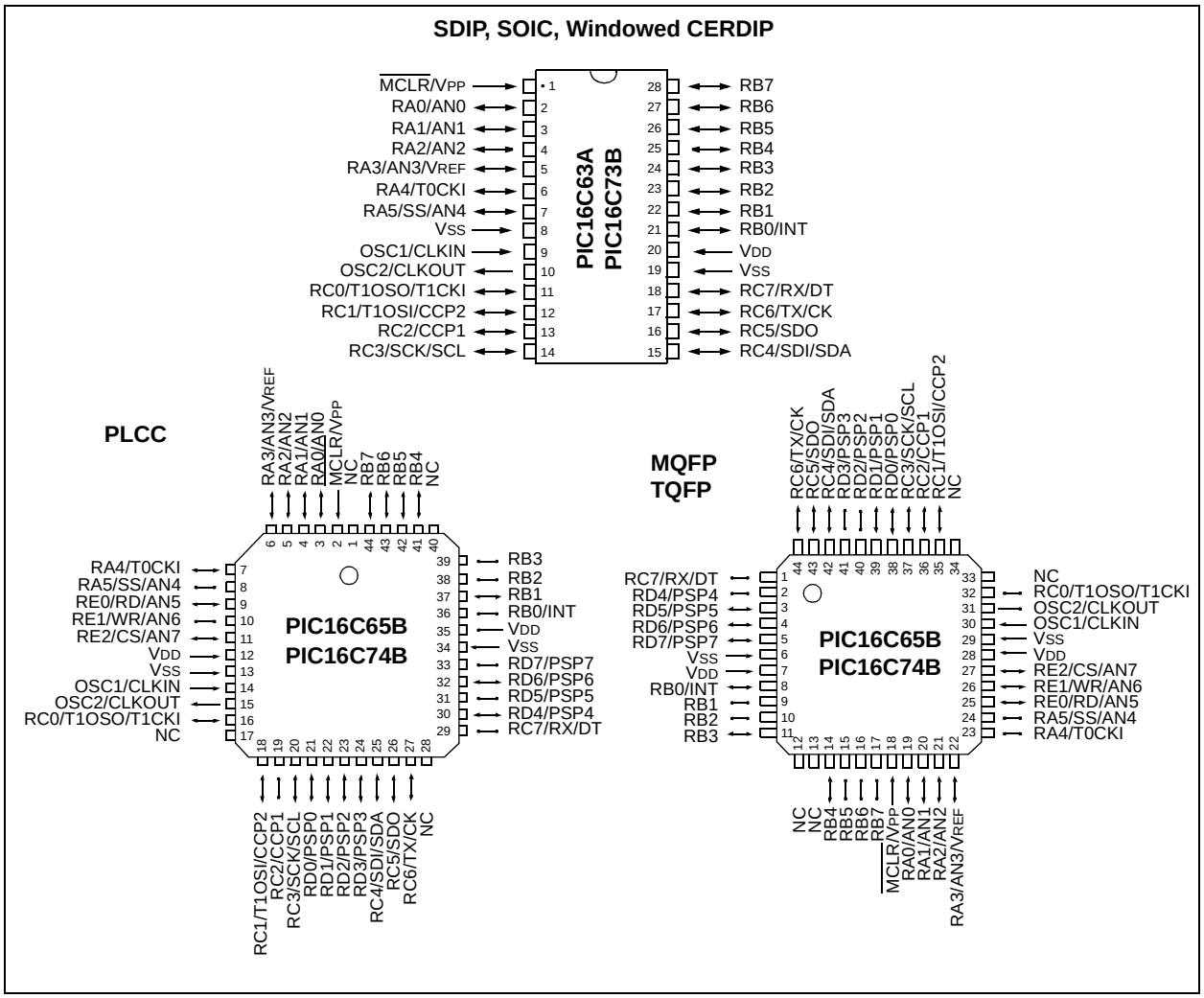



| Key Features<br>PIC® Mid-Range MCU Family Reference Manual (DS33023) | PIC16C63A | PIC16C65B | PIC16C73B | PIC16C74B |
|---|---|---|---|---|
| Program Memory (EPROM) x 14 | 4 K | 4 K | 4 K | 4 K |
| Data Memory (Bytes) x 8 | 192 | 192 | 192 | 192 |
| Pins | 28 | 40 | 28 | 40 |
| Parallel Slave Port | — | Yes | — | Yes |
| Capture/Compare/PWM Modules | 2 | 2 | 2 | 2 |
| Timer Modules | 3 | 3 | 3 | 3 |
| A/D Channels | — | — | 5 | 8 |
| Serial Communication | SPI/I$^2$C, USART | SPI/I$^2$C, USART | SPI/I$^2$C, USART | SPI/I$^2$C, USART |
| In-Circuit Serial Programming | Yes | Yes | Yes | Yes |
| Brown-out Reset | Yes | Yes | Yes | Yes |
| Interrupt Sources | 10 | 11 | 11 | 12 |
| Packages | 28-pin SDIP, SOIC, SSOP, Windowed CERDIP | 40-pin PDIP; 44-pin PLCC, MQFP, TQFP, Windowed CERDIP | 28-pin SDIP, SOIC, SSOP, Windowed CERDIP | 40-pin PDIP; 44-pin PLCC, MQFP, TQFP, Windowed CERDIP |


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## Table of Contents

1.0 General Description ..... 5
2.0 PIC16C63A/65B/73B/74B Device Varieties ..... 7
3.0 Architectural Overview ..... 9
4.0 Memory Organization ..... 15
5.0 I/O Ports ..... 29
6.0 Timer0 Module ..... 39
7.0 Timer1 Module ..... 43
8.0 Timer2 Module ..... 47
9.0 Capture/Compare/PWM Modules ..... 49
10.0 Synchronous Serial Port (SSP) Module ..... 55
11.0 Addressable Universal Synchronous Asynchronous Receiver Transmitter (USART) ..... 65
12.0 Analog-to-Digital Converter (A/D) Module ..... 79
13.0 Special Features of the CPU ..... 85
14.0 Instruction Set Summary ..... 99
15.0 Development Support ..... 107
16.0 Electrical Characteristics ..... 113
17.0 DC and AC Characteristics Graphs and Tables ..... 139
18.0 Packaging Information ..... 153
Appendix A: Revision History ..... 165
Appendix B: Device Differences ..... 165
Appendix C: Device Migrations - PIC16C63/65A/73A/74A → PIC16C63A/65B/73B/74B ..... 166
Appendix D: Migration from Baseline to Mid-Range Devices ..... 168
On-Line Support ..... 175
Reader Response ..... 176
Product Identification System ..... 177

## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@mail.microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our web site at **www.microchip.com/cn** to receive the most current information on all of our products.



Downloaded from Arrow.com.

**NOTES:**


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 1.0 GENERAL DESCRIPTION

The PIC16C63A/65B/73B/74B devices are low cost, high performance, CMOS, fully-static, 8-bit microcontrollers in the PIC16CXX mid-range family.

All PIC® 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.

The **PIC16C63A/73B** devices have 22 I/O pins. The **PIC16C65B/74B** devices have 33 I/O pins. Each device has 192 bytes of RAM. In addition, several peripheral features are available, including: three timer/counters, two Capture/Compare/PWM modules, and two serial ports. The Synchronous Serial Port (SSP) 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 on the **PIC16C73B,** while the **PIC16C74B** offers 8 channels. The 8-bit resolution is ideally suited for applications requiring low cost analog interface, e.g., thermostat control, pressure sensing, etc.

The PIC16C63A/65B/73B/74B devices have 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 (WDT), with its own on-chip RC oscillator, provides protection against software lockup, and also provides one way of waking the device from SLEEP.

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 PIC16C63A/65B/73B/74B devices fit nicely in many 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 PIC16C63A/65B/73B/74B devices 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

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

Please refer to Section 15.0 for more details about Microchip's development tools.

© 1998-2013 Microchip Technology Inc.


Downloaded from Arrow.com.

**NOTES:**


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 2.0 PIC16C63A/65B/73B/74B 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 PIC16C63A/65B/73B/74B 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 windowed CERDIP packages, 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 PIC16C63A/65B/73B/74B.

### 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.



Downloaded from Arrow.com.

**NOTES:**


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 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, most instructions execute in a single cycle (200 ns @ 20 MHz) except for program branches.

All devices covered by this data sheet contain 4K x 14-bit program memory and 192 x 8-bit data memory.

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.

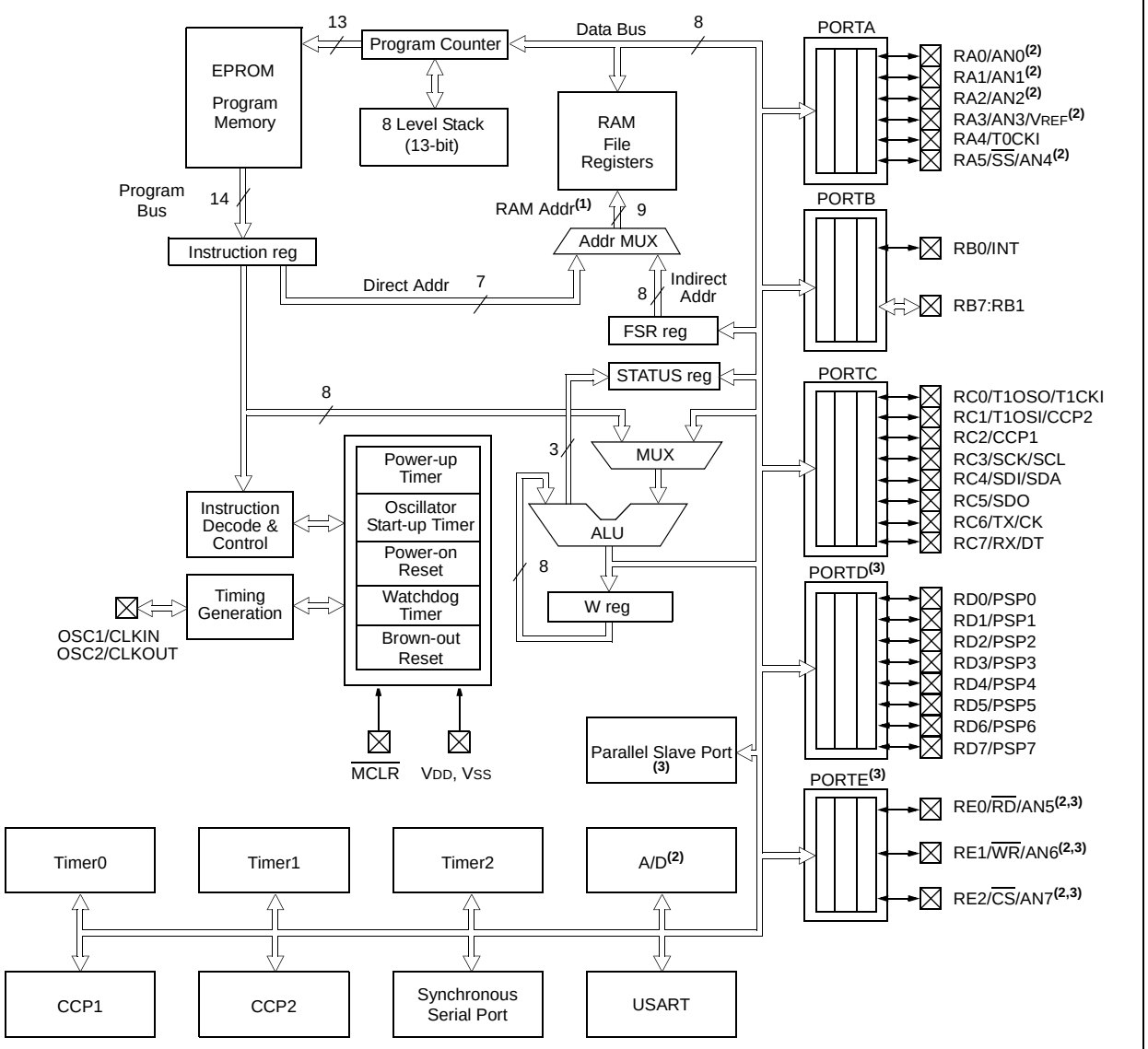
The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

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 borrow bit and a digit borrow out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.



Downloaded from Arrow.com.

**FIGURE 3-1: PIC16C63A/65B/73B/74B BLOCK DIAGRAM**




**Note 1:** Higher order bits are from the STATUS register.
**2:** A/D is not available on the PIC16C63A/65B.
**3:** PSP and Ports D and E are not available on PIC16C63A/73B.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 3-1: PIC16C63A/73B 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(4) | 2 | 2 | I/O | TTL | RA0 can also be analog input 0(4). |
| RA1/AN1(4) | 3 | 3 | I/O | TTL | RA1 can also be analog input 1(4). |
| RA2/AN2(4) | 4 | 4 | I/O | TTL | RA2 can also be analog input 2(4). |
| RA3/AN3/VREF(4) | 5 | 5 | I/O | TTL | RA3 can also be analog input 3 or analog reference voltage(4). |
| RA4/T0CKI | 6 | 6 | I/O | ST | RA4 can also be the clock input to the Timer0 module. Output is open drain type. |
| RA5/SS/AN4(4) | 7 | 7 | I/O | TTL | RA5 can also be analog input 4(4) 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.
**4:** A/D module is not available in the PIC16C63A.



Downloaded from Arrow.com.

**TABLE 3-2: PIC16C65B/74B PINOUT DESCRIPTION**

| Pin Name | DIP Pin# | PLCC Pin# | TQFP MQFP 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(5) | 2 | 3 | 19 | I/O | TTL | RA0 can also be analog input 0(5). |
| RA1/AN1(5) | 3 | 4 | 20 | I/O | TTL | RA1 can also be analog input 1(5). |
| RA2/AN2(5) | 4 | 5 | 21 | I/O | TTL | RA2 can also be analog input 2(5). |
| RA3/AN3/VREF(5) | 5 | 6 | 22 | I/O | TTL | RA3 can also be analog input 3 or analog reference voltage(5). |
| 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(5) | 7 | 8 | 24 | I/O | TTL | RA5 can also be analog input 4(5) 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 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 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.
**5:** A/D is not available on the PIC16C65B.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 3-2: PIC16C65B/74B PINOUT DESCRIPTION (CONTINUED)**

| Pin Name | DIP Pin# | PLCC Pin# | TQFP MQFP 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(5) | 8 | 9 | 25 | I/O | ST/TTL(3) | RE0 can also be read control for the parallel slave port, or analog input 5(5). |
| RE1/$\overline{WR}$/AN6(5) | 9 | 10 | 26 | I/O | ST/TTL(3) | RE1 can also be write control for the parallel slave port, or analog input 6(5). |
| RE2/$\overline{CS}$/AN7(5) | 10 | 11 | 27 | I/O | ST/TTL(3) | RE2 can also be select control for the parallel slave port, or analog input 7(5). |
| 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 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 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.
**5:** A/D is not available on the PIC16C65B.



Downloaded from Arrow.com.

## 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 3-2.

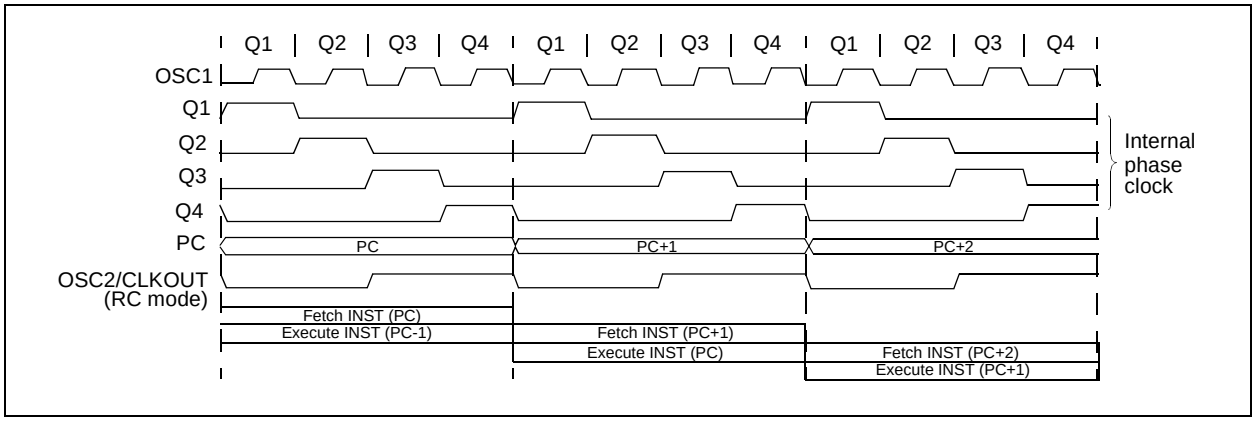
## 3.2 Instruction Flow/Pipelining

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).

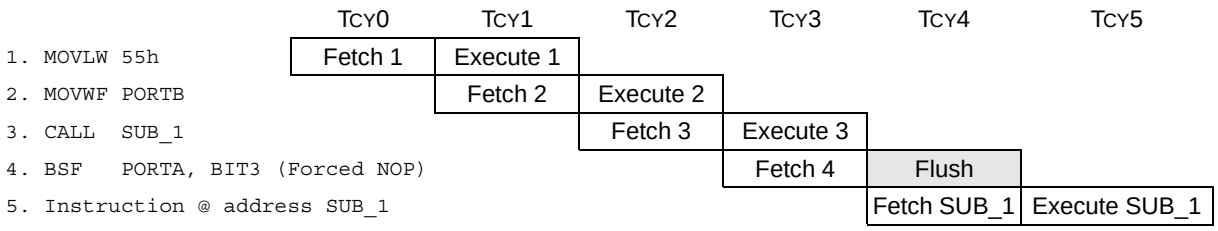
A fetch cycle begins with the program counter (PC) incrementing in Q1.

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 3-2: CLOCK/INSTRUCTION CYCLE**




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




**Note:** 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.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

# 4.0 MEMORY ORGANIZATION

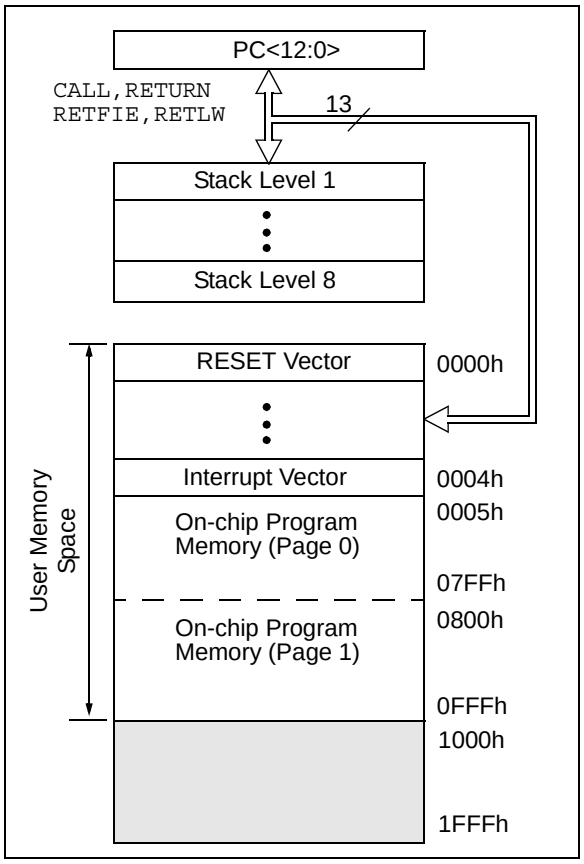
## 4.1 Program Memory Organization

The PIC16C63A/65B/73B/74B has a 13-bit program counter capable of addressing an 8K x 14 program memory space. All devices covered by this data sheet have 4K x 14 bits of program memory. The address range is 0000h - 0FFFh for all devices.

Accessing a location above 0FFFh will cause a wrap-around.

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

**FIGURE 4-1: PIC16C63A/65B/73B/74B PROGRAM MEMORY MAP AND STACK**




## 4.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers (GPR) and the Special Function Registers (SFR). 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 SFRs. Above the SFRs are GPRs, implemented as static RAM.

All implemented banks contain SFRs. Frequently used SFRs from one bank may be mirrored in another bank for code reduction and quicker access.

| **Note:** | Maintain the IRP and RP1 bits clear in these devices. |
|---|---|

### 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).



Downloaded from Arrow.com.

**FIGURE 4-2: REGISTER FILE MAP**

| File Address | Bank 0 | Bank 1 | File Address |
|---|---|---|---|
| 00h | INDF(1) | INDF(1) | 80h |
| 01h | TMR0 | OPTION_REG | 81h |
| 02h | PCL | PCL | 82h |
| 03h | STATUS | STATUS | 83h |
| 04h | FSR | FSR | 84h |
| 05h | PORTA | TRISA | 85h |
| 06h | PORTB | TRISB | 86h |
| 07h | PORTC | TRISC | 87h |
| 08h | PORTD(2) | TRISD(2) | 88h |
| 09h | PORTE(2) | TRISE(2) | 89h |
| 0Ah | PCLATH | PCLATH | 8Ah |
| 0Bh | INTCON | INTCON | 8Bh |
| 0Ch | PIR1 | PIE1 | 8Ch |
| 0Dh | PIR2 | PIE2 | 8Dh |
| 0Eh | TMR1L | PCON | 8Eh |
| 0Fh | TMR1H | | 8Fh |
| 10h | T1CON | | 90h |
| 11h | TMR2 | | 91h |
| 12h | T2CON | PR2 | 92h |
| 13h | SSPBUF | SSPADD | 93h |
| 14h | SSPCON | SSPSTAT | 94h |
| 15h | CCPR1L | | 95h |
| 16h | CCPR1H | | 96h |
| 17h | CCP1CON | | 97h |
| 18h | RCSTA | TXSTA | 98h |
| 19h | TXREG | SPBRG | 99h |
| 1Ah | RCREG | | 9Ah |
| 1Bh | CCPR2L | | 9Bh |
| 1Ch | CCPR2H | | 9Ch |
| 1Dh | CCP2CON | | 9Dh |
| 1Eh | ADRES(3) | | 9Eh |
| 1Fh | ADCON0(3) | ADCON1(3) | 9Fh |
| 20h | General Purpose Register | General Purpose Register | A0h |
| 7Fh | | | FFh |

Unimplemented data memory locations, read as '0'.

**Note 1:** Not a physical register.
**2:** These registers are not implemented on the PIC16C63A/73B, read as '0'.
**3:** These registers are not implemented on the PIC16C63A/65B, read as '0'.

### 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.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 4-1: 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 | INDF(4) | 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 | PCL(4) | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 03h | STATUS(4) | IRP(2) | RP1(2) | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 04h | FSR(4) | 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 | PORTD(5) | PORTD Data Latch when written: PORTD pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 09h | PORTE(5) | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |
| 0Ah | PCLATH(1,4) | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 0Bh | INTCON(4) | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(5) | ADIF(6) | 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(6) | A/D Result register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0(6) | 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>.
**2:** The IRP and RP1 bits are reserved; always maintain these bits clear.
**3:** Other (non power-up) RESETS include external RESET through $\overline{MCLR}$ and Watchdog Timer Reset.
**4:** These registers can be addressed from either bank.
**5:** PORTD, PORTE and the parallel slave port are not implemented on the PIC16C63A/73B; always maintain these bits and registers clear.
**6:** The A/D is not implemented on the PIC16C63A/65B; always maintain these bits and registers clear.



Downloaded from Arrow.com.

**TABLE 4-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)**

| 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 | INDF(4) | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 0000 0000 |
| 81h | OPTION_REG | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 82h | PCL(4) | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 83h | STATUS(4) | IRP(2) | RP1(2) | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 84h | FSR(4) | 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 | TRISD(5) | PORTD Data Direction register | | | | | | | | 1111 1111 | 1111 1111 |
| 89h | TRISE(5) | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction bits | | | 0000 -111 | 0000 -111 |
| 8Ah | PCLATH(1,4) | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 8Bh | INTCON(4) | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 8Ch | PIE1 | PSPIE(5) | ADIE(6) | 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 | — | — | 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(6) | — | — | — | — | — | 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>.
**2:** The IRP and RP1 bits are reserved; always maintain these bits clear.
**3:** Other (non power-up) RESETS include external RESET through $\overline{MCLR}$ and Watchdog Timer Reset.
**4:** These registers can be addressed from either bank.
**5:** PORTD, PORTE and the parallel slave port are not implemented on the PIC16C63A/73B; always maintain these bits and registers clear.
**6:** The A/D is not implemented on the PIC16C63A/65B; always maintain these bits and registers clear.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

#### 4.2.2.1 STATUS Register

The STATUS register, shown in Register 4-1, 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 that only `BCF, BSF, SWAPF` and `MOVWF` instructions be used to alter the STATUS register. These instructions do not affect the Z, C or DC bits in the STATUS register. For other instructions which do not affect status bits, see the "Instruction Set Summary."

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

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

**REGISTER 4-1: STATUS REGISTER (ADDRESS 03h, 83h)**

| R/W-0 | R/W-0 | R/W-0 | R-1 | R-1 | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| IRP(1) | RP1(1) | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C(2) |
| bit 7 | | | | | | | bit 0 |

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

bit 6-5 **RP1(1):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(2):** 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 1:** Maintain the IRP and RP1 bits clear.

**2:** For $\overline{borrow}$ and digit $\overline{borrow}$, 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.

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |



Downloaded from Arrow.com.

#### 4.2.2.2 OPTION Register

The OPTION_REG 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. |
|---|---|

**REGISTER 4-2: OPTION_REG REGISTER (ADDRESS 81h)**

| 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 |
| bit 7 | | | | | | | bit 0 |

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 |

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| -n = Value at POR | ’1’ = Bit is set | ’0’ = Bit is cleared | x = Bit is unknown |


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

#### 4.2.2.3 INTCON Register

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 are 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. |
|---|---|

**REGISTER 4-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-x |
|---|---|---|---|---|---|---|---|
| GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF |
| bit 7 | | | | | | | bit 0 |

bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts

bit 6 **PEIE**: Peripheral Interrupt Enable bit
1 = Enables all unmasked 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**(1)**
0 = None of the RB7:RB4 pins have changed state

**Note 1:** A mismatch condition will exist until PORTB is read. After reading PORTB, the RBIF flag bit can be cleared.

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| -n = Value at POR | ’1’ = Bit is set | ’0’ = Bit is cleared | x = Bit is unknown |



Downloaded from Arrow.com.

### 4.2.2.4 PIE1 Register

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

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

**REGISTER 4-4: PIE1 REGISTER (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(2) | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE |
| bit 7 | | | | | | | bit 0 |

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(2):** 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:** PIC16C63A/73B devices do not have a parallel slave port implemented; always maintain this bit clear.
**2:** PIC16C63A/65B devices do not have an A/D implemented; always maintain this bit clear.

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

### 4.2.2.5 PIR1 Register

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

| **Note:** | Interrupt flag bits are 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. |
|---|---|

**REGISTER 4-5: PIR1 REGISTER (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(2) | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF |
| bit 7 | | | | | | | bit 0 |

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(2)**: 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 (clear 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 (clear 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:** PIC16C63A/73B devices do not have a parallel slave port implemented. This bit location is reserved on these devices.
**2:** PIC16C63A/65B devices do not have an A/D implemented. This bit location is reserved on these devices.

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |



Downloaded from Arrow.com.

### 4.2.2.6 PIE2 Register

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

**REGISTER 4-6: PIE2 REGISTER (ADDRESS 8Dh)**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | CCP2IE |
| bit 7 | | | | | | | bit 0 |

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

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

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

### 4.2.2.7 PIR2 Register

This register contains the CCP2 interrupt flag bit.

**Note:** Interrupt flag bits are 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.

**REGISTER 4-7: PIR2 REGISTER (ADDRESS 0Dh)**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | CCP2IF |
| bit 7 | | | | | | | bit 0 |

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

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

#### 4.2.2.8 PCON Register

The Power Control (PCON) register contains flag bits to allow differentiation between a Power-on Reset (POR), a Brown-out Reset (BOR), a Watchdog Reset (WDT) and an external MCLR Reset.

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

**REGISTER 4-8: PCON REGISTER (ADDRESS 8Eh)**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-q |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ |
| bit 7 | | | | | | | bit 0 |

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

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

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

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| -n = Value at POR | ’1’ = Bit is set | ’0’ = Bit is cleared | x = Bit is unknown |

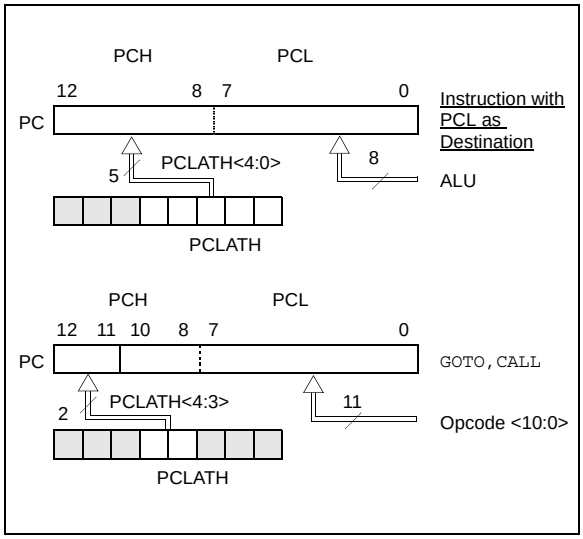


Downloaded from Arrow.com.

## 4.3 PCL and PCLATH

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 4-3 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 4-3: 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.

**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

PIC16CXX 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 executing 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 popped from 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 1:** The contents of PCLATH are unchanged after a return or RETFIE instruction is executed. The user must set up PCLATH for any subsequent CALL's or GOTO's

**2:** PCLATH<4> is not used in these PIC® devices. 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)
    CALL  SUB1_P1  ;Call subroutine in
    :              ;page 1 (800h-FFFh)
    :
    ORG   0x900    ;page 1 (800h-FFFh)
SUB1_P1
    :              ;called subroutine
    :              ;page 1 (800h-FFFh)
    :
    RETURN         ;return to Call subroutine
                   ;in page 0 (000h-7FFh)
```


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 4.5 Indirect Addressing, INDF and FSR Registers

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 4-4.

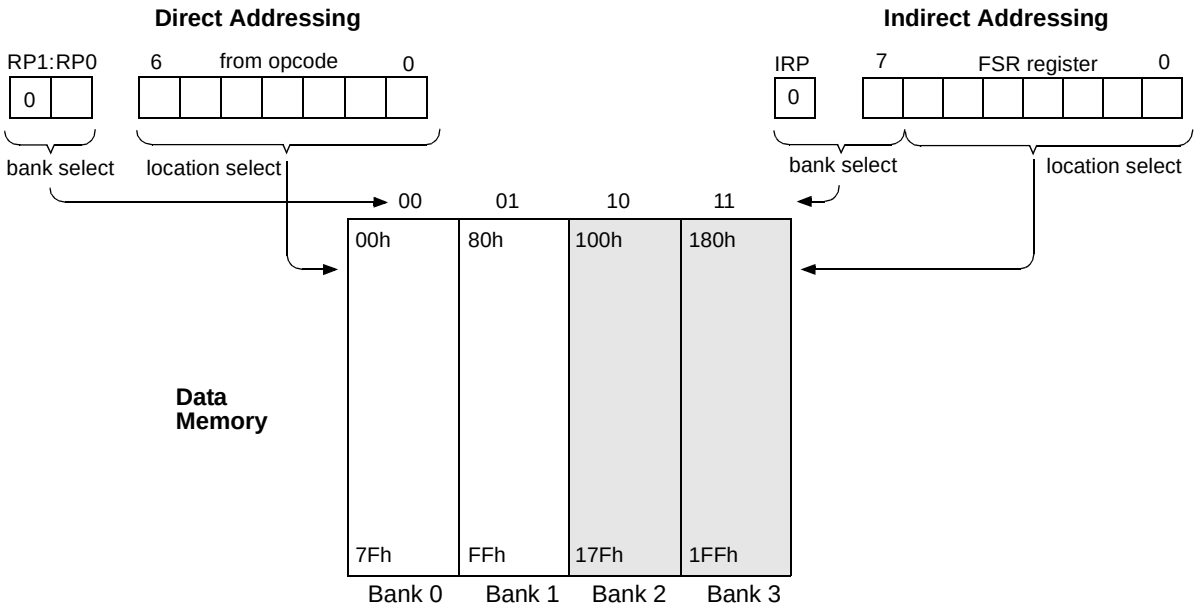
A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 4-2.

**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
```

| **Note:** | Maintain the IRP and RP1 bits clear. |
|---|---|

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




**Note 1:** For register file map detail, see Figure 4-2.
**2:** Shaded portions are not implemented; maintain the IRP and RP1 bits clear.



Downloaded from Arrow.com.

**NOTES:**


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 5.0 I/O PORTS

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

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, the 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.

On the PIC16C73B/74B, 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).

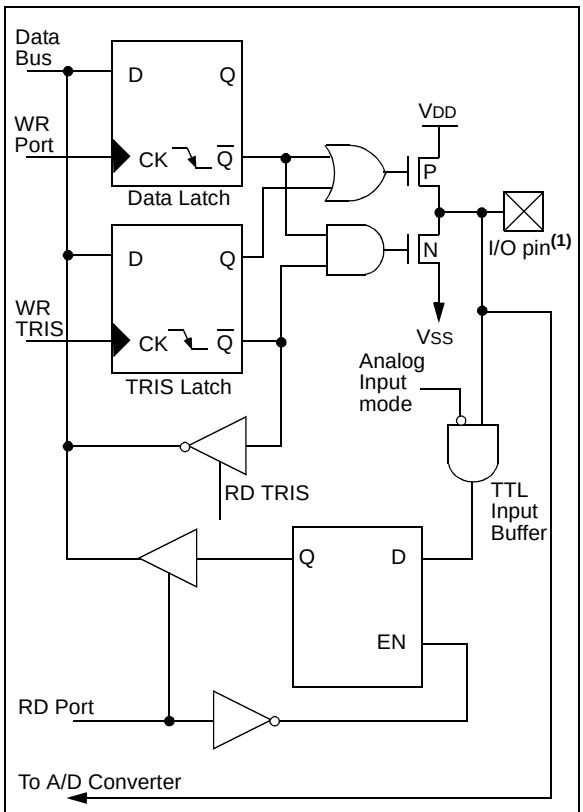
| **Note:** | On all RESETS, pins with analog functions are configured as analog and digital inputs. |
|---|---|

