ | |
| A40 | IAD | A/D conversion current (VDD) | PIC16**C**73/74 | — | 180 | — | µA | Average current consumption when A/D is on. (Note 1) |
| | | | PIC16**LC**73/74 | — | 90 | — | µA | |
| A50 | IREF | VREF input current (Note 2) | | 10 | — | 1000 | µA | During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see Section 13.1. |
| | | | | — | — | 10 | µA | During A/D Conversion cycle |

* These parameters are characterized but not tested.

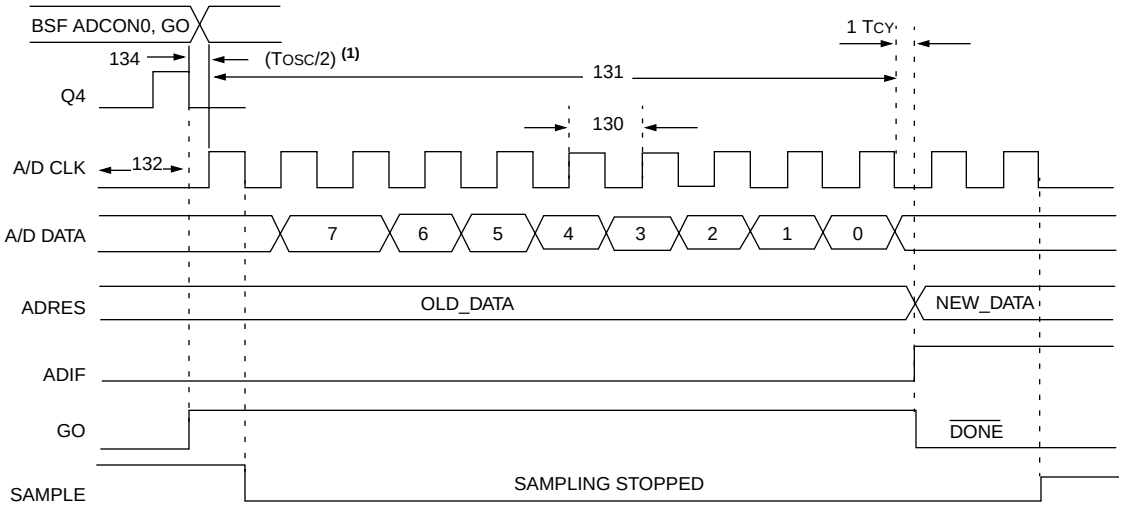
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

2: VREF current is from RA3 pin or VDD pin, whichever is selected as reference input.

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

**FIGURE 18-13: 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 18-14: A/D CONVERSION REQUIREMENTS**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 130 | TAD | A/D clock period | PIC16**C**73/74 | 1.6 | — | — | µs | TOSC based, VREF ≥ 3.0V |
| | | | PIC16**LC**73/74 | 2.0 | — | — | µs | TOSC based, VREF full range |
| | | | PIC16**C**73/74 | 2.0 | 4.0 | 6.0 | µs | A/D RC Mode |
| | | | PIC16**LC**73/74 | 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<br><br>5* | 20<br><br>— | —<br><br>— | µs<br><br>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., 20 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.

§ This specification ensured by design.

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

2: See Section 13.1 for min conditions.

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

## 19.0 ELECTRICAL CHARACTERISTICS FOR PIC16C73A/74A

**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) ..... 1.0W

Maximum current out of $V_{SS}$ pin ..... 300 mA

Maximum current into $V_{DD}$ pin ..... 250 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 ..... 25 mA

Maximum current sunk by PORTA, PORTB, and PORTE (combined) (Note 3) ..... 200 mA

Maximum current sourced by PORTA, PORTB, and PORTE (combined) (Note 3) ..... 200 mA

Maximum current sunk by PORTC and PORTD (combined) (Note 3) ..... 200 mA

Maximum current sourced by PORTC and PORTD (combined) (Note 3) ..... 200 mA

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

**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 $V_{SS}$.

**Note 3:** PORTD and PORTE are not implemented on the PIC16C73A.

† 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 19-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)**

| OSC | PIC16C73A-04<br>PIC16C74A-04 | PIC16C73A-10<br>PIC16C74A-10 | PIC16C73A-20<br>PIC16C74A-20 | PIC16LC73A-04<br>PIC16LC74A-04 | JW Devices |
|---|---|---|---|---|---|
| RC | $V_{DD}$: 4.0V to 6.0V<br>$I_{DD}$: 5 mA max. at 5.5V<br>$I_{PD}$: 16 µA max. at 4V<br>Freq: 4 MHz max. | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 2.7 mA typ. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 2.7 mA typ. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | $V_{DD}$: 2.5V to 6.0V<br>$I_{DD}$: 3.8 mA max. at 3.0V<br>$I_{PD}$: 5 µA max. at 3V<br>Freq: 4 MHz max. | $V_{DD}$: 4.0V to 6.0V<br>$I_{DD}$: 5 mA max. at 5.5V<br>$I_{PD}$: 16 µA max. at 4V<br>Freq: 4 MHz max. |
| XT | $V_{DD}$: 4.0V to 6.0V<br>$I_{DD}$: 5 mA max. at 5.5V<br>$I_{PD}$: 16 µA max. at 4V<br>Freq: 4 MHz max. | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 2.7 mA typ. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 2.7 mA typ. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | $V_{DD}$: 2.5V to 6.0V<br>$I_{DD}$: 3.8 mA max. at 3.0V<br>$I_{PD}$: 5 µA max. at 3V<br>Freq: 4 MHz max. | $V_{DD}$: 4.0V to 6.0V<br>$I_{DD}$: 5 mA max. at 5.5V<br>$I_{PD}$: 16 µA max. at 4V<br>Freq: 4 MHz max. |
| HS | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 13.5 mA typ. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4.5V<br>Freq: 4 MHz max. | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 10 mA max. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4.5V<br>Freq: 10 MHz max. | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 20 mA max. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4.5V<br>Freq: 20 MHz max. | Not recommended for use in HS mode | $V_{DD}$: 4.5V to 5.5V<br>$I_{DD}$: 20 mA max. at 5.5V<br>$I_{PD}$: 1.5 µA typ. at 4.5V<br>Freq: 20 MHz max. |
| LP | $V_{DD}$: 4.0V to 6.0V<br>$I_{DD}$: 52.5 µA typ. at 32 kHz, 4.0V<br>$I_{PD}$: 0.9 µA typ. at 4.0V<br>Freq: 200 kHz max. | Not recommended for use in LP mode | Not recommended for use in LP mode | $V_{DD}$: 2.5V to 6.0V<br>$I_{DD}$: 48 µA max. at 32 kHz, 3.0V<br>$I_{PD}$: 5.0 µA max. at 3.0V<br>Freq: 200 kHz max. | $V_{DD}$: 2.5V to 6.0V<br>$I_{DD}$: 48 µA max. at 32 kHz, 3.0V<br>$I_{PD}$: 5.0 µA max. at 3.0V<br>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 | 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>Brown-out Reset Current (Note 6) | IPD<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 |
|---|---|---|---|---|---|---|---|

## 19.2 DC Characteristics: PIC16LC73A/74A-04 (Commercial, Industrial)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -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 | Supply Voltage | VDD | 2.5 | - | 6.0 | V | LP, XT, RC osc configuration (DC - 4 MHz) |
| 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 | 4.0 | 4.3 | V | BODEN bit in configuration word enabled |
| D010 | Supply Current (Note 2,5) | IDD | - | 2.0 | 3.8 | mA | XT, RC osc configuration<br>FOSC = 4 MHz, VDD = 3.0V (Note 4) |
| D010A | | | - | 22.5 | 48 | µA | LP osc configuration<br>FOSC = 32 kHz, VDD = 3.0V, WDT disabled |
| D015* | Brown-out Reset Current (Note 6) | ΔIBOR | - | 350 | 425 | µA | BOR enabled VDD = 5.0V |
| D020 | Power-down Current | IPD | - | 7.5 | 30 | µA | VDD = 3.0V, WDT enabled, -40°C to +85°C |
| D021 | (Note 3,5) | | - | 0.9 | 5 | µA | VDD = 3.0V, WDT disabled, 0°C to +70°C |
| D021A | | | - | 0.9 | 5 | µA | VDD = 3.0V, WDT disabled, -40°C to +85°C |
| D023* | Brown-out Reset Current (Note 6) | ΔIBOR | - | 350 | 425 | 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 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 |
|---|---|---|---|---|---|---|---|

### 19.3 DC Characteristics: PIC16C73A/74A-04 (Commercial, Industrial, Extended)
### PIC16C73A/74A-10 (Commercial, Industrial, Extended)
### PIC16C73A/74A-20 (Commercial, Industrial, Extended)
### PIC16LC73A/74A-04 (Commercial, Industrial)

| 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<br>Operating voltage VDD range as described in DC spec Section 19.1 and Section 19.2. |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Characteristic** | **Sym** | **Min** | **Typ †** | **Max** | **Units** | **Conditions** |
| | **Input Low Voltage** | | | | | | |
| | I/O ports | VIL | | | | | |
| D030 | with TTL buffer | | VSS | - | 0.15VDD | V | For entire VDD range |
| D030A | | | VSS | - | 0.8V | | 4.5V ≤ VDD ≤ 5.5V |
| D031 | with Schmitt Trigger buffer | | VSS | - | 0.2VDD | V | |
| D032 | $\overline{MCLR}$, OSC1 (in RC mode) | | VSS | - | 0.2VDD | V | |
| D033 | OSC1 (in XT, HS and LP) | | VSS | - | 0.3VDD | V | Note1 |
| | **Input High Voltage** | | | | | | |
| | I/O ports | VIH | | - | | | |
| D040 | with TTL buffer | | 2.0 | - | VDD | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | 0.25VDD + 0.8V | - | VDD | V | For entire VDD range |
| D041 | with Schmitt Trigger buffer | | 0.8VDD | - | VDD | V | For entire VDD range |
| D042 | $\overline{MCLR}$ | | 0.8VDD | - | VDD | V | |
| D042A | OSC1 (XT, HS and LP) | | 0.7VDD | - | VDD | V | Note1 |
| D043 | OSC1 (in RC mode) | | 0.9VDD | - | VDD | V | |
| D070 | PORTB weak pull-up current | IPURB | 50 | 250 | 400 | µA | VDD = 5V, VPIN = VSS |
| | **Input Leakage Current** (Notes 2, 3) | | | | | | |
| D060 | I/O ports | IIL | - | - | ±1 | µA | Vss ≤ VPIN ≤ VDD, Pin at hi-impedance |
| D061 | $\overline{MCLR}$, RA4/T0CKI | | - | - | ±5 | µA | Vss ≤ VPIN ≤ VDD |
| D063 | OSC1 | | - | - | ±5 | µA | Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration |
| | **Output Low Voltage** | | | | | | |
| D080 | I/O ports | VOL | - | - | 0.6 | V | IOL = 8.5 mA, VDD = 4.5V, -40°C to +85°C |
| D080A | | | - | - | 0.6 | V | IOL = 7.0 mA, VDD = 4.5V, -40°C to +125°C |
| D083 | OSC2/CLKOUT (RC osc config) | | - | - | 0.6 | V | IOL = 1.6 mA, VDD = 4.5V, -40°C to +85°C |
| D083A | | | - | - | 0.6 | V | IOL = 1.2 mA, VDD = 4.5V, -40°C to +125°C |

* 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.

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

**DC CHARACTERISTICS**

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40˚C ≤ TA ≤ +125˚C for extended,
-40˚C ≤ TA ≤ +85˚C for industrial and
0˚C ≤ TA ≤ +70˚C for commercial
Operating voltage VDD range as described in DC spec Section 19.1 and Section 19.2.

| 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 |
| D150* | **Open-Drain High Voltage** | VOD | - | - | 14 | V | RA4 pin |
| | **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 | CIO | - | - | 50 | pF | |
| D102 | mode) SCL, SDA in $I^2C$ mode | CB | - | - | 400 | 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.

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

### 19.4 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 | $\overline{WR}$ |

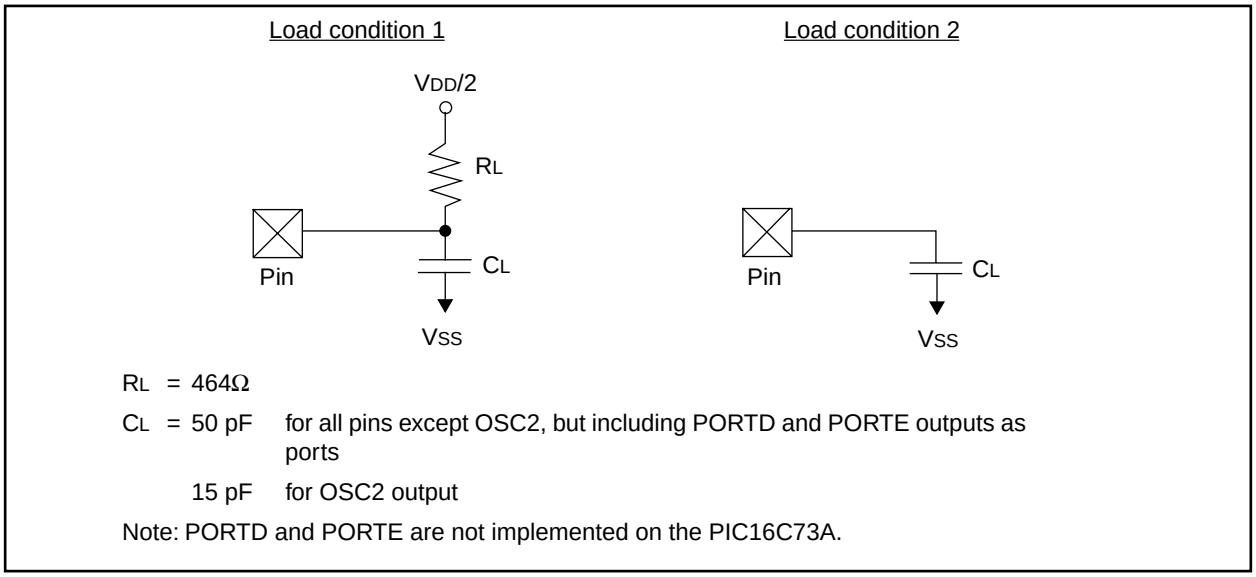
Uppercase letters and their meanings:

| | | | |
|---|---|---|---|
| **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 19-1: 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 PIC16C73A.

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

## 19.5 Timing Diagrams and Specifications

**FIGURE 19-2: EXTERNAL CLOCK TIMING**

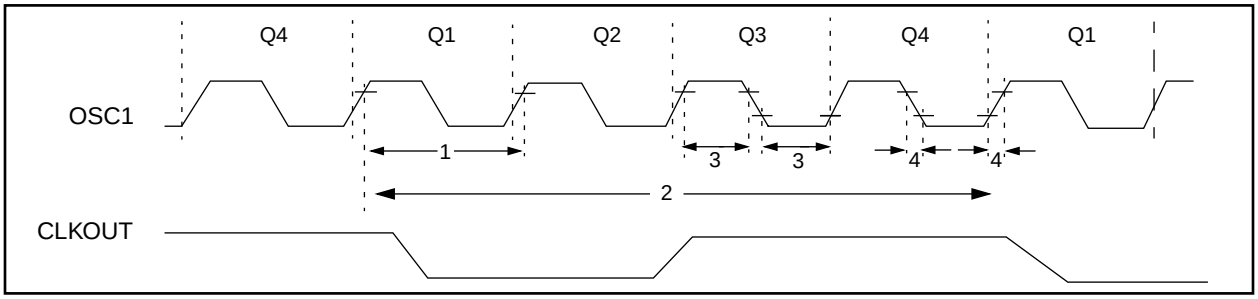



**TABLE 19-2: EXTERNAL CLOCK TIMING REQUIREMENTS**

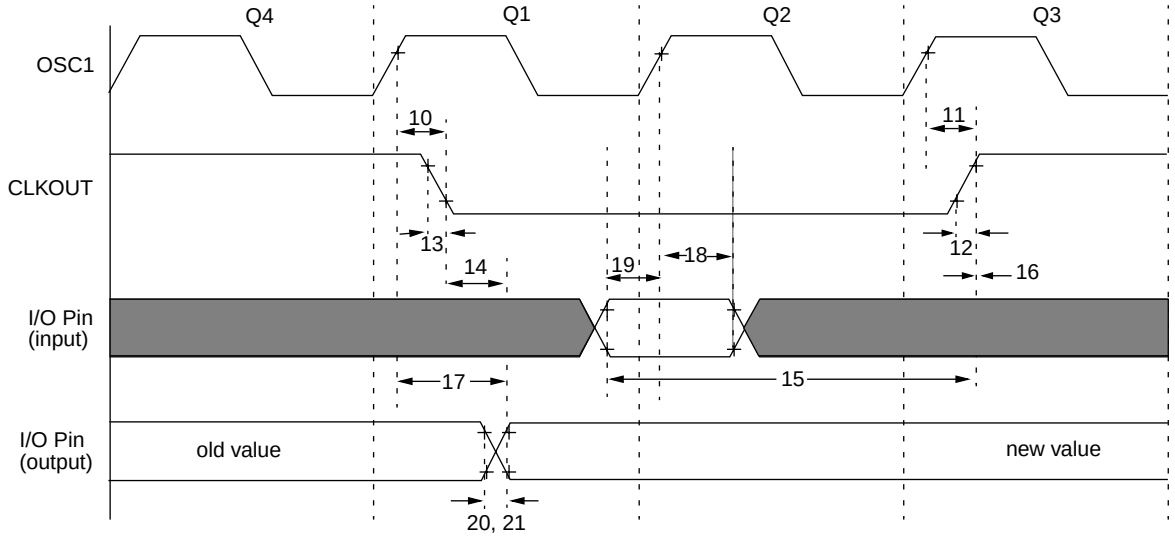
| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | Fosc | **External CLKIN Frequency (Note 1)** | DC | — | 4 | MHz | XT and RC osc mode |
| | | | DC | — | 4 | MHz | HS osc mode (-04) |
| | | | DC | — | 10 | MHz | HS osc mode (-10) |
| | | | DC | — | 20 | MHz | HS osc mode (-20) |
| | | | DC | — | 200 | kHz | LP osc mode |
| | | **Oscillator Frequency (Note 1)** | DC | — | 4 | MHz | RC osc mode |
| | | | 0.1 | — | 4 | MHz | XT osc mode |
| | | | 4 | — | 20 | MHz | HS osc mode |
| | | | 5 | — | 200 | kHz | LP osc mode |
| 1 | Tosc | **External CLKIN Period (Note 1)** | 250 | — | — | ns | XT and RC osc mode |
| | | | 250 | — | — | ns | HS osc mode (-04) |
| | | | 100 | — | — | ns | HS osc mode (-10) |
| | | | 50 | — | — | ns | HS osc mode (-20) |
| | | | 5 | — | — | µs | LP osc mode |
| | | **Oscillator Period (Note 1)** | 250 | — | — | ns | RC osc mode |
| | | | 250 | — | 10,000 | ns | XT osc mode |
| | | | 250 | — | 250 | ns | HS osc mode (-04) |
| | | | 100 | — | 250 | ns | HS osc mode (-10) |
| | | | 50 | — | 250 | ns | HS osc mode (-20) |
| | | | 5 | — | — | µs | LP osc mode |
| 2 | TCY | **Instruction Cycle Time (Note 1)** | 200 | TCY | DC | ns | TCY = 4/FOSC |
| 3 | TosL, TosH | **External Clock in (OSC1) High or Low Time** | 100 | — | — | ns | XT oscillator |
| | | | 2.5 | — | — | µs | LP oscillator |
| | | | 15 | — | — | ns | HS oscillator |
| 4 | TosR, TosF | **External Clock in (OSC1) Rise or Fall Time** | — | — | 25 | ns | XT oscillator |
| | | | — | — | 50 | ns | LP oscillator |
| | | | — | — | 15 | ns | HS oscillator |

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

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

**FIGURE 19-3: CLKOUT AND I/O TIMING**




Note: Refer to Figure 19-1 for load conditions.

**TABLE 19-3: CLKOUT AND I/O TIMING REQUIREMENTS**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 10* | TosH2ckL | OSC1↑ to CLKOUT↓ | | — | 75 | 200 | ns | Note 1 |
| 11* | TosH2ckH | OSC1↑ to CLKOUT↑ | | — | 75 | 200 | ns | Note 1 |
| 12* | TckR | CLKOUT rise time | | — | 35 | 100 | ns | Note 1 |
| 13* | TckF | CLKOUT fall time | | — | 35 | 100 | ns | Note 1 |
| 14* | TckL2ioV | CLKOUT ↓ to Port out valid | | — | — | 0.5TCY + 20 | ns | Note 1 |
| 15* | TioV2ckH | Port in valid before CLKOUT ↑ | | TOSC + 200 | — | — | ns | Note 1 |
| 16* | TckH2ioI | Port in hold after CLKOUT ↑ | | 0 | — | — | ns | Note 1 |
| 17* | TosH2ioV | OSC1↑ (Q1 cycle) to Port out valid | | — | 50 | 150 | ns | |
| 18* | TosH2ioI | OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time) | PIC16**C**73A/74A | 100 | — | — | ns | |
| | | | PIC16**LC**73A/74A | 200 | — | — | ns | |
| 19* | TioV2osH | Port input valid to OSC1↑ (I/O in setup time) | | 0 | — | — | ns | |
| 20* | TioR | Port output rise time | PIC16**C**73A/74A | — | 10 | 40 | ns | |
| | | | PIC16**LC**73A/74A | — | — | 80 | ns | |
| 21* | TioF | Port output fall time | PIC16**C**73A/74A | — | 10 | 40 | ns | |
| | | | PIC16**LC**73A/74A | — | — | 80 | ns | |
| 22††* | Tinp | INT pin high or low time | | TCY | — | — | ns | |
| 23††* | Trbp | RB7:RB4 change INT high or low time | | TCY | — | — | ns | |

* These parameters are characterized but not tested.

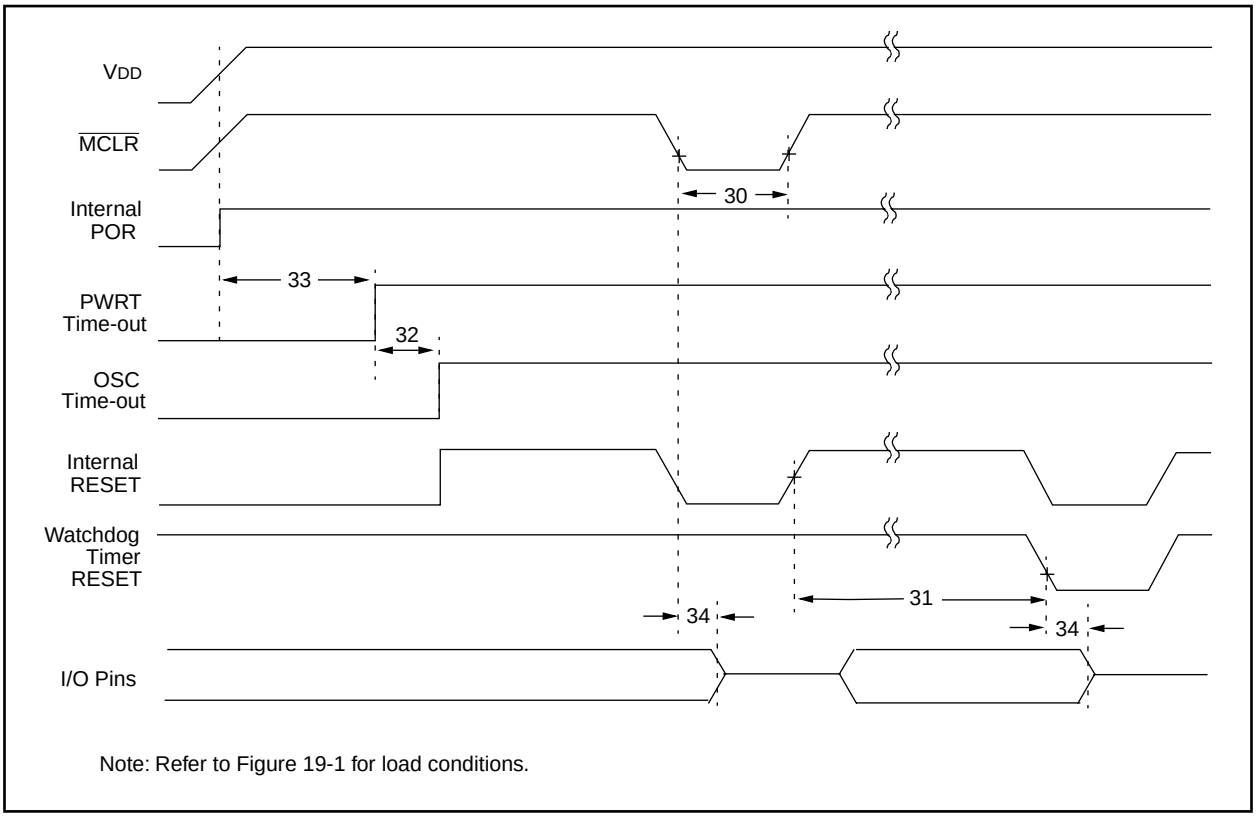
†Data in "Typ" column is at 5V, 25˚C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† 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 TOSC.

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

**FIGURE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**




Note: Refer to Figure 19-1 for load conditions.

**FIGURE 19-5: BROWN-OUT RESET TIMING**

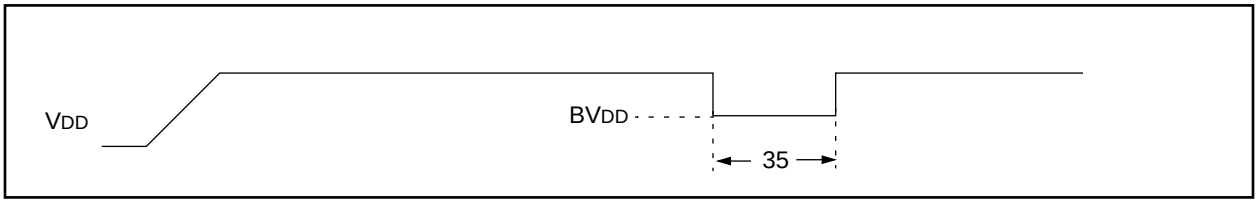



**TABLE 19-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS**

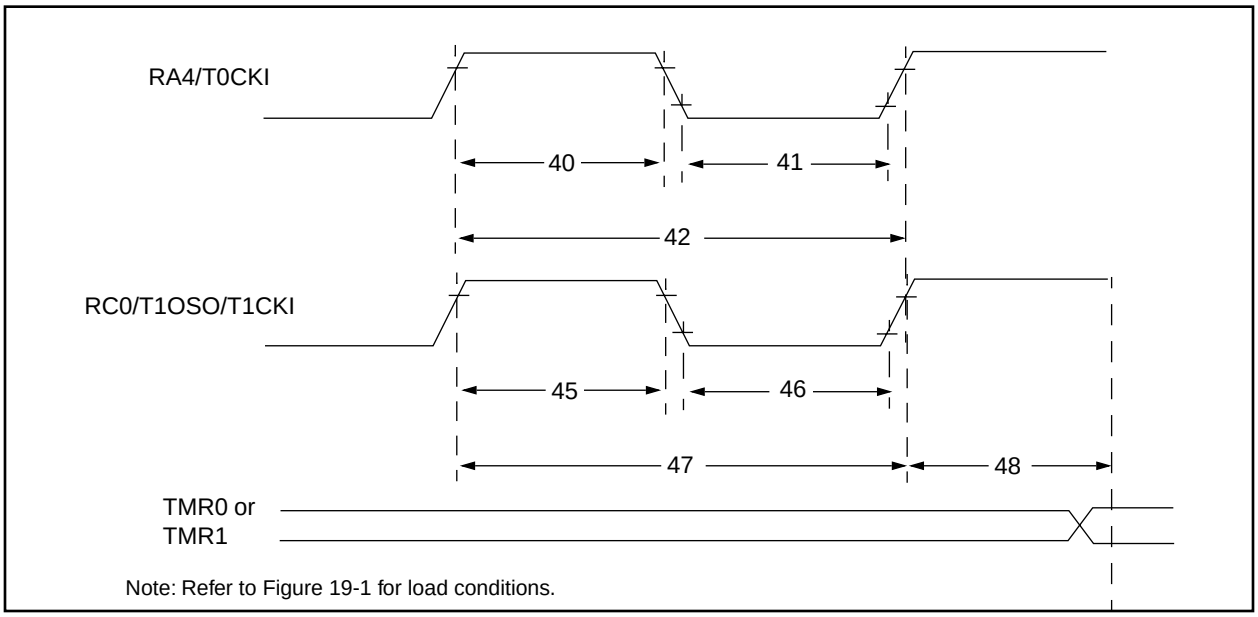
| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 30 | TmcL | $\overline{MCLR}$ Pulse Width (low) | 2 | — | — | µs | VDD = 5V, -40˚C to +125˚C |
| 31* | Twdt | Watchdog Timer Time-out Period (No Prescaler) | 7 | 18 | 33 | ms | VDD = 5V, -40˚C to +125˚C |
| 32 | Tost | Oscillation Start-up Timer Period | — | 1024TOSC | — | — | TOSC = OSC1 period |
| 33* | Tpwrt | Power up Timer Period | 28 | 72 | 132 | ms | VDD = 5V, -40˚C to +125˚C |
| 34 | TIOZ | I/O Hi-impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset | — | — | 2.1 | µs | |
| 35 | TBOR | Brown-out Reset pulse width | 100 | — | — | µs | VDD ≤ BVDD (D005) |

* 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.

