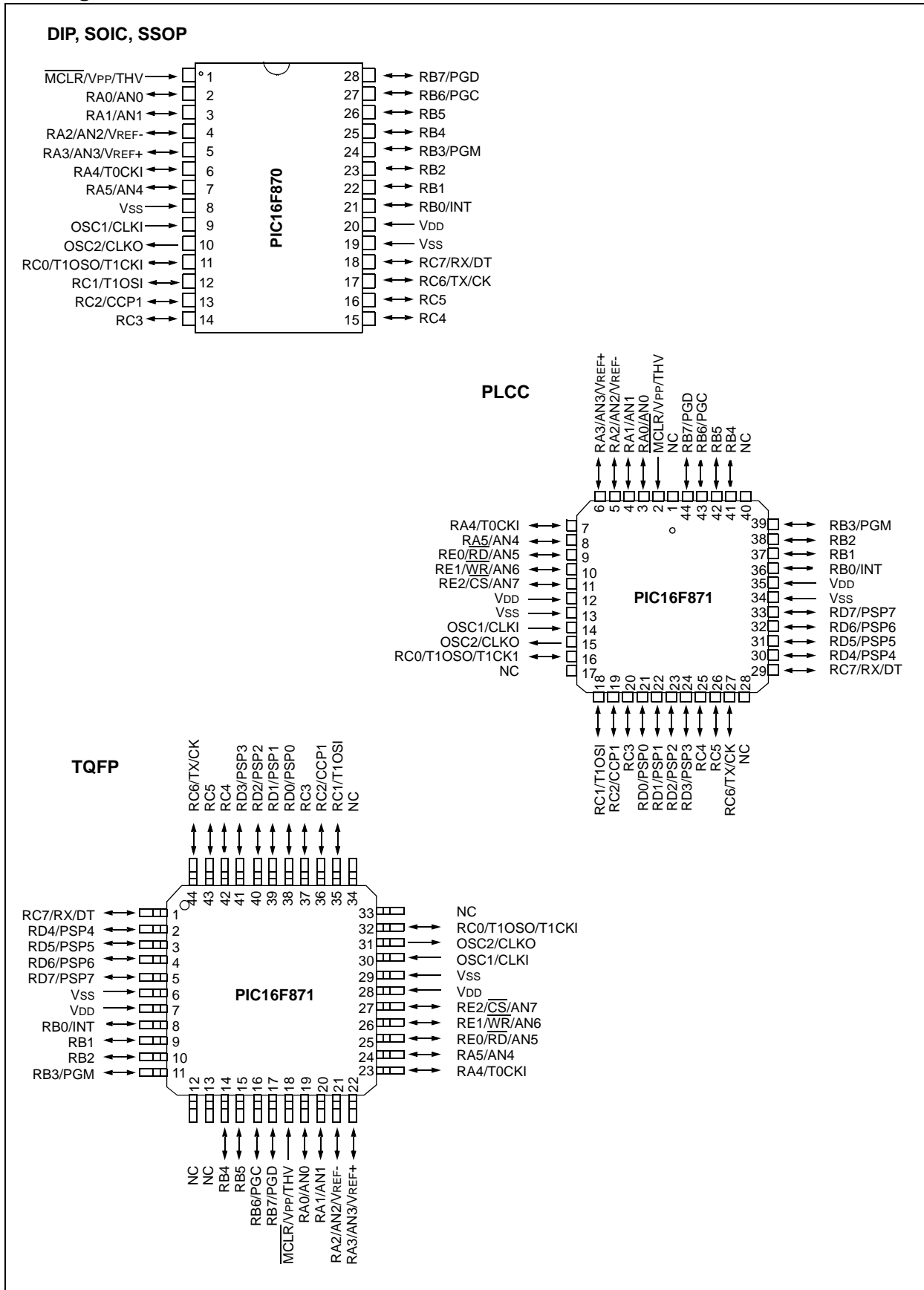


PIC16F870/871

Pin Diagrams



PIC16F870/871

| Key Features PICmicro™ Mid-Range MCU Family Reference Manual (DS33023) | PIC16F870 | PIC16F871 |
|---|----------------------|----------------------|
| Operating Frequency | DC - 20 MHz | DC - 20 MHz |
| RESETS (and Delays) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) |
| FLASH Program Memory (14-bit words) | 2K | 2K |
| Data Memory (bytes) | 128 | 128 |
| EEPROM Data Memory | 64 | 64 |
| Interrupts | 10 | 11 |
| I/O Ports | Ports A,B,C | Ports A,B,C,D,E |
| Timers | 3 | 3 |
| Capture/Compare/PWM modules | 1 | 1 |
| Serial Communications | USART | USART |
| Parallel Communications | — | PSP |
| 10-bit Analog-to-Digital Module | 5 input channels | 8 input channels |
| Instruction Set | 35 Instructions | 35 Instructions |

PIC16F870/871

Table of Contents

| | | |
|------|---|-----|
| 1.0 | Device Overview | 5 |
| 2.0 | Memory Organization | 11 |
| 3.0 | Data EEPROM and Flash Program Memory | 27 |
| 4.0 | I/O Ports | 33 |
| 5.0 | Timer0 Module | 45 |
| 6.0 | Timer1 Module | 49 |
| 7.0 | Timer2 Module | 53 |
| 8.0 | Capture/Compare/PWM Modules | 55 |
| 9.0 | Addressable Universal Synchronous Asynchronous Receiver Transmitter (USART) | 61 |
| 10.0 | Analog-to-Digital (A/D) Converter Module | 79 |
| 11.0 | Special Features of the CPU | 87 |
| 12.0 | Instruction Set Summary | 103 |
| 13.0 | Development Support | 111 |
| 14.0 | Electrical Characteristics | 117 |
| 15.0 | DC and AC Characteristics Graphs and Tables | 137 |
| 16.0 | Packaging Information | 149 |
| | Appendix A: Revision History | 157 |
| | Appendix B: Device Differences | 157 |
| | Appendix C: Conversion Considerations | 158 |
| | Appendix D: Migration from Mid-Range to Enhanced Devices | 158 |
| | Appendix E: Migration from High-End to Enhanced Devices | 159 |
| | Index | 161 |
| | On-Line Support | 167 |
| | Systems Information and Upgrade Hot Line | 167 |
| | Reader Response | 168 |
| | PIC16F870/871 Product Identification System | 169 |

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.

PIC16F870/871

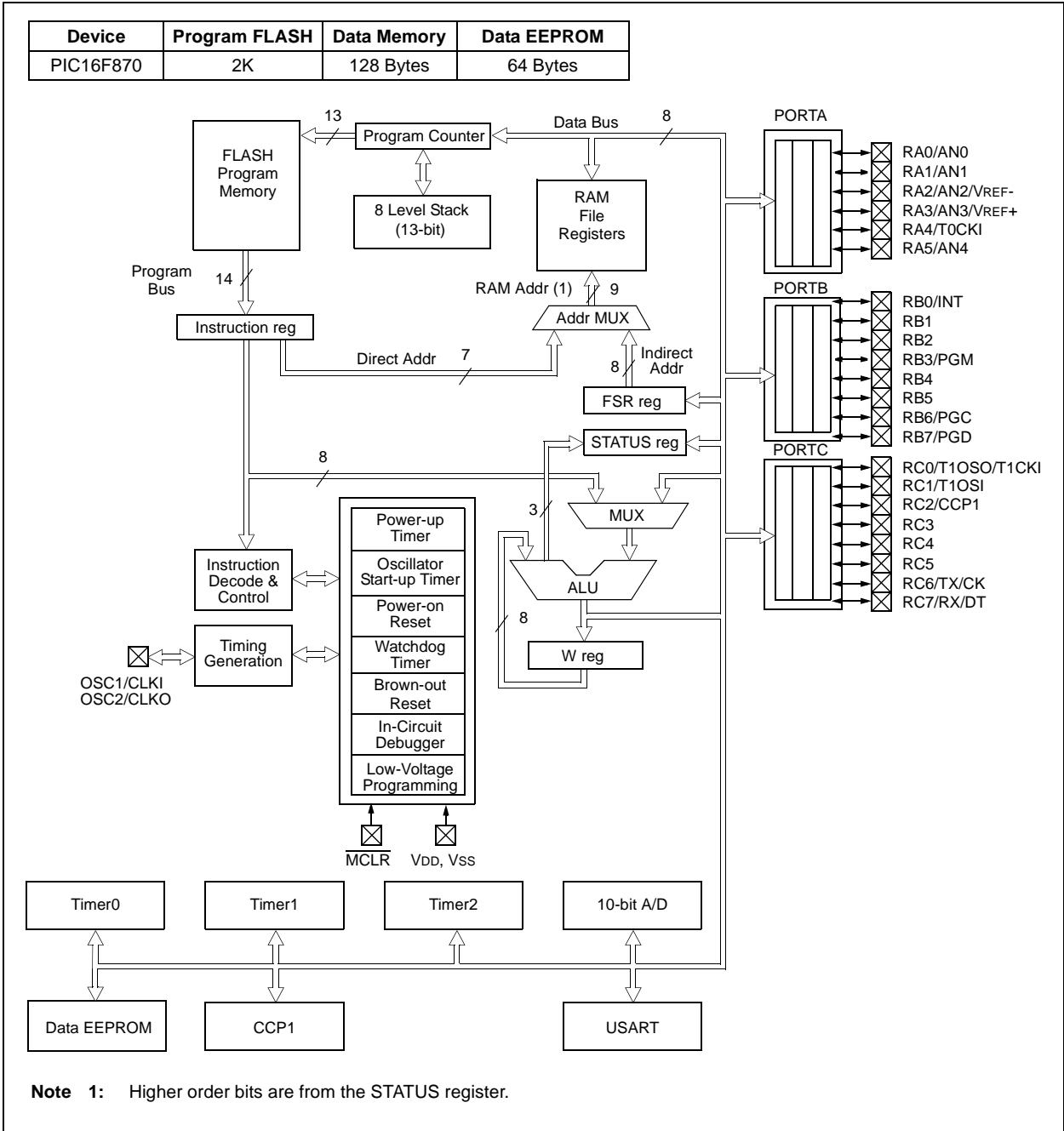
1.0 DEVICE OVERVIEW

This document contains device specific information. Additional information may be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

There are two devices (PIC16F870 and PIC16F871) covered by this data sheet. The PIC16F870 device comes in a 28-pin package and the PIC16F871 device comes in a 40-pin package. The 28-pin device does not have a Parallel Slave Port implemented.

The following two figures are device block diagrams sorted by pin number: 28-pin for Figure 1-1 and 40-pin for Figure 1-2. The 28-pin and 40-pin pinouts are listed in Table 1-1 and Table 1-2, respectively.

FIGURE 1-1: PIC16F870 BLOCK DIAGRAM



PIC16F870/871

FIGURE 1-2: PIC16F871 BLOCK DIAGRAM

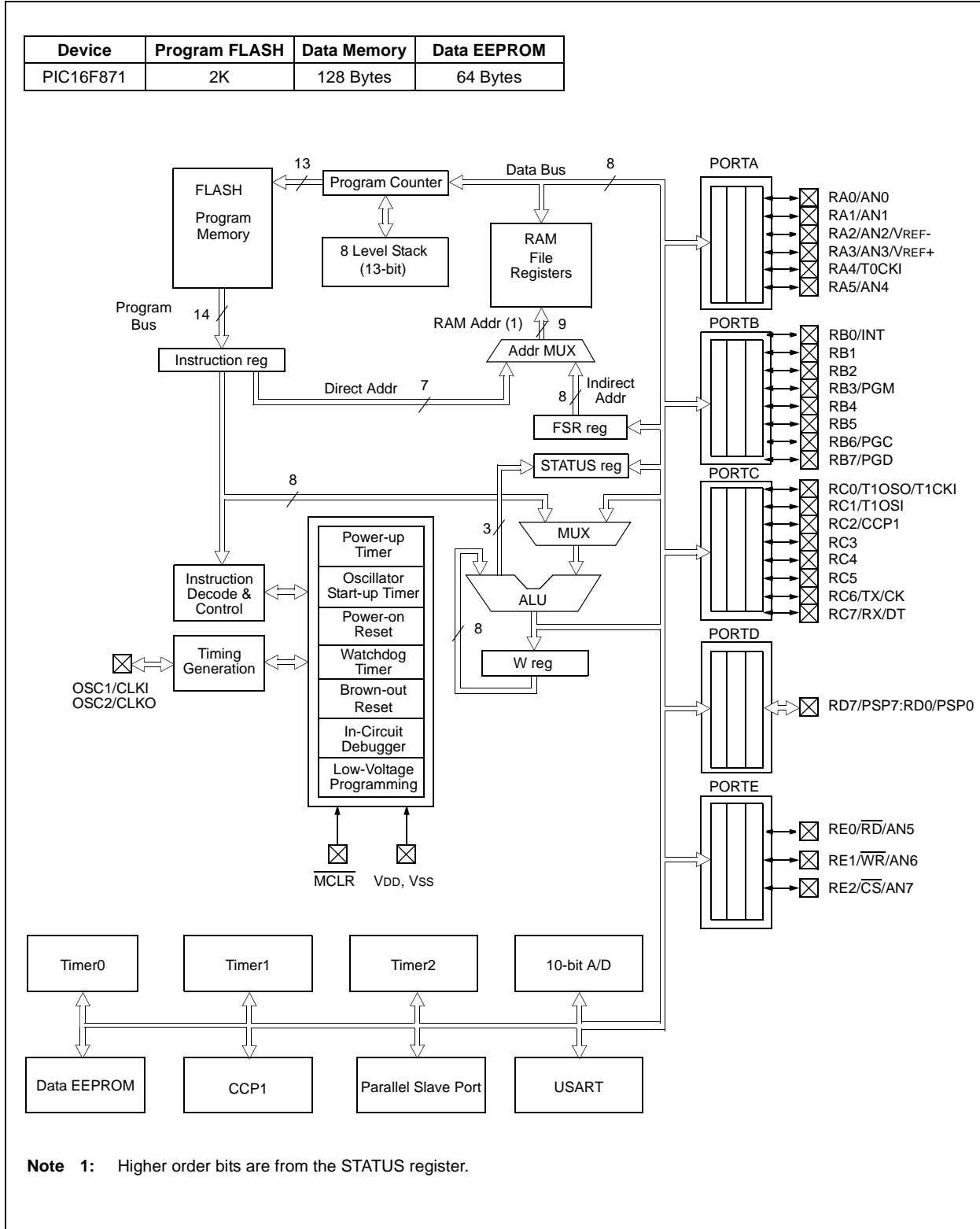


TABLE 1-1: PIC16F870 PINOUT DESCRIPTION

| Pin Name | DIP Pin# | SOIC Pin# | I/O/P Type | Buffer Type | Description |
|-----------------|----------|-----------|------------|------------------------|---|
| OSC1/CLKI | 9 | 9 | I | ST/CMOS ⁽³⁾ | Oscillator crystal input/external clock source input. |
| OSC2/CLKO | 10 | 10 | O | — | Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, the OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate. |
| MCLR/VPP/THV | 1 | 1 | I/P | ST | Master Clear (Reset) input or programming voltage input or High Voltage Test mode control. This pin is an active low RESET to the device. |
| RA0/AN0 | 2 | 2 | I/O | TTL | PORTA is a bi-directional I/O port. RA0 can also be analog input 0. RA1 can also be analog input 1. RA2 can also be analog input 2 or negative analog reference voltage. RA3 can also be analog input 3 or positive analog reference voltage. RA4 can also be the clock input to the Timer0 module. Output is open drain type. RA5 can also be analog input 4. |
| RA1/AN1 | 3 | 3 | I/O | TTL | |
| RA2/AN2/VREF- | 4 | 4 | I/O | TTL | |
| RA3/AN3/VREF+ | 5 | 5 | I/O | TTL | |
| RA4/T0CKI | 6 | 6 | I/O | ST/OD | |
| RA5/AN4 | 7 | 7 | I/O | TTL | |
| RB0/INT | 21 | 21 | I/O | TTL/ST ⁽¹⁾ | PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin. RB3 can also be the low voltage programming input. Interrupt-on-change pin. Interrupt-on-change pin. Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock. Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data. |
| RB1 | 22 | 22 | I/O | TTL | |
| RB2 | 23 | 23 | I/O | TTL | |
| RB3/PGM | 24 | 24 | I/O | TTL/ST ⁽¹⁾ | |
| RB4 | 25 | 25 | I/O | TTL | |
| RB5 | 26 | 26 | I/O | TTL | |
| RB6/PGC | 27 | 27 | I/O | TTL/ST ⁽²⁾ | |
| RB7/PGD | 28 | 28 | I/O | TTL/ST ⁽²⁾ | |
| RC0/T1OSO/T1CKI | 11 | 11 | I/O | ST | PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or Timer1 clock input. RC1 can also be the Timer1 oscillator input. RC2 can also be the Capture1 input/Compare1 output/PWM1 output. RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. RC7 can also be the USART Asynchronous Receive or Synchronous Data. |
| RC1/T1OSI | 12 | 12 | I/O | ST | |
| RC2/CCP1 | 13 | 13 | I/O | ST | |
| RC3 | 14 | 14 | I/O | ST | |
| RC4 | 15 | 15 | I/O | ST | |
| RC5 | 16 | 16 | I/O | ST | |
| RC6/TX/CK | 17 | 17 | I/O | ST | |
| RC7/RX/DT | 18 | 18 | I/O | ST | |
| 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
 OD = Open Drain — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note**
- 1: This buffer is a Schmitt Trigger input when configured as the external interrupt or LVP mode.
 - 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.

PIC16F870/871

TABLE 1-2: PIC16F871 PINOUT DESCRIPTION

| Pin Name | DIP Pin# | PLCC Pin# | QFP Pin# | I/O/P Type | Buffer Type | Description |
|-----------------|----------|-----------|----------|------------|------------------------|--|
| OSC1/CLKI | 13 | 14 | 30 | I | ST/CMOS ⁽⁴⁾ | Oscillator crystal input/external clock source input. |
| OSC2/CLKO | 14 | 15 | 31 | O | — | Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate. |
| MCLR/VPP/THV | 1 | 2 | 18 | I/P | ST | Master Clear (Reset) input or programming voltage input or High Voltage Test mode control. This pin is an active low RESET to the device. |
| RA0/AN0 | 2 | 3 | 19 | I/O | TTL | <p>PORTA is a bi-directional I/O port.</p> <p>RA0 can also be analog input 0.</p> <p>RA1 can also be analog input 1.</p> <p>RA2 can also be analog input 2 or negative analog reference voltage.</p> <p>RA3 can also be analog input 3 or positive analog reference voltage.</p> <p>RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type.</p> <p>RA5 can also be analog input 4.</p> |
| RA1/AN1 | 3 | 4 | 20 | I/O | TTL | |
| RA2/AN2/VREF- | 4 | 5 | 21 | I/O | TTL | |
| RA3/AN3/VREF+ | 5 | 6 | 22 | I/O | TTL | |
| RA4/T0CKI | 6 | 7 | 23 | I/O | ST | |
| RA5/AN4 | 7 | 8 | 24 | I/O | TTL | |
| RB0/INT | 33 | 36 | 8 | I/O | TTL/ST ⁽¹⁾ | <p>PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.</p> <p>RB0 can also be the external interrupt pin.</p> <p>RB3 can also be the low voltage programming input.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.</p> |
| RB1 | 34 | 37 | 9 | I/O | TTL | |
| RB2 | 35 | 38 | 10 | I/O | TTL | |
| RB3/PGM | 36 | 39 | 11 | I/O | TTL/ST ⁽¹⁾ | |
| RB4 | 37 | 41 | 14 | I/O | TTL | |
| RB5 | 38 | 42 | 15 | I/O | TTL | |
| RB6/PGC | 39 | 43 | 16 | I/O | TTL/ST ⁽²⁾ | |
| RB7/PGD | 40 | 44 | 17 | I/O | TTL/ST ⁽²⁾ | |
| RC0/T1OSO/T1CKI | 15 | 16 | 32 | I/O | ST | <p>PORTC is a bi-directional I/O port.</p> <p>RC0 can also be the Timer1 oscillator output or a Timer1 clock input.</p> <p>RC1 can also be the Timer1 oscillator input.</p> <p>RC2 can also be the Capture1 input/Compare1 output/PWM1 output.</p> <p>RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.</p> <p>RC7 can also be the USART Asynchronous Receive or Synchronous Data.</p> |
| RC1/T1OSI | 16 | 18 | 35 | I/O | ST | |
| RC2/CCP1 | 17 | 19 | 36 | I/O | ST | |
| RC3 | 18 | 20 | 37 | I/O | ST | |
| RC4 | 23 | 25 | 42 | I/O | ST | |
| RC5 | 24 | 26 | 43 | I/O | ST | |
| RC6/TX/CK | 25 | 27 | 44 | I/O | ST | |
| RC7/RX/DT | 26 | 29 | 1 | I/O | ST | |

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

- Note**
- 1: This buffer is a Schmitt Trigger input when configured as an external interrupt or LVP mode.
 - 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
 - 3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
 - 4: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

TABLE 1-2: PIC16F871 PINOUT DESCRIPTION (CONTINUED)

| Pin Name | DIP Pin# | PLCC Pin# | QFP Pin# | I/O/P Type | Buffer Type | Description |
|----------------------------------|----------|------------|-------------|------------|-----------------------|--|
| RD0/PSP0 | 19 | 21 | 38 | I/O | ST/TTL ⁽³⁾ | PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus. |
| 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⁾ | |
| RE0/ $\overline{\text{RD}}$ /AN5 | 8 | 9 | 25 | I/O | ST/TTL ⁽³⁾ | PORTE is a bi-directional I/O port. RE0 can also be read control for the parallel slave port, or analog input 5. RE1 can also be write control for the parallel slave port, or analog input 6. RE2 can also be select control for the parallel slave port, or analog input 7. |
| RE1/ $\overline{\text{WR}}$ /AN6 | 9 | 10 | 26 | I/O | ST/TTL ⁽³⁾ | |
| RE2/ $\overline{\text{CS}}$ /AN7 | 10 | 11 | 27 | I/O | ST/TTL ⁽³⁾ | |
| Vss | 12,31 | 13,34 | 6,29 | P | — | Ground reference for logic and I/O pins. |
| Vdd | 11,32 | 12,35 | 7,28 | P | — | Positive supply for logic and I/O pins. |
| NC | — | 1,17,28,40 | 12,13,33,34 | | — | These pins are not internally connected. These pins should be left unconnected. |

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

- Note**
- 1: This buffer is a Schmitt Trigger input when configured as an external interrupt or LVP mode.
 - 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.

PIC16F870/871

NOTES:

2.0 MEMORY ORGANIZATION

The PIC16F870/871 devices have three memory blocks. The Program Memory and Data Memory have separate buses, so that concurrent access can occur, and is detailed in this section. The EEPROM data memory block is detailed in Section 3.0.

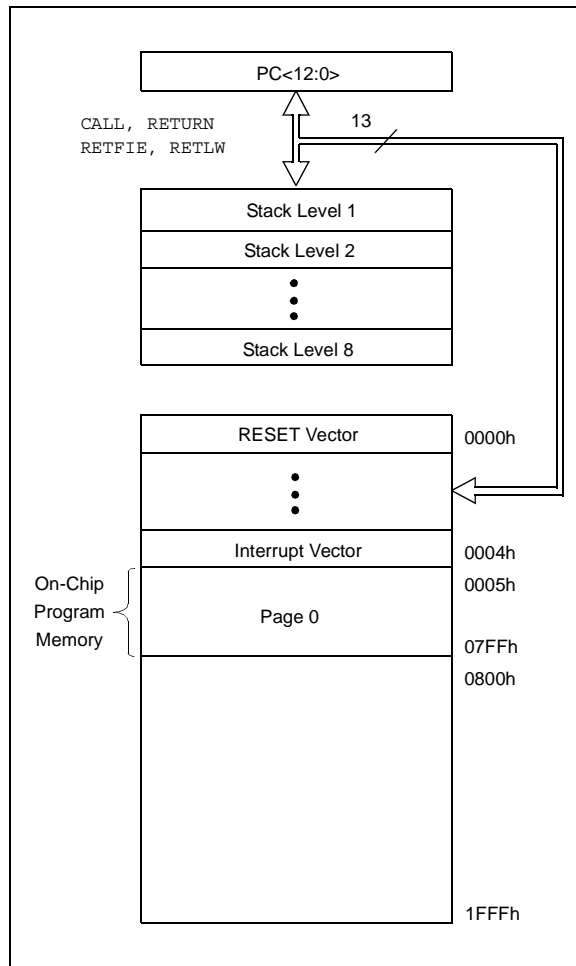
Additional information on device memory may be found in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

2.1 Program Memory Organization

The PIC16F870/871 devices have a 13-bit program counter capable of addressing an 8K x 14 program memory space. The PIC16F870/871 devices have 2K x 14 words of FLASH program memory. Accessing a location above the physically implemented address will cause a wraparound.

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

FIGURE 2-1: PIC16F870/871 PROGRAM MEMORY MAP AND STACK



2.2 Data Memory Organization

The data memory is partitioned into multiple banks which contain the General Purpose Registers and the Special Function Registers. Bits RP1 (STATUS<6>) and RP0 (STATUS<5>) are the bank select bits.

| RP<1:0> | Bank |
|---------|------|
| 00 | 0 |
| 01 | 1 |
| 10 | 2 |
| 11 | 3 |

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

Note: EEPROM Data Memory description can be found in Section 3.0 of this Data Sheet.

2.2.1 GENERAL PURPOSE REGISTER FILE

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

PIC16F870/871

FIGURE 2-2: PIC16F870/871 REGISTER FILE MAP

| File Address | | File Address | | File Address | | File Address | | |
|--------------------------------------|-----|--------------------------------------|------|-------------------------------|------|-------------------------------|---------------------|------|
| Indirect addr. ^(*) | 00h | Indirect addr. ^(*) | 80h | Indirect addr. ^(*) | 100h | Indirect addr. ^(*) | 180h | |
| TMR0 | 01h | OPTION_REG | 81h | TMR0 | 101h | OPTION_REG | 181h | |
| PCL | 02h | PCL | 82h | PCL | 102h | PCL | 182h | |
| STATUS | 03h | STATUS | 83h | STATUS | 103h | STATUS | 183h | |
| FSR | 04h | FSR | 84h | FSR | 104h | FSR | 184h | |
| PORTA | 05h | TRISA | 85h | | 105h | | 185h | |
| PORTB | 06h | TRISB | 86h | PORTB | 106h | TRISB | 186h | |
| PORTC | 07h | TRISC | 87h | | 107h | | 187h | |
| PORTD ⁽²⁾ | 08h | TRISD ⁽²⁾ | 88h | | 108h | | 188h | |
| PORTE ⁽²⁾ | 09h | TRISE ⁽²⁾ | 89h | | 109h | | 189h | |
| PCLATH | 0Ah | PCLATH | 8Ah | PCLATH | 10Ah | PCLATH | 18Ah | |
| INTCON | 0Bh | INTCON | 8Bh | INTCON | 10Bh | INTCON | 18Bh | |
| PIR1 | 0Ch | PIE1 | 8Ch | EEDATA | 10Ch | EECON1 | 18Ch | |
| PIR2 | 0Dh | PIE2 | 8Dh | EEADR | 10Dh | EECON2 | 18Dh | |
| TMR1L | 0Eh | PCON | 8Eh | EEDATH | 10Eh | Reserved ⁽¹⁾ | 18Eh | |
| TMR1H | 0Fh | | 8Fh | EEADRH | 10Fh | Reserved ⁽¹⁾ | 18Fh | |
| T1CON | 10h | | 90h | | 110h | | 190h | |
| TMR2 | 11h | | 91h | | | | | |
| T2CON | 12h | PR2 | 92h | | | | | |
| | 13h | | 93h | | | | | |
| | 14h | | 94h | | | | | |
| CCPR1L | 15h | | 95h | | | | | |
| CCPR1H | 16h | | 96h | | | | | |
| CCP1CON | 17h | | 97h | | | | | |
| RCSTA | 18h | TXSTA | 98h | | | | | |
| TXREG | 19h | SPBRG | 99h | | | | | |
| RCREG | 1Ah | | 9Ah | | | | | |
| | 1Bh | | 9Bh | | | | | |
| | 1Ch | | 9Ch | | | | | |
| | 1Dh | | 9Dh | | | | | |
| ADRESH | 1Eh | ADRESL | 9Eh | | | | | |
| ADCON0 | 1Fh | ADCON1 | 9Fh | | | | | |
| General Purpose Register 96 Bytes | 20h | General Purpose Register 32 Bytes | A0h | accesses 20h-7Fh | 120h | accesses A0h - BFh | 1A0h | |
| | | | BFh | | | | | 1BFh |
| | | | C0h | | | | | 1C0h |
| | | | EFh | | | | accesses 70h-7Fh | 16Fh |
| | F0h | 170h | 1F0h | | | | | |
| | 7Fh | | FFh | | 17Fh | | 1FFh | |

Bank 0 Bank 1 Bank 2 Bank 3

■ Unimplemented data memory locations, read as '0'.
* Not a physical register.

Note 1: These registers are reserved; maintain these registers clear.
Note 2: These registers are not implemented on the PIC16F870.

PIC16F870/871

2.2.2 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. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral feature section.

TABLE 2-1: 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 ⁽²⁾ | |
|----------------------|---------|--|---------|---|--|-----------------|-----------------------|---------|---------|--------------------|--|-----------|
| Bank 0 | | | | | | | | | | | | |
| 00h ⁽⁴⁾ | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | | 0000 0000 | 0000 0000 |
| 01h | TMR0 | Timer0 Module's Register | | | | | | | | | xxxx xxxx | uuuu uuuu |
| 02h ⁽⁴⁾ | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | | 0000 0000 | 0000 0000 |
| 03h ⁽⁴⁾ | STATUS | IRP | RP1 | RP0 | \overline{TO} | \overline{PD} | Z | DC | C | 0001 1xxx | 000q quuu | |
| 04h ⁽⁴⁾ | FSR | Indirect Data Memory Address Pointer | | | | | | | | | xxxx xxxx | uuuu uuuu |
| 05h | PORTA | — | — | PORTA Data Latch when written: PORTA pins when read | | | | | | | --0x 0000 | --0u 0000 |
| 06h | PORTB | PORTB Data Latch when written: PORTB pins when read | | | | | | | | | xxxx xxxx | uuuu uuuu |
| 07h | PORTC | PORTC Data Latch when written: PORTC pins when read | | | | | | | | | xxxx xxxx | uuuu uuuu |
| 08h ⁽⁵⁾ | PORTD | PORTD Data Latch when written: PORTD pins when read | | | | | | | | | xxxx xxxx | uuuu uuuu |
| 09h ⁽⁵⁾ | PORTE | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu | |
| 0Ah ^(1,4) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | | ---0 0000 | ---0 0000 |
| 0Bh ⁽⁴⁾ | INTCON | GIE | PEIE | TOIE | INTE | RBIE | T0IF | INTF | RBF | 0000 000x | 0000 000u | |
| 0Ch | PIR1 | PSPIF ⁽³⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 | |
| 0Dh | PIR2 | — | — | — | EEIF | — | — | — | — | ---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 | 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 | — | Unimplemented | | | | | | | | | — | — |
| 14h | — | Unimplemented | | | | | | | | | — | — |
| 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 | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x | |
| 19h | TXREG | USART Transmit Data Register | | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | USART Receive Data Register | | | | | | | | | 0000 0000 | 0000 0000 |
| 1Bh | — | Unimplemented | | | | | | | | | — | — |
| 1Ch | — | Unimplemented | | | | | | | | | — | — |
| 1Dh | — | Unimplemented | | | | | | | | | — | — |
| 1Eh | ADRESH | A/D Result Register High Byte | | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/ \overline{DONE} | — | ADON | 0000 00-0 | 0000 00-0 | |

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
- 2:** Other (non Power-up) Resets include external RESET through \overline{MCLR} and Watchdog Timer Reset.
- 3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.
- 4:** These registers can be addressed from any bank.
- 5:** PORTD, PORTE, TRISD and TRISE are not physically implemented on the 28-pin devices, read as '0'.

PIC16F870/871

TABLE 2-1: 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 ⁽²⁾ | |
|----------------------|------------|--|--------|-------------------------------|--|-----------------|---------------------------|------------------|--------|-----------------------|--|-----------|
| Bank 1 | | | | | | | | | | | | |
| 80h ⁽⁴⁾ | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | | 0000 0000 | 0000 0000 |
| 81h | OPTION_REG | RBP \overline{U} | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 | |
| 82h ⁽⁴⁾ | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | | 0000 0000 | 0000 0000 |
| 83h ⁽⁴⁾ | STATUS | IRP | RP1 | RP0 | \overline{TO} | \overline{PD} | Z | DC | C | 0001 1xxx | 000q quuu | |
| 84h ⁽⁴⁾ | FSR | Indirect Data Memory Address Pointer | | | | | | | | | xxxx xxxx | uuuu uuuu |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | —11 1111 | —11 1111 | |
| 86h | TRISB | PORTB Data Direction Register | | | | | | | | | 1111 1111 | 1111 1111 |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | | 1111 1111 | 1111 1111 |
| 88h ⁽⁵⁾ | TRISD | PORTD Data Direction Register | | | | | | | | | 1111 1111 | 1111 1111 |
| 89h ⁽⁵⁾ | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction Bits | | | | 0000 -111 | 0000 -111 |
| 8Ah ^(1,4) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | | ---0 0000 | ---0 0000 |
| 8Bh ⁽⁴⁾ | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u | |
| 8Ch | PIE1 | PSPIE ⁽³⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 | |
| 8Dh | PIE2 | — | — | — | EEIE | — | — | — | — | ---0 ---- | ---0 ---- | |
| 8Eh | PCON | — | — | — | — | — | — | \overline{POR} | BOR | ---- -gq | ---- -uu | |
| 8Fh | — | Unimplemented | | | | | | | | | — | — |
| 90h | — | Unimplemented | | | | | | | | | — | — |
| 91h | — | Unimplemented | | | | | | | | | — | — |
| 92h | PR2 | Timer2 Period Register | | | | | | | | | 1111 1111 | 1111 1111 |
| 93h | — | Unimplemented | | | | | | | | | — | — |
| 94h | — | Unimplemented | | | | | | | | | — | — |
| 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 | ADRESL | A/D Result Register Low Byte | | | | | | | | | xxxx xxxx | uuuu uuuu |
| 9Fh | ADCON1 | ADFM | — | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 | 0--- 0000 | 0--- 0000 | |

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
- 2:** Other (non Power-up) Resets include external RESET through \overline{MCLR} and Watchdog Timer Reset.
- 3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.
- 4:** These registers can be addressed from any bank.
- 5:** PORTD, PORTE, TRISD and TRISE are not physically implemented on the 28-pin devices, read as '0'.