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 (PIC16C73B/74B)**

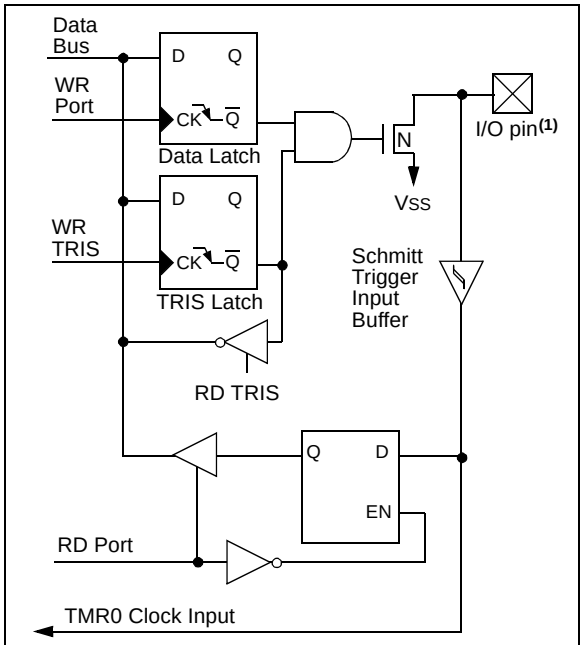
```
BCF     STATUS, RP0   ;
CLRF    PORTA         ; Initialize PORTA by
                      ; clearing output
                      ; data latches
BSF     STATUS, RP0   ; Select Bank 1
MOVLW   0x06          ; Configure all pins
MOVWF   ADCON1        ; as digital inputs
MOVLW   0xCF          ; Value 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 pins have protection diodes to VDD and VSS.



Downloaded from Arrow.com.

**TABLE 5-1: PORTA FUNCTIONS**

| Name | Bit# | Buffer | Function |
|---|---|---|---|
| RA0/AN0(1) | bit0 | TTL | Digital input/output or analog input. |
| RA1/AN1(1) | bit1 | TTL | Digital input/output or analog input. |
| RA2/AN2(1) | bit2 | TTL | Digital input/output or analog input. |
| RA3/AN3/VREF(1) | bit3 | TTL | Digital input/output or analog input or VREF. |
| RA4/T0CKI | bit4 | ST | Digital input/output or external clock input for Timer0.<br>Output is open drain type. |
| RA5/$\overline{SS}$/AN4(1) | bit5 | TTL | Input/output or slave select input for synchronous serial port or analog input. |

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

**Note 1:** The A/D is not implemented on the PIC16C63A/65B. Pins will operate as digital I/O only. ADCON1 is not implemented; maintain this register clear.

**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(1) | — | — | — | — | — | PCFG2 | PCFG1 | PCFG0 | ---- -000 | ---- -000 |

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

**Note 1:** The A/D is not implemented on the PIC16C63A/65B. Pins will operate as digital I/O only. ADCON1 is not implemented; maintain this register clear.


© 1998-2013 Microchip Technology Inc.

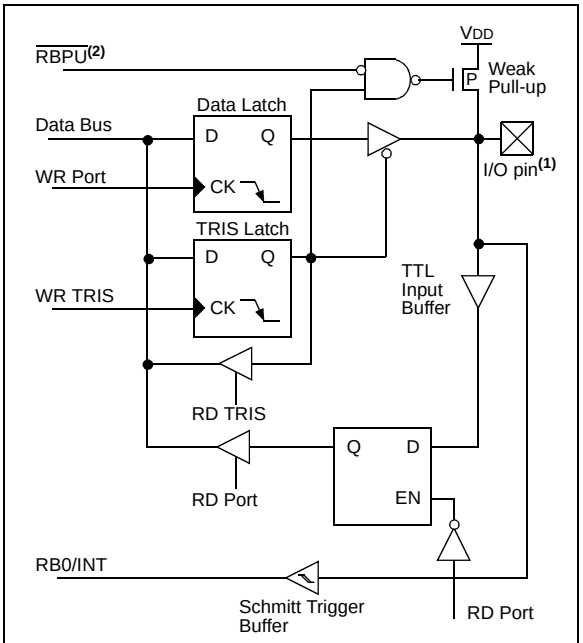
Downloaded from Arrow.com.

## 5.2 PORTB and TRISB Registers

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).

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_REG<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 RBPU bit (OPTION_REG<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 value latched on the last read of PORTB. The "mismatch" outputs of RB7:RB4 are OR'd 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.

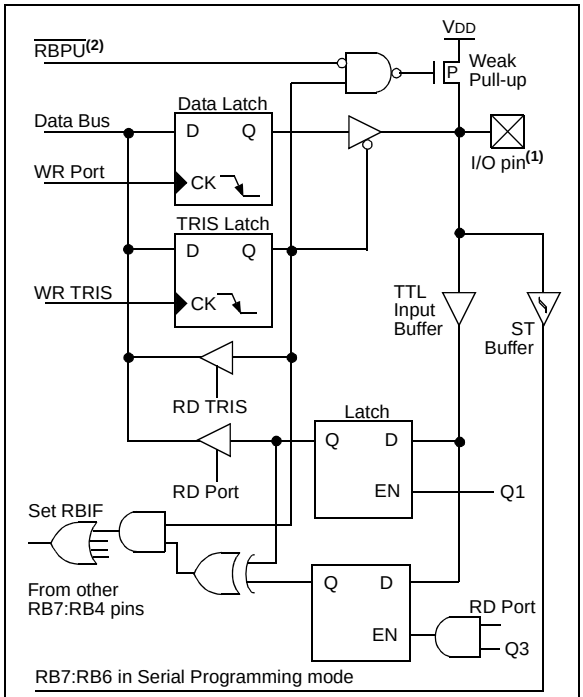
This interrupt-on-mismatch feature, together with software configurable pull-ups on these four pins, allow easy interface to a keypad and make it possible for wake-up on key depression. Refer to the Embedded Control Handbook, *"Implementing Wake-up on Key Stroke"* (AN552).

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.

RB0/INT is an external interrupt input pin and is configured using the INTEDG bit (OPTION_REG<6>).

RB0/INT is discussed in detail in Section 13.5.1.

**FIGURE 5-4: BLOCK DIAGRAM OF RB7:RB4 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 RBPU bit (OPTION_REG<7>).



Downloaded from Arrow.com.

**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 | PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | uuuu uuuu |
| 86h | TRISB | PORTB Data Direction register | | | | | | | | 1111 1111 | 1111 1111 |
| 81h | OPTION_REG | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |

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


© 1998-2013 Microchip Technology Inc.

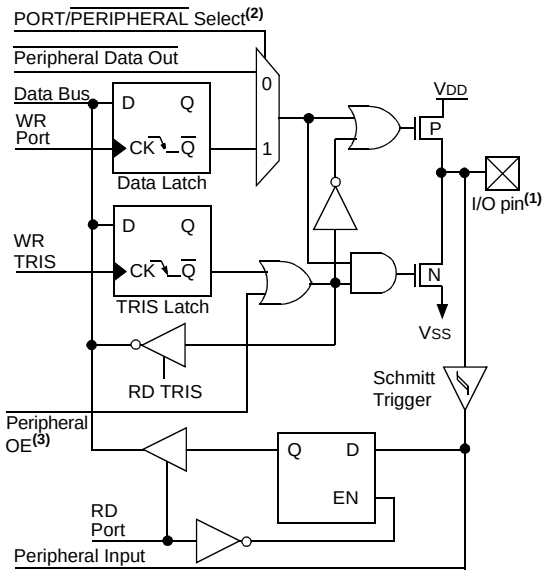
Downloaded from Arrow.com.

## 5.3 PORTC and TRISC Registers

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.

**FIGURE 5-5: PORTC BLOCK DIAGRAM**




**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 | 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 | bit6 | ST | Input/output port pin or USART Asynchronous Transmit, or USART Synchronous Clock. |
| RC7/RX/DT | bit7 | ST | Input/output port pin or USART Asynchronous Receive, or USART Synchronous Data. |

Legend: ST = Schmitt Trigger input

**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



Downloaded from Arrow.com.

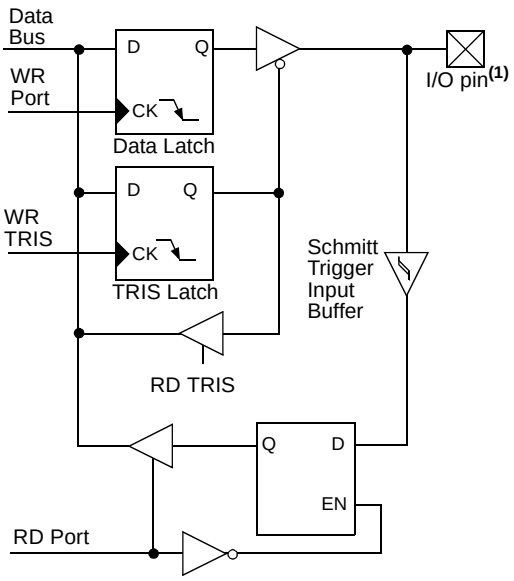
## 5.4 PORTD and TRISD Registers

| Note: | The PIC16C63A and PIC16C73B do not provide PORTD. The PORTD and TRISD registers are not implemented. |
|---|---|

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configured 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 5-6: PORTD BLOCK DIAGRAM**




**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.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 5.5 PORTE and TRISE Register

**Note 1:** The PIC16C63A and PIC16C73B do not provide PORTE. The PORTE and TRISE registers are not implemented.

**2:** The PIC16C63A/65B does not provide an A/D module. A/D functions are not implemented.

PORTE has three pins: RE0/$\overline{RD}$/AN5, RE1/$\overline{WR}$/AN6 and RE2/$\overline{CS}$/AN7, which are individually configured 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.

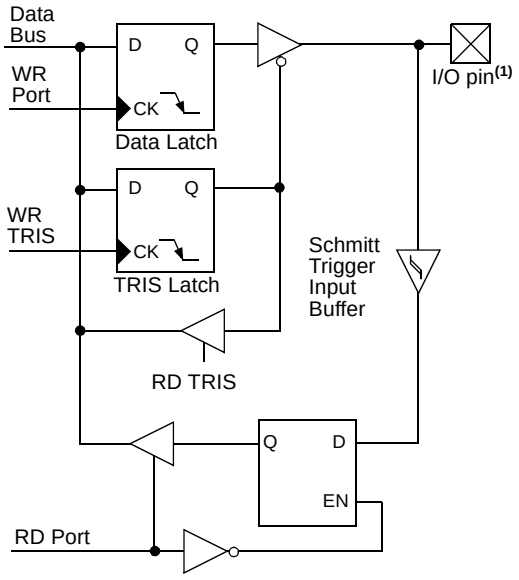
Register 5-1 shows the TRISE register, which also controls the parallel slave port operation.

PORTE pins may be multiplexed with analog inputs (PIC16C74B only). 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 and read as '0's.

**FIGURE 5-7: PORTE BLOCK DIAGRAM**




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

**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 = Idle<br>0 = Read operation. Contents of PORTD register is output to PORTD I/O pins (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 = Idle<br>0 = Write operation. Value of PORTD I/O pins is latched into 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.



Downloaded from Arrow.com.

**REGISTER 5-1: 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 | — | TRISE2 | TRISE1 | TRISE0 |
| bit 7 | | | | | | | bit 0 |

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'

bit 2 **TRISE2:** Direction Control bit for pin RE2/$\overline{CS}$/AN7
1 = Input
0 = Output

bit 1 **TRISE1:** Direction Control bit for pin RE1/$\overline{WR}$/AN6
1 = Input
0 = Output

bit 0 **TRISE0:** Direction Control bit for pin RE0/$\overline{RD}$/AN5
1 = Input
0 = Output

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

**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.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 5.6 Parallel Slave Port (PSP)

**Note:** The PIC16C63A and PIC16C73B do not provide a parallel slave port. The PORTD, PORTE, TRISD and TRISE registers are not implemented.

PORTD operates as an 8-bit wide Parallel Slave Port (PSP), 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 PIC® MCU) 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 external device is controlling the direction of data flow.

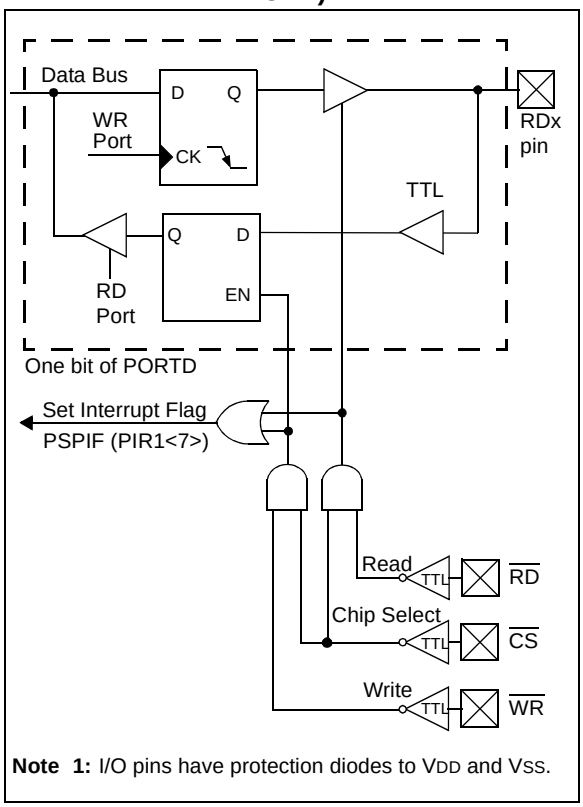
A write to the PSP occurs when both the $\overline{CS}$ and $\overline{WR}$ lines are first detected low. When either the $\overline{CS}$ or $\overline{WR}$ lines become high (level triggered), then the Input Buffer Full (IBF) status flag bit (TRISE<7>) is set on the Q4 clock cycle, following the next Q2 cycle, to signal the write is complete (Figure 5-9). 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 (IBOV) status flag bit (TRISE<5>) is set if a second write to the PSP 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 (OBF) status flag bit (TRISE<6>) is cleared immediately (Figure 5-10), 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 PSP 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 5-8: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)**




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



Downloaded from Arrow.com.

**FIGURE 5-9: PARALLEL SLAVE PORT WRITE WAVEFORMS**

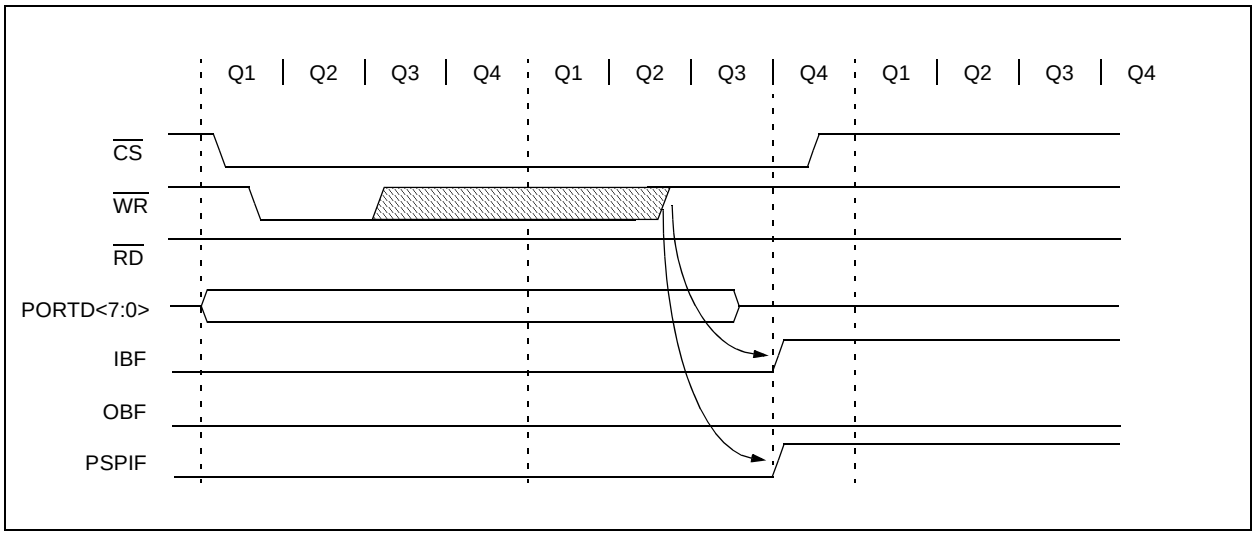



**FIGURE 5-10: PARALLEL SLAVE PORT READ WAVEFORMS**

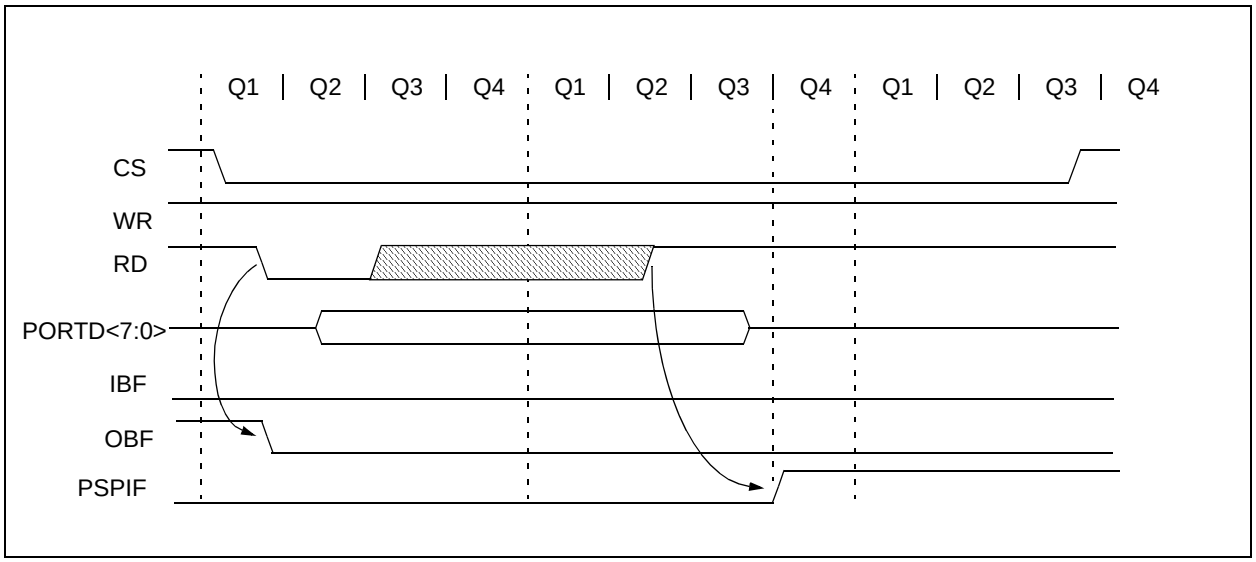



**TABLE 5-11: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT**

| 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 |
| 0Bh, 8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 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.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 6.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

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

Figure 6-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

Additional information on the Timer0 module is available in the PIC® Mid-Range MCU Family Reference Manual (DS33023).

Timer mode is selected by clearing bit T0CS (OPTION_REG<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. The user can work around this by writing an adjusted value to the TMR0 register.

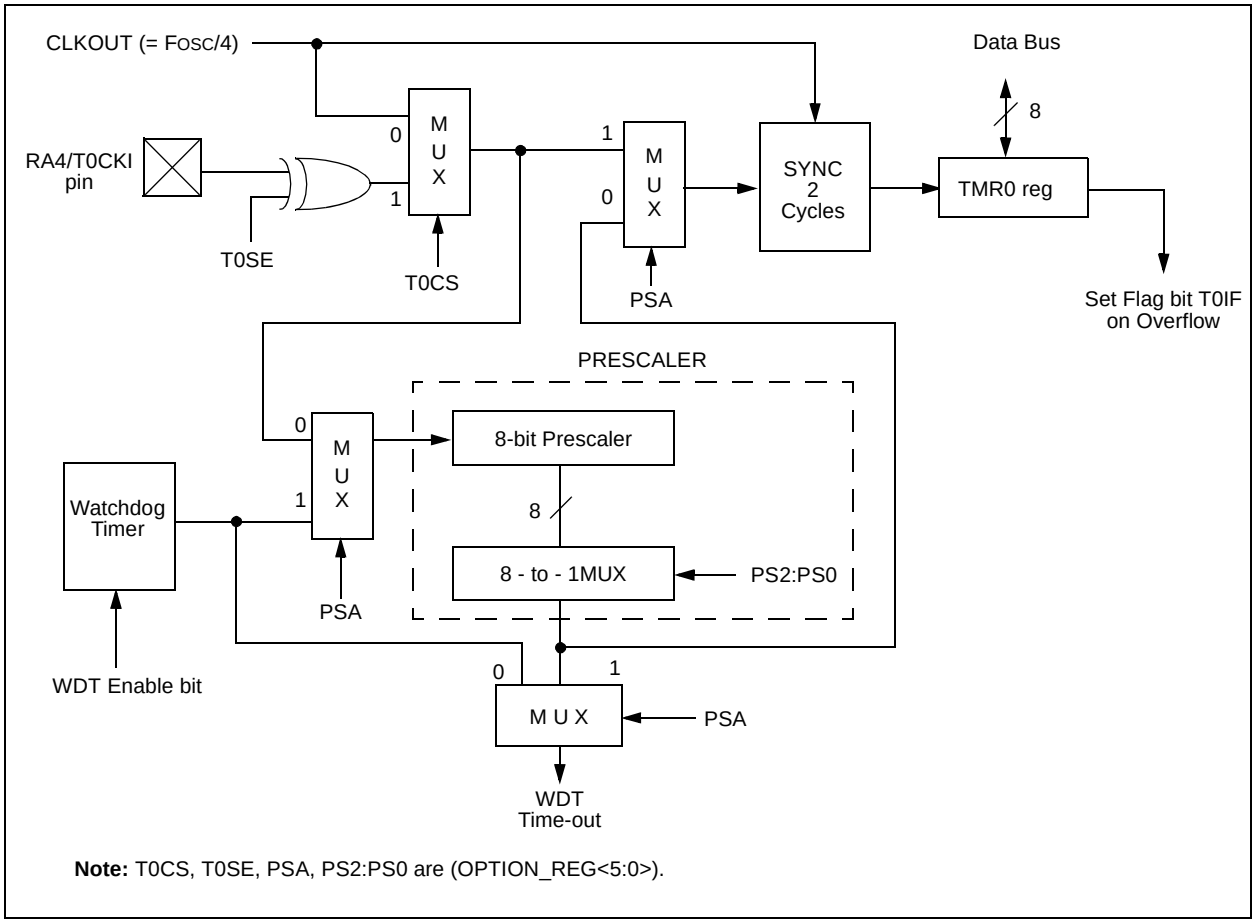
Counter mode is selected by setting bit T0CS (OPTION_REG<5>). In counter mode, Timer0 will increment, either on every rising, or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions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

The prescaler is mutually exclusively shared between the Timer0 module and the watchdog timer. The prescaler is not readable or writable. Section 6.3 details the operation of the prescaler.

### 6.1 Timer0 Interrupt

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

**FIGURE 6-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**




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



Downloaded from Arrow.com.

## 6.2 Using Timer0 with an External Clock

The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the synchronized input on the Q2 and Q4 cycles of the internal phase clocks. Therefore, it is necessary for T0CKI to be high for at least 2 TOSC (and a small RC delay of 20 ns) and low for at least 2 TOSC (and a small RC delay of 20 ns). Refer to the electrical specification for the desired device.

## 6.3 Prescaler

There is only one prescaler available which is mutually exclusively shared between the Timer0 module and the watchdog timer. A prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer, and vice-versa. This prescaler is not readable or writable (see Figure 6-1).

The PSA and PS2:PS0 bits (OPTION_REG<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. |
|---|---|

**REGISTER 6-1: OPTION_REG REGISTER**

| 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 |
| bit 7 | | | | | | | bit 0 |

bit 7 **$\overline{\text{RBPU}}$**

bit 6 **INTEDG**

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

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

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

| **Note:** | To avoid an unintended device RESET, the instruction sequence shown in the PIC® Mid-Range MCU Family Reference Manual (DS33023, Section 11.6) must be executed when changing the prescaler assignment from Timer0 to the WDT. This sequence must be followed even if the WDT is disabled. |
|---|---|


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 6-1: 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 | TMR0 | Timer0 Module's register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Bh,8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 81h | OPTION_REG | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |

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



Downloaded from Arrow.com.

**NOTES:**


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 7.0 TIMER1 MODULE

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 9.0) using the special event trigger. Register 7-1 shows the Timer1 control register.

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, and these pins read as ‘0’.

Additional information on timer modules is available in the PIC® Mid-range MCU Family Reference Manual (DS33023).

**REGISTER 7-1: 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 | T1SYNC | TMR1CS | TMR1ON |
| bit 7 | | | | | | | bit 0 |

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 (The oscillator inverter is turned off to eliminate power drain)

bit 2 **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

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| -n = Value at POR | ’1’ = Bit is set | ’0’ = Bit is cleared | x = Bit is unknown |



Downloaded from Arrow.com.

## 7.1 Timer1 Operation in Timer Mode

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.

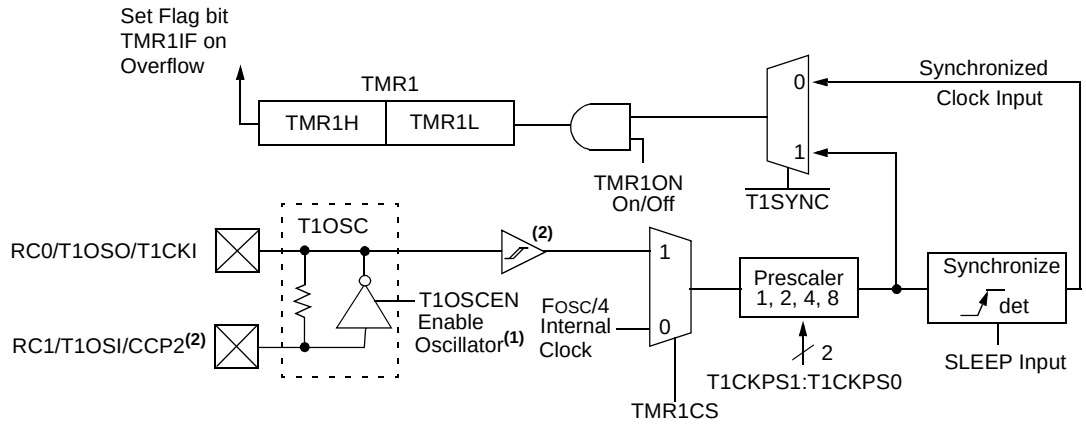
## 7.2 Timer1 Operation in Synchronized Counter Mode

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 on 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.

**FIGURE 7-1: TIMER1 BLOCK DIAGRAM**




**Note 1:** When the T1OSCEN bit is cleared, the inverter is turned off. This eliminates power drain.
**2:** For the PIC16C65B/73B/74B, the Schmitt Trigger is not implemented in External Clock mode.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 7.3 Timer1 Operation in Asynchronous Counter Mode

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 7.3.1).

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

### 7.3.1 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. Examples 12-2 and 12-3 in the PIC® Mid-Range MCU Family Reference Manual (DS33023) show how to read and write Timer1 when it is running in Asynchronous mode.

## 7.4 Timer1 Oscillator

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 use with a 32 kHz crystal. Table 7-1 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 7-1: 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.

## 7.5 Resetting Timer1 using a CCP Trigger Output

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 register pair effectively becomes the period register for Timer1.

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

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.

## 7.7 Timer1 Prescaler

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



Downloaded from Arrow.com.

**TABLE 7-2: 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 | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF(2) | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE(2) | RCIE | TXIE | 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 | 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 PIC16C63A/73B; always maintain these bits clear.

**2:** Bits ADIE and ADIF are reserved on the PIC16C63A/65B; always maintain these bits clear.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 8.0 TIMER2 MODULE

Timer2 is an 8-bit timer with a prescaler and a postscaler. It can be used as the PWM time-base for the 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.

Register 8-1 shows the Timer2 control register.

Additional information on timer modules is available in the PIC® Mid-Range MCU Family Reference Manual (DS33023).

### 8.1 Timer2 Prescaler and Postscaler

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

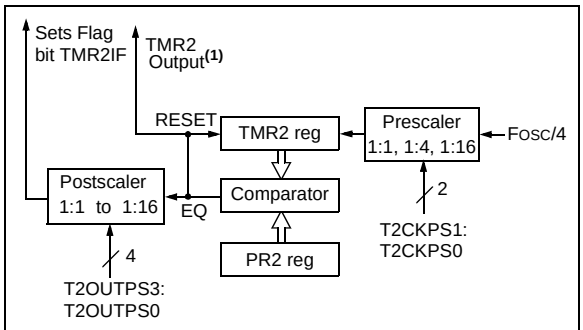
- a write to the TMR2 register
- a write to the T2CON register
- any device RESET (POR, BOR, $\overline{\text{MCLR}}$ Reset, or WDT Reset)

TMR2 is not cleared when T2CON is written.

### 8.2 Output of TMR2

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

**FIGURE 8-1: TIMER2 BLOCK DIAGRAM**




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

**REGISTER 8-1: 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 |
| bit 7 | | | | | | | bit 0 |

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

bit 6-3 **TOUTPS3:TOUTPS0**: Timer2 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
0010 = 1:3 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

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |



Downloaded from Arrow.com.

**TABLE 8-1: 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 | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF(2) | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE(2) | RCIE | TXIE | 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 PIC16C63A/73B; always maintain these bits clear.

**2:** Bits ADIE and ADIF are reserved on the PIC16C63A/65B; always maintain these bits clear.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 9.0 CAPTURE/COMPARE/PWM MODULES

Each Capture/Compare/PWM (CCP) module contains a 16-bit register which can operate as a:

- 16-bit Capture register
- 16-bit Compare register
- PWM Master/Slave Duty Cycle register

Both the CCP1 and CCP2 modules are identical in operation, with the exception being the operation of the special event trigger. Table 9-1 and Table 9-2 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. The special event trigger is generated by a compare match and will reset Timer1.

**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. The special event trigger is generated by a compare match and will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

Additional information on CCP modules is available in the PIC® Mid-Range MCU Family Reference Manual (DS33023) and in “Using the CCP Modules” (AN594).

**TABLE 9-1: CCP MODE - TIMER RESOURCES REQUIRED**

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

**TABLE 9-2: 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. |



Downloaded from Arrow.com.

**REGISTER 9-1: CCP1CON REGISTER/CCP2CON REGISTER**

| 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 |
| bit 7 | | | | | | | bit 0 |

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 disabled (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, CCPx pin is unaffected); CCP1 resets TMR1; CCP2 resets TMR1 and starts an A/D conversion (if A/D module is enabled)
11xx = PWM mode

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 9.1 Capture Mode

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 one of the following and is configured using CCPxCON<3:0>:

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

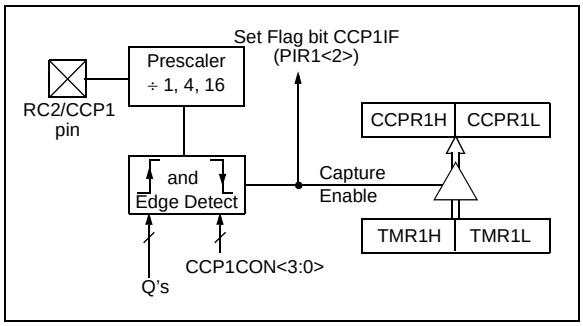
An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit CCP1IF (PIR1<2>) is set. The interrupt flag must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the previous captured value is overwritten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

### 9.1.1 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 pin is configured as an output, a write to the port can cause a capture condition. |
|---|---|

**FIGURE 9-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**




### 9.1.2 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.

### 9.1.3 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.

### 9.1.4 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. 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 9-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

**EXAMPLE 9-1: CHANGING BETWEEN CAPTURE PRESCALERS**

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



Downloaded from Arrow.com.

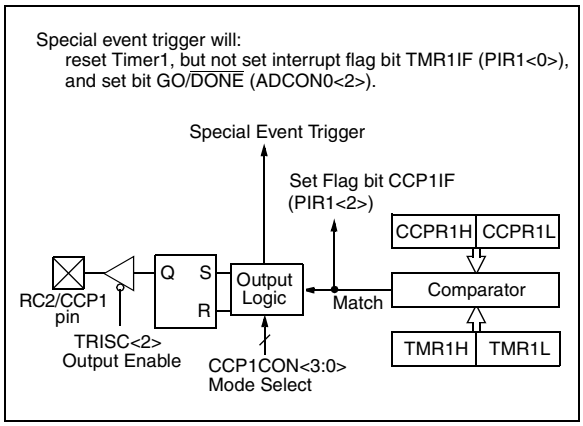
## 9.2 Compare Mode

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 9-2: COMPARE MODE OPERATION BLOCK DIAGRAM**




### 9.2.1 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 PORTC I/O data latch. |

### 9.2.2 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.

### 9.2.3 SOFTWARE INTERRUPT MODE

When Generate Software Interrupt mode is chosen, the CCP1 pin is not affected. The CCPIF bit is set, causing a CCP interrupt (if enabled).

### 9.2.4 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 event trigger output of CCP2 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>). |

## 9.3 PWM Mode (PWM)

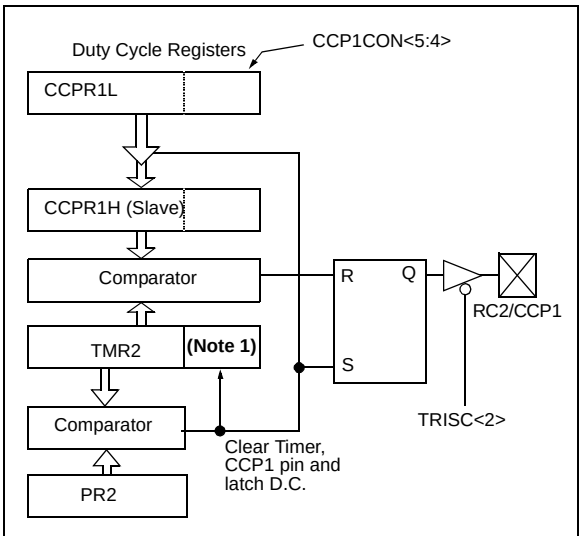
In Pulse Width Modulation mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

| | |
|---|---|
| **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 9-3 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 9.3.3.

**FIGURE 9-3: SIMPLIFIED PWM BLOCK DIAGRAM**



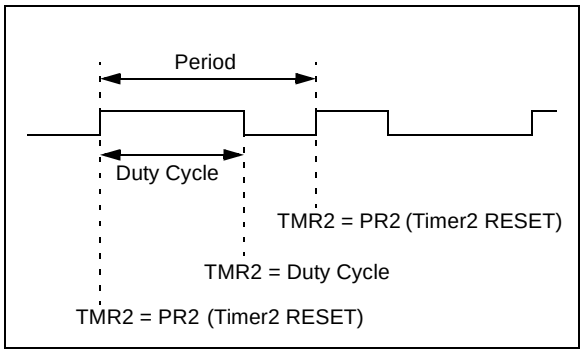

**Note 1:** 8-bit timer is concatenated with 2-bit internal Q clock, or 2 bits of the prescale, to create 10-bit time-base.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

A PWM output (Figure 9-4) 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 9-4: PWM OUTPUT**




### 9.3.1 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 8.1) 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. |
|---|---|

### 9.3.2 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:

$$\text{PWM duty cycle} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot T_{OSC} \cdot (\text{TMR2 prescale value})$$

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:

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

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

### 9.3.3 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 9-3: 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 |



Downloaded from Arrow.com.

**TABLE 9-4: 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 | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF(2) | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 0Dh | PIR2 | — | — | — | — | — | — | — | CCP2IF | ---- ---0 | ---- ---0 |
| 8Ch | PIE1 | PSPIE(1) | ADIE(2) | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 8Dh | 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 TMR1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | 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 | 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 |

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by Capture and Timer1.

**Note 1:** The PSP is not implemented on the PIC16C63A/73B; always maintain these bits clear.

**2:** The A/D is not implemented on the PIC16C63A/65B; always maintain these bits clear.

**TABLE 9-5: 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 | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF(2) | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 0Dh | PIR2 | — | — | — | — | — | — | — | CCP2IF | ---- ---0 | ---- ---0 |
| 8Ch | PIE1 | PSPIE(1) | ADIE(2) | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 8Dh | 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 | 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 |

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 PIC16C63A/73B; always maintain these bits clear.

**2:** Bits ADIE and ADIF are reserved on the PIC16C63A/65B; always maintain these bits clear.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 10.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

### 10.1 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 ($I^2C$)

An overview of $I^2C$ operations and additional information on the SSP module can be found in the PIC® Mid-Range MCU Family Reference Manual (DS33023).

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

### 10.2 SPI Mode

This section contains register definitions and operational characteristics of the SPI module.

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

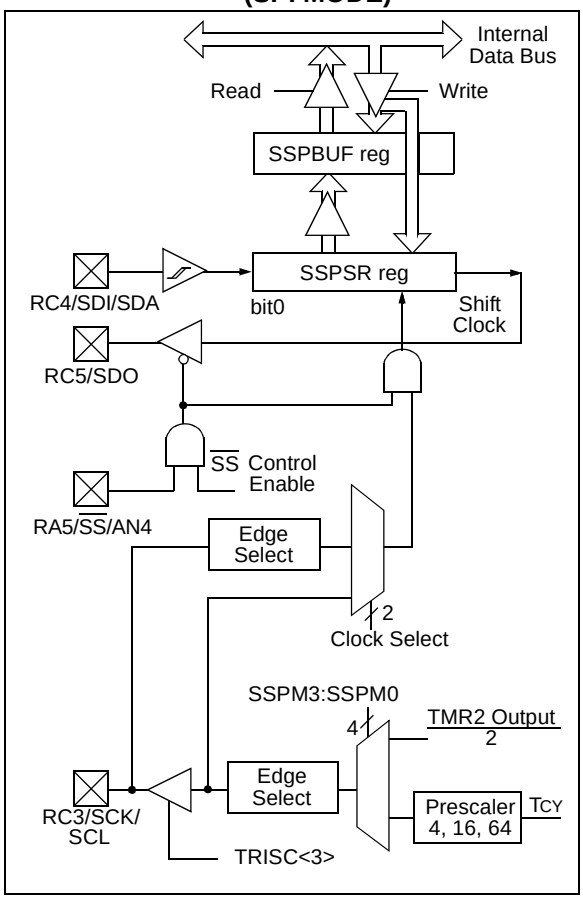
Additionally, a fourth pin may be used when in a Slave mode of operation:

- 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)

**FIGURE 10-1: 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 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
- ADCON1 must configure RA5 as a digital I/O pin.

.

**Note 1:** 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 VDD.

**2:** If the SPI is used in Slave mode with CKE = '1', then the $\overline{SS}$ pin control must be enabled.



Downloaded from Arrow.com.

**REGISTER 10-1: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS 94h)**

| 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 |
| bit 7 | | | | | | | bit 0 |

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 (Microwire®)
SPI Slave mode:
SMP must be cleared when SPI is used in Slave mode
$I^2C$ mode:
This bit must be maintained clear

bit 6 **CKE:** SPI Clock Edge Select (see Figure 10-2, Figure 10-3, and Figure 10-4)
SPI mode:
CKP = 0:
1 = Data transmitted on rising edge of SCK (Microwire alternate)
0 = Data transmitted on falling edge of SCK
CKP = 1:
1 = Data transmitted on falling edge of SCK (Microwire default)
0 = Data transmitted on rising edge of SCK
$I^2C$ mode:
This bit must be maintained clear

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/$\overline{Write}$ 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

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**REGISTER 10-2: 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 |
| bit 7 | | | | | | | bit 0 |

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

bit 6 **SSPOV:** Synchronous Serial Port Overflow Flag bit
In SPI mode:
1 = A new byte was received while the SSPBUF register is still holding the previous unread 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 was received while the SSPBUF register is still holding the previous unread 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. When enabled, the SSP pins must be properly configured as input or output.
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

bit 4 **CKP:** Clock Polarity Select bit
In SPI mode:
1 = Idle state for clock is a high level (Microwire default)
0 = Idle state for clock is a low level (Microwire alternate)
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

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |



Downloaded from Arrow.com.

**FIGURE 10-2: SPI MODE TIMING, MASTER MODE**

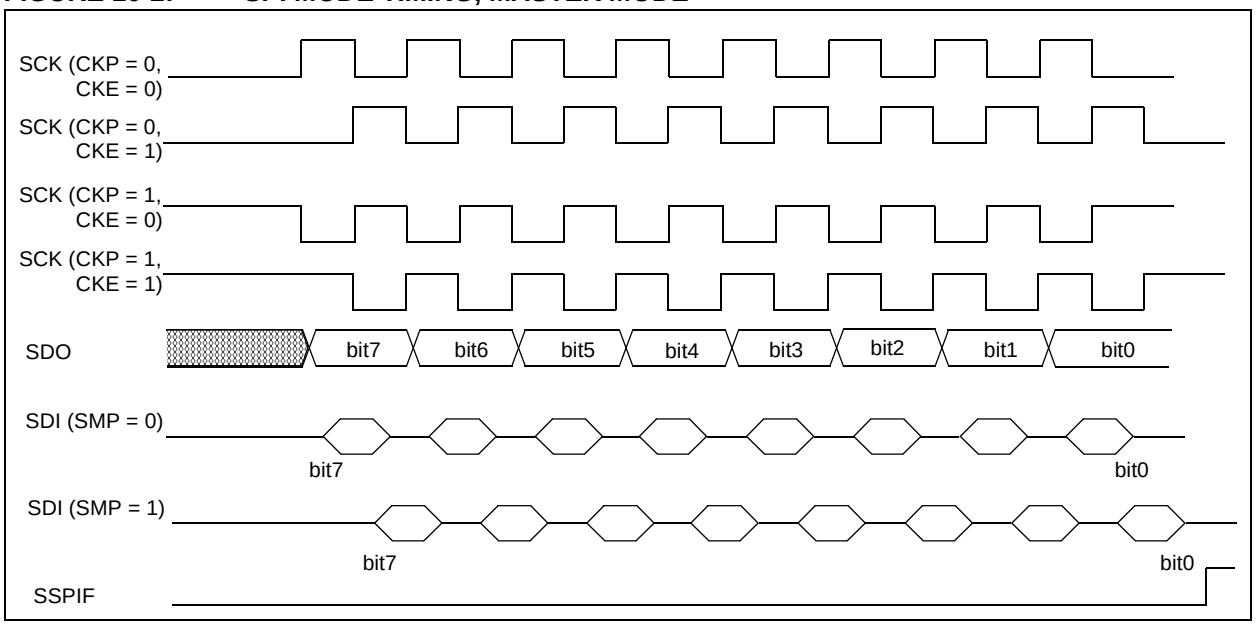
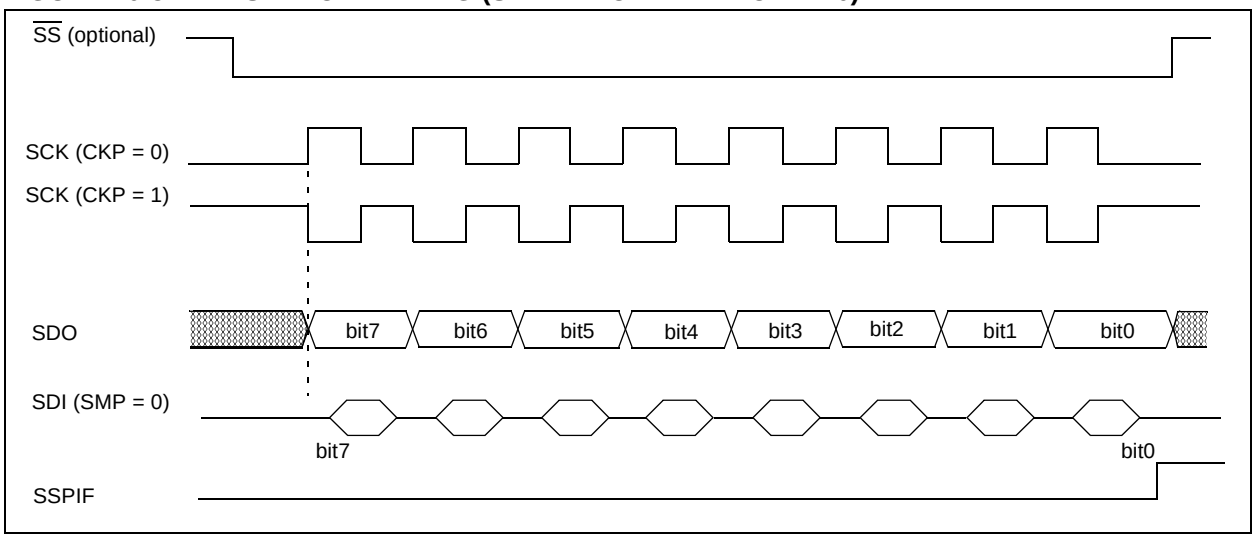



**FIGURE 10-3: SPI MODE TIMING (SLAVE MODE WITH CKE = 0)**





© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 10-4: SPI MODE TIMING (SLAVE MODE WITH CKE = 1)**

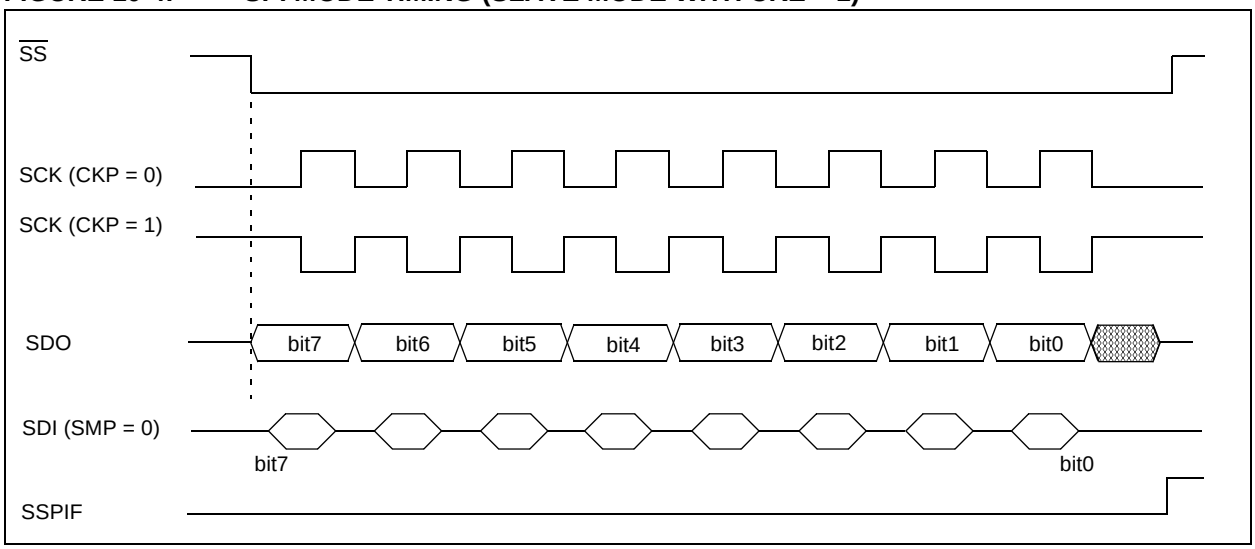



**TABLE 10-1: REGISTERS ASSOCIATED WITH SPI OPERATION**

| 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) | ADIF(2) | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE(2) | 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 PIC16C63A/73B; always maintain these bits clear.

**2:** Bits ADIE and ADIF are reserved on the PIC16C63A/65B; always maintain these bits clear.



Downloaded from Arrow.com.

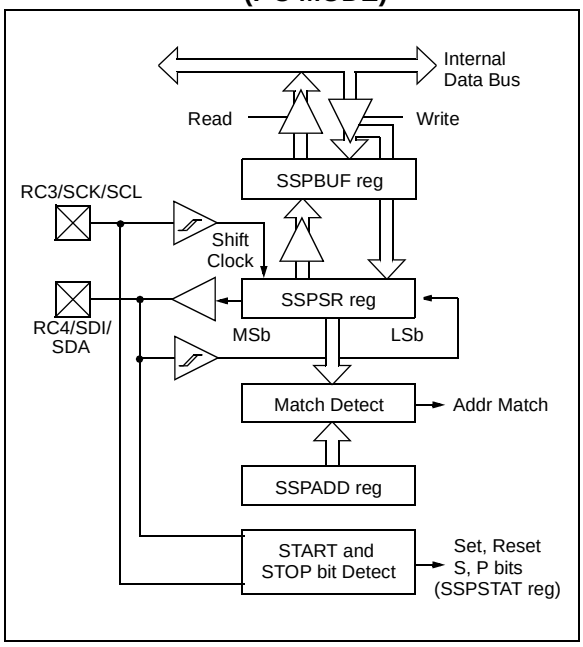
## 10.3 SSP I²C Operation

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 implementation 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, 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. External pull-up resistors for the SCL and SDA pins must be provided in the application circuit for proper operation of the I²C module.

The SSP module functions are enabled by setting SSP enable bit SSPEN (SSPCON<5>).

**FIGURE 10-5: 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 to support firmware Master mode
- I²C Slave mode (10-bit address), with START and STOP bit interrupts enabled to support firmware Master mode
- I²C START and STOP bit interrupts enabled to support firmware 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.

Additional information on SSP I²C operation can be found in the PIC® Mid-Range MCU Family Reference Manual (DS33023).

### 10.3.1 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 generates the acknowledge ($\overline{ACK}$) pulse, and then loads 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. They include (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 10-2 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 minimum high and low times for proper operation. The high and low times of the I²C specification, as well as the requirement of the SSP module, is shown in timing parameter #100 and parameter #101.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

#### 10.3.1.1 Addressing

Once the SSP module has been enabled, it waits for a START condition to occur. Following the START condition, 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 10-7). 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 10-2: DATA TRANSFER RECEIVED BYTE ACTIONS**

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

**Note:** Shaded cells show the conditions where the user software did not properly clear the overflow condition.



Downloaded from Arrow.com.

#### 10.3.1.2 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.

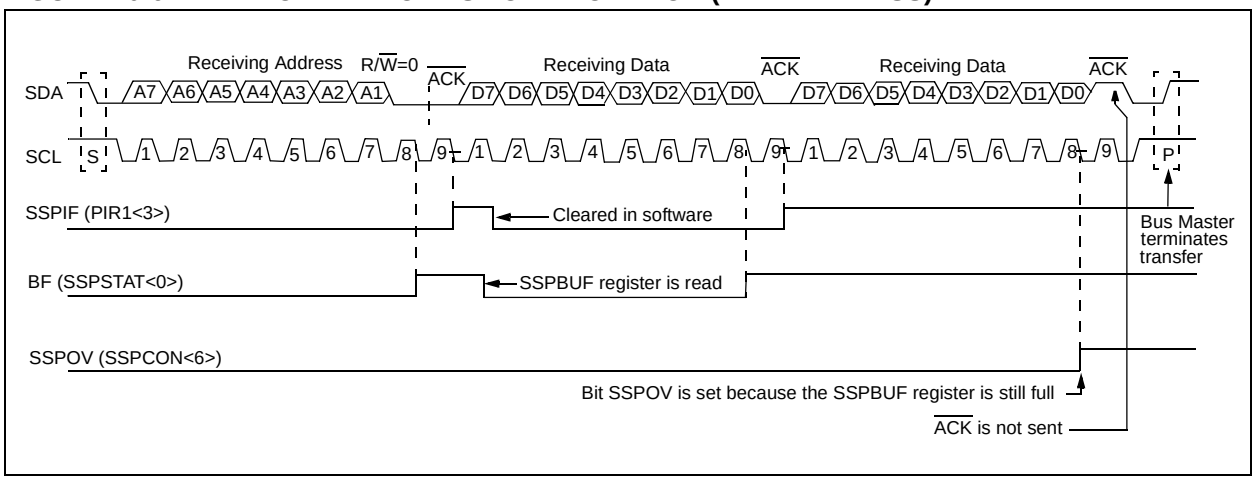
When the address byte overflow condition exists, then no acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as any situation where a received byte in SSPBUF is overwritten by the next received byte before it has been read. An overflow has occurred when:

a) The Buffer Full flag bit, BF(SSPSTAT<0>) was set, indicating that the byte in SSPBUF was waiting to be read when another byte was received. This sets the SSPOV flag.

b) The overflow flag, SSPOV (SSPCON1<6>) was 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 10-6: I²C WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**





© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

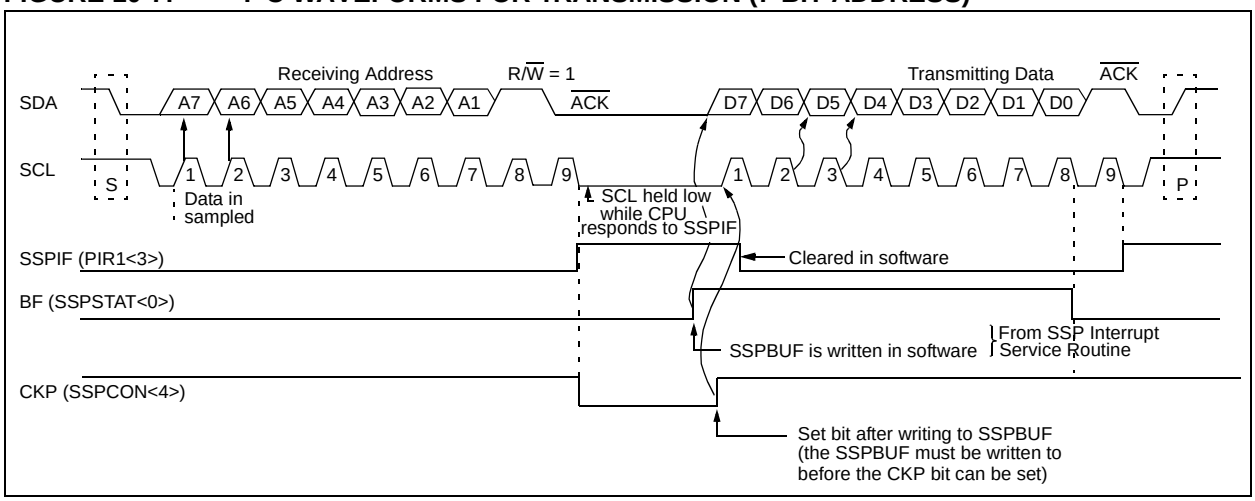
#### 10.3.1.3 Transmission

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 10-7).

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 10-7: I²C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**






Downloaded from Arrow.com.

### 10.3.2 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 (an SSP Interrupt will occur, 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.

### 10.3.3 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 10-3: 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 | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF(2) | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | ADIE(2) | 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(3) | CKE(3) | 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 $I^2C$ mode.

**Note 1:** PSPIF and PSPIE are reserved on the PIC16C63A/73B; always maintain these bits clear.
**2:** ADIF and ADIE are reserved on the PIC16C63A/65B; always maintain these bits clear.
**3:** Maintain these bits clear in $I^2C$ mode.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 11.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART)

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)

Bits SPEN (RCSTA<7>) and 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.

**REGISTER 11-1: 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 |
| bit 7 | | | | | | | bit 0 |

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
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.

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |



Downloaded from Arrow.com.

**REGISTER 11-2: 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 |
| bit 7 | | | | | | | bit 0 |

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:
Don't care

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. Calculated by firmware.)

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 11.1 USART Baud Rate Generator (BRG)

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 11-1 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 11-1. From this, the error in baud rate can be determined.

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.

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.

### 11.1.1 SAMPLING

The data on the RC7/RX/DT pin is sampled three times near the center of each bit time by a majority detect circuit to determine if a high or a low level is present at the RX pin.

**TABLE 11-1: BAUD RATE FORMULA**

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

**TABLE 11-2: 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.



Downloaded from Arrow.com.

## 11.2 USART Asynchronous Mode

In this mode, the USART uses standard non-return-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

### 11.2.1 USART ASYNCHRONOUS TRANSMITTER

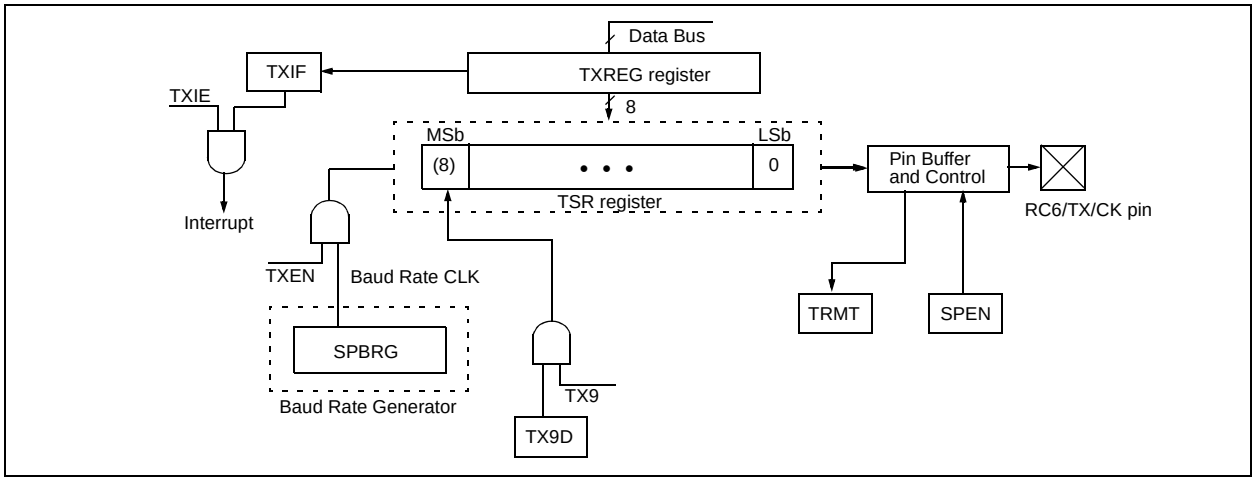
The USART transmitter block diagram is shown in Figure 11-1. The heart of the transmitter is the transmit (serial) shift register (TSR). The shift register obtains its data from the read/write transmit buffer, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until 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 the USART Transmit Flag bit TXIF (PIR1<4>) is set. This interrupt can be enabled/disabled by setting/clearing the USART Transmit Enable bit TXIE (PIE1<4>). The 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. 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. |
| **2:** | Flag bit TXIF is set when enable bit TXEN is set. TXIF is cleared by loading TXREG. |

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 11-2). 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. At that point, 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 11-3). 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 may be loaded in the TSR register.

**FIGURE 11-1: USART TRANSMIT BLOCK DIAGRAM**





© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

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 11.1)
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, set interrupt enable bits TXIE (PIE1<4>), PEIE (INTCON<6>), and GIE (INTCON<7>), as required.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN, which will also set flag 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 11-2: ASYNCHRONOUS MASTER TRANSMISSION**

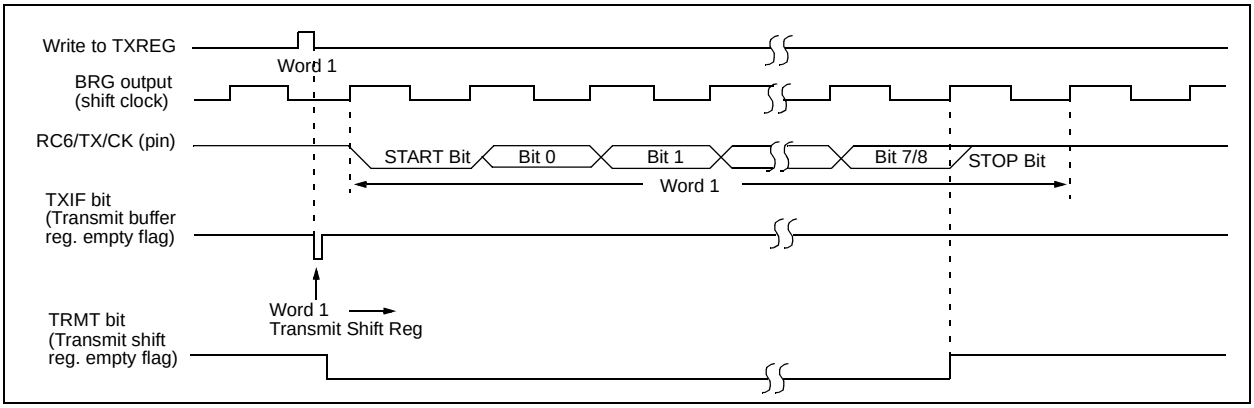
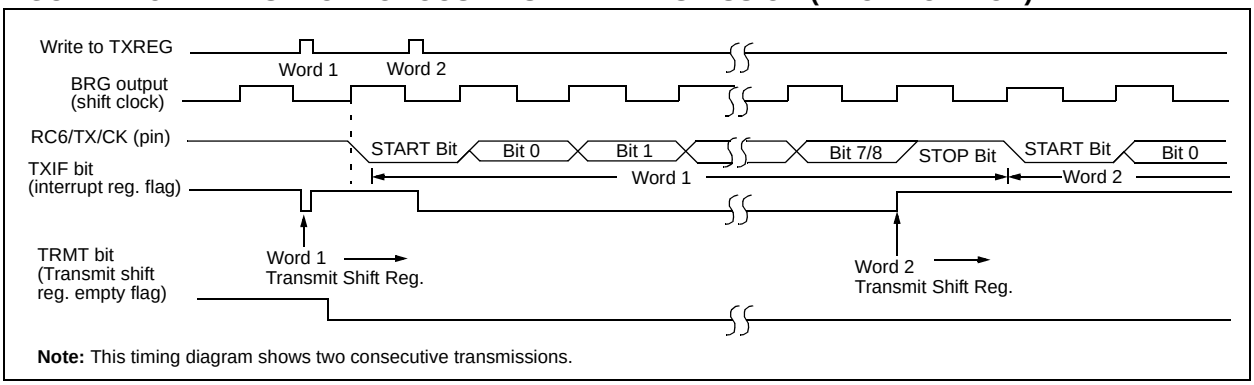



**FIGURE 11-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)**




**Note:** This timing diagram shows two consecutive transmissions.

**TABLE 11-3: 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF(2) | 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(2) | 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: u = unchanged, 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 PIC16C63A/73B; always maintain these bits clear.
**2:** Bits ADIE and ADIF are reserved on the PIC16C63A/65B; always maintain these bits clear.



Downloaded from Arrow.com.

### 11.2.2 USART ASYNCHRONOUS RECEIVER

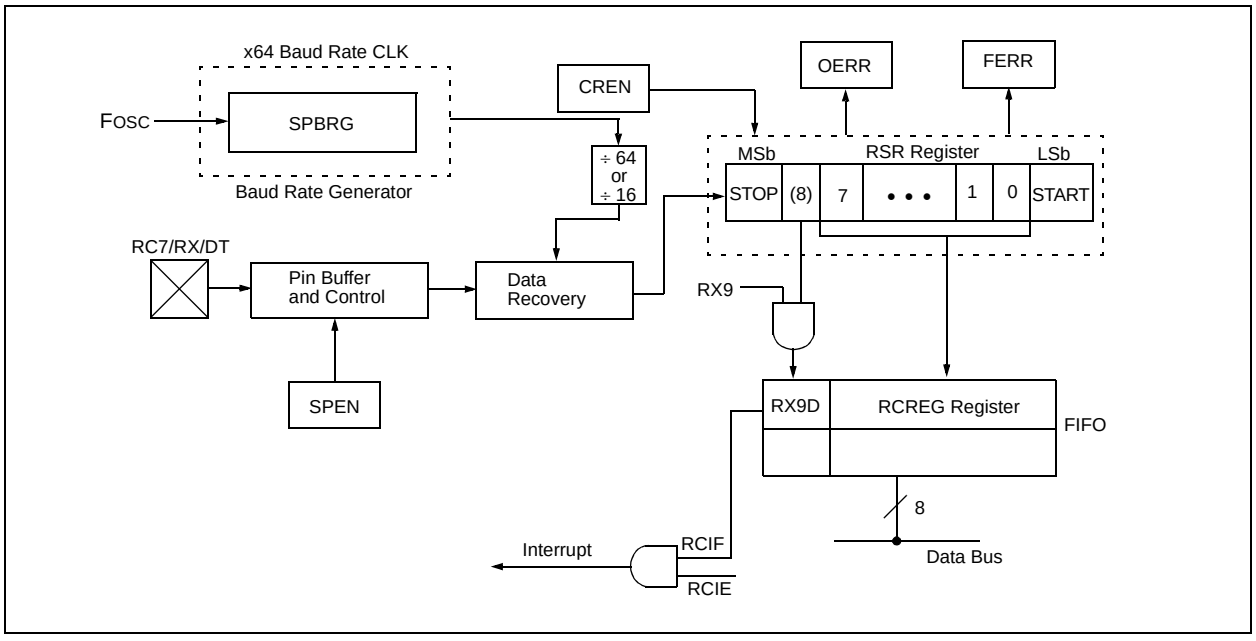
The receiver block diagram is shown in Figure 11-4. The data is received on the RC7/RX/DT pin and drives the data recovery block. The data recovery block is actually a high speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC.

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, USART Receive Flag bit RCIF (PIR1<5>) is set. This interrupt can be enabled/disabled by setting/clearing the USART Receive 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 to 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, and no further data will be received; therefore, 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 <u>before</u> reading the RCREG register, in order not to lose the old FERR and RX9D information.

**FIGURE 11-4: USART RECEIVE BLOCK DIAGRAM**





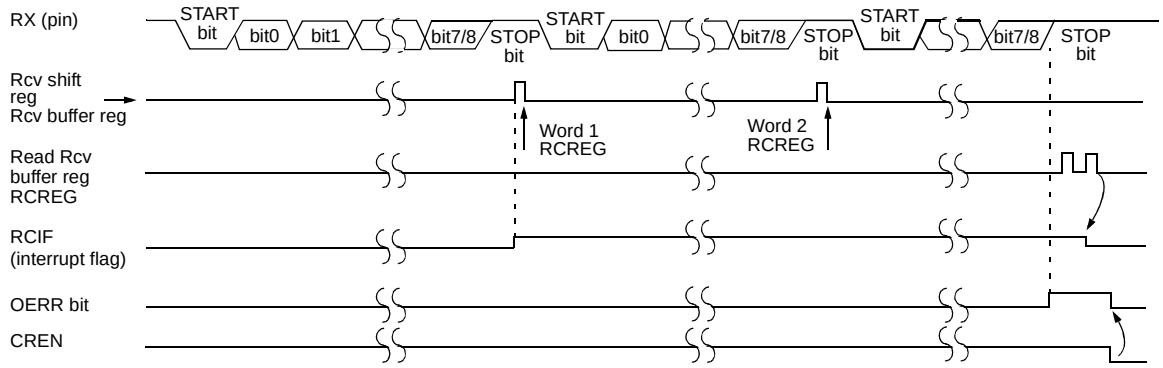
© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

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 11.1).
2. Enable the asynchronous serial port by clearing bit SYNC, and setting bit SPEN.
3. If interrupts are desired, set interrupt enable bits RCIE (PIE1<5>), PEIE (INTCON<6>), and GIE (INTCON<7>), as required.
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.

**FIGURE 11-5: 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. An overrun error indicates an error in user's firmware.

**TABLE 11-4: 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF(2) | 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(2) | 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: u = unchanged, 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.
**2:** Bits ADIE and ADIF are reserved on the PIC16C63A/65B; always maintain these bits clear.



Downloaded from Arrow.com.

### 11.2.3 USART SYNCHRONOUS MASTER MODE

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>).

### 11.2.4 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 11-1. 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 flag 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 11-6). The transmission can also be started by first loading the TXREG register and then setting bit TXEN (Figure 11-7). 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 11.1).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. If interrupts are desired, set interrupt enable bits TXIE (PIE1<4>), PEIE (INTCON<6>), and GIE (INTCON<7>), as required.
4. If 9-bit transmission is desired, 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.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

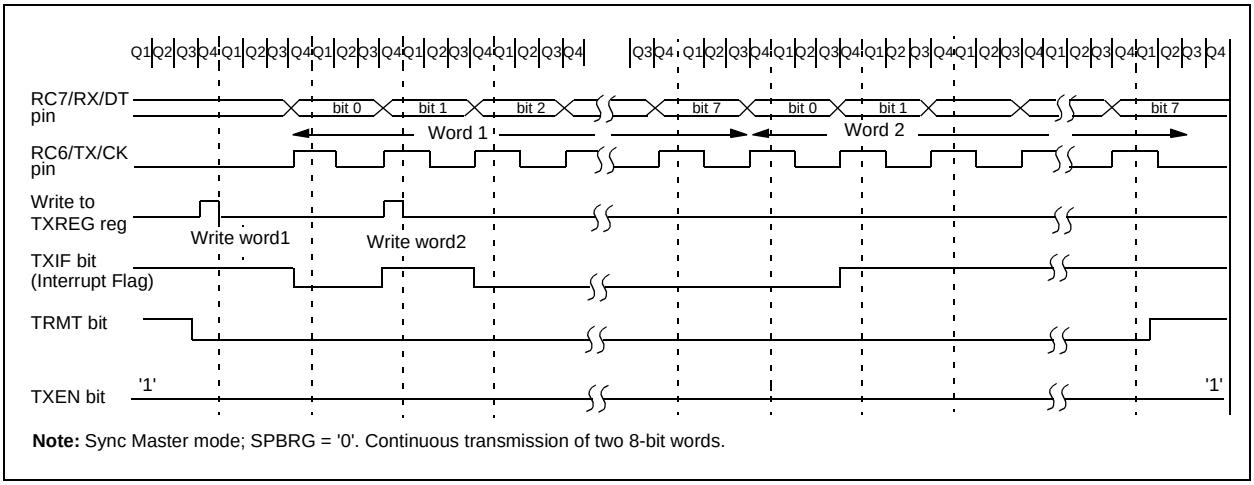
**TABLE 11-5: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF(2) | 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(2) | 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: u = unchanged, x = unknown, - = unimplemented, read as '0'.
Shaded cells are not used for synchronous master transmission.

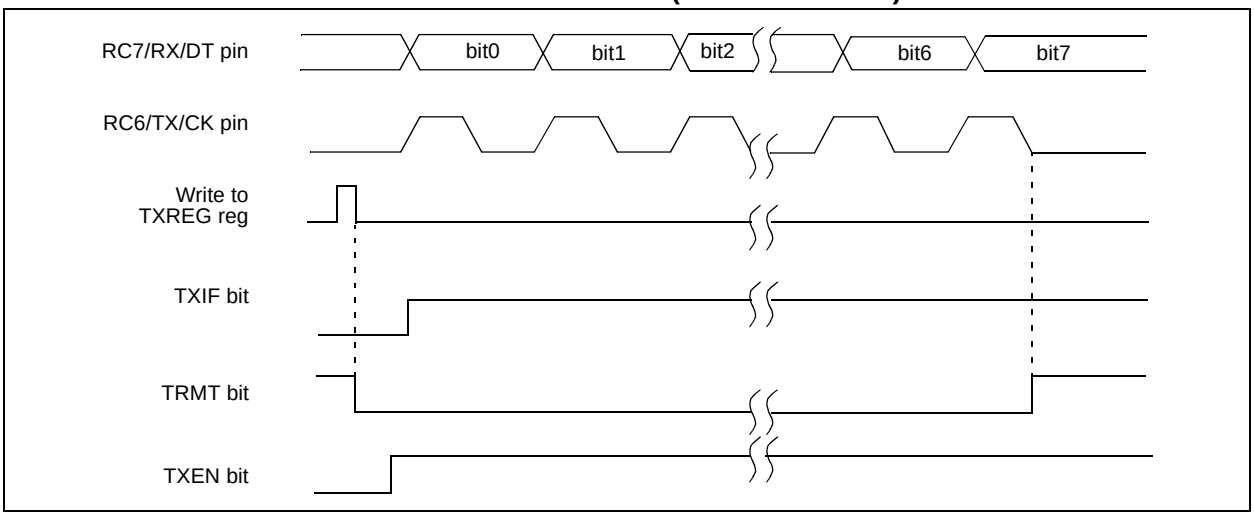
**Note 1:** Bits PSPIE and PSPIF are reserved on the PIC16C63A/73B; always maintain these bits clear.
**2:** Bits ADIE and ADIF are reserved on the PIC16C63A/65B; always maintain these bits clear.

**FIGURE 11-6: SYNCHRONOUS TRANSMISSION**




**Note:** Sync Master mode; SPBRG = '0'. Continuous transmission of two 8-bit words.

**FIGURE 11-7: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**






Downloaded from Arrow.com.

### 11.2.5 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, 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 interrupt from the USART 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, and no further data will be received; therefore, it is essential to clear bit OERR if it is set. The ninth 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 11.1)
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, set interrupt enable bits RCIE (PIE1<5>), PEIE (INTCON<6>), and GIE (INTCON<7>), as required.
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.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

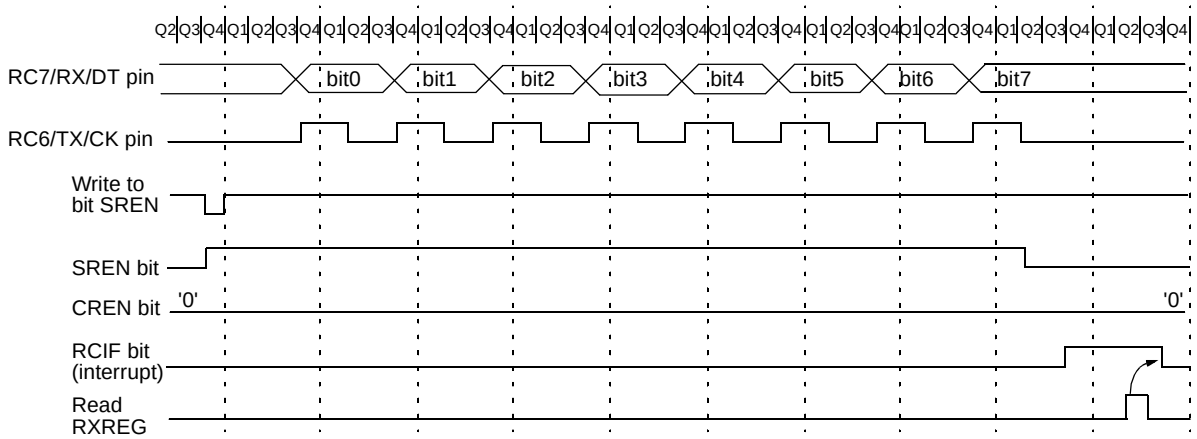
**TABLE 11-6: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF(2) | 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(2) | 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: u = unchanged, 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 PIC16C63A/73B; always maintain these bits clear.
**2:** Bits ADIE and ADIF are reserved on the PIC16C63A/65B; always maintain these bits clear.

**FIGURE 11-8: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**




**Note:** Timing diagram demonstrates SYNC Master mode with bit SREN = '1' and bit BRG = '0'.



Downloaded from Arrow.com.

## 11.3 USART Synchronous Slave Mode

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>).

### 11.3.1 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 interrupt enable bits TXIE and PEIE are set, the interrupt will wake the chip from SLEEP. If GIE is set, the program will branch to the interrupt vector (0004h), otherwise execution will resume from the instruction following the SLEEP instruction.

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, set interrupt enable bits TXIE (PIE1<4>), PEIE (INTCON<6>), and GIE (INTCON<7>), as required.
4. If 9-bit transmission is desired, 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.

### 11.3.2 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, a word may be received during SLEEP. On completely receiving the word, the RSR register will transfer the data to the RCREG register. If interrupt enable bits RCIE and PEIE are 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), otherwise execution will resume from the instruction following the SLEEP instruction.

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, set interrupt enable bits RCIE (PIE1<5>), PEIE (INTCON<6>), and GIE (INTCON<7>), as required.
3. If 9-bit reception is desired, 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.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 11-7: 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF(2) | 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(2) | 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: u = unchanged, 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 PIC16C63A/73B; always maintain these bits clear.
**2:** Bits ADIE and ADIF are reserved on the PIC16C63A/65B; always maintain these bits clear.

**TABLE 11-8: 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh,8Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF(1) | ADIF(2) | 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(2) | 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: u = unchanged, 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 PIC16C63A/73B; always maintain these bits clear.
**2:** Bits ADIE and ADIF are reserved on the PIC16C63A/65B; always maintain these bits clear.



Downloaded from Arrow.com.

**NOTES:**


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 12.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

| **Note:** | The PIC16C63A and PIC16C65B do not include A/D modules. ADCON0, ADCON1 and ADRES registers are not implemented. ADIF and ADIE bits are reserved and should be maintained clear. |
|---|---|

The 8-bit Analog-to-Digital (A/D) converter module has five inputs for the PIC16C73B and eight for the PIC16C74B.

The A/D allows conversion of an analog input signal to a corresponding 8-bit digital number. 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 Register 12-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 12-2, 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.

Additional information on using the A/D module can be found in the PIC® Mid-Range MCU Family Reference Manual (DS33023) and in Application Note, AN546.

**REGISTER 12-1: 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 |
| bit 7 | | | | | | | bit 0 |

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 the internal A/D module 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 shut-off and consumes no operating current

**Note 1:** A/D channels 5, 6 and 7 are implemented on the PIC16C74B only.

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |



Downloaded from Arrow.com.

**REGISTER 12-2: 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 |
| bit 7 | | | | | | | bit 0 |

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 | VDD |

A = Analog input D = Digital I/O

**Note 1:** RE0, RE1 and RE2 are implemented on the PIC16C74B only.

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

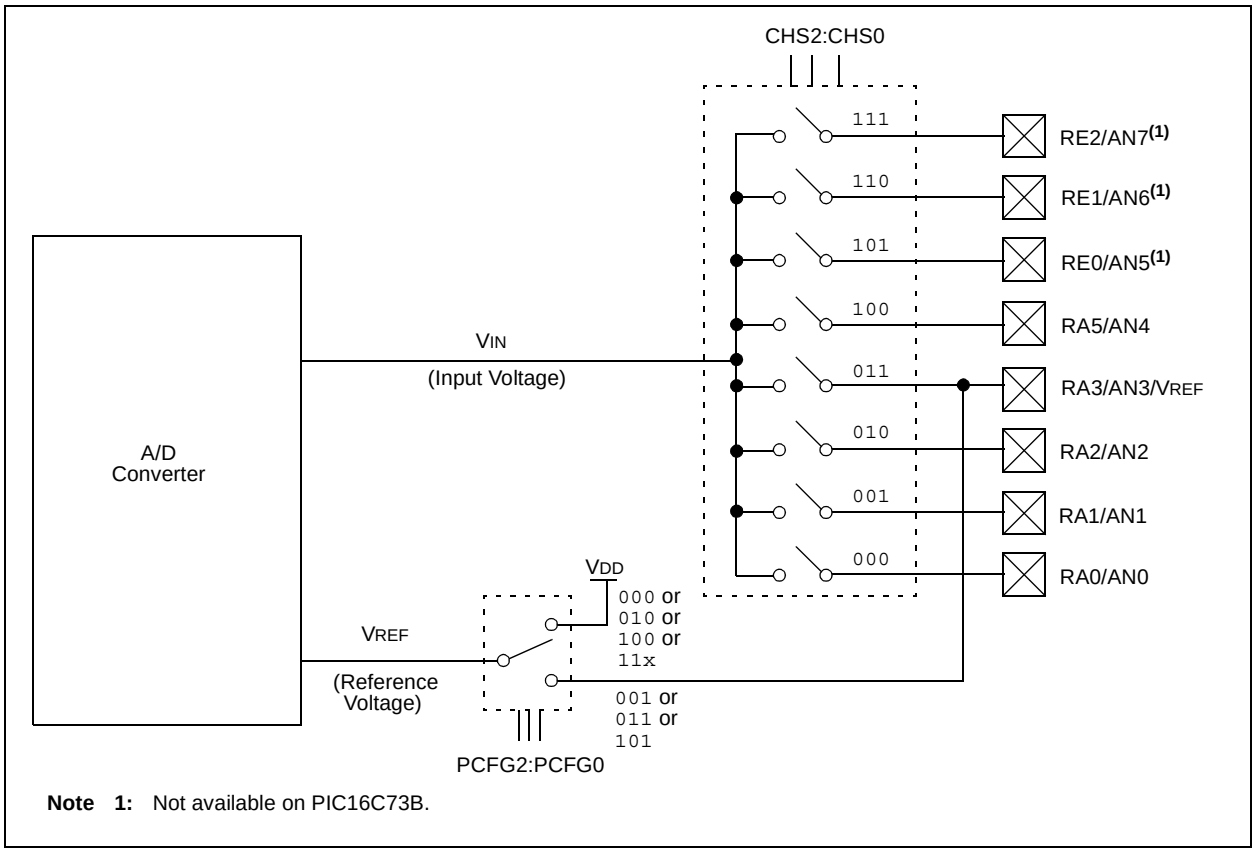
The following steps should be followed for doing an A/D conversion:

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 (PIR1<6>)
   - Set ADIE bit (PIE1<6>)
   - Set PEIE bit (INTCON<6>)
   - Set GIE bit (INTCON<7>)
3. Wait the required acquisition time.
4. Set GO/$\overline{\text{DONE}}$ bit (ADCON0) to start conversion.
5. Wait for A/D conversion to complete, by either: Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared (if interrupts are disabled);

   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 2 TAD is required before next acquisition starts.

**FIGURE 12-1: A/D BLOCK DIAGRAM**




**Note 1:** Not available on PIC16C73B.



Downloaded from Arrow.com.

## 12.1 A/D Acquisition Requirements

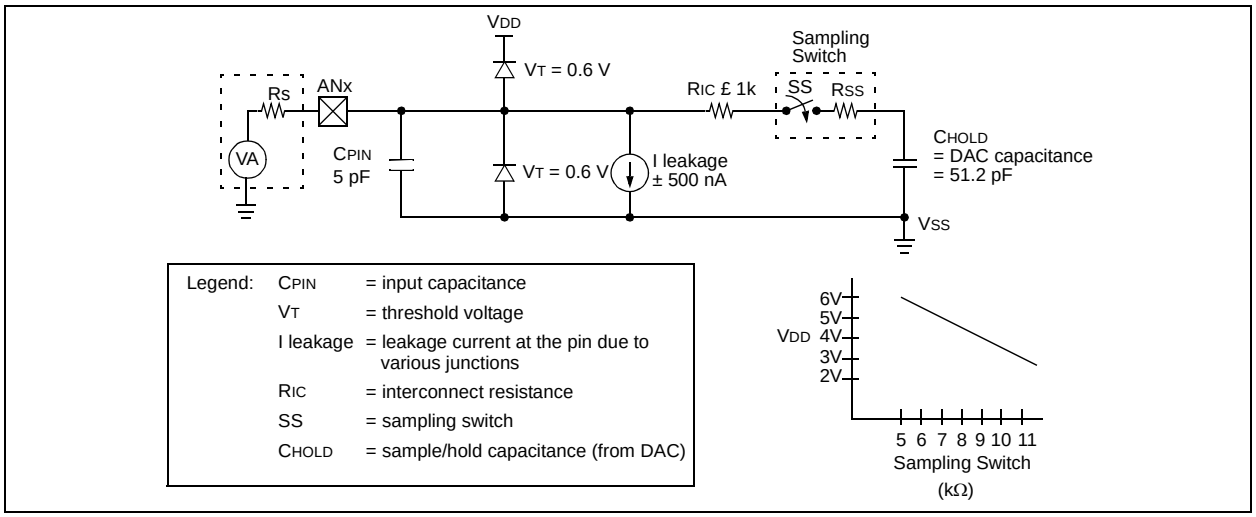
For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 12-2. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), Figure 12-2. 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), the acquisition time (TACQ) must pass before the conversion can be started.

To calculate the minimum acquisition time, Equation 12-1 may be used. This equation assumes that 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 resolution.

For more information, see the PIC® Mid-Range MCU Family Reference Manual (DS33023). In general, however, given a maximum source impedance of 10 kΩ and a worst case temperature of 100°C, TACQ will be no more than 16 µsec.

**FIGURE 12-2: ANALOG INPUT MODEL**




**EQUATION 12-1: ACQUISITION TIME**

TACQ = Amplifier Settling Time +
Hold Capacitor Charging Time +
Temperature Coefficient

= TAMP + TC + TCOFF
TAMP = 5 µS
TC = - (51.2 pF)(1 kΩ + RSS + RS) In(1/511)
TCOFF = (Temp -25°C)(0.05 µS/°C)


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 12.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 9.5 TAD per 8-bit conversion. The source of the A/D conversion clock is software selectable. The four possible options for TAD are:

- 2 TOSC
- 8 TOSC
- 32 TOSC
- Internal RC oscillator (2 - 6 μS)

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time (parameter #130).

## 12.3 Configuring Analog Port Pins

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. |
| **2:** | Analog levels on any pin that is defined as a digital input, but not as an analog input, may cause the input buffer to consume current that is out of the devices specification. |
| **3:** | The TRISE register is not provided on the PIC16C73B. |

## 12.4 A/D Conversions

| | |
|---|---|
| **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 2 TAD wait is required before the next acquisition is started. After this 2 TAD wait, an acquisition is automatically started on the selected channel. The GO/$\overline{\text{DONE}}$ bit can then be set to start another conversion.

## 12.5 A/D Operation During SLEEP

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. |

## 12.6 Effects of a RESET

A device RESET forces all registers to their RESET state. The A/D module is disabled and any conversion in progress is aborted. All pins with analog functions are configured as analog inputs.

The ADRES register will contain unknown data after a Power-on Reset.

## 12.7 Use of the CCP Trigger

An A/D conversion can be started by the “special event trigger” of the CCP2 module. 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.



Downloaded from Arrow.com.

**TABLE 12-1: SUMMARY OF A/D REGISTERS (PIC16C73B/74B ONLY)**

| 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) | 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 | — | — | — | — | — | — | — | CCP1IF | ---- ---0 | ---- ---0 |
| 8Dh | PIE2 | — | — | — | — | — | — | — | CCP1IE | ---- ---0 | ---- ---0 |
| 1Eh | ADRES | A/D Result register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/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 PIC6C63A/73B; always maintain these bits clear.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 13.0 SPECIAL FEATURES OF THE CPU

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 (ICSP)

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 and is 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, WDT 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.

### 13.1 Configuration Bits

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, and can be accessed only during programming.

**REGISTER 13-1: CONFIGURATION WORD (CONFIG 2007h)**

| CP1 | CP0 | CP1 | CP0 | CP1 | CP0 | — | BODEN | CP1 | CP0 | $\overline{\text{PWRTE}}$ | WDTE | FOSC1 | FOSC0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bit 13 | | | | | | | | | | | | | bit 0 |

bits 13-8, 5-4 **CP1:CP0**: Code Protection bits(2)
- 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 $\overline{\text{PWRTE}}$.

**2:** All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed.



Downloaded from Arrow.com.

## 13.2 Oscillator Configurations

### 13.2.1 OSCILLATOR TYPES

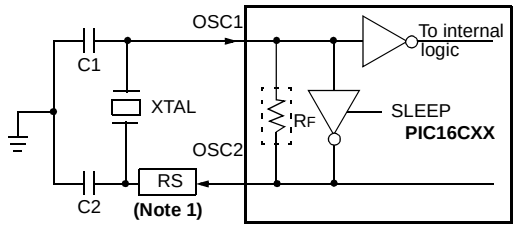
The PIC16CXX can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

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

### 13.2.2 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 13-1). 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 13-2). See the PIC® Mid-Range MCU Reference Manual (DS33023) for details on building an external oscillator.

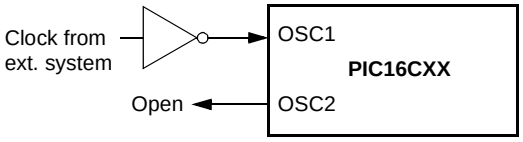
**FIGURE 13-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)**




See Table 13-1 and Table 13-2 for recommended values of C1 and C2.

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

**FIGURE 13-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)**





© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 13-1: 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 |
| **Note:** **These values are for design guidance only.** See notes following Table 13-1 and Table 13-2. | | | |
| **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% | |
| **Note:** Resonators used did not have built-in capacitors. | | | |

**TABLE 13-2: 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 |
| Note: **These values are for design guidance only.** See notes following Table 13-1 and Table 13-2. | | | |
| **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:** Higher capacitance increases the stability of the 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.

**3:** Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.

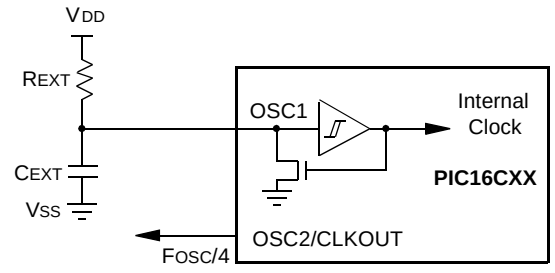
**4:** Oscillator performance should be verified at the expected voltage and temperature extremes in which the application is expected to operate.

### 13.2.3 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. The oscillator frequency will vary from unit to unit due to normal process variation. 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 13-3 shows how the R/C combination is connected to the PIC16CXX.

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 3-2 for waveform).

**FIGURE 13-3: RC OSCILLATOR MODE**




Recommended Values: REXT = 3 kW to 100 kW
CEXT = 20 pf to 30 pF



Downloaded from Arrow.com.

## 13.3 RESET

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)

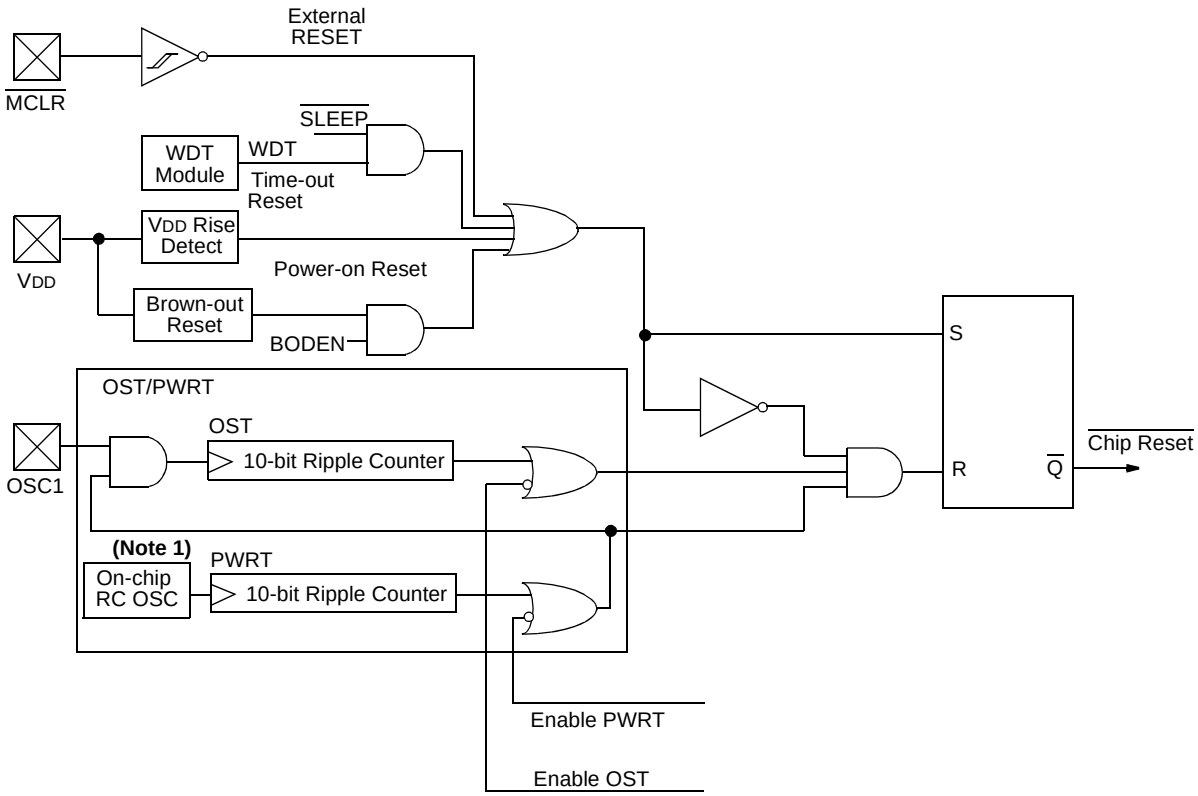
Some registers are not affected in any RESET condition; their status is unknown on POR and unchanged in any other RESET. Most other registers are reset to a "RESET state" on POR, on the $\overline{MCLR}$ and WDT Reset, on $\overline{MCLR}$ Reset during SLEEP, and on BOR. The $\overline{TO}$ and $\overline{PD}$ bits are set or cleared differently in different RESET situations, as indicated in Table 13-4. These bits are used in software to determine the nature of the RESET. See Table 13-6 for a full description of RESET states of all registers.

A simplified block diagram of the on-chip RESET circuit is shown in Figure 13-4.

The PIC devices 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 internal RESET sources do not drive $\overline{MCLR}$ pin low.

**FIGURE 13-4: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**




**Note 1:** This is a separate oscillator from the RC oscillator of the CLKIN pin.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 13.4 RESETS

### 13.4.1 POWER-ON RESET (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected (parameters D003 and D004, in the range of 1.5V - 2.1V). To take advantage of the POR, just tie the MCLR pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create a POR.

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. The device may be held in RESET by keeping MCLR at Vss.

For additional information, refer to Application Note AN607, "*Power-up Trouble Shooting*."

### 13.4.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 72 ms nominal time-out on power-up from the POR. The PWRT operates on an internal RC oscillator. The device 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 (TPWRT, parameter #33).

### 13.4.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer provides a delay of 1024 oscillator cycles (from OSC1 input) after the PWRT delay, if enabled. This helps to ensure 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.

### 13.4.4 BROWN-OUT RESET (BOR)

The configuration bit, BODEN, can enable or disable the Brown-out Reset circuit. If VDD falls below VBOR (parameter D005, about 4V) for longer than TBOR (parameter #35, about 100μS), the brown-out situation will reset the device. If VDD falls below VBOR for less than TBOR, a RESET may not occur.

Once the brown-out occurs, the device will remain in Brown-out Reset until VDD rises above VBOR. The Power-up Timer then keeps the device in RESET for TPWRT (parameter #33, about 72mS). If VDD should fall below VBOR during TPWRT, the Brown-out Reset process will restart when VDD rises above VBOR with the Power-up Timer Reset. The Power-up Timer is always enabled when the Brown-out Reset circuit is enabled, regardless of the state of the PWRT configuration bit.

### 13.4.5 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows: the PWRT delay starts (if enabled) when a POR occurs. Then, OST starts counting 1024 oscillator cycles when PWRT ends (LP, XT, HS). When the OST ends, the device comes out of RESET.

If MCLR is kept low long enough, the time-outs will expire. Bringing MCLR high will begin execution immediately. This is useful for testing purposes or to synchronize more than one PIC16CXX device operating in parallel.

Table 13-5 shows the RESET conditions for the STATUS, PCON and PC registers, while Table 13-6 shows the RESET conditions for all the registers.

### 13.4.6 POWER CONTROL/STATUS REGISTER (PCON)

The Brown-out Reset Status bit, BOR, is unknown on a POR. It must be set by the user and checked on subsequent RESETS to see if bit BOR was cleared, indicating a BOR occurred. The BOR bit is not predictable if the Brown-out Reset circuitry is disabled.

The Power-on Reset Status bit, POR, is cleared on a POR and unaffected otherwise. The user must set this bit following a POR and check it on subsequent RESETS to see if it has been cleared.



Downloaded from Arrow.com.

**TABLE 13-3: TIME-OUT IN VARIOUS SITUATIONS**

| 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 13-4: STATUS BITS AND THEIR SIGNIFICANCE**

| $\overline{POR}$ | $\overline{BOR}$ | $\overline{TO}$ | $\overline{PD}$ | |
|---|---|---|---|---|
| 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 | 1 | 1 | 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 |

Legend: x = don't care, u = unchanged

**TABLE 13-5: RESET CONDITION FOR SPECIAL REGISTERS**

| Condition | Program Counter | STATUS Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 000h | 0001 1xxx | ---- --0x |
| $\overline{MCLR}$ Reset during normal operation | 000h | 000u uuuu | ---- --uu |
| $\overline{MCLR}$ Reset during SLEEP | 000h | 0001 0uuu | ---- --uu |
| WDT Reset | 000h | 0000 1uuu | ---- --uu |
| WDT Wake-up | PC + 1 | uuu0 0uuu | ---- --uu |
| Brown-out Reset | 000h | 000x xuuu | ---- --u0 |
| Interrupt wake-up from SLEEP | PC + 1(1) | uuu1 0uuu | ---- --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).

**REGISTER 13-2: STATUS REGISTER**

| IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C |
|---|---|---|---|---|---|---|---|

**REGISTER 13-3: PCON REGISTER**

| — | — | — | — | — | — | $\overline{POR}$ | $\overline{BOR}$ |
|---|---|---|---|---|---|---|---|


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 13-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS**

| Register | Applicable Devices | | | | Power-on Reset Brown-out Reset | $\overline{\text{MCLR}}$ Resets WDT Reset | Wake-up via WDT or Interrupt |
|---|---|---|---|---|---|---|---|
| W | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| INDF | 63A | 65B | 73B | 74B | N/A | N/A | N/A |
| TMR0 | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PCL | 63A | 65B | 73B | 74B | 0000h | 0000h | PC + 1(2) |
| STATUS | 63A | 65B | 73B | 74B | 0001 1xxx | 000q quuu(3) | uuuq quuu(3) |
| FSR | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTA | 63A | 65B | 73B | 74B | --0x 0000 | --0u 0000 | --uu uuuu |
| PORTB | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTC | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTD | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTE | 63A | 65B | 73B | 74B | ---- -xxx | ---- -uuu | ---- -uuu |
| PCLATH | 63A | 65B | 73B | 74B | ---0 0000 | ---0 0000 | ---u uuuu |
| INTCON | 63A | 65B | 73B | 74B | 0000 000x | 0000 000u | uuuu uuuu(1) |
| PIR1 | 63A | 65B | 73B | 74B | -0-- 0000 | -0-- 0000 | -u-- uuuu(1) |
| | 63A | 65B | 73B | 74B | -000 0000 | -000 0000 | -uuu uuuu(1) |
| | 63A | 65B | 73B | 74B | 0000 0000 | 0000 0000 | uuuu uuuu(1) |
| | 63A | 65B | 73B | 74B | 0000 0000 | 0000 0000 | uuuu uuuu(1) |
| PIR2 | 63A | 65B | 73B | 74B | ---- ---0 | ---- ---0 | ---- ---u(1) |
| TMR1L | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| TMR1H | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| T1CON | 63A | 65B | 73B | 74B | --00 0000 | --uu uuuu | --uu uuuu |
| TMR2 | 63A | 65B | 73B | 74B | 0000 0000 | 0000 0000 | uuuu uuuu |
| T2CON | 63A | 65B | 73B | 74B | -000 0000 | -000 0000 | -uuu uuuu |
| SSPBUF | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| SSPCON | 63A | 65B | 73B | 74B | 0000 0000 | 0000 0000 | uuuu uuuu |
| CCPR1L | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCPR1H | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCP1CON | 63A | 65B | 73B | 74B | --00 0000 | --00 0000 | --uu uuuu |
| RCSTA | 63A | 65B | 73B | 74B | 0000 -00x | 0000 -00x | uuuu -uuu |
| TXREG | 63A | 65B | 73B | 74B | 0000 0000 | 0000 0000 | uuuu uuuu |
| RCREG | 63A | 65B | 73B | 74B | 0000 0000 | 0000 0000 | uuuu uuuu |
| CCPR2L | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCPR2H | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCP2CON | 63A | 65B | 73B | 74B | 0000 0000 | 0000 0000 | uuuu uuuu |
| ADRES | 63A | 65B | 73B | 74B | xxxx xxxx | uuuu uuuu | uuuu 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 13-5 for RESET value for specific condition.



Downloaded from Arrow.com.

**TABLE 13-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)**

| Register | Applicable Devices | | | | Power-on Reset Brown-out Reset | $\overline{\text{MCLR}}$ Resets WDT Reset | Wake-up via WDT or Interrupt |
|---|---|---|---|---|---|---|---|
| ADCON0 | 63A | 65B | 73B | 74B | 0000 00-0 | 0000 00-0 | uuuu uu-u |
| OPTION_REG | 63A | 65B | 73B | 74B | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISA | 63A | 65B | 73B | 74B | --11 1111 | --11 1111 | --uu uuuu |
| TRISB | 63A | 65B | 73B | 74B | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISC | 63A | 65B | 73B | 74B | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISD | 63A | 65B | 73B | 74B | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISE | 63A | 65B | 73B | 74B | 0000 -111 | 0000 -111 | uuuu -uuu |
| PIE1 | 63A | 65B | 73B | 74B | --00 0000 | --00 0000 | --uu uuuu |
| | 63A | 65B | 73B | 74B | 0-00 0000 | 0-00 0000 | u-uu uuuu |
| | 63A | 65B | 73B | 74B | -000 0000 | -000 0000 | -uuu uuuu |
| | 63A | 65B | 73B | 74B | 0000 0000 | 0000 0000 | uuuu uuuu |
| PIE2 | 63A | 65B | 73B | 74B | ---- ---0 | ---- ---0 | ---- ---u |
| PCON | 63A | 65B | 73B | 74B | ---- --0q(3) | ---- --uu | ---- --uu |
| PR2 | 63A | 65B | 73B | 74B | 1111 1111 | 1111 1111 | 1111 1111 |
| SSPADD | 63A | 65B | 73B | 74B | 0000 0000 | 0000 0000 | uuuu uuuu |
| SSPSTAT | 63A | 65B | 73B | 74B | --00 0000 | --00 0000 | --uu uuuu |
| TXSTA | 63A | 65B | 73B | 74B | 0000 -010 | 0000 -010 | uuuu -uuu |
| SPBRG | 63A | 65B | 73B | 74B | 0000 0000 | 0000 0000 | uuuu uuuu |
| ADCON1 | 63A | 65B | 73B | 74B | ---- -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 13-5 for RESET value for specific condition.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 13.5 Interrupts

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 unmasked 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. 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, PEIE bit, or the GIE bit.

**Note:** 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 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.

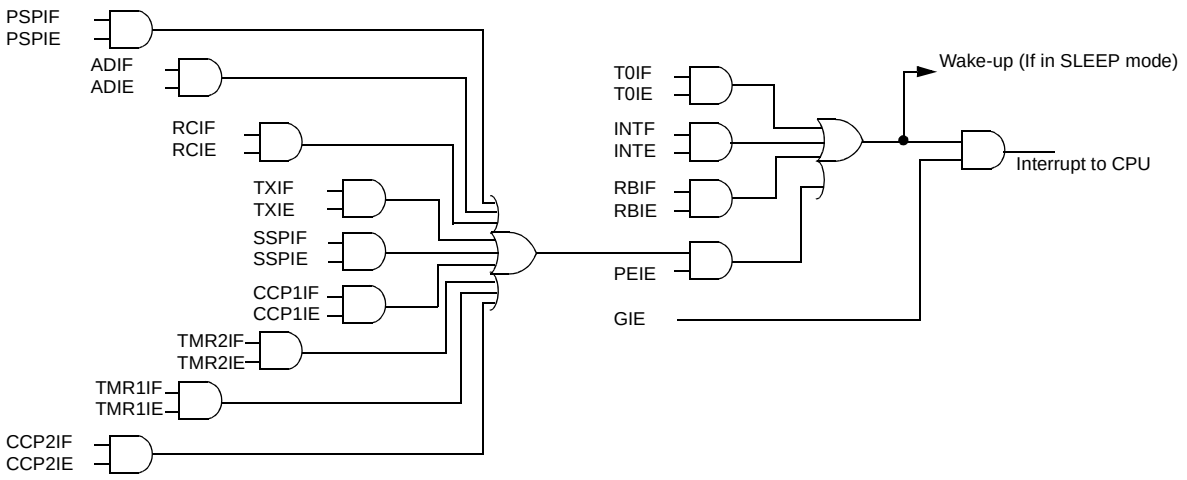
Perform the following to ensure that interrupts are globally disabled:

```
LOOP BCF   INTCON, GIE  ; Disable global
                        ; interrupt bit
     BTFSC INTCON, GIE  ; Global interrupt
                        ; disabled?
     GOTO  LOOP         ; NO, try again
        :               ; Yes, continue
                        ; with program
                        ; flow
```



Downloaded from Arrow.com.

**FIGURE 13-5: INTERRUPT LOGIC**




The following table shows which devices have which interrupts.

| Device | T0IF | INTF | RBIF | PSPIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | CCP2IF |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIC16C63A | Yes | Yes | Yes | – | – | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| PIC16C65B | Yes | Yes | Yes | Yes | – | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| PIC16C73B | Yes | Yes | Yes | – | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| PIC16C74B | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |

### 13.5.1 INT INTERRUPT

The external interrupt on RB0/INT pin is edge triggered: either rising if bit INTEDG (OPTION_REG<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 13.8 for details on SLEEP mode.

### 13.5.2 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>) (see Section 6.0).

### 13.5.3 PORTB INTERRUPT-ON-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:** | 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. |
|---|---|


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 13.6 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. 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 13-1 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 13-1: 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
:
(ISR)                         ;User ISR code goes here
:
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
```

## 13.7 Watchdog Timer (WDT)

The Watchdog Timer is 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. 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 resume normal operation (Watchdog Timer Wake-up).

The WDT can be permanently disabled by clearing configuration bit WDTE (Section 13.1).

### 13.7.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms (parameter #31, TWDT). The time-out periods vary with temperature, VDD, and process variations. 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. Time-out periods up to 128 TWDT can be realized.

The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT. In addition, the SLEEP instruction prevents the WDT from generating a RESET, but will allow the WDT to wake the device from SLEEP mode.

The $\overline{TO}$ bit in the STATUS register will be cleared upon a WDT time-out.



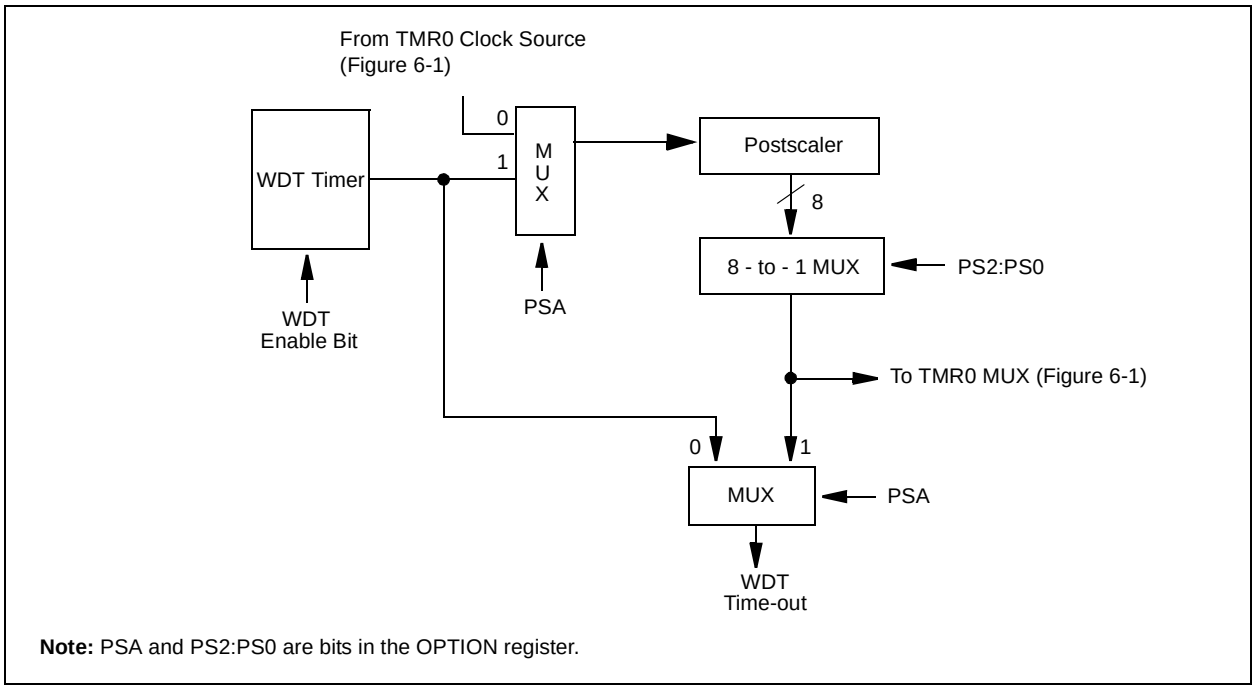
Downloaded from Arrow.com.

### 13.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken into account that under worst case conditions ($V_{DD}$ = 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 13-6: WATCHDOG TIMER BLOCK DIAGRAM**




**Note:** PSA and PS2:PS0 are bits in the OPTION register.

**TABLE 13-7: 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 | – | BODEN(1) | CP1 | CP0 | $\overline{\text{PWRTE}}$(1) | WDTE | FOSC1 | FOSC0 |
| 81h | OPTION_REG | $\overline{\text{RBPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |

Legend: Shaded cells are not used by the Watchdog Timer.
**Note 1:** See Register 13-1 for operation of these bits.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 13.8 Power-down Mode (SLEEP)

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

If enabled, the WDT 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 VDD or VSS, ensure no external circuitry is drawing current from the I/O pin, power-down the A/D, and 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 VDD or VSS for lowest current consumption. The contribution from on-chip pull-ups on PORTB should also be considered.

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

### 13.8.1 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 a Peripheral Interrupt.

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 (PIC16C65B/74B only).
6. A/D conversion (when A/D clock source is RC).
7. 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.