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

**FIGURE 19-6: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**




Note: Refer to Figure 19-1 for load conditions.

**TABLE 19-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

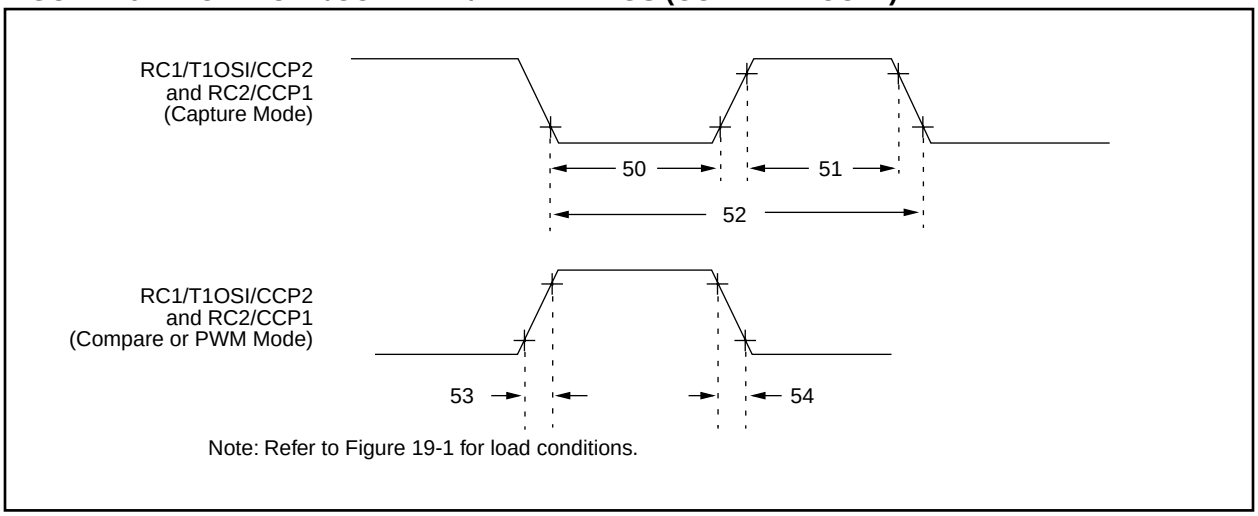
| Param No. | Sym | Characteristic | | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 40* | Tt0H | T0CKI High Pulse Width | | No Prescaler | 0.5TCY + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With Prescaler | 10 | — | — | ns | |
| 41* | Tt0L | T0CKI Low Pulse Width | | No Prescaler | 0.5TCY + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With Prescaler | 10 | — | — | ns | |
| 42* | Tt0P | T0CKI Period | | No Prescaler | TCY + 40 | — | — | ns | |
| | | | | With Prescaler | Greater of: 20 or TCY + 40 / N | — | — | ns | N = prescale value (2, 4, ..., 256) |
| 45* | Tt1H | T1CKI High Time | Synchronous, Prescaler = 1 | | 0.5TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | PIC16**C**7X | 15 | — | — | ns | |
| | | | | PIC16**LC**7X | 25 | — | — | ns | |
| | | | Asynchronous | PIC16**C**7X | 30 | — | — | ns | |
| | | | | PIC16**LC**7X | 50 | — | — | ns | |
| 46* | Tt1L | T1CKI Low Time | Synchronous, Prescaler = 1 | | 0.5TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | PIC16**C**7X | 15 | — | — | ns | |
| | | | | PIC16**LC**7X | 25 | — | — | ns | |
| | | | Asynchronous | PIC16**C**7X | 30 | — | — | ns | |
| | | | | PIC16**LC**7X | 50 | — | — | ns | |
| 47* | Tt1P | T1CKI input period | Synchronous | PIC16**C**7X | Greater of: 30 OR TCY + 40 / N | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | | PIC16**LC**7X | Greater of: 50 OR TCY + 40 / N | | | | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | PIC16**C**7X | 60 | — | — | ns | |
| | | | | PIC16**LC**7X | 100 | — | — | ns | |
| | Ft1 | Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN) | | | DC | — | 200 | kHz | |
| 48 | TCKEZtmr1 | Delay from external clock edge to timer increment | | | 2Tosc | — | 7Tosc | — | |

* 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.

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

**FIGURE 19-7: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)**




Note: Refer to Figure 19-1 for load conditions.

**TABLE 19-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)**

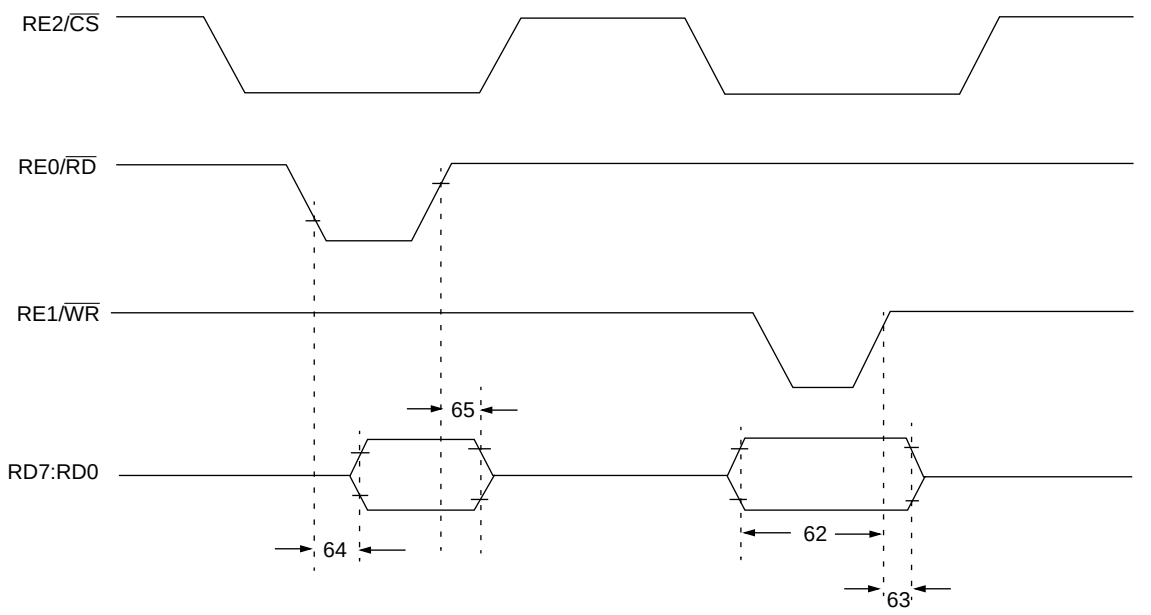
| Param No. | Sym | Characteristic | | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 50* | TccL | CCP1 and CCP2 input low time | No Prescaler | | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | PIC16**C**73A/74A | 10 | — | — | ns | |
| | | | | PIC16**LC**73A/74A | 20 | — | — | ns | |
| 51* | TccH | CCP1 and CCP2 input high time | No Prescaler | | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | PIC16**C**73A/74A | 10 | — | — | ns | |
| | | | | PIC16**LC**73A/74A | 20 | — | — | ns | |
| 52* | TccP | CCP1 and CCP2 input period | | | $\frac{3T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1,4 or 16) |
| 53* | TccR | CCP1 and CCP2 output rise time | | PIC16**C**73A/74A | — | 10 | 25 | ns | |
| | | | | PIC16**LC**73A/74A | — | 25 | 45 | ns | |
| 54* | TccF | CCP1 and CCP2 output fall time | | PIC16**C**73A/74A | — | 10 | 25 | ns | |
| | | | | PIC16**LC**73A/74A | — | 25 | 45 | 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.

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

**FIGURE 19-8: PARALLEL SLAVE PORT TIMING (PIC16C74A)**




Note: Refer to Figure 19-1 for load conditions

**TABLE 19-7: PARALLEL SLAVE PORT REQUIREMENTS (PIC16C74A)**

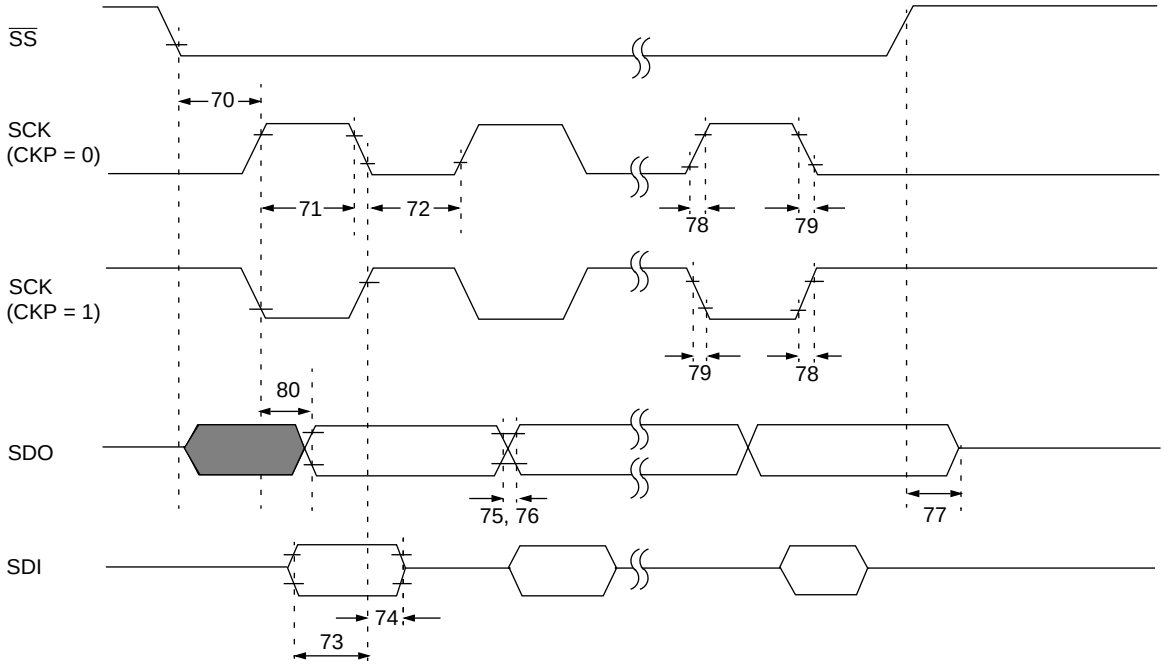
| Parameter No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 62 | TdtV2wrH | Data in valid before WR↑ or CS↑ (setup time) | | 20<br>25 | —<br>— | —<br>— | ns<br>ns | <br>Extended Range Only |
| 63* | TwrH2dtI | WR↑ or CS↑ to data–in invalid (hold time) | PIC16**C**74A | 20 | — | — | ns | |
| | | | PIC16**LC**74A | 35 | — | — | ns | |
| 64 | TrdL2dtV | RD↓ and CS↓ to data–out valid | | —<br>— | —<br>— | 80<br>90 | ns<br>ns | <br>Extended Range Only |
| 65 | TrdH2dtI | RD↑ or CS↓ to data–out invalid | | 10 | — | 30 | 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.

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

**FIGURE 19-9: SPI MODE TIMING**




Note: Refer to Figure 19-1 for load conditions

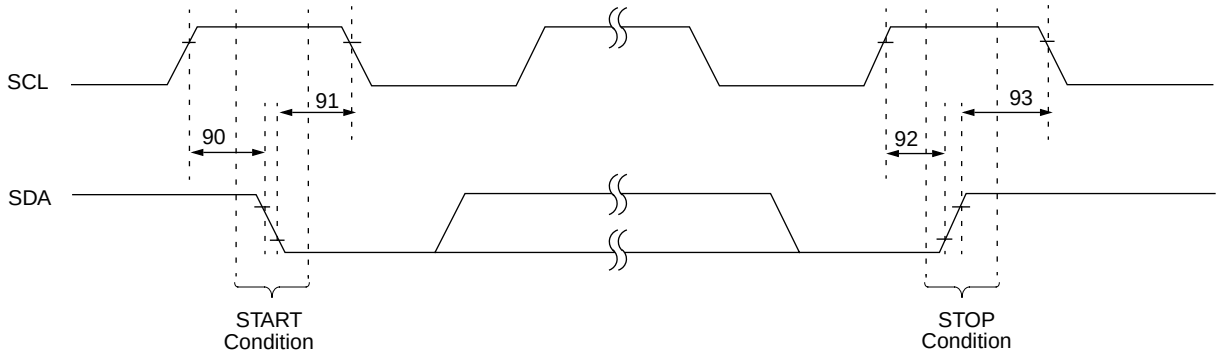
**TABLE 19-8: SPI MODE REQUIREMENTS**

| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 70 | TssL2scH, TssL2scL | $\overline{SS}\downarrow$ to SCK$\downarrow$ or SCK$\uparrow$ input | TCY | — | — | ns | |
| 71 | TscH | SCK input high time (slave mode) | TCY + 20 | — | — | ns | |
| 72 | TscL | SCK input low time (slave mode) | TCY + 20 | — | — | ns | |
| 73 | TdiV2scH, TdiV2scL | Setup time of SDI data input to SCK edge | 100 | — | — | ns | |
| 74 | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | 100 | — | — | ns | |
| 75 | TdoR | SDO data output rise time | — | 10 | 25 | ns | |
| 76 | TdoF | SDO data output fall time | — | 10 | 25 | ns | |
| 77 | TssH2doZ | $\overline{SS}\uparrow$ to SDO output hi-impedance | 10 | — | 50 | ns | |
| 78 | TscR | SCK output rise time (master mode) | — | 10 | 25 | ns | |
| 79 | TscF | SCK output fall time (master mode) | — | 10 | 25 | ns | |
| 80 | TscH2doV, TscL2doV | SDO data output valid after SCK edge | — | — | 50 | ns | |

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

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

**FIGURE 19-10: $I^2C$ BUS START/STOP BITS TIMING**



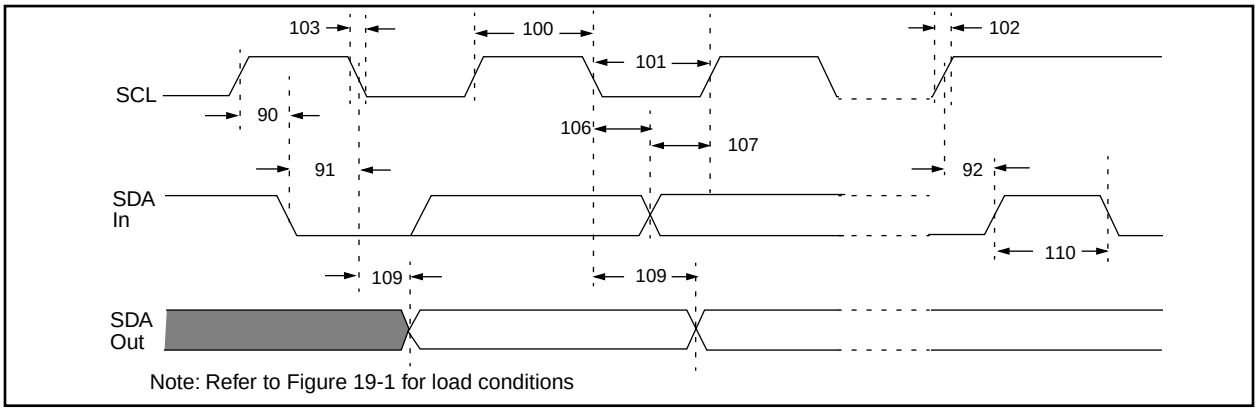

Note: Refer to Figure 19-1 for load conditions

**TABLE 19-9: $I^2C$ BUS START/STOP BITS REQUIREMENTS**

| Parameter No. | Sym | Characteristic | | Min | Typ | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 90 | TSU:STA | START condition Setup time | 100 kHz mode | 4700 | — | — | ns | Only relevant for repeated START condition |
| | | | 400 kHz mode | 600 | — | — | | |
| 91 | THD:STA | START condition Hold time | 100 kHz mode | 4000 | — | — | ns | After this period the first clock pulse is generated |
| | | | 400 kHz mode | 600 | — | — | | |
| 92 | TSU:STO | STOP condition Setup time | 100 kHz mode | 4700 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |
| 93 | THD:STO | STOP condition Hold time | 100 kHz mode | 4000 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |

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

**FIGURE 19-11: I²C BUS DATA TIMING**




Note: Refer to Figure 19-1 for load conditions

**TABLE 19-10: I²C BUS DATA REQUIREMENTS**

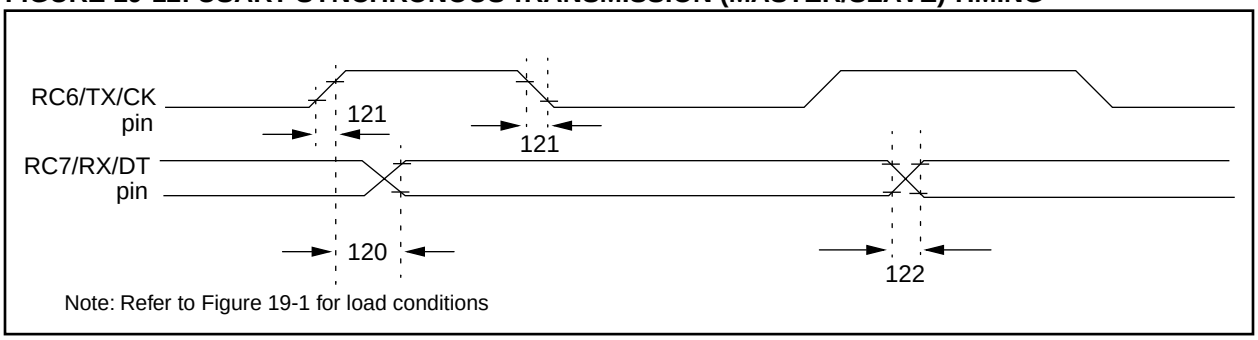
| Parameter No. | Sym | Characteristic | | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 100 | THIGH | Clock high time | 100 kHz mode | 4.0 | — | µs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 0.6 | — | µs | Device must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5TCY | — | | |
| 101 | TLOW | Clock low time | 100 kHz mode | 4.7 | — | µs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 1.3 | — | µs | Device must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5TCY | — | | |
| 102 | TR | SDA and SCL rise time | 100 kHz mode | — | 1000 | ns | |
| | | | 400 kHz mode | 20 + 0.1Cb | 300 | ns | Cb is specified to be from 10 to 400 pF |
| 103 | TF | SDA and SCL fall time | 100 kHz mode | — | 300 | ns | |
| | | | 400 kHz mode | 20 + 0.1Cb | 300 | ns | Cb is specified to be from 10 to 400 pF |
| 90 | TSU:STA | START condition setup time | 100 kHz mode | 4.7 | — | µs | Only relevant for repeated START condition |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 91 | THD:STA | START condition hold time | 100 kHz mode | 4.0 | — | µs | After this period the first clock pulse is generated |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 106 | THD:DAT | Data input hold time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | µs | |
| 107 | TSU:DAT | Data input setup time | 100 kHz mode | 250 | — | ns | Note 2 |
| | | | 400 kHz mode | 100 | — | ns | |
| 92 | TSU:STO | STOP condition setup time | 100 kHz mode | 4.7 | — | µs | |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 109 | TAA | Output valid from clock | 100 kHz mode | — | 3500 | ns | Note 1 |
| | | | 400 kHz mode | — | — | ns | |
| 110 | TBUF | Bus free time | 100 kHz mode | 4.7 | — | µs | Time the bus must be free before a new transmission can start |
| | | | 400 kHz mode | 1.3 | — | µs | |
| | Cb | Bus capacitive loading | | — | 400 | pF | |

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A fast-mode (400 kHz) I²C-bus device can be used in a standard-mode (100 kHz) I²C-bus system, but the requirement tsu;DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line TR max.+tsu;DAT = 1000 + 250 = 1250 ns (according to the standard-mode I²C bus specification) before the SCL line is released.

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

**FIGURE 19-12: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**



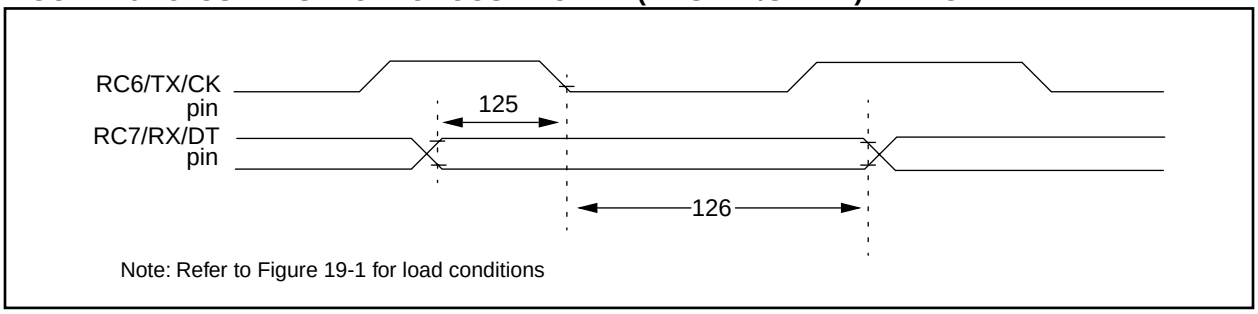

Note: Refer to Figure 19-1 for load conditions

**TABLE 19-11: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 120 | TckH2dtV | SYNC XMIT (MASTER & SLAVE) Clock high to data out valid | PIC16**C**73A/74A | — | — | 80 | ns | |
| | | | PIC16**LC**73A/74A | — | — | 100 | ns | |
| 121 | Tckrf | Clock out rise time and fall time (Master Mode) | PIC16**C**73A/74A | — | — | 45 | ns | |
| | | | PIC16**LC**73A/74A | — | — | 50 | ns | |
| 122 | Tdtrf | Data out rise time and fall time | PIC16**C**73A/74A | — | — | 45 | ns | |
| | | | PIC16**LC**73A/74A | — | — | 50 | ns | |

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

**FIGURE 19-13: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**




Note: Refer to Figure 19-1 for load conditions

**TABLE 19-12: USART SYNCHRONOUS RECEIVE REQUIREMENTS**

| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 125 | TdtV2ckL | SYNC RCV (MASTER & SLAVE) Data setup before CK ↓ (DT setup time) | 15 | — | — | ns | |
| 126 | TckL2dtl | Data hold after CK ↓ (DT hold time) | 15 | — | — | ns | |

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

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

**TABLE 19-13: A/D CONVERTER CHARACTERISTICS:**

**PIC16C73A/74A-04 (Commercial, Industrial, Extended)**
**PIC16C73A/74A-10 (Commercial, Industrial, Extended)**
**PIC16C73A/74A-20 (Commercial, Industrial, Extended)**
**PIC16LC73A/74A-04 (Commercial, Industrial)**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| A01 | NR | Resolution | | — | — | 8-bits | bit | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A02 | EABS | Total Absolute error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A03 | EIL | Integral linearity error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A04 | EDL | Differential linearity error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A05 | EFS | Full scale error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A06 | EOFF | Offset error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A10 | — | Monotonicity | | — | guaranteed | — | — | VSS ≤ VAIN ≤ VREF |
| A20 | VREF | Reference voltage | | 3.0V | — | VDD + 0.3 | V | |
| A25 | VAIN | Analog input voltage | | VSS - 0.3 | — | VREF + 0.3 | V | |
| A30 | ZAIN | Recommended impedance of analog voltage source | | — | — | 10.0 | kΩ | |
| A40 | IAD | A/D conversion current (VDD) | PIC16**C**73A/74A | — | 180 | — | µA | Average current consumption when A/D is on. (Note 1) |
| | | | PIC16**LC**73A/74A | — | 90 | — | µA | |
| A50 | IREF | VREF input current (Note 2) | | 10 | — | 1000 | µA | During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see Section 13.1. |
| | | | | — | — | 10 | µA | During A/D Conversion cycle |

* These parameters are characterized but not tested.

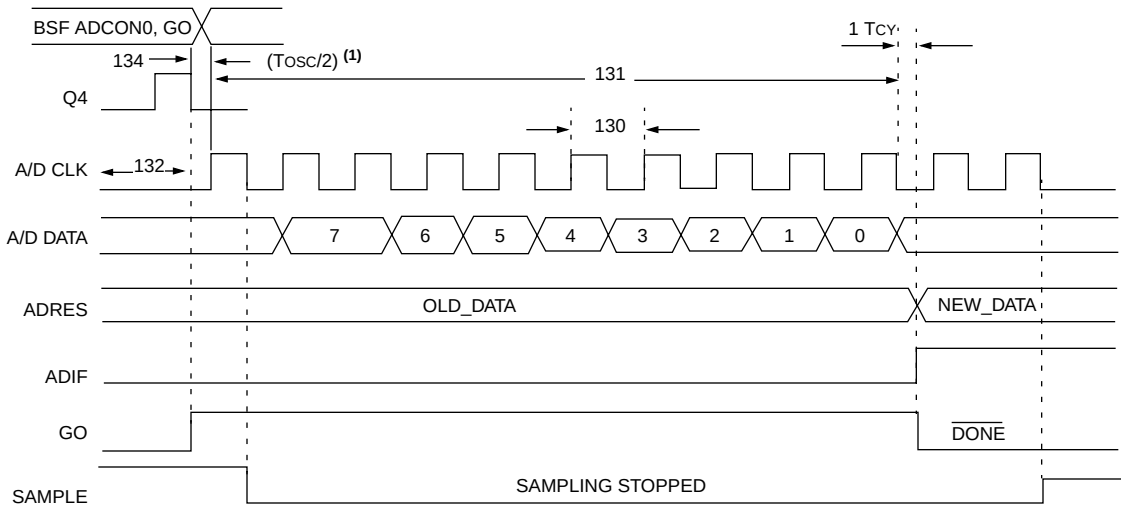
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

2: VREF current is from RA3 pin or VDD pin, whichever is selected as reference input.

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

**FIGURE 19-14: 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 19-14: A/D CONVERSION REQUIREMENTS**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 130 | TAD | A/D clock period | PIC16**C**73A/74A | 1.6 | — | — | µs | TOSC based, VREF ≥ 3.0V |
| | | | PIC16**LC**73A/74A | 2.0 | — | — | µs | TOSC based, VREF full range |
| | | | PIC16**C**73A/74A | 2.0 | 4.0 | 6.0 | µs | A/D RC Mode |
| | | | PIC16**LC**73A/74A | 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., 20.0 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.

§ This specification ensured by design.

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

2: See Section 13.1 for min conditions.

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

## 20.0 ELECTRICAL CHARACTERISTICS FOR PIC16C76/77

**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) ..... 1.0W

Maximum current out of $V_{SS}$ pin ..... 300 mA

Maximum current into $V_{DD}$ pin ..... 250 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 ..... 25 mA

Maximum current sunk by PORTA, PORTB, and PORTE (combined) (Note 3) ..... 200 mA

Maximum current sourced by PORTA, PORTB, and PORTE (combined) (Note 3) ..... 200 mA

Maximum current sunk by PORTC and PORTD (combined) (Note 3) ..... 200 mA

Maximum current sourced by PORTC and PORTD (combined) (Note 3) ..... 200 mA

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

**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 $V_{SS}$.

**Note 3:** PORTD and PORTE are not implemented on the PIC16C76.

† 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.