TABLE 2-1: 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 ⁽²⁾ |
|-----------------------|------------|--|--------|--------------------------------|--|-------|-------|-------|-----------|--------------------|--|
| Bank 2 | | | | | | | | | | | |
| 100h ⁽⁴⁾ | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 0000 0000 |
| 101h | TMR0 | Timer0 Module's Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 102h ⁽⁴⁾ | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 103h ⁽⁴⁾ | STATUS | IRP | RP1 | RP0 | TO | PD | Z | DC | C | 0001 1xxx | 000q quuu |
| 104h ⁽⁴⁾ | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 105h | — | Unimplemented | | | | | | | | — | — |
| 106h | PORTB | PORTB Data Latch when written: PORTB pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 107h | — | Unimplemented | | | | | | | | — | — |
| 108h | — | Unimplemented | | | | | | | | — | — |
| 109h | — | Unimplemented | | | | | | | | — | — |
| 10Ah ^(1,4) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 10Bh ⁽⁴⁾ | INTCON | GIE | PEIE | TOIE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 10Ch | EEDATA | EEPROM Data Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10Dh | EEADR | EEPROM Address Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10Eh | EEDATH | — | — | EEPROM Data Register High Byte | | | | | xxxx xxxx | uuuu uuuu | |
| 10Fh | EEADRH | — | — | — | EEPROM Address Register High Byte | | | | | xxxx xxxx | uuuu uuuu |
| Bank 3 | | | | | | | | | | | |
| 180h ⁽⁴⁾ | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 0000 0000 |
| 181h | OPTION_REG | RBPU | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 182h ⁽⁴⁾ | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 183h ⁽⁴⁾ | STATUS | IRP | RP1 | RP0 | TO | PD | Z | DC | C | 0001 1xxx | 000q quuu |
| 184h ⁽⁴⁾ | FSR | Indirect Data Memory Address Pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 185h | — | Unimplemented | | | | | | | | — | — |
| 186h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 187h | — | Unimplemented | | | | | | | | — | — |
| 188h | — | Unimplemented | | | | | | | | — | — |
| 189h | — | Unimplemented | | | | | | | | — | — |
| 18Ah ^(1,4) | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 18Bh ⁽⁴⁾ | INTCON | GIE | PEIE | TOIE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 18Ch | EECON1 | EEPGD | — | — | — | WRERR | WREN | WR | RD | x--- x000 | x--- u000 |
| 18Dh | EECON2 | EEPROM Control Register2 (not a physical register) | | | | | | | | ---- ---- | ---- ---- |
| 18Eh | — | Reserved maintain clear | | | | | | | | 0000 0000 | 0000 0000 |
| 18Fh | — | Reserved maintain clear | | | | | | | | 0000 0000 | 0000 0000 |

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.
- 2:** Other (non Power-up) Resets include external RESET through $\overline{\text{MCLR}}$ and Watchdog Timer Reset.
- 3:** Bits PSPIE and PSPIF are reserved on the 28-pin devices; always maintain these bits clear.
- 4:** These registers can be addressed from any bank.
- 5:** PORTD, PORTE, TRISD and TRISE are not physically implemented on the 28-pin devices, read as '0'.

PIC16F870/871

2.2.2.1 STATUS Register

The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

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

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

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

Note 1: The C and DC bits operate as a borrow and digit borrow bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

REGISTER 2-1: STATUS REGISTER (ADDRESS: 03h, 83h, 103h, 183h)

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

bit 7

bit 0

- bit 7-6 **IRP:** Register Bank Select bit (used for indirect addressing)
 1 = Bank 2, 3 (100h - 1FFh)
 0 = Bank 0, 1 (00h - FFh)
- bit 6-5 **RP1:RP0:** Register Bank Select bits (used for direct addressing)
 11 = Bank 3 (180h - 1FFh)
 10 = Bank 2 (100h - 17Fh)
 11 = Bank 1 (80h - FFh)
 10 = 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/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
 (for borrow, the polarity is reversed)
 1 = A carry-out from the 4th low order bit of the result occurred
 0 = No carry-out from the 4th low order bit of the result
- bit 0 **C:** Carry/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
 1 = A carry-out from the Most Significant bit of the result occurred
 0 = No carry-out from the Most Significant bit of the result occurred

Note: For 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 |

2.2.2.2 OPTION_REG Register

The OPTION_REG register is a readable and writable register, which contains various control bits to configure the TMR0 prescaler/WDT postscaler (single assignable register known also as the 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 2-2: OPTION_REG REGISTER (ADDRESS: 81h,181h)

| | | | | | | | | |
|-------|--------------------------------|---------------|-------------|-------------|------------|------------|------------|------------|
| | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
| | RBP\bar{U} | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
| bit 7 | | | | | | | | bit 0 |

- bit 7 **RBP \bar{U}** : 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 (CLKO)
- 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

PIC16F870/871

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

REGISTER 2-3: INTCON REGISTER (ADDRESS: 0Bh, 8Bh, 10Bh, 18Bh)

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-x | |
| GIE | PEIE | TOIE | INTE | RBIE | TOIF | 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 **TOIE:** 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 **TOIF:** TMR0 Overflow Interrupt Flag bit
 1 = TMR0 register has overflowed (must be cleared in software)
 0 = TMR0 register did not overflow
- bit 1 **INTF:** RB0/INT External Interrupt Flag bit
 1 = The RB0/INT external interrupt occurred (must be cleared in software)
 0 = The RB0/INT external interrupt did not occur
- bit 0 **RBIF:** RB Port Change Interrupt Flag bit
 1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
 0 = None of the RB7:RB4 pins have changed state

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 |

2.2.2.4 PIE1 Register

The PIE1 register contains the individual enable bits for the peripheral interrupts.

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

REGISTER 2-4: PIE1 REGISTER (ADDRESS: 8Ch)

| | | | | | | | | |
|-------|----------------------|-------|-------|-------|-----|--------|--------|--------|
| | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
| | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | 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:** A/D Converter Interrupt Enable bit
 1 = Enables the A/D converter interrupt
 0 = Disables the A/D converter 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 **Unimplemented:** Read as '0'
- 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: PSPIE is reserved on the PIC16F870; 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 |

PIC16F870/871

2.2.2.5 PIR1 Register

The PIR1 register contains the individual flag bits for the peripheral interrupts.

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

REGISTER 2-5: PIR1 REGISTER (ADDRESS: 0Ch)

| R/W-0 | R/W-0 | R-0 | R-0 | U-0 | R/W-0 | R/W-0 | R/W-0 |
|----------------------|-------|------|------|-----|--------|--------|--------|
| PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | 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**: A/D Converter Interrupt Flag bit
1 = An A/D conversion completed
0 = The A/D conversion is not complete
- bit 5 **RCIF**: USART Receive Interrupt Flag bit
1 = The USART receive buffer is full
0 = The USART receive buffer is empty
- bit 4 **TXIF**: USART Transmit Interrupt Flag bit
1 = The USART transmit buffer is empty
0 = The USART transmit buffer is full
- bit 3 **Unimplemented**: Read as '0'
- 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: PSPIF is reserved on the PIC16F870; always maintain this bit clear.

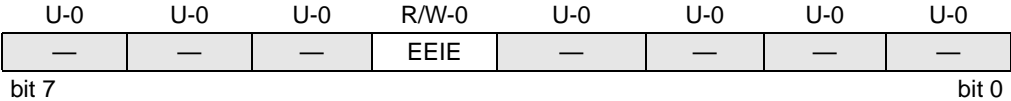
Legend:

| | | |
|--------------------|------------------|--|
| 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 |

2.2.2.6 PIE2 Register

The PIE2 register contains the individual enable bit for the EEPROM write operation interrupt.

REGISTER 2-6: PIE2 REGISTER (ADDRESS: 8Dh)



- bit 7-5 **Unimplemented:** Read as '0'
- bit 4 **EEIE:** EEPROM Write Operation Interrupt Enable bit
1 = Enable EE write interrupt
0 = Disable EE write interrupt
- bit 3-0 **Unimplemented:** Read as '0'

| | | | |
|--------------------|------------------|------------------------------------|--------------------|
| 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 |

PIC16F870/871

2.2.2.7 PIR2 Register

The PIR2 register contains the flag bit for the EEPROM write operation interrupt.

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

REGISTER 2-7: PIR2 REGISTER (ADDRESS: 0Dh)

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

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit
1 = The write operation completed (must be cleared in software)
0 = The write operation is not complete or has not been started

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

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

2.2.2.8 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 BOR is clear, indicating a brown-out has occurred. The BOR status bit is a don't care and is not predictable if the brown-out circuit is disabled (by clearing the BOREN bit in the configuration word).

REGISTER 2-8: PCON REGISTER (ADDRESS: 8Eh)

| | | | | | | | |
|-------|-----|-----|-----|-----|-----|-------|-------------------------|
| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-1 |
| — | — | — | — | — | — | POR | $\overline{\text{BOR}}$ |
| bit 7 | | | | | | bit 0 | |

- bit 7-2 **Unimplemented:** Read as '0'
- bit 1 **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:

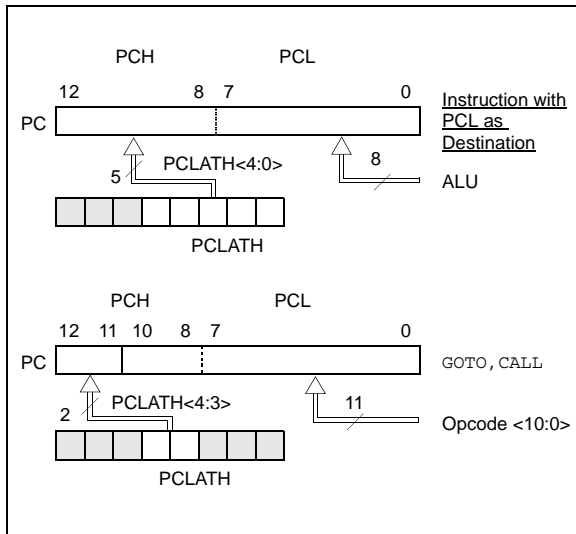
| | | |
|--------------------|------------------|--|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| - n = Value at POR | '1' = Bit is set | '0' = Bit is cleared x = Bit is unknown |

PIC16F870/871

2.3 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 2-3 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 2-3: LOADING OF PC IN DIFFERENT SITUATIONS



2.3.1 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).

2.3.2 STACK

The PIC16FXXX 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.

2.4 Program Memory Paging

The PIC16FXXX architecture is capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide 11 bits of the address, which allows branches within any 2K program memory page. Therefore, the 8K words of program memory are broken into four pages. Since the PIC16F872 has only 2K words of program memory or one page, additional code is not required to ensure that the correct page is selected before a CALL or GOTO instruction is executed. The PCLATH<4:3> bits should always be maintained as zeros. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is popped off the stack. Manipulation of the PCLATH is not required for the return instructions.

2.5 Indirect Addressing, INDF and FSR Registers

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

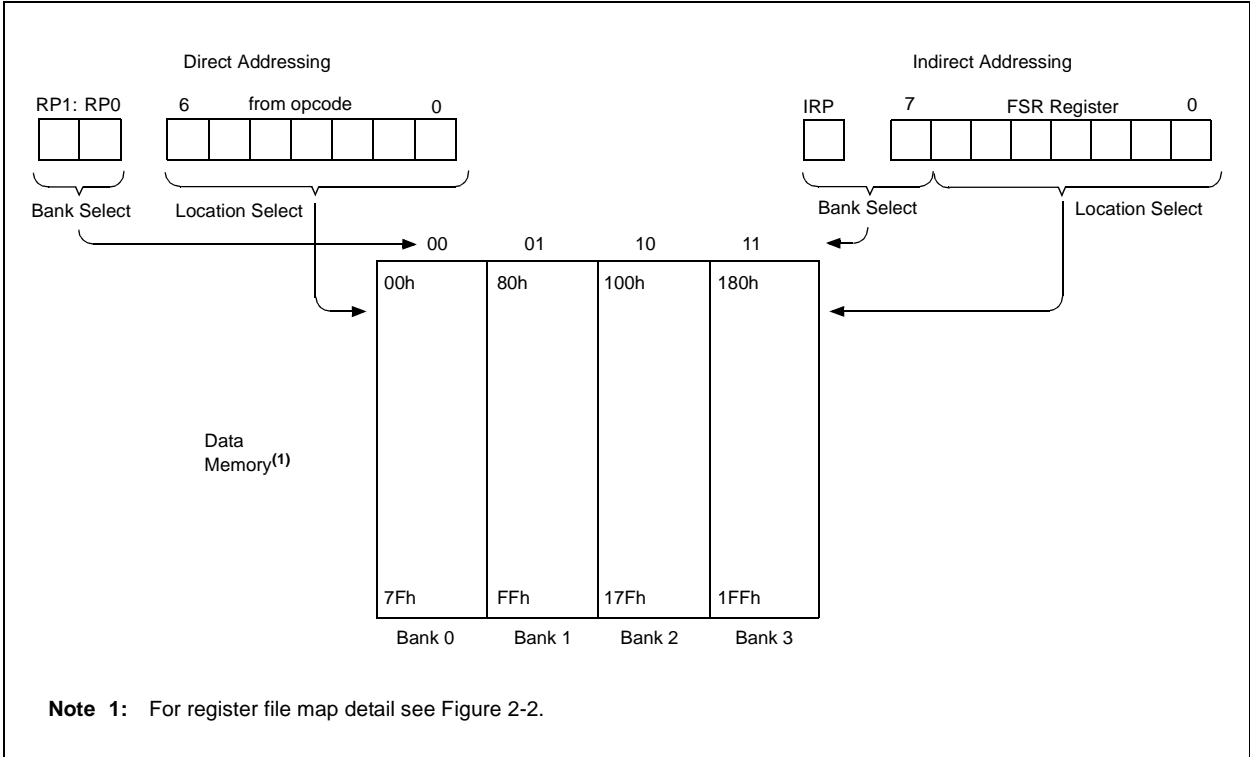
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select register, FSR. Reading the INDF register itself indirectly (FSR = 0) will read 00h. Writing to the INDF register indirectly results in a no 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 2-4.

A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-1.

EXAMPLE 2-1: INDIRECT ADDRESSING

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

FIGURE 2-4: DIRECT/INDIRECT ADDRESSING



PIC16F870/871

NOTES:

3.0 DATA EEPROM AND FLASH PROGRAM MEMORY

The Data EEPROM and FLASH Program Memory are readable and writable during normal operation over the entire VDD range. A bulk erase operation may not be issued from user code (which includes removing code protection). The data memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers (SFR).

There are six SFRs used to read and write the program and data EEPROM memory. These registers are:

- EECON1
- EECON2
- EEDATA
- EEDATH
- EEADR
- EEADRH

The EEPROM data memory allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. The registers EEDATH and EEADRH are not used for data EEPROM access. The PIC16F870/871 devices have 64 bytes of data EEPROM with an address range from 0h to 3Fh.

The EEPROM data memory is rated for high erase/write cycles. The write time is controlled by an on-chip timer. The write time will vary with voltage and temperature, as well as from chip-to-chip. Please refer to the specifications for exact limits.

The program memory allows word reads and writes. Program memory access allows for checksum calculation and calibration table storage. A byte or word write automatically erases the location and writes the new data (erase before write). Writing to program memory will cease operation until the write is complete. The program memory cannot be accessed during the write, therefore code cannot execute. During the write operation, the oscillator continues to clock the peripherals, and therefore, they continue to operate. Interrupt events will be detected and essentially “queued” until the write is completed. When the write completes, the next instruction in the pipeline is executed and the branch to the interrupt vector address will occur.

When interfacing to the program memory block, the EEDATH:EEDATA registers form a two-byte word, which holds the 14-bit data for read/write. The EEADRH:EEADR registers form a two-byte word, which holds the 13-bit address of the FLASH location being accessed. The PIC16F870/871 devices have 2K words of program FLASH with an address range from 0h to 7FFh. The unused upper bits in both the EEDATH and EEDATA registers all read as '0's.

The value written to program memory does not need to be a valid instruction. Therefore, up to 14-bit numbers can be stored in memory for use as calibration parameters, serial numbers, packed 7-bit ASCII, etc. Executing a program memory location containing data that forms an invalid instruction results in a NOP.

3.1 EEADR

The address registers can address up to a maximum of 256 bytes of data EEPROM or up to a maximum of 8K words of program FLASH. However, the PIC16F870/871 have 64 bytes of data EEPROM and 2K words of program FLASH.

When selecting a program address value, the MSByte of the address is written to the EEADRH register and the LSByte is written to the EEADR register. When selecting a data address value, only the LSByte of the address is written to the EEADR register.

On the PIC16F870/871 devices, the upper two bits of the EEADR must always be cleared to prevent inadvertent access to the wrong location in data EEPROM. This also applies to the program memory. The upper five MSbits of EEADRH must always be clear during program FLASH access.

3.2 EECON1 and EECON2 Registers

The EECON1 register is the control register for configuring and initiating the access. The EECON2 register is not a physically implemented register, but is used exclusively in the memory write sequence to prevent inadvertent writes.

There are many bits used to control the read and write operations to EEPROM data and FLASH program memory. The EEPGD bit determines if the access will be a program or data memory access. When clear, any subsequent operations will work on the EEPROM data memory. When set, all subsequent operations will operate in the program memory.

Read operations only use one additional bit, RD, which initiates the read operation from the desired memory location. Once this bit is set, the value of the desired memory location will be available in the data registers. This bit cannot be cleared by firmware. It is automatically cleared at the end of the read operation. For EEPROM data memory reads, the data will be available in the EEDATA register in the very next instruction cycle after the RD bit is set. For program memory reads, the data will be loaded into the EEDATH:EEDATA registers, following the second instruction after the RD bit is set.

PIC16F870/871

Write operations have two control bits, WR and WREN, and two status bits, WRERR and EEIF. The WREN bit is used to enable or disable the write operation. When WREN is clear, the write operation will be disabled. Therefore, the WREN bit must be set before executing a write operation. The WR bit is used to initiate the write operation. It also is automatically cleared at the end of the write operation. The interrupt flag EEIF is used to determine when the memory write completes. This flag must be cleared in software before setting the WR bit. For EEPROM data memory, once the WREN bit and the WR bit have been set, the desired memory address in EEADR will be erased, followed by a write of the data in EEDATA. This operation takes place in parallel with the microcontroller continuing to execute normally. When the write is complete, the EEIF flag bit will be set. For program memory, once the WREN bit and the WR bit have been set, the microcontroller will cease to execute instructions.

The desired memory location pointed to by EEADRH:EEADR will be erased. Then, the data value in EEDATH:EEDATA will be programmed. When complete, the EEIF flag bit will be set and the microcontroller will continue to execute code.

The WRERR bit is used to indicate when the PIC16F870/871 devices have been reset during a write operation. WRERR should be cleared after Power-on Reset. Thereafter, it should be checked on any other RESET. The WRERR bit is set when a write operation is interrupted by a $\overline{\text{MCLR}}$ Reset, or a WDT Time-out Reset, during normal operation. In these situations, following a RESET, the user should check the WRERR bit and rewrite the memory location, if set. The contents of the data registers, address registers and EEPGD bit are not affected by either $\overline{\text{MCLR}}$ Reset, or WDT Time-out Reset, during normal operation.

REGISTER 3-1: EECON1 REGISTER (ADDRESS: 18Ch)

| R/W-x | U-0 | U-0 | U-0 | R/W-x | R/W-0 | R/S-0 | R/S-0 |
|-------|-----|-----|-----|-------|-------|-------|-------|
| EEPGD | — | — | — | WRERR | WREN | WR | RD |
| bit 7 | | | | bit 0 | | | |

- bit 7 **EEPGD:** Program/Data EEPROM Select bit
 1 = Accesses program memory
 0 = Accesses data memory
 (This bit cannot be changed while a read or write operation is in progress.)
- bit 6-4 **Unimplemented:** Read as '0'
- bit 3 **WRERR:** EEPROM Error Flag bit
 1 = A write operation is prematurely terminated (any $\overline{\text{MCLR}}$ Reset or any WDT Reset during normal operation)
 0 = The write operation completed
- bit 2 **WREN:** EEPROM Write Enable bit
 1 = Allows write cycles
 0 = Inhibits write to the EEPROM
- bit 1 **WR:** Write Control bit
 1 = Initiates a write cycle. (The bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)
 0 = Write cycle to the EEPROM is complete
- bit 0 **RD:** Read Control bit
 1 = Initiates an EEPROM read. (RD is cleared in hardware. The RD bit can only be set (not cleared) in software.)
 0 = Does not initiate an EEPROM read

| | | | |
|--------------------|------------------|------------------------------------|--------------------|
| 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 |

3.3 Reading the EEPROM Data Memory

Reading EEPROM data memory only requires that the desired address to access be written to the EEADR register and clear the EEPGD bit. After the RD bit is set, data will be available in the EEDATA register on the very next instruction cycle. EEDATA will hold this value until another read operation is initiated or until it is written by firmware.

The steps to reading the EEPROM data memory are:

1. Write the address to EEADR. Make sure that the address is not larger than the memory size of the PIC16F870/871 devices.
2. Clear the EEPGD bit to point to EEPROM data memory.
3. Set the RD bit to start the read operation.
4. Read the data from the EEDATA register.

EXAMPLE 3-1: EEPROM DATA READ

```
BSF STATUS, RP1 ;
BCF STATUS, RP0 ;Bank 2
MOVF ADDR, W ;Write address
MOVWF EEADR ;to read from
BSF STATUS, RP0 ;Bank 3
BCF EECON1, EEPGD ;Point to Data memory
BSF EECON1, RD ;Start read operation
BCF STATUS, RP0 ;Bank 2
MOVF EEDATA, W ;W = EEDATA
```

3.4 Writing to the EEPROM Data Memory

There are many steps in writing to the EEPROM data memory. Both address and data values must be written to the SFRs. The EEPGD bit must be cleared, and the WREN bit must be set, to enable writes. The WREN bit should be kept clear at all times, except when writing to the EEPROM data. The WR bit can only be set if the WREN bit was set in a previous operation (i.e., they both cannot be set in the same operation). The WREN bit should then be cleared by firmware after the write. Clearing the WREN bit before the write actually completes will not terminate the write in progress.

Writes to EEPROM data memory must also be prefaced with a special sequence of instructions that prevent inadvertent write operations. This is a sequence of five instructions that must be executed without interruptions. The firmware should verify that a write is not in progress before starting another cycle.

The steps to write to EEPROM data memory are:

1. If step 10 is not implemented, check the WR bit to see if a write is in progress.
2. Write the address to EEADR. Make sure that the address is not larger than the memory size of the PIC16F870/871 devices.
3. Write the 8-bit data value to be programmed in the EEDATA register.
4. Clear the EEPGD bit to point to EEPROM data memory.
5. Set the WREN bit to enable program operations.
6. Disable interrupts (if enabled).
7. Execute the special five instruction sequence:
 - Write 55h to EECON2 in two steps (first to W, then to EECON2)
 - Write AAh to EECON2 in two steps (first to W, then to EECON2)
 - Set the WR bit
8. Enable interrupts (if using interrupts).
9. Clear the WREN bit to disable program operations.
10. At the completion of the write cycle, the WR bit is cleared and the EEIF interrupt flag bit is set. (EEIF must be cleared by firmware.) If step 1 is not implemented, then firmware should check for EEIF to be set, or WR to clear, to indicate the end of the program cycle.

EXAMPLE 3-2: EEPROM DATA WRITE

```
BSF STATUS, RP1 ;
BSF STATUS, RP0 ;Bank 3
BTFSC EECON1, WR ;Wait for
GOTO $-1 ;write to finish
BCF STATUS, RP0 ;Bank 2
MOVF ADDR, W ;Address to
MOVWF EEADR ;write to
MOVF VALUE, W ;Data to
MOVWF EEDATA ;write
BSF STATUS, RP0 ;Bank 3
BCF EECON1, EEPGD ;Point to Data memory
BSF EECON1, WREN ;Enable writes
;Only disable interrupts
BCF INTCON, GIE ;if already enabled,
;otherwise discard
MOVLW 0x55 ;Write 55h to
MOVWF EECON2 ;EECON2
MOVLW 0xAA ;Write AAh to
MOVWF EECON2 ;EECON2
BSF EECON1, WR ;Start write operation
;Only enable interrupts
BSF INTCON, GIE ;if using interrupts,
;otherwise discard
BCF EECON1, WREN ;Disable writes
```

PIC16F870/871

3.5 Reading the FLASH Program Memory

Reading FLASH program memory is much like that of EEPROM data memory, only two NOP instructions must be inserted after the RD bit is set. These two instruction cycles that the NOP instructions execute, will be used by the microcontroller to read the data out of program memory and insert the value into the EEDATH:EEDATA registers. Data will be available following the second NOP instruction. EEDATH and EEDATA will hold their value until another read operation is initiated, or until they are written by firmware.

The steps to reading the FLASH program memory are:

1. Write the address to EEADRH:EEADR. Make sure that the address is not larger than the memory size of the PIC16F870/871 devices.
2. Set the EEPGD bit to point to FLASH program memory.
3. Set the RD bit to start the read operation.
4. Execute two NOP instructions to allow the microcontroller to read out of program memory.
5. Read the data from the EEDATH:EEDATA registers.

EXAMPLE 3-3: FLASH PROGRAM READ

```
BSF    STATUS, RP1    ;
BCF    STATUS, RP0    ;Bank 2
MOVF   ADDR, W        ;Write the
MOVWF  EEADR          ;address bytes
MOVF   ADDR, W        ;for the desired
MOVWF  EEADRH        ;address to read
BSF    STATUS, RP0    ;Bank 3
BSF    EECON1, EEPGD ;Point to Program memory
BSF    EECON1, RD     ;Start read operation
NOP    ;Required two NOPs
NOP    ;
BCF    STATUS, RP0    ;Bank 2
MOVF   EEDATA, W     ;DATAH = EEDATA
MOVWF  DATAH        ;
MOVF   EEDATH, W     ;DATAH = EEDATH
MOVWF  DATAH        ;
```

3.6 Writing to the FLASH Program Memory

Writing to FLASH program memory is unique, in that the microcontroller does not execute instructions while programming is taking place. The oscillator continues to run and all peripherals continue to operate and queue interrupts, if enabled. Once the write operation completes (specification D133), the processor begins executing code from where it left off. The other important difference when writing to FLASH program memory is that the WRT configuration bit, when clear, prevents any writes to program memory (see Table 3-1).

Just like EEPROM data memory, there are many steps in writing to the FLASH program memory. Both address and data values must be written to the SFRs. The EEPGD bit must be set, and the WREN bit must be set to enable writes. The WREN bit should be kept clear at all times, except when writing to the FLASH program memory. The WR bit can only be set if the WREN bit was set in a previous operation (i.e., they both cannot be set in the same operation). The WREN bit should then be cleared by firmware after the write. Clearing the WREN bit before the write actually completes will not terminate the write in progress.

Writes to program memory must also be prefaced with a special sequence of instructions that prevent inadvertent write operations. This is a sequence of five instructions that must be executed without interruption for each byte written. These instructions must then be followed by two NOP instructions to allow the microcontroller to setup for the write operation. Once the write is complete, the execution of instructions starts with the instruction after the second NOP.

The steps to write to program memory are:

1. Write the address to EEADRH:EEADR. Make sure that the address is not larger than the memory size of the PIC16F870/871 devices.
2. Write the 14-bit data value to be programmed in the EEDATH:EEDATA registers.
3. Set the EEPGD bit to point to FLASH program memory.
4. Set the WREN bit to enable program operations.
5. Disable interrupts (if enabled).
6. Execute the special five instruction sequence:
 - Write 55h to EECON2 in two steps (first to W, then to EECON2)
 - Write AAh to EECON2 in two steps (first to W, then to EECON2)
 - Set the WR bit
7. Execute two NOP instructions to allow the microcontroller to setup for write operation.
8. Enable interrupts (if using interrupts).
9. Clear the WREN bit to disable program operations.

At the completion of the write cycle, the WR bit is cleared and the EEIF interrupt flag bit is set. (EEIF must be cleared by firmware.) Since the microcontroller does not execute instructions during the write cycle, the firmware does not necessarily have to check either EEIF, or WR, to determine if the write had finished.

EXAMPLE 3-4: FLASH PROGRAM WRITE

```
BSF    STATUS, RP1    ;
BCF    STATUS, RP0    ;Bank 2
MOVF   ADDR1, W      ;Write address
MOVWF  EEADR          ;of desired
MOVF   ADDRH, W      ;program memory
MOVWF  EEADRH        ;location
MOVF   VALUEL, W     ;Write value to
MOVWF  EEDATA        ;program at
MOVF   VALUEH, W     ;desired memory
MOVWF  EEDATH        ;location
BSF    STATUS, RP0    ;Bank 3
BSF    EECON1, EEPGD ;Point to Program memory
BSF    EECON1, WREN  ;Enable writes
                ;Only disable interrupts
BCF    INTCON, GIE   ;if already enabled,
                ;otherwise discard
MOVLW  0x55          ;Write 55h to
MOVWF  EECON2        ;EECON2
MOVLW  0xAA          ;Write AAh to
MOVWF  EECON2        ;EECON2
BSF    EECON1, WR    ;Start write operation
NOP    ;Two NOPs to allow micro
NOP    ;to setup for write
                ;Only enable interrupts
BSF    INTCON, GIE   ;if using interrupts,
                ;otherwise discard
BCF    EECON1, WREN  ;Disable writes
```

3.7 Write Verify

The PIC16F870/871 devices do not automatically verify the value written during a write operation. Depending on the application, good programming practice may dictate that the value written to memory be verified against the original value. This should be used in applications where excessive writes can stress bits near the specified endurance limits.

3.8 Protection Against Spurious Writes

There are conditions when the device may not want to write to the EEPROM data memory or FLASH program memory. To protect against these spurious write conditions, various mechanisms have been built into the PIC16F870/871 devices. On power-up, the WREN bit is cleared and the Power-up Timer (if enabled) prevents writes.

The write initiate sequence and the WREN bit together, help prevent any accidental writes during brown-out, power glitches, or firmware malfunction.

3.9 Operation While Code Protected

The PIC16F870/871 devices have two code protect mechanisms, one bit for EEPROM data memory and two bits for FLASH program memory. Data can be read and written to the EEPROM data memory, regardless of the state of the code protection bit, CPD. When code protection is enabled and CPD cleared, external access via ICSP is disabled, regardless of the state of the program memory code protect bits. This prevents the contents of EEPROM data memory from being read out of the device.

The state of the program memory code protect bits, CP0 and CP1, do not affect the execution of instructions out of program memory. The PIC16F870/871 devices can always read the values in program memory, regardless of the state of the code protect bits. However, the state of the code protect bits and the WRT bit will have different effects on writing to program memory. Table 4-1 shows the effect of the code protect bits and the WRT bit on program memory.

Once code protection has been enabled for either EEPROM data memory or FLASH program memory, only a full erase of the entire device will disable code protection.

PIC16F870/871

3.10 FLASH Program Memory Write Protection

The configuration word contains a bit that write protects the FLASH program memory, called WRT. This bit can only be accessed when programming the PIC16F870/871 devices via ICSP. Once write protection is enabled, only an erase of the entire device will disable it. When enabled, write protection prevents any writes to FLASH program memory. Write protection does not affect program memory reads.

TABLE 3-1: READ/WRITE STATE OF INTERNAL FLASH PROGRAM MEMORY

| Configuration Bits | | | Memory Location | Internal Read | Internal Write | ICSP Read | ICSP Write |
|--------------------|-----|-----|--------------------|---------------|----------------|-----------|------------|
| CP1 | CP0 | WRT | | | | | |
| 0 | 0 | x | All program memory | Yes | No | No | No |
| 0 | 1 | 0 | Unprotected areas | Yes | No | Yes | No |
| 0 | 1 | 0 | Protected areas | Yes | No | No | No |
| 0 | 1 | 1 | Unprotected areas | Yes | Yes | Yes | No |
| 0 | 1 | 1 | Protected areas | Yes | No | No | No |
| 1 | 0 | 0 | Unprotected areas | Yes | No | Yes | No |
| 1 | 0 | 0 | Protected areas | Yes | No | No | No |
| 1 | 0 | 1 | Unprotected areas | Yes | Yes | Yes | No |
| 1 | 0 | 1 | Protected areas | Yes | No | No | No |
| 1 | 1 | 0 | All program memory | Yes | No | Yes | Yes |
| 1 | 1 | 1 | All program memory | Yes | Yes | Yes | Yes |

TABLE 3-2: REGISTERS ASSOCIATED WITH DATA EEPROM/PROGRAM FLASH

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|----------------------|--------|--|-------|---------------------------------|---------------------------|-------|-------|-------|-------|--------------------|---------------------------|
| 0Bh, 8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x | 0000 000u |
| 10Dh | EEADR | EEPROM Address Register, Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10Fh | EEADRH | — | — | — | EEPROM Address, High Byte | | | | | xxxx xxxx | uuuu uuuu |
| 10Ch | EEDATA | EEPROM Data Register, Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10Eh | EEDATH | — | — | EEPROM Data Register, High Byte | | | | | | xxxx xxxx | uuuu uuuu |
| 18Ch | EECON1 | EEPGD | — | — | — | WRERR | WREN | WR | RD | x--- x000 | x--- u000 |
| 18Dh | EECON2 | EEPROM Control Register2 (not a physical register) | | | | | | | | — | — |

Legend: x = unknown, u = unchanged, r = reserved, - = unimplemented, read as '0'.
Shaded cells are not used during FLASH/EEPROM access.

Note 1: These bits are reserved; always maintain these bits clear.

4.0 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.

Additional information on I/O ports may be found in the PIC® Mid-Range MCU Family Reference Manual (DS33023).

4.1 PORTA and the TRISA Register

PORTA is a 6-bit wide bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

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. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other PORTA pins have TTL input levels and full CMOS output drivers.

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register 1).