### 13.8.2 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 $\overline{PD}$ bit 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.

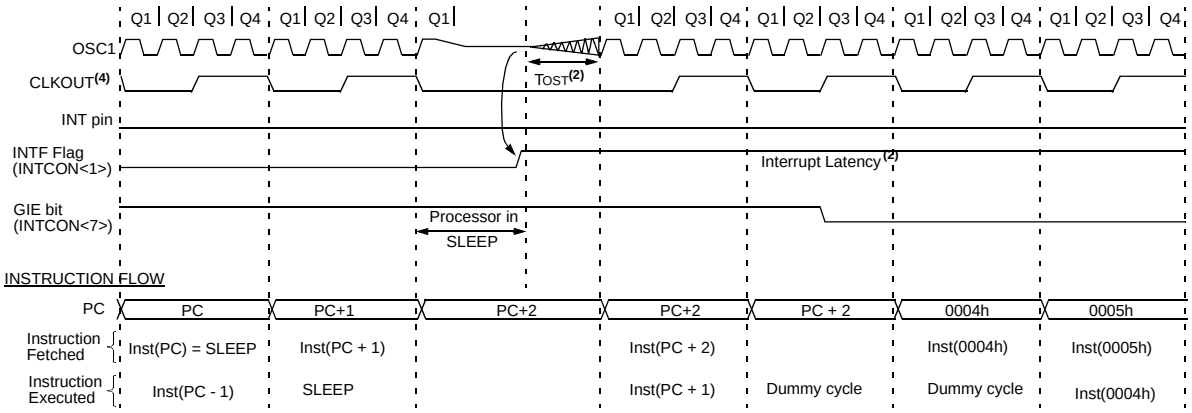
Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the SLEEP instruction was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

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



Downloaded from Arrow.com.

**FIGURE 13-7: WAKE-UP FROM SLEEP THROUGH INTERRUPT**




**Note 1:** XT, HS or LP oscillator mode assumed.
**2:** TOST = 1024Tosc (drawing not to scale). This delay is not present in RC osc mode.
**3:** GIE = '1' assumed. 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.

## 13.9 Program Verification/Code Protection

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

| **Note:** | Microchip does not recommend code protecting windowed devices. Devices that are code protected may be erased, but not programmed again. |
|---|---|

## 13.10 ID Locations

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

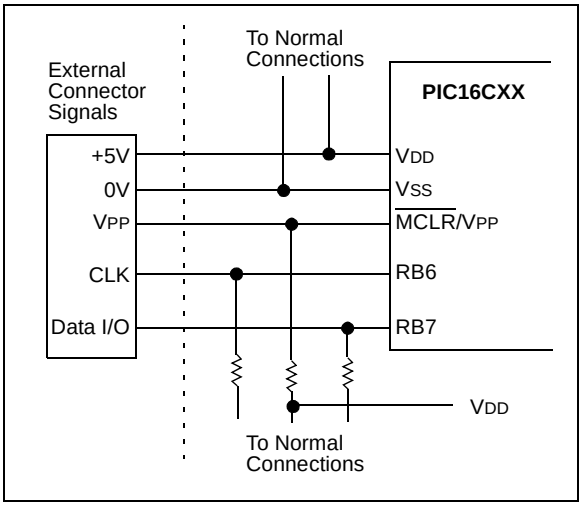
## 13.11 In-Circuit Serial Programming

PIC16CXX microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the RB6 and RB7 pins low, while raising the MCLR (VPP) pin from VIL to VIHH (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt Trigger inputs in this mode.

After RESET, to place the device into Programming/Verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14 bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X Programming Specifications (Literature #DS30228).

**FIGURE 13-8: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**





© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 14.0 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 14-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 14-1 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 address 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 14-1: 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 |
| → | Assigned to |
| < > | Register bit field |
| ∈ | 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 14-2 lists the instructions recognized by the MPASM™ assembler.

Figure 14-1 shows the general formats that the instructions can have.

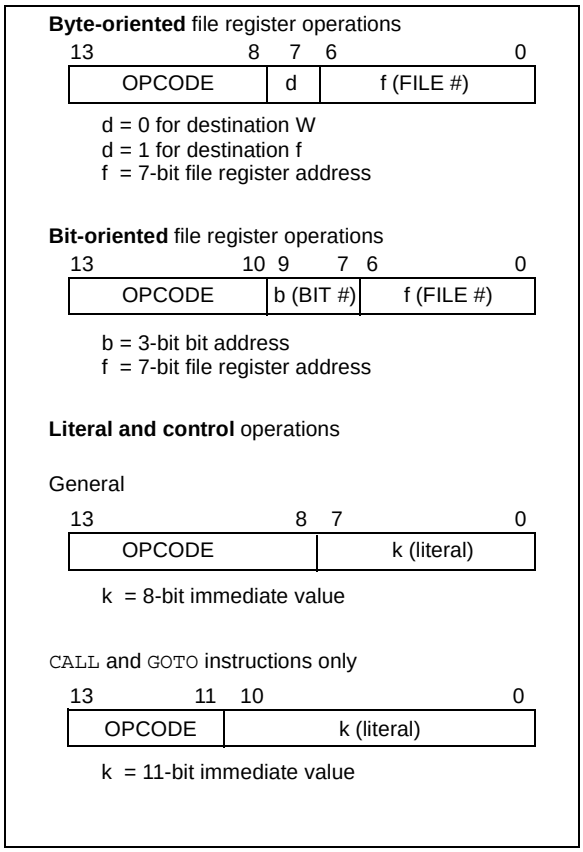
**Note:** To maintain upward compatibility with future PIC16CXX products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

**FIGURE 14-1: GENERAL FORMAT FOR INSTRUCTIONS**






Downloaded from Arrow.com.

**TABLE 14-2: 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 0000 0011 | | 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.

**Note:** Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 14.1 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 |
| Description: | The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register. |

| ADDWF | Add W and f |
|---|---|
| Syntax: | [*label*] ADDWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (W) + (f) → (destination) |
| Status Affected: | C, DC, Z |
| 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'. |

| ANDLW | AND Literal with W |
|---|---|
| Syntax: | [*label*] ANDLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | (W) .AND. (k) → (W) |
| Status Affected: | Z |
| Description: | The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register. |

| ANDWF | AND W with f |
|---|---|
| Syntax: | [*label*] ANDWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (W) .AND. (f) → (destination) |
| Status Affected: | Z |
| 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'. |

| BCF | Bit Clear f |
|---|---|
| Syntax: | [*label*] BCF f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b \le 7$ |
| Operation: | 0 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is cleared. |

| BSF | Bit Set f |
|---|---|
| Syntax: | [*label*] BSF f,b |
| Operands: | $0 \le f \le 127$<br>$0 \le b \le 7$ |
| Operation: | 1 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is set. |



Downloaded from Arrow.com.

| BTFSS | Bit Test f, Skip if Set |
|---|---|
| Syntax: | [*label*] BTFSS f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b < 7 |
| Operation: | skip if (f<b>) = 1 |
| Status Affected: | None |
| Description: | If bit 'b' in register 'f' is '0', the next instruction is executed.<br>If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead making this a 2TCY instruction. |

| BTFSC | Bit Test, Skip if Clear |
|---|---|
| Syntax: | [*label*] BTFSC f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | skip if (f<b>) = 0 |
| Status Affected: | None |
| Description: | If bit 'b' in register 'f' is '1', the next instruction is executed.<br>If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2 TCY instruction. |

| CALL | Call Subroutine |
|---|---|
| Syntax: | [ *label* ] CALL k |
| Operands: | 0 ≤ k ≤ 2047 |
| Operation: | (PC)+ 1→ TOS,<br>k → PC<10:0>,<br>(PCLATH<4:3>) → PC<12:11> |
| Status Affected: | None |
| 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. |

| CLRF | Clear f |
|---|---|
| Syntax: | [*label*] CLRF f |
| Operands: | 0 ≤ f ≤ 127 |
| Operation: | 00h → (f)<br>1 → Z |
| Status Affected: | Z |
| Description: | The contents of register 'f' are cleared and the Z bit is set. |

| CLRW | Clear W |
|---|---|
| Syntax: | [ *label* ] CLRW |
| Operands: | None |
| Operation: | 00h → (W)<br>1 → Z |
| Status Affected: | Z |
| Description: | W register is cleared. Zero bit (Z) is set. |

| CLRWDT | Clear Watchdog Timer |
|---|---|
| Syntax: | [ *label* ] CLRWDT |
| Operands: | None |
| Operation: | 00h → WDT<br>0 → WDT prescaler,<br>1 → $\overline{TO}$<br>1 → $\overline{PD}$ |
| Status Affected: | $\overline{TO}$, $\overline{PD}$ |
| Description: | CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set. |


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

| COMF | Complement f |
|---|---|
| Syntax: | [ *label* ] COMF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | $(\bar{f}) \rightarrow$ (destination) |
| Status Affected: | Z |
| 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'. |

| GOTO | Unconditional Branch |
|---|---|
| Syntax: | [ *label* ] GOTO k |
| Operands: | $0 \le k \le 2047$ |
| Operation: | $k \rightarrow$ PC<10:0><br>PCLATH<4:3> $\rightarrow$ PC<12:11> |
| Status Affected: | None |
| 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. |

| DECF | Decrement f |
|---|---|
| Syntax: | [*label*] DECF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f) - 1 $\rightarrow$ (destination) |
| Status Affected: | Z |
| 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'. |

| INCF | Increment f |
|---|---|
| Syntax: | [ *label* ] INCF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f) + 1 $\rightarrow$ (destination) |
| Status Affected: | Z |
| 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'. |

| DECFSZ | Decrement f, Skip if 0 |
|---|---|
| Syntax: | [ *label* ] DECFSZ f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f) - 1 $\rightarrow$ (destination);<br>skip if result = 0 |
| Status Affected: | None |
| 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'.<br>If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead making it a 2 TCY instruction. |

| INCFSZ | Increment f, Skip if 0 |
|---|---|
| Syntax: | [ *label* ] INCFSZ f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f) + 1 $\rightarrow$ (destination),<br>skip if result = 0 |
| Status Affected: | None |
| 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'.<br>If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead making it a 2 TCY instruction. |



Downloaded from Arrow.com.

| IORLW | Inclusive OR Literal with W |
|---|---|
| Syntax: | [ *label* ] IORLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | (W) .OR. k → (W) |
| Status Affected: | Z |
| Description: | The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register. |

| IORWF | Inclusive OR W with f |
|---|---|
| Syntax: | [ *label* ] IORWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (W) .OR. (f) → (destination) |
| Status Affected: | Z |
| 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'. |

| MOVF | Move f |
|---|---|
| Syntax: | [ *label* ] MOVF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f) → (destination) |
| Status Affected: | Z |
| Description: | The contents of register f are 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. |

| MOVLW | Move Literal to W |
|---|---|
| Syntax: | [ *label* ] MOVLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | k → (W) |
| Status Affected: | None |
| Description: | The eight bit literal 'k' is loaded into W register. The don't cares will assemble as 0's. |

| MOVWF | Move W to f |
|---|---|
| Syntax: | [ *label* ] MOVWF f |
| Operands: | $0 \le f \le 127$ |
| Operation: | (W) → (f) |
| Status Affected: | None |
| Description: | Move data from W register to register 'f'. |

| NOP | No Operation |
|---|---|
| Syntax: | [ *label* ] NOP |
| Operands: | None |
| Operation: | No operation |
| Status Affected: | None |
| Description: | No operation. |


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**RETFIE** **Return from Interrupt**

Syntax: [ *label* ] RETFIE

Operands: None

Operation: TOS → PC,
1 → GIE

Status Affected: None

**RETLW** **Return with Literal in W**

Syntax: [ *label* ] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W);
TOS → PC

Status Affected: None

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.

**RETURN** **Return from Subroutine**

Syntax: [ *label* ] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

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.

**RLF** **Rotate Left f through Carry**

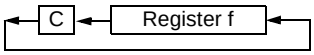
Syntax: [ *label* ] RLF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: See description below

Status Affected: C

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'.




**RRF** **Rotate Right f through Carry**

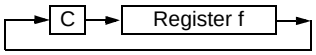
Syntax: [ *label* ] RRF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: See description below

Status Affected: C

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'.




**SLEEP**

Syntax: [ *label* ] SLEEP

Operands: None

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

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

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 13.8 for more details.



Downloaded from Arrow.com.

| SUBLW | Subtract W from Literal |
|---|---|
| Syntax: | [ *label* ] SUBLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | k - (W) → (W) |
| Status Affected: | C, DC, Z |
| Description: | The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register. |

| SUBWF | Subtract W from f |
|---|---|
| Syntax: | [ *label* ] SUBWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f) - (W) → (destination) |
| Status Affected: | C, DC, Z |
| 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'. |

| SWAPF | Swap Nibbles in f |
|---|---|
| Syntax: | [ *label* ] SWAPF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (f<3:0>) → (destination<7:4>),<br>(f<7:4>) → (destination<3:0>) |
| Status Affected: | None |
| 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'. |

| XORLW | Exclusive OR Literal with W |
|---|---|
| Syntax: | [*label*] XORLW k |
| Operands: | $0 \le k \le 255$ |
| Operation: | (W) .XOR. k → (W) |
| Status Affected: | Z |
| Description: | The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register. |

| XORWF | Exclusive OR W with f |
|---|---|
| Syntax: | [*label*] XORWF f,d |
| Operands: | $0 \le f \le 127$<br>$d \in [0,1]$ |
| Operation: | (W) .XOR. (f) → (destination) |
| Status Affected: | Z |
| 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'. |


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 15.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C17 and MPLAB C18 C Compilers
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - ICEPIC™ In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD for PIC16F87X
- Device Programmers
  - PRO MATE® II Universal Device Programmer
  - PICSTART® Plus Entry-Level Development Programmer
- Low Cost Demonstration Boards
  - PICDEM™ 1 Demonstration Board
  - PICDEM 2 Demonstration Board
  - PICDEM 3 Demonstration Board
  - PICDEM 17 Demonstration Board
  - KEELOQ® Demonstration Board

### 15.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8-bit microcontroller market. The MPLAB IDE is a Windows®-based application that contains:

- An interface to debugging tools
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
  - in-circuit debugger (sold separately)
- A full-featured editor
- A project manager
- Customizable toolbar and key mapping
- A status bar
- On-line help

The MPLAB IDE allows you to:

- Edit your source files (either assembly or ‘C’)
- One touch assemble (or compile) and download to PIC emulator and simulator tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
  - machine code

The ability to use MPLAB IDE with multiple debugging tools allows users to easily switch from the cost-effective simulator to a full-featured emulator with minimal retraining.

### 15.2 MPASM Assembler

The MPASM assembler is a full-featured universal macro assembler for all PIC MCUs.

The MPASM assembler has a command line interface and a Windows shell. It can be used as a stand-alone application on a Windows 3.x or greater system, or it can be used through MPLAB IDE. The MPASM assembler generates relocatable object files for the MPLINK object linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, an absolute LST file that contains source lines and generated machine code, and a COD file for debugging.

The MPASM assembler features include:

- Integration into MPLAB IDE projects.
- User-defined macros to streamline assembly code.
- Conditional assembly for multi-purpose source files.
- Directives that allow complete control over the assembly process.

### 15.3 MPLAB C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI ‘C’ compilers for Microchip’s PIC17CXXX and PIC18CXXX family of microcontrollers, respectively. These compilers provide powerful integration capabilities and ease of use not found with other compilers.

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



Downloaded from Arrow.com.

## 15.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can also link relocatable objects from pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian is a librarian for pre-compiled code to be used with the MPLINK object linker. When a routine from a library is called from another source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. The MPLIB object librarian manages the creation and modification of library files.

The MPLINK object linker features include:

- Integration with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- Allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- Easier linking because single libraries can be included instead of many smaller files.
- Helps keep code maintainable by grouping related modules together.
- Allows libraries to be created and modules to be added, listed, replaced, deleted or extracted.

## 15.5 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PIC series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user-defined key press, to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and the MPLAB C18 C compilers and the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent multi-project software development tool.

## 15.6 MPLAB ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB ICE universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers (MCUs). Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE in-circuit 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 platform and Microsoft® Windows environment were chosen to best make these features available to you, the end user.

## 15.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X and PIC16CXXX families of 8-bit One-Time-Programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules, or daughter boards. The emulator is capable of emulating without target application circuitry being present.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 15.8 MPLAB ICD In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD, is a powerful, low cost, run-time development tool. This tool is based on the FLASH PIC16F87X and can be used to develop for this and other PIC microcontrollers from the PIC16CXXX family. The MPLAB ICD utilizes the in-circuit debugging capability built into the PIC16F87X. This feature, along with Microchip's In-Circuit Serial Programming™ protocol, offers cost-effective in-circuit FLASH debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time.

## 15.9 PRO MATE II Universal Device Programmer

The PRO MATE II universal device programmer is a full-featured programmer, capable of operating in stand-alone mode, as well as PC-hosted mode. The PRO MATE II device programmer is CE compliant.

The PRO MATE II device programmer has programmable VDD and VPP supplies, which allow it to verify programmed memory at VDD min and VDD max for maximum reliability. It has an LCD display for instructions and 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 device programmer can read, verify, or program PIC devices. It can also set code protection in this mode.

## 15.10 PICSTART Plus Entry Level Development Programmer

The PICSTART Plus development programmer is an easy-to-use, low cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient.

The PICSTART Plus development programmer supports all PIC devices with up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus development programmer is CE compliant.

## 15.11 PICDEM 1 Low Cost PIC MCU Demonstration Board

The PICDEM 1 demonstration board 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 user can program the sample microcontrollers provided with the PICDEM 1 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The user can also connect the PICDEM 1 demonstration board to the MPLAB ICE in-circuit emulator and download the firmware to the emulator for testing. A 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.

## 15.12 PICDEM 2 Low Cost PIC16CXX Demonstration Board

The PICDEM 2 demonstration board 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 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 2 demonstration board to test firmware. A 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.



Downloaded from Arrow.com.

## 15.13 PICDEM 3 Low Cost PIC16CXXX Demonstration Board

The PICDEM 3 demonstration board 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 an 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 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer with an adapter socket, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 3 demonstration board to test firmware. A prototype area has been provided to the user for adding 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 thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM 3 demonstration board is a LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM 3 demonstration board provides an additional RS-232 interface and Windows 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.

## 15.14 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5-inch disk. A programmed sample is included and the user may erase it and program it with the other sample programs using the PRO MATE II device programmer, or the PICSTART Plus development programmer, and easily debug and test the sample code. In addition, the PICDEM 17 demonstration board supports downloading of programs to and executing out of external FLASH memory on board. The PICDEM 17 demonstration board is also usable with the MPLAB ICE in-circuit emulator, or the PICMASTER emulator and all of the sample programs can be run and modified using either emulator. Additionally, a generous prototype area is available for user hardware.

## 15.15 KEELOQ Evaluation and Programming Tools

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


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 15-1: DEVELOPMENT TOOLS FROM MICROCHIP**

| | | PIC12CXXX | PIC14000 | PIC16C5X | PIC16C6X | PIC16CXXX | PIC16F62X | PIC16C7X | PIC16C7XX | PIC16C8X | PIC16F8XX | PIC16C9XX | PIC17C4X | PIC17C7XX | PIC18CXX2 | 24CXX/ 25CXX/ 93CXX | HCSXXX | MCRFXXX | MCP2510 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Software Tools | MPLAB® Integrated Development Environment | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | |
| | MPLAB® C17 C Compiler | | | | | | | | | | | | ✓ | ✓ | | | | | |
| | MPLAB® C18 C Compiler | | | | | | | | | | | | | | ✓ | | | | |
| | MPASM™ Assembler/ MPLINK™ Object Linker | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | |
| Emulators | MPLAB® ICE In-Circuit Emulator | ✓ | ✓ | ✓ | ✓ | ✓ | ✓** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | |
| | ICEPIC™ In-Circuit Emulator | ✓ | | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | | ✓ | | | | | | | |
| Debugger | MPLAB® ICD In-Circuit Debugger | | | | ✓* | | | ✓* | | | ✓ | | | | | | | | |
| Programmers | PICSTART® Plus Entry Level Development Programmer | ✓ | ✓ | ✓ | ✓ | ✓ | ✓** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | |
| | PRO MATE® II Universal Device Programmer | ✓ | ✓ | ✓ | ✓ | ✓ | ✓** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | |
| Demo Boards and Eval Kits | PICDEM™ 1 Demonstration Board | | | ✓ | | ✓ | | ✓† | | ✓ | | | ✓ | | | | | | |
| | PICDEM™ 2 Demonstration Board | | | | ✓† | | | ✓† | | | | | | | ✓ | | | | |
| | PICDEM™ 3 Demonstration Board | | | | | | | | | | | ✓ | | | | | | | |
| | PICDEM™ 14A Demonstration Board | | ✓ | | | | | | | | | | | | | | | | |
| | PICDEM™ 17 Demonstration Board | | | | | | | | | | | | | ✓ | | | | | |
| | KEELOQ® Evaluation Kit | | | | | | | | | | | | | | | | ✓ | | |
| | KEELOQ® Transponder Kit | | | | | | | | | | | | | | | | ✓ | | |
| | microID™ Programmer's Kit | | | | | | | | | | | | | | | | | ✓ | |
| | 125 kHz microID™ Developer's Kit | | | | | | | | | | | | | | | | | ✓ | |
| | 125 kHz Anticollision microID™ Developer's Kit | | | | | | | | | | | | | | | | | ✓ | |
| | 13.56 MHz Anticollision microID™ Developer's Kit | | | | | | | | | | | | | | | | | ✓ | |
| | MCP2510 CAN Developer's Kit | | | | | | | | | | | | | | | | | | ✓ |

* Contact the Microchip Technology Inc. web site at www.microchip.com for information on how to use the MPLAB® ICD In-Circuit Debugger (DV164001) with PIC16C62, 63, 64, 65, 72, 73, 74, 76, 77.
** Contact Microchip Technology Inc. for availability date.
† Development tool is available on select devices.



Downloaded from Arrow.com.

**NOTES:**


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 16.0 ELECTRICAL CHARACTERISTICS

**Absolute Maximum Ratings(†)**

Ambient temperature under bias ..... -55°C to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on any pin with respect to VSS (except VDD, MCLR and RA4) ..... -0.3V to (VDD + 0.3V)
Voltage on VDD with respect to VSS ..... -0.3V to +7.5V
Voltage on MCLR with respect to VSS **(Note 2)** ..... 0V to +13.25V
Voltage on RA4 with respect to VSS ..... 0V to +8.5V
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 **(Note 3)** (combined) ..... 200 mA
Maximum current sourced by PORTA, PORTB, and PORTE **(Note 3)** (combined) ..... 200 mA
Maximum current sunk by PORTC and PORTD **(Note 3)** (combined) ..... 200 mA
Maximum current sourced by PORTC and PORTD **(Note 3)** (combined) ..... 200 mA

**Note 1:** Power dissipation is calculated as follows: Pdis = VDD x {IDD - ∑ IOH} + ∑ {(VDD-VOH) x IOH} + ∑(Vol x IOL)

**2:** Voltage spikes below VSS at the MCLR/VPP 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 MCLR/VPP pin rather than pulling this pin directly to VSS.

**3:** PORTD and PORTE not available on the PIC16C63A/73B.

† 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.



Downloaded from Arrow.com.

**FIGURE 16-1: PIC16C63A/65B/73B/74B VOLTAGE-FREQUENCY GRAPH**

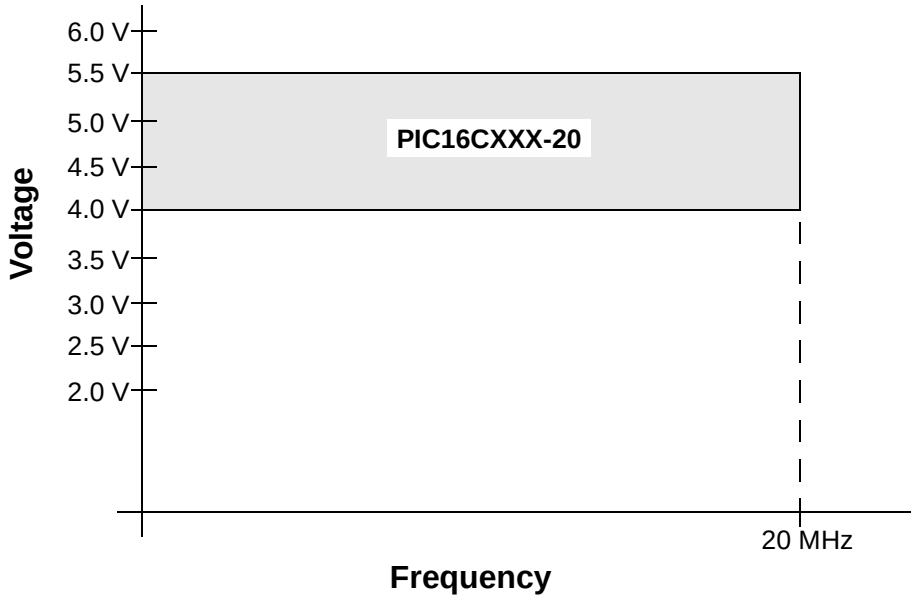
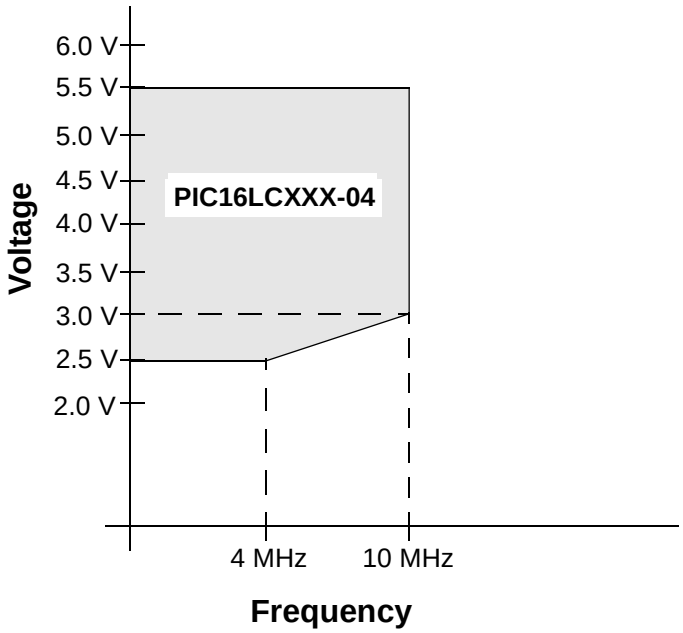



**FIGURE 16-2: PIC16LC63A/65B/73B/74B VOLTAGE-FREQUENCY GRAPH**




FMAX = (12.0 MHz/V) (VDDAPPMIN - 2.5 V) + 4 MHz

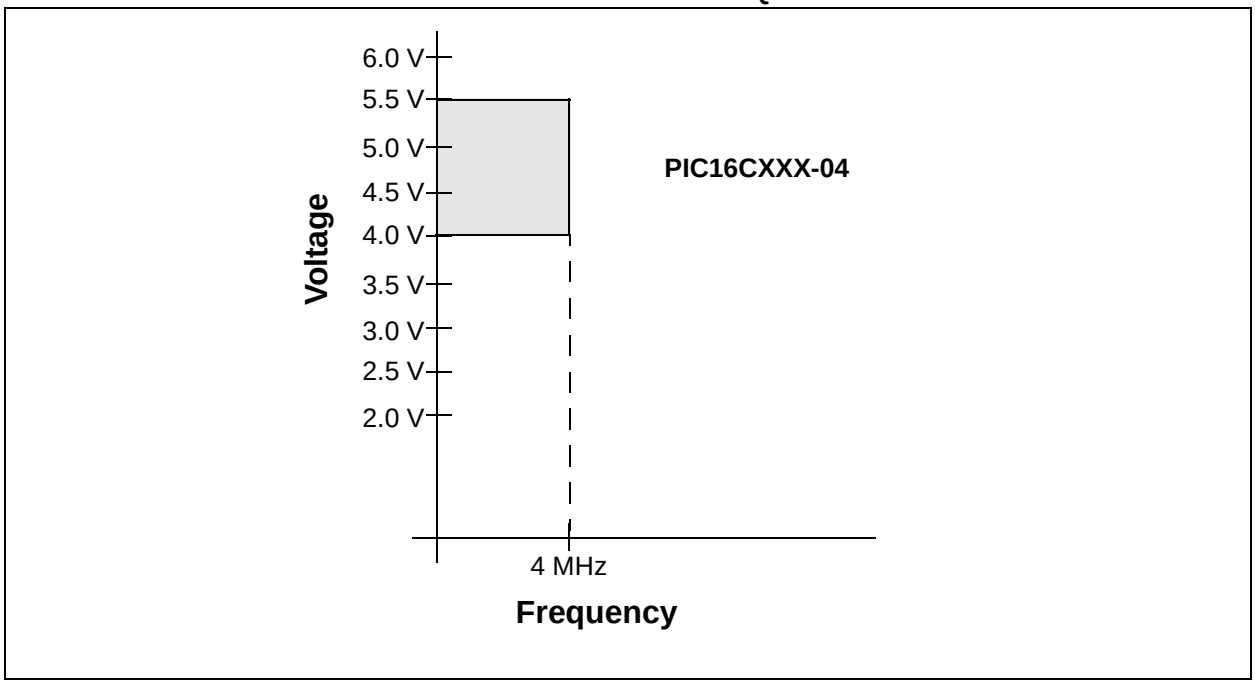
**Note 1:** VDDAPPMIN is the minimum voltage of the PIC® device in the application.

**2:** FMAX has a maximum frequency of 10MHz.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 16-3: PIC16C63A/65B/73B/74B VOLTAGE-FREQUENCY GRAPH**






Downloaded from Arrow.com.

## 16.1 DC Characteristics

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC16LC63A/65B/73B/74B-04** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial | | | | |
| **‡PIC16C63A/65B/73B/74B-04**<br>**‡PIC16C6A/65B/73B/74B-20** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | VDD | **Supply Voltage** | | | | | |
| D001 | | PIC16LCXXX | 2.5<br>VBOR* | –<br>– | 5.5<br>5.5 | V<br>V | LP, XT, RC osc modes (DC - 4 MHz)<br>BOR enabled **(Note 7)** |
| D001<br>D001A | | PIC16CXXX | 4.0<br>4.5<br>VBOR* | –<br>–<br>– | 5.5<br>5.5<br>5.5 | V<br>V<br>V | XT, RC and LP osc mode<br>HS osc mode<br>BOR enabled **(Note 7)** |
| D002* | VDR | **RAM Data Retention Voltage (Note 1)** | – | 1.5 | – | V | |
| D003 | VPOR | **VDD Start Voltage** to ensure internal Power-on Reset signal | – | VSS | – | V | See section on Power-on Reset for details |
| D004*<br>D004A* | SVDD | **VDD Rise Rate** to ensure internal Power-on Reset signal | 0.05<br>TBD | –<br>– | –<br>– | V/mS<br>V/mS | PWRT enabled ($\overline{\text{PWRTE}}$ bit clear)<br>PWRT disabled ($\overline{\text{PWRTE}}$ bit set)<br>See section on Power-on Reset for details |
| D005 | VBOR | **Brown-out Reset** voltage trip point | 3.65 | – | 4.35 | V | BODEN bit set |

* 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.

‡ When specification values of standard devices differ from those of extended voltage devices, they are shown in gray.

**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: $\overline{\text{OSC1}}$ = external square wave, from rail to rail; all I/O pins tri-stated, pulled to VDD, $\overline{\text{MCLR}}$ = 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 mode, 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.

**7:** When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

**8:** In RC oscillator mode, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC device be driven with external clock in RC mode.

**9:** The leakage current on the $\overline{\text{MCLR}}$/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.

**10:** Negative current is defined as current sourced by the pin.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

| PIC16LC63A/65B/73B/74B-04 | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial |
|---|---|
| ‡PIC16C63A/65B/73B/74B-04<br>‡PIC16C6A/65B/73B/74B-20 | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended |

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | IDD | **Supply Current (Notes 2, 5)** | | | | | |
| D010 | | PIC16LCXXX | – | 0.6 | 2.0 | mA | XT, RC osc modes:<br>FOSC = 4 MHz, VDD = 3.0V **(Note 4)** |
| D010A | | | – | 22.5 | 48 | µA | LP osc mode:<br>FOSC = 32 kHz, VDD = 3.0V, WDT disabled |
| D010 | | PIC16CXXX | – | 2.7 | 5 | mA | XT, RC osc modes:<br>FOSC = 4 MHz, VDD = 5.5 V **(Note 4)** |
| D013 | | | – | 7 | 10 | mA | HS osc mode:<br>FOSC = 20 MHz, VDD = 5.5 V |
| | IPD | **Power-down Current (Notes 3, 5)** | | | | | |
| D020 | | PIC16LCXXX | – | 7.5 | 20 | µA | VDD = 3.0V, WDT enabled, -40°C to +85°C |
| D021 | | | – | 0.9 | 3 | µA | VDD = 3.0V, WDT disabled, 0°C to +70°C |
| D021A | | | – | 0.9 | 3 | µA | VDD = 3.0V, WDT disabled, -40°C to +85°C |
| D020 | | PIC16CXXX | – | 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 |

* 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.

‡ When specification values of standard devices differ from those of extended voltage devices, they are shown in gray.

**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 tri-stated, pulled to VDD, MCLR = VDD; WDT enabled/disabled as specified.

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

**4:** For RC osc mode, 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.

**7:** When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

**8:** In RC oscillator mode, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC device be driven with external clock in RC mode.

**9:** The leakage current on the 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.

**10:** Negative current is defined as current sourced by the pin.



Downloaded from Arrow.com.

| PIC16LC63A/65B/73B/74B-04 | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial |
|---|---|
| ‡PIC16C63A/65B/73B/74B-04<br>‡PIC16C6A/65B/73B/74B-20 | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended |

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | | **Module Differential Current (Note 6)** | | | | | |
| D022* | ΔIWDT | Watchdog Timer | – | 6.0 | 20 | μA | WDTE bit set, VDD = 4.0V |
| D022A* | ΔIBOR | Brown-out Reset | – | 100 | 150 | μA | BODEN bit set, VDD = 5.0 |
| | | **Input Low Voltage** | | | | | |
| | VIL | I/O ports | | | | | |
| D030 | | with TTL buffer | VSS | – | 0.15 VDD | V | For entire VDD range |
| D030A | | | VSS | – | 0.8V | V | 4.5V ≤ VDD ≤ 5.5V |
| D031 | | with Schmitt Trigger buffer | VSS | – | 0.2 VDD | V | |
| D032 | | $\overline{MCLR}$, OSC1 (in RC mode) | Vss | – | 0.2 VDD | V | |
| D033 | | OSC1 (in XT, HS, and LP modes) | Vss | – | 0.3 VDD | V | **(Note 8)** |

* 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.

‡ When specification values of standard devices differ from those of extended voltage devices, they are shown in gray.

**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 tri-stated, 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 mode, 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.

**7:** When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

**8:** In RC oscillator mode, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC device be driven with external clock in RC mode.

**9:** 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.

**10:** Negative current is defined as current sourced by the pin.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

| PIC16LC63A/65B/73B/74B-04 | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial |
|---|---|
| ‡PIC16C63A/65B/73B/74B-04<br>‡PIC16C6A/65B/73B/74B-20 | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended |

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | | **Input High Voltage** | | | | | |
| | VIH | I/O ports | | | | | |
| D040 | | with TTL buffer | 2.0 | – | VDD | V | 4.5 V ≤ VDD ≤ 5.5V |
| D040A | | | 0.25 VDD + 0.8V | – | VDD | V | For entire VDD range |
| D041 | | with Schmitt Trigger buffer | 0.8 VDD | – | VDD | V | For entire VDD range |
| D042 | | $\overline{MCLR}$ | 0.8 VDD | – | VDD | V | |
| D042A | | OSC1 (in XT, HS, and LP modes) | 0.7 VDD | – | VDD | V | **(Note 8)** |
| D043 | | OSC1 (in RC mode) | 0.9 VDD | – | VDD | V | |
| | | **Input Leakage Current (Notes 9, 10)** | | | | | |
| D060 | IIL | I/O ports | – | – | ±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 modes |
| D070 | IPURB | **PORTB Weak Pull-up Current** | 50 | 250 | 400 | µA | VDD = 5V, VPIN = VSS |

* 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.

‡ When specification values of standard devices differ from those of extended voltage devices, they are shown in gray.

**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 tri-stated, 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 mode, 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.

**7:** When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

**8:** In RC oscillator mode, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC device be driven with external clock in RC mode.

**9:** 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.

**10:** Negative current is defined as current sourced by the pin.

© 1998-2013 Microchip Technology Inc.


Downloaded from Arrow.com.

| PIC16LC63A/65B/73B/74B-04 | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial |
|---|---|
| ‡PIC16C63A/65B/73B/74B-04<br>‡PIC16C6A/65B/73B/74B-20 | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended |

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | | **Output Low Voltage** | | | | | |
| D080 | VOL | I/O ports | – | – | 0.6 | V | IOL = 8.5 mA, VDD = 4.5V, -40°C to +85°C |
| | | | – | – | 0.6 | V | IOL = 7.0 mA, VDD = 4.5V, -40°C to +125°C |
| D083 | | OSC2/CLKOUT (RC osc mode) | – | – | 0.6 | V | IOL = 1.6 mA, VDD = 4.5V, -40°C to +85°C |
| | | | – | – | 0.6 | V | IOL = 1.2 mA, VDD = 4.5V, -40°C to +125°C |
| | | **Output High Voltage** | | | | | |
| D090 | VOH | I/O ports **(Note 10)** | VDD-0.7 | – | – | V | IOH = -3.0 mA, VDD = 4.5V, -40°C to +85°C |
| | | | VDD-0.7 | – | – | V | IOH = -2.5 mA, VDD = 4.5V, -40°C to +125°C |
| D092 | | OSC2/CLKOUT (RC osc mode) | VDD-0.7 | – | – | V | IOH = -1.3 mA, VDD = 4.5V, -40°C to +85°C |
| | | | VDD-0.7 | – | – | V | IOH = -1.0 mA, VDD = 4.5V, -40°C to +125°C |
| D150* | VOD | **Open-Drain High Voltage** | – | – | 8.5 | 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.

‡ When specification values of standard devices differ from those of extended voltage devices, they are shown in gray.

**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 tri-stated, pulled to VDD, MCLR = VDD; WDT enabled/disabled as specified.

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

**4:** For RC osc mode, 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.

**7:** When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

**8:** In RC oscillator mode, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC device be driven with external clock in RC mode.

**9:** The leakage current on the 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.

**10:** Negative current is defined as current sourced by the pin.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

| PIC16LC63A/65B/73B/74B-04 | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial |
|---|---|
| ‡PIC16C63A/65B/73B/74B-04<br>‡PIC16C6A/65B/73B/74B-20 | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature 0°C ≤ TA ≤ +70°C for commercial<br>-40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended |

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | | **Capacitive Loading Specs on Output Pins** | | | | | |
| D100 | COSC2 | OSC2 pin | – | – | 15 | pF | In XT, HS and LP modes when external clock is used to drive OSC1 |
| D101 | CIO | All I/O pins and OSC2 (in RC mode) | – | – | 50 | pF | |
| D102 | Cb | SCL, SDA (in $I^2C$ mode) | – | – | 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.

‡ When specification values of standard devices differ from those of extended voltage devices, they are shown in gray.

**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 tri-stated, 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 mode, 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.

**7:** When BOR is enabled, the device will operate correctly until the VBOR voltage trip point is reached.

**8:** In RC oscillator mode, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC device be driven with external clock in RC mode.

**9:** 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.

**10:** Negative current is defined as current sourced by the pin.



Downloaded from Arrow.com.

## 16.2 AC (Timing) Characteristics

### 16.2.1 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 | | |


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

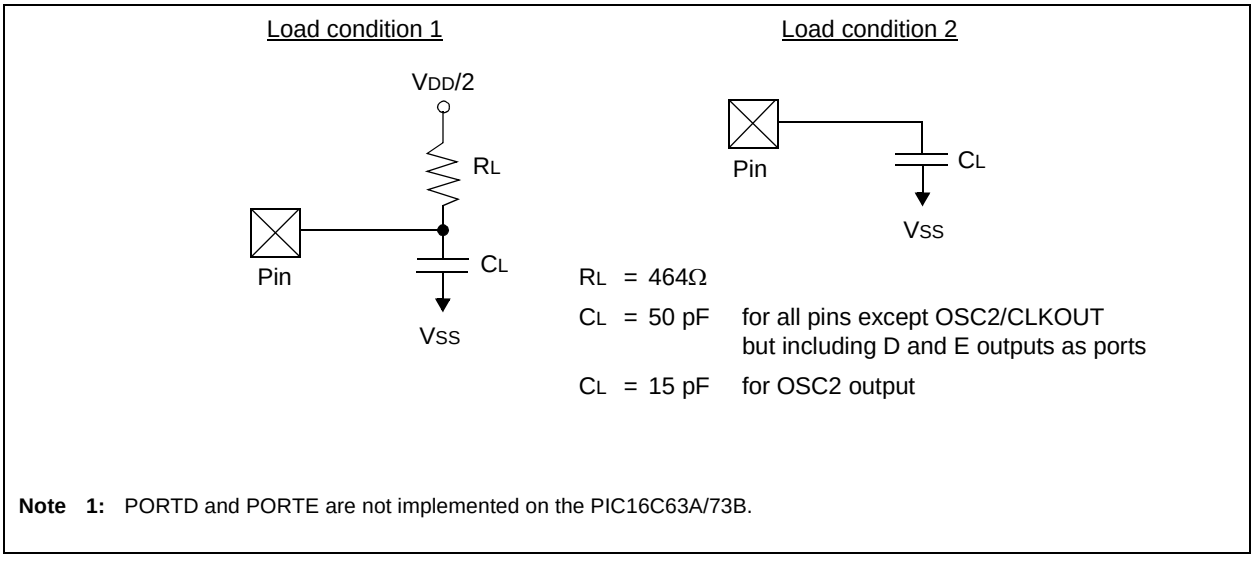
### 16.2.2 TIMING CONDITIONS

The temperature and voltages specified in Table 16-1 apply to all timing specifications unless otherwise noted. Figure 16-4 specifies the load conditions for the timing specifications.

**TABLE 16-1: TEMPERATURE AND VOLTAGE SPECIFICATIONS - AC**

| | | | | | |
|---|---|---|---|---|---|
| AC CHARACTERISTICS | **Standard Operating Conditions (unless otherwise stated)** | | | | |
| | Operating temperature | 0°C | ≤ TA ≤ | +70°C | for commercial |
| | | -40°C | ≤ TA ≤ | +85°C | for industrial |
| | | -40°C | ≤ TA ≤ | +125°C | for extended |
| | Operating voltage VDD range as described in DC spec Section 16.1. | | | | |
| | LC parts operate for commercial/industrial temperatures only. | | | | |

**FIGURE 16-4: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS**




**Note 1:** PORTD and PORTE are not implemented on the PIC16C63A/73B.



Downloaded from Arrow.com.

16.2.3 TIMING DIAGRAMS AND SPECIFICATIONS

**FIGURE 16-5: EXTERNAL CLOCK TIMING**

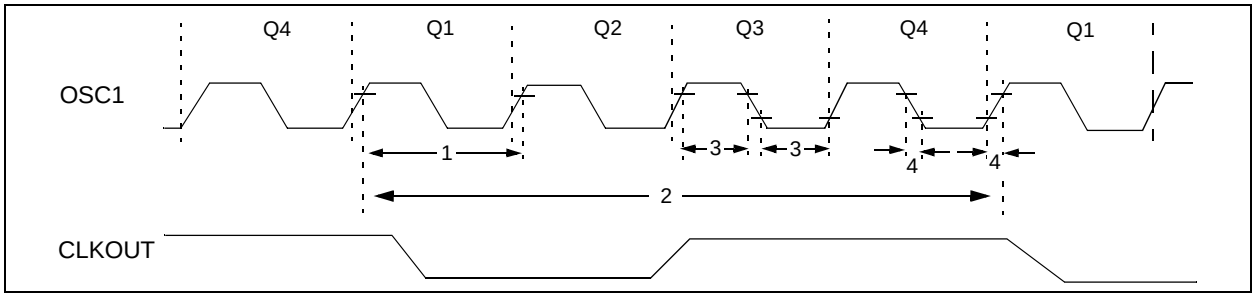



**TABLE 16-2: EXTERNAL CLOCK TIMING REQUIREMENTS**

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 1A | FOSC | External CLKIN Frequency **(Note 1)** | DC | — | 4 | MHz | RC and XT osc modes |
| | | | DC | — | 4 | MHz | HS osc mode (-04) |
| | | | 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 | RC and XT osc modes |
| | | | 250 | — | — | ns | HS osc mode (-04) |
| | | | 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) |
| | | | 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 |

* These parameters are characterized but not tested.

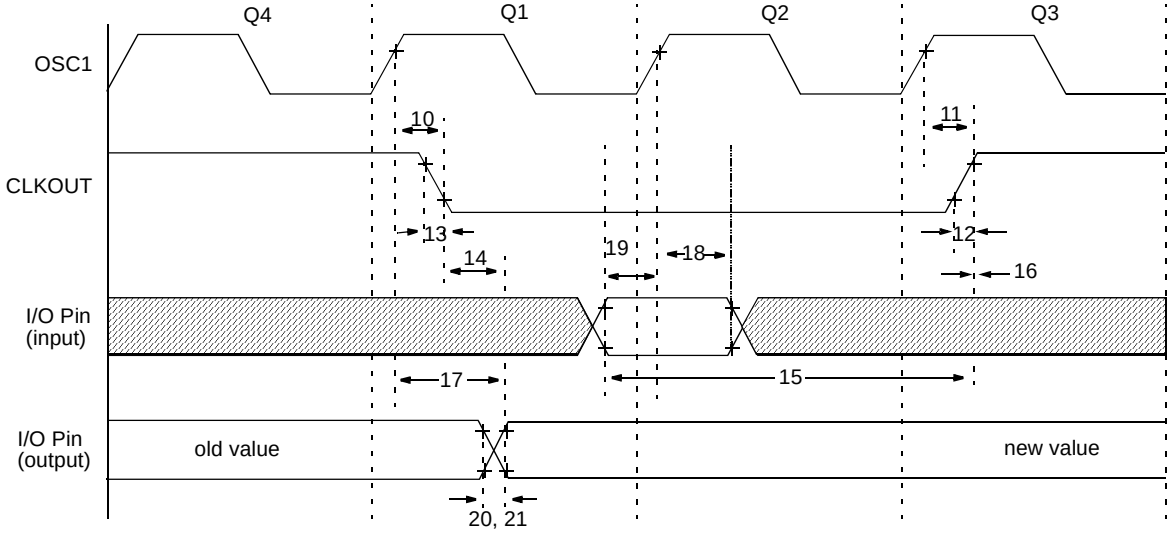
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** 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.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 16-6: CLKOUT AND I/O TIMING**




**Note:** Refer to Figure 16-4 for load conditions.

**TABLE 16-3: 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) | PIC16CXX | 100 | — | — | ns | |
| 18A* | | | PIC16LCXX | 200 | — | — | ns | |
| 19* | TioV2osH | Port input valid to OSC1↑ (I/O in setup time) | | 0 | — | — | ns | |
| 20* | TioR | Port output rise time | PIC16CXX | — | 10 | 40 | ns | |
| 20A* | | | PIC16LCXX | — | — | 80 | ns | |
| 21* | TioF | Port output fall time | PIC16CXX | — | 10 | 40 | ns | |
| 21A* | | | PIC16LCXX | — | — | 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.