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

**TABLE 20-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)**

| OSC | PIC16C76-04<br>PIC16C77-04 | PIC16C76-10<br>PIC16C77-10 | PIC16C76-20<br>PIC16C77-20 | PIC16LC76-04<br>PIC16LC77-04 | JW Devices |
|---|---|---|---|---|---|
| RC | VDD: 4.0V to 6.0V<br>IDD: 5 mA max. at 5.5V<br>IPD: 16 µA max. at 4V<br>Freq: 4 MHz max. | VDD: 4.5V to 5.5V<br>IDD: 2.7 mA typ. at 5.5V<br>IPD: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | VDD: 4.5V to 5.5V<br>IDD: 2.7 mA typ. at 5.5V<br>IPD: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | VDD: 2.5V to 6.0V<br>IDD: 3.8 mA max. at 3.0V<br>IPD: 5 µA max. at 3V<br>Freq: 4 MHz max. | VDD: 4.0V to 6.0V<br>IDD: 5 mA max. at 5.5V<br>IPD: 16 µA max. at 4V<br>Freq: 4 MHz max. |
| XT | VDD: 4.0V to 6.0V<br>IDD: 5 mA max. at 5.5V<br>IPD: 16 µA max. at 4V<br>Freq: 4 MHz max. | VDD: 4.5V to 5.5V<br>IDD: 2.7 mA typ. at 5.5V<br>IPD: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | VDD: 4.5V to 5.5V<br>IDD: 2.7 mA typ. at 5.5V<br>IPD: 1.5 µA typ. at 4V<br>Freq: 4 MHz max. | VDD: 2.5V to 6.0V<br>IDD: 3.8 mA max. at 3.0V<br>IPD: 5 µA max. at 3V<br>Freq: 4 MHz max. | VDD: 4.0V to 6.0V<br>IDD: 5 mA max. at 5.5V<br>IPD: 16 µA max. at 4V<br>Freq: 4 MHz max. |
| HS | VDD: 4.5V to 5.5V<br>IDD: 13.5 mA typ. at 5.5V<br>IPD: 1.5 µA typ. at 4.5V<br>Freq: 4 MHz max. | VDD: 4.5V to 5.5V<br>IDD: 10 mA max. at 5.5V<br>IPD: 1.5 µA typ. at 4.5V<br>Freq: 10 MHz max. | VDD: 4.5V to 5.5V<br>IDD: 20 mA max. at 5.5V<br>IPD: 1.5 µA typ. at 4.5V<br>Freq: 20 MHz max. | Not recommended for use in HS mode | VDD: 4.5V to 5.5V<br>IDD: 20 mA max. at 5.5V<br>IPD: 1.5 µA typ. at 4.5V<br>Freq: 20 MHz max. |
| LP | VDD: 4.0V to 6.0V<br>IDD: 52.5 µA typ. at 32 kHz, 4.0V<br>IPD: 0.9 µA typ. at 4.0V<br>Freq: 200 kHz max. | Not recommended for use in LP mode | Not recommended for use in LP mode | VDD: 2.5V to 6.0V<br>IDD: 48 µA max. at 32 kHz, 3.0V<br>IPD: 5.0 µA max. at 3.0V<br>Freq: 200 kHz max. | VDD: 2.5V to 6.0V<br>IDD: 48 µA max. at 32 kHz, 3.0V<br>IPD: 5.0 µA max. at 3.0V<br>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 | 72 | 73 | 73A | 74 | 74A | 76 | 77 |
|---|---|---|---|---|---|---|---|

## 20.1 DC Characteristics: PIC16C76/77-04 (Commercial, Industrial, Extended) PIC16C76/77-10 (Commercial, Industrial, Extended) PIC16C76/77-20 (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 | Supply Voltage | VDD | 4.0 | - | 6.0 | V | XT, RC and LP osc configuration |
| D001A | | | 4.5 | - | 5.5 | V | 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 | 4.0 | 4.3 | V | BODEN bit in configuration word enabled |
| | | | 3.7 | 4.0 | 4.4 | V | Extended Range Only |
| D010 | Supply Current (Note 2,5) | IDD | - | 2.7 | 5 | mA | XT, RC osc configuration<br>FOSC = 4 MHz, VDD = 5.5V (Note 4) |
| D013 | | | - | 10 | 20 | mA | HS osc configuration<br>FOSC = 20 MHz, VDD = 5.5V |
| D015* | Brown-out Reset Current (Note 6) | ΔIBOR | - | 350 | 425 | µA | BOR enabled VDD = 5.0V |
| D020 | Power-down Current | IPD | - | 10.5 | 42 | µA | VDD = 4.0V, WDT enabled, -40°C to +85°C |
| D021 | (Note 3,5) | | - | 1.5 | 16 | µA | VDD = 4.0V, WDT disabled, -0°C to +70°C |
| D021A | | | - | 1.5 | 19 | µA | VDD = 4.0V, WDT disabled, -40°C to +85°C |
| D021B | | | - | 2.5 | 19 | µA | VDD = 4.0V, WDT disabled, -40°C to +125°C |
| D023* | Brown-out Reset Current (Note 6) | ΔIBOR | - | 350 | 425 | 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 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 |
|---|---|---|---|---|---|---|---|

### 20.2 DC Characteristics: PIC16LC76/77-04 (Commercial, Industrial)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -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 | Supply Voltage | VDD | 2.5 | - | 6.0 | V | LP, XT, RC osc configuration (DC - 4 MHz) |
| 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 | 4.0 | 4.3 | V | BODEN bit in configuration word enabled |
| D010 | Supply Current (Note 2,5) | IDD | - | 2.0 | 3.8 | mA | XT, RC osc configuration<br>FOSC = 4 MHz, VDD = 3.0V (Note 4) |
| D010A | | | - | 22.5 | 48 | µA | LP osc configuration<br>FOSC = 32 kHz, VDD = 3.0V, WDT disabled |
| D015* | Brown-out Reset Current (Note 6) | ΔIBOR | - | 350 | 425 | µA | BOR enabled VDD = 5.0V |
| D020 | Power-down Current | IPD | - | 7.5 | 30 | µA | VDD = 3.0V, WDT enabled, -40°C to +85°C |
| D021 | (Note 3,5) | | - | 0.9 | 5 | µA | VDD = 3.0V, WDT disabled, 0°C to +70°C |
| D021A | | | - | 0.9 | 5 | µA | VDD = 3.0V, WDT disabled, -40°C to +85°C |
| D023* | Brown-out Reset Current (Note 6) | ΔIBOR | - | 350 | 425 | 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 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 |
|---|---|---|---|---|---|---|---|

## 20.3 DC Characteristics: PIC16C76/77-04 (Commercial, Industrial, Extended) PIC16C76/77-10 (Commercial, Industrial, Extended) PIC16C76/77-20 (Commercial, Industrial, Extended) PIC16LC76/77-04 (Commercial, Industrial)

| 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<br>Operating voltage VDD range as described in DC spec Section 20.1 and Section 20.2. |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Characteristic** | **Sym** | **Min** | **Typ †** | **Max** | **Units** | **Conditions** |
| | **Input Low Voltage** | | | | | | |
| | I/O ports | VIL | | | | | |
| D030 | with TTL buffer | | VSS | - | 0.15VDD | V | For entire VDD range |
| D030A | | | VSS | - | 0.8V | V | 4.5V ≤ VDD ≤ 5.5V |
| D031 | with Schmitt Trigger buffer | | VSS | - | 0.2VDD | V | |
| D032 | $\overline{MCLR}$, OSC1 (in RC mode) | | VSS | - | 0.2VDD | V | |
| D033 | OSC1 (in XT, HS and LP) | | VSS | - | 0.3VDD | V | Note1 |
| | **Input High Voltage** | | | | | | |
| | I/O ports | VIH | | - | | | |
| D040 | with TTL buffer | | 2.0 | - | VDD | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | 0.25VDD + 0.8V | - | VDD | V | For entire VDD range |
| D041 | with Schmitt Trigger buffer | | 0.8VDD | - | VDD | V | For entire VDD range |
| D042 | $\overline{MCLR}$ | | 0.8VDD | - | VDD | V | |
| D042A | OSC1 (XT, HS and LP) | | 0.7VDD | - | VDD | V | Note1 |
| D043 | OSC1 (in RC mode) | | 0.9VDD | - | VDD | V | |
| D070 | PORTB weak pull-up current | IPURB | 50 | 250 | 400 | µA | VDD = 5V, VPIN = VSS |
| | **Input Leakage Current** (Notes 2, 3) | | | | | | |
| D060 | I/O ports | IIL | - | - | ±1 | µA | Vss ≤ VPIN ≤ VDD, Pin at hi-impedance |
| D061 | $\overline{MCLR}$, RA4/T0CKI | | - | - | ±5 | µA | Vss ≤ VPIN ≤ VDD |
| D063 | OSC1 | | - | - | ±5 | µA | Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration |
| | **Output Low Voltage** | | | | | | |
| D080 | I/O ports | VOL | - | - | 0.6 | V | IOL = 8.5 mA, VDD = 4.5V, -40°C to +85°C |
| D080A | | | - | - | 0.6 | V | IOL = 7.0 mA, VDD = 4.5V, -40°C to +125°C |
| D083 | OSC2/CLKOUT (RC osc config) | | - | - | 0.6 | V | IOL = 1.6 mA, VDD = 4.5V, -40°C to +85°C |
| D083A | | | - | - | 0.6 | V | IOL = 1.2 mA, VDD = 4.5V, -40°C to +125°C |

* 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.

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

**DC CHARACTERISTICS**

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40˚C ≤ TA ≤ +125˚C for extended,
-40˚C ≤ TA ≤ +85˚C for industrial and
0˚C ≤ TA ≤ +70˚C for commercial
Operating voltage VDD range as described in DC spec Section 20.1 and Section 20.2.

| 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 |
| D150* | **Open-Drain High Voltage** | VOD | - | - | 14 | V | RA4 pin |
| | **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 | CIO | - | - | 50 | pF | |
| D102 | mode) SCL, SDA in $I^2C$ mode | CB | - | - | 400 | 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.

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

## 20.4 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 | $\overline{WR}$ |

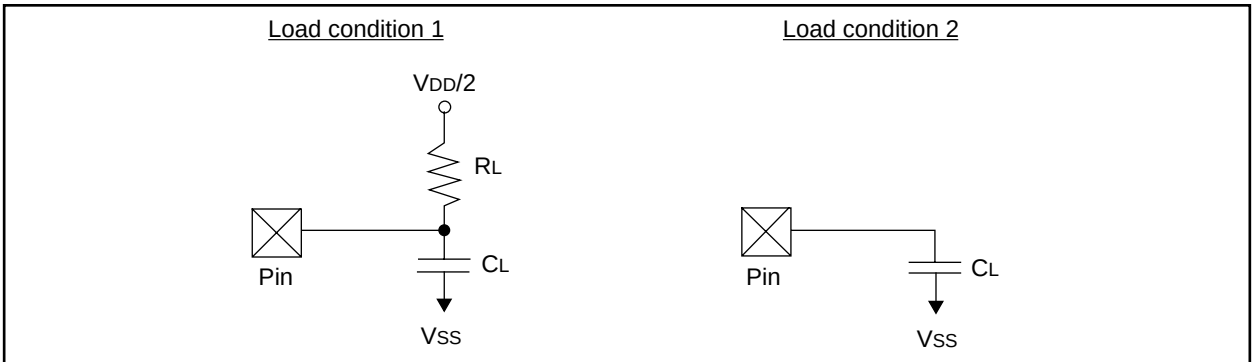
Uppercase letters and their meanings:

| | | | |
|---|---|---|---|
| **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 20-1: 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 PIC16C76.

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

## 20.5 Timing Diagrams and Specifications

**FIGURE 20-2: EXTERNAL CLOCK TIMING**

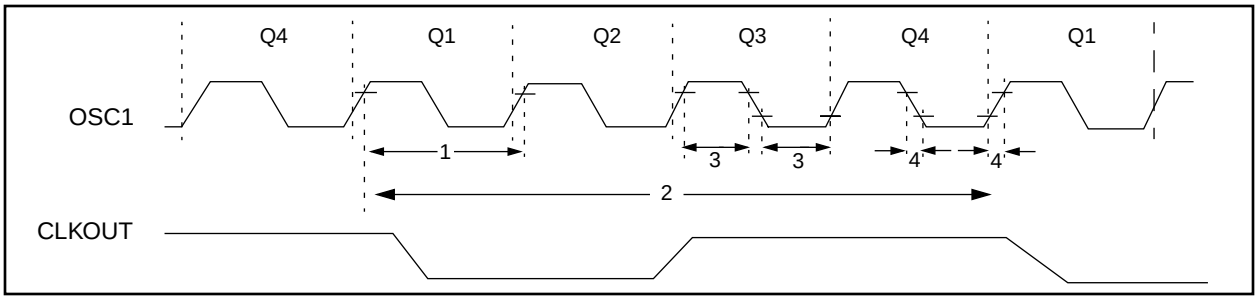



**TABLE 20-2: EXTERNAL CLOCK TIMING REQUIREMENTS**

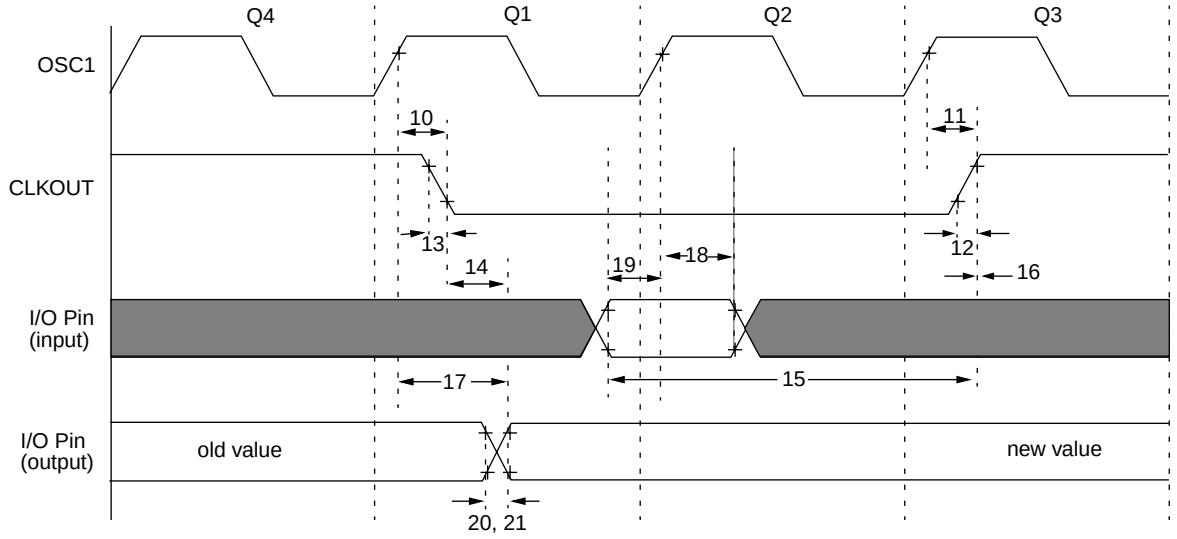
| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | Fosc | **External CLKIN Frequency (Note 1)** | DC | — | 4 | MHz | XT and RC osc mode |
| | | | DC | — | 4 | MHz | HS osc mode (-04) |
| | | | DC | — | 10 | MHz | HS osc mode (-10) |
| | | | DC | — | 20 | MHz | HS osc mode (-20) |
| | | | DC | — | 200 | kHz | LP osc mode |
| | | **Oscillator Frequency (Note 1)** | DC | — | 4 | MHz | RC osc mode |
| | | | 0.1 | — | 4 | MHz | XT osc mode |
| | | | 4 | — | 20 | MHz | HS osc mode |
| | | | 5 | — | 200 | kHz | LP osc mode |
| 1 | Tosc | **External CLKIN Period (Note 1)** | 250 | — | — | ns | XT and RC osc mode |
| | | | 250 | — | — | ns | HS osc mode (-04) |
| | | | 100 | — | — | ns | HS osc mode (-10) |
| | | | 50 | — | — | ns | HS osc mode (-20) |
| | | | 5 | — | — | µs | LP osc mode |
| | | **Oscillator Period (Note 1)** | 250 | — | — | ns | RC osc mode |
| | | | 250 | — | 10,000 | ns | XT osc mode |
| | | | 250 | — | 250 | ns | HS osc mode (-04) |
| | | | 100 | — | 250 | ns | HS osc mode (-10) |
| | | | 50 | — | 250 | ns | HS osc mode (-20) |
| | | | 5 | — | — | µs | LP osc mode |
| 2 | TCY | **Instruction Cycle Time (Note 1)** | 200 | TCY | DC | ns | TCY = 4/FOSC |
| 3 | TosL, TosH | **External Clock in (OSC1) High or Low Time** | 100 | — | — | ns | XT oscillator |
| | | | 2.5 | — | — | µs | LP oscillator |
| | | | 15 | — | — | ns | HS oscillator |
| 4 | TosR, TosF | **External Clock in (OSC1) Rise or Fall Time** | — | — | 25 | ns | XT oscillator |
| | | | — | — | 50 | ns | LP oscillator |
| | | | — | — | 15 | ns | HS oscillator |

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

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

**FIGURE 20-3: CLKOUT AND I/O TIMING**




Note: Refer to Figure 20-1 for load conditions.

**TABLE 20-3: CLKOUT AND I/O TIMING REQUIREMENTS**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 10* | TosH2ckL | OSC1↑ to CLKOUT↓ | | — | 75 | 200 | ns | Note 1 |
| 11* | TosH2ckH | OSC1↑ to CLKOUT↑ | | — | 75 | 200 | ns | Note 1 |
| 12* | TckR | CLKOUT rise time | | — | 35 | 100 | ns | Note 1 |
| 13* | TckF | CLKOUT fall time | | — | 35 | 100 | ns | Note 1 |
| 14* | TckL2ioV | CLKOUT ↓ to Port out valid | | — | — | 0.5TCY + 20 | ns | Note 1 |
| 15* | TioV2ckH | Port in valid before CLKOUT ↑ | | TOSC + 200 | — | — | ns | Note 1 |
| 16* | TckH2ioI | Port in hold after CLKOUT ↑ | | 0 | — | — | ns | Note 1 |
| 17* | TosH2ioV | OSC1↑ (Q1 cycle) to Port out valid | | — | 50 | 150 | ns | |
| 18* | TosH2ioI | OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time) | PIC16**C**76/77 | 100 | — | — | ns | |
| | | | PIC16**LC**76/77 | 200 | — | — | ns | |
| 19* | TioV2osH | Port input valid to OSC1↑ (I/O in setup time) | | 0 | — | — | ns | |
| 20* | TioR | Port output rise time | PIC16**C**76/77 | — | 10 | 40 | ns | |
| | | | PIC16**LC**76/77 | — | — | 80 | ns | |
| 21* | TioF | Port output fall time | PIC16**C**76/77 | — | 10 | 40 | ns | |
| | | | PIC16**LC**76/77 | — | — | 80 | ns | |
| 22††* | Tinp | INT pin high or low time | | TCY | — | — | ns | |
| 23††* | Trbp | RB7:RB4 change INT high or low time | | TCY | — | — | ns | |

* These parameters are characterized but not tested.

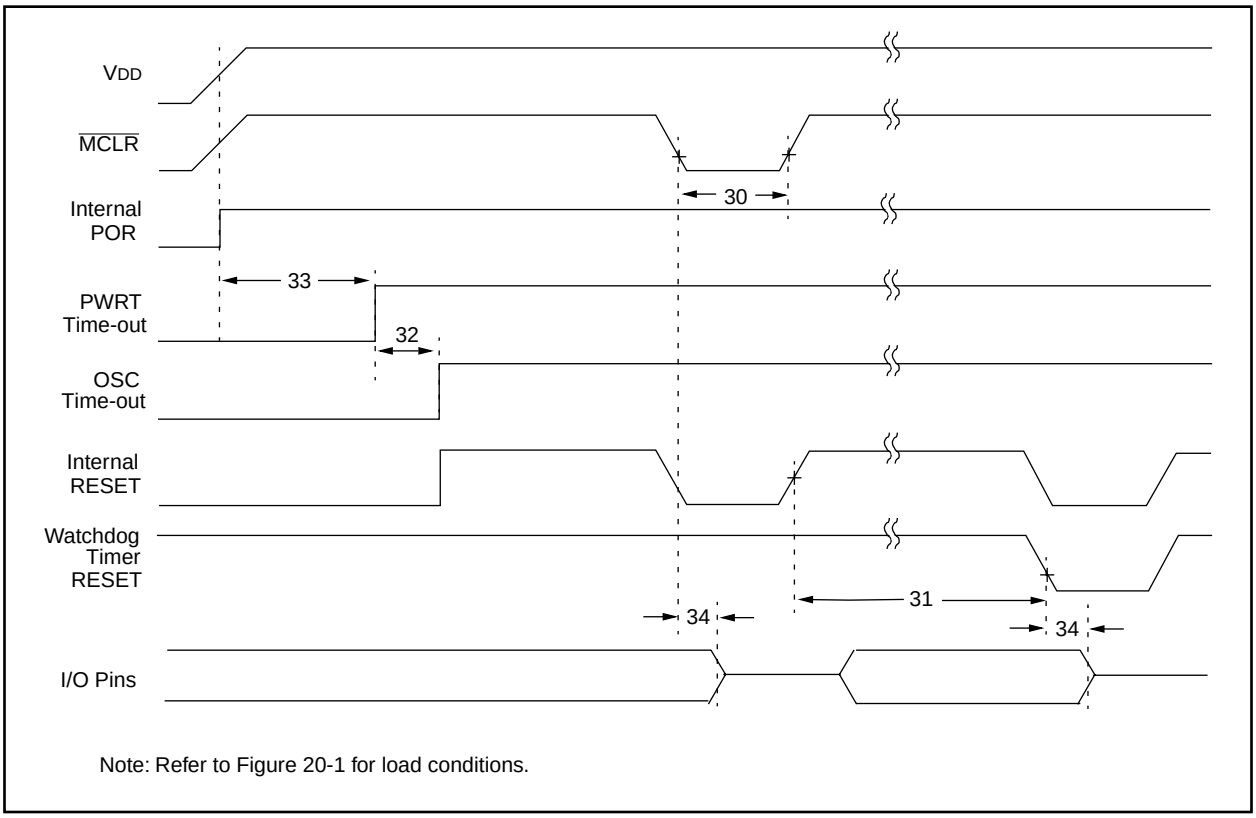
† Data in "Typ" column is at 5V, 25˚C unless otherwise stated. These parameters are for design guidance only and are not tested.

†† 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 TOSC.

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

**FIGURE 20-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**




Note: Refer to Figure 20-1 for load conditions.

**FIGURE 20-5: BROWN-OUT RESET TIMING**

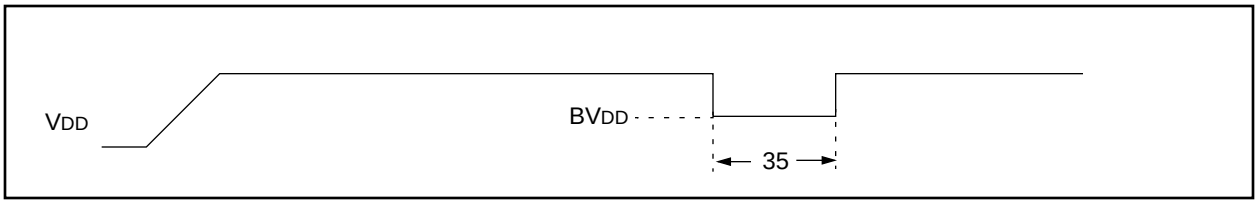



**TABLE 20-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS**

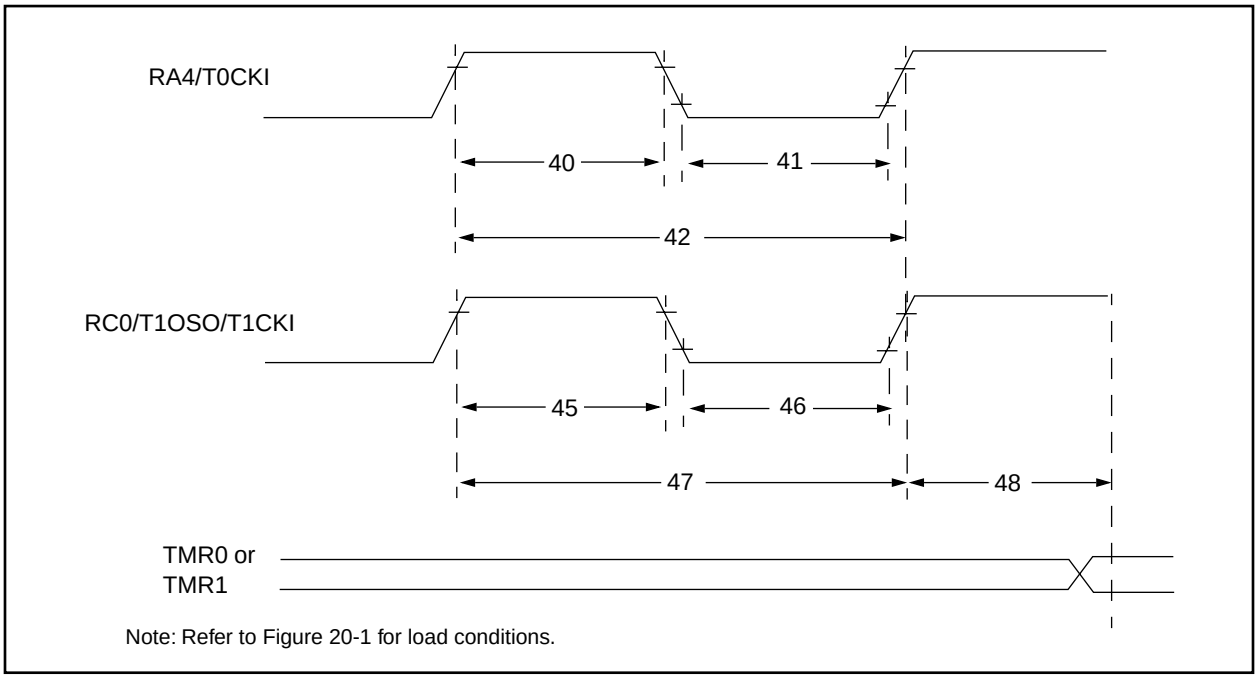
| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 30 | TmcL | $\overline{MCLR}$ Pulse Width (low) | 2 | — | — | µs | VDD = 5V, -40˚C to +125˚C |
| 31* | Twdt | Watchdog Timer Time-out Period (No Prescaler) | 7 | 18 | 33 | ms | VDD = 5V, -40˚C to +125˚C |
| 32 | Tost | Oscillation Start-up Timer Period | — | 1024TOSC | — | — | TOSC = OSC1 period |
| 33* | Tpwrt | Power up Timer Period | 28 | 72 | 132 | ms | VDD = 5V, -40˚C to +125˚C |
| 34 | TIOZ | I/O Hi-impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset | — | — | 2.1 | µs | |
| 35 | TBOR | Brown-out Reset pulse width | 100 | — | — | µs | VDD ≤ BVDD (D005) |

* 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.

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

**FIGURE 20-6: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**




Note: Refer to Figure 20-1 for load conditions.

**TABLE 20-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

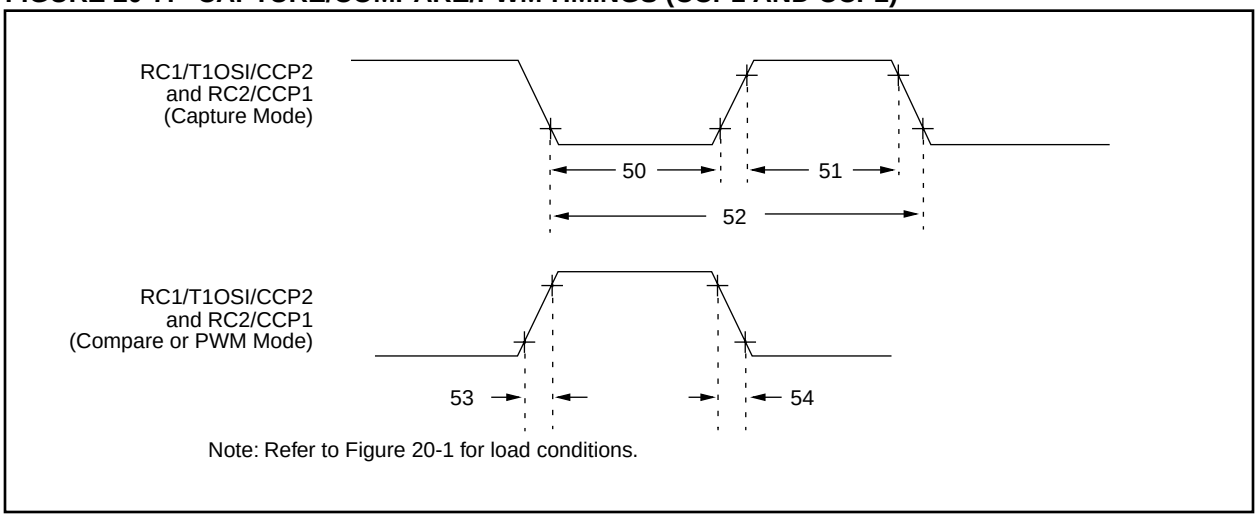
| Param No. | Sym | Characteristic | | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 40* | Tt0H | T0CKI High Pulse Width | | No Prescaler | 0.5TCY + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With Prescaler | 10 | — | — | ns | |
| 41* | Tt0L | T0CKI Low Pulse Width | | No Prescaler | 0.5TCY + 20 | — | — | ns | Must also meet parameter 42 |
| | | | | With Prescaler | 10 | — | — | ns | |
| 42* | Tt0P | T0CKI Period | | No Prescaler | TCY + 40 | — | — | ns | |
| | | | | With Prescaler | Greater of: 20 or TCY + 40 / N | — | — | ns | N = prescale value (2, 4, ..., 256) |
| 45* | Tt1H | T1CKI High Time | Synchronous, Prescaler = 1 | | 0.5TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | PIC16**C**7X | 15 | — | — | ns | |
| | | | | PIC16**LC**7X | 25 | — | — | ns | |
| | | | Asynchronous | PIC16**C**7X | 30 | — | — | ns | |
| | | | | PIC16**LC**7X | 50 | — | — | ns | |
| 46* | Tt1L | T1CKI Low Time | Synchronous, Prescaler = 1 | | 0.5TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | PIC16**C**7X | 15 | — | — | ns | |
| | | | | PIC16**LC**7X | 25 | — | — | ns | |
| | | | Asynchronous | PIC16**C**7X | 30 | — | — | ns | |
| | | | | PIC16**LC**7X | 50 | — | — | ns | |
| 47* | Tt1P | T1CKI input period | Synchronous | PIC16**C**7X | Greater of: 30 OR TCY + 40 / N | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | | PIC16**LC**7X | Greater of: 50 OR TCY + 40 / N | | | | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | PIC16**C**7X | 60 | — | — | ns | |
| | | | | PIC16**LC**7X | 100 | — | — | ns | |
| | Ft1 | Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN) | | | DC | — | 200 | kHz | |
| 48 | TCKEZtmr1 | Delay from external clock edge to timer increment | | | 2Tosc | — | 7Tosc | — | |

* 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.