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

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

EXAMPLE 4-1: INITIALIZING PORTA

```
BCF STATUS, RP0 ;
BCF STATUS, RP1 ;Bank0
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 4-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS

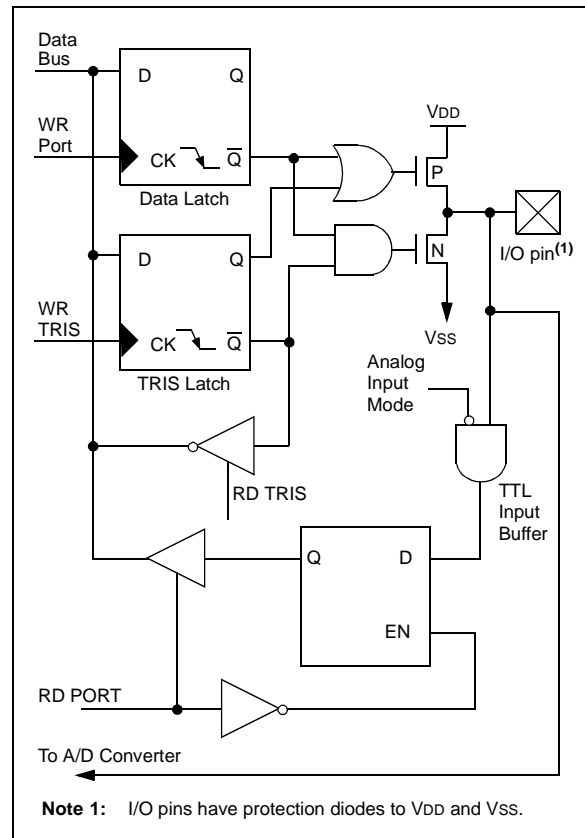
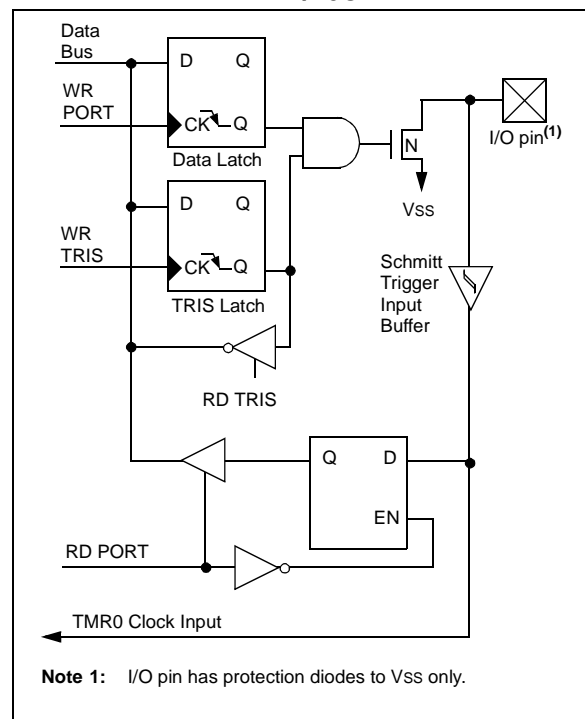


FIGURE 4-2: BLOCK DIAGRAM OF RA4/T0CKI PIN



PIC16F870/871

TABLE 4-1: PORTA FUNCTIONS

| Name | Bit# | Buffer | Function |
|--------------|------|--------|---|
| RA0/AN0 | bit0 | TTL | Input/output or analog input. |
| RA1/AN1 | bit1 | TTL | Input/output or analog input. |
| RA2/AN2 | bit2 | TTL | Input/output or analog input. |
| RA3/AN3/VREF | bit3 | TTL | Input/output or analog input or VREF. |
| RA4/T0CKI | bit4 | ST | Input/output or external clock input for Timer0. Output is open drain type. |
| RA5/AN4 | bit5 | TTL | Input/output or analog input. |

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

TABLE 4-2: 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 | ADFM | — | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 | --0- 0000 | --0- 0000 |

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

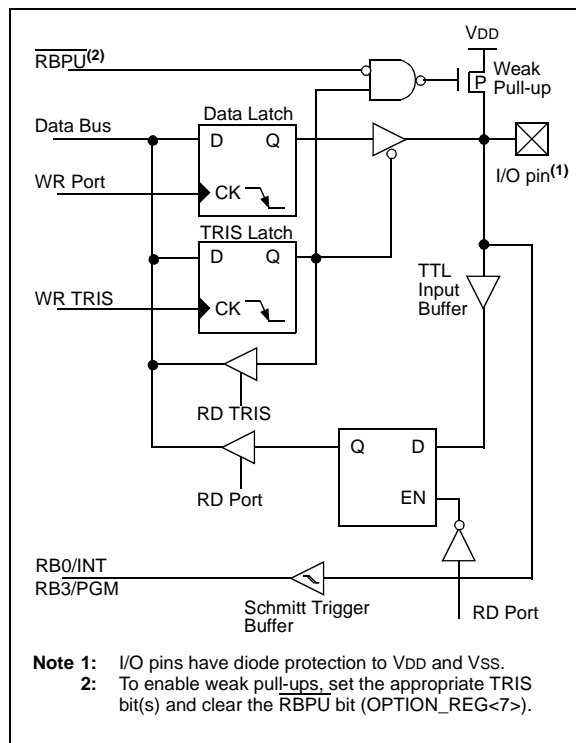
4.2 PORTB and the TRISB Register

PORTB is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., put the contents of the output latch on the selected pin).

Three pins of PORTB are multiplexed with the Low Voltage Programming function: RB3/PGM, RB6/PGC and RB7/PGD. The alternate functions of these pins are described in the Special Features Section.

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 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 4-3: BLOCK DIAGRAM OF RB3:RB0 PINS



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

This interrupt can wake the device from SLEEP. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of PORTB. This will end the mismatch condition.
- 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.

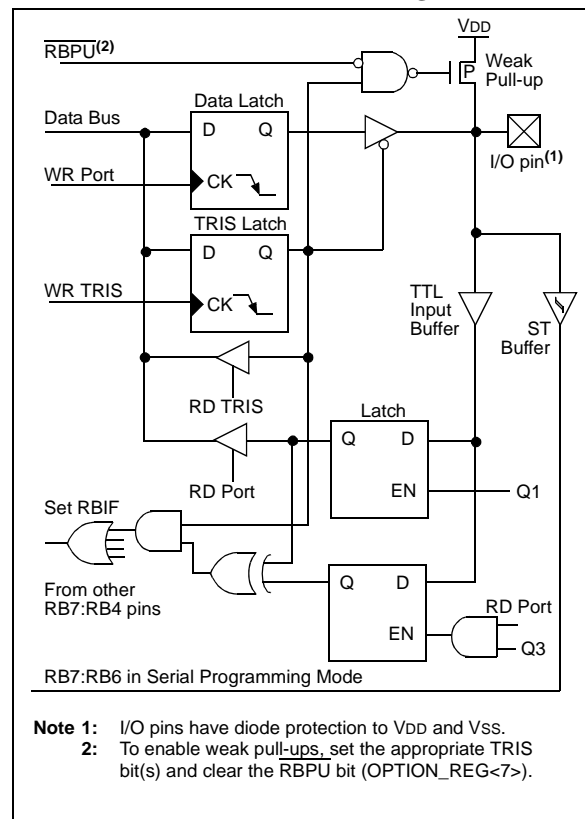
The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

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).

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 11.10.1.

FIGURE 4-4: BLOCK DIAGRAM OF RB7:RB4 PINS



PIC16F870/871

TABLE 4-3: 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/PGM | bit3 | TTL/ST ⁽¹⁾ | Input/output pin or programming pin in LVP mode. 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/PGC | bit6 | TTL/ST ⁽²⁾ | Input/output pin (with interrupt-on-change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming clock. |
| RB7/PGD | bit7 | TTL/ST ⁽²⁾ | Input/output pin (with interrupt-on-change) or In-Circuit Debugger pin. 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 or LVP mode.

2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

TABLE 4-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|-----------|------------|-------------------------------|--------|-------|-------|-------|-------|-------|-------|-----------------------|---------------------------------|
| 06h, 106h | PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | uuuu uuuu |
| 86h, 186h | TRISB | PORTB Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |
| 81h, 181h | OPTION_REG | RBP | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |

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

4.3 PORTC and the TRISC Register

PORTC is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input (i.e., put the corresponding output driver in a Hi-Impedance mode). Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output (i.e., put the contents of the output latch on the selected pin).

PORTC is multiplexed with several peripheral functions (Table 4-5). 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 the destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

FIGURE 4-5: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE)

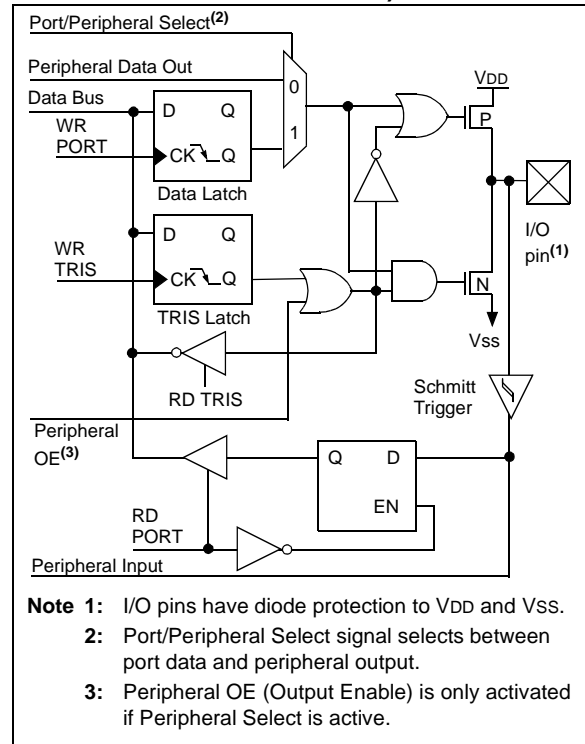


TABLE 4-5: PORTC FUNCTIONS

| Name | Bit# | Buffer Type | Function |
|-----------------|------|-------------|--|
| RC0/T1OSO/T1CKI | bit0 | ST | Input/output port pin or Timer1 oscillator output/Timer1 clock input. |
| RC1/T1OSI | bit1 | ST | Input/output port pin or Timer1 oscillator input. |
| RC2/CCP1 | bit2 | ST | Input/output port pin or Capture1 input/Compare1 output/PWM1 output. |
| RC3 | bit3 | ST | Input/output port pin. |
| RC4 | bit4 | ST | Input/output port pin. |
| RC5 | bit5 | ST | Input/output port pin. |
| RC6/TX/CK | bit6 | ST | Input/output port pin or USART Asynchronous Transmit or Synchronous Clock. |
| RC7/RX/DT | bit7 | ST | Input/output port pin or USART Asynchronous Receive or Synchronous Data. |

Legend: ST = Schmitt Trigger input

TABLE 4-6: 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

PIC16F870/871

4.4 PORTD and TRISD Registers

This section is not applicable to the PIC16F870.

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

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

FIGURE 4-6: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)

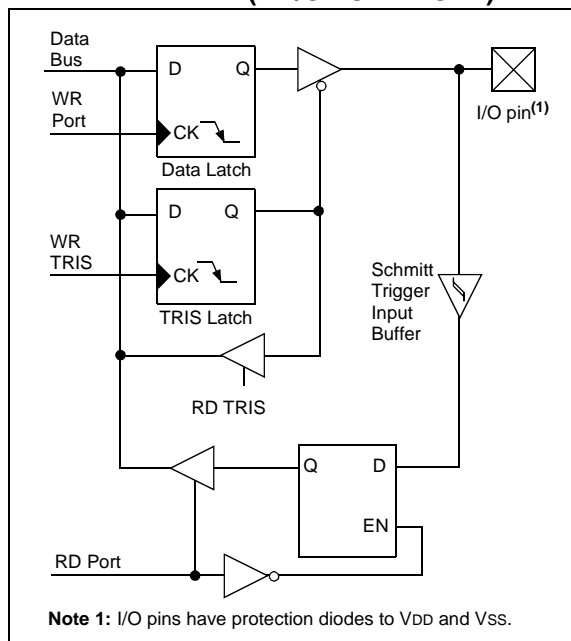


TABLE 4-7: PORTD FUNCTIONS

| 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 buffers when in Parallel Slave Port mode.

TABLE 4-8: 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.

4.5 PORTE and TRISE Register

This section is not applicable to the PIC16F870.

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

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

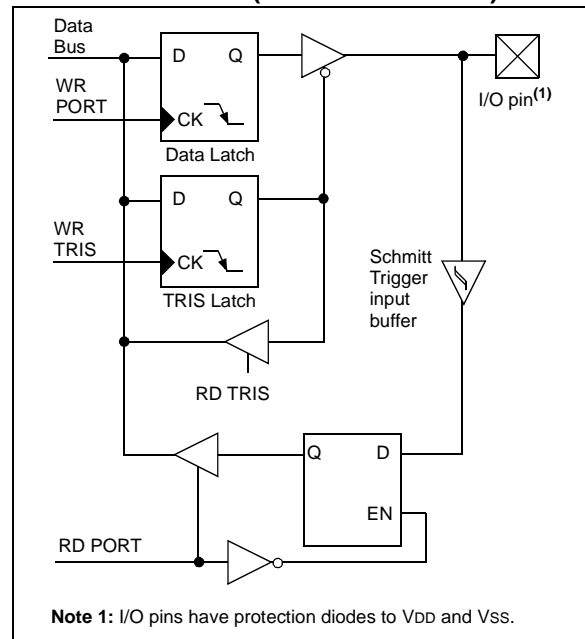
Register 4-1 shows the TRISE register, which also controls the parallel slave port operation.

PORTE pins are multiplexed with analog inputs. When selected as an analog input, these pins will read as '0's.

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

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

FIGURE 4-7: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)



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

PIC16F870/871

REGISTER 4-1: 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 | — | Bit2 | Bit1 | Bit0 |
| bit 7 | | | | | | | bit 0 |

- bit 7 **Parallel Slave Port Status/Control Bits**
IBF: Input Buffer Full Status bit
 1 = A word has been received and is waiting to be read by the CPU
 0 = No word has been received
- bit 6 **OBF:** Output Buffer Full Status bit
 1 = The output buffer still holds a previously written word
 0 = The output buffer has been read
- bit 5 **IBOV:** Input Buffer Overflow Detect bit (in Microprocessor mode)
 1 = A write occurred when a previously input word has not been read
 (must be cleared in software)
 0 = No overflow occurred
- bit 4 **PSPMODE:** Parallel Slave Port Mode Select bit
 1 = Parallel Slave Port mode
 0 = General Purpose I/O mode
- bit 3 **Unimplemented:** Read as '0'
PORTE Data Direction Bits
- bit 2 **Bit2:** Direction Control bit for pin RE2/ $\overline{\text{CS}}$ /AN7
 1 = Input
 0 = Output
- bit 1 **Bit1:** Direction Control bit for pin RE1/ $\overline{\text{WR}}$ /AN6
 1 = Input
 0 = Output
- bit 0 **Bit0:** Direction Control bit for pin RE0/ $\overline{\text{RD}}$ /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 4-9: PORTE FUNCTIONS

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

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

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

TABLE 4-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

| Addr | 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 | ADFM | — | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 | --0- 0000 | --0- 0000 |

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

PIC16F870/871

4.6 Parallel Slave Port

The Parallel Slave Port is not implemented on the PIC16F870.

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

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} to be the \overline{RD} input, RE1/ \overline{WR} to be the \overline{WR} input and RE2/ \overline{CS} 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). The A/D port configuration bits PCFG3:PCFG0 (ADCON1<3:0>) must be set to configure pins RE2:RE0 as digital I/O.

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

A write to the PSP occurs when both the \overline{CS} and \overline{WR} lines are first detected low. When either the \overline{CS} or \overline{WR} lines become high (level triggered), 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 4-9). 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 4-10), 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 4-8: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)

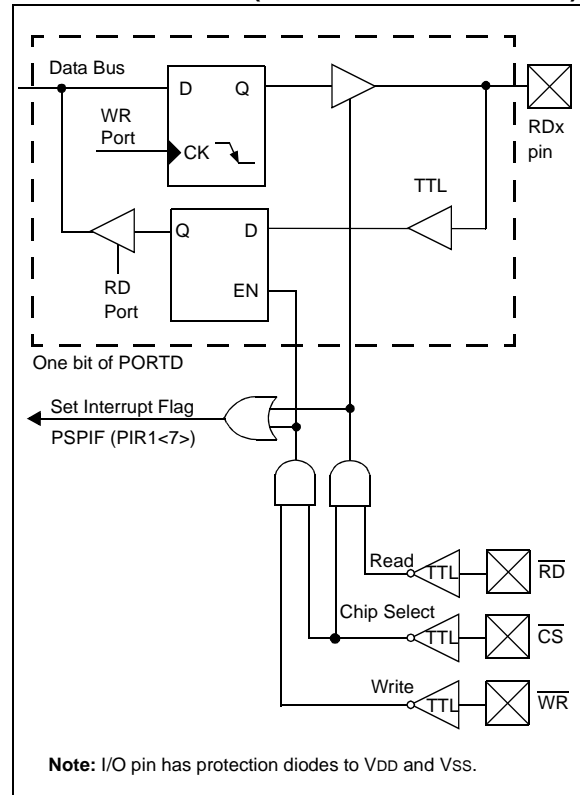


FIGURE 4-9: PARALLEL SLAVE PORT WRITE WAVEFORMS

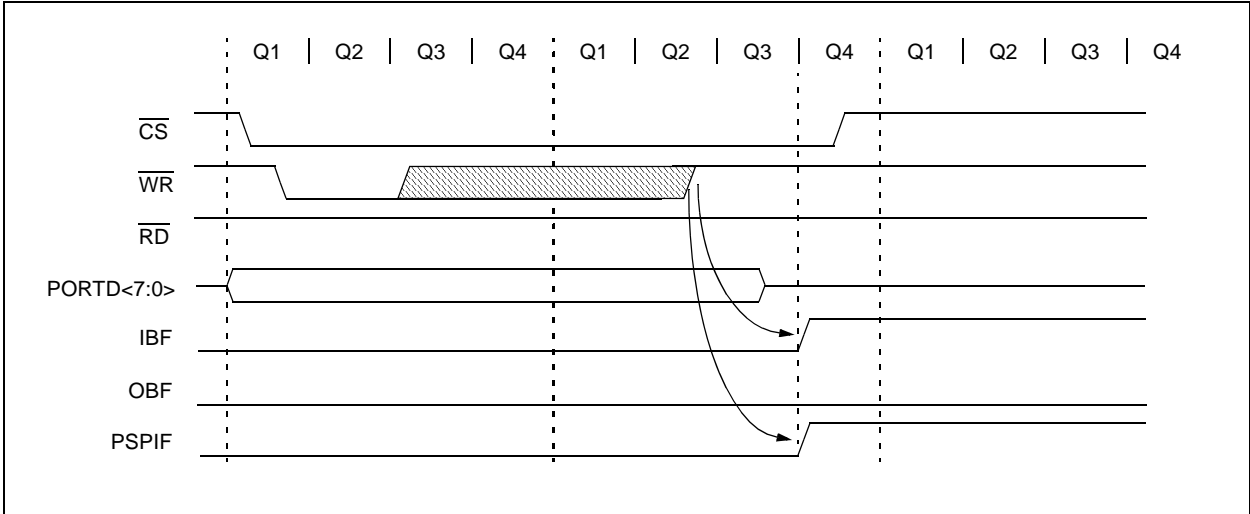


FIGURE 4-10: PARALLEL SLAVE PORT READ WAVEFORMS

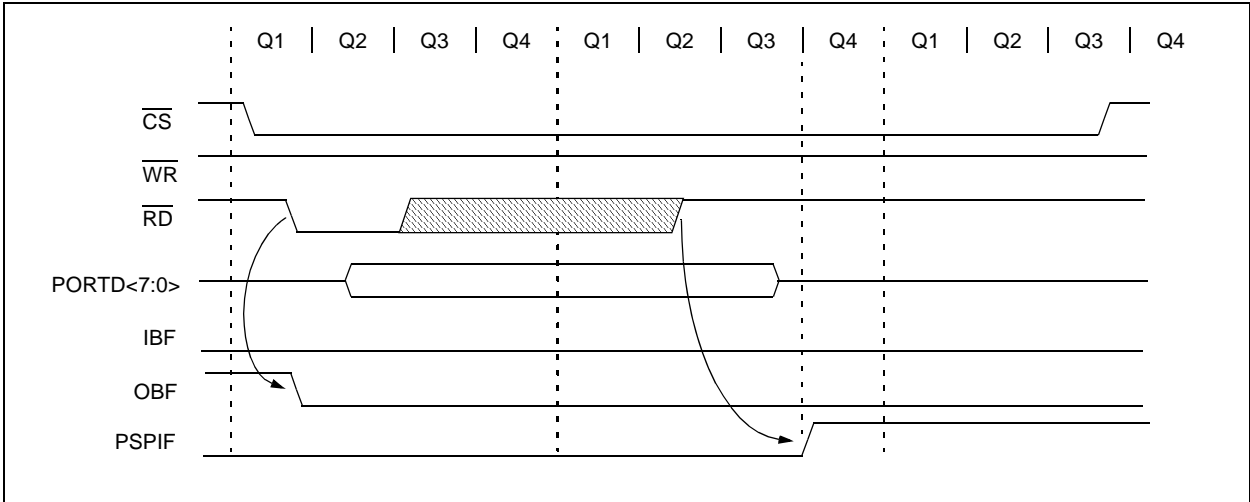


TABLE 4-11: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|---------|--------|---|-------|-------|---------|-------|---------------------------|--------|--------|--------------------|---------------------------|
| 08h | PORTD | Port Data Latch when written: Port pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 09h | PORTE | - | - | - | - | - | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |
| 89h | TRISE | IBF | OBF | IBOV | PSPMODE | - | PORTE Data Direction bits | | | 0000 -111 | 0000 -111 |
| 0Ch | PIR1 | PSPIF | ADIF | RCIF | TXIF | - | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE | ADIE | RCIE | TXIE | - | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 9Fh | ADCON1 | ADFM | - | - | - | PCFG3 | PCFG2 | PCFG1 | PCFG0 | --0- 0000 | --0- 0000 |

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

PIC16F870/871

NOTES:

5.0 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 5-1 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.

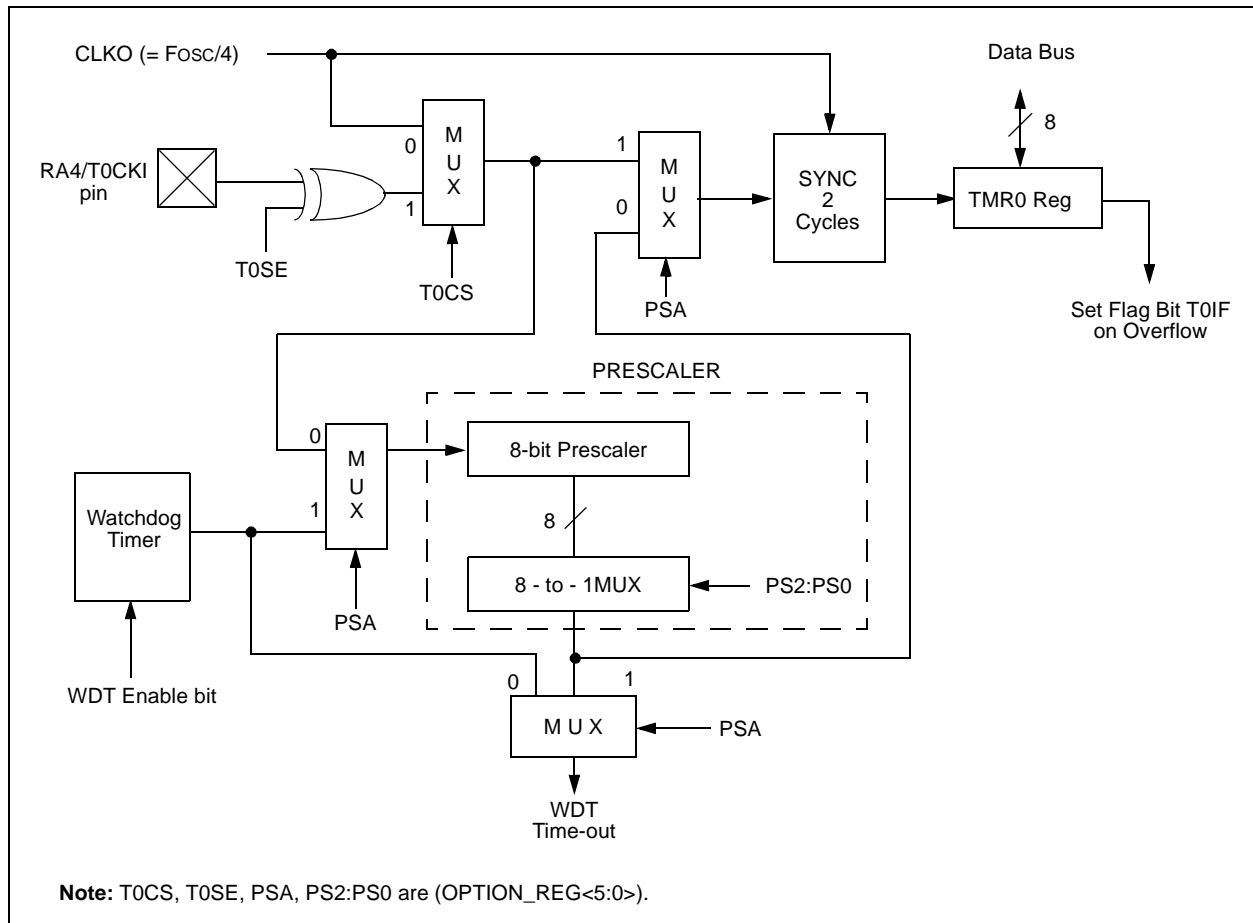
Counter mode is selected by setting bit T0CS (OPTION_REG<5>). In Counter mode, Timer0 will increment either on every rising, or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (OPTION_REG<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 5.2.

The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler is not readable or writable. Section 5.3 details the operation of the prescaler.

5.1 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 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



PIC16F870/871

5.2 Using Timer0 with an External Clock

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. 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 of the desired device.

5.3 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 5-1).

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., CLRFL1, MOVWF1, BSF1, 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 5-1: 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|---|----------|-------|-------|-------|-------|-------|-------|-----------|-----------|----------|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|--------|-------|-----|--------|--------|-----|--------|--------|-----|---------|--------|-----|---------|---------|
| | RBPU | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | bit 7 | | | | | | | bit 0 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bit 7 | RBPU | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bit 6 | INTEDG | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| bit 5 | T0CS: TMR0 Clock Source Select bit 1 = Transition on T0CKI pin 0 = Internal instruction cycle clock (CLKO) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 5px;"> <thead> <tr> <th style="width: 15%;">Bit Value</th> <th style="width: 20%;">TMR0 Rate</th> <th style="width: 20%;">WDT Rate</th> </tr> </thead> <tbody> <tr><td>000</td><td>1 : 2</td><td>1 : 1</td></tr> <tr><td>001</td><td>1 : 4</td><td>1 : 2</td></tr> <tr><td>010</td><td>1 : 8</td><td>1 : 4</td></tr> <tr><td>011</td><td>1 : 16</td><td>1 : 8</td></tr> <tr><td>100</td><td>1 : 32</td><td>1 : 16</td></tr> <tr><td>101</td><td>1 : 64</td><td>1 : 32</td></tr> <tr><td>110</td><td>1 : 128</td><td>1 : 64</td></tr> <tr><td>111</td><td>1 : 256</td><td>1 : 128</td></tr> </tbody> </table> | | | | | | | | 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 |
| 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) must be executed when changing the prescaler assignment from Timer0 to the WDT. This sequence must be followed even if the WDT is disabled.

TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|-----------------------|------------|--------------------------|--------|-------|-------|-------|-------|-------|-------|-----------------------|---------------------------------|
| 01h,101h | TMR0 | Timer0 Module's Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Bh,8Bh, 10Bh,18Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 81h,181h | OPTION_REG | 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.

PIC16F870/871

NOTES:

6.0 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 8.0). Register 6-1 shows the Timer1 control register.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI 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 6-1: 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
When TMR1CS = 1:
 1 = Do not synchronize external clock input
 0 = Synchronize external clock input
When 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 |

PIC16F870/871

6.1 Timer1 Operation in Timer Mode

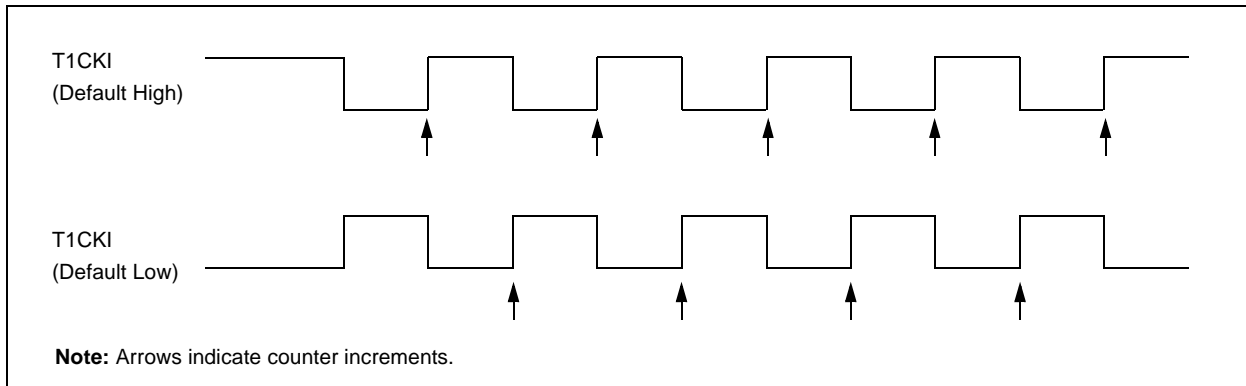
Timer mode is selected by clearing the TMR1CS (T1CON<1>) bit. In this mode, the input clock to the timer is $F_{OSC}/4$. The synchronize control bit, $\overline{T1SYNC}$ (T1CON<2>), has no effect, since the internal clock is always in sync.

6.2 Timer1 Counter Operation

Timer1 may operate in either a Synchronous, or an Asynchronous mode, depending on the setting of the TMR1CS bit.

When Timer1 is being incremented via an external source, increments occur on a rising edge. After Timer1 is enabled in Counter mode, the module must first have a falling edge before the counter begins to increment.

FIGURE 6-1: TIMER1 INCREMENTING EDGE



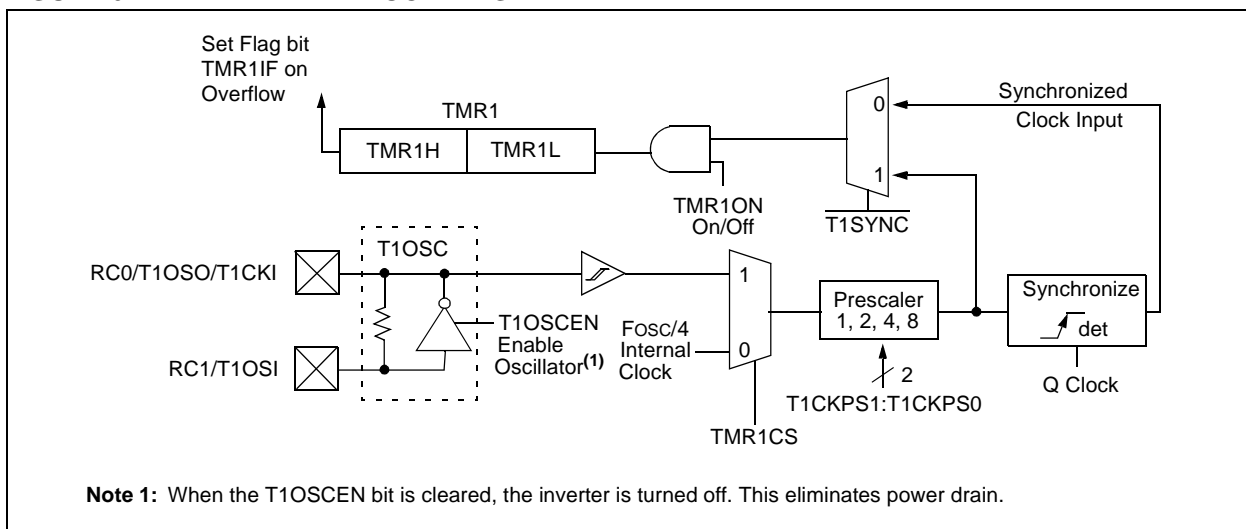
6.3 Timer1 Operation in Synchronized Counter Mode

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, when bit T1OSCEN is set, or on pin RC0/T1OSO/T1CKI, when bit T1OSCEN is cleared.

If $\overline{T1SYNC}$ 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 6-2: TIMER1 BLOCK DIAGRAM



6.4 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{T1SYNC}$ (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 6.4.1).

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

6.4.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock, will ensure 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.

6.5 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 6-1 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 6-1: 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. | | | |

6.6 Resetting Timer1 Using a CCP Trigger Output

If the CCP1 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 module 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, the write will take precedence.

In this mode of operation, the CCPRH:CCPRL register pair effectively becomes the period register for Timer1.

PIC16F870/871

6.7 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 special event trigger.

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.

6.8 Timer1 Prescaler

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

TABLE 6-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|-------------------|--------|---|-------|---------|---------|---------|--------|--------|--------|--------------------|---------------------------|
| 0Bh,8Bh,10Bh,18Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 0Eh | TMR1L | Holding Register for the Least Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding Register for the Most Significant Byte of the 16-bit TMR1 Register | | | | | | | | xxxx xxxxx | 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 PIC16F870; always maintain these bits clear.

7.0 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 ($F_{osc}/4$) 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.

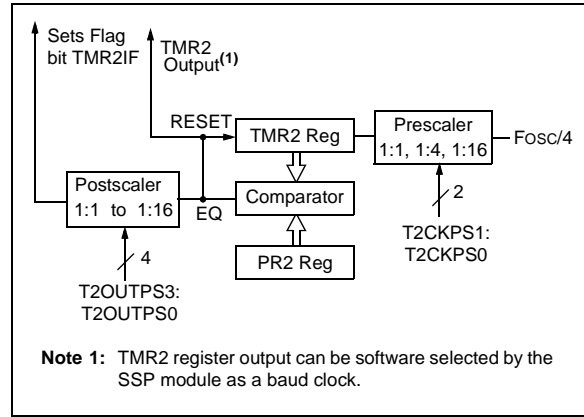
The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit TMR2IF (PIR1<1>)).

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

Register 7-1 shows the Timer2 control register.

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

FIGURE 7-1: TIMER2 BLOCK DIAGRAM



REGISTER 7-1: 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 |

PIC16F870/871

7.1 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, MCLR Reset, WDT Reset, or BOR)

TMR2 is not cleared when T2CON is written.

7.2 Output of TMR2

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

TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|-------------------|--------|--------------------------|---------|---------|---------|---------|--------|---------|---------|--------------------|---------------------------|
| 0Bh,8Bh,10Bh,18Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 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 PIC16F870; always maintain these bits clear.

8.0 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

Table 8-1 shows the resources and interactions of the CCP module. In the following sections, the operation of a CCP module is described.

Additional information on CCP modules is available in the PIC[®] Mid-Range MCU Family Reference Manual (DS33023) and in application note AN594, "Using the CCP Modules" (DS00594).