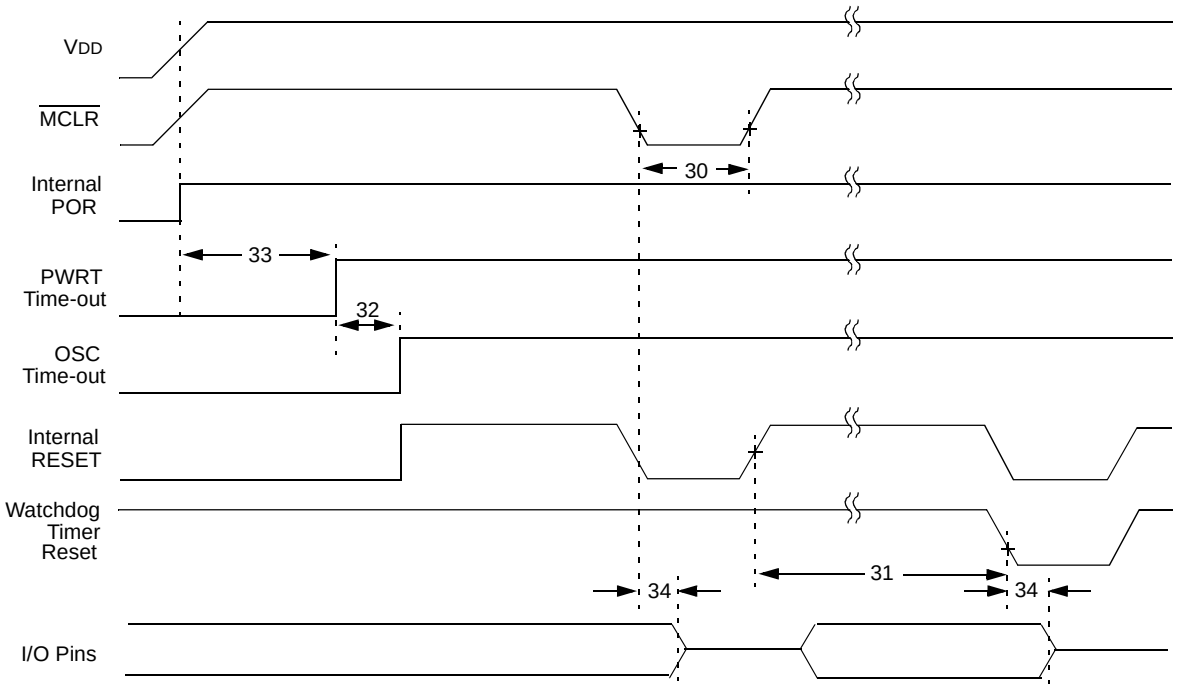
†† These parameters are asynchronous events not related to any internal clock edge.

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



Downloaded from Arrow.com.

**FIGURE 16-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**




**Note:** Refer to Figure 16-4 for load conditions.

**FIGURE 16-8: BROWN-OUT RESET TIMING**

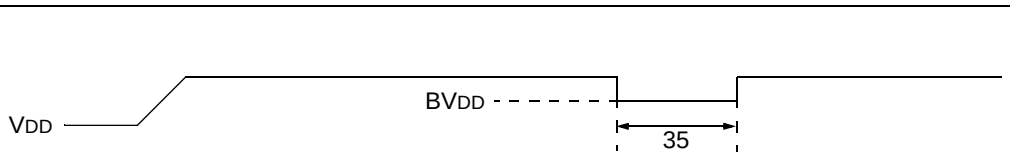



**TABLE 16-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS**

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 30 | TmcL | $\overline{\text{MCLR}}$ 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 | — | 1024 TOSC | — | — | 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 $\overline{\text{MCLR}}$ Low or WDT Reset | — | — | 2.1 | µs | |
| 35 | TBOR | Brown-out Reset Pulse Width | 100 | — | — | µs | VDD ≤ BVDD (D005) |

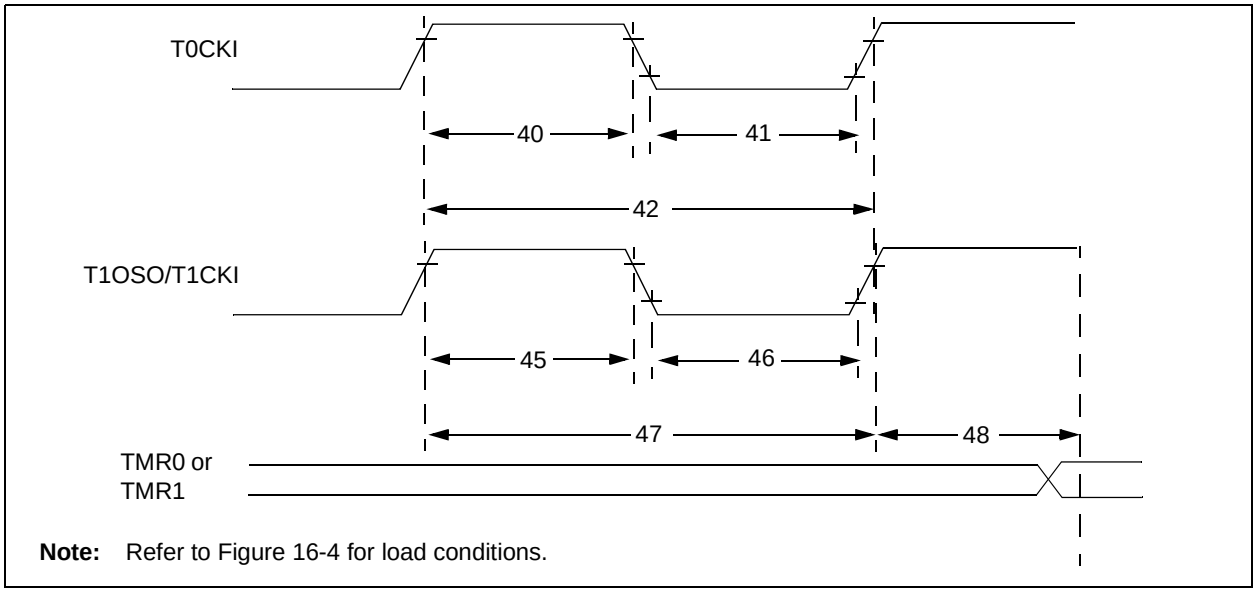
* These parameters are characterized but not tested.

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


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 16-9: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**




**Note:** Refer to Figure 16-4 for load conditions.

**TABLE 16-5: 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 | PIC16CXX | 15 | — | — | ns | |
| | | | | PIC16LCXX | 25 | — | — | ns | |
| | | | Asynchronous | PIC16CXX | 30 | — | — | ns | |
| | | | | PIC16LCXX | 50 | — | — | ns | |
| 46* | Tt1L | T1CKI Low Time | Synchronous, Prescaler = 1 | | 0.5TCY + 20 | — | — | ns | Must also meet parameter 47 |
| | | | Synchronous, Prescaler = 2,4,8 | PIC16CXX | 15 | — | — | ns | |
| | | | | PIC16LCXX | 25 | — | — | ns | |
| | | | Asynchronous | PIC16CXX | 30 | — | — | ns | |
| | | | | PIC16LCXX | 50 | — | — | ns | |
| 47* | Tt1P | T1CKI input period | Synchronous | PIC16CXX | Greater of: 30 or TCY + 40 / N | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | | PIC16LCXX | Greater of: 50 or TCY + 40 / N | | | | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | PIC16CXX | 60 | — | — | ns | |
| | | | | PIC16LCXX | 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 | — | |

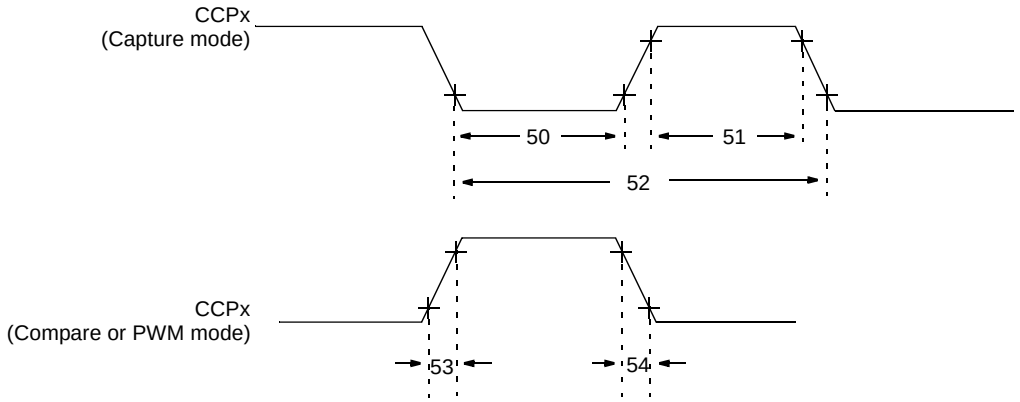
* These parameters are characterized but not tested.

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



Downloaded from Arrow.com.

**FIGURE 16-10: CAPTURE/COMPARE/PWM TIMINGS (CCP1 AND CCP2)**




**Note:** Refer to Figure 16-4 for load conditions.

**TABLE 16-6: 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 | PIC16CXX | 10 | — | — | ns | |
| | | | | PIC16LCXX | 20 | — | — | ns | |
| 51* | TccH | CCP1 and CCP2 input high time | No Prescaler | | 0.5TCY + 20 | — | — | ns | |
| | | | With Prescaler | PIC16CXX | 10 | — | — | ns | |
| | | | | PIC16LCXX | 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 | | PIC16CXX | — | 10 | 25 | ns | |
| | | | | PIC16LCXX | — | 25 | 45 | ns | |
| 54* | TccF | CCP1 and CCP2 output fall time | | PIC16CXX | — | 10 | 25 | ns | |
| | | | | PIC16LCXX | — | 25 | 45 | ns | |

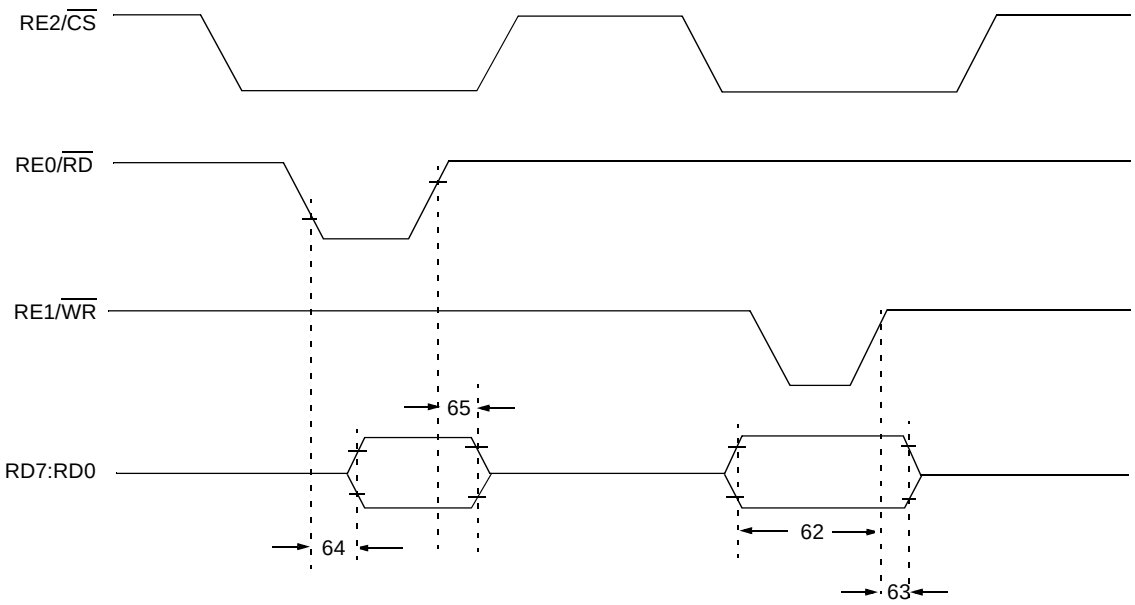
* These parameters are characterized but not tested.

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


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 16-11: PARALLEL SLAVE PORT TIMING (PIC16C65B/74B)**




**Note:** Refer to Figure 16-4 for load conditions.

**TABLE 16-7: PARALLEL SLAVE PORT REQUIREMENTS (PIC16C65B/74B)**

| Param 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) | PIC16CXX | 20 | — | — | ns | |
| | | | PIC16LCXX | 35 | — | — | ns | |
| 64 | TrdL2dtV | $\overline{RD}\downarrow$ and $\overline{CS}\downarrow$ to data out valid | | — | — | 80 | ns | |
| 65* | TrdH2dtI | $\overline{RD}\uparrow$ or $\overline{CS}\uparrow$ to data out invalid | | 10 | — | 30 | ns | |

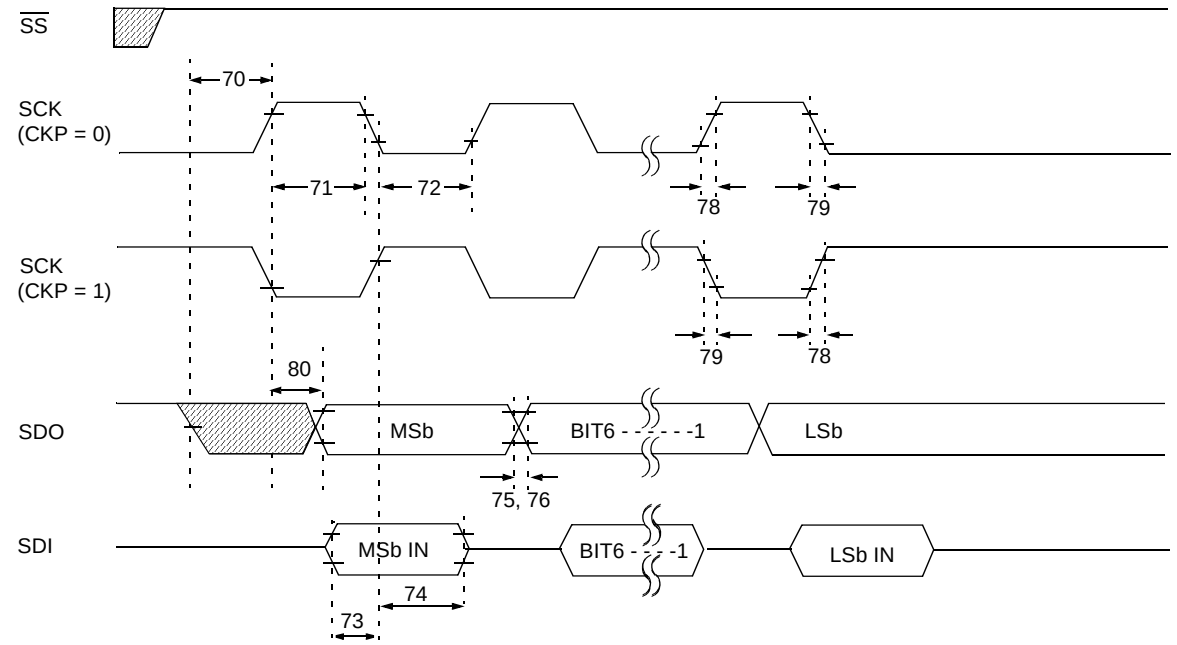
* These parameters are characterized but not tested.

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



Downloaded from Arrow.com.

**FIGURE 16-12: EXAMPLE SPI MASTER MODE TIMING (CKE = 0)**




**Note:** Refer to Figure 16-4 for load conditions.

**TABLE 16-8: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 0)**

| Param No. | Symbol | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 70 | TssL2scH, TssL2scL | $\overline{SS}\downarrow$ to SCK↓ or SCK↑ input | | TCY | — | — | ns | |
| 71 | TscH | SCK input high time (Slave mode) | Continuous | 1.25TCY + 30 | — | — | ns | |
| 71A | | | Single Byte | 40 | — | — | ns | **(Note 1)** |
| 72 | TscL | SCK input low time (Slave mode) | Continuous | 1.25TCY + 30 | — | — | ns | |
| 72A | | | Single Byte | 40 | — | — | ns | **(Note 1)** |
| 73 | TdiV2scH, TdiV2scL | Setup time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 73A | TB2B | Last clock edge of Byte1 to the 1st clock edge of Byte2 | | 1.5TCY + 40 | — | — | ns | **(Note 1)** |
| 74 | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 75 | TdoR | SDO data output rise time | PIC16CXX | — | 10 | 25 | ns | |
| | | | PIC16LCXX | — | 20 | 45 | ns | |
| 76 | TdoF | SDO data output fall time | | — | 10 | 25 | ns | |
| 78 | TscR | SCK output rise time (Master mode) | PIC16CXX | — | 10 | 25 | ns | |
| | | | PIC16LCXX | — | 20 | 45 | ns | |
| 79 | TscF | SCK output fall time (Master mode) | | — | 10 | 25 | ns | |
| 80 | TscH2doV, TscL2doV | SDO data output valid after SCK edge | PIC16CXX | — | — | 50 | ns | |
| | | | PIC16LCXX | — | — | 100 | ns | |

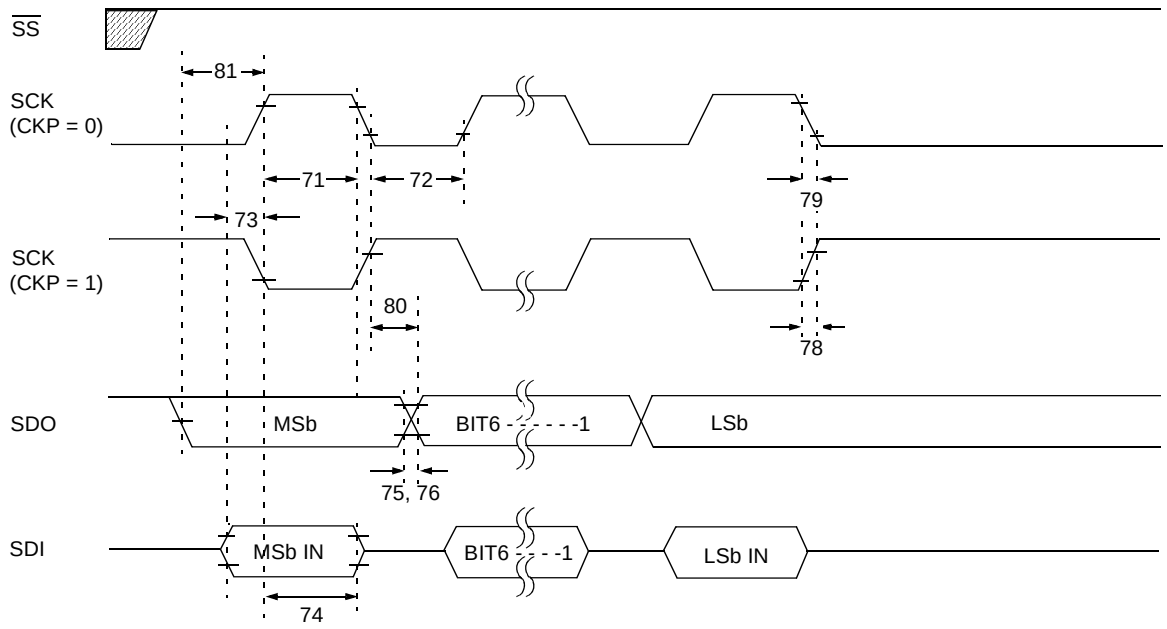
† Data in "Typ" column is at 5 V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Specification 73A is only required if specifications 71A and 72A are used.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 16-13: EXAMPLE SPI MASTER MODE TIMING (CKE = 1)**




**Note:** Refer to Figure 16-4 for load conditions.

**TABLE 16-9: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 1)**

| Param No. | Symbol | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 71 | TscH | SCK input high time (Slave mode) | Continuous | 1.25TCY + 30 | — | — | ns | |
| 71A | | | Single Byte | 40 | — | — | ns | **(Note 1)** |
| 72 | TscL | SCK input low time (Slave mode) | Continuous | 1.25TCY + 30 | — | — | ns | |
| 72A | | | Single Byte | 40 | — | — | ns | **(Note 1)** |
| 73 | TdiV2scH, TdiV2scL | Setup time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 73A | TB2B | Last clock edge of Byte1 to the 1st clock edge of Byte2 | | 1.5TCY + 40 | — | — | ns | **(Note 1)** |
| 74 | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 75 | TdoR | SDO data output rise time | PIC16CXX | — | 10 | 25 | ns | |
| | | | PIC16LCXX | | 20 | 45 | ns | |
| 76 | TdoF | SDO data output fall time | | — | 10 | 25 | ns | |
| 78 | TscR | SCK output rise time (Master mode) | PIC16CXX | — | 10 | 25 | ns | |
| | | | PIC16LCXX | | 20 | 45 | ns | |
| 79 | TscF | SCK output fall time (Master mode) | | — | 10 | 25 | ns | |
| 80 | TscH2doV, TscL2doV | SDO data output valid after SCK edge | PIC16CXX | — | — | 50 | ns | |
| | | | PIC16LCXX | | — | 100 | ns | |
| 81 | TdoV2scH, TdoV2scL | SDO data output setup to SCK edge | | TCY | — | — | ns | |

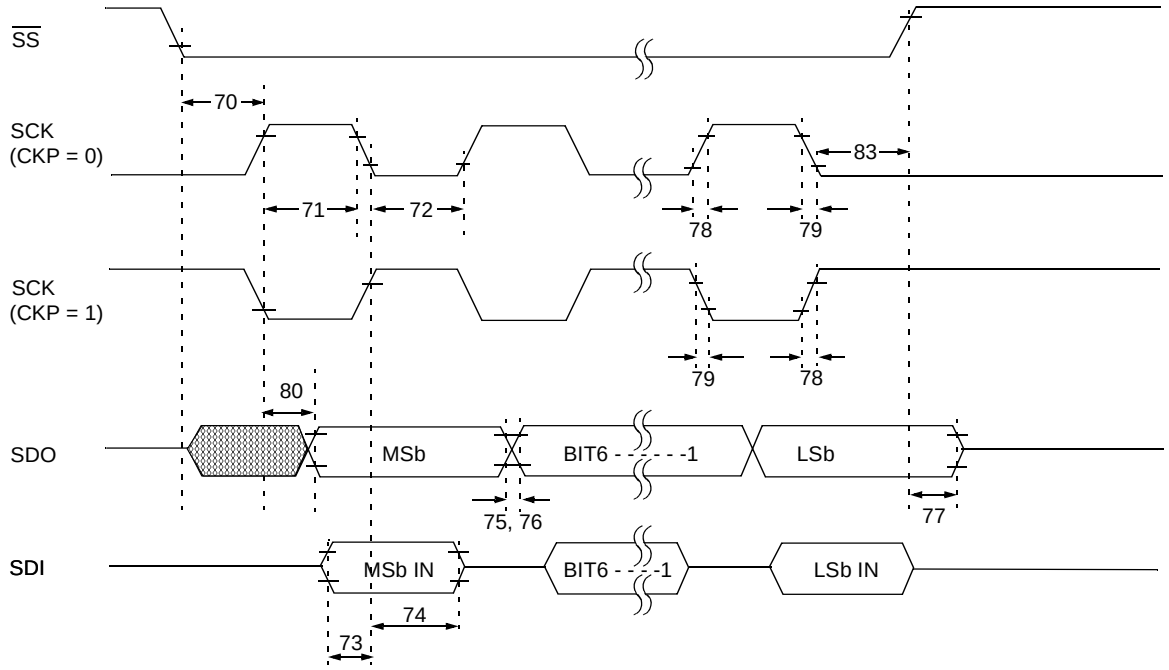
† Data in "Typ" column is at 5 V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Specification 73A is only required if specifications 71A and 72A are used.



Downloaded from Arrow.com.

**FIGURE 16-14: EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)**




**Note:** Refer to Figure 16-4 for load conditions.

**TABLE 16-10: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING (CKE = 0)**

| Param No. | Symbol | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 70 | TssL2scH, TssL2scL | $\overline{SS}$↓ to SCK↓ or SCK↑ input | | TCY | — | — | ns | |
| 71 | TscH | SCK input high time (Slave mode) | Continuous | 1.25TCY + 30 | — | — | ns | |
| 71A | | | Single Byte | 40 | — | — | ns | **(Note 1)** |
| 72 | TscL | SCK input low time (Slave mode) | Continuous | 1.25TCY + 30 | — | — | ns | |
| 72A | | | Single Byte | 40 | — | — | ns | **(Note 1)** |
| 73 | TdiV2scH, TdiV2scL | Setup time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 73A | TB2B | Last clock edge of Byte1 to the 1st clock edge of Byte2 | | 1.5TCY + 40 | — | — | ns | **(Note 1)** |
| 74 | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 75 | TdoR | SDO data output rise time | PIC16CXX | — | 10 | 25 | ns | |
| | | | PIC16LCXX | | 20 | 45 | ns | |
| 76 | TdoF | SDO data output fall time | | — | 10 | 25 | ns | |
| 77 | TssH2doZ | $\overline{SS}$↑ to SDO output hi-impedance | | 10 | — | 50 | ns | |
| 78 | TscR | SCK output rise time (Master mode) | PIC16CXX | — | 10 | 25 | ns | |
| | | | PIC16LCXX | | 20 | 45 | ns | |
| 79 | TscF | SCK output fall time (Master mode) | | — | 10 | 25 | ns | |
| 80 | TscH2doV, TscL2doV | SDO data output valid after SCK edge | PIC16CXX | — | — | 50 | ns | |
| | | | PIC16LCXX | | — | 100 | ns | |
| 83 | TscH2ssH, TscL2ssH | $\overline{SS}$ ↑ after SCK edge | | 1.5TCY + 40 | — | — | ns | |

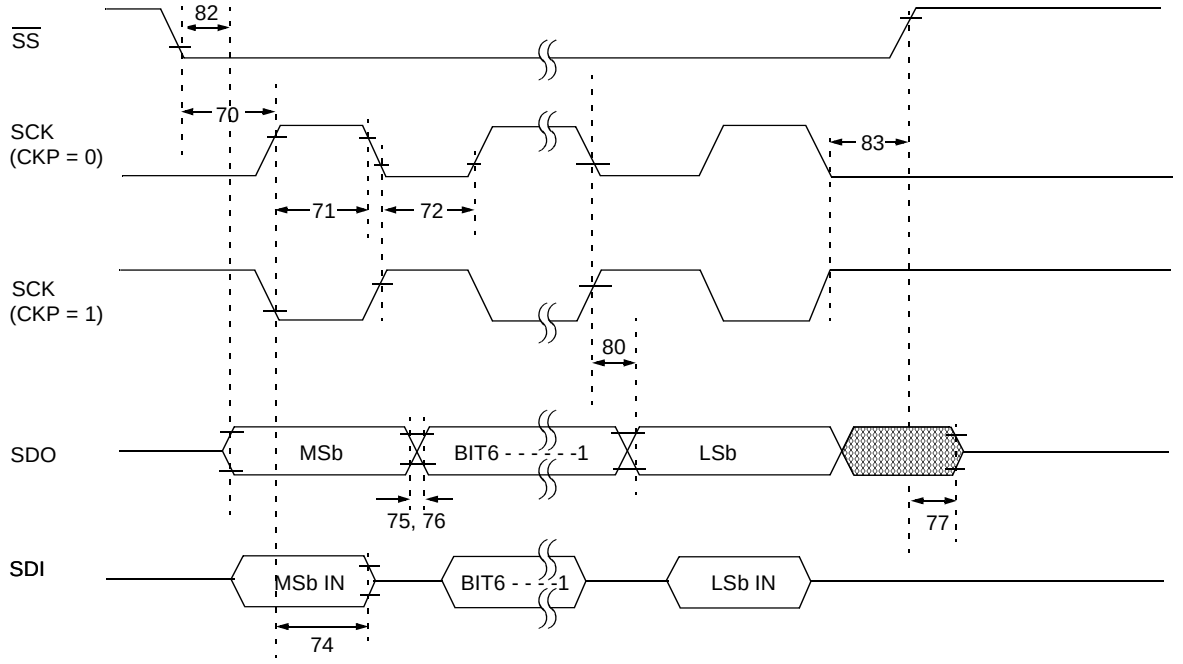
† Data in "Typ" column is at 5 V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Specification 73A is only required if specifications 71A and 72A are used.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 16-15: EXAMPLE SPI SLAVE MODE TIMING (CKE = 1)**




**Note:** Refer to Figure 16-4 for load conditions.

**TABLE 16-11: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)**

| Param No. | Symbol | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 70 | TssL2scH, TssL2scL | $\overline{SS}$↓ to SCK↓ or SCK↑ input | | TCY | — | — | ns | |
| 71 | TscH | SCK input high time (Slave mode) | Continuous | 1.25TCY + 30 | — | — | ns | |
| 71A | | | Single Byte | 40 | — | — | ns | **(Note 1)** |
| 72 | TscL | SCK input low time (Slave mode) | Continuous | 1.25TCY + 30 | — | — | ns | |
| 72A | | | Single Byte | 40 | — | — | ns | **(Note 1)** |
| 73A | TB2B | Last clock edge of Byte1 to the 1st clock edge of Byte2 | | 1.5TCY + 40 | — | — | ns | **(Note 1)** |
| 74 | TscH2diL, TscL2diL | Hold time of SDI data input to SCK edge | | 100 | — | — | ns | |
| 75 | TdoR | SDO data output rise time | PIC16CXX | — | 10 | 25 | ns | |
| | | | PIC16LCXX | | 20 | 45 | ns | |
| 76 | TdoF | SDO data output fall time | | — | 10 | 25 | ns | |
| 77 | TssH2doZ | $\overline{SS}$↑ to SDO output hi-impedance | | 10 | — | 50 | ns | |
| 78 | TscR | SCK output rise time (Master mode) | PIC16CXX | — | 10 | 25 | ns | |
| | | | PIC16LCXX | — | 20 | 45 | ns | |
| 79 | TscF | SCK output fall time (Master mode) | | — | 10 | 25 | ns | |
| 80 | TscH2doV, TscL2doV | SDO data output valid after SCK edge | PIC16CXX | — | — | 50 | ns | |
| | | | PIC16LCXX | — | — | 100 | ns | |
| 82 | TssL2doV | SDO data output valid after $\overline{SS}$↓ edge | PIC16CXX | — | — | 50 | ns | |
| | | | PIC16LCXX | — | — | 100 | ns | |
| 83 | TscH2ssH, TscL2ssH | $\overline{SS}$ ↑ after SCK edge | | 1.5TCY + 40 | — | — | ns | |

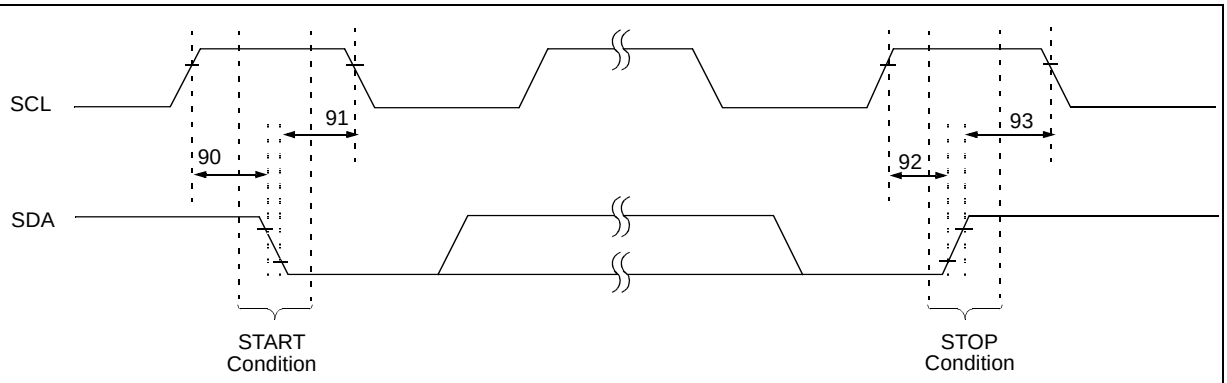
† Data in "Typ" column is at 5 V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Specification 73A is only required if specifications 71A and 72A are used.



Downloaded from Arrow.com.

**FIGURE 16-16: I²C BUS START/STOP BITS TIMING**



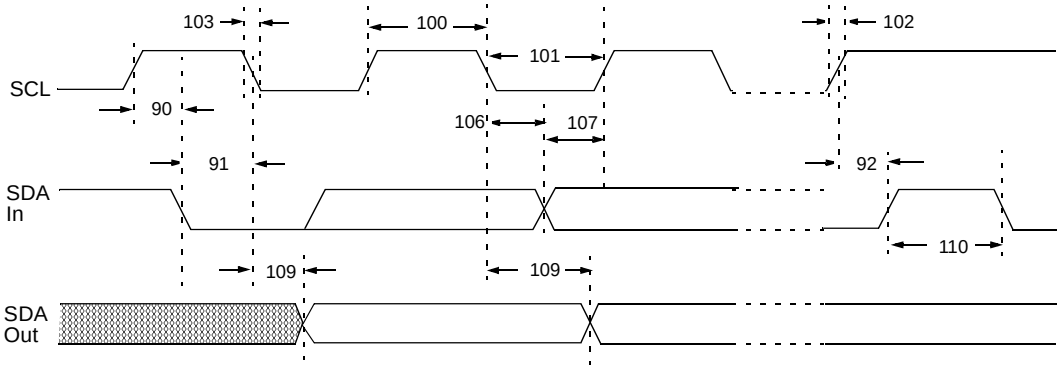

**Note:** Refer to Figure 16-4 for load conditions.

**TABLE 16-12: I²C BUS START/STOP BITS REQUIREMENTS**

| Param 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 | — | — | | |

* These parameters are characterized but not tested.

**FIGURE 16-17: I²C BUS DATA TIMING**




**Note:** Refer to Figure 16-4 for load conditions.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 16-13: I²C BUS DATA REQUIREMENTS**

| Param. 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 10-400 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 10-400 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 | |

* These parameters are characterized but not tested.

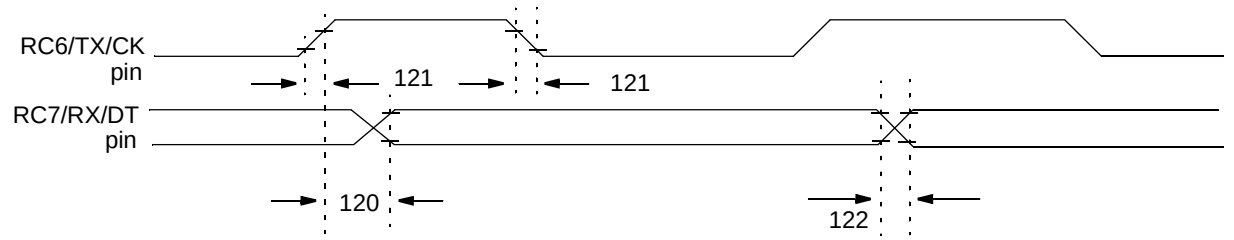
**Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

**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.



Downloaded from Arrow.com.

**FIGURE 16-18: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**




**Note:** Refer to Figure 16-4 for load conditions.

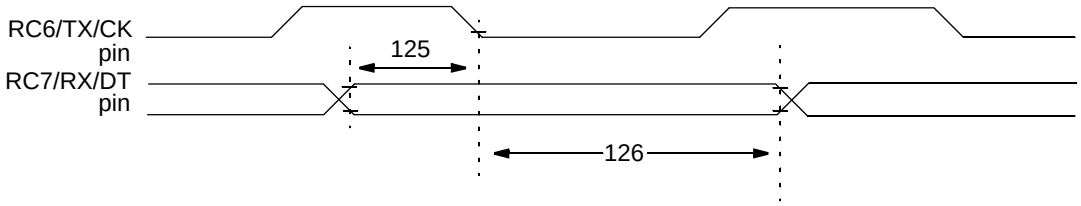
**TABLE 16-14: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 120* | TckH2dtV | SYNC XMIT (MASTER & SLAVE) Clock high to data out valid | PIC16CXX | — | — | 80 | ns | |
| | | | PIC16LCXX | — | — | 100 | ns | |
| 121* | Tckrf | Clock out rise time and fall time (Master mode) | PIC16CXX | — | — | 45 | ns | |
| | | | PIC16LCXX | — | — | 50 | ns | |
| 122* | Tdtrf | Data out rise time and fall time | PIC16CXX | — | — | 45 | ns | |
| | | | PIC16LCXX | — | — | 50 | 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.

**FIGURE 16-19: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**




**Note:** Refer to Figure 16-4 for load conditions.

**TABLE 16-15: USART SYNCHRONOUS RECEIVE REQUIREMENTS**

| Param 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 | |

* 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.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE 16-16: A/D CONVERTER CHARACTERISTICS: PIC16C73B/74B-04 (COMMERCIAL, INDUSTRIAL, EXTENDED) PIC16C73B/74B-20 (COMMERCIAL, INDUSTRIAL, EXTENDED) PIC16LC73B/74B-04 (COMMERCIAL, INDUSTRIAL)**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| A01 | NR | Resolution | PIC16CXX | — | — | 8 bits | bit | VREF = VDD = 5.12 V, VSS ≤ VAIN ≤ VREF |
| | | | PIC16LCXX | — | — | 8 bits | bit | VREF = VDD = 2.5 V |
| A02 | EABS | Total Absolute error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12 V, VSS ≤ VAIN ≤ VREF |
| A03 | EIL | Integral linearity error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12 V, VSS ≤ VAIN ≤ VREF |
| A04 | EDL | Differential linearity error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12 V, VSS ≤ VAIN ≤ VREF |
| A05 | EFS | Full scale error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12 V, VSS ≤ VAIN ≤ VREF |
| A06 | EOFF | Offset error | | — | — | < ± 1 | LSb | VREF = VDD = 5.12 V, VSS ≤ VAIN ≤ VREF |
| A10 | — | Monotonicity **(Note 3)** | | — | guaranteed | — | — | VSS ≤ VAIN ≤ VREF |
| A20 | VREF | Reference voltage | | 2.5V | — | 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) | PIC16CXX | — | 180 | — | μA | Average current consumption when A/D is on **(Note 1)** |
| | | | PIC16LCXX | — | 90 | — | μA | |
| A50 | IREF | VREF input current **(Note 2)** | | 10<br><br><br><br><br><br>— | — | 1000<br><br><br><br><br><br>10 | μA<br><br><br><br><br><br>μA | During VAIN acquisition Based on differential of VHOLD to VAIN to charge CHOLD, see Section 12.1 During A/D Conversion cycle |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5 V, 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.

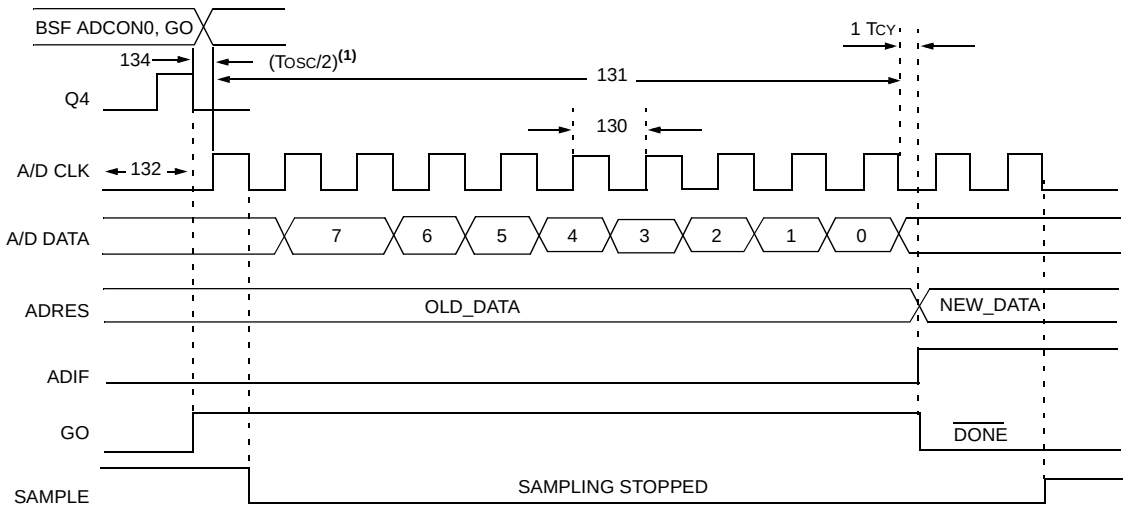
**2:** VREF current is from the RA3 pin or the VDD pin, whichever is selected as a reference input.

**3:** The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.



Downloaded from Arrow.com.

**FIGURE 16-20: 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 16-17: A/D CONVERSION REQUIREMENTS**

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|---|
| 130 | TAD | A/D clock period | PIC16CXX | 1.6 | — | — | µs | TOSC based, VREF ≥ 3.0 V |
| | | | PIC16LCXX | 2.0 | — | — | µs | TOSC based, 2.5V ≤ VREF ≤ 5.5 V |
| | | | PIC16CXX | 2.0 | 4.0 | 6.0 | µs | A/D RC mode |
| | | | PIC16LCXX | 3.0 | 6.0 | 9.0 | µs | A/D RC mode |
| 131 | TCNV | Conversion time (not including S/H time) **(Note 1)** | | 11 | — | 11 | TAD | |
| 132 | TACQ | Acquisition time | | 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., 20.0mV @ 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.

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

**2:** See Section 12.1 for minimum conditions.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 17.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

The graphs and tables provided in this section are for design guidance and are not tested nor guaranteed. In some graphs or tables the data presented is outside specified operating range (e.g., outside specified VDD range). This is for information only and devices are ensured to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time.

**Note:** Standard deviation is denoted by sigma (σ).

- **Typ** or **Typical** represents the mean of the distribution at 25°C.
- **Max** or **Maximum** represents the mean + 3σ over the temperature range of -40°C to 85°C.
- **Min** or **Minimum** represents the mean - 3σ over the temperature range of -40°C to 85°C.



Downloaded from Arrow.com.