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

**FIGURE 20-7: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)**




Note: Refer to Figure 20-1 for load conditions.

**TABLE 20-6: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1 AND CCP2)**

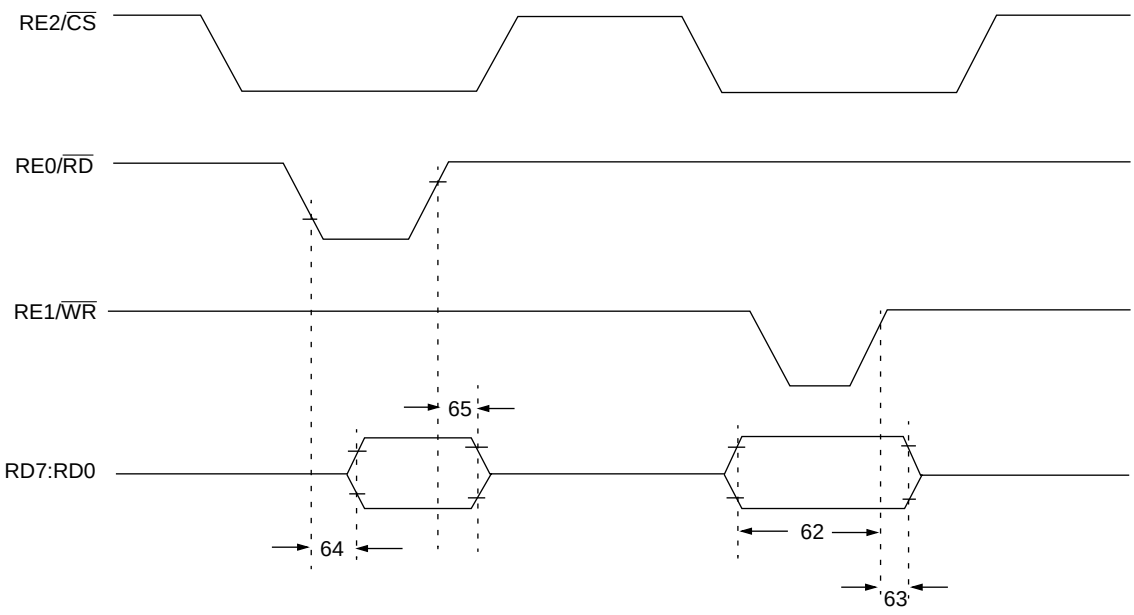
| Param No. | Sym | Characteristic | | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|---|
| 50* | TccL | CCP1 and CCP2 input low time | No Prescaler | | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | PIC16**C**76/77 | 10 | — | — | ns | |
| | | | | PIC16**LC**76/77 | 20 | — | — | ns | |
| 51* | TccH | CCP1 and CCP2 input high time | No Prescaler | | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | PIC16**C**76/77 | 10 | — | — | ns | |
| | | | | PIC16**LC**76/77 | 20 | — | — | ns | |
| 52* | TccP | CCP1 and CCP2 input period | | | $\frac{3T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1,4 or 16) |
| 53* | TccR | CCP1 and CCP2 output rise time | | PIC16**C**76/77 | — | 10 | 25 | ns | |
| | | | | PIC16**LC**76/77 | — | 25 | 45 | ns | |
| 54* | TccF | CCP1 and CCP2 output fall time | | PIC16**C**76/77 | — | 10 | 25 | ns | |
| | | | | PIC16**LC**76/77 | — | 25 | 45 | 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.

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

**FIGURE 20-8: PARALLEL SLAVE PORT TIMING (PIC16C77)**




Note: Refer to Figure 20-1 for load conditions

**TABLE 20-7: PARALLEL SLAVE PORT REQUIREMENTS (PIC16C77)**

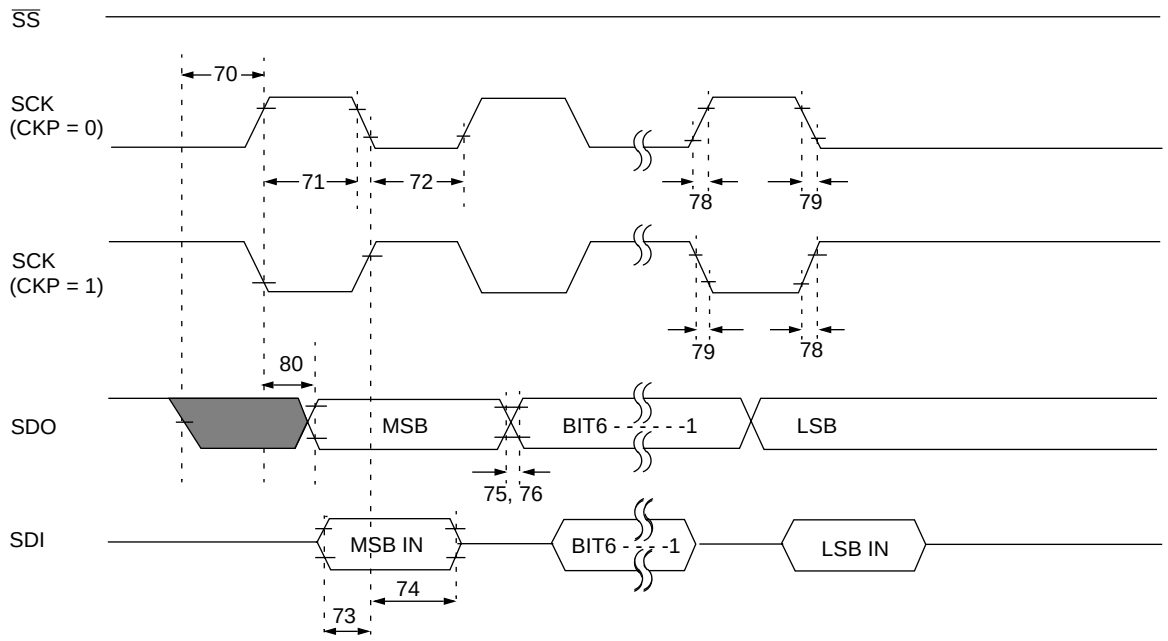
| Parameter No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 62 | TdtV2wrH | Data in valid before WR↑ or CS↑ (setup time) | | 20<br>25 | —<br>— | —<br>— | ns<br>ns | <br>Extended Range Only |
| 63* | TwrH2dtI | WR↑ or CS↑ to data–in invalid (hold time) | PIC16**C**77 | 20 | — | — | ns | |
| | | | PIC16**LC**77 | 35 | — | — | ns | |
| 64 | TrdL2dtV | RD↓ and CS↓ to data–out valid | | —<br>— | —<br>— | 80<br>90 | ns<br>ns | <br>Extended Range Only |
| 65 | TrdH2dtI | RD↑ or CS↓ to data–out invalid | | 10 | — | 30 | 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.

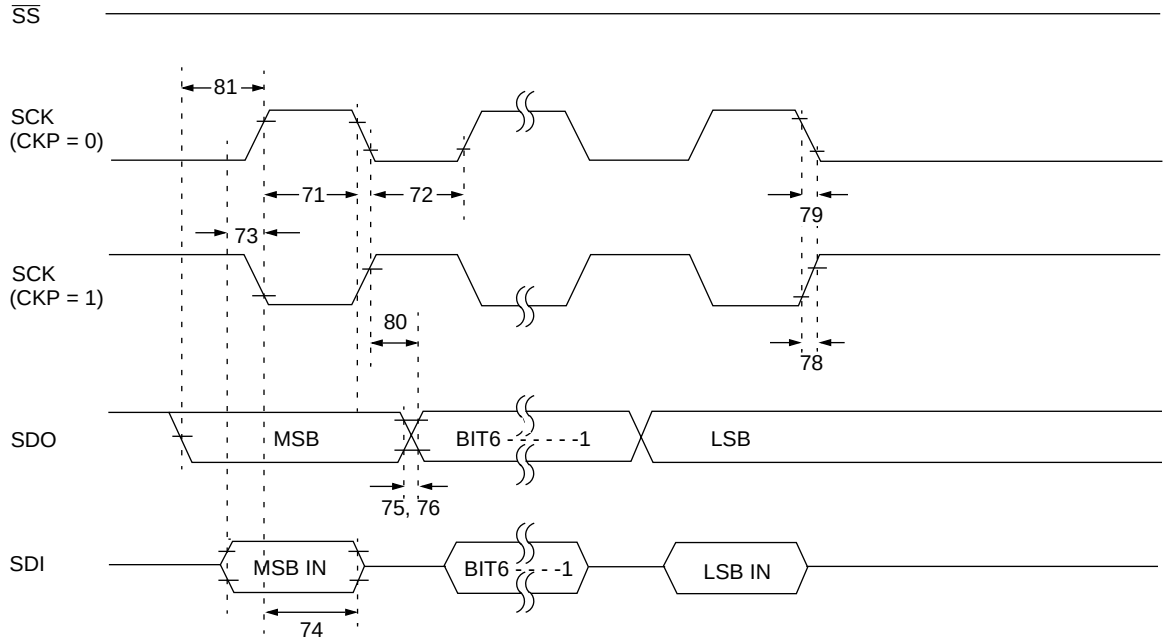
| Applicable Devices | 72 | 73 | 73A | 74 | 74A | 76 | 77 |
|---|---|---|---|---|---|---|---|

**FIGURE 20-9: SPI MASTER MODE TIMING (CKE = 0)**




Refer to Figure 20-1 for load conditions.

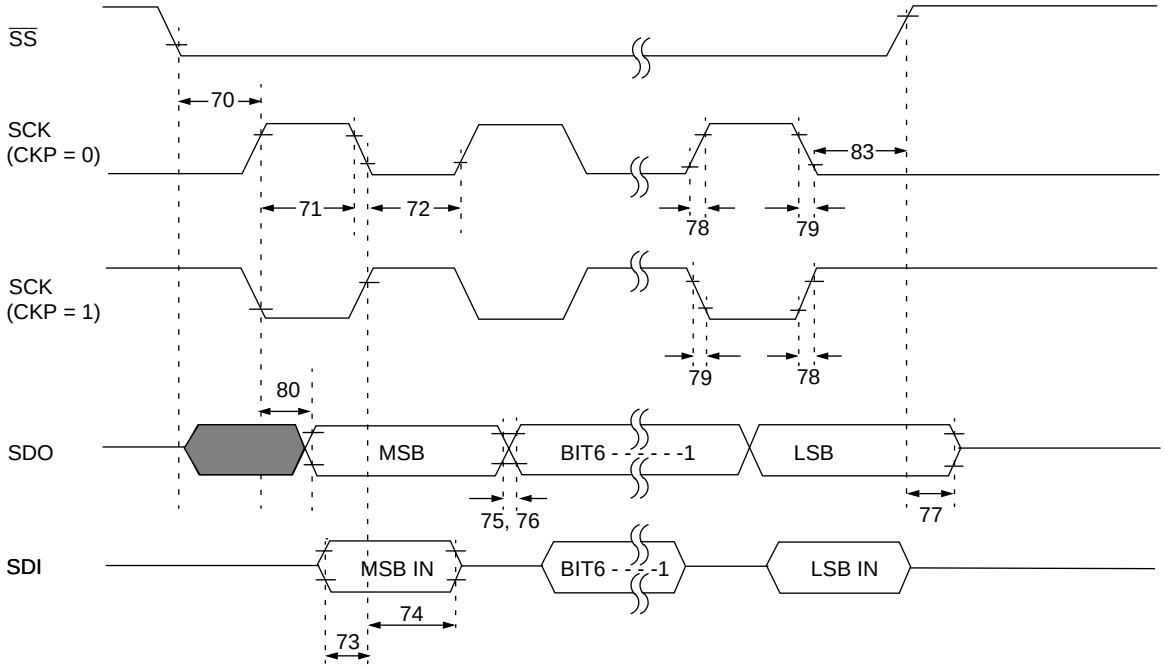
**FIGURE 20-10: SPI MASTER MODE TIMING (CKE = 1)**




Refer to Figure 20-1 for load conditions.

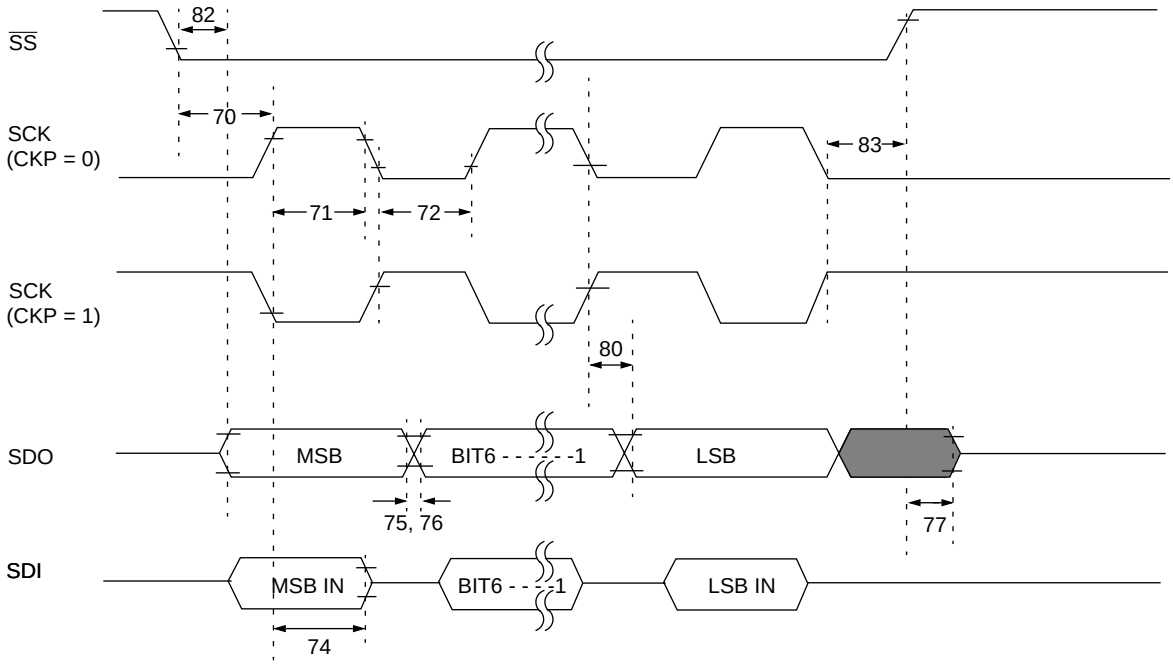
| Applicable Devices | 72 | 73 | 73A | 74 | 74A | 76 | 77 |
|---|---|---|---|---|---|---|---|

**FIGURE 20-11: SPI SLAVE MODE TIMING (CKE = 0)**




Refer to Figure 20-1 for load conditions.

**FIGURE 20-12: SPI SLAVE MODE TIMING (CKE = 1)**




Refer to Figure 20-1 for load conditions.

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

**TABLE 20-8: SPI MODE REQUIREMENTS**

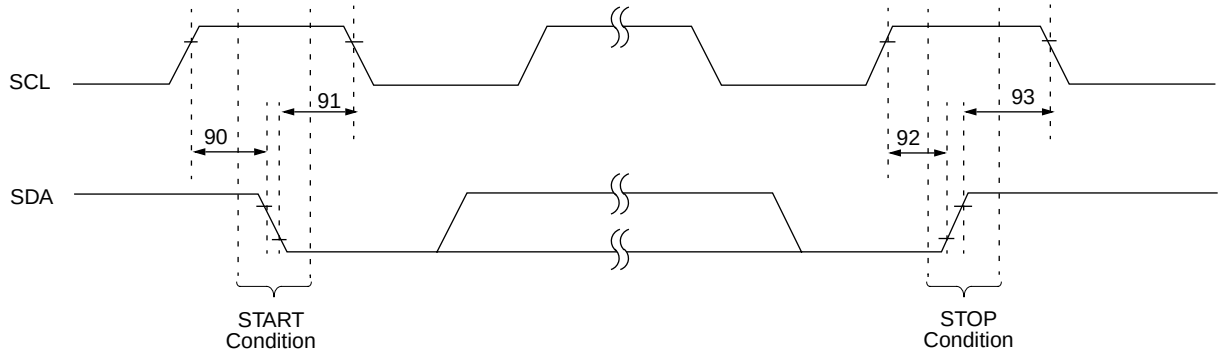
| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 70* | TssL2scH, TssL2scL | $\overline{SS}\downarrow$ to SCK$\downarrow$ or SCK$\uparrow$ input | TCY | — | — | ns | |
| 71* | TscH | SCK input high time (slave mode) | TCY + 20 | — | — | ns | |
| 72* | TscL | SCK input low time (slave mode) | TCY + 20 | — | — | ns | |
| 73* | TdiV2scH, TdiV2scL | Setup time of SDI data input to SCK edge | 100 | — | — | ns | |
| 74* | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | 100 | — | — | ns | |
| 75* | TdoR | SDO data output rise time | — | 10 | 25 | ns | |
| 76* | TdoF | SDO data output fall time | — | 10 | 25 | ns | |
| 77* | TssH2doZ | $\overline{SS}\uparrow$ to SDO output hi-impedance | 10 | — | 50 | ns | |
| 78* | TscR | SCK output rise time (master mode) | — | 10 | 25 | ns | |
| 79* | TscF | SCK output fall time (master mode) | — | 10 | 25 | ns | |
| 80* | TscH2doV, TscL2doV | SDO data output valid after SCK edge | — | — | 50 | ns | |
| 81* | TdoV2scH, TdoV2scL | SDO data output setup to SCK edge | TCY | — | — | ns | |
| 82* | TssL2doV | SDO data output valid after $\overline{SS}\downarrow$ edge | — | — | 50 | ns | |
| 83* | TscH2ssH, TscL2ssH | $\overline{SS}\uparrow$ after SCK edge | 1.5TCY + 40 | — | — | 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.

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

**FIGURE 20-13: $I^2C$ BUS START/STOP BITS TIMING**



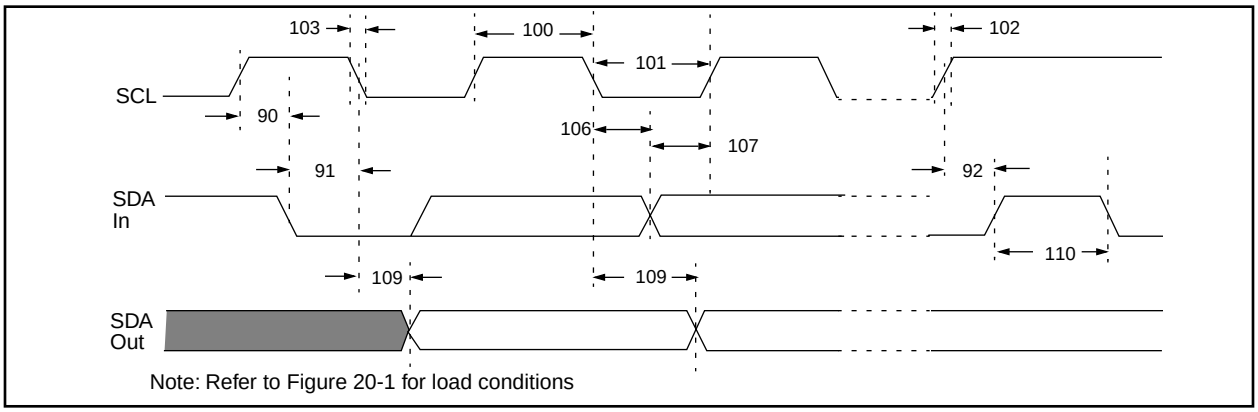

Note: Refer to Figure 20-1 for load conditions

**TABLE 20-9: $I^2C$ BUS START/STOP BITS REQUIREMENTS**

| Parameter No. | Sym | Characteristic | | Min | Typ | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 90 | TSU:STA | START condition Setup time | 100 kHz mode | 4700 | — | — | ns | Only relevant for repeated START condition |
| | | | 400 kHz mode | 600 | — | — | | |
| 91 | THD:STA | START condition Hold time | 100 kHz mode | 4000 | — | — | ns | After this period the first clock pulse is generated |
| | | | 400 kHz mode | 600 | — | — | | |
| 92 | TSU:STO | STOP condition Setup time | 100 kHz mode | 4700 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |
| 93 | THD:STO | STOP condition Hold time | 100 kHz mode | 4000 | — | — | ns | |
| | | | 400 kHz mode | 600 | — | — | | |

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

**FIGURE 20-14: I²C BUS DATA TIMING**




Note: Refer to Figure 20-1 for load conditions

**TABLE 20-10: I²C BUS DATA REQUIREMENTS**

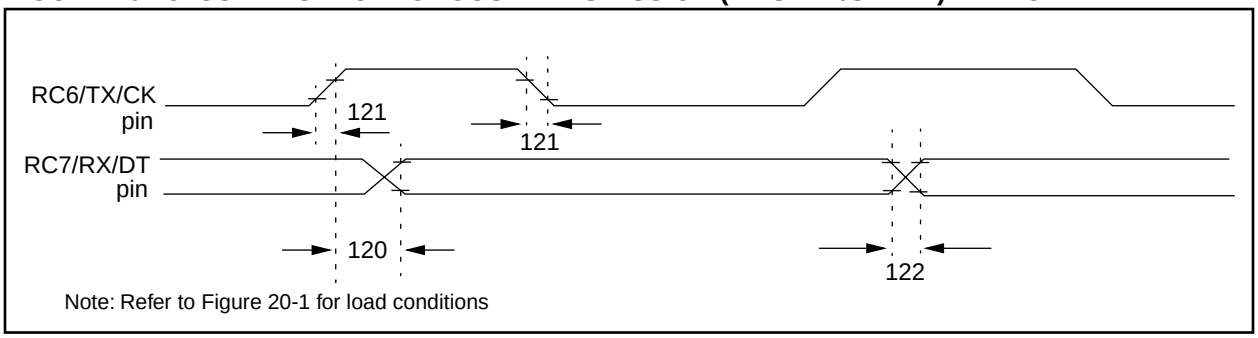
| Parameter No. | Sym | Characteristic | | Min | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 100 | THIGH | Clock high time | 100 kHz mode | 4.0 | — | µs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 0.6 | — | µs | Device must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5TCY | — | | |
| 101 | TLOW | Clock low time | 100 kHz mode | 4.7 | — | µs | Device must operate at a minimum of 1.5 MHz |
| | | | 400 kHz mode | 1.3 | — | µs | Device must operate at a minimum of 10 MHz |
| | | | SSP Module | 1.5TCY | — | | |
| 102 | TR | SDA and SCL rise time | 100 kHz mode | — | 1000 | ns | |
| | | | 400 kHz mode | 20 + 0.1Cb | 300 | ns | Cb is specified to be from 10 to 400 pF |
| 103 | TF | SDA and SCL fall time | 100 kHz mode | — | 300 | ns | |
| | | | 400 kHz mode | 20 + 0.1Cb | 300 | ns | Cb is specified to be from 10 to 400 pF |
| 90 | TSU:STA | START condition setup time | 100 kHz mode | 4.7 | — | µs | Only relevant for repeated START condition |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 91 | THD:STA | START condition hold time | 100 kHz mode | 4.0 | — | µs | After this period the first clock pulse is generated |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 106 | THD:DAT | Data input hold time | 100 kHz mode | 0 | — | ns | |
| | | | 400 kHz mode | 0 | 0.9 | µs | |
| 107 | TSU:DAT | Data input setup time | 100 kHz mode | 250 | — | ns | Note 2 |
| | | | 400 kHz mode | 100 | — | ns | |
| 92 | TSU:STO | STOP condition setup time | 100 kHz mode | 4.7 | — | µs | |
| | | | 400 kHz mode | 0.6 | — | µs | |
| 109 | TAA | Output valid from clock | 100 kHz mode | — | 3500 | ns | Note 1 |
| | | | 400 kHz mode | — | — | ns | |
| 110 | TBUF | Bus free time | 100 kHz mode | 4.7 | — | µs | Time the bus must be free before a new transmission can start |
| | | | 400 kHz mode | 1.3 | — | µs | |
| | Cb | Bus capacitive loading | | — | 400 | pF | |

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A fast-mode (400 kHz) I²C-bus device can be used in a standard-mode (100 kHz) I²C-bus system, but the requirement tsu;DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line TR max.+tsu;DAT = 1000 + 250 = 1250 ns (according to the standard-mode I²C bus specification) before the SCL line is released.

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

**FIGURE 20-15: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**



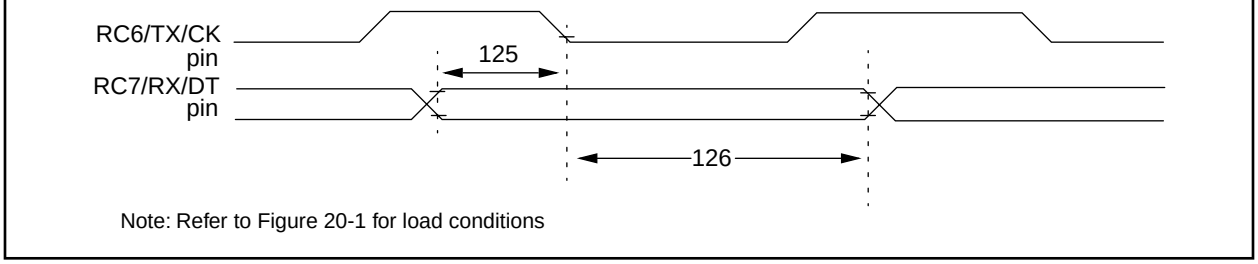

Note: Refer to Figure 20-1 for load conditions

**TABLE 20-11: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 120 | TckH2dtV | SYNC XMIT (MASTER & SLAVE) Clock high to data out valid | PIC16**C**76/77 | — | — | 80 | ns | |
| | | | PIC16**LC**76/77 | — | — | 100 | ns | |
| 121 | Tckrf | Clock out rise time and fall time (Master Mode) | PIC16**C**76/77 | — | — | 45 | ns | |
| | | | PIC16**LC**76/77 | — | — | 50 | ns | |
| 122 | Tdtrf | Data out rise time and fall time | PIC16**C**76/77 | — | — | 45 | ns | |
| | | | PIC16**LC**76/77 | — | — | 50 | ns | |

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

**FIGURE 20-16: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**




Note: Refer to Figure 20-1 for load conditions

**TABLE 20-12: USART SYNCHRONOUS RECEIVE REQUIREMENTS**

| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 125 | TdtV2ckL | SYNC RCV (MASTER & SLAVE) Data setup before CK ↓ (DT setup time) | 15 | — | — | ns | |
| 126 | TckL2dtl | Data hold after CK ↓ (DT hold time) | 15 | — | — | ns | |

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

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

**TABLE 20-13: A/D CONVERTER CHARACTERISTICS:**

**PIC16C76/77-04 (Commercial, Industrial, Extended)**
**PIC16C76/77-10 (Commercial, Industrial, Extended)**
**PIC16C76/77-20 (Commercial, Industrial, Extended)**
**PIC16LC76/77-04 (Commercial, Industrial)**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| A01 | NR | Resolution | | — | — | 8-bits | bit | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A02 | EABS | Total Absolute error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A03 | EIL | Integral linearity error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A04 | EDL | Differential linearity error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A05 | EFS | Full scale error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A06 | EOFF | Offset error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF |
| A10 | — | Monotonicity | | — | guaranteed | — | — | VSS ≤ VAIN ≤ VREF |
| A20 | VREF | Reference voltage | | 3.0V | — | VDD + 0.3 | V | |
| A25 | VAIN | Analog input voltage | | VSS - 0.3 | — | VREF + 0.3 | V | |
| A30 | ZAIN | Recommended impedance of analog voltage source | | — | — | 10.0 | kΩ | |
| A40 | IAD | A/D conversion current (VDD) | PIC16**C**76/77 | — | 180 | — | μA | Average current consumption when A/D is on. (Note 1) |
| | | | PIC16**LC**76/77 | — | 90 | — | μA | |
| A50 | IREF | VREF input current (Note 2) | | 10 | — | 1000 | μA | During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see Section 13.1. |
| | | | | — | — | 10 | μA | During A/D Conversion cycle |

* These parameters are characterized but not tested.