TABLE 8-1: CCP MODE - TIMER RESOURCES REQUIRED

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

8.1 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 and start an A/D conversion (if the A/D module is enabled).

REGISTER 8-1: CCP1CON REGISTER REGISTER (ADDRESS: 17h/1Dh)

| | | | | | | | | |
|-------|-----|-------|-------|--------|--------|--------|--------|-------|
| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | |
| — | — | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | |
| bit 7 | | | | | | | | bit 0 |

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

bit 5-4 **CCP1X:CCP1Y:** 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 CCPR1L.

bit 3-0 **CCP1M3:CCP1M0:** CCP1 Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCP1 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 (CCP1IF bit is set)

1001 = Compare mode, clear output on match (CCP1IF bit is set)

1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP1 pin is unaffected)

1011 = Compare mode, trigger special event (CCP1IF bit is set, CCP1 pin is unaffected); CCP1 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

PIC16F870/871

8.2 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:

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

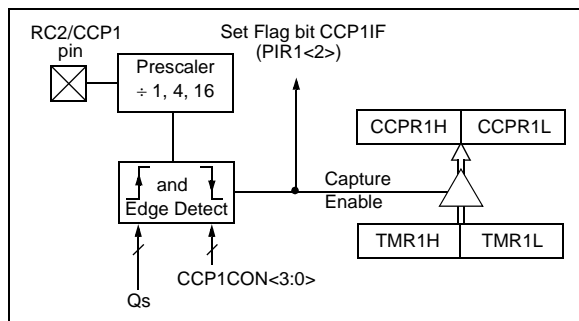
The type of event is configured by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit CCP1IF (PIR1<2>) is set. The interrupt flag must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value is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

8.2.1 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 8-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



8.2.2 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.

8.2.3 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.

8.2.4 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 8-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

EXAMPLE 8-1: CHANGING BETWEEN CAPTURE PRESCALERS

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

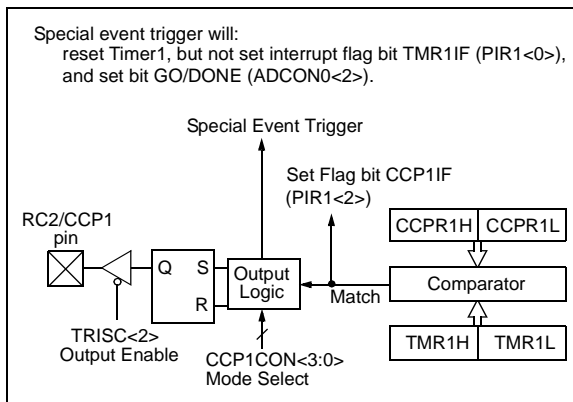
8.3 Compare Mode

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 8-2: COMPARE MODE OPERATION BLOCK DIAGRAM



8.3.1 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.

8.3.2 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.

8.3.3 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).

8.3.4 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, and starts an A/D conversion (if A/D module is enabled). This allows the CCPR1 register to effectively be a 16-bit programmable period register for Timer1.

Note: The special event trigger from the CCP1 module will not set interrupt flag bit TMR1IF (PIR1<0>).

PIC16F870/871

8.4 PWM Mode (PWM)

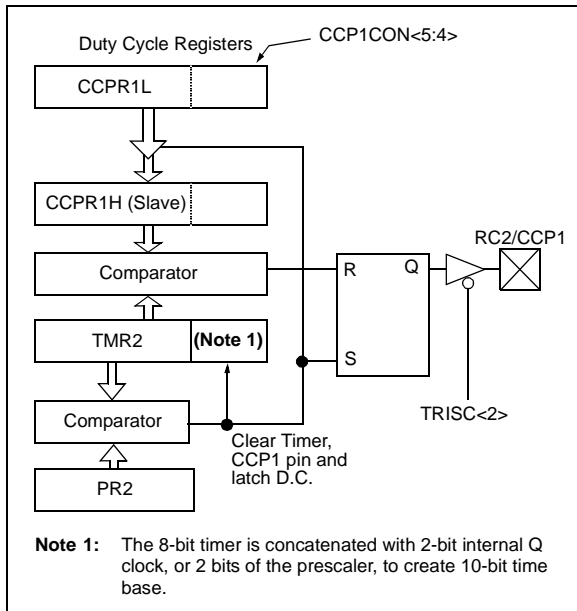
In Pulse Width Modulation mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTC data latch, the TRISC<2> bit must be cleared to make the CCP1 pin an output.

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 8-3 shows a simplified block diagram of the CCP module in PWM mode.

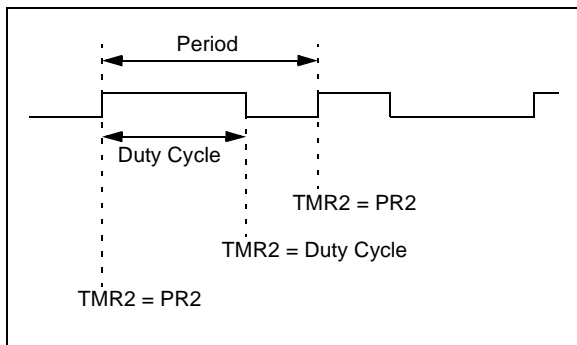
For a step-by-step procedure on how to set up the CCP module for PWM operation, see Section 8.4.3.

FIGURE 8-3: SIMPLIFIED PWM BLOCK DIAGRAM



A PWM output (Figure 8-4) 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 8-4: PWM OUTPUT



8.4.1 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 $1 / [\text{PWM period}]$.

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 7.1) 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.

8.4.2 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 glitch-free 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.

The maximum PWM resolution (bits) for a given PWM frequency is given by the formula:

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

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

8.4.3 SETUP 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 8-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz

| PWM Frequency | 1.22 kHz | 4.88 kHz | 19.53 kHz | 78.12kHz | 156.3 kHz | 208.3 kHz |
|----------------------------|----------|----------|-----------|----------|-----------|-----------|
| Timer Prescaler (1, 4, 16) | 16 | 4 | 1 | 1 | 1 | 1 |
| PR2 Value | 0xFFh | 0xFFh | 0xFFh | 0x3Fh | 0x1Fh | 0x17h |
| Maximum Resolution (bits) | 10 | 10 | 10 | 8 | 7 | 6.5 |

TABLE 8-3: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, AND TIMER1

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|---------------------|---------|---|-------|---------|---------|---------|--------|--------|--------|--------------------|---------------------------|
| 0Bh,8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 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 |

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 PIC16F870; always maintain these bits clear.

PIC16F870/871

TABLE 8-4: REGISTERS ASSOCIATED WITH PWM AND TIMER2

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|------------------------|---------|-------------------------------------|---------|---------|---------|---------|--------|---------|---------|-----------------------|---------------------------------|
| 0Bh,8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 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 |

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 PIC16F870; always maintain these bits clear.

9.0 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)

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

The USART module also has a multi-processor communication capability using 9-bit address detection.

REGISTER 9-1: 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 |

PIC16F870/871

REGISTER 9-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

| | | | | | | | |
|-------|-------|-------|-------|-------|------|-------|------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R-0 | R-x |
| SPEN | RX9 | SREN | CREN | ADDEN | 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 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
 1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8> is set
 0 = Disables address detection, all bytes are received, and ninth bit can be used as parity bit
- 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, but must be calculated by user 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 |

9.1 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 9-1 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 9-1. 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 $F_{OSC}/(16(X + 1))$ 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.

9.1.1 SAMPLING

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

TABLE 9-1: BAUD RATE FORMULA

| SYNC | BRGH = 0 (Low Speed) | BRGH = 1 (High Speed) |
|------|--|---------------------------------|
| 0 | (Asynchronous) Baud Rate = $F_{OSC}/(64(X+1))$ | Baud Rate = $F_{OSC}/(16(X+1))$ |
| 1 | (Synchronous) Baud Rate = $F_{OSC}/(4(X+1))$ | N/A |

Legend: X = value in SPBRG (0 to 255)

TABLE 9-2: 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 | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 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.

PIC16F870/871

TABLE 9-3: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

| BAUD RATE (K) | Fosc = 20 MHz | | | Fosc = 16 MHz | | | Fosc = 10 MHz | | |
|---------------|---------------|---------|-----------------------|---------------|---------|-----------------------|---------------|---------|-----------------------|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 0.3 | - | - | - | - | - | - | - | - | - |
| 1.2 | 1.221 | 1.75 | 255 | 1.202 | 0.17 | 207 | 1.202 | 0.17 | 129 |
| 2.4 | 2.404 | 0.17 | 129 | 2.404 | 0.17 | 103 | 2.404 | 0.17 | 64 |
| 9.6 | 9.766 | 1.73 | 31 | 9.615 | 0.16 | 25 | 9.766 | 1.73 | 15 |
| 19.2 | 19.531 | 1.72 | 15 | 19.231 | 0.16 | 12 | 19.531 | 1.72 | 7 |
| 28.8 | 31.250 | 8.51 | 9 | 27.778 | 3.55 | 8 | 31.250 | 8.51 | 4 |
| 33.6 | 34.722 | 3.34 | 8 | 35.714 | 6.29 | 6 | 31.250 | 6.99 | 4 |
| 57.6 | 62.500 | 8.51 | 4 | 62.500 | 8.51 | 3 | 52.083 | 9.58 | 2 |
| HIGH | 1.221 | - | 255 | 0.977 | - | 255 | 0.610 | - | 255 |
| LOW | 312.500 | - | 0 | 250.000 | - | 0 | 156.250 | - | 0 |

| BAUD RATE (K) | Fosc = 4 MHz | | | Fosc = 3.6864 MHz | | |
|---------------|--------------|---------|-----------------------|-------------------|---------|-----------------------|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 0.3 | 0.300 | 0 | 207 | 0.3 | 0 | 191 |
| 1.2 | 1.202 | 0.17 | 51 | 1.2 | 0 | 47 |
| 2.4 | 2.404 | 0.17 | 25 | 2.4 | 0 | 23 |
| 9.6 | 8.929 | 6.99 | 6 | 9.6 | 0 | 5 |
| 19.2 | 20.833 | 8.51 | 2 | 19.2 | 0 | 2 |
| 28.8 | 31.250 | 8.51 | 1 | 28.8 | 0 | 1 |
| 33.6 | - | - | - | - | - | - |
| 57.6 | 62.500 | 8.51 | 0 | 57.6 | 0 | 0 |
| HIGH | 0.244 | - | 255 | 0.225 | - | 255 |
| LOW | 62.500 | - | 0 | 57.6 | - | 0 |

TABLE 9-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

| BAUD RATE (K) | Fosc = 20 MHz | | | Fosc = 16 MHz | | | Fosc = 10 MHz | | |
|---------------|---------------|---------|-----------------------|---------------|---------|-----------------------|---------------|---------|-----------------------|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 0.3 | - | - | - | - | - | - | - | - | - |
| 1.2 | - | - | - | - | - | - | - | - | - |
| 2.4 | - | - | - | - | - | - | 2.441 | 1.71 | 255 |
| 9.6 | 9.615 | 0.16 | 129 | 9.615 | 0.16 | 103 | 9.615 | 0.16 | 64 |
| 19.2 | 19.231 | 0.16 | 64 | 19.231 | 0.16 | 51 | 19.531 | 1.72 | 31 |
| 28.8 | 29.070 | 0.94 | 42 | 29.412 | 2.13 | 33 | 28.409 | 1.36 | 21 |
| 33.6 | 33.784 | 0.55 | 36 | 33.333 | 0.79 | 29 | 32.895 | 2.10 | 18 |
| 57.6 | 59.524 | 3.34 | 20 | 58.824 | 2.13 | 16 | 56.818 | 1.36 | 10 |
| HIGH | 4.883 | - | 255 | 3.906 | - | 255 | 2.441 | - | 255 |
| LOW | 1250.000 | - | 0 | 1000.000 | - | 0 | 625.000 | - | 0 |

| BAUD RATE (K) | Fosc = 4 MHz | | | Fosc = 3.6864 MHz | | |
|---------------|--------------|---------|-----------------------|-------------------|---------|-----------------------|
| | KBAUD | % ERROR | SPBRG value (decimal) | KBAUD | % ERROR | SPBRG value (decimal) |
| 0.3 | - | - | - | - | - | - |
| 1.2 | 1.202 | 0.17 | 207 | 1.2 | 0 | 191 |
| 2.4 | 2.404 | 0.17 | 103 | 2.4 | 0 | 95 |
| 9.6 | 9.615 | 0.16 | 25 | 9.6 | 0 | 23 |
| 19.2 | 19.231 | 0.16 | 12 | 19.2 | 0 | 11 |
| 28.8 | 27.798 | 3.55 | 8 | 28.8 | 0 | 7 |
| 33.6 | 35.714 | 6.29 | 6 | 32.9 | 2.04 | 6 |
| 57.6 | 62.500 | 8.51 | 3 | 57.6 | 0 | 3 |
| HIGH | 0.977 | - | 255 | 0.9 | - | 255 |
| LOW | 250.000 | - | 0 | 230.4 | - | 0 |

PIC16F870/871

9.2 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 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

9.2.1 USART ASYNCHRONOUS TRANSMITTER

The USART transmitter block diagram is shown in Figure 9-1. 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 T_{cy}), the TXREG register is empty and flag bit TXIF (PIR1<4>) is set. This interrupt can be

enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set, regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF 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.

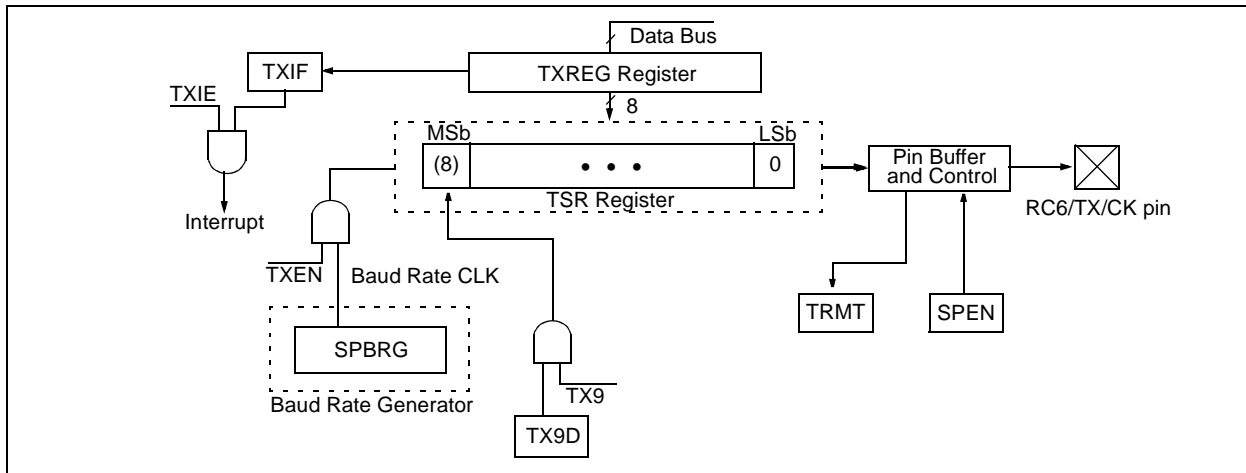
Note 1: The TSR register is not mapped in data memory, so it is not available to the user.

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 9-2). 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 9-3). 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 9-1: USART TRANSMIT BLOCK DIAGRAM



When setting up an Asynchronous Transmission, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH (Section 9.1).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit TXIE.
4. If 9-bit transmission is desired, then set transmit bit TX9.
5. Enable the transmission by setting bit TXEN, which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

FIGURE 9-2: ASYNCHRONOUS MASTER TRANSMISSION

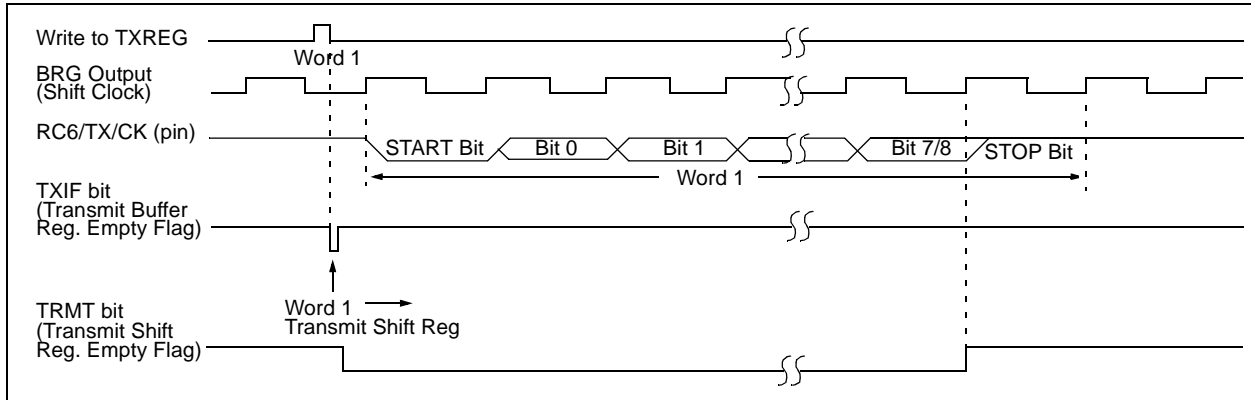


FIGURE 9-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)

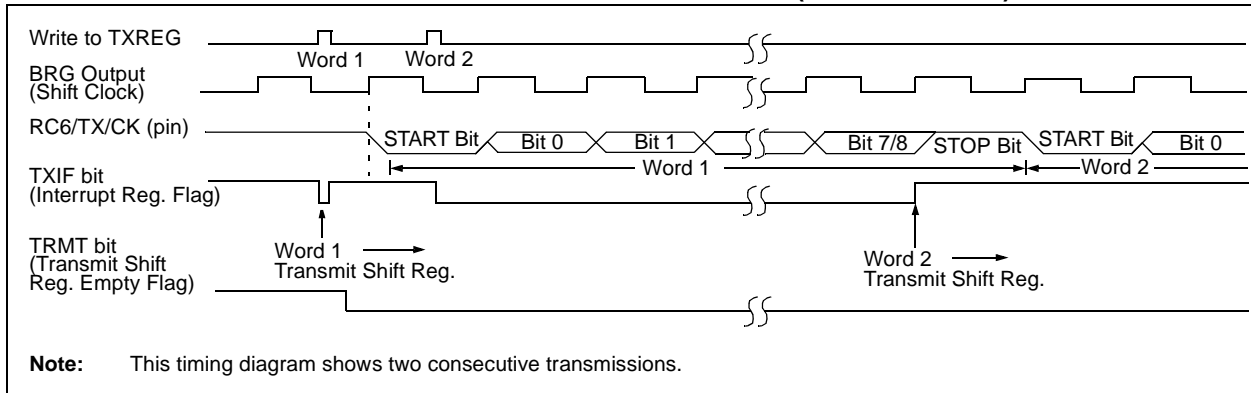


TABLE 9-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

| 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, 10Bh, 18Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | T0IF | INTF | R0IF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 19h | TXREG | USART Transmit Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F870; always maintain these bits clear.

PIC16F870/871

9.2.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 9-4. 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, flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>). Flag bit RCIF is a read only bit, which is cleared by the hardware. It is cleared when the RCREG register has been read and is empty. The RCREG is a double-buffered register (i.e., it is a two-deep FIFO). It

is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte 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, the 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. It is therefore, essential to clear error bit OERR if it is set. Framing error bit FERR (RCSTA<2>) is set if a STOP bit is detected as clear. Bit FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG will load bits RX9D and FERR with new values, therefore, it is essential for the user to read the RCSTA register before reading the RCREG register in order not to lose the old FERR and RX9D information.

FIGURE 9-4: USART RECEIVE BLOCK DIAGRAM

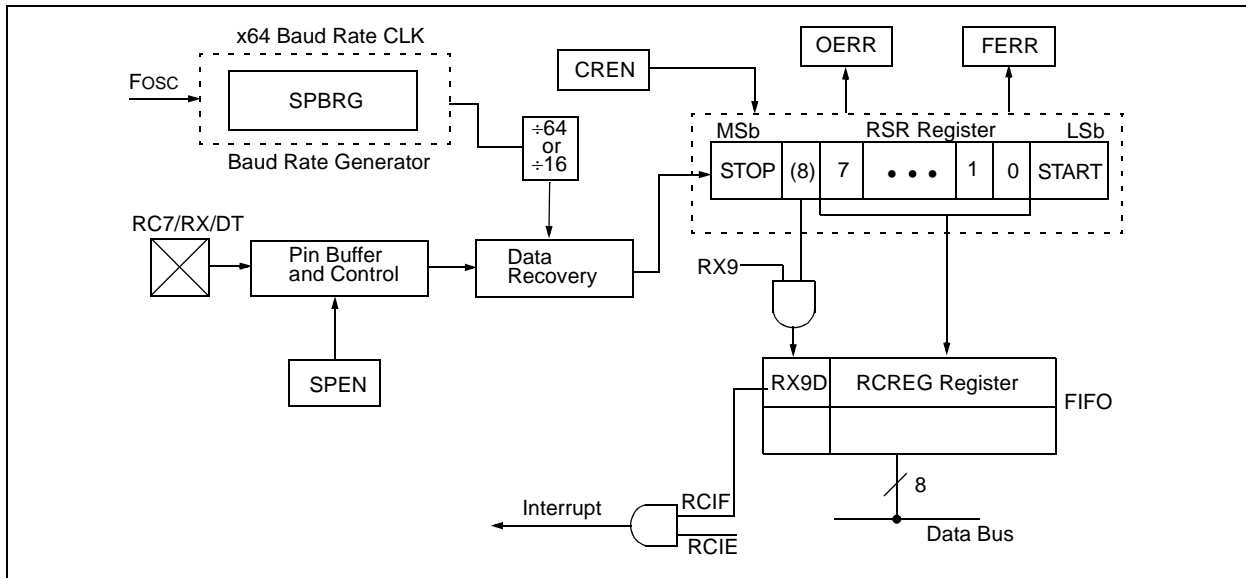
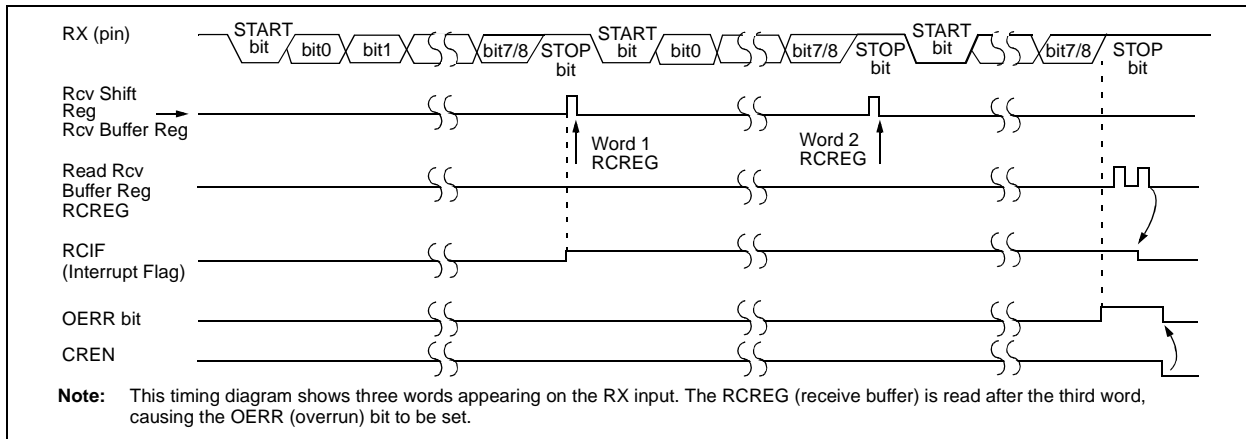


FIGURE 9-5: ASYNCHRONOUS RECEPTION



When setting up an Asynchronous Reception, follow these steps:

1. Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH (Section 9.1).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, then set enable bit RCIE.
4. If 9-bit reception is desired, then set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE is 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.
10. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

TABLE 9-6: 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, 10Bh, 18Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | T0IF | INTF | R0IF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 1Ah | RCREG | USART Receive Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F870; always maintain these bits clear.

PIC16F870/871

9.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

When setting up an Asynchronous Reception with Address Detect enabled:

- Initialize the SPBRG register for the appropriate baud rate. If a high speed baud rate is desired, set bit BRGH.
 - Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
 - If interrupts are desired, then set enable bit RCIE.
 - Set bit RX9 to enable 9-bit reception.
 - Set ADDEN to enable address detect.
 - Enable the reception by setting enable bit CREN.
- Flag bit RCIF will be set when reception is complete, and an interrupt will be generated if enable bit RCIE was set.
 - Read the RCSTA register to get the ninth bit and determine if any error occurred during reception.
 - Read the 8-bit received data by reading the RCREG register, to determine if the device is being addressed.
 - If any error occurred, clear the error by clearing enable bit CREN.
 - If the device has been addressed, clear the ADDEN bit to allow data bytes and address bytes to be read into the receive buffer, and interrupt the CPU.

FIGURE 9-6: USART RECEIVE BLOCK DIAGRAM

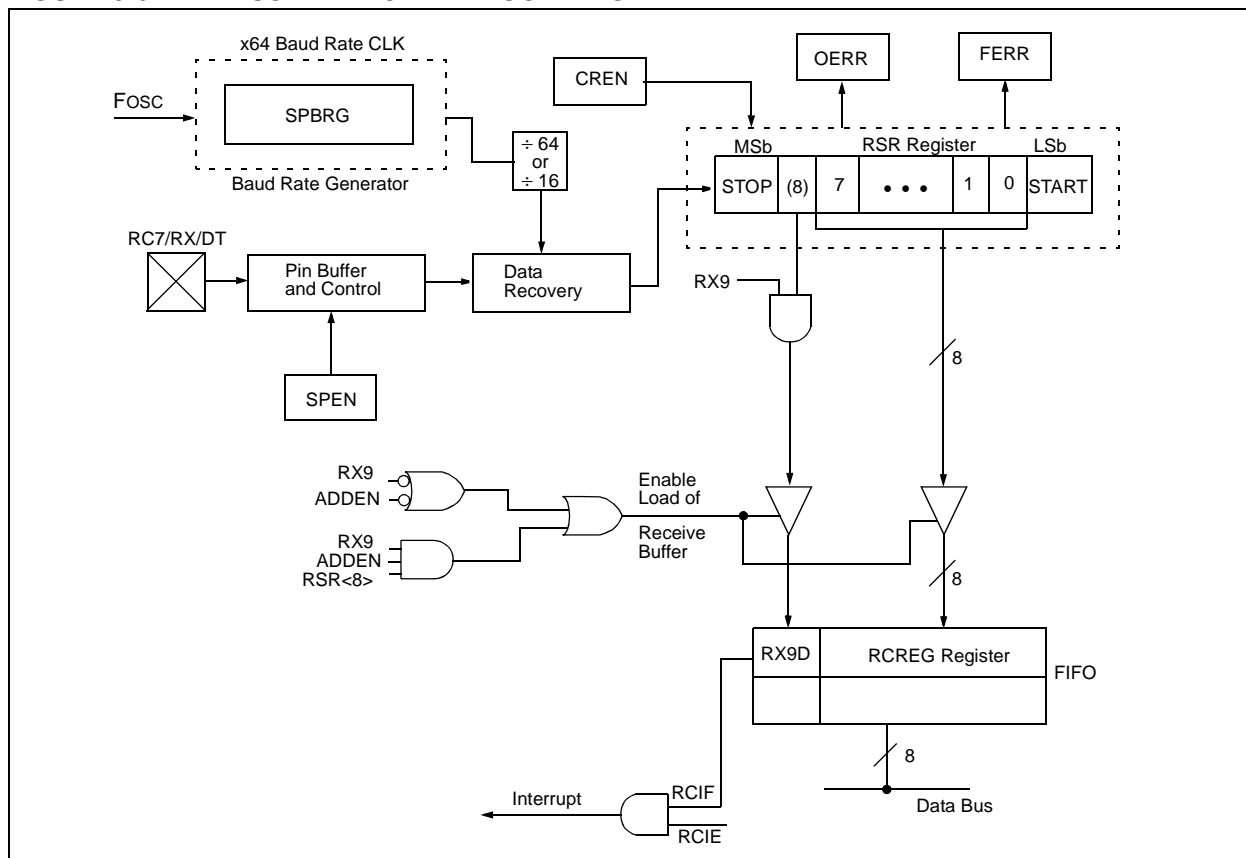


FIGURE 9-7: ASYNCHRONOUS RECEPTION WITH ADDRESS DETECT

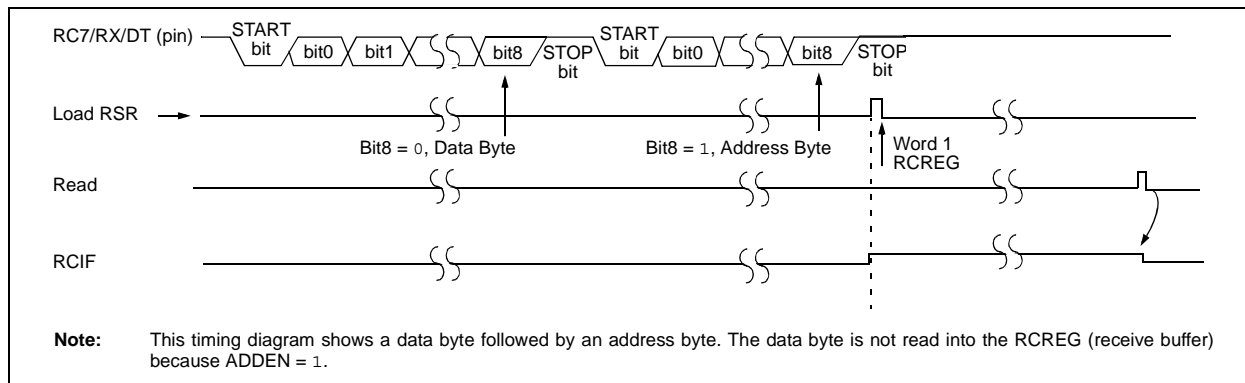


FIGURE 9-8: ASYNCHRONOUS RECEPTION WITH ADDRESS BYTE FIRST

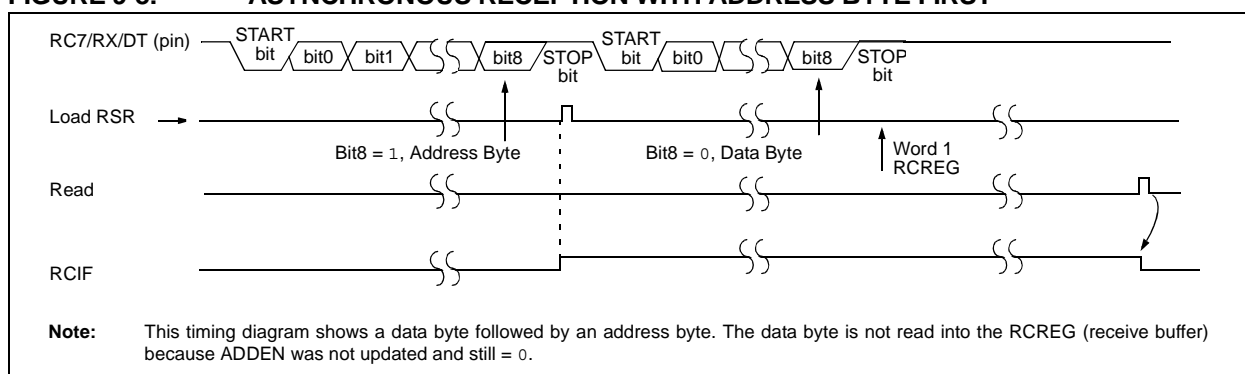


TABLE 9-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|----------------------|--------|------------------------------|-------|-------|-------|-------|--------|--------|--------|--------------------|---------------------------|
| 0Bh, 8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | ROIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 1Ah | RCREG | USART Receive Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F870; always maintain these bits clear.

9.3 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>).

9.3.1 USART SYNCHRONOUS MASTER TRANSMISSION

The USART transmitter block diagram is shown in Figure 9-6. 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 T_{CYCLE}), the TXREG is empty and interrupt bit TXIF (PIR1<4>) is set. The interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1<4>). Flag bit TXIF will be set, regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit TXIF indicates the status of the TXREG register, another bit TRMT (TXSTA<1>) shows the status of the TSR register. TRMT is a read only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory, so it is not available to the user.

Transmission is enabled by setting enable bit TXEN (TXSTA<5>). The actual transmission will not occur until the TXREG register has been loaded with data. The first data bit will be shifted out on the next available rising edge of the clock on the CK line. Data out is stable around the falling edge of the synchronous clock (Figure 9-9). The transmission can also be started by first loading the TXREG register and then setting bit TXEN (Figure 9-10). 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 9.1).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit TXIE.
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.
8. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

TABLE 9-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|----------------------|--------|------------------------------|-------|-------|-------|-------|--------|--------|--------|--------------------|---------------------------|