**FIGURE 17-1: TYPICAL IDD vs. FOSC OVER VDD – HS MODE**

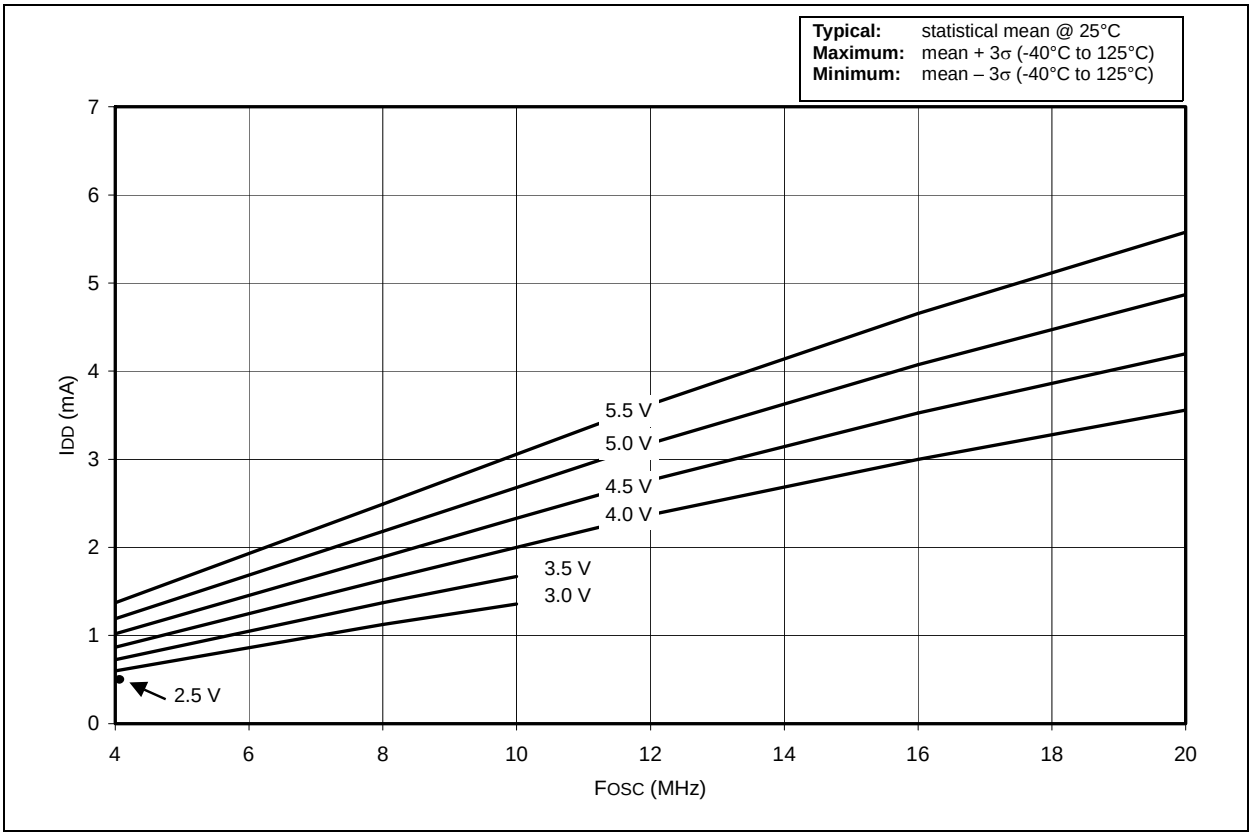
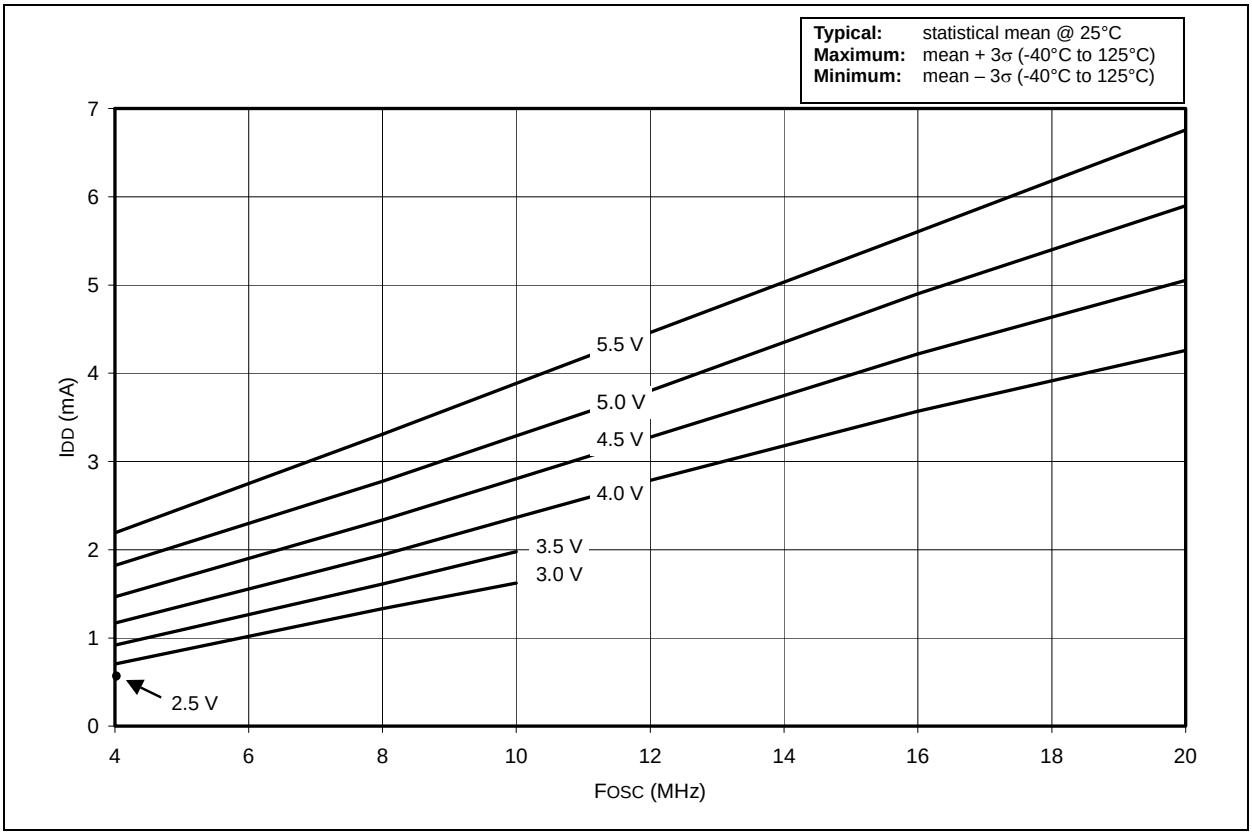



**FIGURE 17-2: MAXIMUM IDD vs. FOSC OVER VDD – HS MODE**





© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 17-3: TYPICAL IDD vs. FOSC OVER VDD – LP MODE**

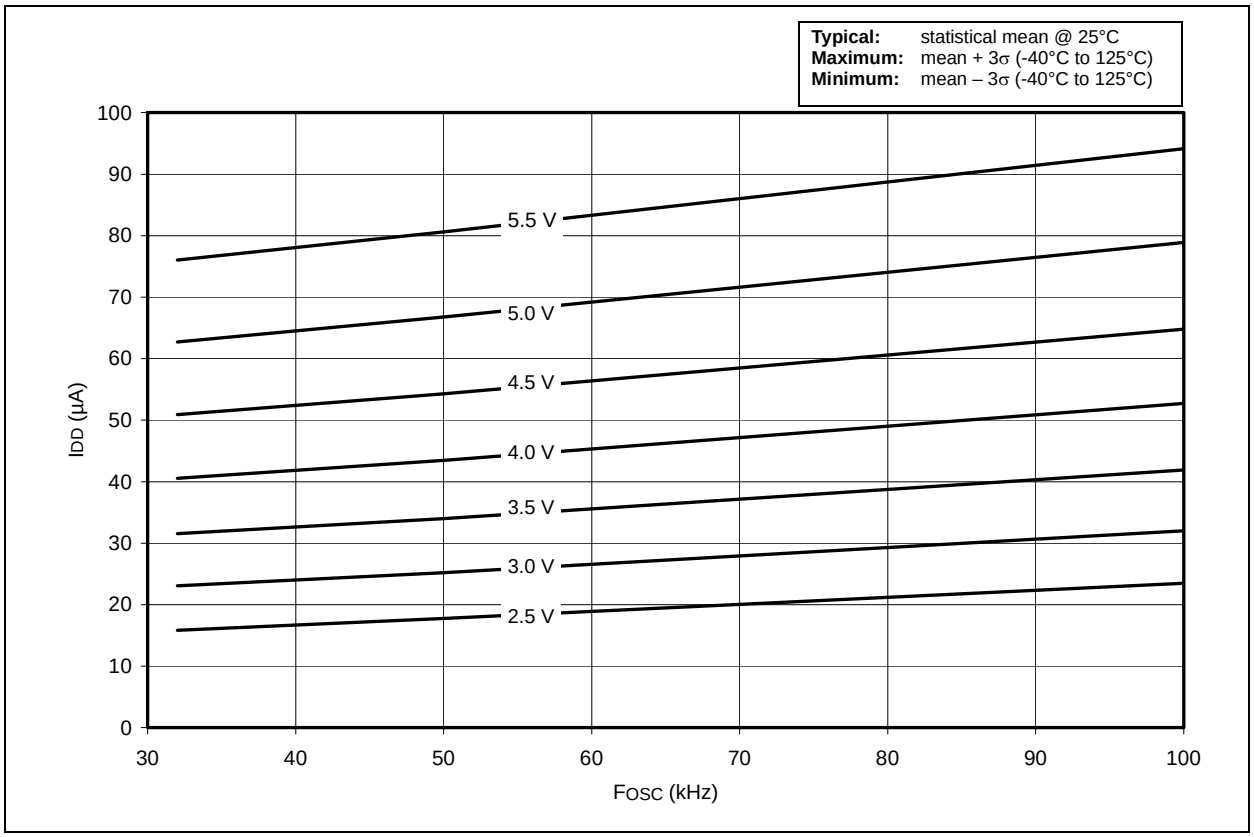
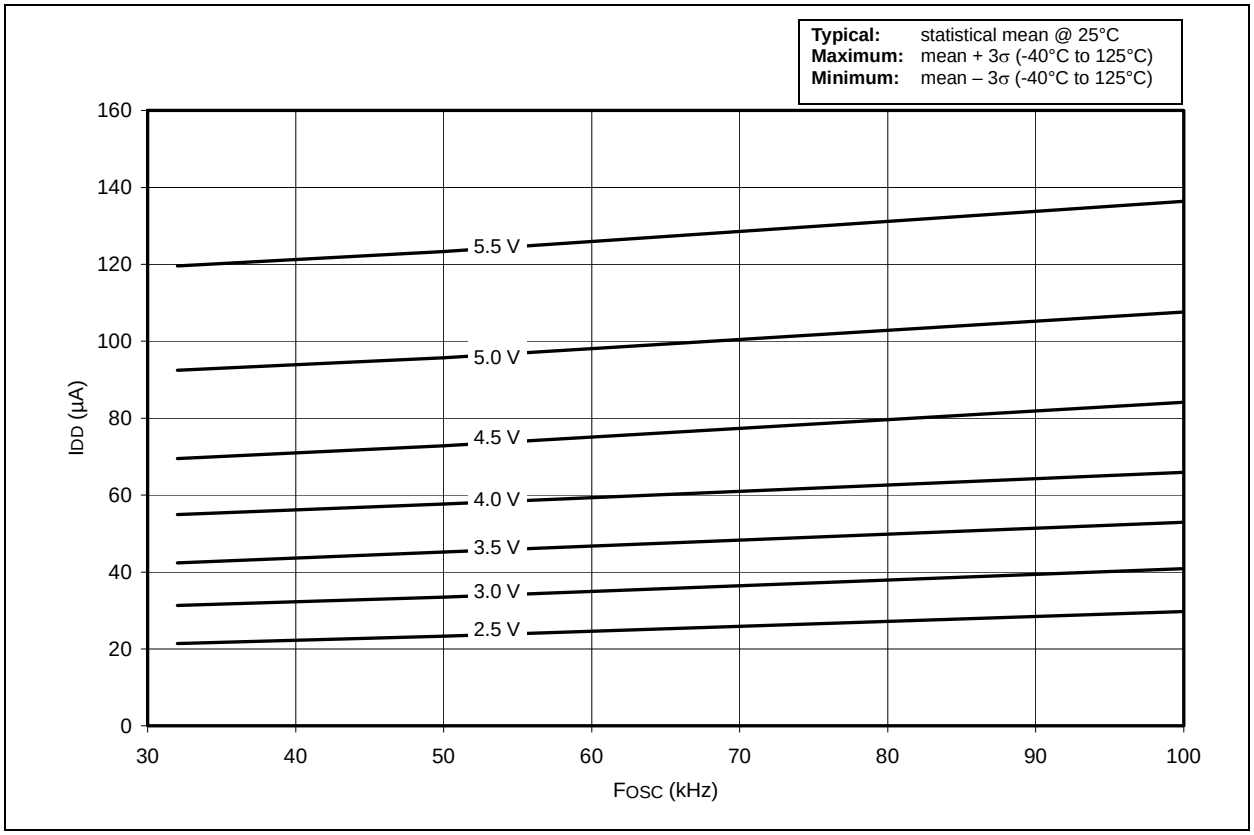



**FIGURE 17-4: MAXIMUM IDD vs. FOSC OVER VDD – LP MODE**






Downloaded from Arrow.com.

**FIGURE 17-5: TYPICAL IDD vs. FOSC OVER VDD – XT MODE**

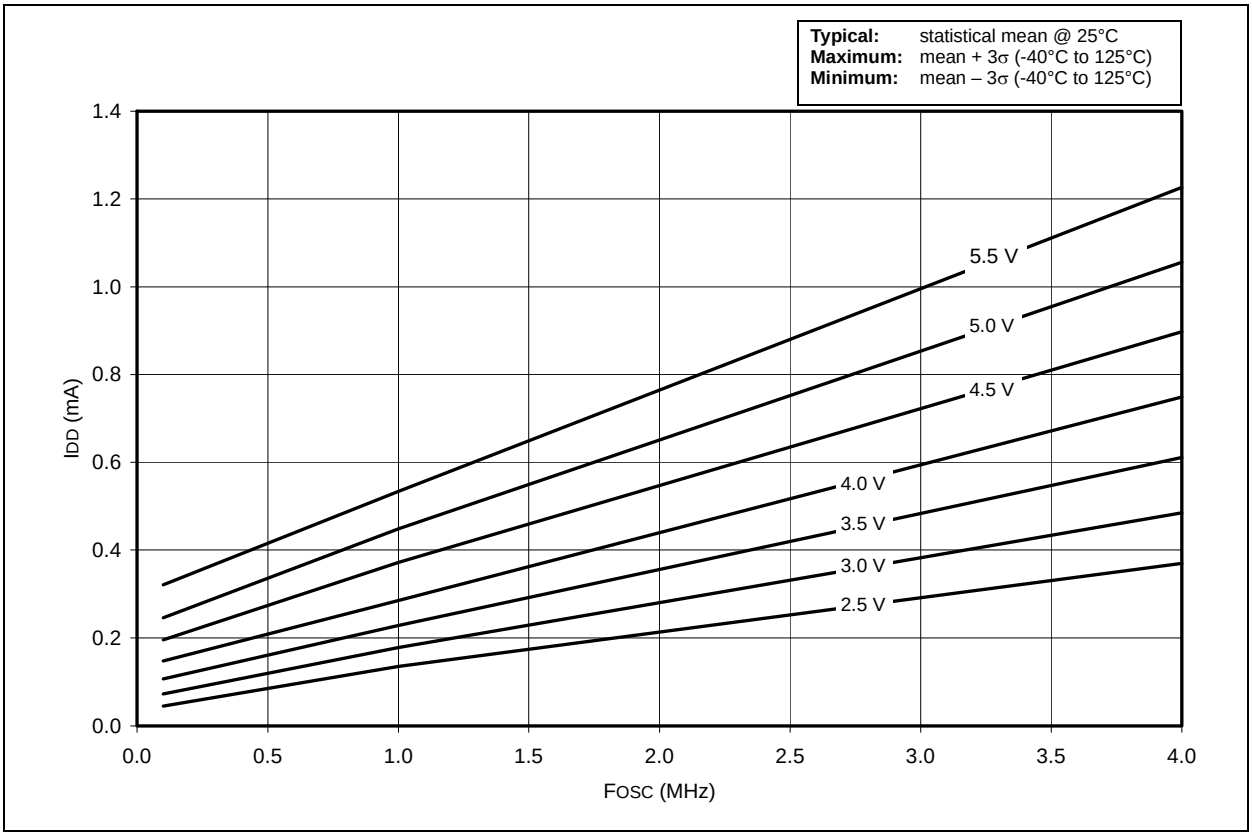
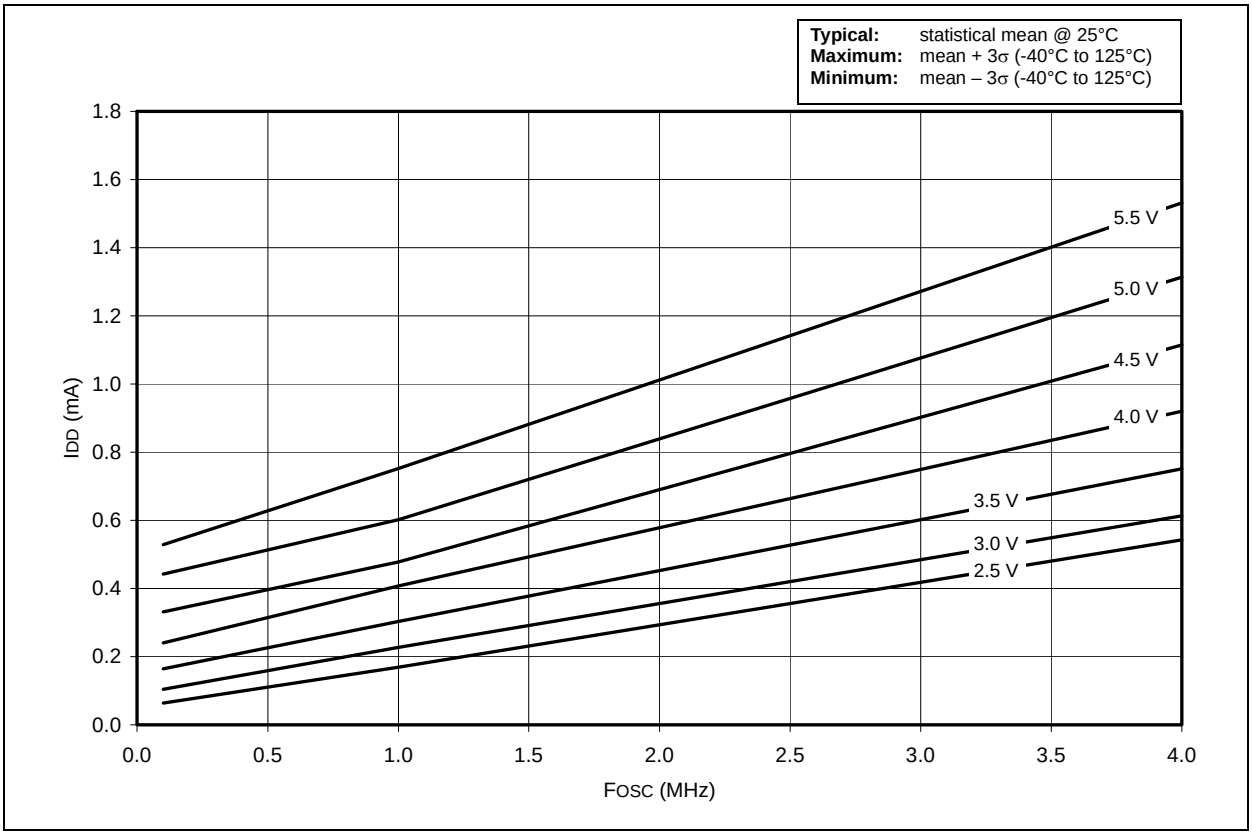



**FIGURE 17-6: MAXIMUM IDD vs. FOSC OVER VDD – XT MODE**





© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 17-7: AVERAGE FOSC VS. VDD FOR VARIOUS RESISTANCES – RC MODE; C = 20 PF**

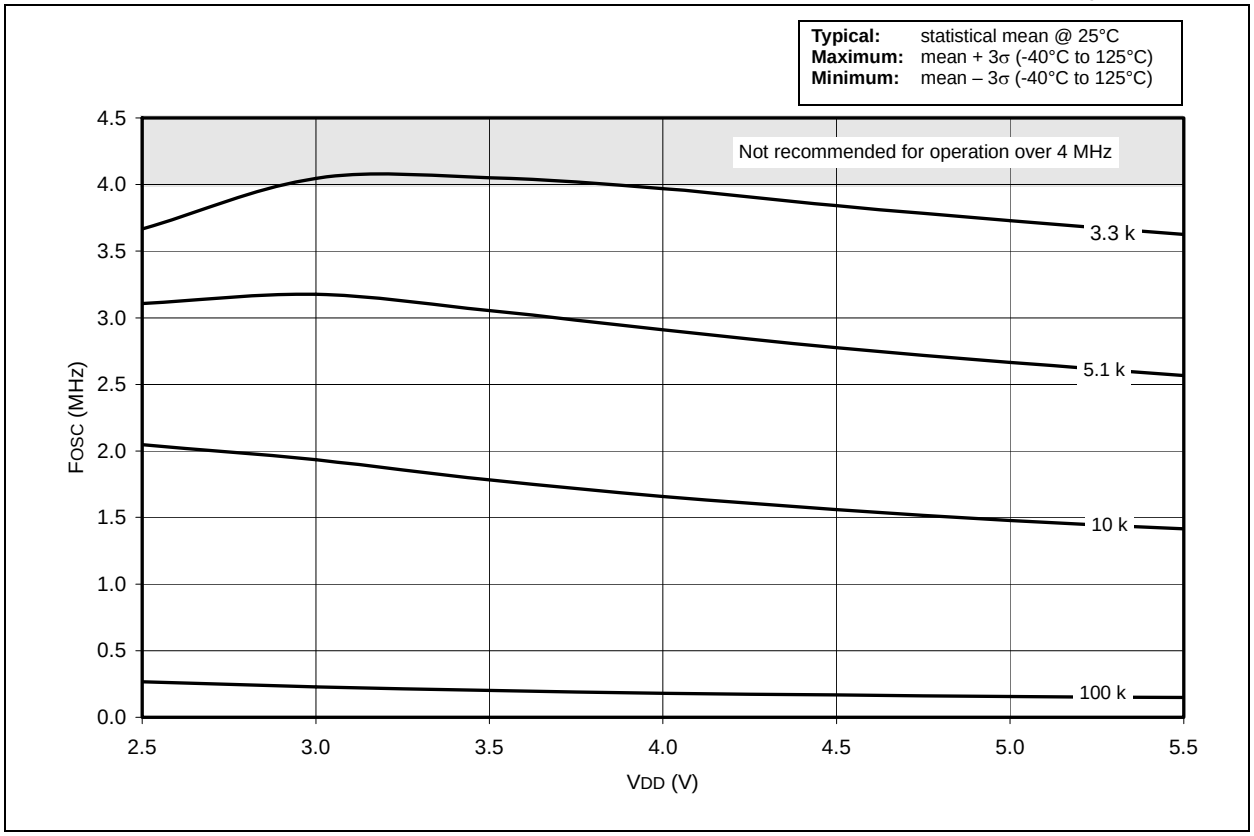
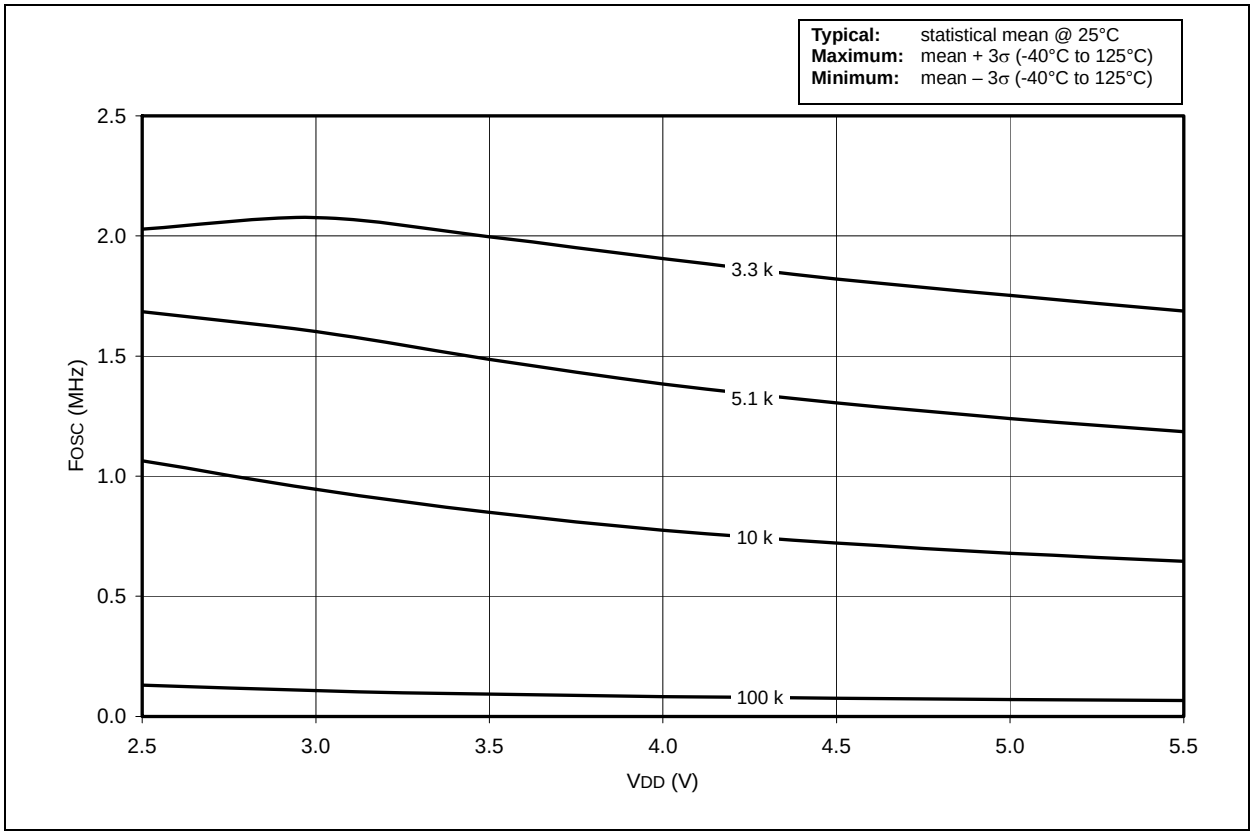



**FIGURE 17-8: AVERAGE FOSC VS. VDD FOR VARIOUS RESISTANCES – RC MODE; C = 100 PF**






Downloaded from Arrow.com.

**FIGURE 17-9: AVERAGE FOSC VS. VDD FOR VARIOUS RESISTANCES – RC MODE; C = 300 PF**

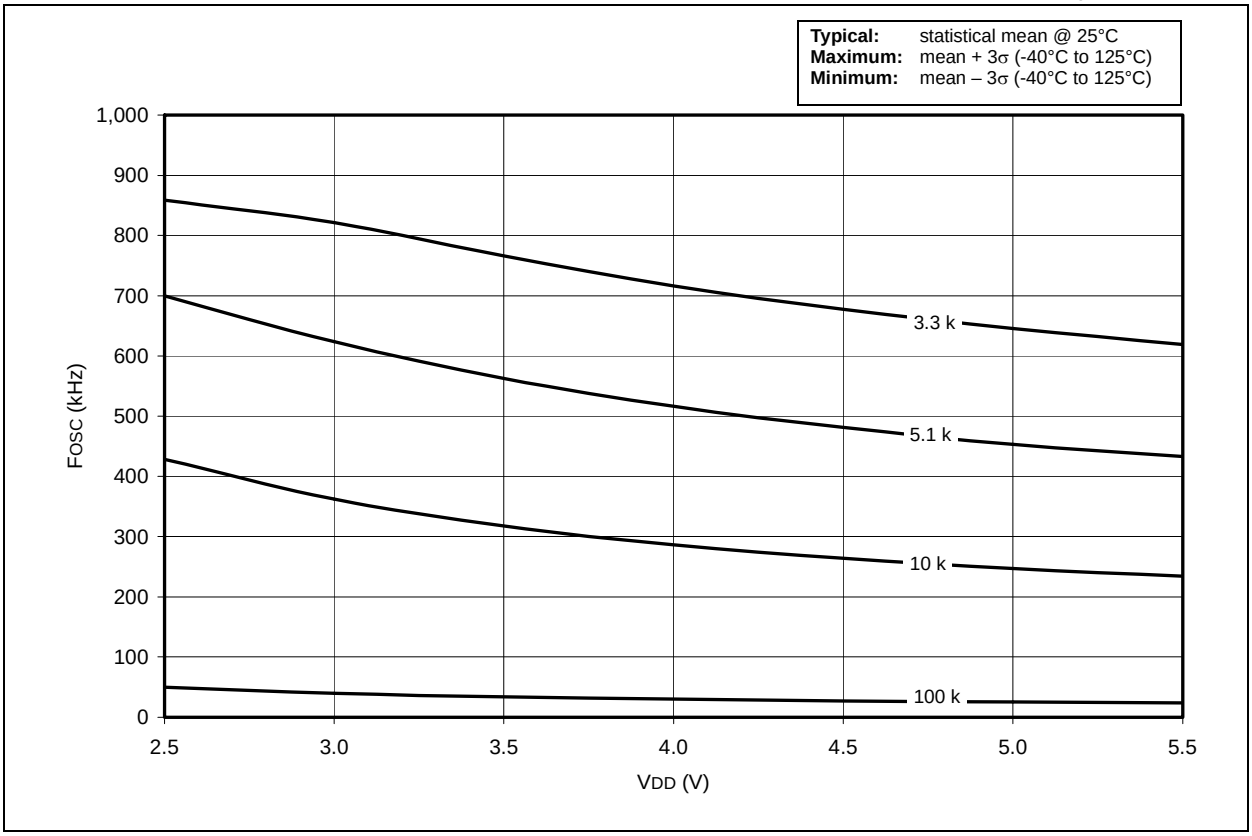
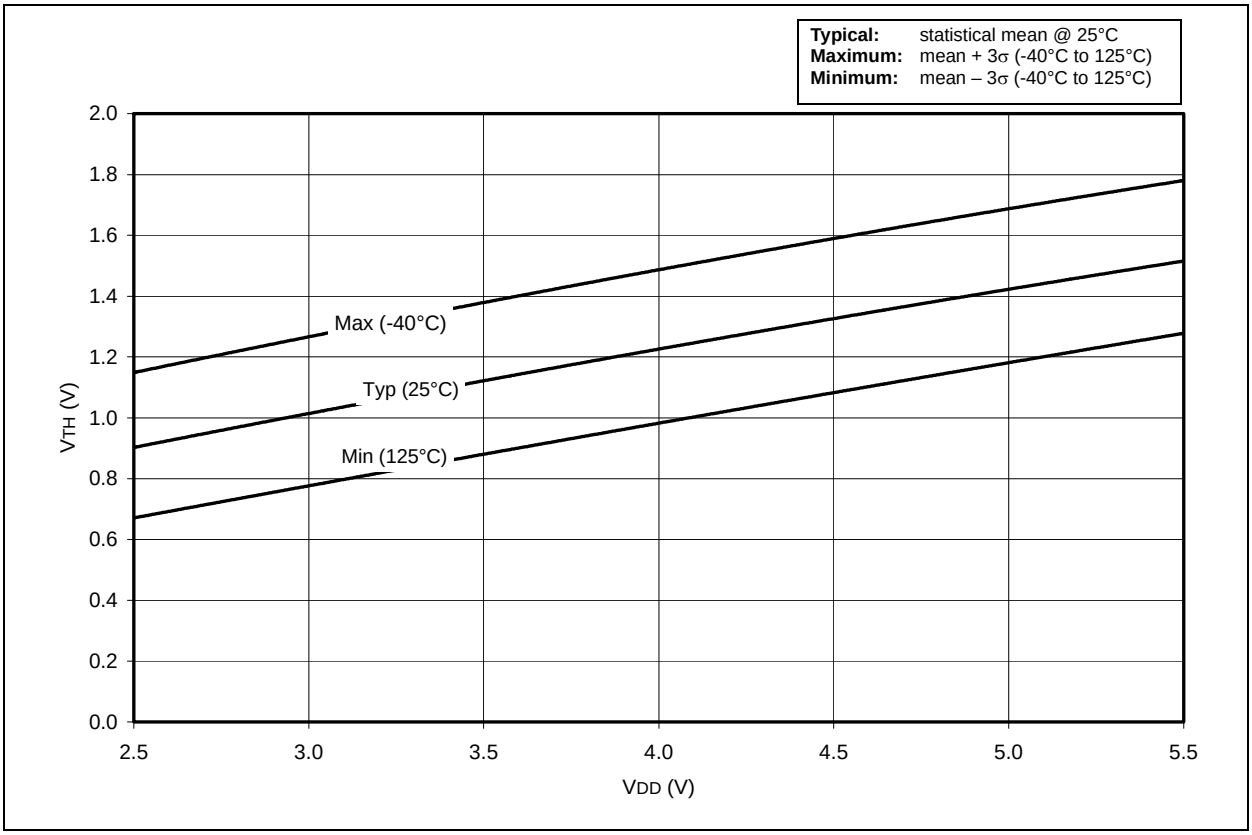



**FIGURE 17-10: VTH VS. VDD OVER TEMPERATURE – TTL INPUT**





© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 17-11: VIL, VIH vs. VDD OVER TEMPERATURE – SCHMITT TRIGGER INPUT ($I^2C$)**

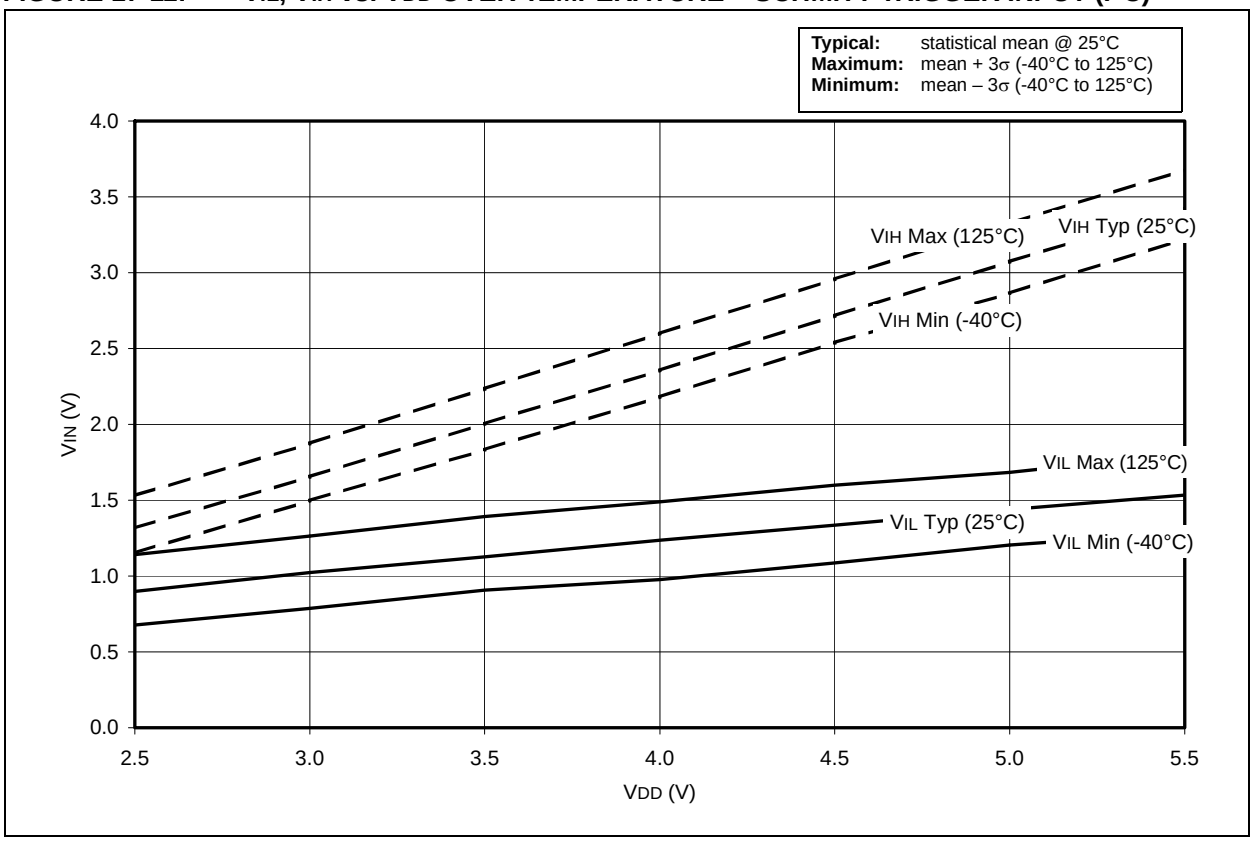
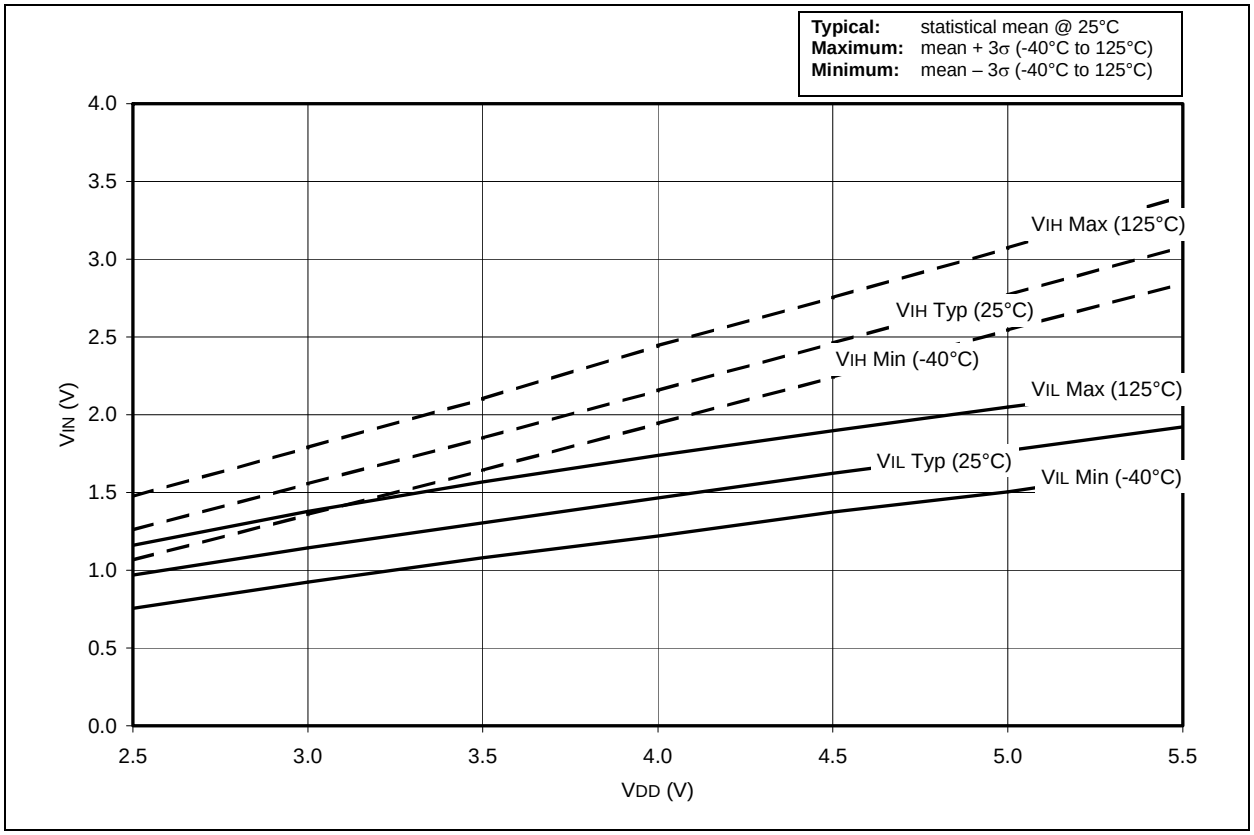



**FIGURE 17-12: VIL, VIH vs. VDD OVER TEMPERATURE – SCHMITT TRIGGER INPUT**






Downloaded from Arrow.com.

**FIGURE 17-13: $V_{OH}$ vs. $I_{OH}$ AT $V_{DD}$ = 3.0 V**

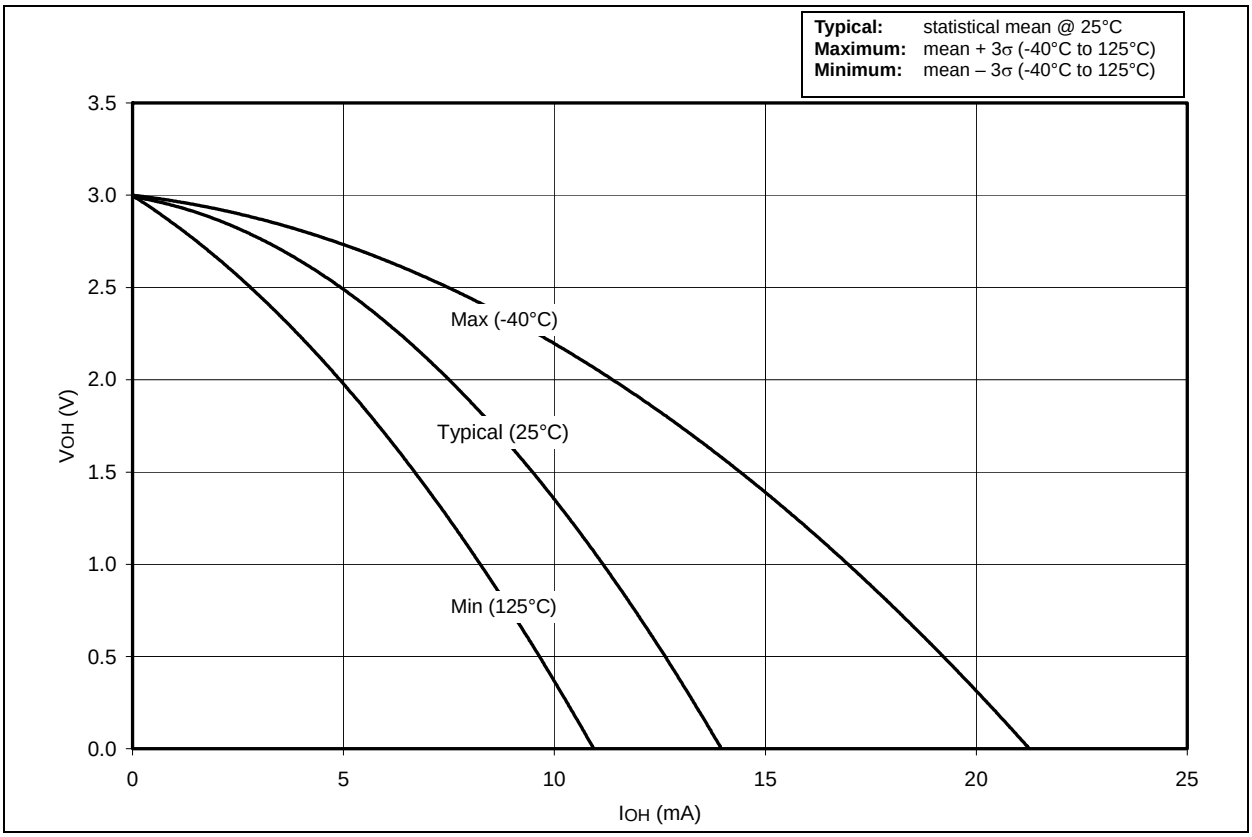
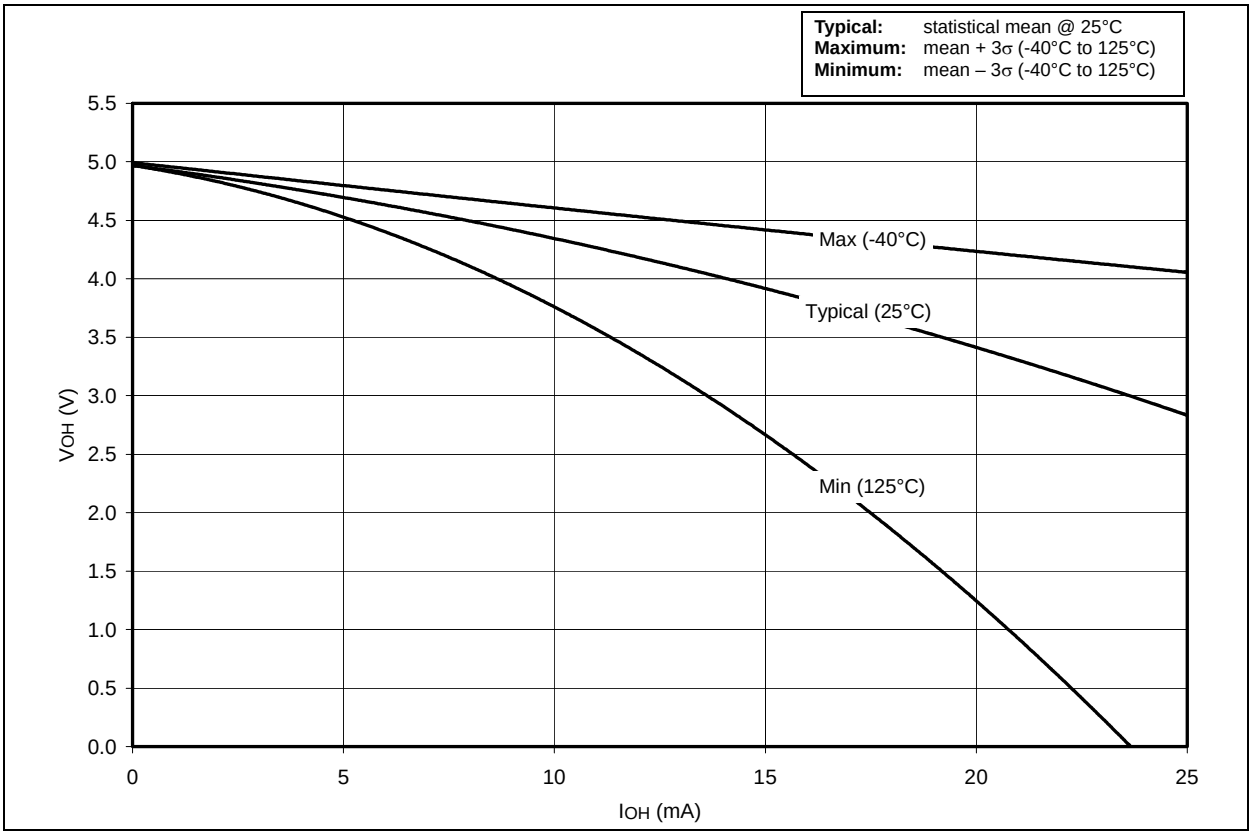



**FIGURE 17-14: $V_{OH}$ vs. $I_{OH}$ AT $V_{DD}$ = 5.0 V**





© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 17-15: VOL VS. IOL AT VDD = 3.0 V**

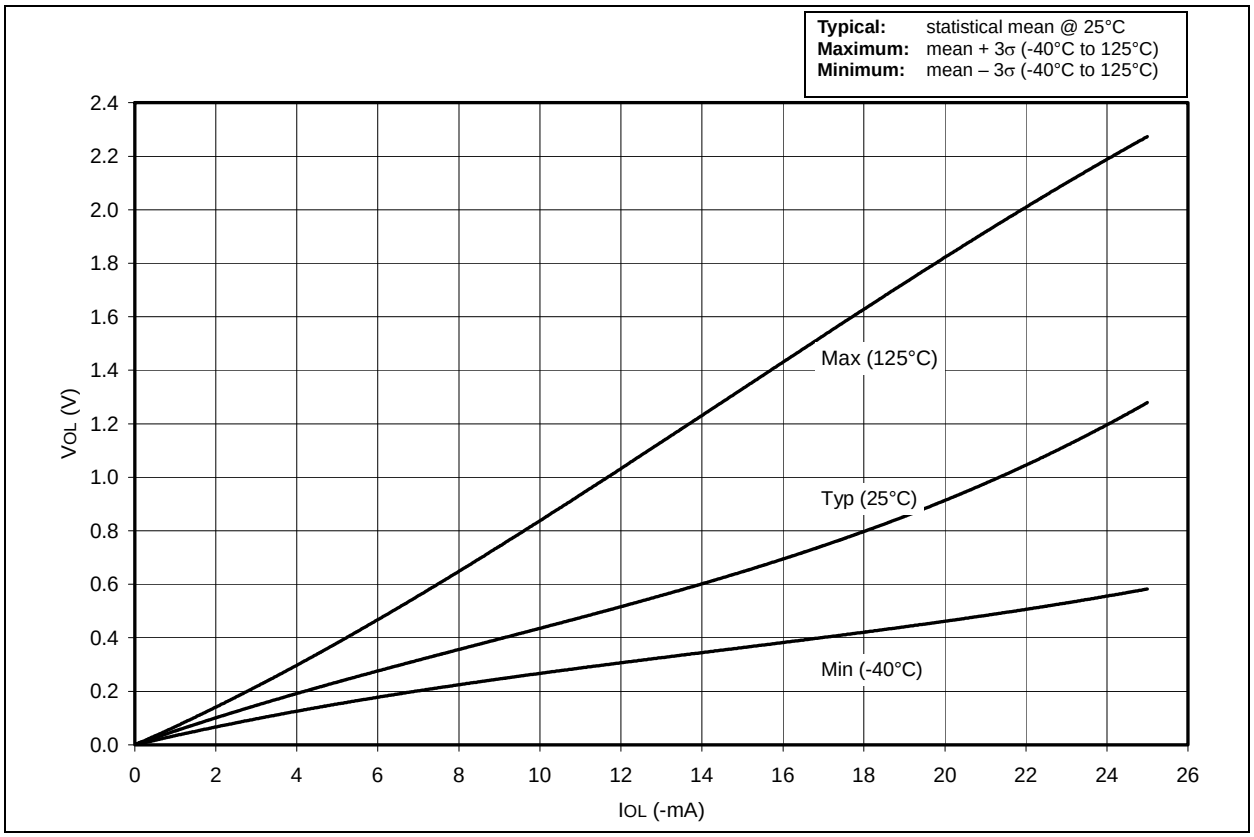
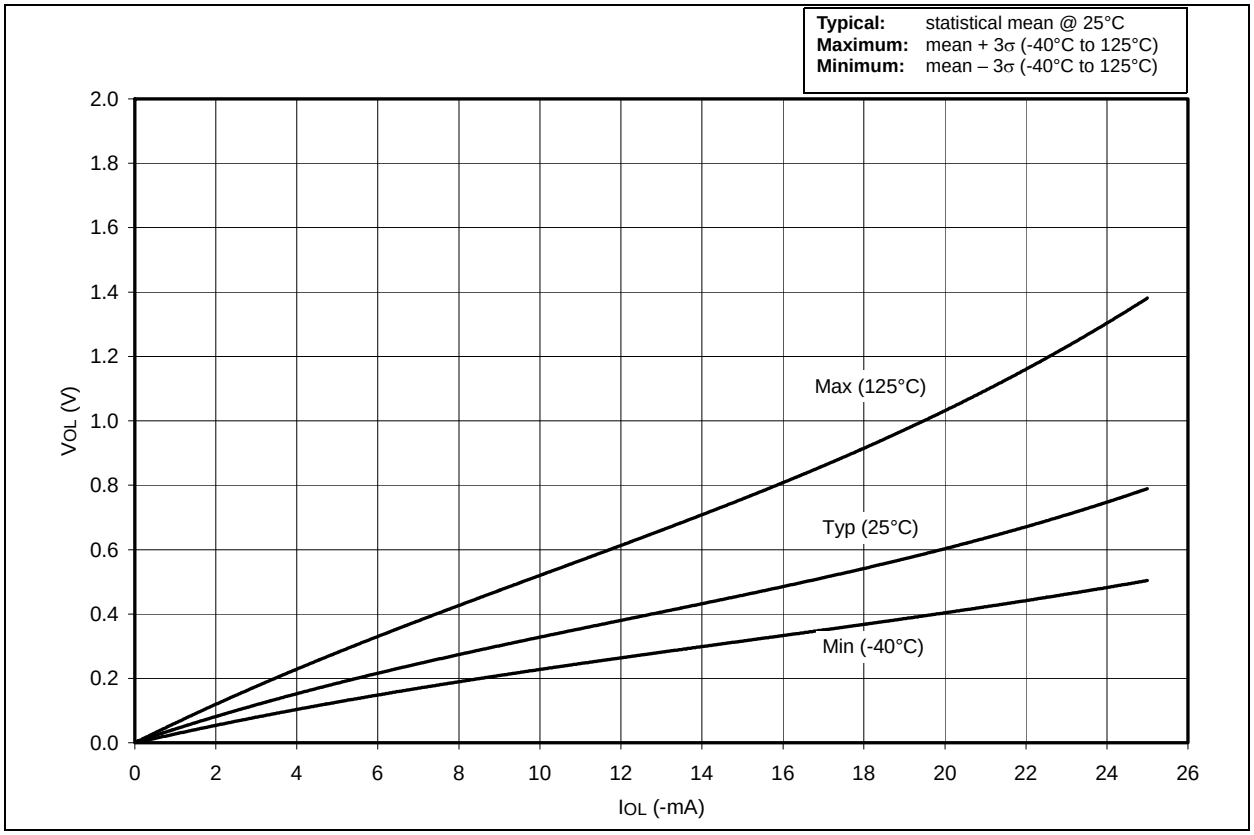



**FIGURE 17-16: VOL VS. IOL AT VDD = 5.0 V**






Downloaded from Arrow.com.