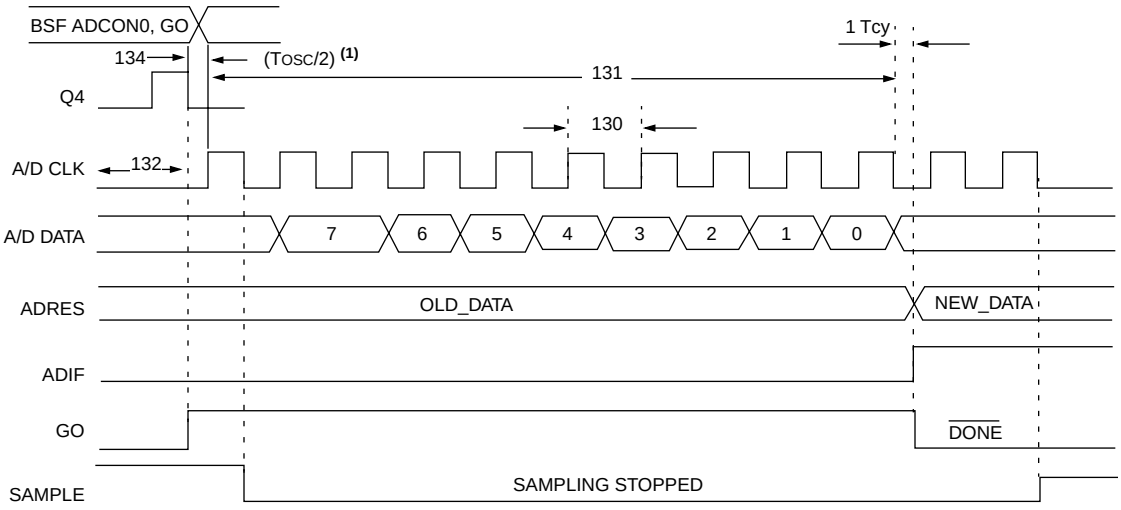
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

2: VREF current is from RA3 pin or VDD pin, whichever is selected as reference input.

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

**FIGURE 20-17: 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 20-14: A/D CONVERSION REQUIREMENTS**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 130 | TAD | A/D clock period | PIC16**C**76/77 | 1.6 | — | — | µs | TOSC based, VREF ≥ 3.0V |
| | | | PIC16**LC**76/77 | 2.0 | — | — | µs | TOSC based, VREF full range |
| | | | PIC16**C**76/77 | 2.0 | 4.0 | 6.0 | µs | A/D RC Mode |
| | | | PIC16**LC**76/77 | 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<br><br>5* | 20<br><br>— | —<br><br>— | µs<br><br>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., 20.0 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.

§ This specification ensured by design.

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

2: See Section 13.1 for min conditions.

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

**NOTES:**

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

## 21.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

The graphs and tables provided in this section are for design guidance and are not tested or guaranteed.

**In some graphs or tables the data presented are outside specified operating range (i.e., outside specified $V_{DD}$ range). This is for information only and devices are guaranteed to operate properly only within the specified range.**

**Note:** The data presented in this section is a statistical summary of data collected on units from different lots over a period of time and matrix samples. 'Typical' represents the mean of the distribution at, 25°C, while 'max' or 'min' represents (mean +3σ) and (mean -3σ) respectively where σ is standard deviation.

**FIGURE 21-1: TYPICAL $I_{PD}$ vs. $V_{DD}$ (WDT DISABLED, RC MODE)**

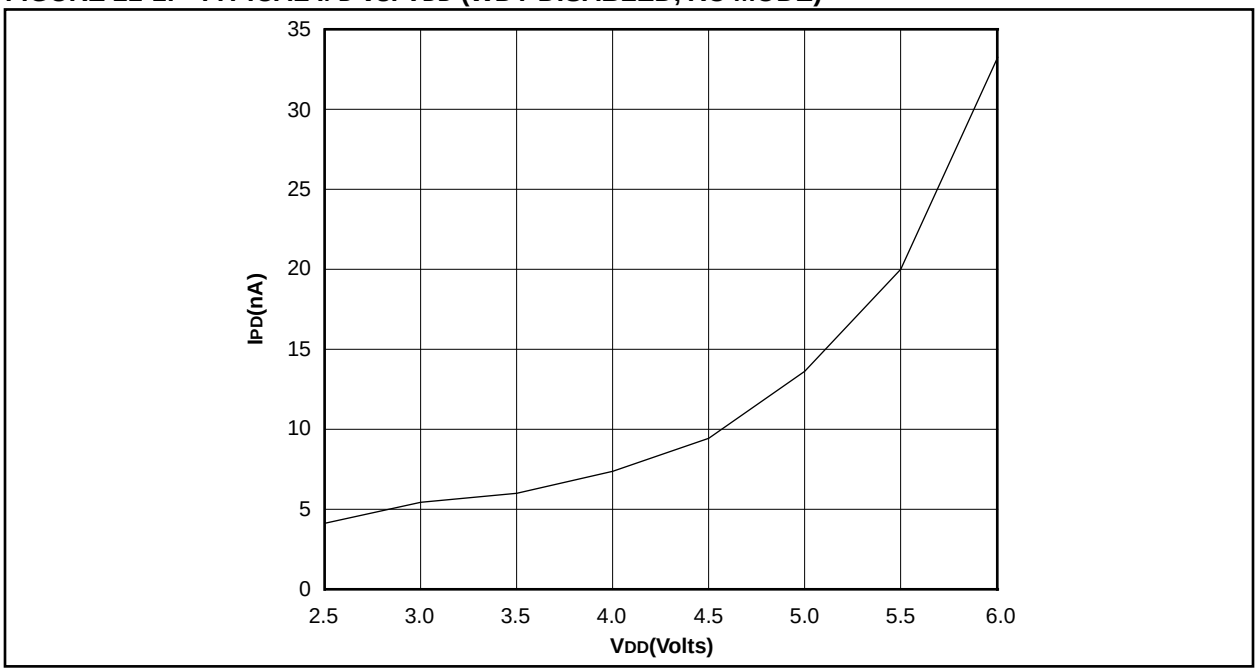
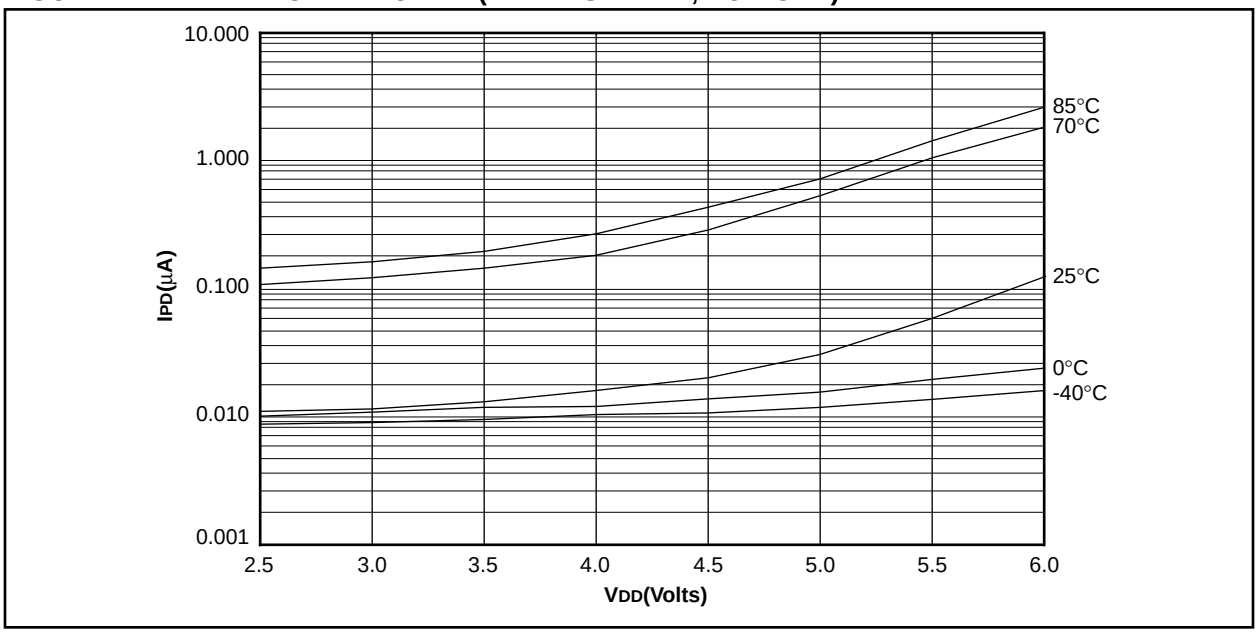



**FIGURE 21-2: MAXIMUM $I_{PD}$ vs. $V_{DD}$ (WDT DISABLED, RC MODE)**



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

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

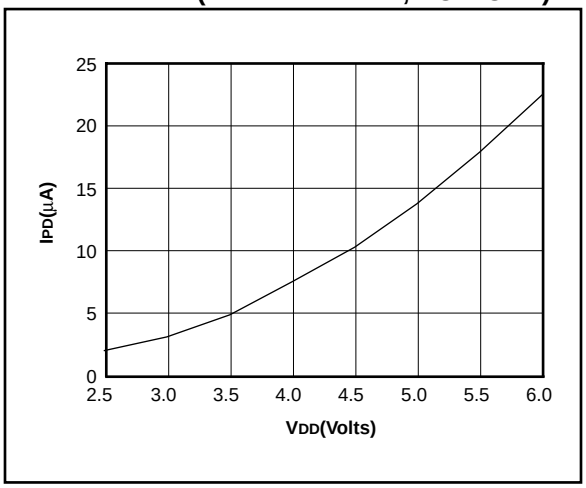



**FIGURE 21-4: MAXIMUM IPD vs. VDD (WDT ENABLED, RC MODE)**

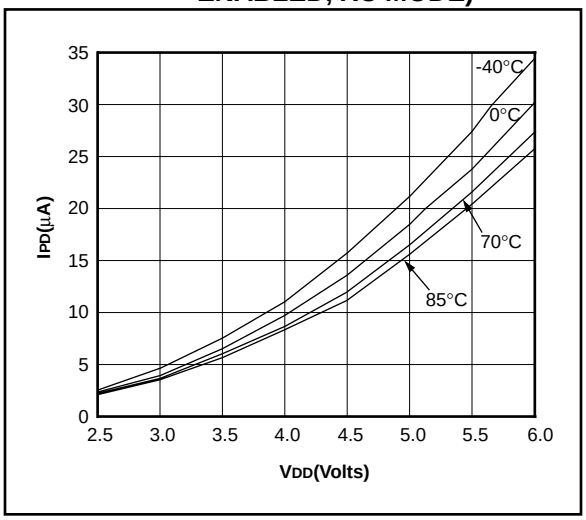
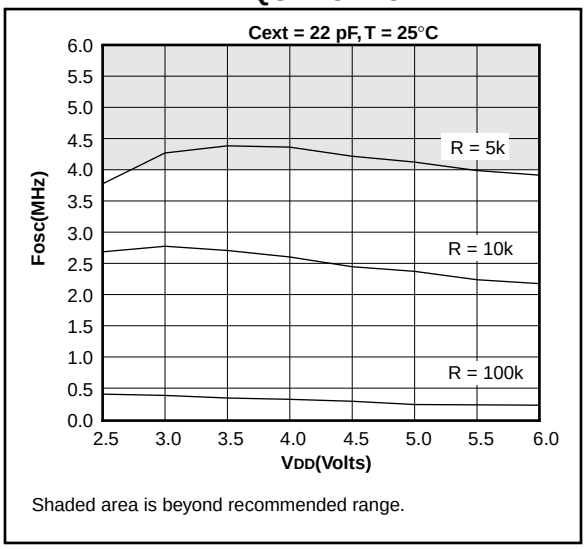



**FIGURE 21-5: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD**




Shaded area is beyond recommended range.

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

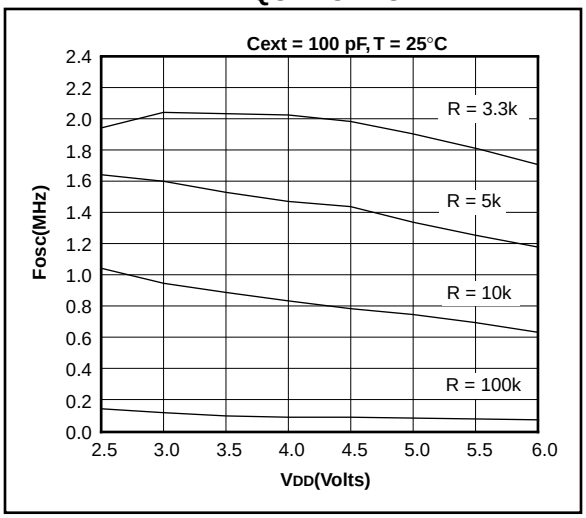
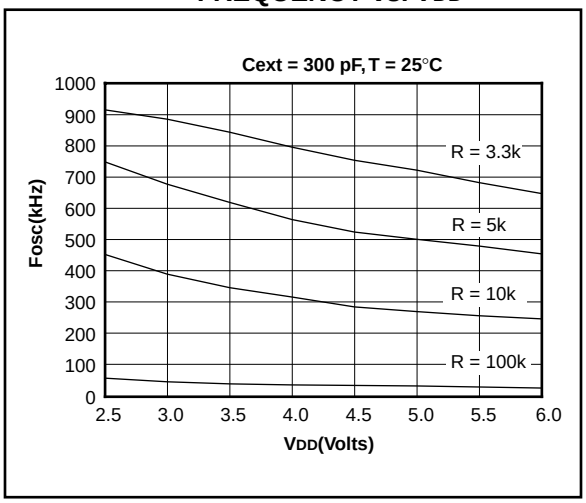


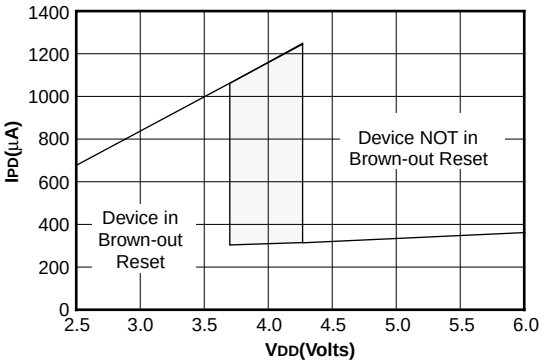

**FIGURE 21-7: TYPICAL RC OSCILLATOR FREQUENCY vs. VDD**




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

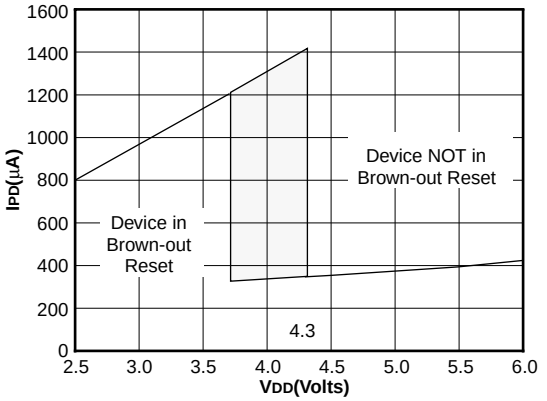
| Applicable Devices | 72 | 73 | 73A | 74 | 74A | 76 | 77 |
|---|---|---|---|---|---|---|---|

**FIGURE 21-8: TYPICAL IPD vs. VDD BROWN-OUT DETECT ENABLED (RC MODE)**




The shaded region represents the built-in hysteresis of the brown-out reset circuitry.

**FIGURE 21-9: MAXIMUM IPD vs. VDD BROWN-OUT DETECT ENABLED (85°C TO -40°C, RC MODE)**




The shaded region represents the built-in hysteresis of the brown-out reset circuitry.

**FIGURE 21-10: TYPICAL IPD vs. TIMER1 ENABLED (32 kHz, RC0/RC1 = 33 pF/33 pF, RC MODE)**

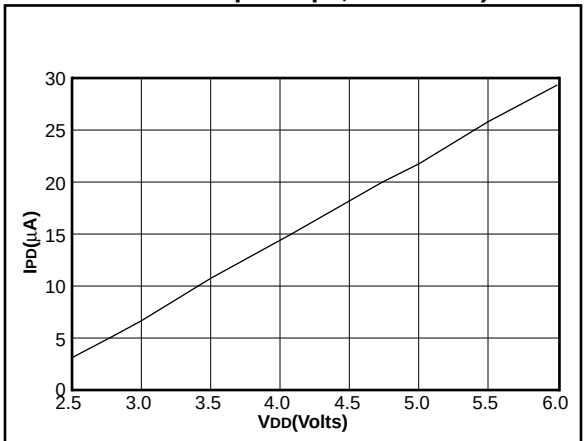
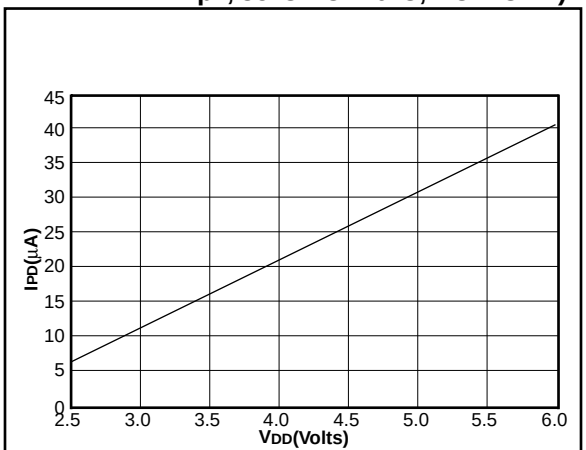



**FIGURE 21-11: MAXIMUM IPD vs. TIMER1 ENABLED (32 kHz, RC0/RC1 = 33 pF/33 pF, 85°C TO -40°C, RC MODE)**




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

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

**FIGURE 21-12: TYPICAL IDD vs. FREQUENCY (RC MODE @ 22 pF, 25°C)**

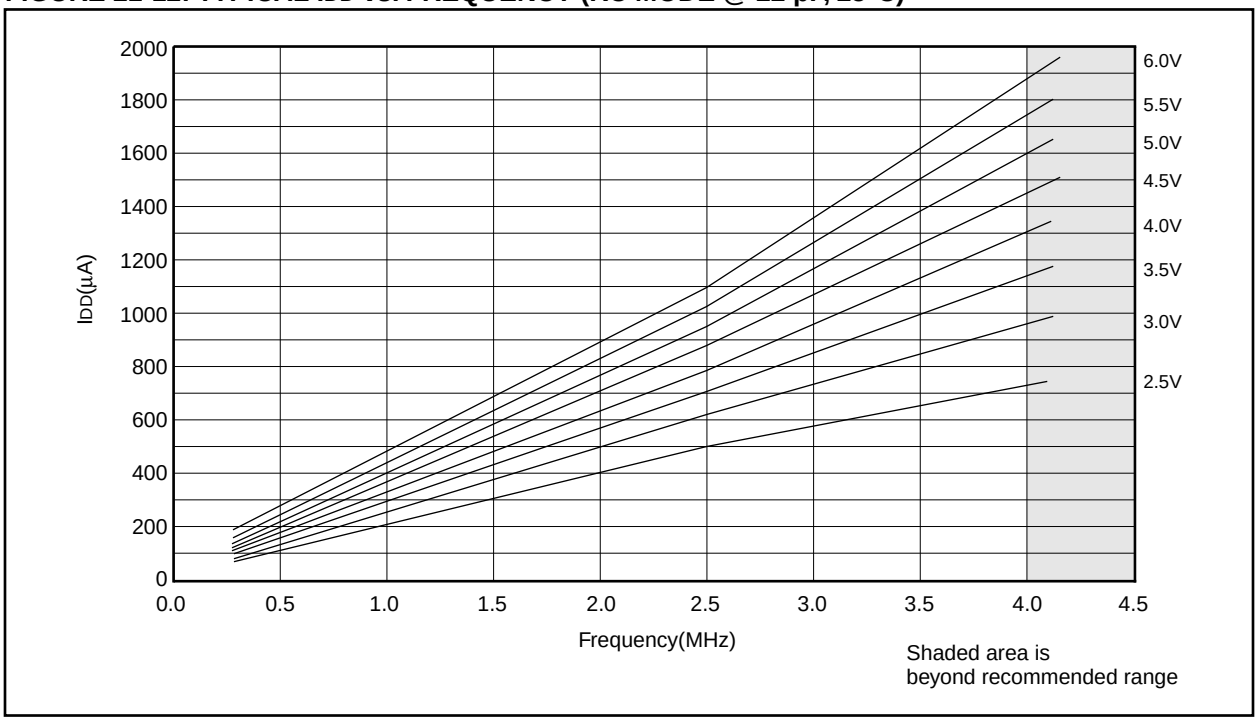
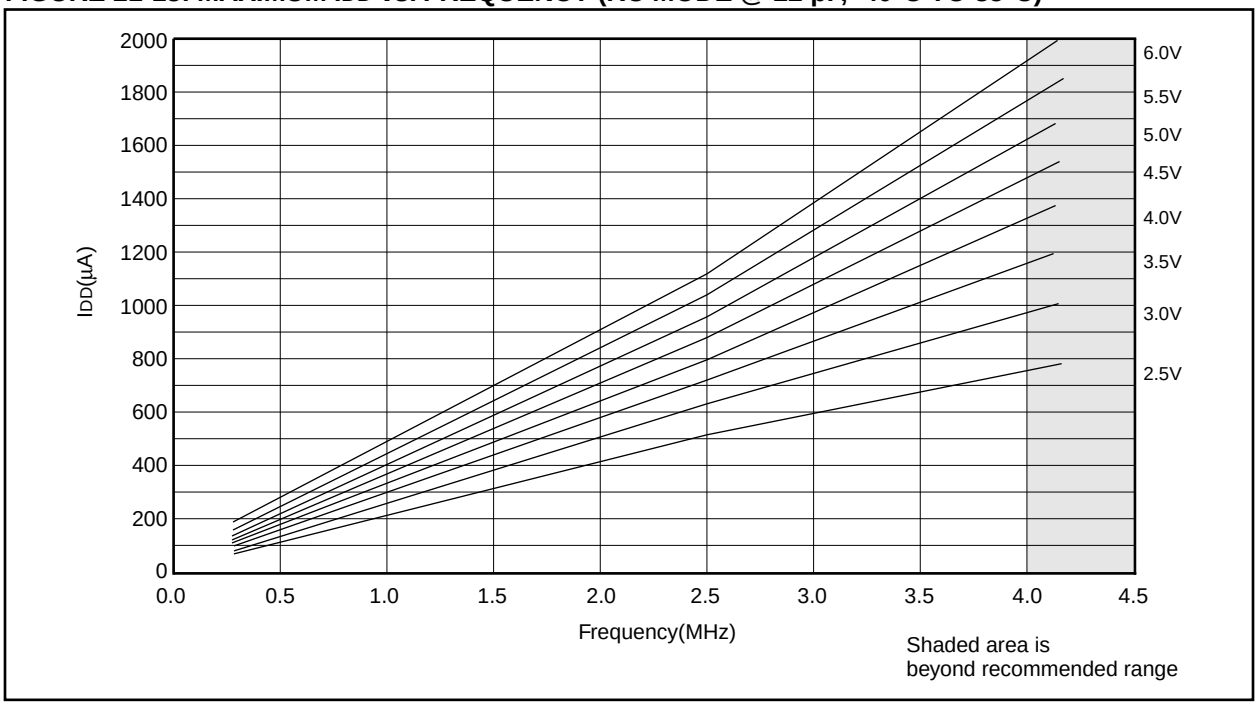



**FIGURE 21-13: MAXIMUM IDD vs. FREQUENCY (RC MODE @ 22 pF, -40°C TO 85°C)**




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

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

**FIGURE 21-14: TYPICAL IDD vs. FREQUENCY (RC MODE @ 100 pF, 25°C)**

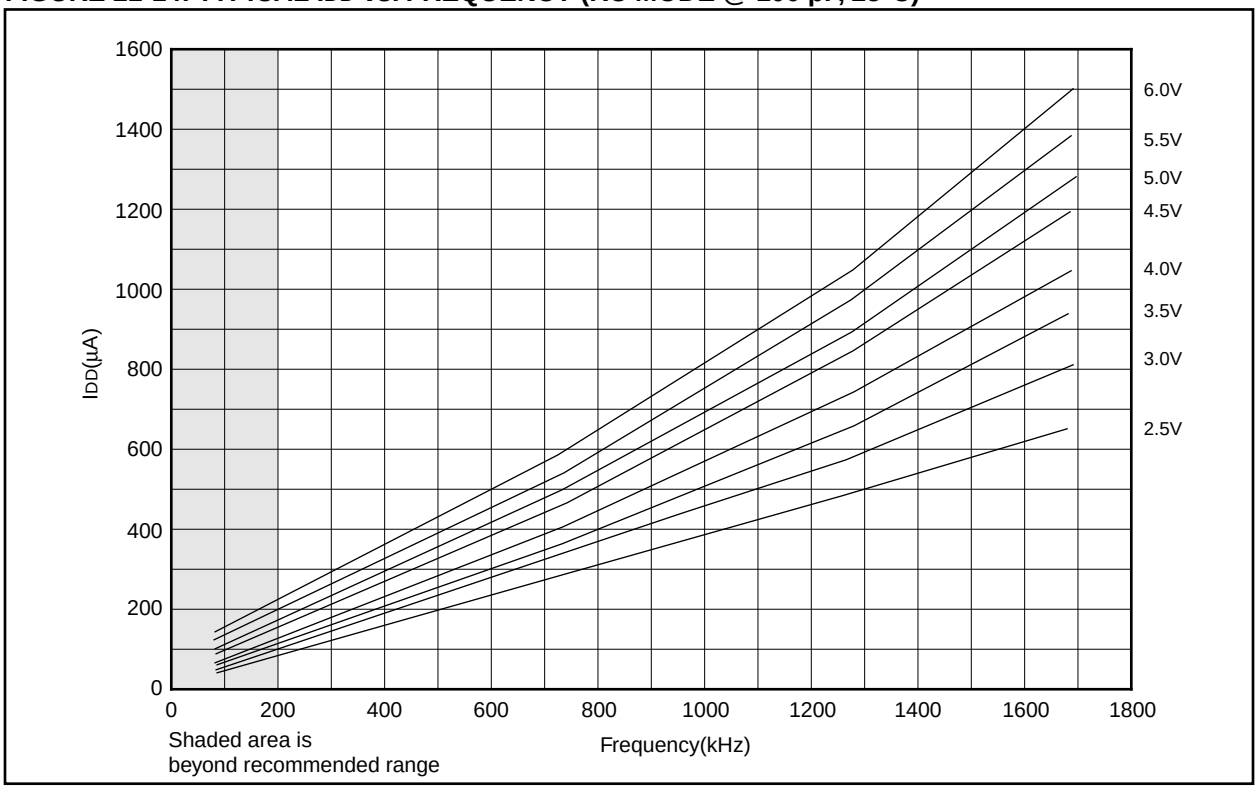
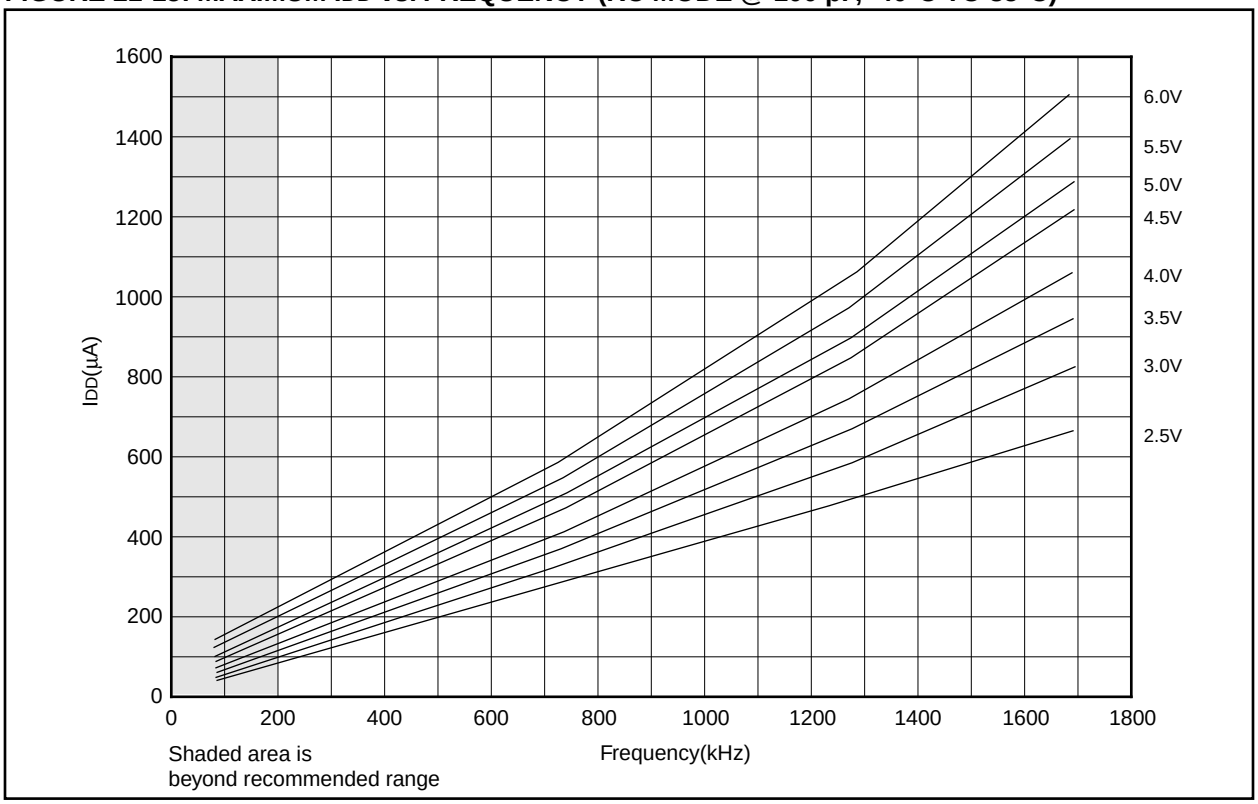