| 0Bh, 8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | R0IF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 19h | TXREG | USART Transmit Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F870; always maintain these bits clear.

FIGURE 9-9: SYNCHRONOUS TRANSMISSION

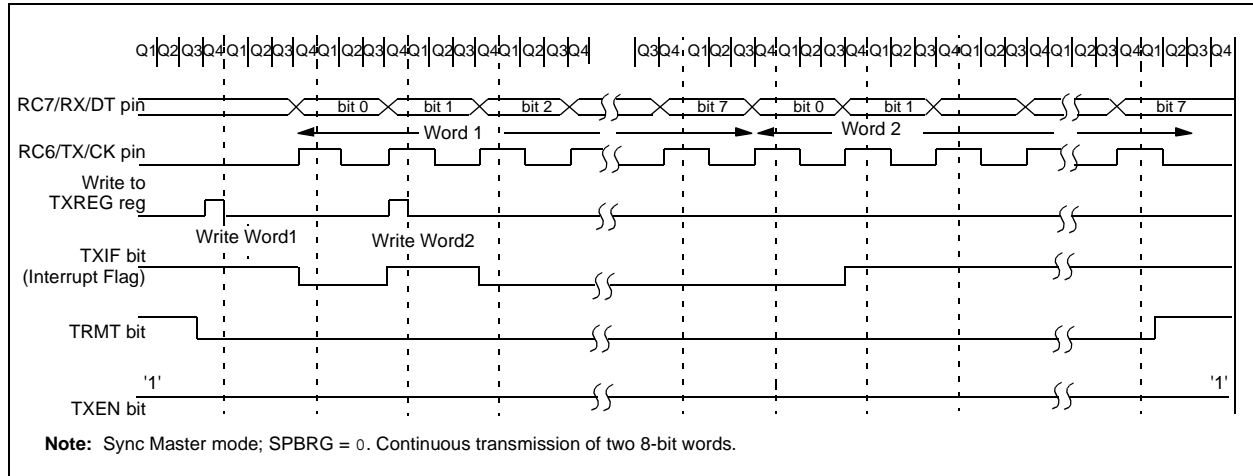
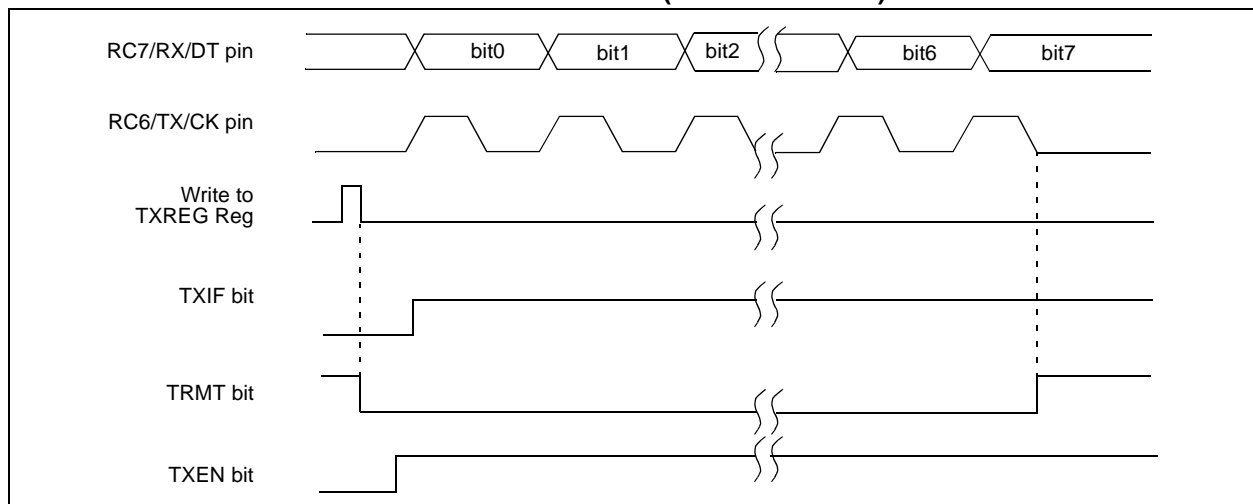


FIGURE 9-10: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)



PIC16F870/871

9.3.2 USART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either enable bit SREN (RCSTA<5>), or enable bit CREN (RCSTA<4>). Data is sampled on the RC7/RX/DT pin on the falling edge of the clock. If enable bit SREN is set, then only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, CREN takes precedence. After clocking the last bit, the received data in the Receive Shift register (RSR) is transferred to the RCREG register (if it is empty). When the transfer is complete, interrupt flag bit RCIF (PIR1<5>) is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1<5>). Flag bit RCIF is a read only bit, which is reset by the hardware. In this case, it is reset when the RCREG register has been read and is empty. The RCREG is a double-buffered register (i.e., it is a two-deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte to begin shifting into the RSR register. On the clocking of the last bit of the third byte, if the RCREG register is still full, then overrun error bit OERR (RCSTA<1>) is set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Bit OERR has to be cleared in software (by clearing bit CREN). If bit OERR is set, transfers from the RSR to the RCREG are inhibited, so it is essential to clear bit OERR if it is set. The 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.

When setting up a Synchronous Master Reception:

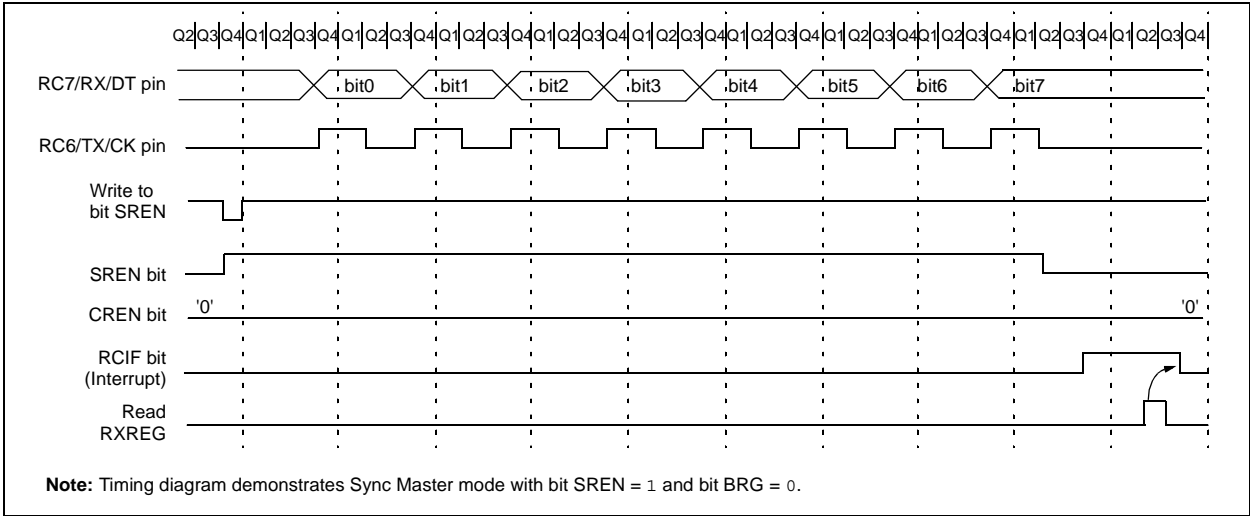
1. Initialize the SPBRG register for the appropriate baud rate (Section 9.1).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, then set enable bit RCIE.
5. If 9-bit reception is desired, then set bit RX9.
6. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
7. Interrupt flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
8. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG register.
10. If any error occurred, clear the error by clearing bit CREN.
11. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

TABLE 9-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|----------------------|--------|------------------------------|-------|-------|-------|-------|--------|--------|--------|--------------------|---------------------------|
| 0Bh, 8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | R0IF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 1Ah | RCREG | USART Receive Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master reception.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F870; always maintain these bits clear.

FIGURE 9-11: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



PIC16F870/871

9.4 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>).

9.4.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes is 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:

- The first word will immediately transfer to the TSR register and transmit.
- The second word will remain in TXREG register.
- Flag bit TXIF will not be set.
- 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.
- If enable bit TXIE is set, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

When setting up a Synchronous Slave Transmission, follow these steps:

- Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
- Clear bits CREN and SREN.
- If interrupts are desired, then set enable bit TXIE.
- If 9-bit transmission is desired, then set bit TX9.
- Enable the transmission by setting enable bit TXEN.
- If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- Start transmission by loading data to the TXREG register.
- If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

TABLE 9-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|----------------------|--------|------------------------------|-------|-------|-------|-------|--------|--------|--------|--------------------|---------------------------|
| 0Bh, 8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | ROIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 19h | TXREG | USART Transmit Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous slave transmission.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F870; always maintain these bits clear.

9.4.2 USART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of the SLEEP mode. Bit SREN is a “don't care” in Slave mode.

If receive is enabled by setting bit CREN prior to the SLEEP instruction, then a word may be received during SLEEP. On completely receiving the word, the RSR register will transfer the data to the RCREG register and if enable bit RCIE bit is set, the interrupt generated will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

When setting up a Synchronous Slave Reception, follow these steps:

1. Enable the synchronous master serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. If interrupts are desired, set enable bit RCIE.
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.
9. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

TABLE 9-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other RESETS |
|----------------------|--------|------------------------------|-------|-------|-------|-------|--------|--------|--------|--------------------|---------------------------|
| 0Bh, 8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | R0IF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 1Ah | RCREG | USART Receive Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous slave reception.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F870, always maintain these bits clear.

PIC16F870/871

NOTES:

10.0 ANALOG-TO-DIGITAL (A/D) CONVERTER MODULE

The Analog-to-Digital (A/D) Converter module has five inputs for the 28-pin devices and eight for the other devices.

The analog input charges a sample and hold capacitor. The output of the sample and hold capacitor is the input into the converter. The converter then generates a digital result of this analog level via successive approximation. The A/D conversion of the analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low voltage reference input that is software selectable to some combination of V_{DD}, V_{SS}, RA2, or RA3.

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 clock must be derived from the A/D's internal RC oscillator.

The A/D module has four registers. These registers are:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register0 (ADCON0)
- A/D Control Register1 (ADCON1)

The ADCON0 register, shown in Register 10-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 10-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be the 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).

REGISTER 10-1: 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/DONE | — | ADON |
| | | | | | | | bit 0 |
| | | | | | | | bit 7 |

bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits

- 00 = FOSC/2
- 01 = FOSC/8
- 10 = FOSC/32
- 11 = FRC (clock derived from the internal A/D module RC oscillator)

bit 5-3 **CHS2:CHS0:** Analog Channel Select bits

- 000 = Channel 0, (RA0/AN0)
- 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/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: These channels are not available on the PIC16F870 device.

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

PIC16F870/871

REGISTER 10-2: ADCON1 REGISTER (ADDRESS: 9Fh)

| | | | | | | | |
|-------|-----|-------|-----|-------|-------|-------|-------|
| U-0 | U-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| ADFM | — | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| bit 7 | | | | bit 0 | | | |

- bit 7 **ADFM:** A/D Result Format Select bit
 1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
 0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.
- bit 6-4 **Unimplemented:** Read as '0'
- bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits:

| PCFG3: PCFG0 | AN7 ⁽¹⁾ RE2 | AN6 ⁽¹⁾ RE1 | AN5 ⁽¹⁾ RE0 | AN4 RA5 | AN3 RA3 | AN2 RA2 | AN1 RA1 | AN0 RA0 | VREF+ | VREF- | CHAN/ Refs ⁽²⁾ |
|-----------------|---------------------------|---------------------------|---------------------------|------------|------------|------------|------------|------------|-------|-------|------------------------------|
| 0000 | A | A | A | A | A | A | A | A | VDD | VSS | 8/0 |
| 0001 | A | A | A | A | VREF+ | A | A | A | RA3 | VSS | 7/1 |
| 0010 | D | D | D | A | A | A | A | A | VDD | VSS | 5/0 |
| 0011 | D | D | D | A | VREF+ | A | A | A | RA3 | VSS | 4/1 |
| 0100 | D | D | D | D | A | D | A | A | VDD | VSS | 3/0 |
| 0101 | D | D | D | D | VREF+ | D | A | A | RA3 | VSS | 2/1 |
| 011x | D | D | D | D | D | D | D | D | VDD | VSS | 0/0 |
| 1000 | A | A | A | A | VREF+ | VREF- | A | A | RA3 | RA2 | 6/2 |
| 1001 | D | D | A | A | A | A | A | A | VDD | VSS | 6/0 |
| 1010 | D | D | A | A | VREF+ | A | A | A | RA3 | VSS | 5/1 |
| 1011 | D | D | A | A | VREF+ | VREF- | A | A | RA3 | RA2 | 4/2 |
| 1100 | D | D | D | A | VREF+ | VREF- | A | A | RA3 | RA2 | 3/2 |
| 1101 | D | D | D | D | VREF+ | VREF- | A | A | RA3 | RA2 | 2/2 |
| 1110 | D | D | D | D | D | D | D | A | VDD | VSS | 1/0 |
| 1111 | D | D | D | D | VREF+ | VREF- | D | A | RA3 | RA2 | 1/2 |

A = Analog input D = Digital I/O

- Note 1:** These channels are not available on the PIC16F870 device.
- 2:** This column indicates the number of analog channels available as A/D inputs and the number of analog channels used as voltage reference inputs.

| | | | |
|--------------------|------------------|------------------------------------|--------------------|
| 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 |

The ADRESH:ADRESL registers contain the 10-bit result of the A/D conversion. When the A/D conversion is complete, the result is loaded into this A/D result register pair, the GO/DONE bit (ADCON0<2>) is cleared and the A/D interrupt flag bit ADIF is set. The block diagram of the A/D module is shown in Figure 10-1.

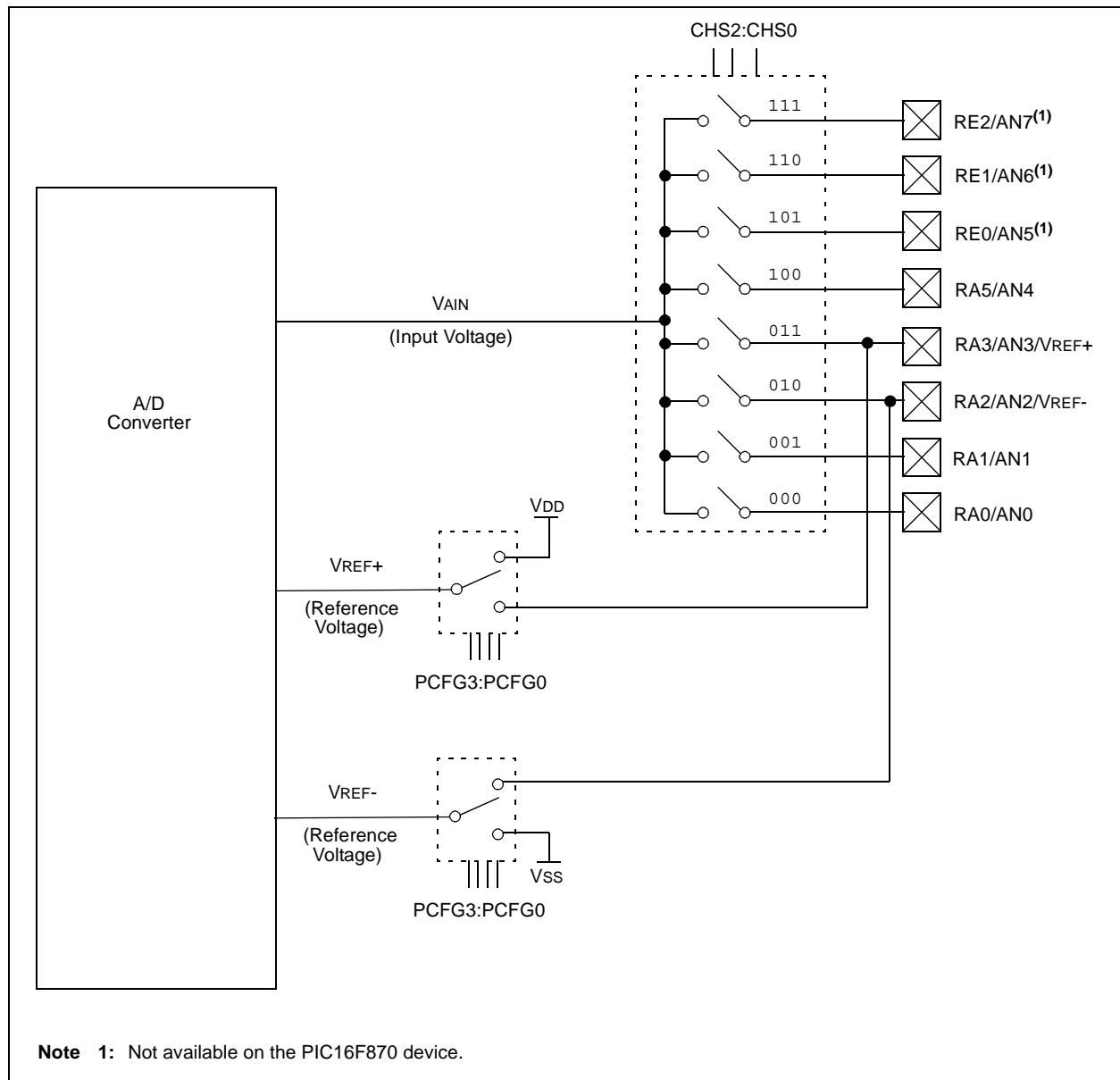
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

To determine sample time, see Section 10.1. After this acquisition time has elapsed, the A/D conversion can be started.

These steps should be followed for doing an A/D Conversion:

1. Configure the A/D module:
 - Configure analog pins/voltage reference and digital I/O (ADCON1)
 - Select A/D input channel (ADCON0)
 - Select A/D conversion clock (ADCON0)
 - Turn on A/D module (ADCON0)
2. Configure A/D interrupt (if desired):
 - Clear ADIF bit
 - Set ADIE bit
 - Set PEIE bit
 - Set GIE bit
3. Wait the required acquisition time.
4. Start conversion:
 - Set GO/DONE bit (ADCON0)
5. Wait for A/D conversion to complete, by either:
 - Polling for the GO/DONE bit to be cleared (with interrupts enabled); OR
 - Waiting for the A/D interrupt
6. Read A/D Result register pair (ADRESH:ADRESL), clear bit ADIF if required.
7. For the next conversion, go to step 1 or step 2, as required. The A/D conversion time per bit is defined as T_{AD}. A minimum wait of 2 T_{AD} is required before the next acquisition starts.

FIGURE 10-1: A/D BLOCK DIAGRAM



PIC16F870/871

10.1 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 10-2. The source impedance (R_S) and the internal sampling switch (R_{SS}) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (R_{SS}) impedance varies over the device voltage (V_{DD}), see Figure 10-2. **The maximum recommended impedance for analog sources is 10 k Ω .** As the impedance is decreased, the acquisition time may

be decreased. After the analog input channel is selected (changed), this acquisition must be done before the conversion can be started.

To calculate the minimum acquisition time, Equation 10-1 may be used. This equation assumes that 1/2 LSB error is used (1024 steps for the A/D). The 1/2 LSB error is the maximum error allowed for the A/D to meet its specified resolution.

To calculate the minimum acquisition time, T_{ACQ} , see the PIC[®] Mid-Range MCU Family Reference Manual (DS33023).

EQUATION 10-1: ACQUISITION TIME

$$\begin{aligned}
 T_{ACQ} &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\
 &= T_{AMP} + T_C + T_{COFF} \\
 &= 2 \mu\text{s} + T_C + [(\text{Temperature} - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})] \\
 T_C &= \text{CHOLD} (R_{IC} + R_{SS} + R_S) \ln(1/2047) \\
 &= -120 \text{ pF} (1 \text{ k}\Omega + 7 \text{ k}\Omega + 10 \text{ k}\Omega) \ln(0.0004885) \\
 &= 16.47 \mu\text{s} \\
 T_{ACQ} &= 2 \mu\text{s} + 16.47 \mu\text{s} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})] \\
 &= 19.72 \mu\text{s}
 \end{aligned}$$

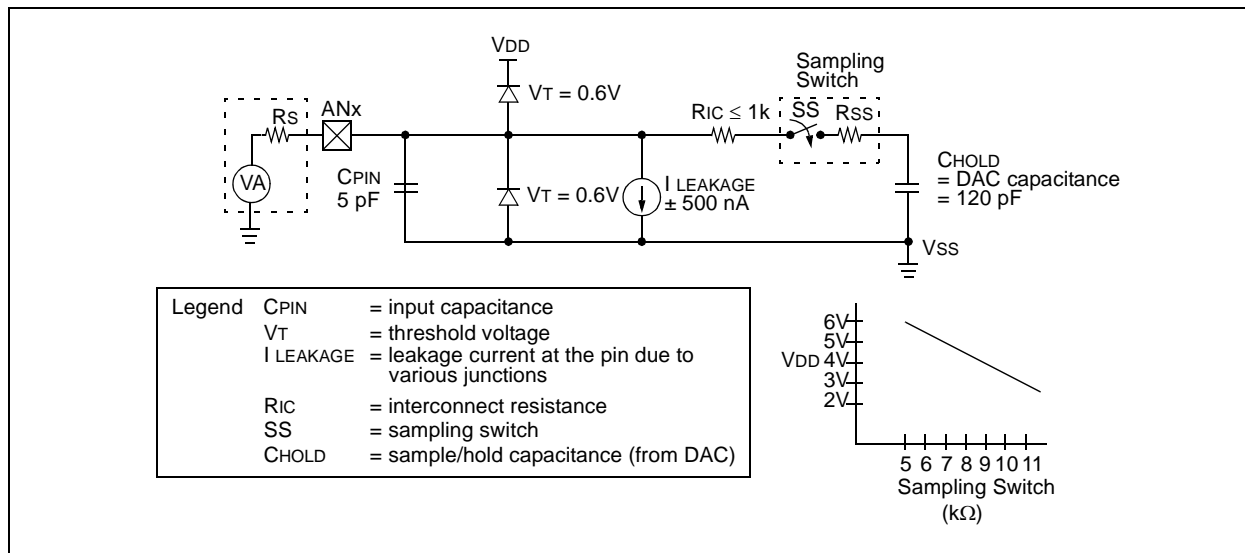
Note 1: The reference voltage (V_{REF}) has no effect on the equation, since it cancels itself out.

2: The charge holding capacitor (CHOLD) is not discharged after each conversion.

3: The maximum recommended impedance for analog sources is 10 k Ω . This is required to meet the pin leakage specification.

4: After a conversion has completed, a 2.0 T_{AD} delay must complete before acquisition can begin again. During this time, the holding capacitor is not connected to the selected A/D input channel.

FIGURE 10-2: ANALOG INPUT MODEL



10.2 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires a minimum 12 TAD per 10-bit conversion. The source of the A/D conversion clock is software selected. The four possible options for TAD are:

- 2 TOSC
- 8 TOSC
- 32 TOSC
- Internal A/D module RC oscillator (2-6 μ s)

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 μ s.

Table 10-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

TABLE 10-1: TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (C))

| AD Clock Source (TAD) | | Maximum Device Frequency |
|-------------------------|-------------|--------------------------|
| Operation | ADCS1:ADCS0 | Max. |
| 2 TOSC | 00 | 1.25 MHz |
| 8 TOSC | 01 | 5 MHz |
| 32 TOSC | 10 | 20 MHz |
| RC ^(1, 2, 3) | 11 | (Note 1) |

- Note 1:** The RC source has a typical TAD time of 4 μ s, but can vary between 2-6 μ s.
- 2:** When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for SLEEP operation.
- 3:** For extended voltage devices (LC), please refer to the Electrical Characteristics (Section 14.1 and 14.2).

10.3 Configuring Analog Port Pins

The ADCON1 and TRIS 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, any pin configured as an analog input channel 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 (including the AN7:AN0 pins), may cause the input buffer to consume current that is out of the device specifications.

PIC16F870/871

10.4 A/D Conversions

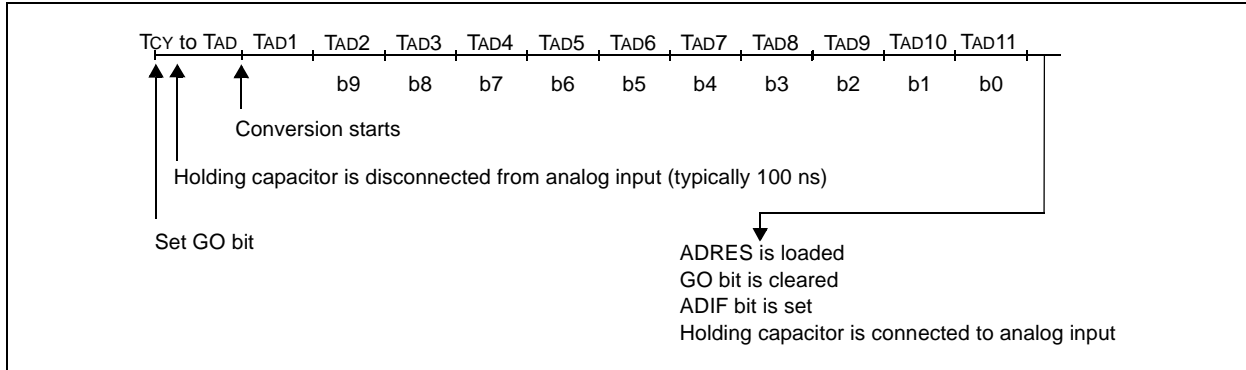
Clearing the $\overline{\text{GO/DONE}}$ bit during a conversion will abort the current conversion. The A/D result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers). After the A/D conversion is aborted, a 2 TAD wait is required before the next

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

In Figure 10-3, after the GO bit is set, the first time segment has a minimum of T_{CY} and a maximum of T_{AD}.

Note: The $\overline{\text{GO/DONE}}$ bit should **NOT** be set in the same instruction that turns on the A/D.

FIGURE 10-3: A/D CONVERSION TAD CYCLES

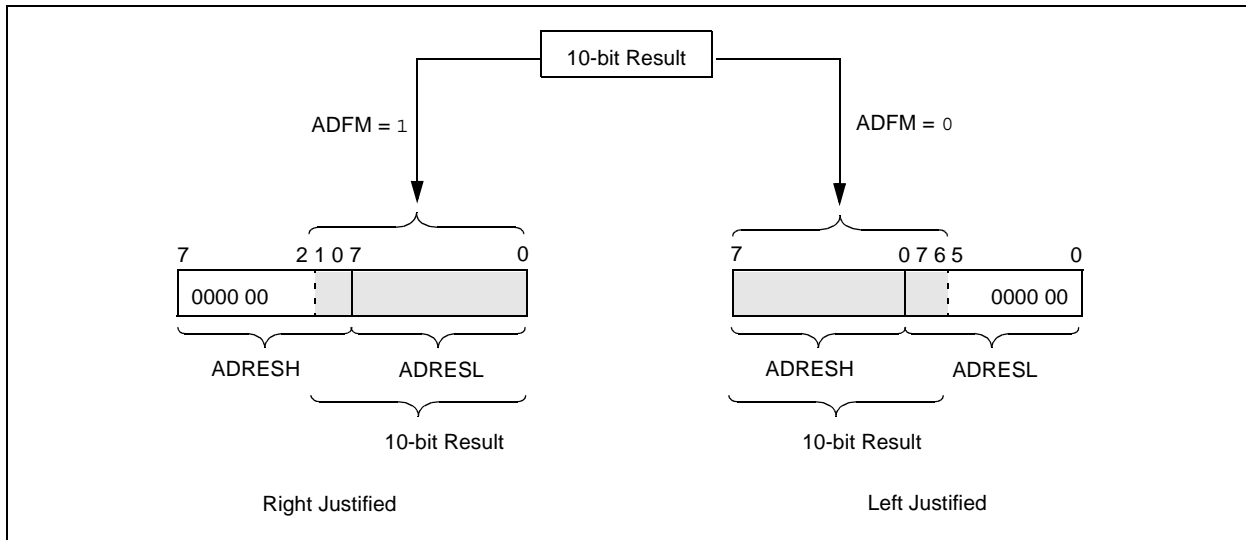


10.4.1 A/D RESULT REGISTERS

The ADRESH:ADRESL register pair is the location where the 10-bit A/D result is loaded at the completion of the A/D conversion. This register pair is 16-bits wide. The A/D module gives the flexibility to left or right justify the 10-bit result in the 16-bit result register. The A/D For-

mat Select bit (ADFM) controls this justification. Figure 10-4 shows the operation of the A/D result justification. The extra bits are loaded with '0'. When an A/D result will not overwrite these locations (A/D disable), these registers may be used as two general purpose 8-bit registers.

FIGURE 10-4: A/D RESULT JUSTIFICATION



10.5 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/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 allow the conversion to occur during SLEEP, ensure the SLEEP instruction immediately follows the instruction that sets the GO/DONE bit.

10.6 Effects of a RESET

A device RESET forces all registers to their RESET state. This forces the A/D module to be turned off, and any conversion is aborted. All A/D input pins are configured as analog inputs.

The value that is in the ADRESH:ADRESL registers is not modified for a Power-on Reset. The ADRESH:ADRESL registers will contain unknown data after a Power-on Reset.

TABLE 10-2: REGISTERS/BITS ASSOCIATED WITH A/D

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on MCLR, WDT |
|--------------------|--------|-------------------------------|-------|---|---------|-------|---------------------------|--------|--------|-------------------|--------------------|
| 0Bh,8Bh,10Bh,18Bh | INTCON | GIE | PEIE | TOIE | INTE | RBIE | TOIF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | — | CCP1IF | TMR2IF | TMR1IF | 0000 -000 | 0000 -000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | — | CCP1IE | TMR2IE | TMR1IE | 0000 -000 | 0000 -000 |
| 1Eh | ADRESH | A/D Result Register High Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 9Eh | ADRESL | A/D Result Register Low Byte | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Fh | ADCON0 | ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/DONE | — | ADON | 0000 00-0 | 0000 00-0 |