**FIGURE 17-17: IPD VS. VDD (85°C) – SLEEP MODE, ALL PERIPHERALS DISABLED**

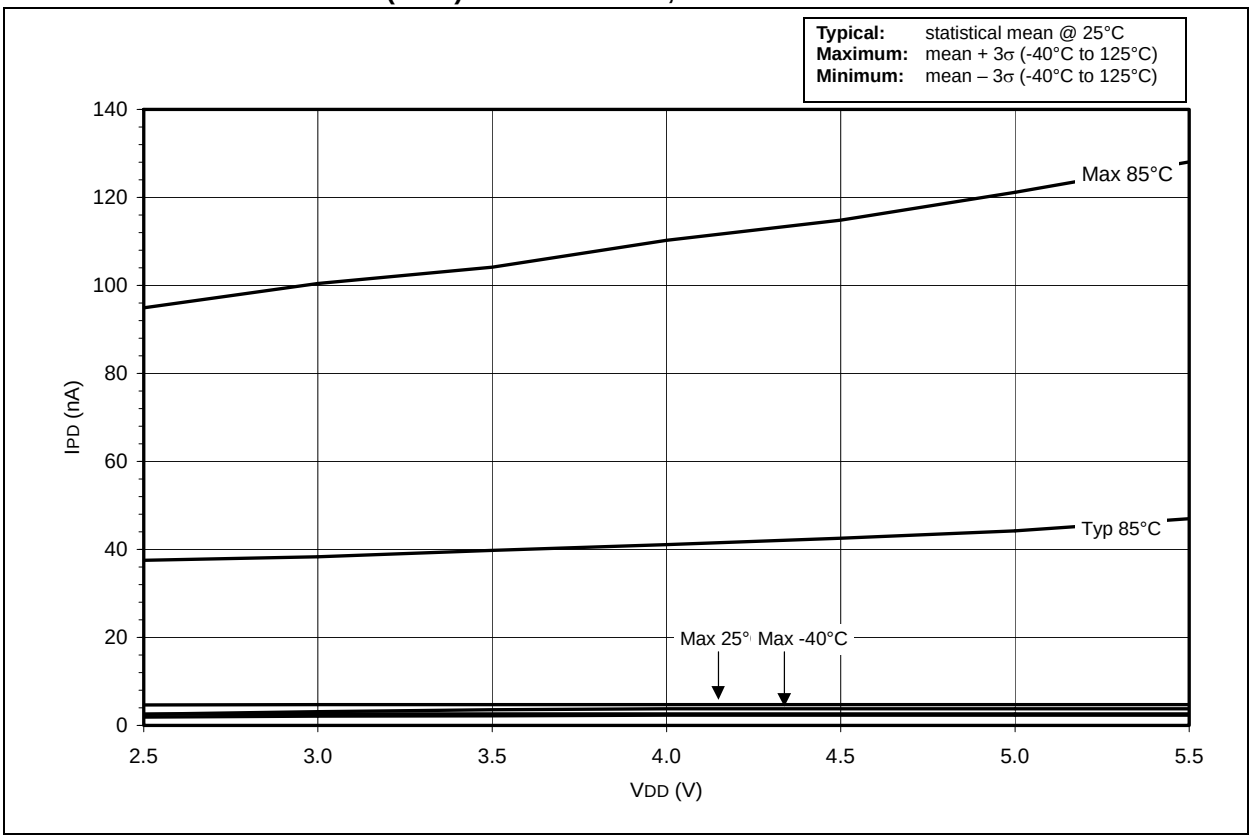
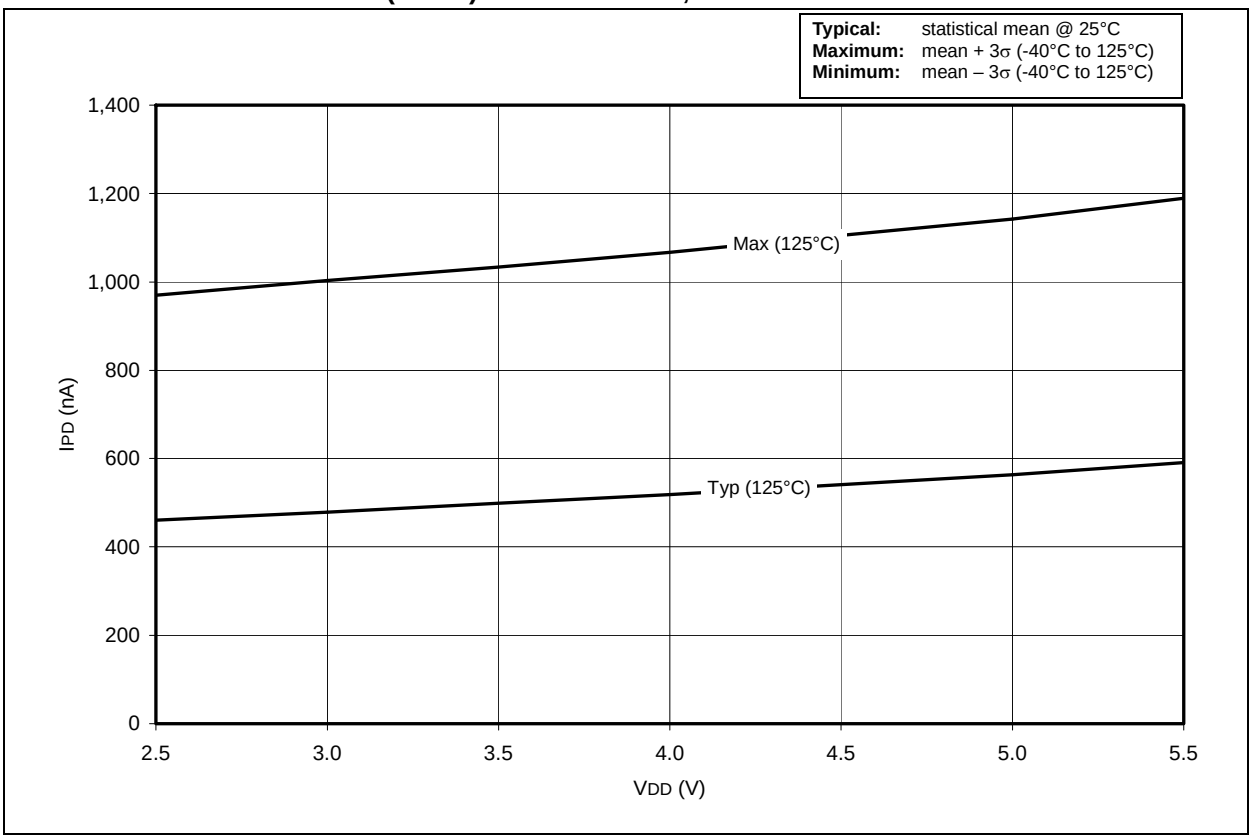



**FIGURE 17-18: IPD VS. VDD (125°C) – SLEEP MODE, ALL PERIPHERALS DISABLED**





© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 17-19: ΔIBOR VS. VDD OVER TEMPERATURE (-40°C TO +125°C)**

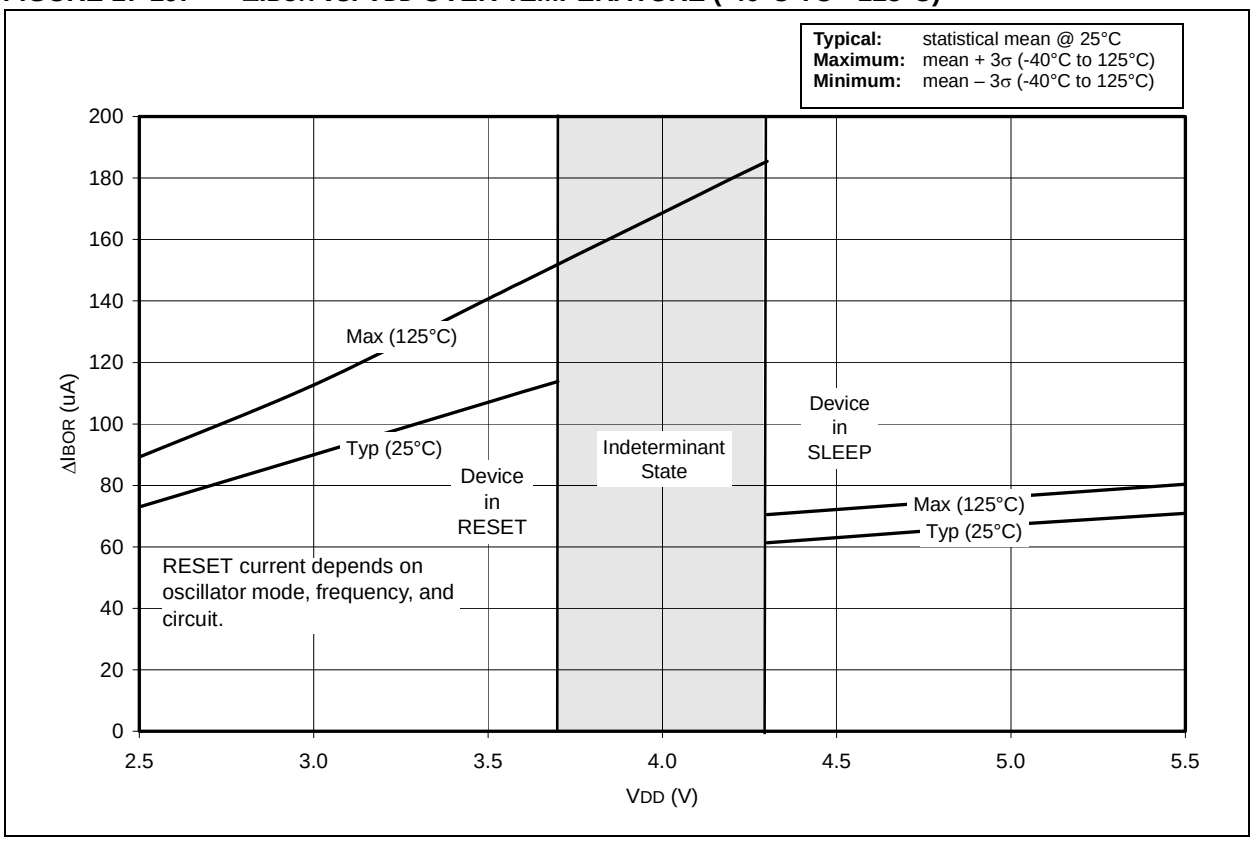
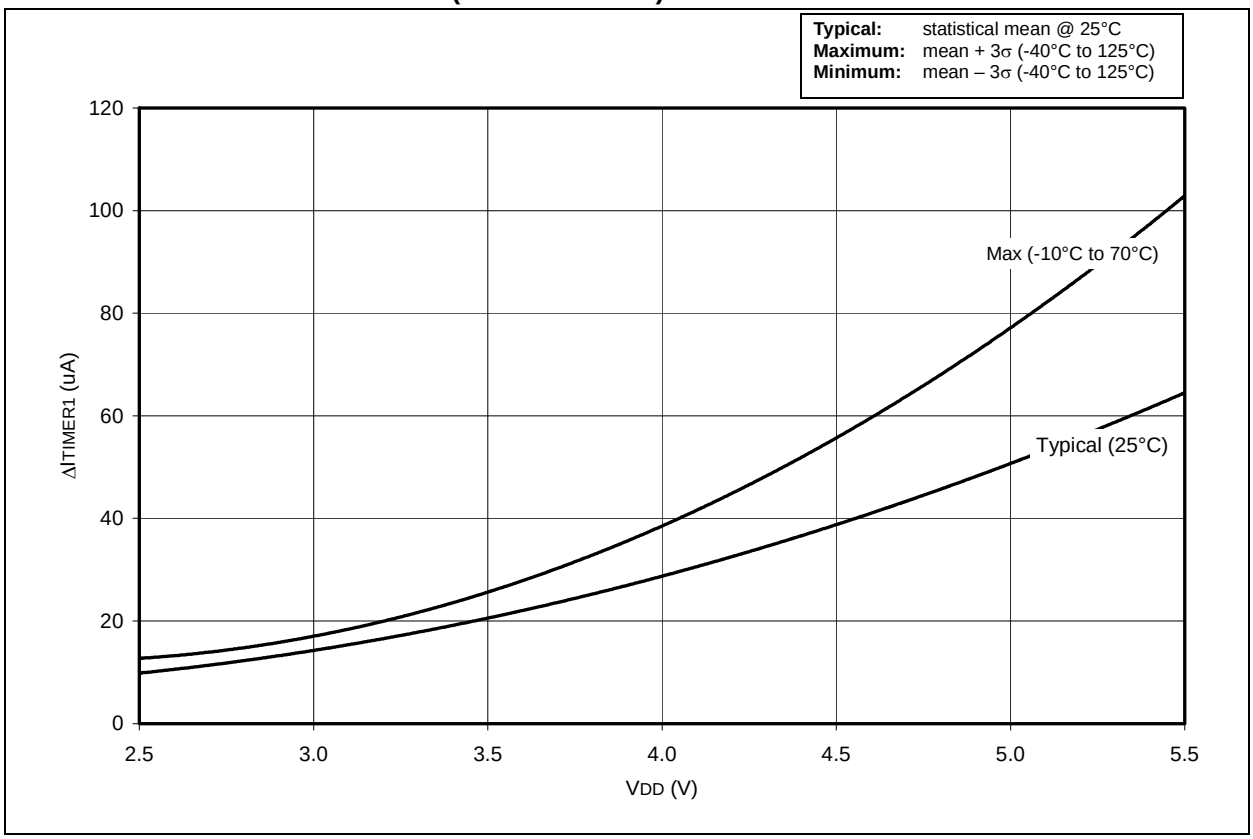



**FIGURE 17-20: ΔITIMER1 VS. VDD (-10°C TO +70°C)**






Downloaded from Arrow.com.

**FIGURE 17-21: ΔIWDT vs. VDD (-40°C TO +125°C)**

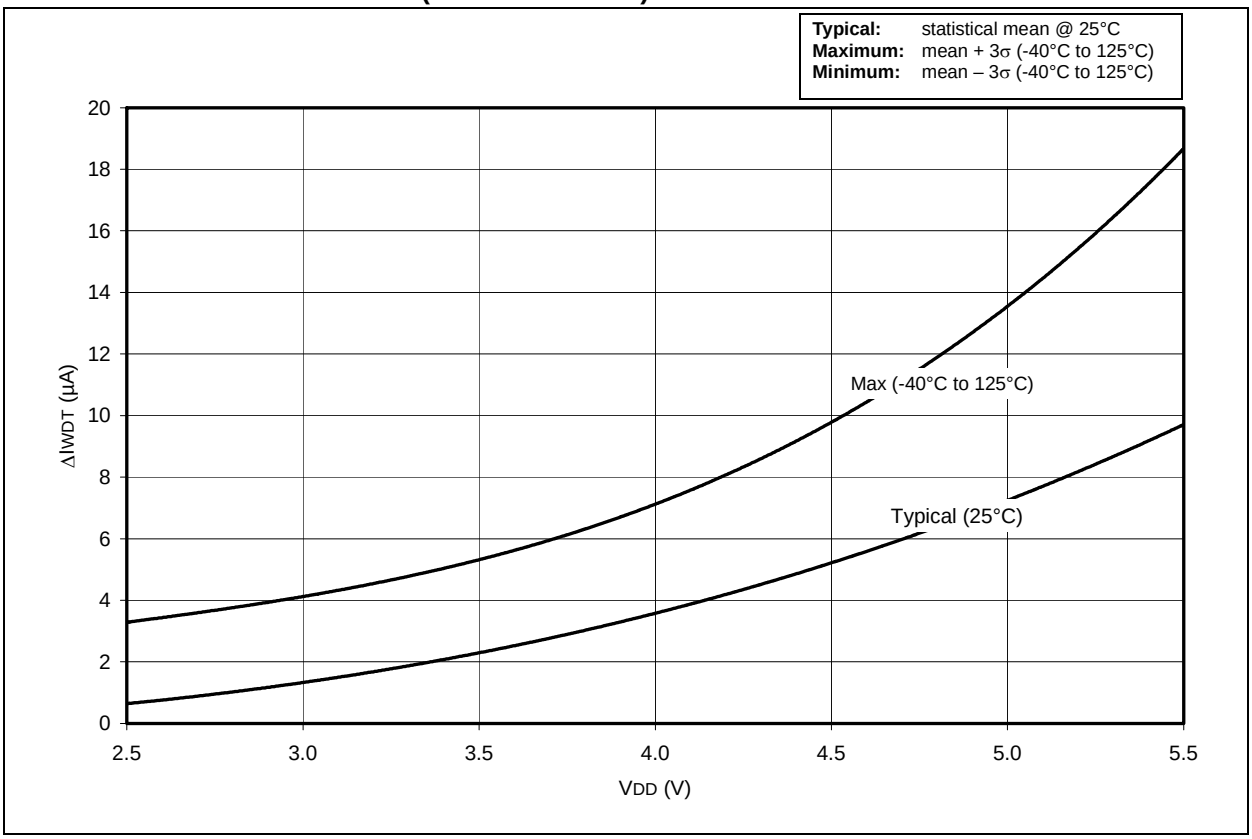
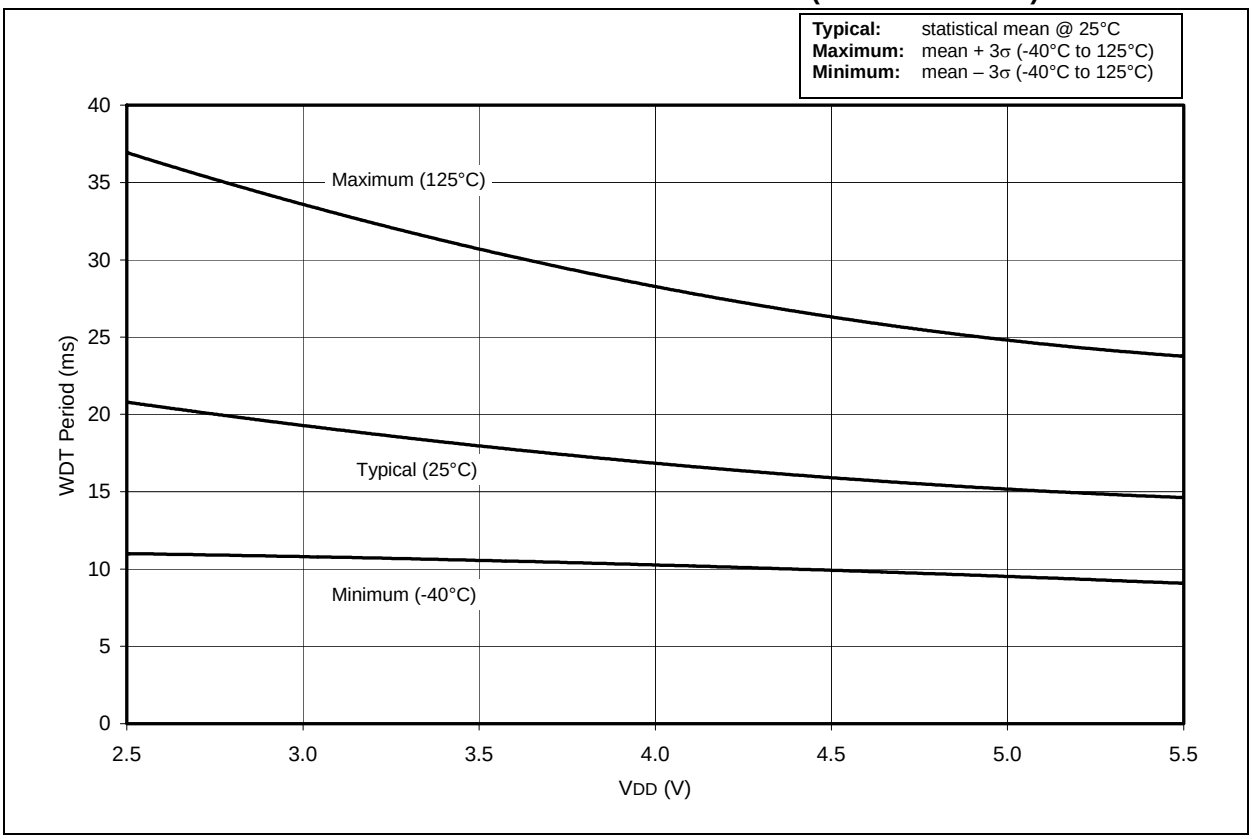



**FIGURE 17-22: WDT PERIOD vs. VDD OVER TEMPERATURE (-40°C TO +125°C)**

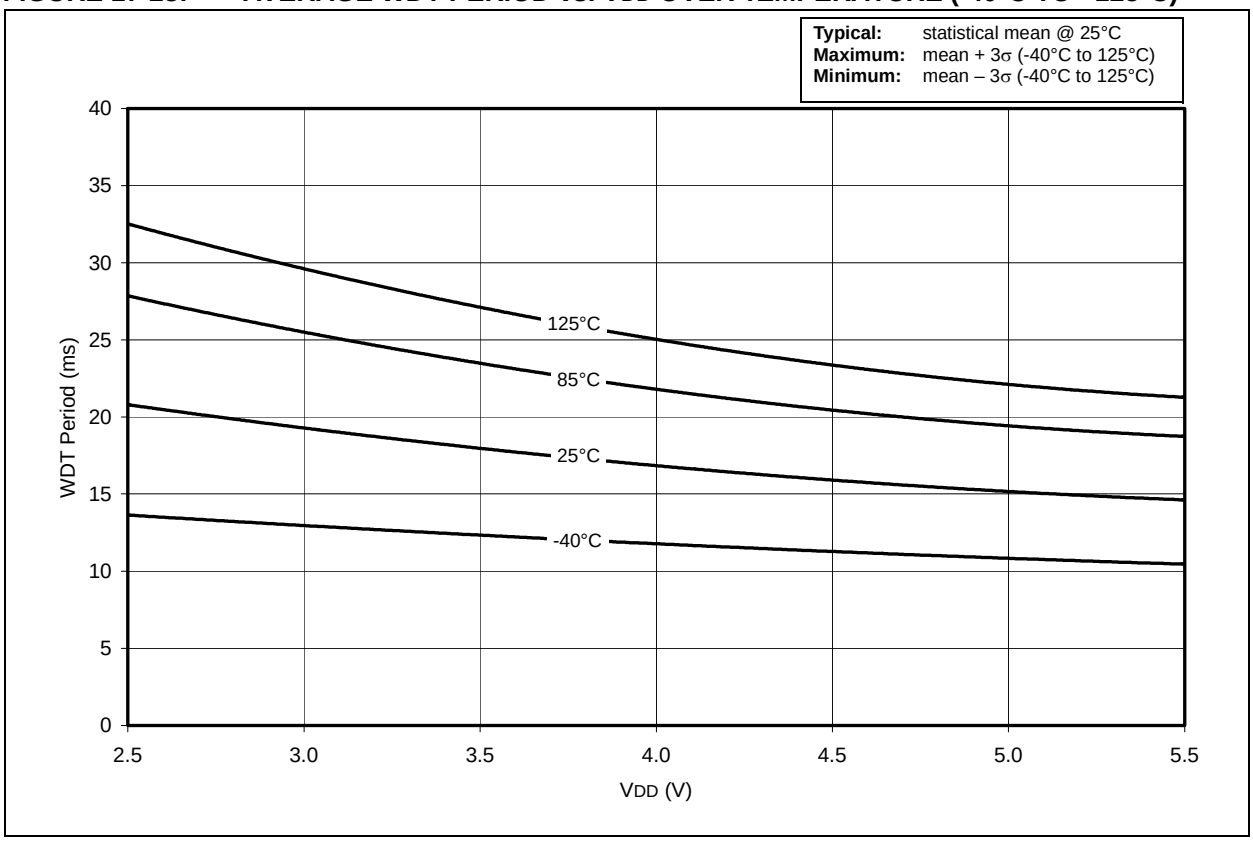




© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**FIGURE 17-23: AVERAGE WDT PERIOD vs. VDD OVER TEMPERATURE (-40°C TO +125°C)**






Downloaded from Arrow.com.

**NOTES:**

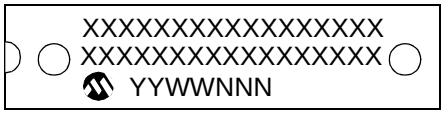

© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

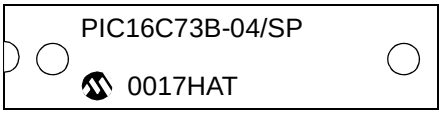
## 18.0 PACKAGING INFORMATION

### 18.1 Package Marking Information

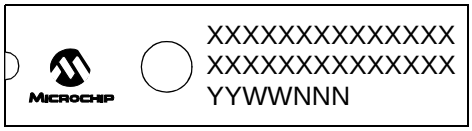
28-Lead PDIP (Skinny DIP)



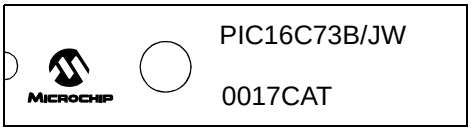

Example



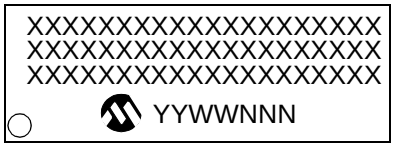

28-Lead CERDIP Windowed



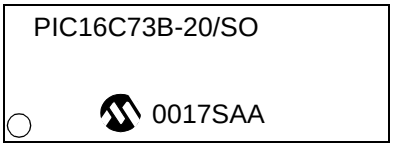

Example




28-Lead SOIC

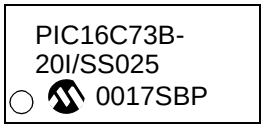



Example




28-Lead SSOP

Example




| | | |
|---|---|---|
| **Legend:** | XX...X | Customer-specific information |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week ‘01’) |
| | NNN | Alphanumeric traceability code |
| | (e3) | Pb-free JEDEC designator for Matte Tin (Sn) |
| | * | This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package. |

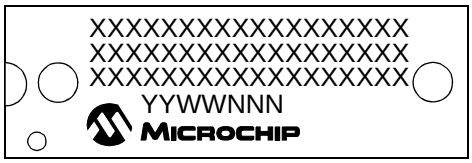
**Note**: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.



Downloaded from Arrow.com.

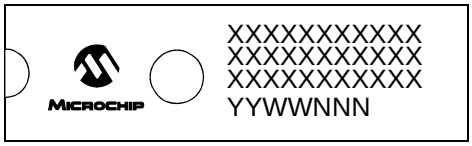
### Package Marking Information (Cont'd)

40-Lead PDIP

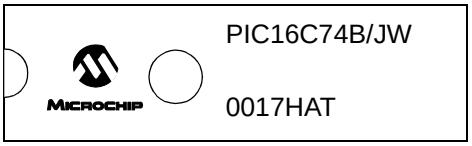



Example

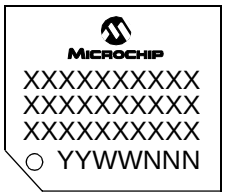
40-Lead CERDIP Windowed



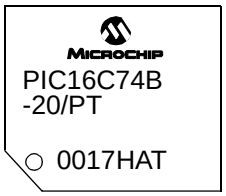

Example



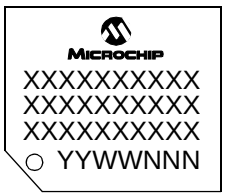

44-Lead TQFP



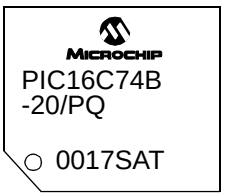

Example



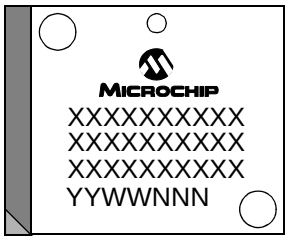

44-Lead MQFP



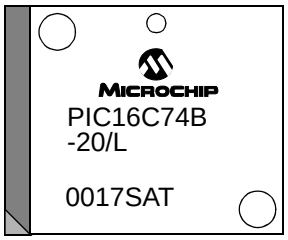

Example




44-Lead PLCC




Example



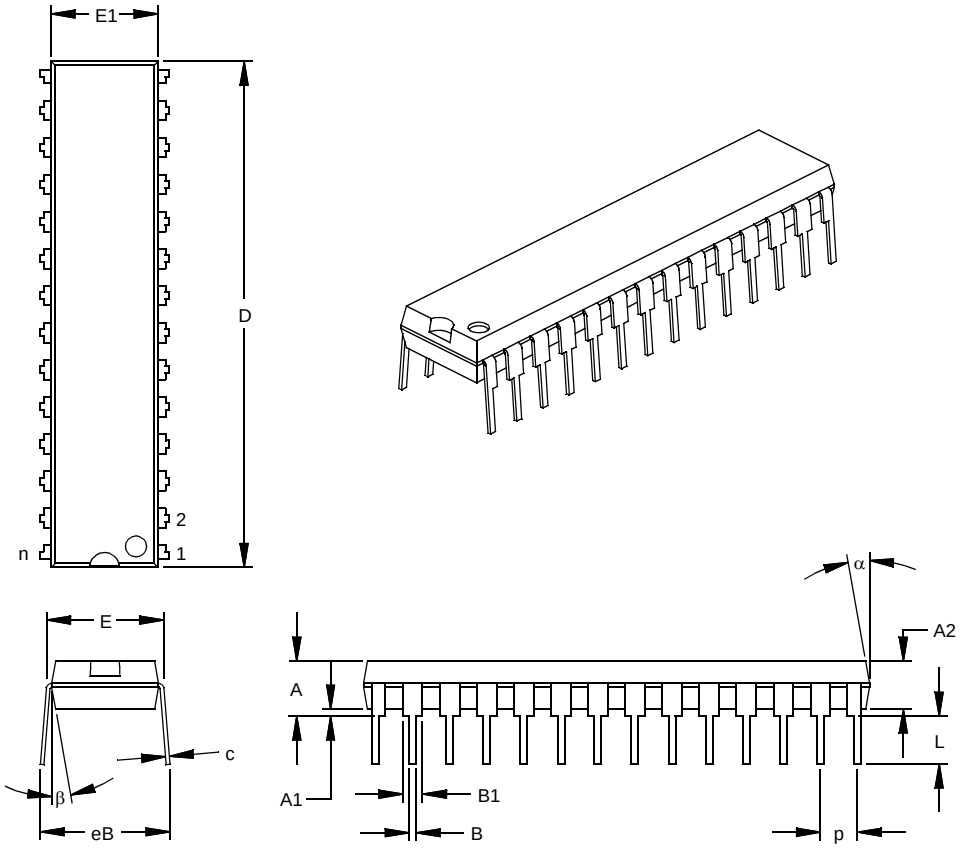


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 18.2 28-Lead Skinny Plastic Dual In-line (SP) – 300 mil (PDIP)

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 28 | | | 28 | |
| Pitch | p | | .100 | | | 2.54 | |
| Top to Seating Plane | A | .140 | .150 | .160 | 3.56 | 3.81 | 4.06 |
| Molded Package Thickness | A2 | .125 | .130 | .135 | 3.18 | 3.30 | 3.43 |
| Base to Seating Plane | A1 | .015 | | | 0.38 | | |
| Shoulder to Shoulder Width | E | .300 | .310 | .325 | 7.62 | 7.87 | 8.26 |
| Molded Package Width | E1 | .275 | .285 | .295 | 6.99 | 7.24 | 7.49 |
| Overall Length | D | 1.345 | 1.365 | 1.385 | 34.16 | 34.67 | 35.18 |
| Tip to Seating Plane | L | .125 | .130 | .135 | 3.18 | 3.30 | 3.43 |
| Lead Thickness | c | .008 | .012 | .015 | 0.20 | 0.29 | 0.38 |
| Upper Lead Width | B1 | .040 | .053 | .065 | 1.02 | 1.33 | 1.65 |
| Lower Lead Width | B | .016 | .019 | .022 | 0.41 | 0.48 | 0.56 |
| Overall Row Spacing § | eB | .320 | .350 | .430 | 8.13 | 8.89 | 10.92 |
| Mold Draft Angle Top | α | 5 | 10 | 15 | 5 | 10 | 15 |
| Mold Draft Angle Bottom | β | 5 | 10 | 15 | 5 | 10 | 15 |

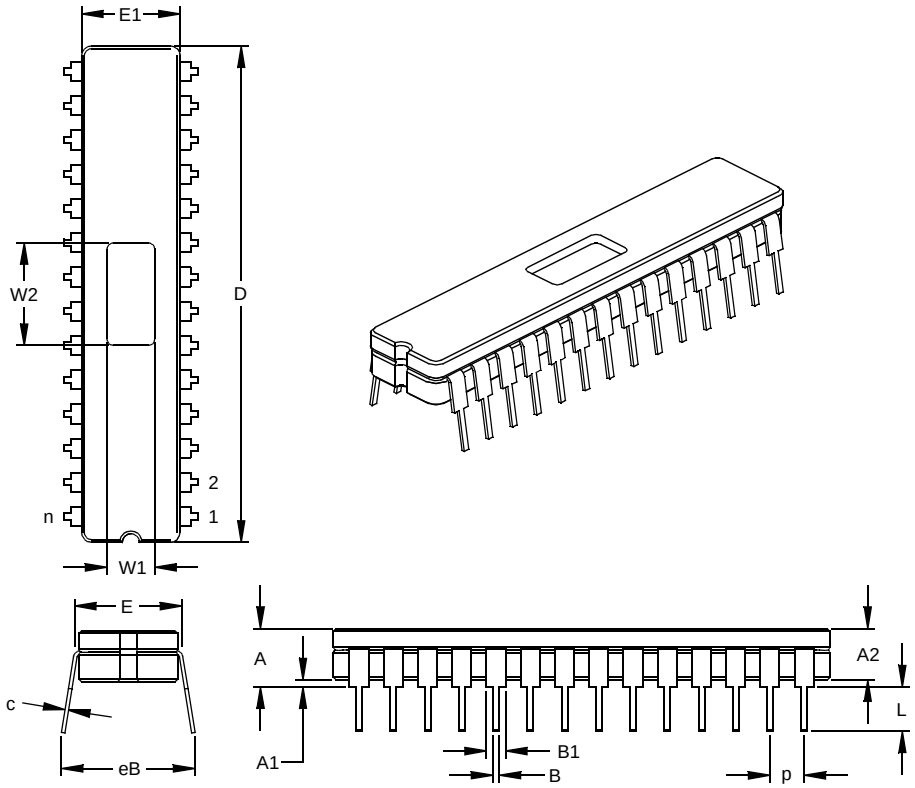
* Controlling Parameter
§ Significant Characteristic
Notes:
Dimension D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MO-095
Drawing No. C04-070



Downloaded from Arrow.com.

## 18.3 28-Lead Ceramic Dual In-line with Window (JW) – 300 mil (CERDIP)

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 28 | | | 28 | |
| Pitch | p | | .100 | | | 2.54 | |
| Top to Seating Plane | A | .170 | .183 | .195 | 4.32 | 4.64 | 4.95 |
| Ceramic Package Height | A2 | .155 | .160 | .165 | 3.94 | 4.06 | 4.19 |
| Standoff | A1 | .015 | .023 | .030 | 0.38 | 0.57 | 0.76 |
| Shoulder to Shoulder Width | E | .300 | .313 | .325 | 7.62 | 7.94 | 8.26 |
| Ceramic Pkg. Width | E1 | .285 | .290 | .295 | 7.24 | 7.37 | 7.49 |
| Overall Length | D | 1.430 | 1.458 | 1.485 | 36.32 | 37.02 | 37.72 |
| Tip to Seating Plane | L | .135 | .140 | .145 | 3.43 | 3.56 | 3.68 |
| Lead Thickness | c | .008 | .010 | .012 | 0.20 | 0.25 | 0.30 |
| Upper Lead Width | B1 | .050 | .058 | .065 | 1.27 | 1.46 | 1.65 |
| Lower Lead Width | B | .016 | .019 | .021 | 0.41 | 0.47 | 0.53 |
| Overall Row Spacing § | eB | .345 | .385 | .425 | 8.76 | 9.78 | 10.80 |
| Window Width | W1 | .130 | .140 | .150 | 3.30 | 3.56 | 3.81 |
| Window Length | W2 | .290 | .300 | .310 | 7.37 | 7.62 | 7.87 |

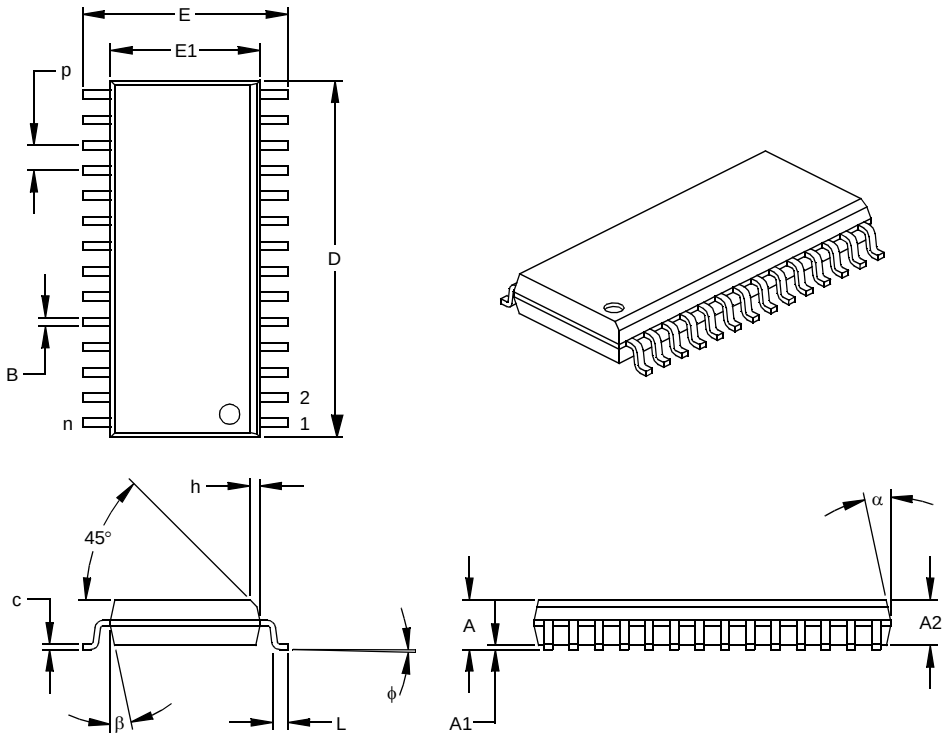
* Controlling Parameter
§ Significant Characteristic
JEDEC Equivalent: MO-058
Drawing No. C04-080


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 18.4 28-Lead Plastic Small Outline (SO) – Wide, 300 mil (SOIC)

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 28 | | | 28 | |
| Pitch | p | | .050 | | | 1.27 | |
| Overall Height | A | .093 | .099 | .104 | 2.36 | 2.50 | 2.64 |
| Molded Package Thickness | A2 | .088 | .091 | .094 | 2.24 | 2.31 | 2.39 |
| Standoff § | A1 | .004 | .008 | .012 | 0.10 | 0.20 | 0.30 |
| Overall Width | E | .394 | .407 | .420 | 10.01 | 10.34 | 10.67 |
| Molded Package Width | E1 | .288 | .295 | .299 | 7.32 | 7.49 | 7.59 |
| Overall Length | D | .695 | .704 | .712 | 17.65 | 17.87 | 18.08 |
| Chamfer Distance | h | .010 | .020 | .029 | 0.25 | 0.50 | 0.74 |
| Foot Length | L | .016 | .033 | .050 | 0.41 | 0.84 | 1.27 |
| Foot Angle Top | ϕ | 0 | 4 | 8 | 0 | 4 | 8 |
| Lead Thickness | c | .009 | .011 | .013 | 0.23 | 0.28 | 0.33 |
| Lead Width | B | .014 | .017 | .020 | 0.36 | 0.42 | 0.51 |
| Mold Draft Angle Top | α | 0 | 12 | 15 | 0 | 12 | 15 |
| Mold Draft Angle Bottom | β | 0 | 12 | 15 | 0 | 12 | 15 |

* Controlling Parameter
§ Significant Characteristic

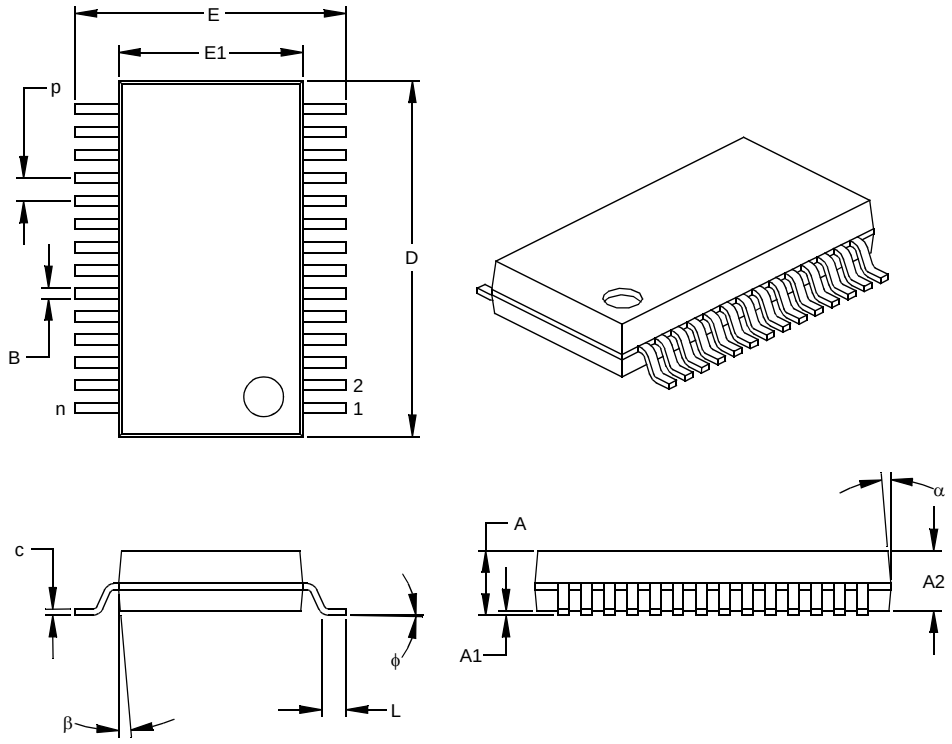
Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-013
Drawing No. C04-052



Downloaded from Arrow.com.

## 18.5 28-Lead Plastic Shrink Small Outline (SS) – 209 mil, 5.30 mm (SSOP)

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | INCHES | | | MILLIMETERS* | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 28 | | | 28 | |
| Pitch | p | | .026 | | | 0.65 | |
| Overall Height | A | .068 | .073 | .078 | 1.73 | 1.85 | 1.98 |
| Molded Package Thickness | A2 | .064 | .068 | .072 | 1.63 | 1.73 | 1.83 |
| Standoff § | A1 | .002 | .006 | .010 | 0.05 | 0.15 | 0.25 |
| Overall Width | E | .299 | .309 | .319 | 7.59 | 7.85 | 8.10 |
| Molded Package Width | E1 | .201 | .207 | .212 | 5.11 | 5.25 | 5.38 |
| Overall Length | D | .396 | .402 | .407 | 10.06 | 10.20 | 10.34 |
| Foot Length | L | .022 | .030 | .037 | 0.56 | 0.75 | 0.94 |
| Lead Thickness | c | .004 | .007 | .010 | 0.10 | 0.18 | 0.25 |
| Foot Angle | ϕ | 0 | 4 | 8 | 0.00 | 101.60 | 203.20 |
| Lead Width | B | .010 | .013 | .015 | 0.25 | 0.32 | 0.38 |
| Mold Draft Angle Top | α | 0 | 5 | 10 | 0 | 5 | 10 |
| Mold Draft Angle Bottom | β | 0 | 5 | 10 | 0 | 5 | 10 |

* Controlling Parameter
§ Significant Characteristic

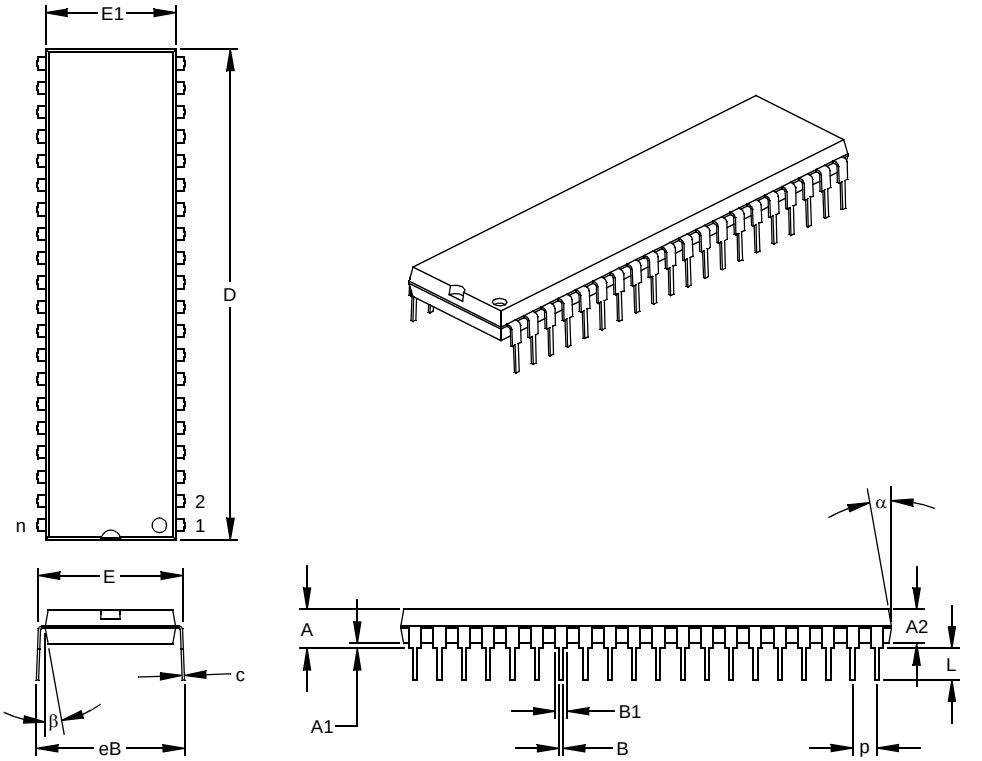
Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-150
Drawing No. C04-073


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 18.6 40-Lead Plastic Dual In-line (P) – 600 mil (PDIP)

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 40 | | | 40 | |
| Pitch | p | | .100 | | | 2.54 | |
| Top to Seating Plane | A | .160 | .175 | .190 | 4.06 | 4.45 | 4.83 |
| Molded Package Thickness | A2 | .140 | .150 | .160 | 3.56 | 3.81 | 4.06 |
| Base to Seating Plane | A1 | .015 | | | 0.38 | | |
| Shoulder to Shoulder Width | E | .595 | .600 | .625 | 15.11 | 15.24 | 15.88 |
| Molded Package Width | E1 | .530 | .545 | .560 | 13.46 | 13.84 | 14.22 |
| Overall Length | D | 2.045 | 2.058 | 2.065 | 51.94 | 52.26 | 52.45 |
| Tip to Seating Plane | L | .120 | .130 | .135 | 3.05 | 3.30 | 3.43 |
| Lead Thickness | c | .008 | .012 | .015 | 0.20 | 0.29 | 0.38 |
| Upper Lead Width | B1 | .030 | .050 | .070 | 0.76 | 1.27 | 1.78 |
| Lower Lead Width | B | .014 | .018 | .022 | 0.36 | 0.46 | 0.56 |
| Overall Row Spacing § | eB | .620 | .650 | .680 | 15.75 | 16.51 | 17.27 |
| Mold Draft Angle Top | α | 5 | 10 | 15 | 5 | 10 | 15 |
| Mold Draft Angle Bottom | β | 5 | 10 | 15 | 5 | 10 | 15 |

* Controlling Parameter
§ Significant Characteristic

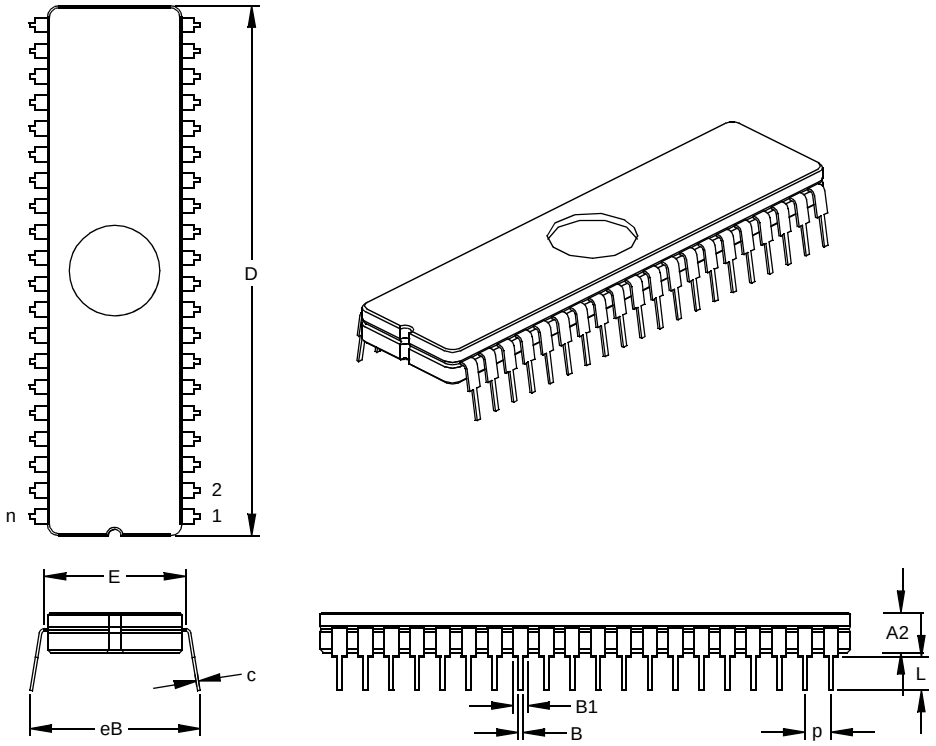
Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MO-011
Drawing No. C04-016



Downloaded from Arrow.com.