**FIGURE 21-15: MAXIMUM IDD vs. FREQUENCY (RC MODE @ 100 pF, -40°C TO 85°C)**




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

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

**FIGURE 21-16: TYPICAL IDD vs. FREQUENCY (RC MODE @ 300 pF, 25°C)**

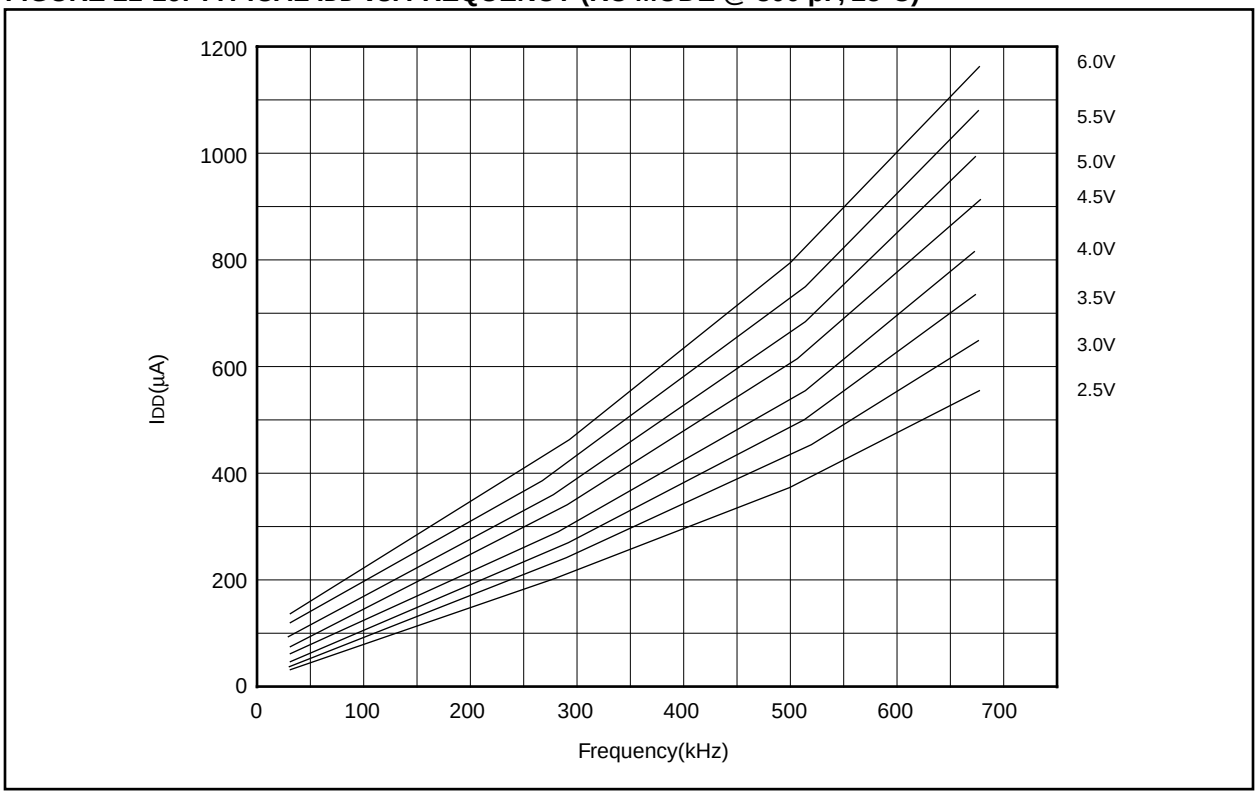
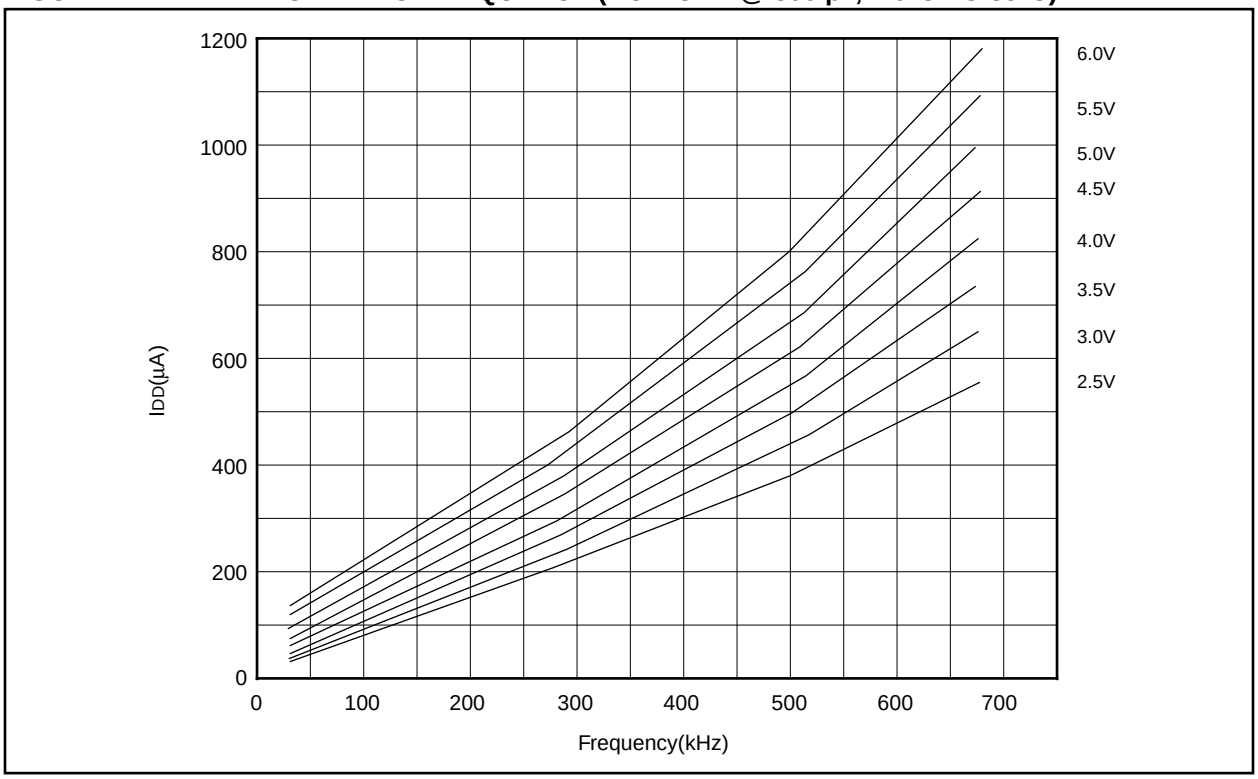



**FIGURE 21-17: MAXIMUM IDD vs. FREQUENCY (RC MODE @ 300 pF, -40°C TO 85°C)**




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

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

**FIGURE 21-18: TYPICAL IDD vs. CAPACITANCE @ 500 kHz (RC MODE)**

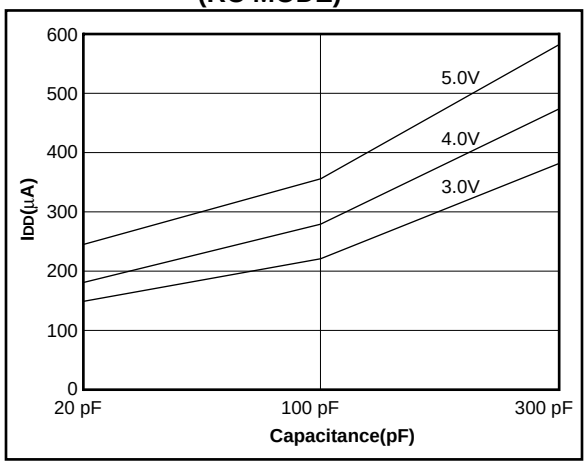



**TABLE 21-1: RC OSCILLATOR FREQUENCIES**

| 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 VDD = 5V.

**FIGURE 21-19: TRANSCONDUCTANCE(gm) OF HS OSCILLATOR vs. VDD**

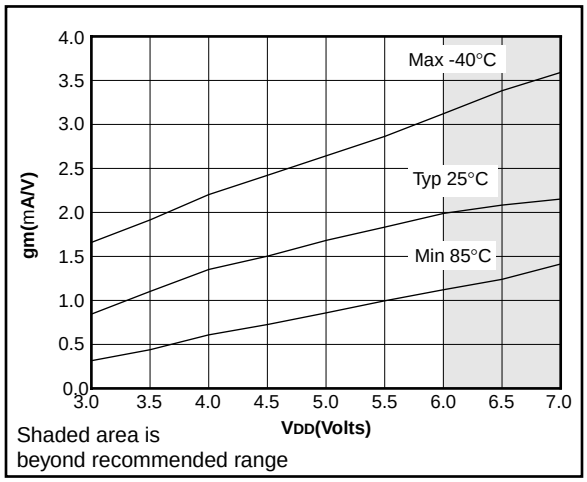



**FIGURE 21-20: TRANSCONDUCTANCE(gm) OF LP OSCILLATOR vs. VDD**

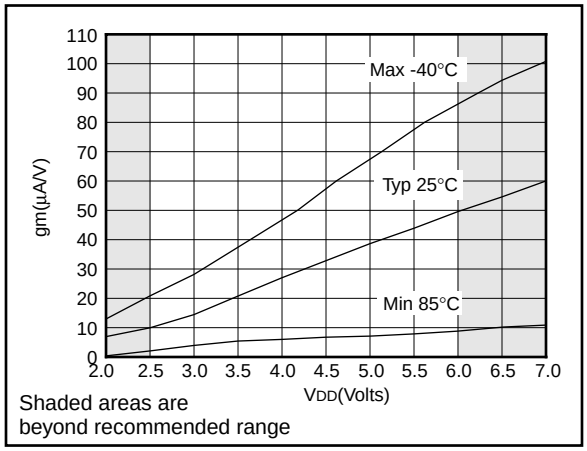
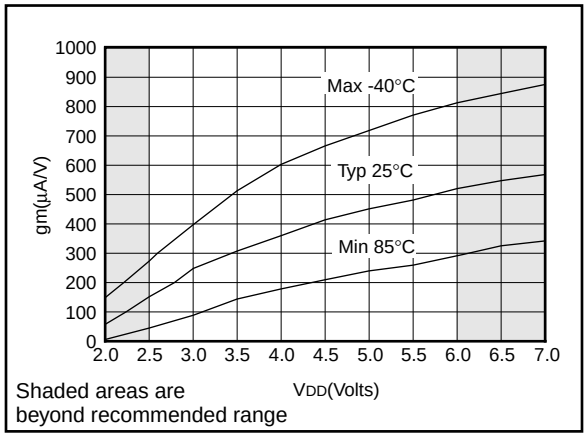



**FIGURE 21-21: TRANSCONDUCTANCE(gm) OF XT OSCILLATOR vs. VDD**




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

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

**FIGURE 21-22: TYPICAL XTAL STARTUP TIME vs. VDD (LP MODE, 25°C)**

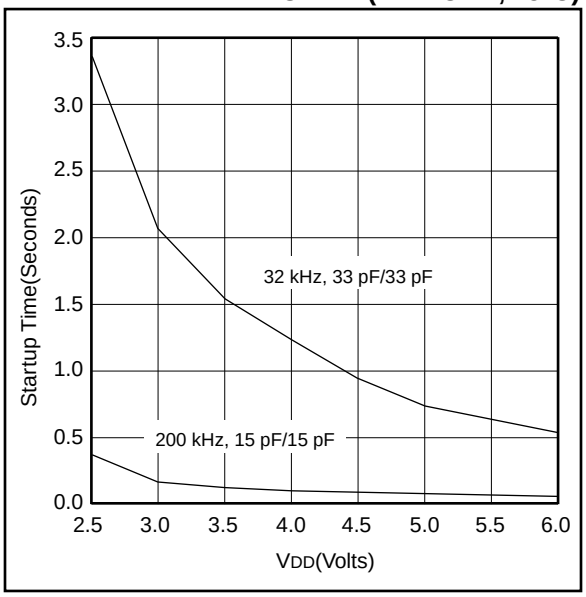



**FIGURE 21-23: TYPICAL XTAL STARTUP TIME vs. VDD (HS MODE, 25°C)**

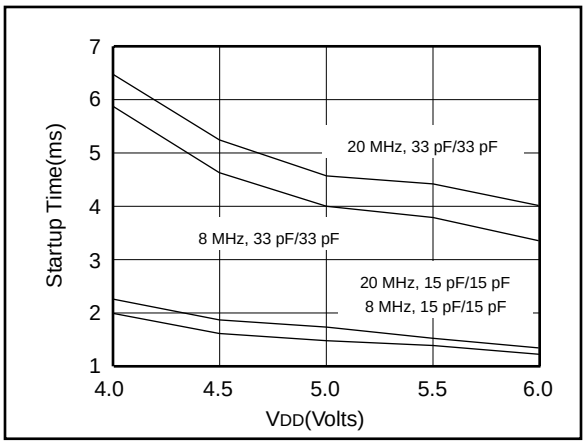



**FIGURE 21-24: TYPICAL XTAL STARTUP TIME vs. VDD (XT MODE, 25°C)**

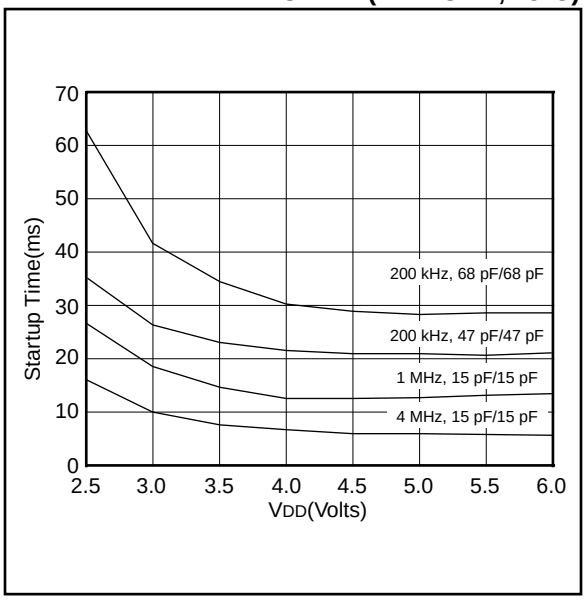



**TABLE 21-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATORS**

| Osc Type | Crystal Freq | Cap. Range C1 | Cap. Range C2 |
|---|---|---|---|
| LP | 32 kHz | 33 pF | 33 pF |
| | 200 kHz | 15 pF | 15 pF |
| XT | 200 kHz | 47-68 pF | 47-68 pF |
| | 1 MHz | 15 pF | 15 pF |
| | 4 MHz | 15 pF | 15 pF |
| HS | 4 MHz | 15 pF | 15 pF |
| | 8 MHz | 15-33 pF | 15-33 pF |
| | 20 MHz | 15-33 pF | 15-33 pF |
| | | | |
| **Crystals Used** | | | |
| 32 kHz | Epson C-001R32.768K-A | ± 20 PPM | |
| 200 kHz | STD XTL 200.000KHz | ± 20 PPM | |
| 1 MHz | ECS ECS-10-13-1 | ± 50 PPM | |
| 4 MHz | ECS ECS-40-20-1 | ± 50 PPM | |
| 8 MHz | EPSON CA-301 8.000M-C | ± 30 PPM | |
| 20 MHz | EPSON CA-301 20.000M-C | ± 30 PPM | |

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

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

**FIGURE 21-25: TYPICAL IDD vs. FREQUENCY (LP MODE, 25°C)**

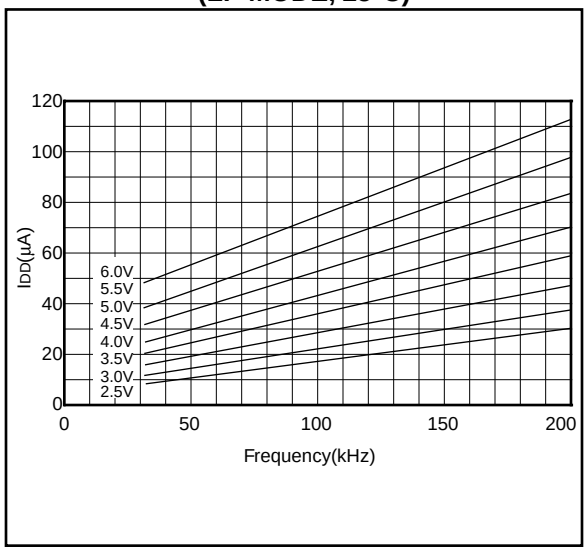



**FIGURE 21-26: MAXIMUM IDD vs. FREQUENCY (LP MODE, 85°C TO -40°C)**

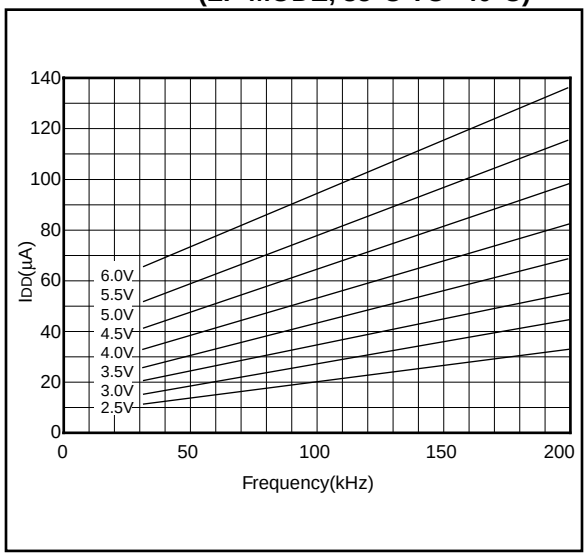



**FIGURE 21-27: TYPICAL IDD vs. FREQUENCY (XT MODE, 25°C)**

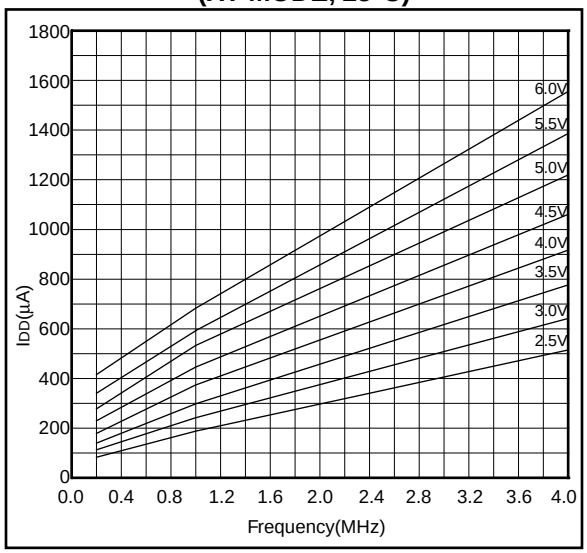
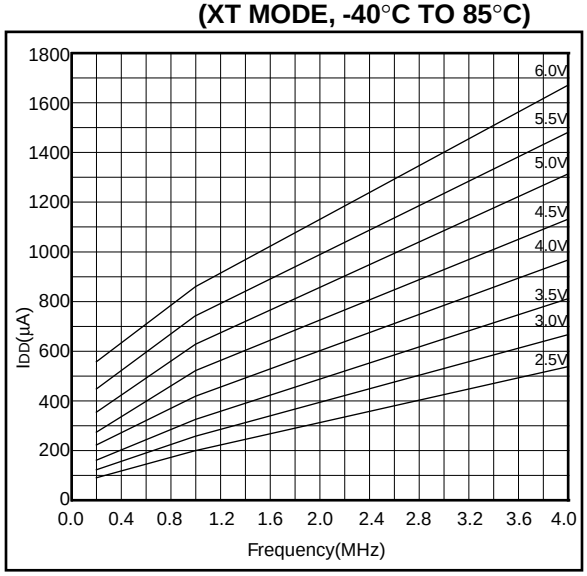



**FIGURE 21-28: MAXIMUM IDD vs. FREQUENCY (XT MODE, -40°C TO 85°C)**




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

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

**FIGURE 21-29: TYPICAL IDD vs. FREQUENCY (HS MODE, 25°C)**

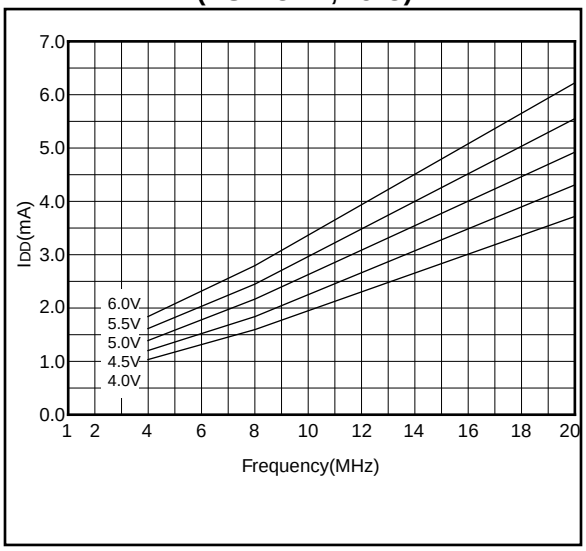
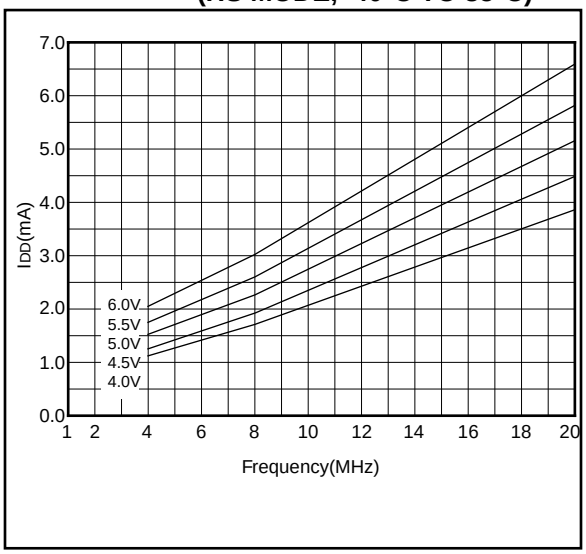


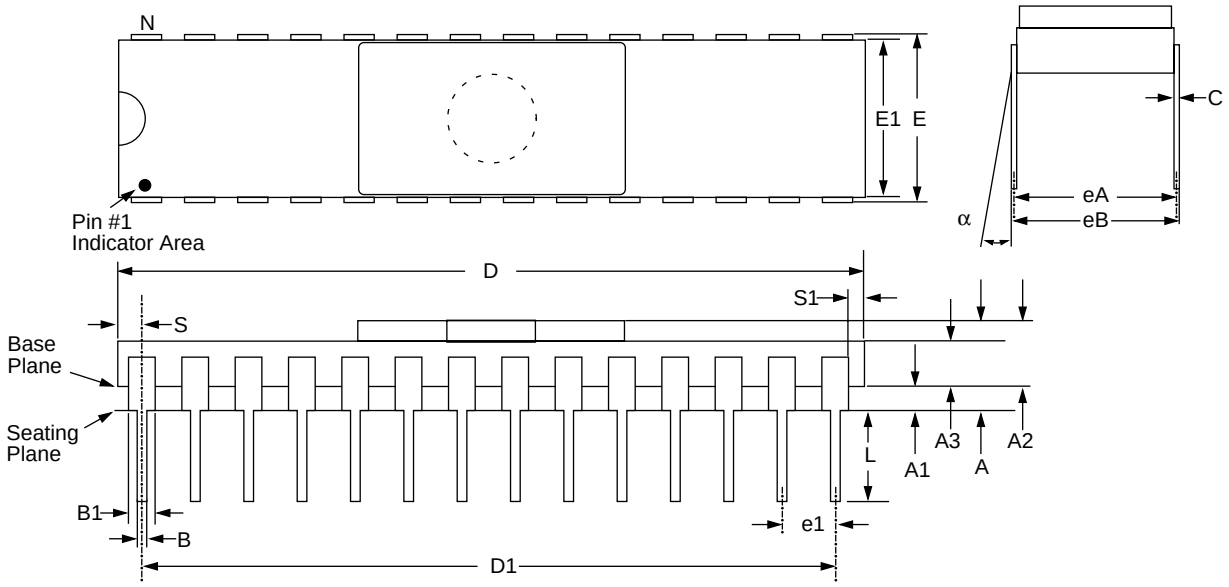

**FIGURE 21-30: MAXIMUM IDD vs. FREQUENCY (HS MODE, -40°C TO 85°C)**




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

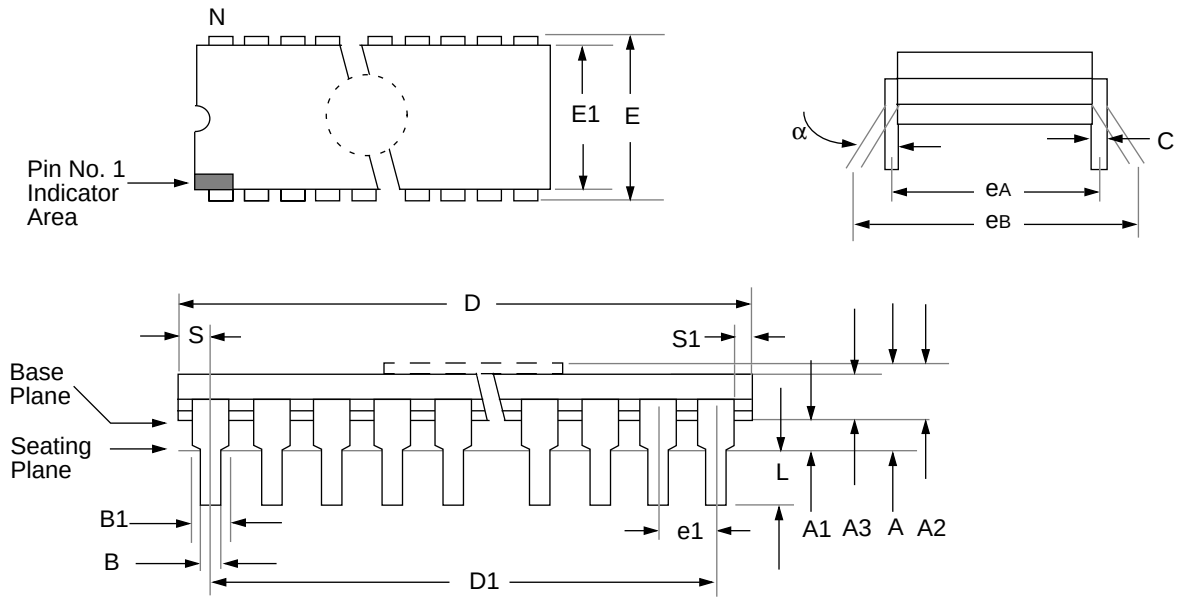
## 22.0 PACKAGING INFORMATION

### 22.1 28-Lead Ceramic Side Brazed Dual In-Line with Window (300 mil)(JW)



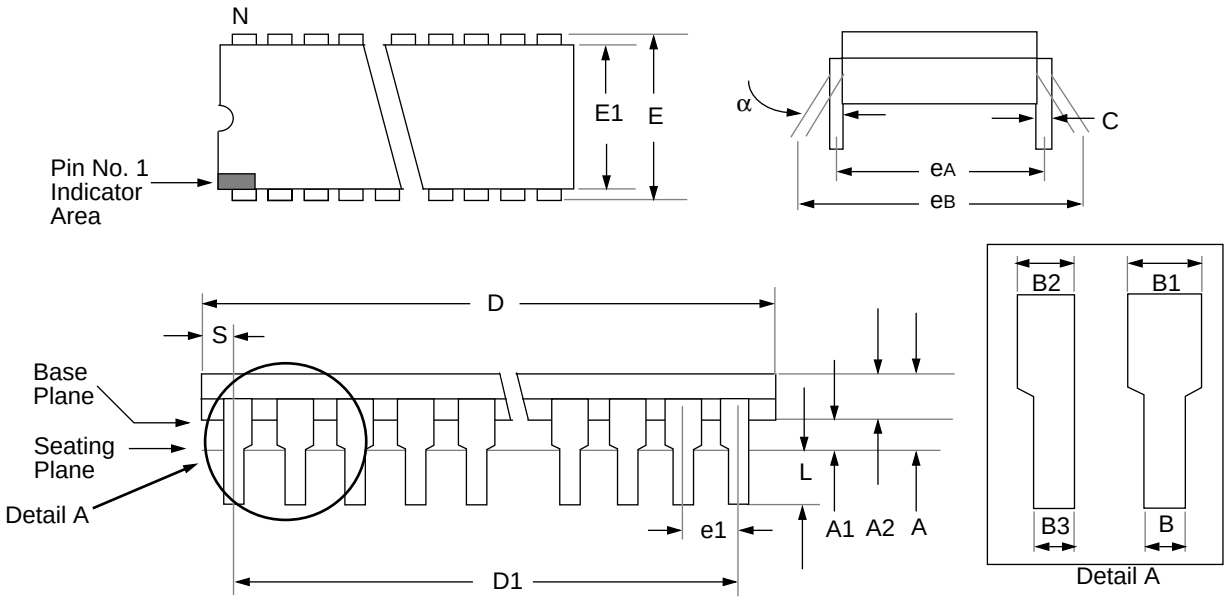

| Package Group: Ceramic Side Brazed Dual In-Line (CER) | | | | | | |
|---|---|---|---|---|---|---|
| Symbol | Millimeters | | | Inches | | |
| | Min | Max | Notes | Min | Max | Notes |
| α | 0° | 10° | | 0° | 10° | |
| A | 3.937 | 5.030 | | 0.155 | 0.198 | |
| A1 | 1.016 | 1.524 | | 0.040 | 0.060 | |
| A2 | 2.921 | 3.506 | | 0.115 | 0.138 | |
| A3 | 1.930 | 2.388 | | 0.076 | 0.094 | |
| B | 0.406 | 0.508 | | 0.016 | 0.020 | |
| B1 | 1.219 | 1.321 | **Typical** | 0.048 | 0.052 | |
| C | 0.228 | 0.305 | **Typical** | 0.009 | 0.012 | |
| D | 35.204 | 35.916 | | 1.386 | 1.414 | |
| D1 | 32.893 | 33.147 | **Reference** | 1.295 | 1.305 | |
| E | 7.620 | 8.128 | | 0.300 | 0.320 | |
| E1 | 7.366 | 7.620 | | 0.290 | 0.300 | |
| e1 | 2.413 | 2.667 | **Typical** | 0.095 | 0.105 | |
| eA | 7.366 | 7.874 | **Reference** | 0.290 | 0.310 | |
| eB | 7.594 | 8.179 | | 0.299 | 0.322 | |
| L | 3.302 | 4.064 | | 0.130 | 0.160 | |
| N | 28 | 28 | | 28 | 28 | |
| S | 1.143 | 1.397 | | 0.045 | 0.055 | |
| S1 | 0.533 | 0.737 | | 0.021 | 0.029 | |