| 9Fh | ADCON1 | ADFM | — | — | — | PCFG3 | PCFG2 | PCFG1 | PCFG0 | --0- 0000 | --0- 0000 |
| 85h | TRISA | — | — | PORTA Data Direction Register | | | | | | --11 1111 | --11 1111 |
| 05h | PORTA | — | — | PORTA Data Latch when written: PORTA pins when read | | | | | | --0x 0000 | --0u 0000 |
| 89h ⁽¹⁾ | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction bits | | | 0000 -111 | 0000 -111 |
| 09h ⁽¹⁾ | PORTE | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used for A/D conversion.

Note 1: These registers/bits are not available on the 28-pin devices.

PIC16F870/871

NOTES:

11.0 SPECIAL FEATURES OF THE CPU

The PIC16F870/871 devices have a host of 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
- Low Voltage In-Circuit Serial Programming
- In-Circuit Debugger

PIC16F870/871 devices have 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. It 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, Watchdog Timer Wake-up, or through an interrupt.

Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost while the LP crystal option saves power. A set of configuration bits is used to select various options.

Additional information on special features is available in the PIC[®] Mid-Range MCU Family Reference Manual (DS33023).

11.1 Configuration Bits

The configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1'), to select various device configurations. The erased, or unprogrammed value of the configuration word is 3FFFh. These bits are mapped in program memory location 2007h.

It is important to note that address 2007h is beyond the user program memory space, which can be accessed only during programming.

PIC16F870/871

REGISTER 11-1: CONFIGURATION WORD (ADDRESS 2007h)⁽¹⁾

| | | | | | | | | | | | | | |
|-----|-----|-------|---|-----|-----|-----|-------|-----|-----|--------|--------|-------|-------|
| CP1 | CP0 | DEBUG | — | WRT | CPD | LVP | BOREN | CP1 | CP0 | PWRTEN | WDTEN | FOSC1 | FOSC0 |
| | | | | | | | | | | | bit 13 | bit 0 | |

| | |
|-----------------------|--|
| bit 13-12, bit 5-4 | CP1:CP0: FLASH Program Memory Code Protection bits ⁽²⁾ 11 = Code protection off 10 = Not supported 01 = Not supported 00 = Code protection on |
| bit 11 | DEBUG: In-Circuit Debugger Mode 1 = In-Circuit Debugger disabled, RB6 and RB7 are general purpose I/O pins 0 = In-Circuit Debugger enabled, RB6 and RB7 are dedicated to the debugger |
| bit 10 | Unimplemented: Read as '1' |
| bit 9 | WRT: FLASH Program Memory Write Enable 1 = Unprotected program memory may be written to by EECON control 0 = Unprotected program memory may not be written to by EECON control |
| bit 8 | CPD: Data EE Memory Code Protection 1 = Code protection off 0 = Data EEPROM memory code protected |
| bit 7 | LVP: Low Voltage In-Circuit Serial Programming Enable bit 1 = RB3/PGM pin has PGM function, low voltage programming enabled 0 = RB3 is digital I/O, HV on MCLR must be used for programming |
| bit 6 | BOREN: Brown-out Reset Enable bit ⁽³⁾ 1 = BOR enabled 0 = BOR disabled |
| bit 3 | PWRTEN: Power-up Timer Enable bit ⁽³⁾ 1 = PWRT disabled 0 = PWRT enabled |
| bit 2 | WDTEN: 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 |

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 1:** The erased (unprogrammed) value of the configuration word is 3FFFh.
- 2:** All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed.
- 3:** Enabling Brown-out Reset automatically enables Power-up Timer (PWRT), regardless of the value of bit PWRTEN. Ensure the Power-up Timer is enabled any time Brown-out Reset is enabled.

11.2 Oscillator Configurations

11.2.1 OSCILLATOR TYPES

The PIC16F870/871 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

11.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKI and OSC2/CLKO pins to establish oscillation (Figure 11-1). The PIC16F870/871 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/CLKI pin (Figure 11-2).

FIGURE 11-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)

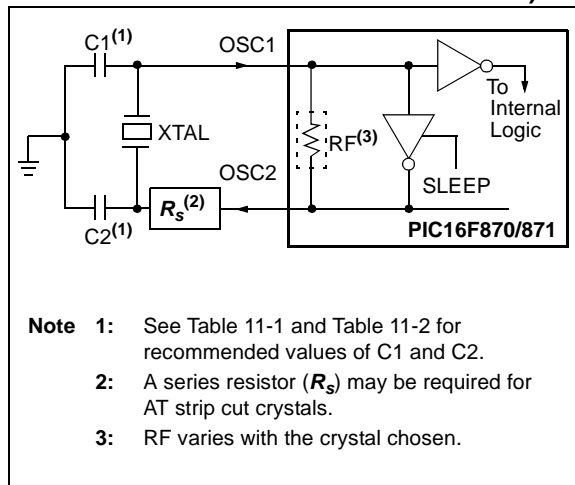


FIGURE 11-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

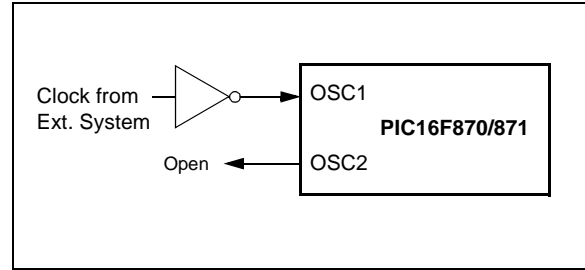


TABLE 11-1: CERAMIC RESONATORS

| Ranges Tested: | | | |
|--|------------------------|-------------|-------------|
| Mode | Freq. | OSC1 | OSC2 |
| XT | 455 kHz | 68 - 100 pF | 68 - 100 pF |
| | 2.0 MHz | 15 - 68 pF | 15 - 68 pF |
| | 4.0 MHz | 15 - 68 pF | 15 - 68 pF |
| HS | 8.0 MHz | 10 - 68 pF | 10 - 68 pF |
| | 16.0 MHz | 10 - 22 pF | 10 - 22 pF |
| These values are for design guidance only. See notes following Table 11-2. | | | |
| 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% | |
| All resonators used did not have built-in capacitors. | | | |

PIC16F870/871

TABLE 11-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

| Osc Type | Crystal Freq. | Cap. Range C1 | Cap. Range C2 |
|----------|---------------|---------------|---------------|
| LP | 32 kHz | 33 pF | 33 pF |
| | 200 kHz | 15 pF | 15 pF |
| XT | 200 kHz | 47-68 pF | 47-68 pF |
| | 1 MHz | 15 pF | 15 pF |
| | 4 MHz | 15 pF | 15 pF |
| HS | 4 MHz | 15 pF | 15 pF |
| | 8 MHz | 15-33 pF | 15-33 pF |
| | 20 MHz | 15-33 pF | 15-33 pF |

These values are for design guidance only.

See notes following this table.

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 |

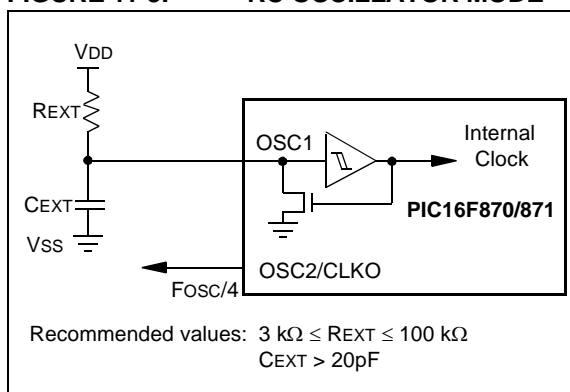
Note 1: Higher capacitance increases the stability of oscillator, but also increases the start-up time.

- 2: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
- 3: R_s may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.
- 4: When migrating from other PIC® devices, oscillator performance should be verified.

11.2.3 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 (R_{EXT}) and capacitor (C_{EXT}) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low C_{EXT} values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 11-3 shows how the R/C combination is connected to the PIC16F870/871.

FIGURE 11-3: RC OSCILLATOR MODE



11.3 RESET

The PIC16F870/871 differentiates between various kinds of RESET:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset during normal operation
- $\overline{\text{MCLR}}$ Reset during SLEEP
- WDT Reset (during normal operation)
- WDT Wake-up (during SLEEP)
- 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 Power-on Reset (POR), on the $\overline{\text{MCLR}}$ and WDT Reset, on $\overline{\text{MCLR}}$ Reset during

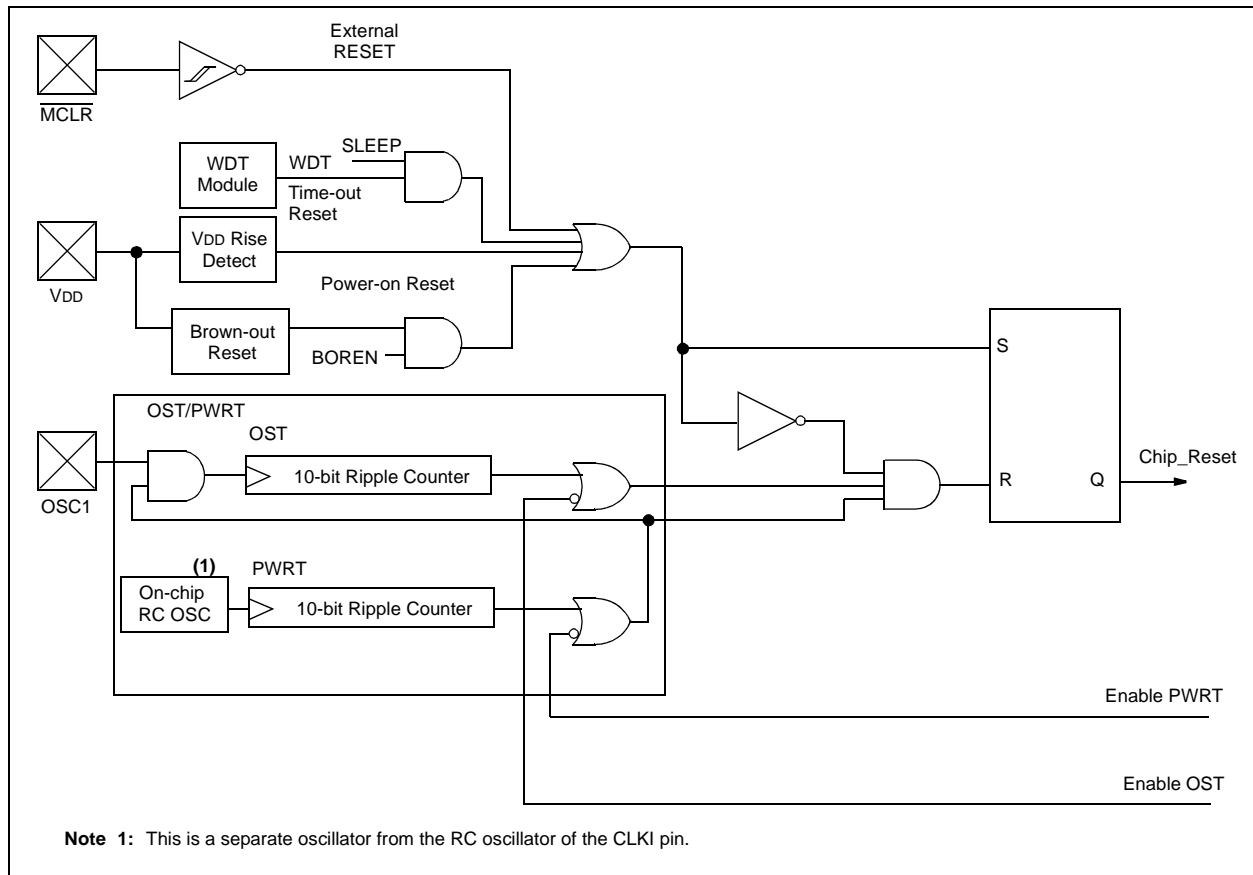
SLEEP, and Brown-out Reset (BOR). They are not affected by a WDT Wake-up, which is viewed as the resumption of normal operation. The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different RESET situations, as indicated in Table 11-4. These bits are used in software to determine the nature of the RESET. See Table 11-6 for a full description of RESET states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 11-4.

These devices have a $\overline{\text{MCLR}}$ noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.

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

FIGURE 11-4: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



PIC16F870/871

11.4 Power-on Reset (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected (in the range of 1.2V - 1.7V). To take advantage of the POR, tie the $\overline{\text{MCLR}}$ pin directly (or through a resistor) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for VDD is specified. See Electrical Specifications for details.

When the device starts normal operation (exits the RESET condition), device operating parameters (voltage, frequency, temperature,...) must be met to ensure operation. If these conditions are not met, the device must be held in RESET until the operating conditions are met. Brown-out Reset may be used to meet the start-up conditions. For additional information, refer to Application Note, AN007, "Power-up Trouble Shooting" (DS00007).

11.5 Power-up Timer (PWRT)

The Power-up Timer provides a fixed 72 ms nominal time-out on power-up only from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as the PWRT is active. The PWRT's time delay allows VDD to rise to an acceptable level. A configuration bit is provided to enable/disable the PWRT.

The power-up time delay will vary from chip to chip due to VDD, temperature and process variation. See DC parameters for details (TPWRT, parameter #33).

11.6 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides a delay of 1024 oscillator cycles (from OSC1 input) after the PWRT delay is over (if PWRT is 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.

11.7 Brown-out Reset (BOR)

The configuration bit, BOREN, 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 100 μS), 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 72 ms). 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.

11.8 Time-out Sequence

On power-up, the time-out sequence is as follows: The PWRT delay starts (if enabled) when a POR Reset 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 $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Bringing $\overline{\text{MCLR}}$ high will begin execution immediately. This is useful for testing purposes or to synchronize more than one PIC16F870/871 device operating in parallel.

Table 11-5 shows the RESET conditions for the STATUS, PCON and PC registers, while Table 11-6 shows the RESET conditions for all the registers.

11.9 Power Control/Status Register (PCON)

The Power Control/Status Register, PCON, has up to two bits depending upon the device.

Bit0 is Brown-out Reset Status bit, $\overline{\text{BOR}}$. Bit $\overline{\text{BOR}}$ is unknown on a Power-on Reset. It must then be set by the user and checked on subsequent RESETS to see if bit $\overline{\text{BOR}}$ cleared, indicating a BOR occurred. When the Brown-out Reset is disabled, the state of the $\overline{\text{BOR}}$ bit is unpredictable and is, therefore, not valid at any time.

Bit1 is $\overline{\text{POR}}$ (Power-on Reset Status bit). It is cleared on a Power-on Reset and unaffected otherwise. The user must set this bit following a Power-on Reset.

TABLE 11-3: TIME-OUT IN VARIOUS SITUATIONS

| Oscillator Configuration | Power-up | | Brown-out | Wake-up from SLEEP |
|--------------------------|-------------------------------|-------------------------------|-------------------|--------------------|
| | $\overline{\text{PWRTE}} = 0$ | $\overline{\text{PWRTE}} = 1$ | | |
| XT, HS, LP | 72 ms + 1024 TOSC | 1024 TOSC | 72 ms + 1024 TOSC | 1024 TOSC |
| RC | 72 ms | — | 72 ms | — |

TABLE 11-4: STATUS BITS AND THEIR SIGNIFICANCE

| POR | BOR | TO | 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 11-5: 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 | 0001 1uuu | ---- --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).

TABLE 11-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS

| Register | Devices | | Power-on Reset, Brown-out Reset | \overline{MCLR} Resets WDT Reset | Wake-up via WDT or Interrupt |
|----------|-----------|-----------|---------------------------------|------------------------------------|------------------------------|
| W | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| INDF | PIC16F870 | PIC16F871 | N/A | N/A | N/A |
| TMR0 | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PCL | PIC16F870 | PIC16F871 | 0000h | 0000h | PC + 1 ⁽²⁾ |
| STATUS | PIC16F870 | PIC16F871 | 0001 1xxx | 000q quuu ⁽³⁾ | uuuq quuu ⁽³⁾ |
| FSR | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTA | PIC16F870 | PIC16F871 | --0x 0000 | --0u 0000 | --uu uuuu |
| PORTB | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTC | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTD | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTE | PIC16F870 | PIC16F871 | ---- -xxx | ---- -uuu | ---- -uuu |
| PCLATH | PIC16F870 | PIC16F871 | ---0 0000 | ---0 0000 | ---u uuuu |
| INTCON | PIC16F870 | PIC16F871 | 0000 000x | 0000 000u | uuuu uuuu ⁽¹⁾ |

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition, r = reserved, maintain clear

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 11-5 for RESET value for specific condition.

PIC16F870/871

TABLE 11-6: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

| Register | Devices | | Power-on Reset, Brown-out Reset | MCLR Resets WDT Reset | Wake-up via WDT or Interrupt |
|------------|-----------|-----------|------------------------------------|--------------------------|---------------------------------|
| PIR1 | PIC16F870 | PIC16F871 | r000 -000 | r000 -000 | ruuu -uuu ⁽¹⁾ |
| | PIC16F870 | PIC16F871 | 0000 -000 | 0000 -000 | uuuu -uuu ⁽¹⁾ |
| PIR2 | PIC16F870 | PIC16F871 | ---0 ---- | ---0 ---- | ---u ---- ⁽¹⁾ |
| TMR1L | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| TMR1H | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| T1CON | PIC16F870 | PIC16F871 | --00 0000 | --uu uuuu | --uu uuuu |
| TMR2 | PIC16F870 | PIC16F871 | 0000 0000 | 0000 0000 | uuuu uuuu |
| T2CON | PIC16F870 | PIC16F871 | -000 0000 | -000 0000 | -uuu uuuu |
| CCPR1L | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCPR1H | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCP1CON | PIC16F870 | PIC16F871 | --00 0000 | --00 0000 | --uu uuuu |
| RCSTA | PIC16F870 | PIC16F871 | 0000 000x | 0000 000x | uuuu uuuu |
| TXREG | PIC16F870 | PIC16F871 | 0000 0000 | 0000 0000 | uuuu uuuu |
| RCREG | PIC16F870 | PIC16F871 | 0000 0000 | 0000 0000 | uuuu uuuu |
| ADRESH | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADCON0 | PIC16F870 | PIC16F871 | 0000 00-0 | 0000 00-0 | uuuu uu-u |
| OPTION_REG | PIC16F870 | PIC16F871 | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISA | PIC16F870 | PIC16F871 | --11 1111 | --11 1111 | --uu uuuu |
| TRISB | PIC16F870 | PIC16F871 | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISC | PIC16F870 | PIC16F871 | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISD | PIC16F870 | PIC16F871 | 1111 1111 | 1111 1111 | uuuu uuuu |
| TRISE | PIC16F870 | PIC16F871 | 0000 -111 | 0000 -111 | uuuu -uuu |
| PIE1 | PIC16F870 | PIC16F871 | r000 -000 | r000 -000 | ruuu -uuu |
| | PIC16F870 | PIC16F871 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PIE2 | PIC16F870 | PIC16F871 | ---0 ---- | ---0 ---- | ---u ---- |
| PCON | PIC16F870 | PIC16F871 | ---- -qg | ---- -uu | ---- -uu |
| PR2 | PIC16F870 | PIC16F871 | 1111 1111 | 1111 1111 | 1111 1111 |
| TXSTA | PIC16F870 | PIC16F871 | 0000 -010 | 0000 -010 | uuuu -uuu |
| SPBRG | PIC16F870 | PIC16F871 | 0000 0000 | 0000 0000 | uuuu uuuu |
| ADRESL | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| ADCON1 | PIC16F870 | PIC16F871 | 0--- 0000 | 0--- 0000 | u--- uuuu |
| EEDATA | PIC16F870 | PIC16F871 | 0--- 0000 | 0--- 0000 | u--- uuuu |
| EEADR | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| EEDATH | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| EEADRH | PIC16F870 | PIC16F871 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| EECON1 | PIC16F870 | PIC16F871 | x--- x000 | u--- u000 | u--- uuuu |
| EECON2 | PIC16F870 | PIC16F871 | ---- ---- | ---- ---- | ---- ---- |

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition, r = reserved, maintain clear

- 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 11-5 for RESET value for specific condition.

FIGURE 11-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO V_{DD})

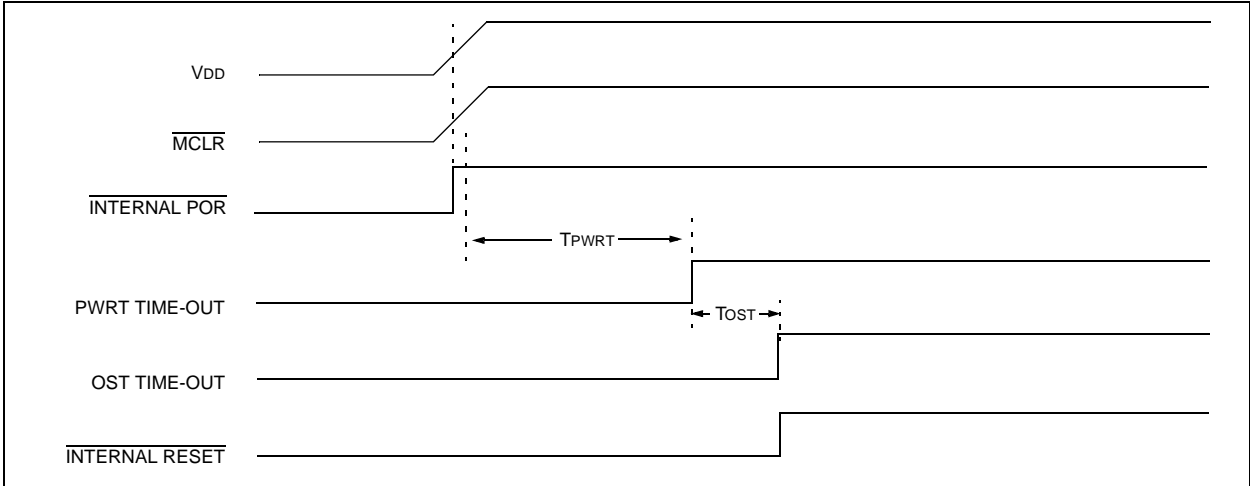


FIGURE 11-6: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 1

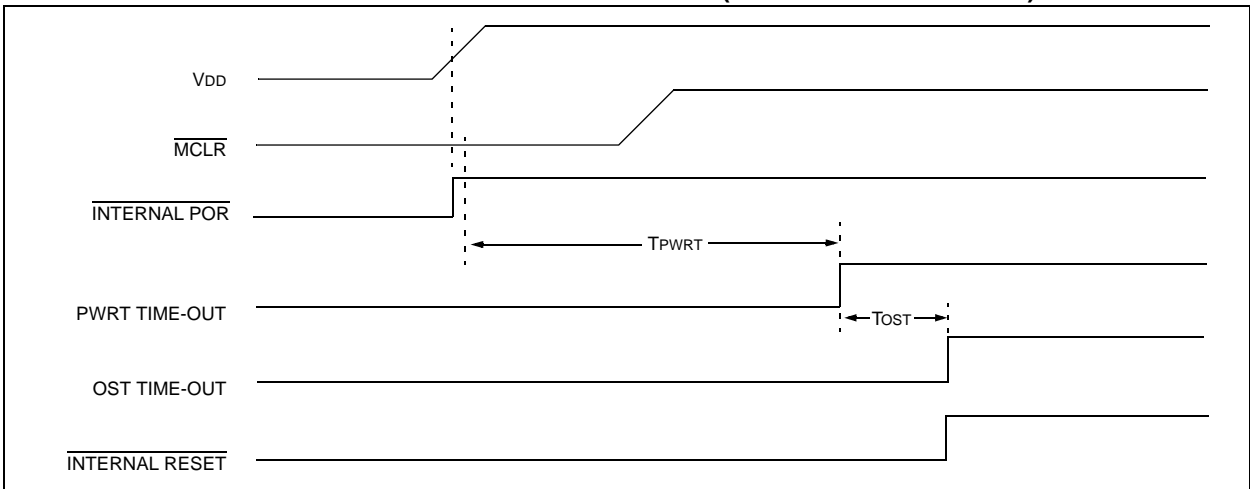
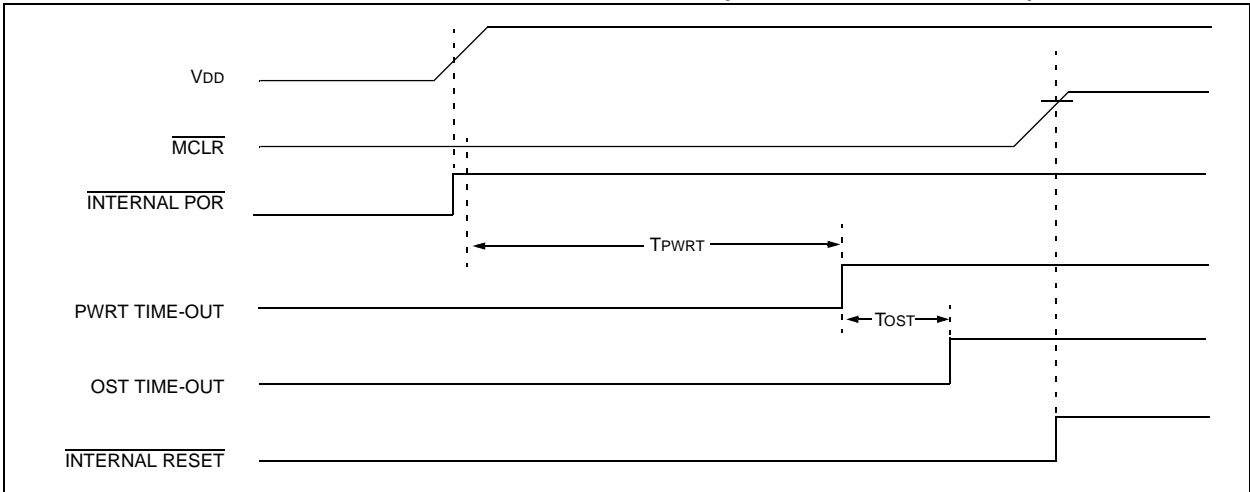
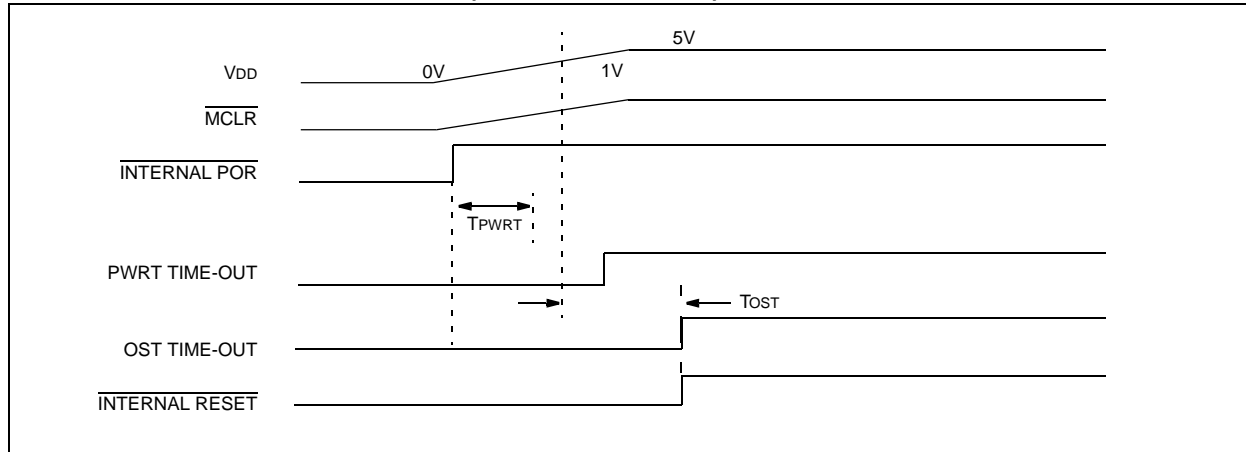


FIGURE 11-7: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 2



PIC16F870/871

FIGURE 11-8: SLOW RISE TIME ($\overline{\text{MCLR}}$ TIED TO V_{DD})



11.10 Interrupts

The PIC16F870/871 family has up to 14 sources of interrupt. The Interrupt Control register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

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

A global interrupt enable bit, GIE (INTCON<7>), enables (if set) all 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.

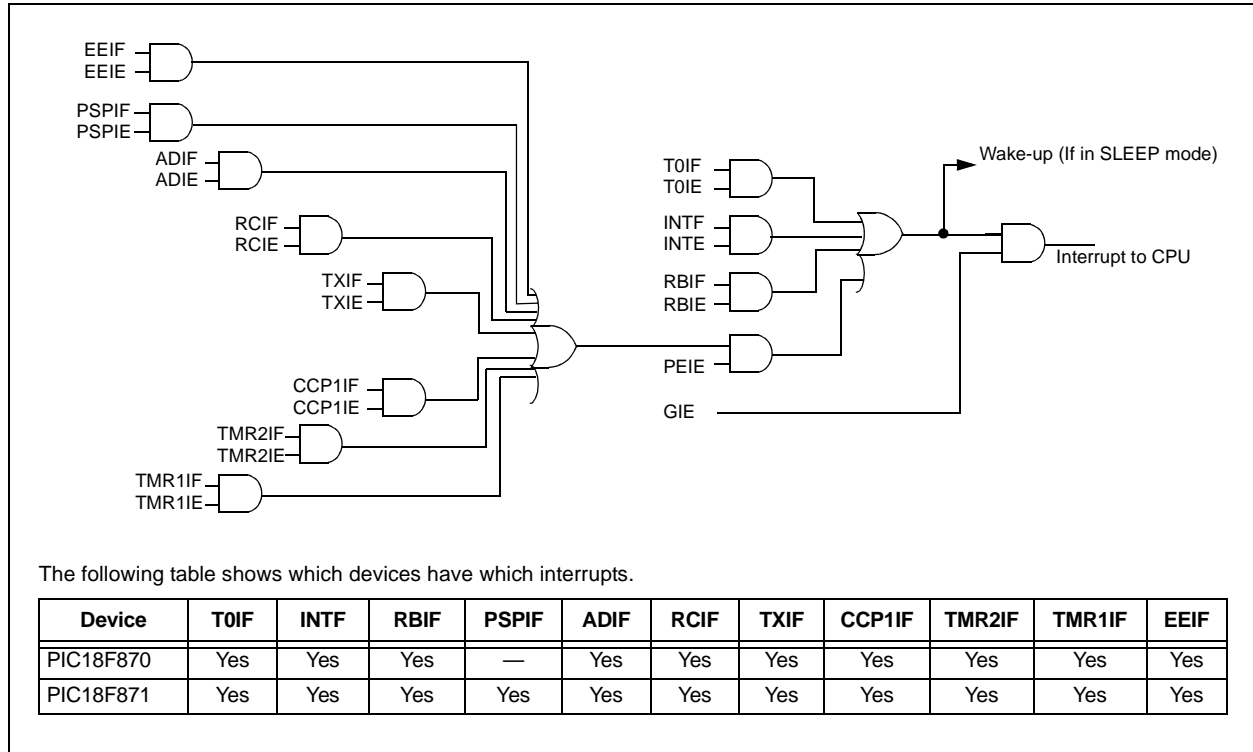
The RB0/INT pin interrupt, the RB port change interrupt, and the TMR0 overflow interrupt flags are contained in the INTCON register.

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 GIE bit.

FIGURE 11-9: INTERRUPT LOGIC



11.10.1 INT INTERRUPT

External interrupt on the 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 11.13 for details on SLEEP mode.

11.10.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set flag bit T0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit T0IE (INTCON<5>) (Section 5.0).

11.10.3 PORTB INTCON CHANGE

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<4>) (Section 4.2).

PIC16F870/871

11.11 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt, (i.e., W register and STATUS register). This will have to be implemented in software.

For the PIC16F870/871 devices, the register W_TEMP must be defined in both banks 0 and 1 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 registers, PCLATH_TEMP and STATUS_TEMP, are only defined in bank 0.

Since the upper 16 bytes of each bank are common in the PIC16F870/871 devices, temporary holding registers W_TEMP, STATUS_TEMP, and PCLATH_TEMP should be placed in here. These 16 locations don't require banking and therefore, make it easier for context save and restore. The same code shown in Example 11-1 can be used.

EXAMPLE 11-1: SAVING STATUS, W, AND PCLATH REGISTERS IN RAM