## 18.7 40-Lead Ceramic Dual In-line with Window (JW) – 600 mil (CERDIP)

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 40 | | | 40 | |
| Pitch | p | | .100 | | | 2.54 | |
| Top to Seating Plane | A | .185 | .205 | .225 | 4.70 | 5.21 | 5.72 |
| Ceramic Package Height | A2 | .155 | .160 | .165 | 3.94 | 4.06 | 4.19 |
| Standoff | A1 | .030 | .045 | .060 | 0.76 | 1.14 | 1.52 |
| Shoulder to Shoulder Width | E | .595 | .600 | .625 | 15.11 | 15.24 | 15.88 |
| Ceramic Pkg. Width | E1 | .514 | .520 | .526 | 13.06 | 13.21 | 13.36 |
| Overall Length | D | 2.040 | 2.050 | 2.060 | 51.82 | 52.07 | 52.32 |
| Tip to Seating Plane | L | .135 | .140 | .145 | 3.43 | 3.56 | 3.68 |
| Lead Thickness | c | .008 | .011 | .014 | 0.20 | 0.28 | 0.36 |
| Upper Lead Width | B | .050 | .053 | .055 | 1.27 | 1.33 | 1.40 |
| Lower Lead Width | B1 | .016 | .020 | .023 | 0.41 | 0.51 | 0.58 |
| Overall Row Spacing § | eB | .610 | .660 | .710 | 15.49 | 16.76 | 18.03 |
| Window Diameter | W | .340 | .350 | .360 | 8.64 | 8.89 | 9.14 |

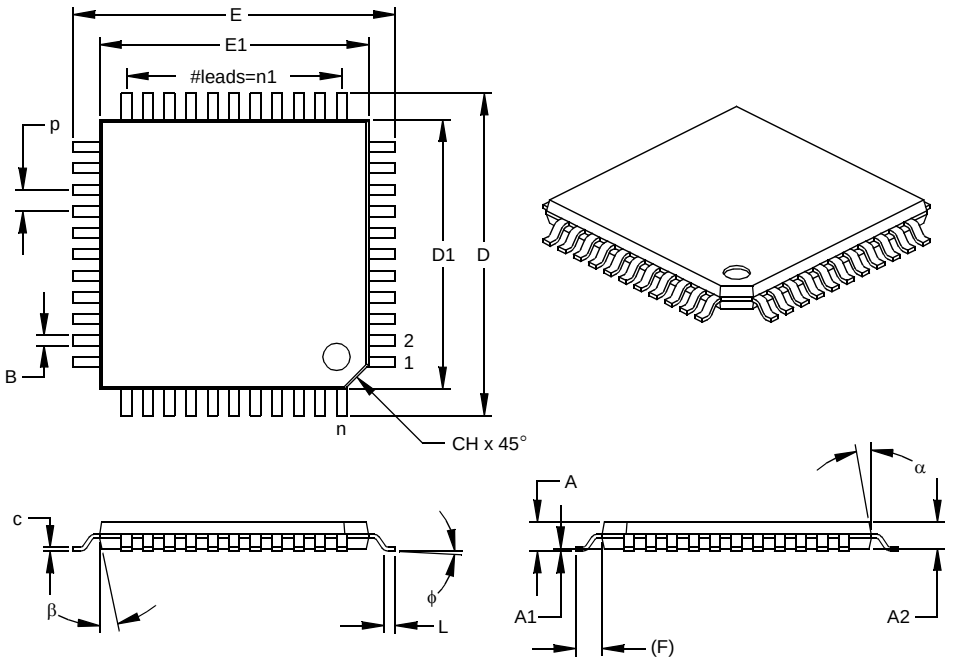
* Controlling Parameter
§ Significant Characteristic
JEDEC Equivalent: MO-103
Drawing No. C04-014


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 18.8 44-Lead Plastic Thin Quad Flatpack (PT) 10x10x1 mm Body, 1.0/0.10 mm Lead Form (TQFP)

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | INCHES | | | MILLIMETERS* | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 44 | | | 44 | |
| Pitch | p | | .031 | | | 0.80 | |
| Pins per Side | n1 | | 11 | | | 11 | |
| Overall Height | A | .039 | .043 | .047 | 1.00 | 1.10 | 1.20 |
| Molded Package Thickness | A2 | .037 | .039 | .041 | 0.95 | 1.00 | 1.05 |
| Standoff § | A1 | .002 | .004 | .006 | 0.05 | 0.10 | 0.15 |
| Foot Length | L | .018 | .024 | .030 | 0.45 | 0.60 | 0.75 |
| Footprint (Reference) | (F) | | .039 | | 1.00 | | |
| Foot Angle | ϕ | 0 | 3.5 | 7 | 0 | 3.5 | 7 |
| Overall Width | E | .463 | .472 | .482 | 11.75 | 12.00 | 12.25 |
| Overall Length | D | .463 | .472 | .482 | 11.75 | 12.00 | 12.25 |
| Molded Package Width | E1 | .390 | .394 | .398 | 9.90 | 10.00 | 10.10 |
| Molded Package Length | D1 | .390 | .394 | .398 | 9.90 | 10.00 | 10.10 |
| Lead Thickness | c | .004 | .006 | .008 | 0.09 | 0.15 | 0.20 |
| Lead Width | B | .012 | .015 | .017 | 0.30 | 0.38 | 0.44 |
| Pin 1 Corner Chamfer | CH | .025 | .035 | .045 | 0.64 | 0.89 | 1.14 |
| Mold Draft Angle Top | α | 5 | 10 | 15 | 5 | 10 | 15 |
| Mold Draft Angle Bottom | β | 5 | 10 | 15 | 5 | 10 | 15 |

* Controlling Parameter
§ Significant Characteristic

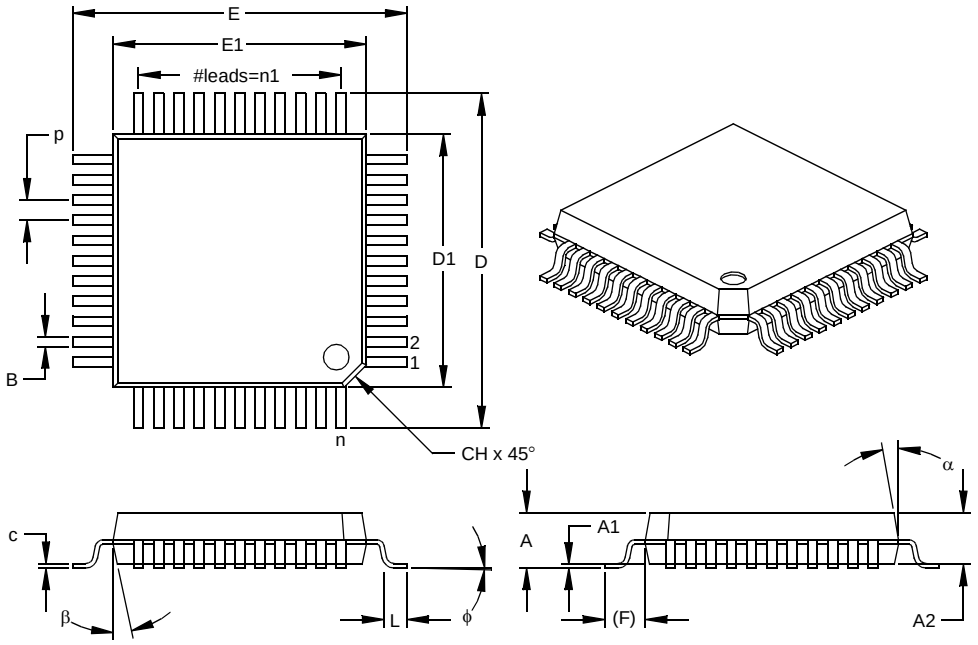
Notes:
Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-026
Drawing No. C04-076



Downloaded from Arrow.com.

## 18.9 44-Lead Plastic Metric Quad Flatpack (PQ) 10x10x2 mm Body, 1.6/0.15 mm Lead Form (MQFP)

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | INCHES | | | MILLIMETERS* | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 44 | | | 44 | |
| Pitch | p | | .031 | | | 0.80 | |
| Pins per Side | n1 | | 11 | | | 11 | |
| Overall Height | A | .079 | .086 | .093 | 2.00 | 2.18 | 2.35 |
| Molded Package Thickness | A2 | .077 | .080 | .083 | 1.95 | 2.03 | 2.10 |
| Standoff § | A1 | .002 | .006 | .010 | 0.05 | 0.15 | 0.25 |
| Foot Length | L | .029 | .035 | .041 | 0.73 | 0.88 | 1.03 |
| Footprint (Reference) | (F) | | .063 | | | 1.60 | |
| Foot Angle | ϕ | 0 | 3.5 | 7 | 0 | 3.5 | 7 |
| Overall Width | E | .510 | .520 | .530 | 12.95 | 13.20 | 13.45 |
| Overall Length | D | .510 | .520 | .530 | 12.95 | 13.20 | 13.45 |
| Molded Package Width | E1 | .390 | .394 | .398 | 9.90 | 10.00 | 10.10 |
| Molded Package Length | D1 | .390 | .394 | .398 | 9.90 | 10.00 | 10.10 |
| Lead Thickness | c | .005 | .007 | .009 | 0.13 | 0.18 | 0.23 |
| Lead Width | B | .012 | .015 | .018 | 0.30 | 0.38 | 0.45 |
| Pin 1 Corner Chamfer | CH | .025 | .035 | .045 | 0.64 | 0.89 | 1.14 |
| Mold Draft Angle Top | α | 5 | 10 | 15 | 5 | 10 | 15 |
| Mold Draft Angle Bottom | β | 5 | 10 | 15 | 5 | 10 | 15 |

* Controlling Parameter
§ Significant Characteristic

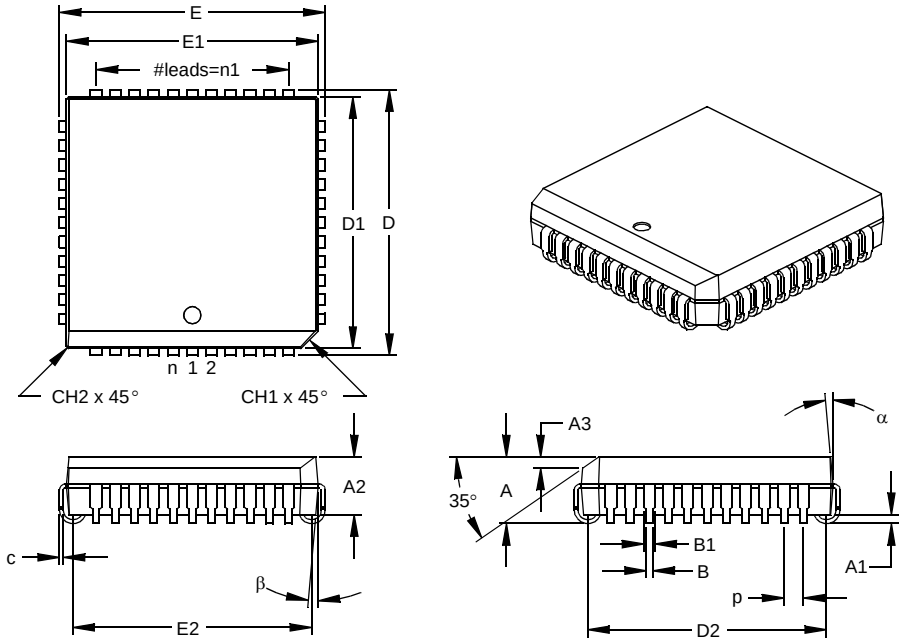
Notes:
Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-022
Drawing No. C04-071


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 18.10 44-Lead Plastic Leaded Chip Carrier (L) – Square (PLCC)

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 44 | | | 44 | |
| Pitch | p | | .050 | | | 1.27 | |
| Pins per Side | n1 | | 11 | | | 11 | |
| Overall Height | A | .165 | .173 | .180 | 4.19 | 4.39 | 4.57 |
| Molded Package Thickness | A2 | .145 | .153 | .160 | 3.68 | 3.87 | 4.06 |
| Standoff § | A1 | .020 | .028 | .035 | 0.51 | 0.71 | 0.89 |
| Side 1 Chamfer Height | A3 | .024 | .029 | .034 | 0.61 | 0.74 | 0.86 |
| Corner Chamfer 1 | CH1 | .040 | .045 | .050 | 1.02 | 1.14 | 1.27 |
| Corner Chamfer (others) | CH2 | .000 | .005 | .010 | 0.00 | 0.13 | 0.25 |
| Overall Width | E | .685 | .690 | .695 | 17.40 | 17.53 | 17.65 |
| Overall Length | D | .685 | .690 | .695 | 17.40 | 17.53 | 17.65 |
| Molded Package Width | E1 | .650 | .653 | .656 | 16.51 | 16.59 | 16.66 |
| Molded Package Length | D1 | .650 | .653 | .656 | 16.51 | 16.59 | 16.66 |
| Footprint Width | E2 | .590 | .620 | .630 | 14.99 | 15.75 | 16.00 |
| Footprint Length | D2 | .590 | .620 | .630 | 14.99 | 15.75 | 16.00 |
| Lead Thickness | c | .008 | .011 | .013 | 0.20 | 0.27 | 0.33 |
| Upper Lead Width | B1 | .026 | .029 | .032 | 0.66 | 0.74 | 0.81 |
| Lower Lead Width | B | .013 | .020 | .021 | 0.33 | 0.51 | 0.53 |
| Mold Draft Angle Top | α | 0 | 5 | 10 | 0 | 5 | 10 |
| Mold Draft Angle Bottom | β | 0 | 5 | 10 | 0 | 5 | 10 |

* Controlling Parameter
§ Significant Characteristic
Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MO-047
Drawing No. C04-048



Downloaded from Arrow.com.

**NOTES:**


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## APPENDIX A: REVISION HISTORY

| Version | Date | Revision Description |
|---|---|---|
| A | 7/98 | This is a new data sheet. However, the devices described in this data sheet are the upgrades to the devices found in the *PIC16C6X Data Sheet*, DS30234, and the *PIC16C7X Data Sheet*, DS30390. |
| B | 1/99 | Corrections to Version A data sheet for technical accuracy.<br>Added data:<br>• Operation of the SMP and CKE bits of the SSPSTAT register in $I^2C$ mode have been specified<br>• Frequency vs. VDD graphs for device operating area (in Electrical Specifications)<br>• Formula for calculating A/D acquisition time, TACQ (in the A/D section)<br>• Brief description of instructions<br>Removed data (see PICmicro™ Mid-Range MCU Family Reference Manual, DS33023, for additional data):<br>• USART Baud Rate Tables (formulas for calculating baud rate remain) |
| C | 12/00 | • Minor changes to text to clarify content<br>• Revised some DC specifications<br>• Included characteristic charts and graphs |
| D | 01/13 | • Added a note to every package drawing. |

## APPENDIX B: DEVICE DIFFERENCES

The differences between the devices in this data sheet are listed in Table B-1.

**TABLE B-1: DEVICE DIFFERENCES**

| Difference | PIC16C63A | PIC16C65B | PIC16C73B | PIC16C74B |
|---|---|---|---|---|
| A/D | no | no | 5 channels, 8 bits | 8 channels, 8 bits |
| Parallel Slave Port | no | yes | no | yes |
| Packages | 28-pin PDIP, 28-pin windowed CERDIP, 28-pin SOIC, 28-pin SSOP | 40-pin PDIP, 40-pin windowed CERDIP, 44-pin TQFP, 44-pin MQFP, 44-pin PLCC | 28-pin PDIP, 28-pin windowed CERDIP, 28-pin SOIC, 28-pin SSOP | 40-pin PDIP, 40-pin windowed CERDIP, 44-pin TQFP, 44-pin MQFP, 44-pin PLCC |



Downloaded from Arrow.com.

## APPENDIX C: DEVICE MIGRATIONS - PIC16C63/65A/73A/74A → PIC16C63A/65B/73B/74B

This document is intended to describe the functional differences and the electrical specification differences that are present when migrating from one device to the next. Table C-1 shows functional differences, while Table C-2 shows electrical and timing differences.

**Note:** Even though compatible devices are specified to be tested to the same electrical specification, the device characteristics may be different from each other (due to process differences). For systems that were designed to the device specifications, these process differences should not cause any issues in the application. For systems that did not tightly meet the electrical specifications, the process differences may cause the device to behave differently in the application.

**Note:** While there are no functional or electrical changes to the device oscillator specifications, the user should verify that the device oscillator starts and performs as expected. Adjusting the loading capacitor values and/or the oscillator mode may be required.

**TABLE C-1: FUNCTIONAL DIFFERENCES**

| No. | Module | Differences from PIC16C63/65A/73A/74A | H/W | S/W | Prog. |
|---|---|---|---|---|---|
| 1 | CCP | CCP Special Event Trigger clears Timer1. | — | ✔ | — |
| 2 | | Compare mode drives pin correctly. | — | ✔ | — |
| 3 | Timers | Writing to TMR1L does not affect TMR1H. | — | ✔ | — |
| 4 | | WDT/TMR0 prescaler assignment changes do not affect TMR0 count. | — | ✔ | — |
| 5 | SSP | TMR2 SPI clock synchronized to start of SPI Transmission. | — | ✔ | — |
| 6 | | Can now transmit multiple words in SPI mode. | — | ✔ | — |
| 7 | | Supports all four SPI modes. (Now uses SSP vs. BSSP module.) See SSP module in the PIC® Mid-Range MCU Family Reference Manual (DS33023). | — | ✔ | — |
| 8 | | $I^2C$ no longer generates ACK pulses when module is enabled. | — | ✔ | — |
| 9 | USART | Async receive errors due to BRGH setting corrected. | — | ✔ | — |
| 10 | A/D | $V_{REF} = V_{DD}$ when all inputs are configured as digital. This allows conversion of digital inputs. (A/D on PIC16C73X/74X only.) | — | ✔ | — |

H/W - Issues may exist with regard to the application circuits.
S/W - Issues may exist with regard to the user program.
Prog. - Issues may exist when writing the program to the controller.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**TABLE C-2: SPECIFICATION DIFFERENCES**

| Param No. | Symbol | Characteristic | | PIC16C63/65A/73A/74A | | | PIC16C63A/65B/73B/74B | | | Unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Min | Typ† | Max | Min | Typ† | Max | |
| **Core** | | | | | | | | | | |
| D001 | VDD | Supply Voltage | | 4.0 | — | 6.0 | 4.0 | — | **5.5** | V |
| D001A | | | | — | — | — | **VBOR(1)** | — | 5.5 | V |
| D005 | BVDD | Brown-out Reset Voltage | | 3.7 | 4.0 | 4.3 | **3.65** | — | **4.35** | V |
| D150* | VOD | Open-Drain High Voltage on RA4 | | — | — | 14.0 | - | — | **8.5** | V |
| **A/D Converter** | | | | | | | | | | |
| A20 | VREF | Reference voltage | | 3.0 | — | VDD + 0.3 | **2.5** | — | VDD + 0.3 | V |
| 131 | TCNV | Conversion time **(Note 2)** (not including S/H time) | | — | 9.5 **(Note 3)** | — | **11 (Note 4)** | — | **11 (Note 4)** | TAD |
| **SSP in SPI mode** | | | | | | | | | | |
| 71 | TscH | SCK input high time (Slave mode) | Continuous | TCY+20 | — | — | **1.25TCY + 30** | — | — | ns |
| 71A | | | Single Byte | | | | **40** | — | — | ns |
| 72 | TscL | SCK input low time (Slave mode) | Continuous | TCY+20 | — | — | **1.25TCY + 30** | — | — | ns |
| 72A | | | Single Byte | | | | **40** | — | — | ns |
| 73 | TdiV2scH TdiV2scL | Setup time of SDI data input to SCK edge | | 50 | — | — | **100** | — | — | ns |
| 73A **(Note 5)** | TB2B | Last clock edge of Byte1 to the 1st clock edge of Byte2 | | — | — | — | **1.5TCY + 40** | — | — | ns |
| 74 | TscH2diL TscL2diL | Hold time of SDI data input to SCK edge | | 50 | — | — | **100** | — | — | ns |
| 75 | TdoR | SDO data output rise time | PIC16CXX | — | 10 | 25 | — | 10 | 25 | ns |
| | | | PIC16LCXX | | | | — | **20** | **45** | ns |
| 78 | TscR | SCK output rise time (Master mode) | PIC16CXX | — | 10 | 25 | — | 10 | 25 | ns |
| | | | PIC16LCXX | | | | — | **20** | **45** | ns |
| 80 | TscH2doV TscL2doV | SDO data output valid after SCK edge | PIC16CXX | — | — | 50 | — | — | 50 | ns |
| | | | PIC16LCXX | | | | — | — | **100** | ns |
| 83 | TscH2ssH TscL2ssH | $\overline{SS}$ ↑ after SCK edge | | — | — | 50 | **1.5TCY + 40** | — | — | ns |

† 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 BOR is enabled, the device will operate until VDD drops below VBOR.
**2:** ADRES register may be read on the following TCY cycle.
**3:** This is the time that the actual conversion requires.
**4:** This is the time from when the GO/$\overline{DONE}$ bit is set, to when the conversion result appears in ADRES.
**5:** Specification 73A is only required if specifications 71A and 72A are used.



Downloaded from Arrow.com.

## APPENDIX D: MIGRATION FROM BASELINE TO MID-RANGE DEVICES

This section discusses how to migrate from a baseline device (i.e., PIC16C5X) to a mid-range device (i.e., PIC16CXXX).

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 (2 K 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 and 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 8-bit register.
15. “In-Circuit Serial Programming” (ICSP) 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.

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.


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## INDEX

### A


A/D
- ADCON0 Register ... 79
- ADCON1 Register ... 80
- Analog Input Model Block Diagram ... 82
- Analog-to-Digital Converter ... 79
- Block Diagram ... 81
- Configuring Analog Port Pins ... 83
- Configuring the Interrupt ... 81
- Configuring the Module ... 81
- Conversion Clock ... 83
- Conversions ... 83
- Converter Characteristics ... 137
- Effects of a RESET ... 83
- Faster Conversion - Lower Resolution Trade-off ... 83
- Internal Sampling Switch (Rss) Impedance ... 82
- Operation During SLEEP ... 83
- Sampling Requirements ... 82
- Source Impedance ... 82
- Timing Diagram ... 138
- Using the CCP Trigger ... 83

Absolute Maximum Ratings ... 113
ACK ... 60, 62
ADRES Register ... 17, 79
Application Notes
- AN552 (Implementing Wake-up on Key Strokes Using PIC16CXXX) ... 31
- AN556 (Table Reading Using PIC16CXX) ... 26
- AN578 (Use of the SSP Module in the $I^2C$ Multi-Master Environment) ... 55
- AN607, (Power-up Trouble Shooting) ... 89

Architecture
- Overview ... 9

Assembler
- MPASM® Assembler ... 107


### B


Baud Rate Formula ... 67
BF ... 56, 60
Block Diagrams
- A/D ... 81
- Analog Input Model ... 82
- Capture ... 51
- Compare ... 52
- $I^2C$ Mode ... 60
- On-Chip Reset Circuit ... 88
- PIC16C74 ... 10
- PIC16C74A ... 10
- PIC16C77 ... 10
- PORTC ... 33
- PORTD (In I/O Port Mode) ... 34
- PORTD and PORTE as a Parallel Slave Port ... 37
- PORTE (In I/O Port Mode) ... 35
- PWM ... 52
- RA4/T0CKI Pin ... 29
- RB3:RB0 Port Pins ... 31
- RB7:RB4 Port Pins ... 31
- SSP in $I^2C$ Mode ... 60
- SSP in SPI Mode ... 55
- Timer0/WDT Prescaler ... 39
- Timer2 ... 47
- USART Receive ... 70
- USART Transmit ... 68
- Watchdog Timer ... 96

$\overline{BOR}$ bit ... 25, 89
BRGH bit ... 67
Brown-out Reset (BOR)
- Timing Diagram ... 126

Buffer Full Status bit, BF ... 56


### C


C bit ... 19
Capture/Compare/PWM
- Capture
  - Block Diagram ... 51
  - CCP1CON Register ... 50
  - CCP1IF ... 51
  - Mode ... 51
  - Prescaler ... 51
- CCP Timer Resources ... 49
- Compare
  - Block Diagram ... 52
  - Mode ... 52
  - Software Interrupt Mode ... 52
  - Special Event Trigger ... 52
  - Special Trigger Output of CCP1 ... 52
  - Special Trigger Output of CCP2 ... 52
- Interaction of Two CCP Modules ... 49
- Section ... 49
- Special Event Trigger and A/D Conversions ... 52

Capture/Compare/PWM (CCP)
- PWM Block Diagram ... 52
- PWM Mode ... 52
- PWM, Example Frequencies/Resolutions ... 53
- Timing Diagram ... 128

CCP2IE bit ... 24
CCP2IF bit ... 24
CCPR1H Register ... 17, 49
CCPR1L Register ... 49
CCPR2H Register ... 17
CCPR2L Register ... 17
CCPxM0 bit ... 50
CCPxM1 bit ... 50
CCPxM2 bit ... 50
CCPxM3 bit ... 50
CCPxX bit ... 50
CCPxY bit ... 50
CKE ... 56
CKP ... 57
Clock Polarity Select bit, CKP ... 57
Clocking Scheme ... 14
Code Examples
- Call of a Subroutine in Page 1 from Page 0 ... 26
- Indirect Addressing ... 27
- Initializing PORTA ... 29

Code Protection ... 85, 98
Computed GOTO ... 26
Configuration Bits ... 85
CREN bit ... 66
$\overline{CS}$ pin ... 37




Downloaded from Arrow.com.

## D

D/$\overline{A}$ ..... 56
Data Memory
  Register File Map ..... 16
Data/Address bit, D/$\overline{A}$ ..... 56
DC bit ..... 19
Development Support ..... 5
Device Differences ..... 165
Direct Addressing ..... 27

## E

Electrical Characteristics ..... 113
Errata ..... 3

## F

FERR bit ..... 66
FSR Register ..... 17, 18, 27

## G

General Description ..... 5
GIE bit ..... 93

## I

I/O Ports
  PORTA ..... 29
  PORTB ..... 31
  PORTC ..... 33
  PORTD ..... 34, 37
  PORTE ..... 35
  Section ..... 29
$I^2C$
  Addressing ..... 61
  Block Diagram ..... 60
  $I^2C$ Operation ..... 60
  Master Mode ..... 64
  Mode ..... 60
  Mode Selection ..... 60
  Multi-Master Mode ..... 64
  Reception ..... 62
  Reception Timing Diagram ..... 62
  SCL and SDA pins ..... 60
  Slave Mode ..... 60
  Transmission ..... 63
$I^2C$ (SSP Module)
  Timing Diagram, Data ..... 134
  Timing Diagram, START/STOP Bits ..... 134
In-Circuit Serial Programming ..... 85, 98
INDF Register ..... 17, 18, 27
Indirect Addressing ..... 27
Instruction Cycle ..... 14
Instruction Flow/Pipelining ..... 14
Instruction Format ..... 99
Instruction Set
  ADDLW ..... 101
  ADDWF ..... 101
  ANDLW ..... 101
  ANDWF ..... 101
  BCF ..... 101
  BSF ..... 101
  BTFSC ..... 102
  BTFSS ..... 102
  CALL ..... 102
  CLRF ..... 102
  CLRW ..... 102
  CLRWDT ..... 102
  COMF ..... 103
  DECF ..... 103
  DECFSZ ..... 103
  GOTO ..... 103
  INCF ..... 103
  INCFSZ ..... 103
  IORLW ..... 104
  IORWF ..... 104
  MOVF ..... 104
  MOVLW ..... 104
  MOVWF ..... 104
  NOP ..... 104
  RETFIE ..... 105
  RETLW ..... 105
  RETURN ..... 105
  RLF ..... 105
  RRF ..... 105
  SLEEP ..... 105
  SUBLW ..... 106
  SUBWF ..... 106
  SWAPF ..... 106
  XORLW ..... 106
  XORWF ..... 106
  Section ..... 99
  Summary Table ..... 100
INT Interrupt ..... 94
INTCON Register ..... 21
INTEDG bit ..... 20, 94
Internal Sampling Switch (Rss) Impedance ..... 82
Interrupts ..... 85
  PORTB Change ..... 94
  RB7:RB4 Port Change ..... 31
  Section ..... 93
  TMR0 ..... 94
IRP bit ..... 19

## K

KEELOQ Evaluation and Programming Tools ..... 110

## L

Loading of PC ..... 26

## M

$\overline{MCLR}$ ..... 87, 90
Memory
  Data Memory ..... 15
  Program Memory ..... 15
  Program Memory Maps
    PIC16C73 ..... 15
    PIC16C73A ..... 15
    PIC16C74 ..... 15
    PIC16C74A ..... 15
  Register File Maps
    PIC16C73 ..... 16
    PIC16C73A ..... 16
    PIC16C74 ..... 16
    PIC16C74A ..... 16
    PIC16C76 ..... 16
    PIC16C77 ..... 16
MPLAB® Integrated Development
Environment Software ..... 107



© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## O


OERR bit ..... 66
OPCODE ..... 99
OPTION Register ..... 20
OSC Selection ..... 85
Oscillator
- HS ..... 86, 90
- LP ..... 86, 90
- RC ..... 86
- XT ..... 86, 90

Oscillator Configurations ..... 86
Output of TMR2 ..... 47


## P


P ..... 56
Packaging ..... 153
Paging, Program Memory ..... 26
Parallel Slave Port ..... 34, 37
Parallel Slave Port (PSP)
- Timing Diagram ..... 129

PCFG0 bit ..... 80
PCFG1 bit ..... 80
PCFG2 bit ..... 80
PCL Register ..... 17, 18, 26
PCLATH ..... 91
PCLATH Register ..... 17, 18, 26
PCON Register ..... 25, 89
PD bit ..... 19, 87
PICDEM™ 1 Low Cost PIC MCU Demonstration Board ..... 109
PICDEM™ 2 Low Cost PIC16CXX Demonstration Board ..... 109
PICDEM™ 3 Low Cost PIC16CXXX Demonstration Board ..... 110
PICSTART® Plus Entry Level Development System ..... 109
PIE1 Register ..... 22
PIE2 Register ..... 24
Pin Functions
- MCLR/VPP ..... 11, 12
- OSC1/CLKIN ..... 11, 12
- OSC2/CLKOUT ..... 11, 12
- RA0/AN0 ..... 11, 12
- RA1/AN1 ..... 11, 12
- RA2/AN2 ..... 11, 12
- RA3/AN3/VREF ..... 11, 12
- RA4/T0CKI ..... 11, 12
- RA5/AN4/SS ..... 11, 12
- RB0/INT ..... 11, 12
- RB1 ..... 11, 12
- RB2 ..... 11, 12
- RB3 ..... 11, 12
- RB4 ..... 11, 12
- RB5 ..... 11, 12
- RB6 ..... 11, 12
- RB7 ..... 11, 12
- RC0/T1OSO/T1CKI ..... 11, 13
- RC1/T1OSI/CCP2 ..... 11, 13
- RC2/CCP1 ..... 11, 13
- RC3/SCK/SCL ..... 11, 13
- RC4/SDI/SDA ..... 11, 13
- RC5/SDO ..... 11, 13
- RC6/TX/CK ..... 11, 13, 65–76
- RC7/RX/DT ..... 11, 13, 65–76
- RD0/PSP0 ..... 13
- RD1/PSP1 ..... 13
- RD2/PSP2 ..... 13
- RD3/PSP3 ..... 13
- RD4/PSP4 ..... 13
- RD5/PSP5 ..... 13
- RD6/PSP6 ..... 13
- RD7/PSP7 ..... 13
- RE0/RD/AN5 ..... 13
- RE1/WR/AN6 ..... 13
- RE2/CS/AN7 ..... 13
- VDD ..... 11, 13
- VSS ..... 11, 13

Pinout Descriptions
- PIC16C73 ..... 11
- PIC16C73A ..... 11
- PIC16C74 ..... 12
- PIC16C74A ..... 12
- PIC16C76 ..... 11
- PIC16C77 ..... 12

PIR1 Register ..... 23
PIR2 Register ..... 24
POP ..... 26
POR ..... 89
- Oscillator Start-up Timer (OST) ..... 85, 89
- Power Control Register (PCON) ..... 89
- Power-on Reset (POR) ..... 85, 89, 91
- Power-up Timer (PWRT) ..... 85
- Power-Up-Timer (PWRT) ..... 89
- TO ..... 87

POR bit ..... 25, 89
Port RB Interrupt ..... 94
PORTA ..... 91
PORTA Register ..... 17, 29
PORTB ..... 91
PORTB Register ..... 17, 31
PORTC ..... 91
PORTC Register ..... 17, 33
PORTD ..... 91
PORTD Register ..... 17, 34
PORTE ..... 91
PORTE Register ..... 17, 35
Power-down Mode (SLEEP) ..... 97
Power-on Reset (POR)
- Timing Diagram ..... 126

PR2 Register ..... 18, 47
PRO MATE® II Universal Programmer ..... 109
Product Identification System ..... 177
Program Memory
- Paging ..... 26

Program Memory Maps
- PIC16C73 ..... 15
- PIC16C73A ..... 15
- PIC16C74 ..... 15
- PIC16C74A ..... 15

Program Verification ..... 98
PS0 bit ..... 20
PS1 bit ..... 20
PS2 bit ..... 20
PSA bit ..... 20
PSPMODE bit ..... 34, 35, 37
PUSH ..... 26




Downloaded from Arrow.com.

## R

R/W ..... 56
R/W bit ..... 61, 62, 63
RBIF bit ..... 31, 94
RBPU bit ..... 20
RC Oscillator ..... 87, 90
RCSTA Register ..... 66
RD pin ..... 37
Read/Write bit Information, R/W ..... 56
Receive Overflow Indicator bit, SSPOV ..... 57
Register File ..... 15
Register File Map ..... 16
Registers
  Maps
    PIC16C73 ..... 16
    PIC16C73A ..... 16
    PIC16C74 ..... 16
    PIC16C74A ..... 16
  RESET Conditions ..... 90
  SSPSTAT ..... 56
  Summary ..... 17
RESET ..... 85, 87
  Timing Diagram ..... 126
RESET Conditions for Special Registers ..... 90
Revision History ..... 165
RP0 bit ..... 15, 19
RP1 bit ..... 19
RX9 bit ..... 66
RX9D bit ..... 66

## S

S ..... 56
SCL ..... 60
Serial Communication Interface (SCI) Module,
See USART
Services
  One-Time-Programmable (OTP) ..... 7
  Quick-Turnaround-Production (QTP) ..... 7
  Serialized Quick-Turnaround Production
  (SQTP) ..... 7
Slave Mode
  SCL ..... 60
  SDA ..... 60
SLEEP ..... 85, 87
SMP ..... 56
Software Simulator (MPLAB-SIM) ..... 108
SPBRG Register ..... 18
Special Features of the CPU ..... 85
Special Function Registers
  PIC16C73 ..... 17
  PIC16C73A ..... 17
  PIC16C74 ..... 17
  PIC16C74A ..... 17
Special Function Registers, Section ..... 16
SPEN bit ..... 66
SPI
  Block Diagram ..... 55
  Master Mode Timing ..... 58
  Serial Clock ..... 55
  Serial Data In ..... 55
  Serial Data Out ..... 55
  Slave Mode Timing ..... 59
  Slave Mode Timing Diagram ..... 58
  Slave Select ..... 55
  SSPCON ..... 57
  SSPSTAT ..... 56
SPI Clock Edge Select bit, CKE ..... 56
SPI Data Input Sample Phase Select bit, SMP ..... 56
SREN bit ..... 66
SSP
  Module Overview ..... 55
  Section ..... 55
  SSPCON ..... 57
  SSPSTAT ..... 56
SSPADD Register ..... 18
SSPBUF Register ..... 17
SSPCON ..... 57
SSPCON Register ..... 17
SSPEN ..... 57
SSPM3:SSPM0 ..... 57
SSPOV ..... 57, 60
SSPSTAT Register ..... 18, 56
Stack ..... 26
  Overflows ..... 26
  Underflow ..... 26
START bit, S ..... 56
STATUS Register ..... 19
STOP bit, P ..... 56
Synchronous Serial Port Enable bit, SSPEN ..... 57
Synchronous Serial Port Mode Select bits,
SSPM3:SSPM0 ..... 57
Synchronous Serial Port Module ..... 55
Synchronous Serial Port Status Register ..... 56

## T

T0CS bit ..... 20
T1CKPS0 bit ..... 43
T1CKPS1 bit ..... 43
T1CON Register ..... 43
T1OSCEN bit ..... 43
T1SYNC bit ..... 43
T2CKPS0 bit ..... 47
T2CKPS1 bit ..... 47
T2CON Register ..... 47
TAD ..... 83
Timer0
  RTCC ..... 91
  Timing Diagram ..... 127
Timer1
  Timing Diagram ..... 127
Timers
  Timer0
    External Clock ..... 40
    Interrupt ..... 39
    Prescaler ..... 40
    Prescaler Block Diagram ..... 39
    Section ..... 39
    T0CKI ..... 40
    T0IF ..... 94
    TMR0 Interrupt ..... 94
  Timer1
    Asynchronous Counter Mode ..... 45
    Capacitor Selection ..... 45
    Operation in Timer Mode ..... 44
    Oscillator ..... 45
    Prescaler ..... 45
    Resetting of Timer1 Registers ..... 45
    Resetting Timer1 using a CCP
    Trigger Output ..... 45
    Synchronized Counter Mode ..... 44
    T1CON ..... 43



© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

TMR1H ..... 45
TMR1L ..... 45
Timer2
Block Diagram ..... 47
Module ..... 47
Postscaler ..... 47
Prescaler ..... 47
T2CON ..... 47
Timing Diagrams
I2C Reception (7-bit Address) ..... 62
SPI Master Mode ..... 58
SPI Slave Mode (CKE = 1) ..... 59
SPI Slave Mode Timing (CKE = 0) ..... 58
USART Asynchronous Master Transmission ..... 69
USART Asynchronous Reception ..... 71
USART Synchronous Reception ..... 75
USART Synchronous Transmission ..... 73
Wake-up from SLEEP via Interrupt ..... 98
Timing Diagrams and Specifications ..... 124
A/D Conversion ..... 138
Brown-out Reset (BOR) ..... 126
Capture/Compare/PWM (CCP) ..... 128
CLKOUT and I/O ..... 125
External Clock ..... 124
I2C Bus Data ..... 134
I2C Bus START/STOP Bits ..... 134
Oscillator Start-up Timer (OST) ..... 126
Parallel Slave Port (PSP) ..... 129
Power-up Timer (PWRT) ..... 126
RESET ..... 126
Timer0 and Timer1 ..... 127
USART Synchronous Receive (Master/Slave) ..... 136
USART SynchronousTransmission (Master/Slave) .. 136
Watchdog Timer (WDT) ..... 126
TMR0 Register ..... 17
TMR1CS bit ..... 43
TMR1H Register ..... 17
TMR1L Register ..... 17
TMR1ON bit ..... 43
TMR2 Register ..... 17
TMR2ON bit ..... 47
TO bit ..... 19
TOUTPS0 bit ..... 47
TOUTPS1 bit ..... 47
TOUTPS2 bit ..... 47
TOUTPS3 bit ..... 47
TRISA Register ..... 18, 29
TRISB Register ..... 18, 31
TRISC Register ..... 18, 33
TRISD Register ..... 18, 34
TRISE Register ..... 18, 35, 36
TXSTA Register ..... 65


## U


UA ..... 56
Universal Synchronous Asynchronous Receiver Transmitter (USART) ..... 65
Update Address bit, UA ..... 56
USART
Asynchronous Mode ..... 68
Asynchronous Receiver ..... 70
Asynchronous Reception ..... 71
Asynchronous Transmitter ..... 68
Baud Rate Generator (BRG) ..... 67
Receive Block Diagram ..... 70
Sampling ..... 67
Synchronous Master Mode ..... 72
Timing Diagram, Synchronous Receive ..... 136
Timing Diagram, Synchronous Transmission ... 136
Synchronous Master Reception ..... 74
Synchronous Master Transmission ..... 72
Synchronous Slave Mode ..... 76
Synchronous Slave Reception ..... 76
Synchronous Slave Transmit ..... 76
Transmit Block Diagram ..... 68
UV Erasable Devices ..... 7


## W


Wake-up from SLEEP ..... 97
Watchdog Timer (WDT) ..... 85, 87, 90, 95
Timing Diagram ..... 126
WCOL ..... 57
WDT ..... 90
Block Diagram ..... 96
Period ..... 95
Programming Considerations ..... 96
Time-out ..... 91
WR pin ..... 37
Write Collision Detect bit, WCOL ..... 57
WWW, On-Line Support ..... 3


## Z


Z bit ..... 19




Downloaded from Arrow.com.

**NOTES:**


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

## CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the web site at: http://microchip.com/support**



Downloaded from Arrow.com.

## 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 (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

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:  Literature Number: DS30605D

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 document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document 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?


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

## 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 |
|---|---|---|---|---|
| Device | Frequency Range | Temperature Range | Package | Pattern |

| | | |
|---|---|---|
| Device | PIC16C6X(1), PIC16C6XT(2); VDD range 4.0V to 5.5V<br>PIC16LC6X(1), PIC16LC6XT(2); VDD range 2.5V to 5.5V<br>PIC16C7X(1), PIC16C7XT(2); VDD range 4.0V to 5.5V<br>PIC16LC7X(1), PIC16LC7XT(2); VDD range 2.5V to 5.5V | |
| Frequency Range | 04<br>20 | = 4 MHz<br>= 20 MHz |
| Temperature Range | blank<br>I<br>E | = 0°C to 70°C (Commercial)<br>= -40°C to +85°C (Industrial)<br>= -40°C to +125°C (Extended) |
| Package | JW<br>PQ<br>PT<br>SO<br>SP<br>P<br>L<br>SS | = Windowed CERDIP<br>= MQFP (Metric PQFP)<br>= TQFP (Thin Quad Flatpack)<br>= SOIC<br>= Skinny plastic dip<br>= PDIP<br>= PLCC<br>= SSOP |
| Pattern | QTP, SQTP, Code or Special Requirements (blank otherwise) | |

**Examples:**

a) PIC16C74B -04/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301.

b) PIC16LC63A - 04I/SO = Industrial temp., SOIC package, 200 kHz, Extended VDD limits.

c) PIC16C65B - 20I/P = Industrial temp., PDIP package, 20 MHz, normal VDD limits.

**Note 1:** C = CMOS
LC = Low Power CMOS
**2:** T = in tape and reel - SOIC, SSOP, PLCC, QFP, TQ and FP packages only.

* 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

**Data Sheets**
Products supported by a preliminary Data Sheet may 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. Your local Microchip sales office
2. The Microchip Worldwide Site (www.microchip.com)



Downloaded from Arrow.com.

**NOTES:**


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**NOTES:**



Downloaded from Arrow.com.

**NOTES:**


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**NOTES:**



Downloaded from Arrow.com.

**NOTES:**


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**NOTES:**



Downloaded from Arrow.com.

© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families 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 Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is 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 products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.** Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

**Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC$^{32}$ logo, rfPIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MTP, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

Analog-for-the-Digital Age, Application Maestro, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, SQI, Serial Quad I/O, Total Endurance, TSHARC, UniWinDriver, WiperLock, ZENA and Z-Scale are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

GestIC and ULPP are registered trademarks of Microchip Technology Germany II GmbH & Co. & KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 1998-2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

ISBN: 9781620769324

**QUALITY MANAGEMENT SYSTEM**
**CERTIFIED BY DNV**
**ISO/TS 16949**

*Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*



Downloaded from Arrow.com.

# 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:
http://www.microchip.com/support
Web Address:
www.microchip.com

**Atlanta**
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

**Boston**
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

**Chicago**
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

**Cleveland**
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

**Dallas**
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

**Detroit**
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

**Indianapolis**
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

**Los Angeles**
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

**Santa Clara**
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

**Toronto**
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

## ASIA/PACIFIC

**Asia Pacific Office**
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

**Australia - Sydney**
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

**China - Beijing**
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

**China - Chengdu**
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

**China - Chongqing**
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

**China - Hangzhou**
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

**China - Hong Kong SAR**
Tel: 852-2943-5100
Fax: 852-2401-3431

**China - Nanjing**
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

**China - Qingdao**
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

**China - Shanghai**
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

**China - Shenyang**
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

**China - Shenzhen**
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

**China - Wuhan**
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

**China - Xian**
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

**China - Xiamen**
Tel: 86-592-2388138
Fax: 86-592-2388130

**China - Zhuhai**
Tel: 86-756-3210040
Fax: 86-756-3210049

## ASIA/PACIFIC

**India - Bangalore**
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

**India - New Delhi**
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

**India - Pune**
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

**Japan - Osaka**
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

**Japan - Tokyo**
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

**Korea - Daegu**
Tel: 82-53-744-4301
Fax: 82-53-744-4302

**Korea - Seoul**
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

**Malaysia - Kuala Lumpur**
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

**Malaysia - Penang**
Tel: 60-4-227-8870
Fax: 60-4-227-4068

**Philippines - Manila**
Tel: 63-2-634-9065
Fax: 63-2-634-9069

**Singapore**
Tel: 65-6334-8870
Fax: 65-6334-8850

**Taiwan - Hsin Chu**
Tel: 886-3-5778-366
Fax: 886-3-5770-955

**Taiwan - Kaohsiung**
Tel: 886-7-213-7828
Fax: 886-7-330-9305

**Taiwan - Taipei**
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

**Thailand - Bangkok**
Tel: 66-2-694-1351
Fax: 66-2-694-1350

## EUROPE

**Austria - Wels**
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

**Denmark - Copenhagen**
Tel: 45-4450-2828
Fax: 45-4485-2829

**France - Paris**
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

**Germany - Munich**
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

**Italy - Milan**
Tel: 39-0331-742611
Fax: 39-0331-466781

**Netherlands - Drunen**
Tel: 31-416-690399
Fax: 31-416-690340

**Spain - Madrid**
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

**UK - Wokingham**
Tel: 44-118-921-5869
Fax: 44-118-921-5820

11/29/12


© 1998-2013 Microchip Technology Inc.

Downloaded from Arrow.com.