## 22.2 40-Lead Ceramic CERDIP Dual In-line with Window (600 mil) (JW)



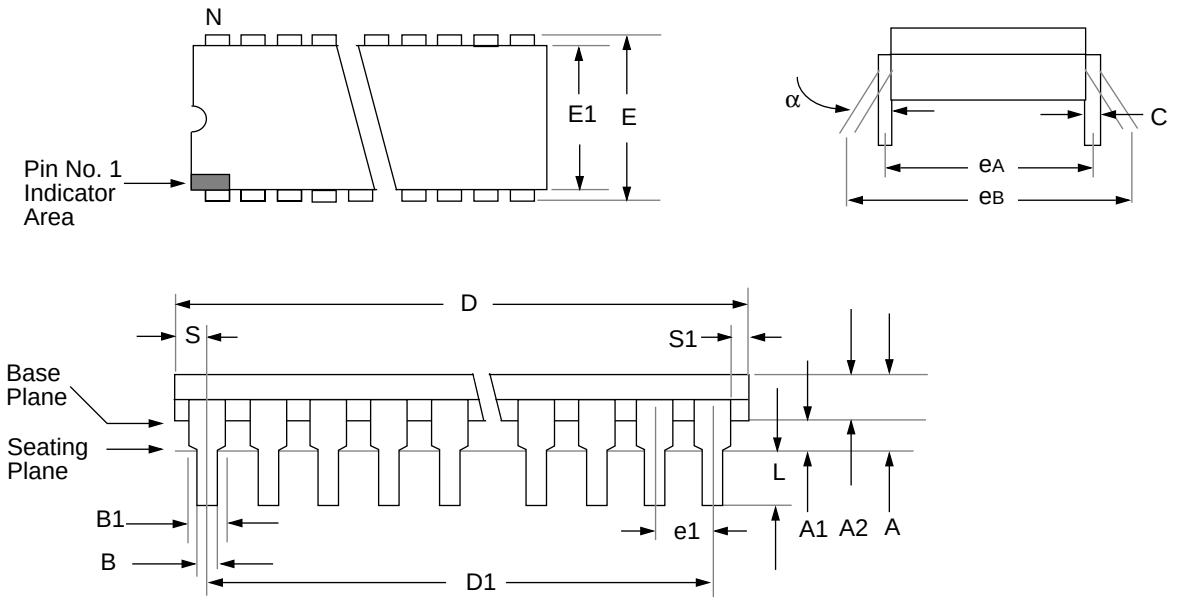

| Package Group: Ceramic CERDIP Dual In-Line (CDP) | | | | | | |
|---|---|---|---|---|---|---|
| | Millimeters | | | Inches | | |
| Symbol | Min | Max | Notes | Min | Max | Notes |
| α | 0° | 10° | | 0° | 10° | |
| A | 4.318 | 5.715 | | 0.170 | 0.225 | |
| A1 | 0.381 | 1.778 | | 0.015 | 0.070 | |
| A2 | 3.810 | 4.699 | | 0.150 | 0.185 | |
| A3 | 3.810 | 4.445 | | 0.150 | 0.175 | |
| B | 0.355 | 0.585 | | 0.014 | 0.023 | |
| B1 | 1.270 | 1.651 | **Typical** | 0.050 | 0.065 | **Typical** |
| C | 0.203 | 0.381 | **Typical** | 0.008 | 0.015 | **Typical** |
| D | 51.435 | 52.705 | | 2.025 | 2.075 | |
| D1 | 48.260 | 48.260 | **Reference** | 1.900 | 1.900 | **Reference** |
| E | 15.240 | 15.875 | | 0.600 | 0.625 | |
| E1 | 12.954 | 15.240 | | 0.510 | 0.600 | |
| e1 | 2.540 | 2.540 | **Reference** | 0.100 | 0.100 | **Reference** |
| eA | 14.986 | 16.002 | **Typical** | 0.590 | 0.630 | **Typical** |
| eB | 15.240 | 18.034 | | 0.600 | 0.710 | |
| L | 3.175 | 3.810 | | 0.125 | 0.150 | |
| N | 40 | 40 | | 40 | 40 | |
| S | 1.016 | 2.286 | | 0.040 | 0.090 | |
| S1 | 0.381 | 1.778 | | 0.015 | 0.070 | |

## 22.3 28-Lead Plastic Dual In-line (300 mil) (SP)



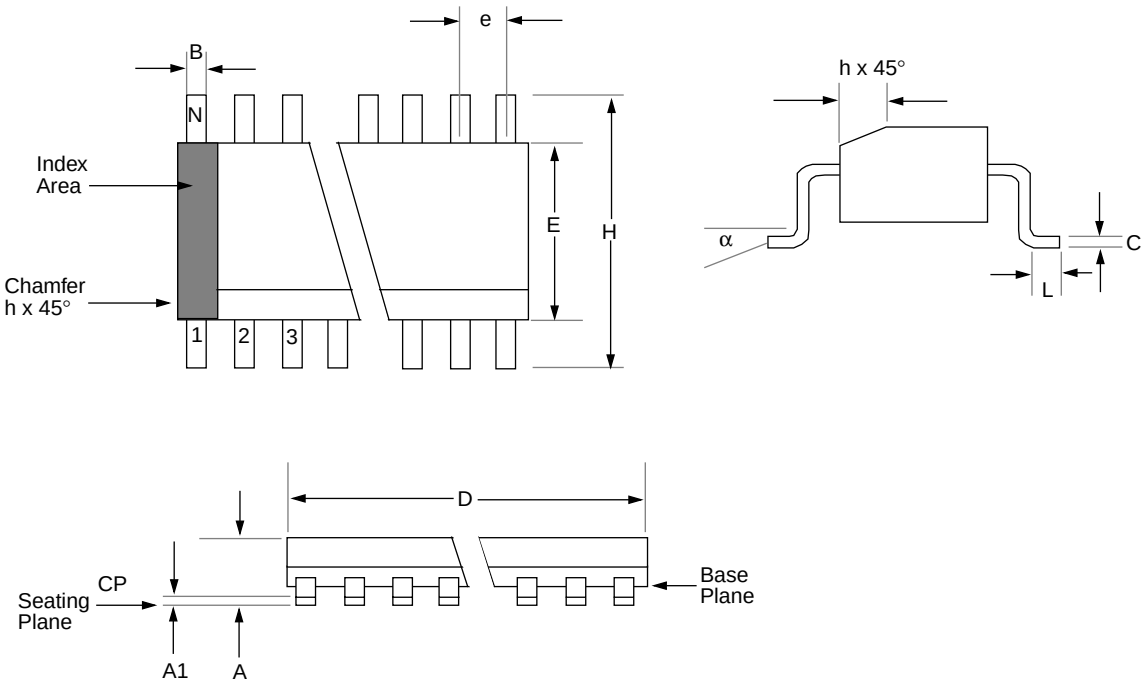

| Package Group: Plastic Dual In-Line (PLA) | | | | | | |
|---|---|---|---|---|---|---|
| | Millimeters | | | Inches | | |
| Symbol | Min | Max | Notes | Min | Max | Notes |
| α | 0° | 10° | | 0° | 10° | |
| A | 3.632 | 4.572 | | 0.143 | 0.180 | |
| A1 | 0.381 | – | | 0.015 | – | |
| A2 | 3.175 | 3.556 | | 0.125 | 0.140 | |
| B | 0.406 | 0.559 | | 0.016 | 0.022 | |
| B1 | 1.016 | 1.651 | **Typical** | 0.040 | 0.065 | **Typical** |
| B2 | 0.762 | 1.016 | **4 places** | 0.030 | 0.040 | **4 places** |
| B3 | 0.203 | 0.508 | **4 places** | 0.008 | 0.020 | **4 places** |
| C | 0.203 | 0.331 | **Typical** | 0.008 | 0.013 | **Typical** |
| D | 34.163 | 35.179 | | 1.385 | 1.395 | |
| D1 | 33.020 | 33.020 | **Reference** | 1.300 | 1.300 | **Reference** |
| E | 7.874 | 8.382 | | 0.310 | 0.330 | |
| E1 | 7.112 | 7.493 | | 0.280 | 0.295 | |
| e1 | 2.540 | 2.540 | **Typical** | 0.100 | 0.100 | **Typical** |
| eA | 7.874 | 7.874 | **Reference** | 0.310 | 0.310 | **Reference** |
| eB | 8.128 | 9.652 | | 0.320 | 0.380 | |
| L | 3.175 | 3.683 | | 0.125 | 0.145 | |
| N | 28 | - | | 28 | - | |
| S | 0.584 | 1.220 | | 0.023 | 0.048 | |

### 22.4 40-Lead Plastic Dual In-line (600 mil) (P)



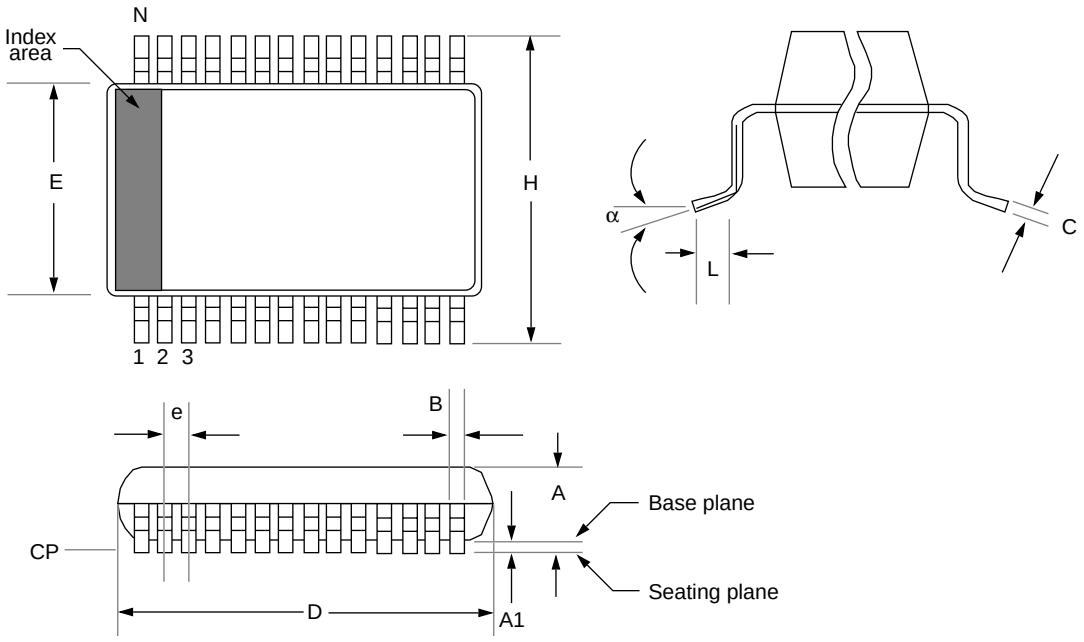

| Package Group: Plastic Dual In-Line (PLA) | | | | | | |
|---|---|---|---|---|---|---|
| | Millimeters | | | Inches | | |
| Symbol | Min | Max | Notes | Min | Max | Notes |
| α | 0° | 10° | | 0° | 10° | |
| A | – | 5.080 | | – | 0.200 | |
| A1 | 0.381 | – | | 0.015 | – | |
| A2 | 3.175 | 4.064 | | 0.125 | 0.160 | |
| B | 0.355 | 0.559 | | 0.014 | 0.022 | |
| B1 | 1.270 | 1.778 | **Typical** | 0.050 | 0.070 | **Typical** |
| C | 0.203 | 0.381 | **Typical** | 0.008 | 0.015 | **Typical** |
| D | 51.181 | 52.197 | | 2.015 | 2.055 | |
| D1 | 48.260 | 48.260 | **Reference** | 1.900 | 1.900 | **Reference** |
| E | 15.240 | 15.875 | | 0.600 | 0.625 | |
| E1 | 13.462 | 13.970 | | 0.530 | 0.550 | |
| e1 | 2.489 | 2.591 | **Typical** | 0.098 | 0.102 | **Typical** |
| eA | 15.240 | 15.240 | **Reference** | 0.600 | 0.600 | **Reference** |
| eB | 15.240 | 17.272 | | 0.600 | 0.680 | |
| L | 2.921 | 3.683 | | 0.115 | 0.145 | |
| N | 40 | 40 | | 40 | 40 | |
| S | 1.270 | – | | 0.050 | – | |
| S1 | 0.508 | – | | 0.020 | – | |

## 22.5 28-Lead Plastic Surface Mount (SOIC - Wide, 300 mil Body) (SO)



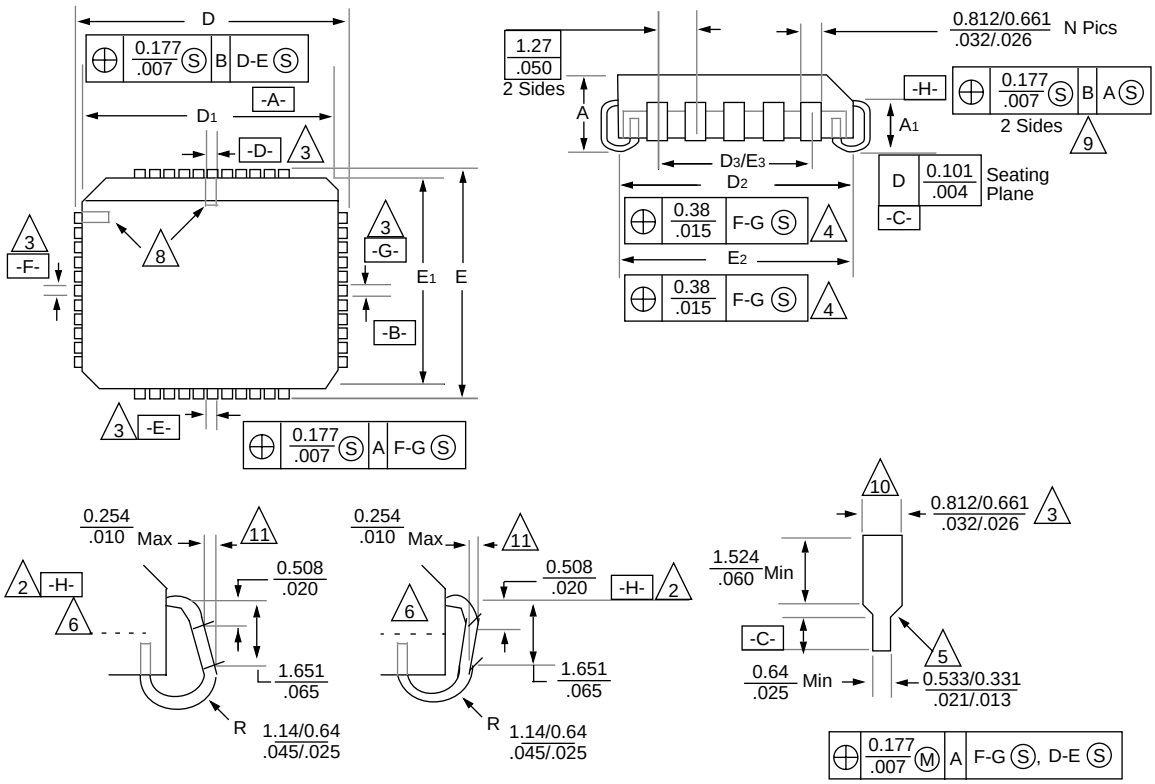

| Package Group: Plastic SOIC (SO) | | | | | | |
|---|---|---|---|---|---|---|
| | Millimeters | | | Inches | | |
| Symbol | Min | Max | Notes | Min | Max | Notes |
| α | 0° | 8° | | 0° | 8° | |
| A | 2.362 | 2.642 | | 0.093 | 0.104 | |
| A1 | 0.101 | 0.300 | | 0.004 | 0.012 | |
| B | 0.355 | 0.483 | | 0.014 | 0.019 | |
| C | 0.241 | 0.318 | | 0.009 | 0.013 | |
| D | 17.703 | 18.085 | | 0.697 | 0.712 | |
| E | 7.416 | 7.595 | | 0.292 | 0.299 | |
| e | 1.270 | 1.270 | **Typical** | 0.050 | 0.050 | **Typical** |
| H | 10.007 | 10.643 | | 0.394 | 0.419 | |
| h | 0.381 | 0.762 | | 0.015 | 0.030 | |
| L | 0.406 | 1.143 | | 0.016 | 0.045 | |
| N | 28 | 28 | | 28 | 28 | |
| CP | – | 0.102 | | – | 0.004 | |

## 22.6 28-Lead Plastic Surface Mount (SSOP - 209 mil Body 5.30 mm) (SS)



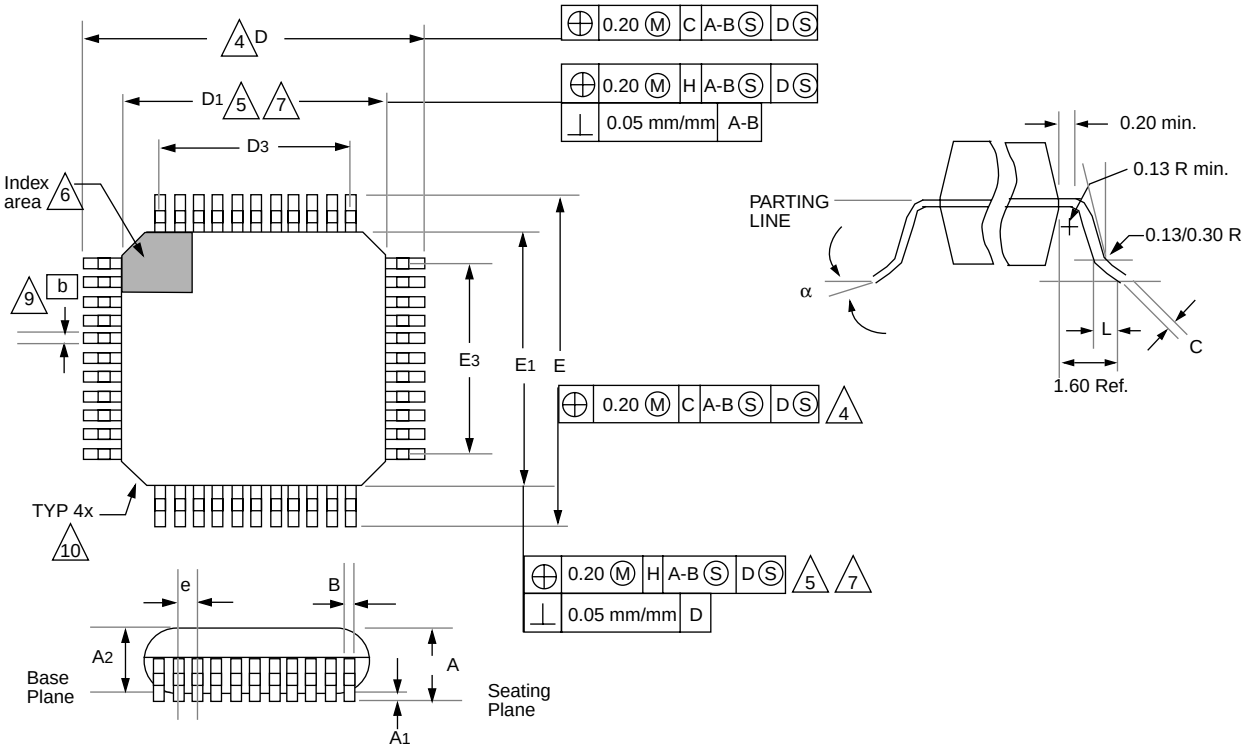

| Package Group: Plastic SSOP | | | | | | |
|---|---|---|---|---|---|---|
| | Millimeters | | | Inches | | |
| Symbol | Min | Max | Notes | Min | Max | Notes |
| α | 0° | 8° | | 0° | 8° | |
| A | 1.730 | 1.990 | | 0.068 | 0.078 | |
| A1 | 0.050 | 0.210 | | 0.002 | 0.008 | |
| B | 0.250 | 0.380 | | 0.010 | 0.015 | |
| C | 0.130 | 0.220 | | 0.005 | 0.009 | |
| D | 10.070 | 10.330 | | 0.396 | 0.407 | |
| E | 5.200 | 5.380 | | 0.205 | 0.212 | |
| e | 0.650 | 0.650 | **Reference** | 0.026 | 0.026 | **Reference** |
| H | 7.650 | 7.900 | | 0.301 | 0.311 | |
| L | 0.550 | 0.950 | | 0.022 | 0.037 | |
| N | 28 | 28 | | 28 | 28 | |
| CP | - | 0.102 | | - | 0.004 | |

## 22.7 44-Lead Plastic Leaded Chip Carrier (Square)(PLCC)



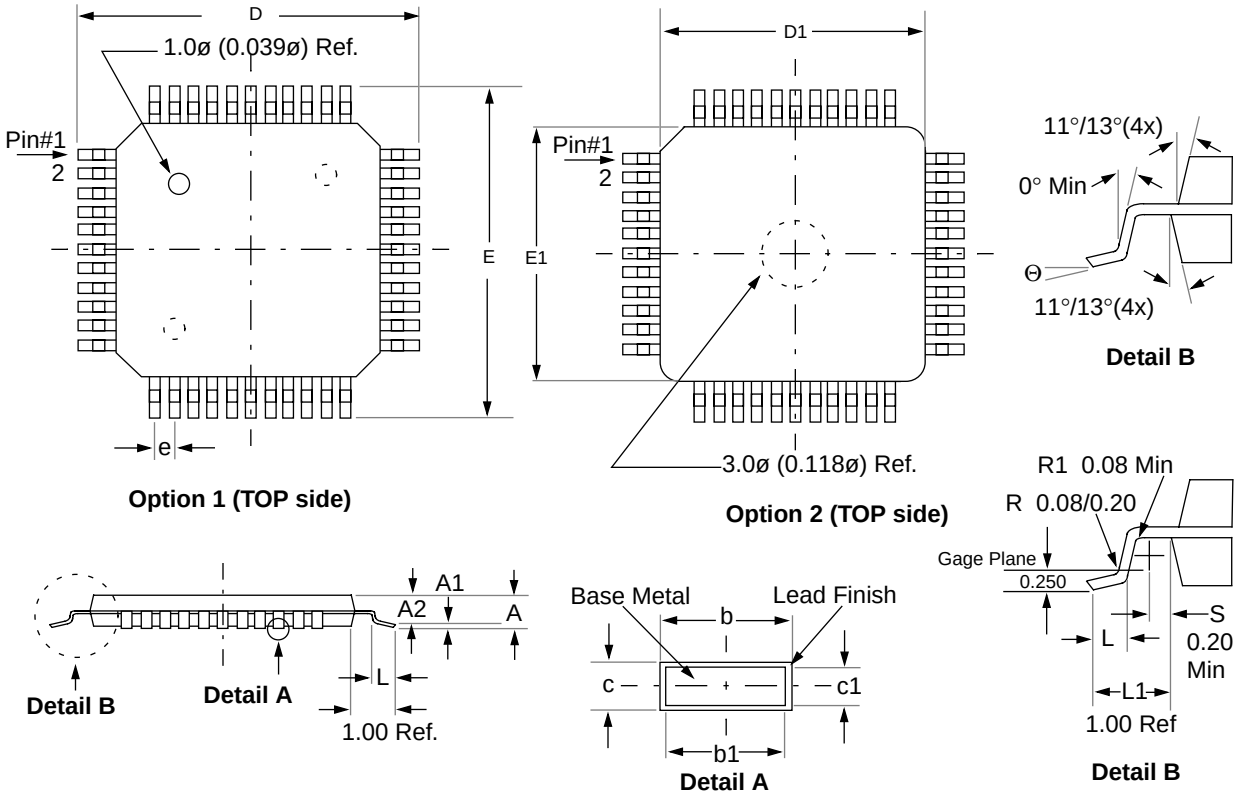

| Package Group: Plastic Leaded Chip Carrier (PLCC) | | | | | | |
|---|---|---|---|---|---|---|
| | Millimeters | | | Inches | | |
| Symbol | Min | Max | Notes | Min | Max | Notes |
| A | 4.191 | 4.572 | | 0.165 | 0.180 | |
| A1 | 2.413 | 2.921 | | 0.095 | 0.115 | |
| D | 17.399 | 17.653 | | 0.685 | 0.695 | |
| D1 | 16.510 | 16.663 | | 0.650 | 0.656 | |
| D2 | 15.494 | 16.002 | | 0.610 | 0.630 | |
| D3 | 12.700 | 12.700 | **Reference** | 0.500 | 0.500 | **Reference** |
| E | 17.399 | 17.653 | | 0.685 | 0.695 | |
| E1 | 16.510 | 16.663 | | 0.650 | 0.656 | |
| E2 | 15.494 | 16.002 | | 0.610 | 0.630 | |
| E3 | 12.700 | 12.700 | **Reference** | 0.500 | 0.500 | **Reference** |
| N | 44 | 44 | | 44 | 44 | |
| CP | – | 0.102 | | – | 0.004 | |
| LT | 0.203 | 0.381 | | 0.008 | 0.015 | |

## 22.8 44-Lead Plastic Surface Mount (MQFP 10x10 mm Body 1.6/0.15 mm Lead Form) (PQ)




| Package Group: Plastic MQFP | | | | | | |
|---|---|---|---|---|---|---|
| | Millimeters | | | Inches | | |
| Symbol | Min | Max | Notes | Min | Max | Notes |
| α | 0° | 7° | | 0° | 7° | |
| A | 2.000 | 2.350 | | 0.078 | 0.093 | |
| A1 | 0.050 | 0.250 | | 0.002 | 0.010 | |
| A2 | 1.950 | 2.100 | | 0.768 | 0.083 | |
| b | 0.300 | 0.450 | **Typical** | 0.011 | 0.018 | **Typical** |
| C | 0.150 | 0.180 | | 0.006 | 0.007 | |
| D | 12.950 | 13.450 | | 0.510 | 0.530 | |
| D1 | 9.900 | 10.100 | | 0.390 | 0.398 | |
| D3 | 8.000 | 8.000 | **Reference** | 0.315 | 0.315 | **Reference** |
| E | 12.950 | 13.450 | | 0.510 | 0.530 | |
| E1 | 9.900 | 10.100 | | 0.390 | 0.398 | |
| E3 | 8.000 | 8.000 | **Reference** | 0.315 | 0.315 | **Reference** |
| e | 0.800 | 0.800 | | 0.031 | 0.032 | |
| L | 0.730 | 1.030 | | 0.028 | 0.041 | |
| N | 44 | 44 | | 44 | 44 | |
| CP | 0.102 | – | | 0.004 | – | |

## 22.9 44-Lead Plastic Surface Mount (TQFP 10x10 mm Body 1.0/0.10 mm Lead Form) (TQ)




| Package Group: Plastic TQFP | | | | | | |
|---|---|---|---|---|---|---|
| | Millimeters | | | Inches | | |
| Symbol | Min | Max | Notes | Min | Max | Notes |
| A | 1.00 | 1.20 | | 0.039 | 0.047 | |
| A1 | 0.05 | 0.15 | | 0.002 | 0.006 | |
| A2 | 0.95 | 1.05 | | 0.037 | 0.041 | |
| D | 11.75 | 12.25 | | 0.463 | 0.482 | |
| D1 | 9.90 | 10.10 | | 0.390 | 0.398 | |
| E | 11.75 | 12.25 | | 0.463 | 0.482 | |
| E1 | 9.90 | 10.10 | | 0.390 | 0.398 | |
| L | 0.45 | 0.75 | | 0.018 | 0.030 | |
| e | 0.80 BSC | | | 0.031 BSC | | |
| b | 0.30 | 0.45 | | 0.012 | 0.018 | |
| b1 | 0.30 | 0.40 | | 0.012 | 0.016 | |
| c | 0.09 | 0.20 | | 0.004 | 0.008 | |
| c1 | 0.09 | 0.16 | | 0.004 | 0.006 | |
| N | 44 | 44 | | 44 | 44 | |
| Θ | 0° | 7° | | 0° | 7° | |

Note 1: Dimensions D1 and E1 do not include mold protrusion. Allowable mold protrusion is 0.25m/m (0.010”) per side. D1 and E1 dimensions including mold mismatch.