```
MOVWF    W_TEMP          ;Copy W to TEMP register
SWAPF    STATUS,W        ;Swap status to be saved into W
CLRF     STATUS          ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF    STATUS_TEMP     ;Save status to bank zero STATUS_TEMP register
MOVF     PCLATH, W       ;Only required if using pages 1, 2 and/or 3
MOVWF    PCLATH_TEMP     ;Save PCLATH into W
CLRF     PCLATH          ;Page zero, regardless of current page
:
:(ISR)           ;(Insert user code 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
```

11.12 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/CLKI pin. That means that the WDT will run, even if the clock on the OSC1/CLKI and OSC2/CLKO pins of the device has been stopped, for example, by execution of a SLEEP instruction.

During normal operation, a WDT time-out generates a device RESET (Watchdog Timer Reset). If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer Wake-up). The TO bit in the STATUS register will be cleared upon a Watchdog Timer time-out.

The WDT can be permanently disabled by clearing configuration bit WDTEN (Section 11.1).

WDT time-out period values may be found in the Electrical Specifications section under parameter #31. Values for the WDT prescaler (actually a postscaler, but shared with the Timer0 prescaler) may be assigned using the OPTION_REG register.

Note 1: The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET condition.

2: 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 11-10: WATCHDOG TIMER BLOCK DIAGRAM

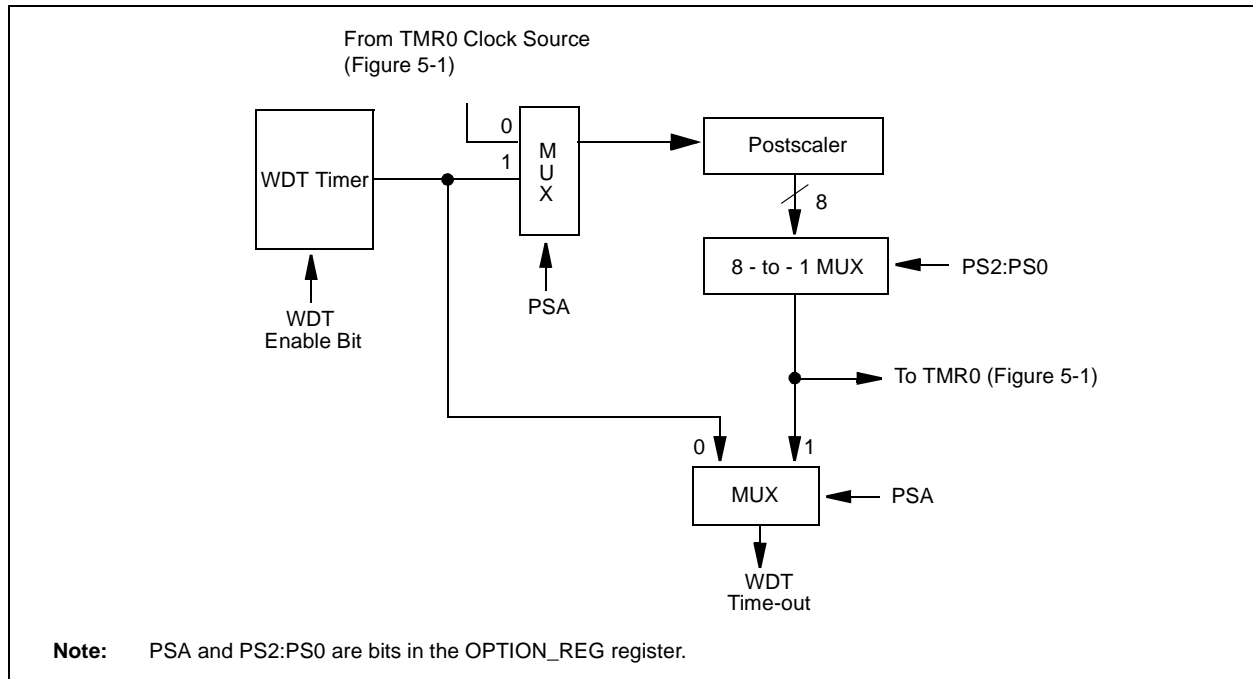


TABLE 11-7: SUMMARY OF WATCHDOG TIMER REGISTERS

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|----------|--------------|-------|----------------------|-------|-------|-----------------------|-------|-------|-------|
| 2007h | Config. bits | (1) | BOREN ⁽¹⁾ | CP1 | CP0 | PWRTEN ⁽¹⁾ | WDTEN | FOSC1 | FOSC0 |
| 81h,181h | OPTION_REG | RBPU | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |

Legend: Shaded cells are not used by the Watchdog Timer.

Note 1: See Register 11-1 for operation of these bits.

11.13 Power-down Mode (SLEEP)

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

If enabled, the Watchdog Timer will be cleared but keeps running, the 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).

11.13.1 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 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. PSP read or write (PIC16F874/877 only).
2. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
3. CCP Capture mode interrupt.
4. Special event trigger (Timer1 in Asynchronous mode using an external clock).
5. SSP (START/STOP) bit detect interrupt.
6. SSP transmit or receive in Slave mode (SPI/I²C).
7. USART RX or TX (Synchronous Slave mode).
8. A/D conversion (when A/D clock source is RC).
9. EEPROM write operation completion

Other peripherals cannot generate interrupts, since during SLEEP, no on-chip 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.

11.13.2 WAKE-UP USING INTERRUPTS

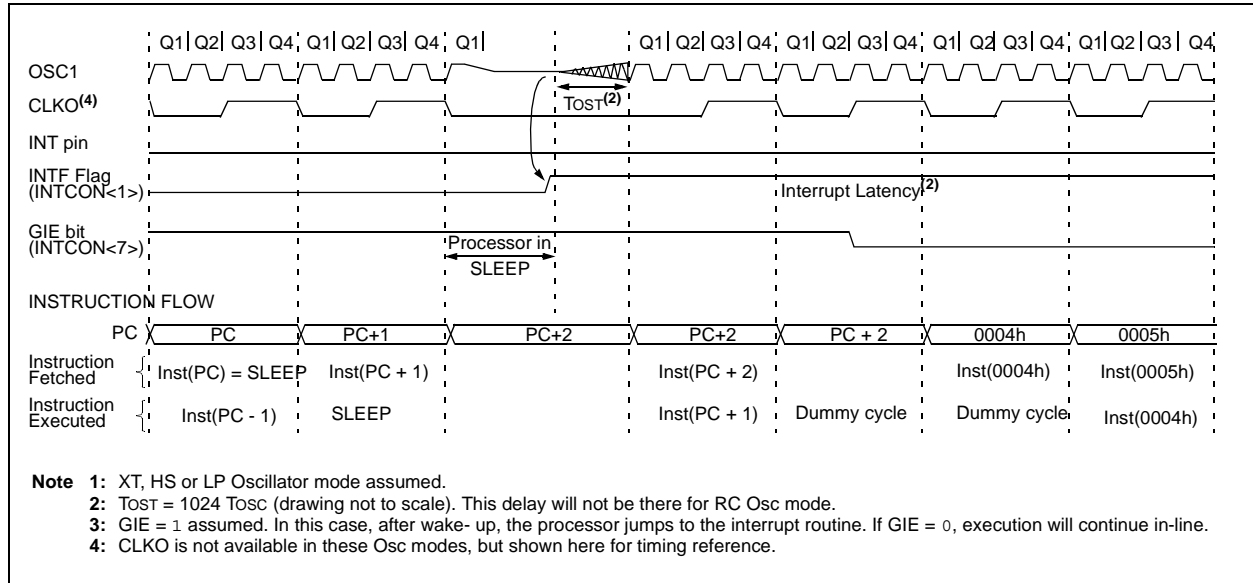
When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a `SLEEP` instruction, the `SLEEP` instruction will complete as a `NOP`. Therefore, the WDT and WDT postscaler will not be cleared, the \overline{TO} bit will not be set and \overline{PD} bits will not be cleared.
- If the interrupt occurs **during or after** the execution of a `SLEEP` instruction, the device will immediately wake-up from SLEEP. The `SLEEP` instruction will be completely executed before the wake-up. Therefore, the WDT and WDT postscaler will be cleared, the \overline{TO} bit will be set and the \overline{PD} bit will be cleared.

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

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

FIGURE 11-11: WAKE-UP FROM SLEEP THROUGH INTERRUPT



11.14 In-Circuit Debugger

When the DEBUG bit in the configuration word is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB® ICD. When the microcontroller has this feature enabled, some of the resources are not available for general use. Table 11-8 shows which features are consumed by the background debugger.

TABLE 11-8: DEBUGGER RESOURCES

| | |
|----------------|--|
| I/O pins | RB6, RB7 |
| Stack | 1 level |
| Program Memory | Address 0000h must be NOP |
| | Last 100h words |
| Data Memory | 0x070 (0x0F0, 0x170, 0x1F0) 0x1EB - 0x1EF |

To use the In-Circuit Debugger function of the microcontroller, the design must implement In-Circuit Serial Programming connections to MCLR/VPP, VDD, GND, RB7 and RB6. This will interface to the In-Circuit Debugger module available from Microchip, or one of the third party development tool companies.

11.15 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.

11.16 ID Locations

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

PIC16F870/871

11.17 In-Circuit Serial Programming

PIC16F870/871 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.

When using ICSP, the part must be supplied at 4.5V to 5.5V, if a bulk erase will be executed. This includes reprogramming of the code protect, both from an on-state to off-state. For all other cases of ICSP, the part may be programmed at the normal operating voltages. This means calibration values, unique user IDs, or user code can be reprogrammed or added.

For complete details of serial programming, please refer to the EEPROM Memory Programming Specification for the PIC16F87X (DS39025).

11.18 Low Voltage ICSP Programming

The LVP bit of the configuration word enables low voltage ICSP programming. This mode allows the microcontroller to be programmed via ICSP, using a V_{DD} source in the operating voltage range. This only means that V_{PP} does not have to be brought to V_{IH} , but can instead be left at the normal operating voltage. In this mode, the RB3/PGM pin is dedicated to the programming function and ceases to be a general purpose I/O pin. During programming, V_{DD} is applied to the \overline{MCLR} pin. To enter Programming mode, V_{DD} must be applied to the RB3/PGM pin, provided the LVP bit is set. The LVP bit defaults to on ('1') from the factory.

Note 1: The High Voltage Programming mode is always available, regardless of the state of the LVP bit, by applying V_{IH} to the \overline{MCLR} pin.

- 2:** While in Low Voltage ICSP mode, the RB3 pin can no longer be used as a general purpose I/O pin.
- 3:** When using low voltage ICSP programming (LVP) and the pull-ups on PORTB are enabled, bit 3 in the TRISB register must be cleared to disable the pull-up on RB3 and ensure the proper operation of the device.
- 4:** RB3 should not be allowed to float if LVP is enabled. An external pull-down device should be used to default the device to normal Operating mode. If RB3 floats high, the PIC16F870/871 devices will enter Programming mode.
- 5:** LVP mode is enabled by default on all devices shipped from Microchip. It can be disabled by clearing the LVP bit in the CONFIG register.
- 6:** Disabling LVP will provide maximum compatibility to other PIC16CXXX devices.

If Low Voltage Programming mode is not used, the LVP bit can be programmed to a '0' and RB3/PGM becomes a digital I/O pin. However, the LVP bit may only be programmed when programming is entered with V_{IH} on \overline{MCLR} . The LVP bit can only be changed when using high voltage on \overline{MCLR} .

It should be noted, that once the LVP bit is programmed to 0, only the High Voltage Programming mode is available and only High Voltage Programming mode can be used to program the device.

When using low voltage ICSP, the part must be supplied at 4.5V to 5.5V, if a bulk erase will be executed. This includes reprogramming of the code protect bits from an on-state to off-state. For all other cases of low voltage ICSP, the part may be programmed at the normal operating voltage. This means calibration values, unique user IDs, or user code can be reprogrammed or added.

12.0 INSTRUCTION SET SUMMARY

Each PIC16F870/871 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 PIC16F870/871 instruction set summary in Table 12-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 12-1 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 12-1: 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). 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, d = 1: store result in file register f. Default is d = 1. |
| PC | Program Counter |
| TO | Time-out bit |
| PD | Power-down bit |

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 μ s. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μ s.

Table 12-2 lists the instructions recognized by the MPASM™ assembler.

Figure 12-1 shows the general formats that the instructions can have.

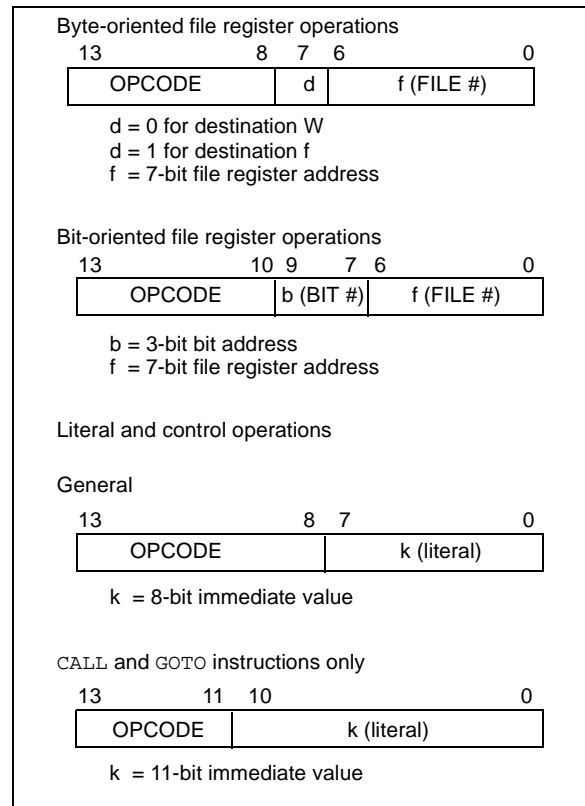
Note: To maintain upward compatibility with future PIC16F870/871 products, do not use the OPTION and TRIS instructions.

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

0xhh

where h signifies a hexadecimal digit.

FIGURE 12-1: GENERAL FORMAT FOR INSTRUCTIONS



A description of each instruction is available in the PIC® Mid-Range MCU Family Reference Manual (DS33023).

PIC16F870/871

TABLE 12-2: PIC16F870/871 INSTRUCTION SET

| Mnemonic, Operands | Description | Cycles | 14-Bit Opcode | | | | Status Affected | Notes | |
|---|-------------|------------------------------|---------------|-----|------|------|--------------------|--------------------------------|-------|
| | | | MSb | LSb | | | | | |
| 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 | 1fff | ffff | Z | 2 |
| CLRWF | - | Clear W | 1 | 00 | 0001 | 0xxx | xxxx | Z | |
| COMF | f, d | Complement f | 1 | 00 | 1001 | dfff | ffff | Z | 1,2 |
| DECf | f, d | Decrement f | 1 | 00 | 0011 | dfff | ffff | Z | 1,2 |
| DECFSZ | f, d | Decrement f, Skip if 0 | 1(2) | 00 | 1011 | dfff | ffff | | 1,2,3 |
| INCF | f, d | Increment f | 1 | 00 | 1010 | dfff | ffff | Z | 1,2 |
| INCFSZ | f, d | Increment f, Skip if 0 | 1(2) | 00 | 1111 | dfff | ffff | | 1,2,3 |
| IORWF | f, d | Inclusive OR W with f | 1 | 00 | 0100 | dfff | ffff | Z | 1,2 |
| MOVF | f, d | Move f | 1 | 00 | 1000 | dfff | ffff | Z | 1,2 |
| MOVWF | f | Move W to f | 1 | 00 | 0000 | 1fff | 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 | | |
| CLRWD _T | - | 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 PIC[®] Mid-Range MCU Family Reference Manual (DS33023).

12.1 Instruction Descriptions

ADDLW **Add Literal and W**

Syntax: [*label*] ADDLW *k*

Operands: $0 \leq k \leq 255$

Operation: $(W) + k \rightarrow (W)$

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

Operation: $(W) + (f) \rightarrow (\text{destination})$

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: $0 \leq k \leq 255$

Operation: $(W) .\text{AND.} (k) \rightarrow (W)$

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

Operation: $(W) .\text{AND.} (f) \rightarrow (\text{destination})$

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: $0 \leq f \leq 127$
 $0 \leq b \leq 7$

Operation: $0 \rightarrow (f)$

Status Affected: None

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

BSF **Bit Set f**

Syntax: [*label*] BSF *f,b*

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

Operation: $1 \rightarrow (f)$

Status Affected: None

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

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

Syntax: [*label*] BTFSS *f,b*

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

Operation: skip if $(f) = 1$

Status Affected: None

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

BTFSC **Bit Test, Skip if Clear**

Syntax: [*label*] BTFSC *f,b*

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

Operation: skip if $(f) = 0$

Status Affected: None

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

PIC16F870/871

CALL **Call Subroutine**

Syntax: [*label*] CALL k
Operands: $0 \leq k \leq 2047$
Operation: (PC)+ 1 → TOS,
 k → PC<10:0>,
 (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.

CLRWDT **Clear Watchdog Timer**

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

CLRF **Clear f**

Syntax: [*label*] CLRF f
Operands: $0 \leq f \leq 127$
Operation: 00h → (f)
 1 → Z
Status Affected: Z
Description: The contents of register 'f' are cleared and the Z bit is set.

COMF **Complement f**

Syntax: [*label*] COMF f,d
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: (\bar{f}) → (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'.

CLRW **Clear W**

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

DECF **Decrement f**

Syntax: [*label*] DECF f,d
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
Operation: (f) - 1 → (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'.

DECFSZ Decrement f, Skip if 0

Syntax: [*label*] DECFSZ f,d

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

Operation: (f) - 1 → (destination);
 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'.
 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: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: (f) + 1 → (destination),
 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'.
 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.

GOTO Unconditional Branch

Syntax: [*label*] GOTO k

Operands: $0 \leq k \leq 2047$

Operation: $k \rightarrow PC<10:0>$
 $PCLATH<4:3> \rightarrow PC<12:11>$

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.

IORLW Inclusive OR Literal with W

Syntax: [*label*] IORLW k

Operands: $0 \leq k \leq 255$

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.

INCF Increment f

Syntax: [*label*] INCF f,d

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

Operation: (f) + 1 → (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'.

IORWF Inclusive OR W with f

Syntax: [*label*] IORWF f,d

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

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'.

PIC16F870/871

MOVF **Move f**

Syntax: [*label*] MOVF f,d
Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
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.

NOP **No Operation**

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

MOVLW **Move Literal to W**

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

RETFIE **Return from Interrupt**

Syntax: [*label*] RETFIE
Operands: None
Operation: TOS → PC,
 1 → GIE
Status Affected: None

MOVWF **Move W to f**

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

RETLW **Return with Literal in W**

Syntax: [*label*] RETLW k
Operands: $0 \leq k \leq 255$
Operation: $k \rightarrow (W)$;
 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.

RLF Rotate Left f through Carry

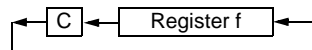
Syntax: [*label*] RLF f,d

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

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'.



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.

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.

SUBLW Subtract W from Literal

Syntax: [*label*] SUBLW k

Operands: $0 \leq k \leq 255$

Operation: $k - (W) \rightarrow (W)$

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.

RRF Rotate Right f through Carry

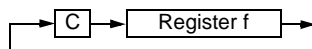
Syntax: [*label*] RRF f,d

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

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'.



SUBWF Subtract W from f

Syntax: [*label*] SUBWF f,d

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

Operation: $(f) - (W) \rightarrow (\text{destination})$

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'.

PIC16F870/871

SWAPF **Swap Nibbles in f**

Syntax: [*label*] SWAPF f,d

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

Operation: (f<3:0>) → (destination<7:4>),
 (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 the W register. If 'd' is 1, the result is placed in register 'f'.

XORWF **Exclusive OR W with f**

Syntax: [*label*] XORWF f,d

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

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'.

XORLW **Exclusive OR Literal with W**

Syntax: [*label*] XORLW k

Operands: $0 \leq k \leq 255$

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.

13.0 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
 - MPLAB C30 C Compiler
 - MPLAB ASM30 Assembler/Linker/Library
- Simulators
 - MPLAB SIM Software Simulator
 - MPLAB dsPIC30 Software Simulator
- Emulators
 - MPLAB ICE 2000 In-Circuit Emulator
 - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
 - MPLAB ICD 2
- Device Programmers
 - PRO MATE[®] II Universal Device Programmer
 - PICSTART[®] Plus Development Programmer
- Low Cost Demonstration Boards
 - PICDEM[™] 1 Demonstration Board
 - PICDEM.net[™] Demonstration Board
 - PICDEM 2 Plus Demonstration Board
 - PICDEM 3 Demonstration Board
 - PICDEM 4 Demonstration Board
 - PICDEM 17 Demonstration Board
 - PICDEM 18R Demonstration Board
 - PICDEM LIN Demonstration Board
 - PICDEM USB Demonstration Board
- Evaluation Kits
 - KEELOQ[®]
 - PICDEM MSC
 - microID[®]
 - CAN
 - PowerSmart[®]
 - Analog

13.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-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 with color coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High level source code debugging
- Mouse over variable inspection
- Extensive 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 MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
 - source files (assembly or C)
 - absolute listing file (mixed assembly and C)
 - machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost effective simulators, through low cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increasing flexibility and power.

13.2 MPASM Assembler

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

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, absolute LST files that contain source lines and generated machine code and COFF files 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

PIC16F870/871

13.3 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. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

13.4 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 link relocatable objects from pre-compiled libraries, using directives from a linker script.

The MPLIB object librarian manages the creation and modification of library files of pre-compiled code. When a routine from a library is called from a 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 object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

13.5 MPLAB C30 C Compiler

The MPLAB C30 C compiler is a full-featured, ANSI compliant, optimizing compiler that translates standard ANSI C programs into dsPIC30F assembly language source. The compiler also supports many command-line options and language extensions to take full advantage of the dsPIC30F device hardware capabilities, and afford fine control of the compiler code generator.

MPLAB C30 is distributed with a complete ANSI C standard library. All library functions have been validated and conform to the ANSI C library standard. The library includes functions for string manipulation, dynamic memory allocation, data conversion, time-keeping, and math functions (trigonometric, exponential and hyperbolic). The compiler provides symbolic information for high level source debugging with the MPLAB IDE.

13.6 MPLAB ASM30 Assembler, Linker, and Librarian

MPLAB ASM30 assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

13.7 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 pin. 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 MPLAB C18 C Compilers, as well as the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

13.8 MPLAB SIM30 Software Simulator

The MPLAB SIM30 software simulator allows code development in a PC hosted environment by simulating the dsPIC30F 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 MPLAB SIM30 simulator fully supports symbolic debugging using the MPLAB C30 C Compiler and MPLAB ASM30 assembler. The simulator runs in either a Command Line mode for automated tasks, or from MPLAB IDE. This high speed simulator is designed to debug, analyze and optimize time intensive DSP routines.

13.9 MPLAB ICE 2000 High Performance Universal In-Circuit Emulator

The MPLAB ICE 2000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers. Software control of the MPLAB ICE 2000 in-circuit emulator is advanced by the MPLAB Integrated Development Environment, 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 2000 in-circuit emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft® Windows 32-bit operating system were chosen to best make these features available in a simple, unified application.

13.10 MPLAB ICE 4000 High Performance Universal In-Circuit Emulator

The MPLAB ICE 4000 universal in-circuit emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PIC microcontrollers. Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 4000 is a premium emulator system, providing the features of MPLAB ICE 2000, but with increased emulation memory and high speed performance for dsPIC30F and PIC18XXX devices. Its advanced emulator features include complex triggering and timing, up to 2 Mb of emulation memory, and the ability to view variables in real-time.

The MPLAB ICE 4000 in-circuit emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft Windows 32-bit operating system were chosen to best make these features available in a simple, unified application.

13.11 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low cost, run-time development tool, connecting to the host PC via an RS-232 or high speed USB interface. This tool is based on the FLASH PIC MCUs and can be used to develop for these and other PIC microcontrollers. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the FLASH devices. This feature, along with Microchip's In-Circuit Serial Programming™ (ICSP™) 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 setting breakpoints, single-stepping and watching variables, CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real-time. MPLAB ICD 2 also serves as a development programmer for selected PIC devices.

13.12 PRO MATE II Universal Device Programmer

The PRO MATE II is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features an LCD display for instructions and error messages and a modular detachable socket assembly to support various package types. In Stand-Alone mode, the PRO MATE II device programmer can read, verify, and program PIC devices without a PC connection. It can also set code protection in this mode.

13.13 PICSTART Plus 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 most PIC devices 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.

PIC16F870/871

13.14 PICDEM 1 PIC MCU Demonstration Board

The PICDEM 1 demonstration board demonstrates the capabilities of the 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 sample microcontrollers provided with the PICDEM 1 demonstration board can be programmed with a PRO MATE II device programmer, or a PICSTART Plus development programmer. The PICDEM 1 demonstration board can be connected to the MPLAB ICE in-circuit emulator for testing. A prototype area extends the circuitry for additional application components. Features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs.

13.15 PICDEM.net Internet/Ethernet Demonstration Board

The PICDEM.net demonstration board is an Internet/Ethernet demonstration board using the PIC18F452 microcontroller and TCP/IP firmware. The board supports any 40-pin DIP device that conforms to the standard pinout used by the PIC16F877 or PIC18C452. This kit features a user friendly TCP/IP stack, web server with HTML, a 24L256 Serial EEPROM for Xmodem download to web pages into Serial EEPROM, ICSP/MPLAB ICD 2 interface connector, an Ethernet interface, RS-232 interface, and a 16 x 2 LCD display. Also included is the book and CD-ROM *"TCP/IP Lean, Web Servers for Embedded Systems,"* by Jeremy Bentham

13.16 PICDEM 2 Plus Demonstration Board

The PICDEM 2 Plus demonstration board supports many 18-, 28-, and 40-pin microcontrollers, including PIC16F87X and PIC18FXX2 devices. All the necessary hardware and software is included to run the demonstration programs. The sample microcontrollers provided with the PICDEM 2 demonstration board can be programmed with a PRO MATE II device programmer, PICSTART Plus development programmer, or MPLAB ICD 2 with a Universal Programmer Adapter. The MPLAB ICD 2 and MPLAB ICE in-circuit emulators may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area extends the circuitry for additional application components. Some of the features include an RS-232 interface, a 2 x 16 LCD display, a piezo speaker, an on-board temperature sensor, four LEDs, and sample PIC18F452 and PIC16F877 FLASH microcontrollers.

13.17 PICDEM 3 PIC16C92X Demonstration Board

The PICDEM 3 demonstration board supports the PIC16C923 and PIC16C924 in the PLCC package. All the necessary hardware and software is included to run the demonstration programs.

13.18 PICDEM 4 8/14/18-Pin Demonstration Board

The PICDEM 4 can be used to demonstrate the capabilities of the 8-, 14-, and 18-pin PIC16XXXX and PIC18XXXX MCUs, including the PIC16F818/819, PIC16F87/88, PIC16F62XA and the PIC18F1320 family of microcontrollers. PICDEM 4 is intended to showcase the many features of these low pin count parts, including LIN and Motor Control using ECCP. Special provisions are made for low power operation with the supercapacitor circuit, and jumpers allow on-board hardware to be disabled to eliminate current draw in this mode. Included on the demo board are provisions for Crystal, RC or Canned Oscillator modes, a five volt regulator for use with a nine volt wall adapter or battery, DB-9 RS-232 interface, ICD connector for programming via ICSP and development with MPLAB ICD 2, 2x16 liquid crystal display, PCB footprints for H-Bridge motor driver, LIN transceiver and EEPROM. Also included are: header for expansion, eight LEDs, four potentiometers, three push buttons and a prototyping area. Included with the kit is a PIC16F627A and a PIC18F1320. Tutorial firmware is included along with the User's Guide.

13.19 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. A programmed sample is included. The PRO MATE II device programmer, or the PICSTART Plus development programmer, can be used to reprogram the device for user tailored application development. The PICDEM 17 demonstration board supports program download and execution from external on-board FLASH memory. A generous prototype area is available for user hardware expansion.

13.20 PICDEM 18R PIC18C601/801 Demonstration Board

The PICDEM 18R demonstration board serves to assist development of the PIC18C601/801 family of Microchip microcontrollers. It provides hardware implementation of both 8-bit Multiplexed/De-multiplexed and 16-bit Memory modes. The board includes 2 Mb external FLASH memory and 128 Kb SRAM memory, as well as serial EEPROM, allowing access to the wide range of memory types supported by the PIC18C601/801.

13.21 PICDEM LIN PIC16C43X Demonstration Board