2: Dimension “b” does not include Dambar protrusion, allowable Dambar protrusion shall be 0.08m/m (0.003”)max.

3: This outline conforms to JEDEC MS-026.

### 22.10 Package Marking Information

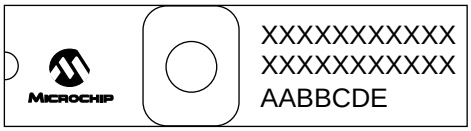
28-Lead SSOP


Example

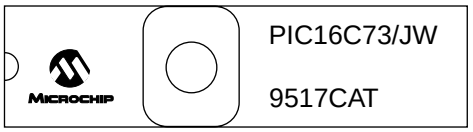

28-Lead PDIP (Skinny DIP)


Example

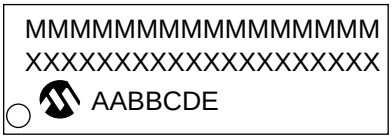

28-Lead Side Brazed Skinny Windowed



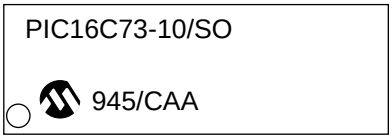

Example




28-Lead SOIC




Example



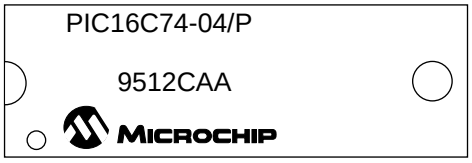

| | | |
|---|---|---|
| Legend: | MM...M | Microchip part number information |
| | XX...X | Customer specific information* |
| | AA | Year code (last 2 digits of calender year) |
| | BB | Week code (week of January 1 is week '01') |
| | C | Facility code of the plant at which wafer is manufactured.<br>C = Chandler, Arizona, U.S.A.<br>S = Tempe, Arizona, U.S.A. |
| | $D_1$ | Mask revision number for microcontroller |
| | E | Assembly code of the plant or country of origin in which part was assembled. |
| **Note**: | In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information. | |

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

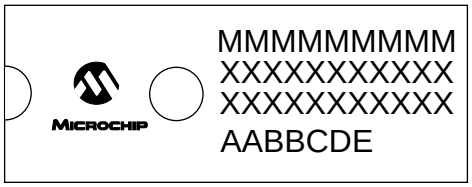
**Package Marking Information (Cont'd)**

40-Lead PDIP

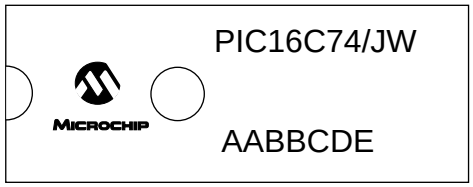

Example



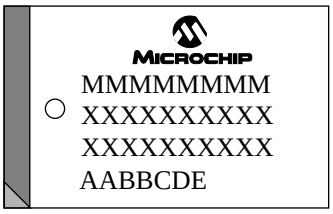

40-Lead CERDIP Windowed



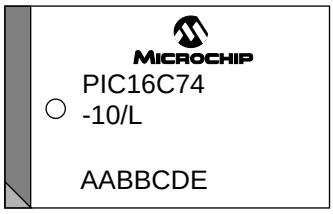

Example



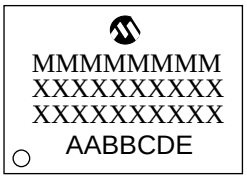

44-Lead PLCC



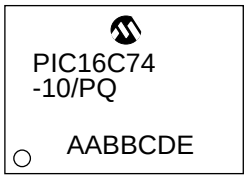

Example




44-Lead MQFP




Example

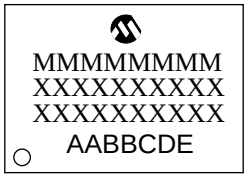



| | | |
|---|---|---|
| Legend: | MM...M | Microchip part number information |
| | XX...X | Customer specific information* |
| | AA | Year code (last 2 digits of calender year) |
| | BB | Week code (week of January 1 is week '01') |
| | C | Facility code of the plant at which wafer is manufactured.<br>C = Chandler, Arizona, U.S.A.<br>S = Tempe, Arizona, U.S.A. |
| | $D_1$ | Mask revision number for microcontroller |
| | E | Assembly code of the plant or country of origin in which part was assembled. |
| **Note**: | In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information. | |

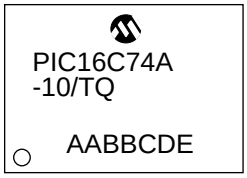
* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

**Package Marking Information (Cont'd)**

44-Lead TQFP




Example




| | | |
|---|---|---|
| Legend: | MM...M | Microchip part number information |
| | XX...X | Customer specific information* |
| | AA | Year code (last 2 digits of calender year) |
| | BB | Week code (week of January 1 is week '01') |
| | C | Facility code of the plant at which wafer is manufactured.<br>C = Chandler, Arizona, U.S.A.<br>S = Tempe, Arizona, U.S.A. |
| | $D_1$ | Mask revision number for microcontroller |
| | E | Assembly code of the plant or country of origin in which part was assembled. |
| **Note**: | In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information. | |

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

## APPENDIX A:

The following are the list of modifications over the PIC16C5X microcontroller family:

1. Instruction word length is increased to 14-bits. This allows larger page sizes both in program memory (2K now as opposed to 512 before) and register file (128 bytes now versus 32 bytes before).
2. A PC high latch register (PCLATH) is added to handle program memory paging. Bits PA2, PA1, PA0 are removed from STATUS register.
3. Data memory paging is redefined slightly. STATUS register is modified.
4. Four new instructions have been added: `RETURN, RETFIE, ADDLW`, and `SUBLW`. Two instructions `TRIS` and `OPTION` are being phased out although they are kept for compatibility with PIC16C5X.
5. OPTION and TRIS registers are made addressable.
6. Interrupt capability is added. Interrupt vector is at 0004h.
7. Stack size is increased to 8 deep.
8. Reset vector is changed to 0000h.
9. Reset of all registers is revisited. Five different reset (and wake-up) types are recognized. Registers are reset differently.
10. Wake up from SLEEP through interrupt is added.
11. Two separate timers, Oscillator Start-up Timer (OST) and Power-up Timer (PWRT) are included for more reliable power-up. These timers are invoked selectively to avoid unnecessary delays on power-up and wake-up.
12. PORTB has weak pull-ups and interrupt on change feature.
13. T0CKI pin is also a port pin (RA4) now.
14. FSR is made a full eight bit register.
15. “In-circuit serial programming” is made possible. The user can program PIC16CXX devices using only five pins: $V_{DD}$, $V_{SS}$, $\overline{MCLR}/V_{PP}$, RB6 (clock) and RB7 (data in/out).
16. PCON status register is added with a Power-on Reset status bit ($\overline{POR}$).
17. Code protection scheme is enhanced such that portions of the program memory can be protected, while the remainder is unprotected.
18. Brown-out protection circuitry has been added. Controlled by configuration word bit BODEN. Brown-out reset ensures the device is placed in a reset condition if $V_{DD}$ dips below a fixed setpoint.

## APPENDIX B: COMPATIBILITY

To convert code written for PIC16C5X to PIC16CXX, the user should take the following steps:

1. Remove any program memory page select operations (PA2, PA1, PA0 bits) for `CALL`, `GOTO`.
2. Revisit any computed jump operations (write to PC or add to PC, etc.) to make sure page bits are set properly under the new scheme.
3. Eliminate any data memory page switching. Redefine data variables to reallocate them.
4. Verify all writes to STATUS, OPTION, and FSR registers since these have changed.
5. Change reset vector to 0000h.

## APPENDIX C: WHAT'S NEW

Added the following devices:

- PIC16C76
- PIC16C77

Removed the PIC16C710, PIC16C71, PIC16C711 from this datasheet.

Added PIC16C76 and PIC16C77 devices. The PIC16C76/77 devices have 368 bytes of data memory distributed in 4 banks and 8K of program memory in 4 pages. These two devices have an enhanced SPI that supports both clock phase and polarity. The USART has been enhanced.

When upgrading to the PIC16C76/77 please note that the upper 16 bytes of data memory in banks 1,2, and 3 are mapped into bank 0. This may require relocation of data memory usage in the user application code.

Added Q-cycle definitions to the Instruction Set Summary section.

## APPENDIX D: WHAT'S CHANGED

Minor changes, spelling and grammatical changes.

Added the following note to the USART section. This note applies to all devices except the PIC16C76 and PIC16C77.

For the PIC16C73/73A/74/74A the asynchronous high speed mode (BRGH = 1) may experience a high rate of receive errors. It is recommended that BRGH = 0. If you desire a higher baud rate than BRGH = 0 can support, refer to the device errata for additional information or use the PIC16C76/77.

Divided SPI section into SPI for the PIC16C76/77 and SPI for all other devices.

# APPENDIX E: PIC16/17 MICROCONTROLLERS

## E.1 PIC12CXXX Family of Devices

| | | PIC12C508 | PIC12C509 | PIC12C671 | PIC12C672 |
|---|---|---|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 4 | 4 | 4 | 4 |
| **Memory** | EPROM Program Memory | 512 x 12 | 1024 x 12 | 1024 x 14 | 2048 x 14 |
| | Data Memory (bytes) | 25 | 41 | 128 | 128 |
| **Peripherals** | Timer Module(s) | TMR0 | TMR0 | TMR0 | TMR0 |
| | A/D Converter (8-bit) Channels | — | — | 4 | 4 |
| **Features** | Wake-up from SLEEP on pin change | Yes | Yes | Yes | Yes |
| | I/O Pins | 5 | 5 | 5 | 5 |
| | Input Pins | 1 | 1 | 1 | 1 |
| | Internal Pull-ups | Yes | Yes | Yes | Yes |
| | Voltage Range (Volts) | 2.5-5.5 | 2.5-5.5 | 2.5-5.5 | 2.5-5.5 |
| | In-Circuit Serial Programming | Yes | Yes | Yes | Yes |
| | Number of Instructions | 33 | 33 | 35 | 35 |
| | Packages | 8-pin DIP, SOIC | 8-pin DIP, SOIC | 8-pin DIP, SOIC | 8-pin DIP, SOIC |

All PIC12C5XX devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability.
All PIC12C5XX devices use serial programming with data pin GP1 and clock pin GP0.

## E.2 PIC14C000 Family of Devices

| | | PIC14C000 |
|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 20 |
| **Memory** | EPROM Program Memory (x14 words) | 4K |
| | Data Memory (bytes) | 192 |
| | Timer Module(s) | TMR0<br>ADTMR |
| **Peripherals** | Serial Port(s) (SPI/$I^2C$, USART) | $I^2C$ with SMBus Support |
| **Features** | Slope A/D Converter Channels | 8 External; 6 Internal |
| | Interrupt Sources | 11 |
| | I/O Pins | 22 |
| | Voltage Range (Volts) | 2.7-6.0 |
| | In-Circuit Serial Programming | Yes |
| | Additional On-chip Features | Internal 4MHz Oscillator, Bandgap Reference,Temperature Sensor, Calibration Factors, Low Voltage Detector, SLEEP, HIBERNATE, Comparators with Programmable References (2) |
| | Packages | 28-pin DIP (.300 mil), SOIC, SSOP |

## E.3 PIC16C15X Family of Devices

| | | PIC16C154 | PIC16CR154 | PIC16C156 | PIC16CR156 | PIC16C158 | PIC16CR158 |
|---|---|---|---|---|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 20 | 20 | 20 | 20 | 20 | 20 |
| **Memory** | EPROM Program Memory (x12 words) | 512 | — | 1K | — | 2K | — |
| | ROM Program Memory (x12 words) | — | 512 | — | 1K | — | 2K |
| | RAM Data Memory (bytes) | 25 | 25 | 25 | 25 | 73 | 73 |
| **Peripherals** | Timer Module(s) | TMR0 | TMR0 | TMR0 | TMR0 | TMR0 | TMR0 |
| **Features** | I/O Pins | 12 | 12 | 12 | 12 | 12 | 12 |
| | Voltage Range (Volts) | 3.0-5.5 | 2.5-5.5 | 3.0-5.5 | 2.5-5.5 | 3.0-5.5 | 2.5-5.5 |
| | Number of Instructions | 33 | 33 | 33 | 33 | 33 | 33 |
| | Packages | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP |

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

## E.4 PIC16C5X Family of Devices

| | | PIC16C52 | PIC16C54 | PIC16C54A | PIC16CR54A | PIC16C55 | PIC16C56 |
|---|---|---|---|---|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 4 | 20 | 20 | 20 | 20 | 20 |
| **Memory** | EPROM Program Memory (x12 words) | 384 | 512 | 512 | — | 512 | 1K |
| | ROM Program Memory (x12 words) | — | — | — | 512 | — | — |
| | RAM Data Memory (bytes) | 25 | 25 | 25 | 25 | 24 | 25 |
| **Peripherals** | Timer Module(s) | TMR0 | TMR0 | TMR0 | TMR0 | TMR0 | TMR0 |
| **Features** | I/O Pins | 12 | 12 | 12 | 12 | 20 | 12 |
| | Voltage Range (Volts) | 2.5-6.25 | 2.5-6.25 | 2.0-6.25 | 2.0-6.25 | 2.5-6.25 | 2.5-6.25 |
| | Number of Instructions | 33 | 33 | 33 | 33 | 33 | 33 |
| | Packages | 18-pin DIP, SOIC | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP | 28-pin DIP, SOIC, SSOP | 18-pin DIP, SOIC; 20-pin SSOP |

| | | PIC16C57 | PIC16CR57B | PIC16C58A | PIC16CR58A |
|---|---|---|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 20 | 20 | 20 | 20 |
| **Memory** | EPROM Program Memory (x12 words) | 2K | — | 2K | — |
| | ROM Program Memory (x12 words) | — | 2K | — | 2K |
| | RAM Data Memory (bytes) | 72 | 72 | 73 | 73 |
| **Peripherals** | Timer Module(s) | TMR0 | TMR0 | TMR0 | TMR0 |
| **Features** | I/O Pins | 20 | 20 | 12 | 12 |
| | Voltage Range (Volts) | 2.5-6.25 | 2.5-6.25 | 2.0-6.25 | 2.5-6.25 |
| | Number of Instructions | 33 | 33 | 33 | 33 |
| | Packages | 28-pin DIP, SOIC, SSOP | 28-pin DIP, SOIC, SSOP | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP |

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer (except PIC16C52), selectable code protect and high I/O current capability.

## E.5 PIC16C55X Family of Devices

| | | PIC16C554 | PIC16C556(1) | PIC16C558 |
|---|---|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 20 | 20 | 20 |
| **Memory** | EPROM Program Memory (x14 words) | 512 | 1K | 2K |
| | Data Memory (bytes) | 80 | 80 | 128 |
| **Peripherals** | Timer Module(s) | TMR0 | TMR0 | TMR0 |
| | Comparators(s) | — | — | — |
| | Internal Reference Voltage | — | — | — |
| **Features** | Interrupt Sources | 3 | 3 | 3 |
| | I/O Pins | 13 | 13 | 13 |
| | Voltage Range (Volts) | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 |
| | Brown-out Reset | — | — | — |
| | Packages | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP |

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C5XX Family devices use serial programming with clock pin RB6 and data pin RB7.

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

## E.6 PIC16C62X and PIC16C64X Family of Devices

| | | PIC16C620 | PIC16C621 | PIC16C622 | PIC16C642 | PIC16C662 |
|---|---|---|---|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 20 | 20 | 20 | 20 | 20 |
| **Memory** | EPROM Program Memory (x14 words) | 512 | 1K | 2K | 4K | 4K |
| | Data Memory (bytes) | 80 | 80 | 128 | 176 | 176 |
| **Peripherals** | Timer Module(s) | TMR0 | TMR0 | TMR0 | TMR0 | TMR0 |
| | Comparators(s) | 2 | 2 | 2 | 2 | 2 |
| | Internal Reference Voltage | Yes | Yes | Yes | Yes | Yes |
| **Features** | Interrupt Sources | 4 | 4 | 4 | 4 | 5 |
| | I/O Pins | 13 | 13 | 13 | 22 | 33 |
| | Voltage Range (Volts) | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 | 3.0-6.0 | 3.0-6.0 |
| | Brown-out Reset | Yes | Yes | Yes | Yes | Yes |
| | Packages | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP | 18-pin DIP, SOIC; 20-pin SSOP | 28-pin PDIP, SOIC, Windowed CDIP | 40-pin PDIP, Windowed CDIP; 44-pin PLCC, MQFP |

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C62X and PIC16C64X Family devices use serial programming with clock pin RB6 and data pin RB7.

## E.7 PIC16C6X Family of Devices

| | | PIC16C61 | PIC16C62A | PIC16CR62 | PIC16C63 | PIC16CR63 |
|---|---|---|---|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 20 | 20 | 20 | 20 | 20 |
| **Memory** | EPROM Program Memory (x14 words) | 1K | 2K | — | 4K | — |
| | ROM Program Memory (x14 words) | — | — | 2K | — | 4K |
| | Data Memory (bytes) | 36 | 128 | 128 | 192 | 192 |
| **Peripherals** | Timer Module(s) | TMR0 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 |
| | Capture/Compare/ PWM Module(s) | — | 1 | 1 | 2 | 2 |
| | Serial Port(s) (SPI/$I^2C$, USART) | — | SPI/$I^2C$ | SPI/$I^2C$ | SPI/$I^2C$, USART | SPI/$I^2C$ USART |
| | Parallel Slave Port | — | — | — | — | — |
| **Features** | Interrupt Sources | 3 | 7 | 7 | 10 | 10 |
| | I/O Pins | 13 | 22 | 22 | 22 | 22 |
| | Voltage Range (Volts) | 3.0-6.0 | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 |
| | In-Circuit Serial Programming | Yes | Yes | Yes | Yes | Yes |
| | Brown-out Reset | — | Yes | Yes | Yes | Yes |
| | Packages | 18-pin DIP, SO | 28-pin SDIP, SOIC, SSOP | 28-pin SDIP, SOIC, SSOP | 28-pin SDIP, SOIC | 28-pin SDIP, SOIC |

| | | PIC16C64A | PIC16CR64 | PIC16C65A | PIC16CR65 | PIC16C66 | PIC16C67 |
|---|---|---|---|---|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 20 | 20 | 20 | 20 | 20 | 20 |
| **Memory** | EPROM Program Memory (x14 words) | 2K | — | 4K | — | 8K | 8K |
| | ROM Program Memory (x14 words) | — | 2K | — | 4K | — | — |
| | Data Memory (bytes) | 128 | 128 | 192 | 192 | 368 | 368 |
| **Peripherals** | Timer Module(s) | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 |
| | Capture/Compare/PWM Module(s) | 1 | 1 | 2 | 2 | 2 | 2 |
| | Serial Port(s) (SPI/$I^2C$, USART) | SPI/$I^2C$ | SPI/$I^2C$ | SPI/$I^2C$, USART | SPI/$I^2C$, USART | SPI/$I^2C$, USART | SPI/$I^2C$, USART |
| | Parallel Slave Port | Yes | Yes | Yes | Yes | — | Yes |
| **Features** | Interrupt Sources | 8 | 8 | 11 | 11 | 10 | 11 |
| | I/O Pins | 33 | 33 | 33 | 33 | 22 | 33 |
| | Voltage Range (Volts) | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 | 2.5-6.0 |
| | In-Circuit Serial Programming | Yes | Yes | Yes | Yes | Yes | Yes |
| | Brown-out Reset | Yes | Yes | Yes | Yes | Yes | Yes |
| | 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 | 28-pin SDIP, SOIC | 40-pin DIP; 44-pin PLCC, MQFP, TQFP |

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C6X Family devices use serial programming with clock pin RB6 and data pin RB7.

## E.8 PIC16C8X Family of Devices

| | | PIC16F83 | PIC16CR83 | PIC16F84 | PIC16CR84 |
|---|---|---|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 10 | 10 | 10 | 10 |
| **Memory** | Flash Program Memory | 512 | — | 1K | — |
| | EEPROM Program Memory | — | — | — | — |
| | ROM Program Memory | — | 512 | — | 1K |
| | Data Memory (bytes) | 36 | 36 | 68 | 68 |
| | Data EEPROM (bytes) | 64 | 64 | 64 | 64 |
| **Peripherals** | Timer Module(s) | TMR0 | TMR0 | TMR0 | TMR0 |
| **Features** | Interrupt Sources | 4 | 4 | 4 | 4 |
| | I/O Pins | 13 | 13 | 13 | 13 |
| | Voltage Range (Volts) | 2.0-6.0 | 2.0-6.0 | 2.0-6.0 | 2.0-6.0 |
| | Packages | 18-pin DIP, SOIC | 18-pin DIP, SOIC | 18-pin DIP, SOIC | 18-pin DIP, SOIC |

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C8X Family devices use serial programming with clock pin RB6 and data pin RB7.

## E.9 PIC16C9XX Family Of Devices

| | | PIC16C923 | PIC16C924 |
|---|---|---|---|
| **Clock** | Maximum Frequency of Operation (MHz) | 8 | 8 |
| **Memory** | EPROM Program Memory | 4K | 4K |
| | Data Memory (bytes) | 176 | 176 |
| **Peripherals** | Timer Module(s) | TMR0, TMR1, TMR2 | TMR0, TMR1, TMR2 |
| | Capture/Compare/PWM Module(s) | 1 | 1 |
| | Serial Port(s) (SPI/$I^2C$, USART) | SPI/$I^2C$ | SPI/$I^2C$ |
| | Parallel Slave Port | — | — |
| | A/D Converter (8-bit) Channels | — | 5 |
| | LCD Module | 4 Com, 32 Seg | 4 Com, 32 Seg |
| **Features** | Interrupt Sources | 8 | 9 |
| | I/O Pins | 25 | 25 |
| | Input Pins | 27 | 27 |
| | Voltage Range (Volts) | 3.0-6.0 | 3.0-6.0 |
| | In-Circuit Serial Programming | Yes | Yes |
| | Brown-out Reset | — | — |
| | Packages | 64-pin SDIP(1), TQFP; 68-pin PLCC, Die | 64-pin SDIP(1), TQFP; 68-pin PLCC, Die |

All PIC16/17 Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16C9XX Family devices use serial programming with clock pin RB6 and data pin RB7.

## 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 |

**NOTES:**

# INDEX

## L



## M



## O



## P

## R

## S

## LIST OF EXAMPLES

## LIST OF FIGURES

## LIST OF TABLES

## ON-LINE SUPPORT

Microchip provides two methods of on-line support. These are the Microchip BBS and the Microchip World Wide Web (WWW) site.

Use Microchip's Bulletin Board Service (BBS) to get current information and help about Microchip products. Microchip provides the BBS communication channel for you to use in extending your technical staff with microcontroller and memory experts.

To provide you with the most responsive service possible, the Microchip systems team monitors the BBS, posts the latest component data and software tool updates, provides technical help and embedded systems insights, and discusses how Microchip products provide project solutions.

The web site, like the BBS, is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape or Microsoft Explorer. Files are also available for FTP download from our FTP site.

### Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite Internet browser to attach to:

**www.microchip.com**

The file transfer site is available by using an FTP service to connect to:

**ftp://ftp.futureone.com/pub/microchip**

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products

### Connecting to the Microchip BBS

Connect worldwide to the Microchip BBS using either the Internet or the CompuServe® communications network.

**Internet:**

You can telnet or ftp to the Microchip BBS at the address: **mchipbbs.microchip.com**

**CompuServe Communications Network:**

When using the BBS via the Compuserve Network, in most cases, a local call is your only expense. The Microchip BBS connection does not use CompuServe membership services, therefore you do not need CompuServe membership to join Microchip's BBS. There is no charge for connecting to the Microchip BBS. The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allow multiple users various baud rates depending on the local point of access.

The following connect procedure applies in most locations.

1. Set your modem to 8-bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
2. Dial your local CompuServe access number.
3. Depress the **<Enter>** key and a garbage string will appear because CompuServe is expecting a 7E1 setting.
4. Type +, depress the **<Enter>** key and "`Host Name:`" will appear.
5. Type MCHIPBBS, depress the **<Enter>** key and you will be connected to the Microchip BBS.

In the United States, to find the CompuServe phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with "`Host Name:`", type `NETWORK`, depress the **<Enter>** key and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 723-1550 for your local CompuServe number.

Microchip regularly uses the Microchip BBS to distribute technical information, application notes, source code, errata sheets, bug reports, and interim patches for Microchip systems software products. For each SIG, a moderator monitors, scans, and approves or disapproves files submitted to the SIG. No executable files are accepted from the user community in general to limit the spread of computer viruses.

### Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits.The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and

1-602-786-7302 for the rest of the world.

970301

## READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (602) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: **PIC16C6X** Literature Number: **DS30390E**

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this data sheet easy to follow? If not, why?

4. What additions to the data sheet do you think would enhance the structure and subject?

5. What deletions from the data sheet could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

8. How would you improve our software, systems, and silicon products?

## PIC16C7X PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery refer to the factory or the listed sales office.

**PART NO. -XX X /XX XXX**

| | | | |
|---|---|---|---|
| **Pattern:** | QTP, SQTP, Code or Special Requirements | | |
| **Package:** | JW | = | Windowed CERDIP |
| | PQ | = | MQFP (Metric PQFP) |
| | TQ | = | TQFP (Thin Quad Flatpack) |
| | SO | = | SOIC |
| | SP | = | Skinny plastic dip |
| | P | = | PDIP |
| | L | = | PLCC |
| | SS | = | SSOP |
| **Temperature Range:** | - | = | 0°C to +70°C |
| | I | = | -40°C to +85°C |
| | E | = | -40°C to +125°C |
| **Frequency Range:** | 04 | = | 200 kHz (PIC16C7X-04) |
| | 04 | = | 4 MHz |
| | 10 | = | 10 MHz |
| | 20 | = | 20 MHz |
| **Device** | PIC16C7X | :VDD range 4.0V to 6.0V | |
| | PIC16C7XT | :VDD range 4.0V to 6.0V (Tape/Reel) | |
| | PIC16LC7X | :VDD range 2.5V to 6.0V | |
| | PIC16LC7XT | :VDD range 2.5V to 6.0V (Tape/Reel) | |

**Examples**

a) PIC16C72 - 04/P 301 Commercial Temp., PDIP Package, 4 MHz, normal VDD limits, QTP pattern #301

b) PIC16LC76 - 041/SO Industrial Temp., SOIC package, 4 MHz, extended VDD limits

c) PIC16C74A - 10E/P Automotive Temp., PDIP package, 10 MHz, normal VDD limits

* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type (including LC devices).

### Sales and Support

Products supported by a preliminary Data Sheet may possibly have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. The Microchip Website at www.microchip.com
2. Your local Microchip sales office (see following page)
3. The Microchip Corporate Literature Center U.S. FAX: (602) 786-7277
4. The Microchip's Bulletin Board, via your local CompuServe number (CompuServe membership NOT required).

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.
For latest version information and upgrade kits for Microchip Development Tools, please call 1-800-755-2345 or 1-602-786-7302.

**Note the following details of the code protection feature on PICmicro® MCUs.**

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable”.
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.



DNV Certification, Inc. USA — ANSI-RAB QMS ACCREDITED — DNV MSC The Netherlands Accredited by the RvA — DNV — ISO 9001 / QS-9000 REGISTERED FIRM

*Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company’s quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001 certified.*

# WORLDWIDE SALES AND SERVICE

## AMERICAS

**Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200 Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: http://www.microchip.com

**Rocky Mountain**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-7456

**Atlanta**
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

**Boston**
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

**Chicago**
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

**Dallas**
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

**Detroit**
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

**Kokomo**
2767 S. Albright Road
Kokomo, Indiana 46902
Tel: 765-864-8360 Fax: 765-864-8387

**Los Angeles**
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

**New York**
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

**San Jose**
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

**Toronto**
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

## ASIA/PACIFIC

**Australia**
Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

**China - Beijing**
Microchip Technology Consulting (Shanghai) Co., Ltd., Beijing Liaison Office
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

**China - Chengdu**
Microchip Technology Consulting (Shanghai) Co., Ltd., Chengdu Liaison Office
Rm. 2401, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-6766200 Fax: 86-28-6766599

**China - Fuzhou**
Microchip Technology Consulting (Shanghai) Co., Ltd., Fuzhou Liaison Office
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

**China - Shanghai**
Microchip Technology Consulting (Shanghai) Co., Ltd.
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

**China - Shenzhen**
Microchip Technology Consulting (Shanghai) Co., Ltd., Shenzhen Liaison Office
Rm. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-2350361 Fax: 86-755-2366086

**Hong Kong**
Microchip Technology Hongkong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

**India**
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

**Japan**
Microchip Technology Japan K.K.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

**Korea**
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5934

**Singapore**
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-334-8870 Fax: 65-334-8850

**Taiwan**
Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

## EUROPE

**Denmark**
Microchip Technology Nordic ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

**France**
Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

**Germany**
Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

**Italy**
Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

**United Kingdom**
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

01/18/02