The powerful LIN hardware and software kit includes a series of boards and three PIC microcontrollers. The small footprint PIC16C432 and PIC16C433 are used as slaves in the LIN communication and feature on-board LIN transceivers. A PIC16F874 FLASH microcontroller serves as the master. All three microcontrollers are programmed with firmware to provide LIN bus communication.

13.22 PICKit™ 1 FLASH Starter Kit

A complete "development system in a box", the PICKit FLASH Starter Kit includes a convenient multi-section board for programming, evaluation, and development of 8/14-pin FLASH PIC® microcontrollers. Powered via USB, the board operates under a simple Windows GUI. The PICKit 1 Starter Kit includes the user's guide (on CD ROM), PICKit 1 tutorial software and code for various applications. Also included are MPLAB® IDE (Integrated Development Environment) software, software and hardware "Tips 'n Tricks for 8-pin FLASH PIC® Microcontrollers" Handbook and a USB Interface Cable. Supports all current 8/14-pin FLASH PIC microcontrollers, as well as many future planned devices.

13.23 PICDEM USB PIC16C7X5 Demonstration Board

The PICDEM USB Demonstration Board shows off the capabilities of the PIC16C745 and PIC16C765 USB microcontrollers. This board provides the basis for future USB products.

13.24 Evaluation and Programming Tools

In addition to the PICDEM series of circuits, Microchip has a line of evaluation kits and demonstration software for these products.

- KEELOQ evaluation and programming tools for Microchip's HCS Secure Data Products
- CAN developers kit for automotive network applications
- Analog design boards and filter design software
- PowerSmart battery charging evaluation/calibration kits
- IrDA® development kit
- microID development and rfLab™ development software
- SEEVAL® designer kit for memory evaluation and endurance calculations
- PICDEM MSC demo boards for Switching mode power supply, high power IR driver, delta sigma ADC, and flow rate sensor

Check the Microchip web page and the latest Product Line Card for the complete list of demonstration and evaluation kits.

PIC16F870/871

NOTES:

14.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings †

| | |
|--|-----------------------------------|
| Ambient temperature under bias | -55 to +125°C |
| Storage temperature | -65°C to +150°C |
| Voltage on any pin with respect to V _{SS} (except V _{DD} , $\overline{\text{MCLR}}$, and RA4) | -0.3V to (V _{DD} + 0.3V) |
| Voltage on V _{DD} with respect to V _{SS} | -0.3 to +7.5V |
| Voltage on $\overline{\text{MCLR}}$ with respect to V _{SS} (Note 2) | 0 to +13.25V |
| Voltage on RA4 with respect to V _{SS} | 0 to +8.5V |
| Total power dissipation (Note 1) | 1.0W |
| Maximum current out of V _{SS} pin | 300 mA |
| Maximum current into V _{DD} pin | 250 mA |
| Input clamp current, I _{IK} (V _I < 0 or V _I > V _{DD}) | ± 20 mA |
| Output clamp current, I _{OK} (V _O < 0 or V _O > V _{DD}) | ± 20 mA |
| Maximum output current sunk by any I/O pin | 25 mA |
| Maximum output current sourced by any I/O pin | 25 mA |
| Maximum current sunk by PORTA, PORTB, and PORTE (combined) (Note 3) | 200 mA |
| Maximum current sourced by PORTA, PORTB, and PORTE (combined) (Note 3) | 200 mA |
| Maximum current sunk by PORTC and PORTD (combined) (Note 3) | 200 mA |
| Maximum current sourced by PORTC and PORTD (combined) (Note 3) | 200 mA |

Note 1: Power dissipation is calculated as follows: $P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

2: Voltage spikes below V_{SS} at the $\overline{\text{MCLR}}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a “low” level to the $\overline{\text{MCLR}}$ pin, rather than pulling this pin directly to V_{SS}.

3: PORTD and PORTE are not implemented on the 28-pin devices.

† 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.

PIC16F870/871

FIGURE 14-1: PIC16FXXX VOLTAGE-FREQUENCY GRAPH

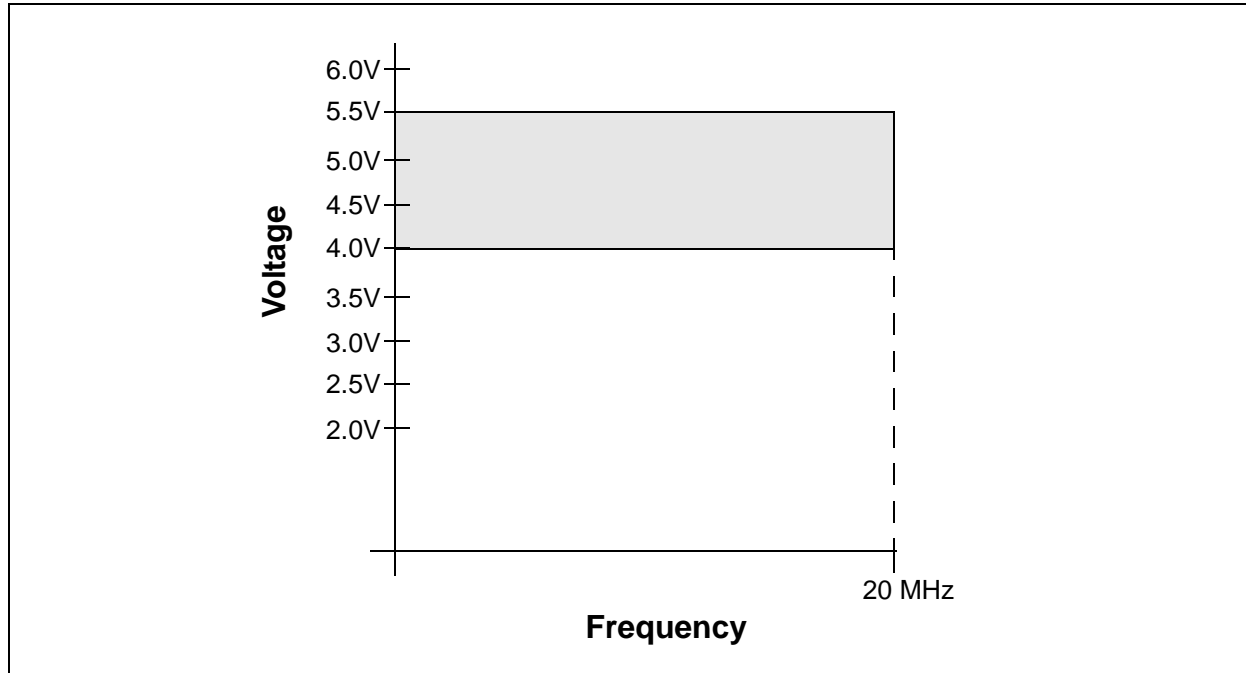
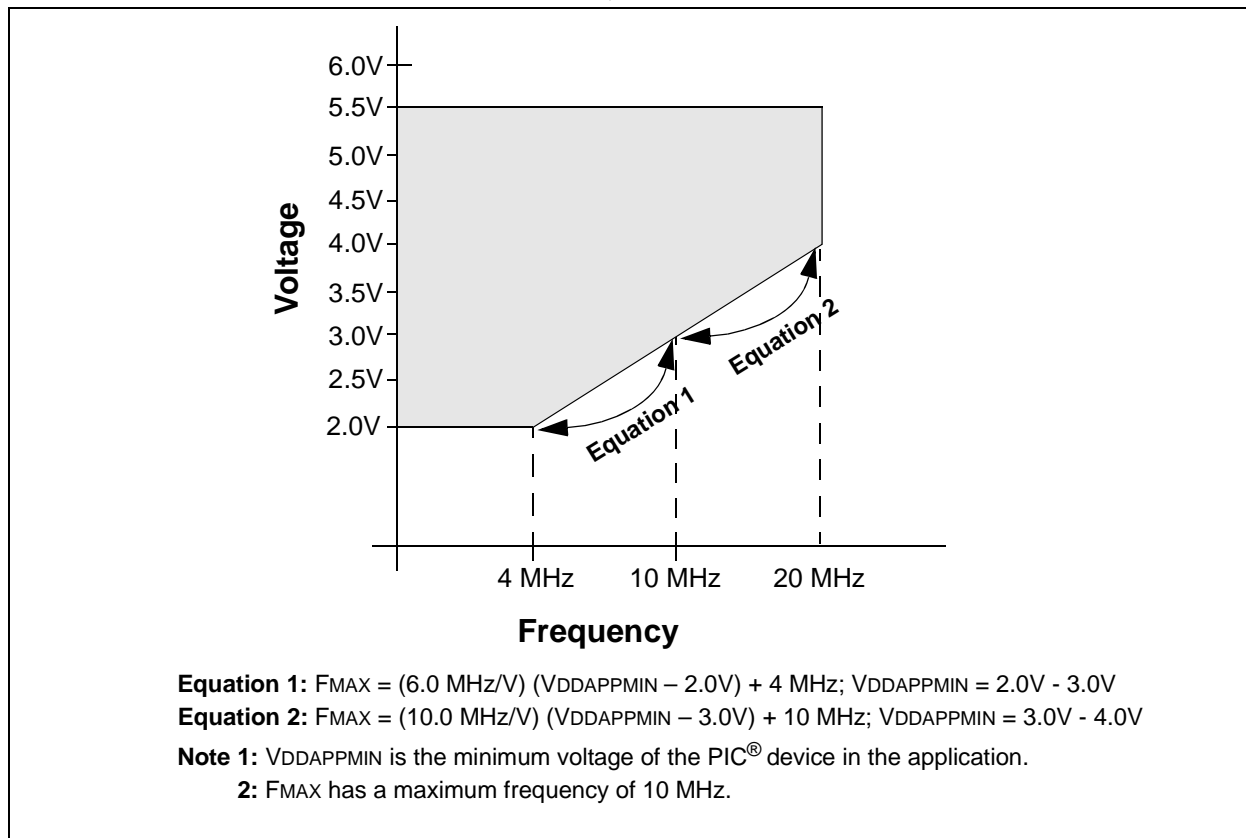


FIGURE 14-2: PIC16LFXXX VOLTAGE-FREQUENCY GRAPH



14.1 DC Characteristics: PIC16F870/871 (Industrial, Extended) PIC16LF870/871 (Commercial, Industrial)

| PIC16LF870/871 (Commercial, Industrial) | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for Commercial | | | | | |
|--|------|--|-------|------|------|-------|--|
| PIC16F870/871 (Industrial, Extended) | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | | |
| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
| D001 | VDD | Supply Voltage | | | | | |
| | | PIC16LF870/871 | 2.0 | — | 5.5 | V | All configurations. See Figure 14-2 for details. |
| D001 D001A | | PIC16F870/871 | 4.0 | — | 5.5 | V | All configurations. BOR enabled, $F_{\text{MAX}} = 14 \text{ MHz}$ (Note 7), -40°C to $+85^{\circ}\text{C}$ |
| | | | VBOR* | — | 5.5 | V | |
| D002* | VDR | RAM Data Retention Voltage⁽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* | SVDD | VDD Rise Rate to ensure internal Power-on Reset signal | 0.05 | — | — | V/ms | See section on Power-on Reset for details |
| D005 | VBOR | Brown-out Reset Voltage | 3.7 | 4.0 | 4.35 | V | BOREN bit in configuration word enabled |

* These parameters are characterized but not tested.

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

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

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature also have an impact on the current consumption.

The test conditions for all IDD measurements in active Operation mode are:

OSC1 = external square wave, from rail to rail; all I/O pins 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 configuration, current through REXT is not included. The current through the resistor can be estimated by the formula $I_r = V_{\text{DD}}/2R_{\text{EXT}}$ (mA) with REXT in kΩ.

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.

PIC16F870/871

14.1 DC Characteristics: PIC16F870/871 (Industrial, Extended) PIC16LF870/871 (Commercial, Industrial) (Continued)

| PIC16LF870/871 (Commercial, Industrial) | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for Commercial | | | | | |
|--|-------|--|-----|------|-----|--|---|
| PIC16F870/871 (Industrial, Extended) | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for Extended | | | | | |
| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
| D010 | IDD | Supply Current^(2,5) | | | | | |
| D010A | | PIC16LF870/871 | — | 0.6 | 2.0 | mA | XT, RC osc configuration FOSC = 4 MHz, VDD = 3.0V (Note 4) |
| D010 | | PIC16F870/871 | — | 20 | 35 | μA | LP osc configuration FOSC = 32 kHz, VDD = 3.0V, WDT disabled |
| D010 | | | | 1.6 | 4 | mA | XT, RC osc configuration FOSC = 4 MHz, VDD = 5.5V (Note 4) |
| D013 | | | | 7 | 15 | mA | HS osc configuration FOSC = 20 MHz, VDD = 5.5V, -40°C to $+85^{\circ}\text{C}$ |
| | | | 7 | 15 | mA | HS osc configuration FOSC = 10 MHz, VDD = 5.5V, -40°C to $+125^{\circ}\text{C}$ | |
| D015* | ΔIBOR | Brown-out Reset Current⁽⁶⁾ | — | 85 | 200 | μA | BOR enabled, VDD = 5.0V |
| D020 | IPD | Power-down Current^(3,5) | | | | | |
| D021 | | PIC16LF870/871 | — | 7.5 | 30 | μA | VDD = 3.0V, WDT enabled, -40°C to $+85^{\circ}\text{C}$ |
| D021A | | | | 0.8 | 4.5 | μA | VDD = 3.0V, WDT disabled, 0°C to $+70^{\circ}\text{C}$ |
| D021A | 0.9 | | | 5 | μA | VDD = 3.0V, WDT disabled, -40°C to $+85^{\circ}\text{C}$ | |
| D020 | | PIC16F870/871 | — | 10.5 | 42 | μA | VDD = 4.0V, WDT enabled, -40°C to $+85^{\circ}\text{C}$ |
| D20A | | | | 10.5 | 60 | μA | VDD = 4.0V, WDT enabled, -40°C to $+125^{\circ}\text{C}$ |
| D021 | | | | 1.5 | 16 | μA | VDD = 4.0V, WDT disabled, 0°C to $+70^{\circ}\text{C}$ |
| D021A | | | | 1.5 | 19 | μA | VDD = 4.0V, WDT disabled, -40°C to $+85^{\circ}\text{C}$ |
| D21B | | | | 1.5 | 30 | μA | VDD = 4.0V, WDT disabled, -40°C to $+125^{\circ}\text{C}$ |
| D023* | ΔIBOR | Brown-out Reset Current⁽⁶⁾ | — | 85 | 200 | μA | BOR enabled, VDD = 5.0V |

* These parameters are characterized but not tested.

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

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

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature also have an impact on the current consumption.

The test conditions for all IDD measurements in active Operation mode are:

OSC1 = external square wave, from rail to rail; all I/O pins 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 configuration, current through REXT is not included. The current through the resistor can be estimated by the formula $I_r = V_{DD}/2R_{EXT}$ (mA) with REXT in kΩ.

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.

14.2 DC Characteristics: PIC16F870/871 (Industrial)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial Operating voltage V_{DD} range as described in DC spec Section 14.1 and Section 14.2. | | | | |
|---|----------|--|--|--------------------------------------|--|---|--|
| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
| D030 D030A D031 D032 D033 D034 D034A | V_{IL} | Input Low Voltage I/O ports: with TTL buffer with Schmitt Trigger buffer $\overline{\text{MCLR}}$, OSC1 (in RC mode) OSC1 (in XT, HS and LP) Ports RC3 and RC4: with Schmitt Trigger buffer with SMBus | V_{SS} V_{SS} V_{SS} V_{SS} V_{SS} V_{SS} -0.5 | — — — — — — — | $0.15 V_{DD}$ 0.8V $0.2 V_{DD}$ $0.2 V_{DD}$ $0.3 V_{DD}$ $0.3 V_{DD}$ 0.6 | V V V V V V V | For entire V_{DD} range $4.5V \leq V_{DD} \leq 5.5V$ (Note 1) For entire V_{DD} range For $V_{DD} = 4.5$ to $5.5V$ |
| D040 D040A D041 D042 D042A D043 D044 D044A | V_{IH} | Input High Voltage I/O ports: with TTL buffer with Schmitt Trigger buffer $\overline{\text{MCLR}}$ OSC1 (XT, HS and LP) OSC1 (in RC mode) Ports RC3 and RC4: with Schmitt Trigger buffer with SMBus | 2.0 $0.25 V_{DD} + 0.8V$ $0.8 V_{DD}$ $0.8 V_{DD}$ $0.7 V_{DD}$ $0.9 V_{DD}$ $0.7 V_{DD}$ 1.4 | — — — — — — — — | V_{DD} V_{DD} V_{DD} V_{DD} V_{DD} V_{DD} V_{DD} 5.5 | V V V V V V V V | $4.5V \leq V_{DD} \leq 5.5V$ For entire V_{DD} range For entire V_{DD} range (Note 1) For entire V_{DD} range for $V_{DD} = 4.5$ to $5.5V$ |
| D070 | IPURB | PORTB Weak Pull-up Current | 50 | 250 | 400 | μA | $V_{DD} = 5V$, $V_{PIN} = V_{SS}$ |
| D060 D061 D063 | I_{IL} | Input Leakage Current (Notes 2, 3) I/O ports $\overline{\text{MCLR}}$, RA4/T0CKI OSC1 | — — — | — — — | ± 1 ± 5 ± 5 | μA μA μA | $V_{SS} \leq V_{PIN} \leq V_{DD}$, Pin at hi-impedance $V_{SS} \leq V_{PIN} \leq V_{DD}$ $V_{SS} \leq V_{PIN} \leq V_{DD}$, XT, HS and LP osc configuration |

* These parameters are characterized but not tested.

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

Note 1: In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC16F870/871 be driven with external clock in RC mode.

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

3: Negative current is defined as current sourced by the pin.

PIC16F870/871

14.2 DC Characteristics: PIC16F870/871 (Industrial) (Continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for Industrial Operating voltage V_{DD} range as described in DC spec Section 14.1 and Section 14.2. | | | | |
|--|-------|---|--|------|-----|-------|--|
| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
| D080 | VOL | Output Low Voltage I/O ports | — | — | 0.6 | V | $I_{OL} = 8.5\text{ mA}$, $V_{DD} = 4.5\text{V}$, -40°C to $+85^{\circ}\text{C}$ |
| D083 | | OSC2/CLKO (RC osc config) | — | — | 0.6 | V | $I_{OL} = 1.6\text{ mA}$, $V_{DD} = 4.5\text{V}$, -40°C to $+85^{\circ}\text{C}$ |
| D090 | VOH | Output High Voltage I/O ports (Note 3) | $V_{DD} - 0.7$ | — | — | V | $I_{OH} = -3.0\text{ mA}$, $V_{DD} = 4.5\text{V}$, -40°C to $+85^{\circ}\text{C}$ |
| D092 | | OSC2/CLKO (RC osc config) | $V_{DD} - 0.7$ | — | — | V | $I_{OH} = -1.3\text{ mA}$, $V_{DD} = 4.5\text{V}$, -40°C to $+85^{\circ}\text{C}$ |
| D150* | VOD | Open Drain High Voltage | — | — | 8.5 | V | RA4 pin |
| 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 ² C mode | — | — | 400 | pF | |
| Data EEPROM Memory | | | | | | | |
| D120 | ED | Endurance | 100K | — | — | E/W | 25°C at 5V |
| D121 | VDRW | VDD for read/write | V _{MIN} | — | 5.5 | V | Using EECON to read/write V _{MIN} = min operating voltage |
| D122 | TDEW | Erase/write cycle time | — | 4 | 8 | ms | |
| Program FLASH Memory | | | | | | | |
| D130 | EP | Endurance | 1000 | — | — | E/W | 25°C at 5V |
| D131 | VPR | VDD for read | V _{MIN} | — | 5.5 | V | V _{MIN} = min operating voltage |
| D132a | | VDD for erase/write | V _{MIN} | — | 5.5 | V | Using EECON to read/write, V _{MIN} = min operating voltage |
| D133 | TPEW | Erase/Write cycle time | — | 4 | 8 | ms | |

* These parameters are characterized but not tested.

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

- Note 1:** In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC16F870/871 be driven with external clock in RC mode.
- 2:** The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as current sourced by the pin.

14.3 DC Characteristics: PIC16F870/871 (Extended)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ Operating voltage V_{DD} range as described in DC specification (Section) | | | | |
|---|----------|--|---|---|---|---|---|
| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
| D030 D030A D031 D032 D033 D034 D034A | V_{IL} | Input Low Voltage I/O ports: with TTL buffer with Schmitt Trigger buffer $\overline{\text{MCLR}}$, OSC1 (in RC mode) OSC1 (in XT, HS and LP) Ports RC3 and RC4: with Schmitt Trigger buffer with SMBus | V_{SS} V_{SS} V_{SS} V_{SS} V_{SS} V_{SS} -0.5 | — — — — — — — | 0.15 V_{DD} 0.8V 0.2 V_{DD} 0.2 V_{DD} 0.3 V_{DD} 0.3 V_{DD} 0.6 | V V V V V V V | For entire V_{DD} range $4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$ (Note 1) For entire V_{DD} range For $V_{DD} = 4.5$ to 5.5V |
| D040 D040A D041 D042 D042A D043 D044 D044A | V_{IH} | Input High Voltage I/O ports: with TTL buffer with Schmitt Trigger buffer $\overline{\text{MCLR}}$ OSC1 (XT, HS and LP) OSC1 (in RC mode) Ports RC3 and RC4: with Schmitt Trigger buffer with SMBus | 2.0 0.25 V_{DD} + 0.8V 0.8 V_{DD} 0.8 V_{DD} 0.7 V_{DD} 0.9 V_{DD} 0.7 V_{DD} 1.4 | — — — — — — — — — | V_{DD} V_{DD} V_{DD} V_{DD} V_{DD} V_{DD} V_{DD} V_{DD} 5.5 | V V V V V V V V V | $4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$ For entire V_{DD} range For entire V_{DD} range (Note 1) For entire V_{DD} range For $V_{DD} = 4.5$ to 5.5V |
| D070A | IPURB | PORTB Weak Pull-up Current | 50 | 250 | 400 | μA | $V_{DD} = 5\text{V}$, $V_{PIN} = V_{SS}$ |
| D060 D061 D063 | I_{IL} | Input Leakage Current^(2,3) I/O ports $\overline{\text{MCLR}}$, RA4/T0CKI OSC1 | — — — | — — — | ± 1 ± 5 ± 5 | μA μA μA | $V_{SS} \leq V_{PIN} \leq V_{DD}$, Pin at hi-impedance $V_{SS} \leq V_{PIN} \leq V_{DD}$ $V_{SS} \leq V_{PIN} \leq V_{DD}$, XT, HS and LP osc configuration |

* These parameters are characterized but not tested.

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

- Note 1:** In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC16F87X be driven with external clock in RC mode.
- 2:** The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as current sourced by the pin.

PIC16F870/871

14.3 DC Characteristics: PIC16F870/871 (Extended) (Continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ Operating voltage V_{DD} range as described in DC specification (Section) | | | | |
|-------------------------------|--------------------|---|---|------|-----------------|---------------|---|
| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
| D080A D083A | VOL | Output Low Voltage I/O ports OSC2/CLKO (RC osc config) | — | — | 0.6 | V | $I_{OL} = 7.0 \text{ mA}$, $V_{DD} = 4.5\text{V}$ $I_{OL} = 1.2 \text{ mA}$, $V_{DD} = 4.5\text{V}$ |
| D090A D092A | VOH | Output High Voltage I/O ports ⁽³⁾ OSC2/CLKO (RC osc config) | $V_{DD} - 0.7$ $V_{DD} - 0.7$ | — | — | V | $I_{OH} = -2.5 \text{ mA}$, $V_{DD} = 4.5\text{V}$ $I_{OH} = -1.0 \text{ mA}$, $V_{DD} = 4.5\text{V}$ |
| D150* | VOD | Open Drain High Voltage | — | — | 8.5 | V | RA4 pin |
| D100 D101 D102 | Cosc2 Cio CB | Capacitive Loading Specs on Output Pins OSC2 pin All I/O pins and OSC2 (RC mode) SCL, SDA (I ² C mode) | — | — | 15 50 400 | pF | In XT, HS and LP modes when external clock is used to drive OSC1 |
| D120 D121 D122 | ED VDRW TDEW | Data EEPROM Memory Endurance VDD for read/write Erase/write cycle time | 100K V_{MIN} | — | — 5.5 | E/W V | 25°C at 5V Using EECON to read/write, V_{MIN} = min. operating voltage |
| D130 D131 D132A D133 | EP VPR TPEW | Program FLASH Memory Endurance VDD for read VDD for erase/write Erase/Write cycle time | 1000 V_{MIN} V_{MIN} | — | — 5.5 5.5 | E/W V V | 25°C at 5V V_{MIN} = min operating voltage Using EECON to read/write, V_{MIN} = min. operating voltage |

* These parameters are characterized but not tested.

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

- Note 1:** In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC16F87X be driven with external clock in RC mode.
- 2:** The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.
- 3:** Negative current is defined as current sourced by the pin.

14.4 Timing Parameter Symbology

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

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

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

Lowercase letters (pp) and their meanings:

| | | | |
|-----------|-------------------|-----|------------------------------------|
| pp | | | |
| cc | CCP1 | osc | OSC1 |
| ck | CLKO | rd | \overline{RD} |
| cs | \overline{CS} | rw | \overline{RD} or \overline{WR} |
| di | SDI | sc | \overline{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 |
| I²C only | | | |
| AA | output access | High | High |
| BUF | Bus free | Low | Low |

TCC:ST (I²C specifications only)

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

PIC16F870/871

FIGURE 14-3: LOAD CONDITIONS

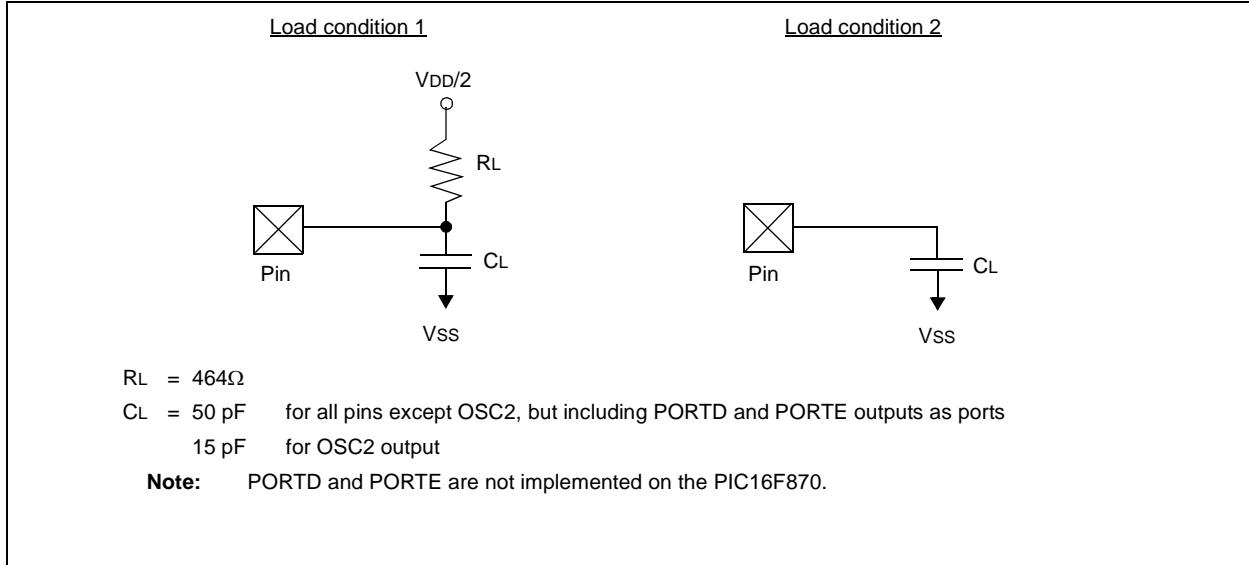


FIGURE 14-4: EXTERNAL CLOCK TIMING

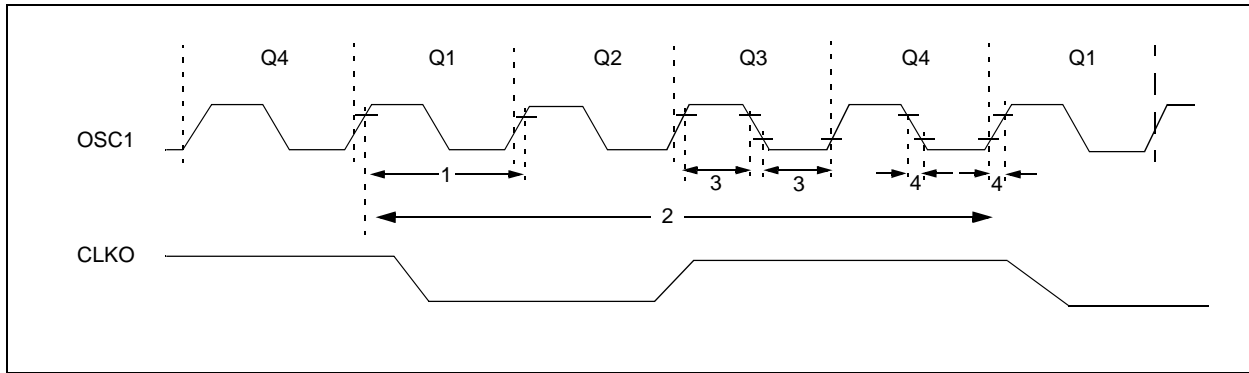


TABLE 14-1: EXTERNAL CLOCK TIMING REQUIREMENTS

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|-----------|-----------------|---|-----|-----------------|--------|-------|--------------------------|
| | FOSC | External CLKI Frequency (Note 1) | DC | — | 4 | MHz | XT and RC Osc mode |
| | | | 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 CLKI Period (Note 1) | 250 | — | — | ns | XT and RC Osc mode |
| | | | 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 |
| | | | 5 | — | — | μs | LP Osc mode |
| 2 | T _{CY} | Instruction Cycle Time (Note 1) | 200 | T _{CY} | DC | ns | T _{CY} = 4/FOSC |
| 3 | TosL, TosH | External Clock in (OSC1) High or Low Time | 100 | — | — | ns | XT oscillator |
| | | | 2.5 | — | — | μs | LP oscillator |
| | | | 15 | — | — | ns | HS oscillator |
| 4 | TosR, TosF | External Clock in (OSC1) Rise or Fall Time | — | — | 25 | ns | XT oscillator |
| | | | — | — | 50 | ns | LP oscillator |
| | | | — | — | 15 | ns | HS oscillator |

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

Note 1: Instruction cycle period (T_{CY}) 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/CLKI pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

PIC16F870/871

FIGURE 14-5: CLKO AND I/O TIMING

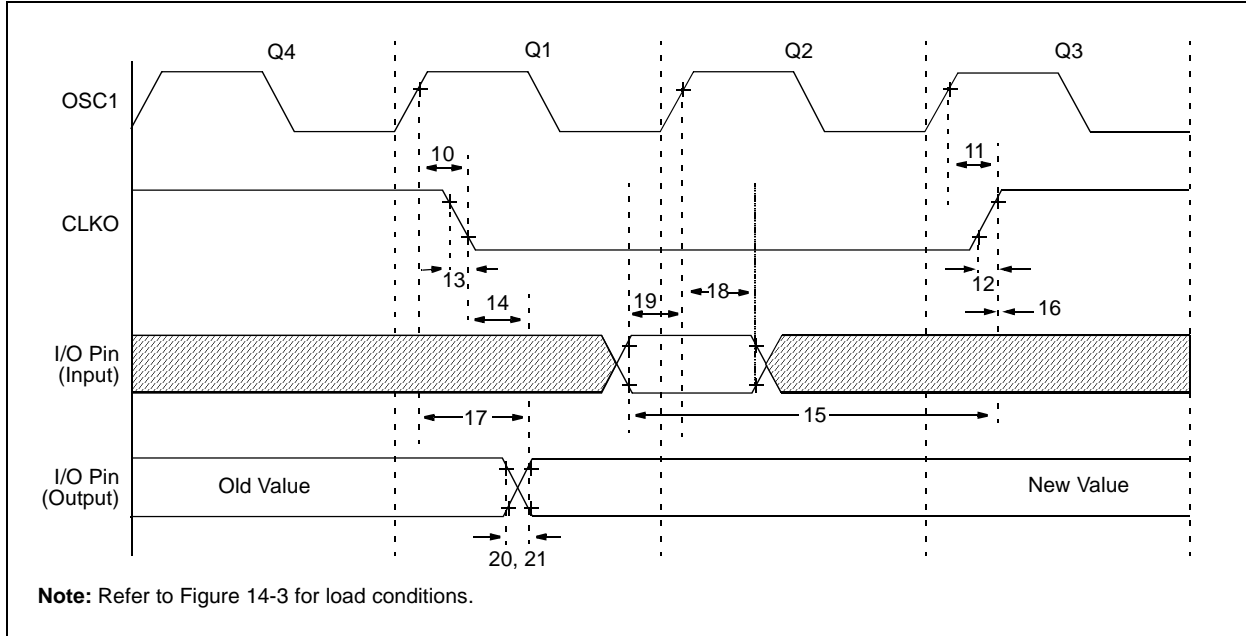


TABLE 14-2: CLKO AND I/O TIMING REQUIREMENTS

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions | |
|-----------|----------|---|---------------|------|--------------|-------|------------|--|
| 10* | TosH2ckL | OSC1↑ to CLKO↓ | — | 75 | 200 | ns | (Note 1) | |
| 11* | TosH2ckH | OSC1↑ to CLKO↑ | — | 75 | 200 | ns | (Note 1) | |
| 12* | TckR | CLKO rise time | — | 35 | 100 | ns | (Note 1) | |
| 13* | TckF | CLKO fall time | — | 35 | 100 | ns | (Note 1) | |
| 14* | TckL2ioV | CLKO↓ to Port out valid | — | — | 0.5 Tcy + 20 | ns | (Note 1) | |
| 15* | TioV2ckH | Port in valid before CLKO↑ | Tosc + 200 | — | — | ns | (Note 1) | |
| 16* | TckH2ioI | Port in hold after CLKO↑ | 0 | — | — | ns | (Note 1) | |
| 17* | TosH2ioV | OSC1↑ (Q1 cycle) to Port out valid | — | 100 | 255 | ns | | |
| 18* | TosH2ioI | OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time) | Standard (F) | 100 | — | — | ns | |
| | | | Extended (LF) | 200 | — | — | ns | |
| 19* | TioV2osH | Port input valid to OSC1↑ (I/O in setup time) | 0 | — | — | ns | | |
| 20* | TioR | Port output rise time | Standard (F) | — | 10 | 40 | ns | |
| | | | Extended (LF) | — | — | 145 | ns | |
| 21* | TioF | Port output fall time | Standard (F) | — | 10 | 40 | ns | |
| | | | Extended (LF) | — | — | 145 | ns | |
| 22††* | TINP | INT pin high or low time | Tcy | — | — | ns | | |
| 23††* | TRBP | RB7:RB4 change INT high or low time | Tcy | — | — | ns | | |

* These parameters are characterized but not tested.

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

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

Note 1: Measurements are taken in RC mode where CLKO output is 4 x Tosc.

FIGURE 14-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

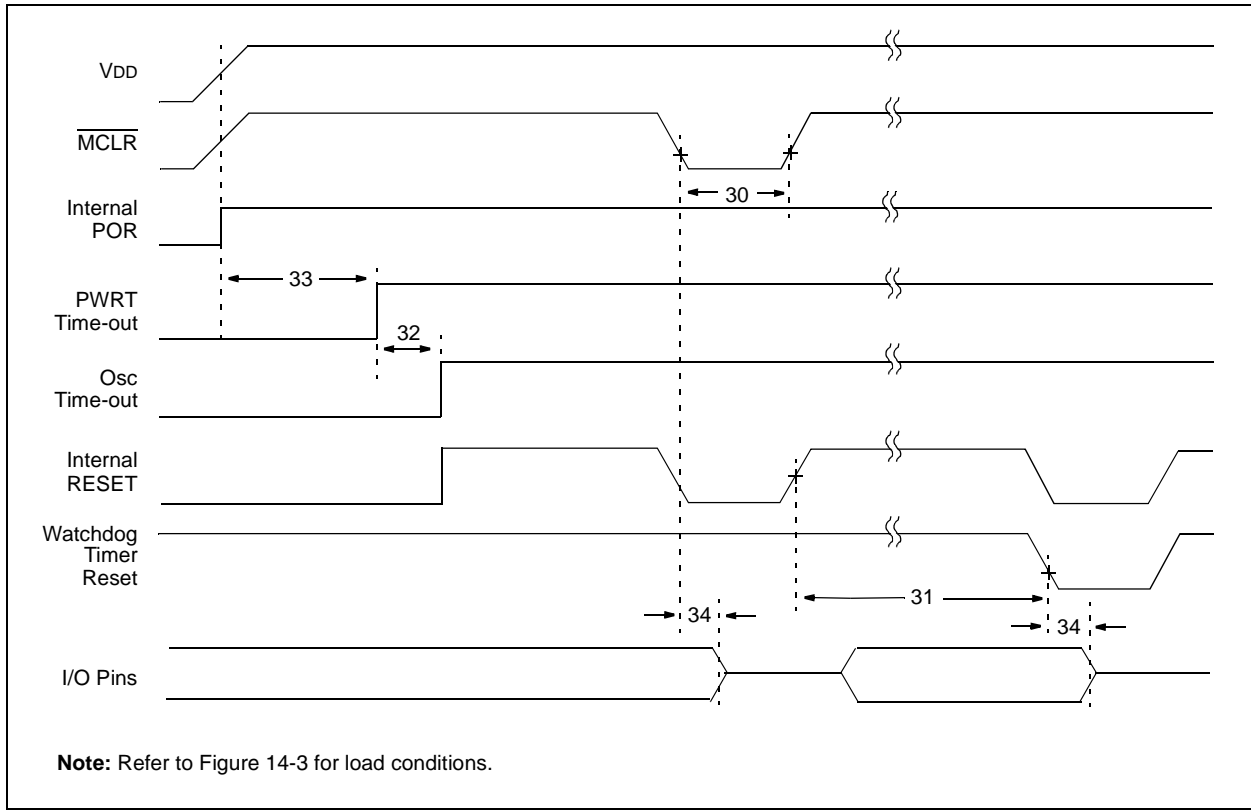


FIGURE 14-7: BROWN-OUT RESET TIMING

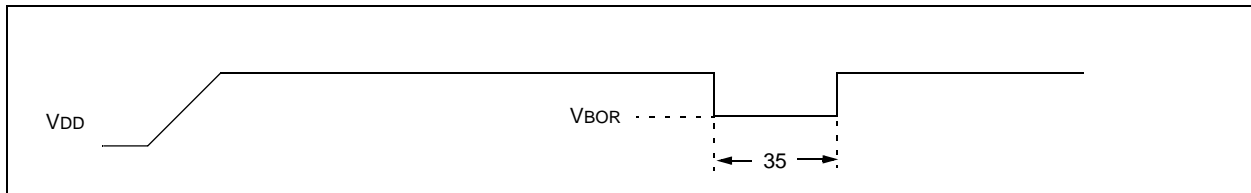


TABLE 14-3: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|-----------|-------|--|-----|-----------|-----|-------|--------------------------|
| 30 | TmCL | MCLR Pulse Width (low) | 2 | — | — | μs | VDD = 5V, -40°C to +85°C |
| 31* | TWDT | Watchdog Timer Time-out Period (No Prescaler) | 7 | 18 | 33 | ms | VDD = 5V, -40°C to +85°C |
| 32 | TOST | Oscillation Start-up Timer Period | — | 1024 TOSC | — | — | TOSC = OSC1 period |
| 33* | TPWRT | Power up Timer Period | 28 | 72 | 132 | ms | VDD = 5V, -40°C to +85°C |
| 34 | TIOZ | I/O Hi-impedance from MCLR Low or Watchdog Timer Reset | — | — | 2.1 | μs | |
| 35 | TBOR | Brown-out Reset pulse width | 100 | — | — | μs | VDD ≤ VBOR (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.

PIC16F870/871

FIGURE 14-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

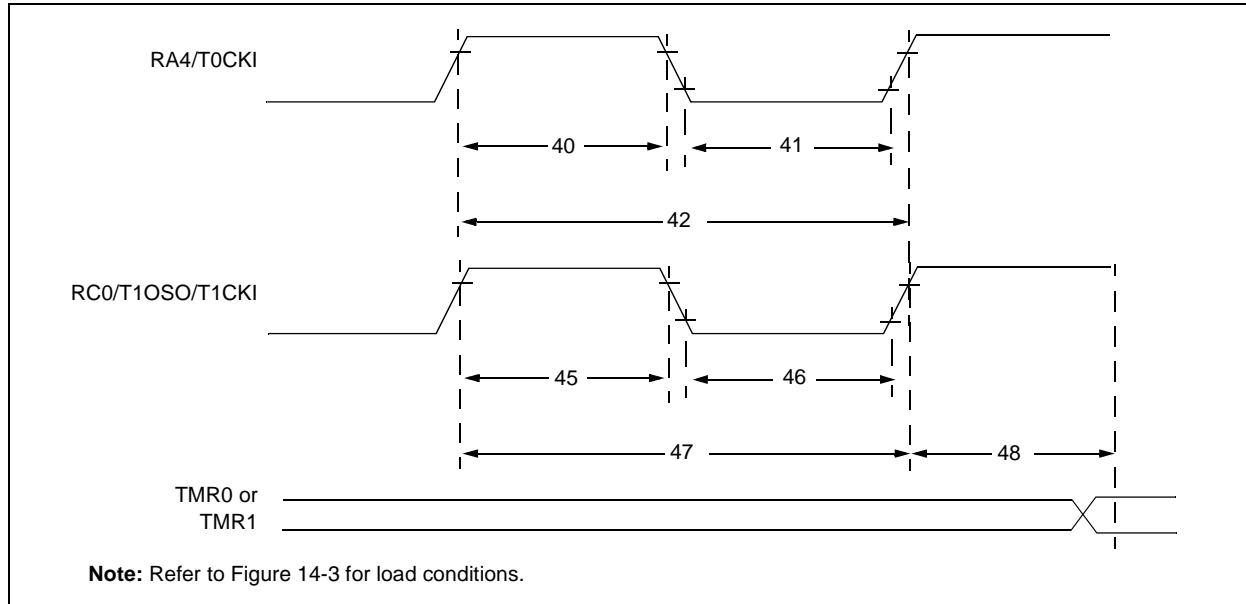


TABLE 14-4: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions | |
|--------------|-----------|---|--------------------------------|--|--|-------------|-------|-------------------------------------|---------------------------------|
| 40* | Tt0H | T0CKI High Pulse Width | No Prescaler | $0.5 T_{CY} + 20$ | — | — | ns | Must also meet parameter 42 | |
| | | | With Prescaler | 10 | — | — | ns | | |
| 41* | Tt0L | T0CKI Low Pulse Width | No Prescaler | $0.5 T_{CY} + 20$ | — | — | ns | Must also meet parameter 42 | |
| | | | With Prescaler | 10 | — | — | ns | | |
| 42* | Tt0P | T0CKI Period | No Prescaler | $T_{CY} + 40$ | — | — | ns | N = prescale value (2, 4, ..., 256) | |
| | | | With Prescaler | Greater of: 20 or $\frac{T_{CY} + 40}{N}$ | — | — | ns | | |
| 45* | Tt1H | T1CKI High Time | Synchronous, Prescaler = 1 | $0.5 T_{CY} + 20$ | — | — | ns | Must also meet parameter 47 | |
| | | | Synchronous, Prescaler = 2,4,8 | Standard(F) | 15 | — | — | | ns |
| | | | | Extended(LF) | 25 | — | — | | ns |
| | | | Asynchronous | Standard(F) | 30 | — | — | | ns |
| Extended(LF) | 50 | — | | — | ns | | | | |
| 46* | Tt1L | T1CKI Low Time | Synchronous, Prescaler = 1 | $0.5 T_{CY} + 20$ | — | — | ns | Must also meet parameter 47 | |
| | | | Synchronous, Prescaler = 2,4,8 | Standard(F) | 15 | — | — | | ns |
| | | | | Extended(LF) | 25 | — | — | | ns |
| | | | Asynchronous | Standard(F) | 30 | — | — | | ns |
| Extended(LF) | 50 | — | | — | ns | | | | |
| 47* | Tt1P | T1CKI input period | Synchronous | Standard(F) | Greater of: 30 or $\frac{T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1, 2, 4, 8) |
| | | | | Extended(LF) | Greater of: 50 or $\frac{T_{CY} + 40}{N}$ | | | | N = prescale value (1, 2, 4, 8) |
| | | | Asynchronous | Standard(F) | 60 | — | — | ns | |
| | | | | Extended(LF) | 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 | | $2 T_{osc}$ | — | $7 T_{osc}$ | — | | |

* 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 14-9: CAPTURE/COMPARE/PWM TIMINGS (CCP1)

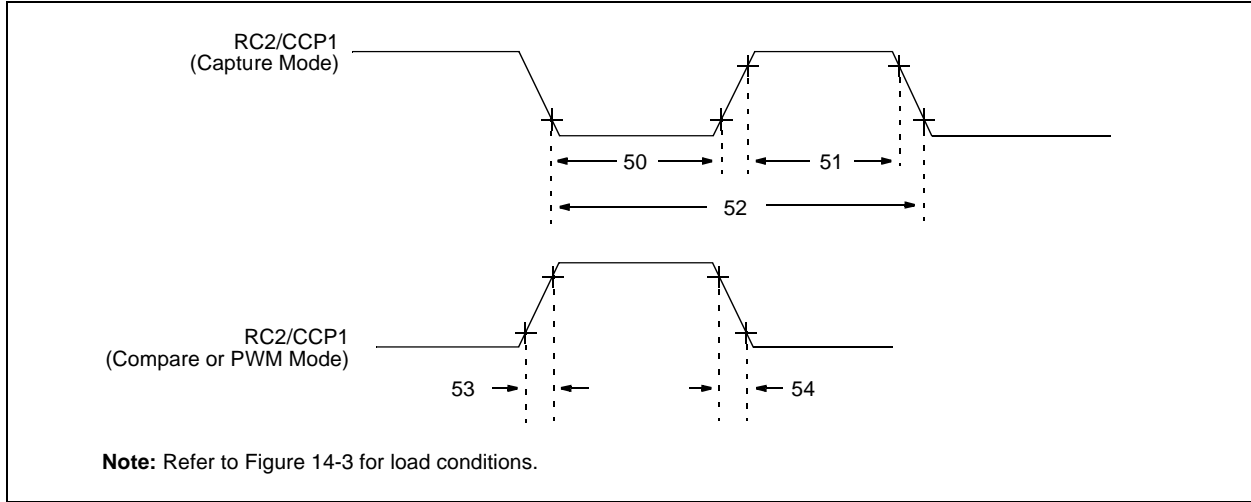


TABLE 14-5: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1)

| Param No. | Sym | Characteristic | | Min | Typ† | Max | Units | Conditions | |
|-----------|------|-----------------------|---------------------------|-------------------|------|-----|--------------------------------|------------|----|
| 50* | TccL | * CCP1 input low time | No Prescaler | $0.5 T_{CY} + 20$ | — | — | ns | | |
| | | | With Prescaler | Standard(F) | 10 | — | — | | ns |
| | | | | Extended(LF) | 20 | — | — | | ns |
| 51* | TccH | CCP1 input high time | No Prescaler | $0.5 T_{CY} + 20$ | — | — | ns | | |
| | | | With Prescaler | Standard(F) | 10 | — | — | | ns |
| | | | | Extended(LF) | 20 | — | — | | ns |
| 52* | TccP | CCP1 input period | $\frac{3 T_{CY} + 40}{N}$ | — | — | ns | N = prescale value (1,4 or 16) | | |
| 53* | TccR | CCP1 output rise time | Standard(F) | — | 10 | 25 | ns | | |
| | | | Extended(LF) | — | 25 | 50 | ns | | |
| 54* | TccF | CCP1 output fall time | Standard(F) | — | 10 | 25 | ns | | |
| | | | Extended(LF) | — | 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.

PIC16F870/871

FIGURE 14-10: PARALLEL SLAVE PORT TIMING (PIC16F871 ONLY)

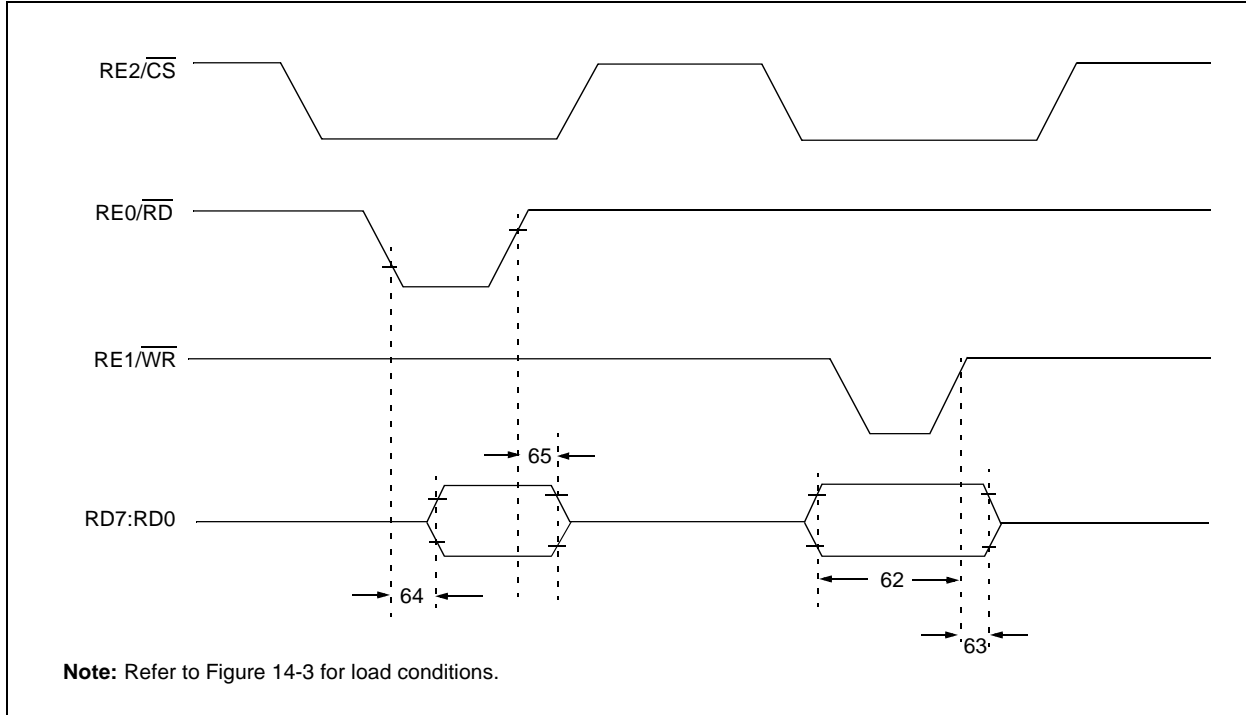


TABLE 14-6: PARALLEL SLAVE PORT REQUIREMENTS (PIC16F871 ONLY)

| 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 | Extended range only | |
| | | | 25 | — | — | ns | | |
| 63* | TwrH2dtl | $\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ to data-in invalid (hold time) | Standard(F) | 20 | — | — | ns | |
| | | | Extended(LF) | 35 | — | — | ns | |
| 64 | TrdL2dtV | $\overline{RD}\downarrow$ and $\overline{CS}\downarrow$ to data-out valid | — | — | 80 | ns | Extended range only | |
| | | | — | — | 90 | ns | | |
| 65 | TrdH2dtl | $\overline{RD}\uparrow$ or $\overline{CS}\downarrow$ to data-out invalid | 10 | — | 30 | ns | | |

* These parameters are characterized but not tested.

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

FIGURE 14-11: USART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

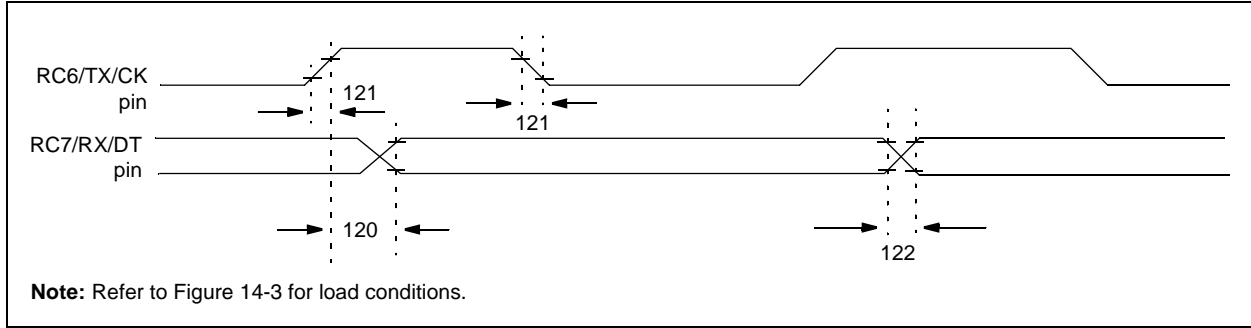


TABLE 14-7: USART SYNCHRONOUS TRANSMISSION REQUIREMENTS

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

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

FIGURE 14-12: USART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

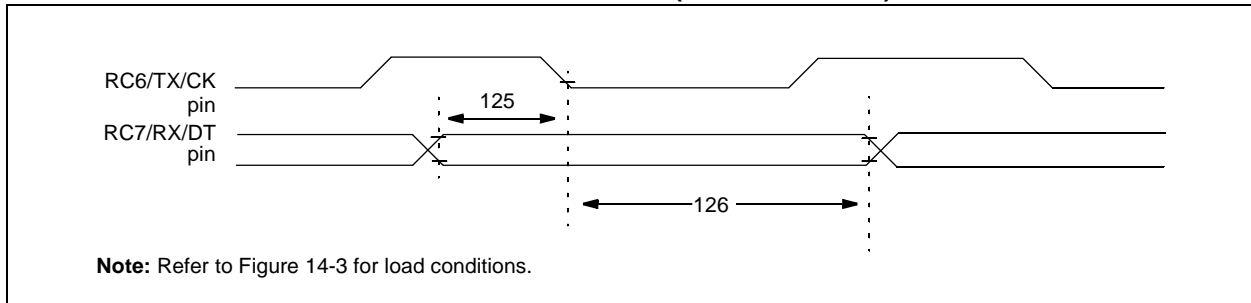


TABLE 14-8: USART SYNCHRONOUS RECEIVE REQUIREMENTS

| 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 | TckL2dtH | Data hold after CK ↓ (DT hold time) | 15 | — | — | ns | |

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

PIC16F870/871

**TABLE 14-9: PIC16F870/871 (INDUSTRIAL)
PIC16LF870/871 (INDUSTRIAL)**

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|-----------|-------------------|--|------------------------|------------|--------------------------|-------|--|
| A01 | NR | Resolution | — | — | 10-bits | bit | $V_{REF} = V_{DD} = 5.12V$, $V_{SS} \leq V_{AIN} \leq V_{REF}$ |
| A03 | EIL | Integral linearity error | — | — | $< \pm 1$ | LSb | $V_{REF} = V_{DD} = 5.12V$, $V_{SS} \leq V_{AIN} \leq V_{REF}$ |
| A04 | EDL | Differential linearity error | — | — | $< \pm 1$ | LSb | $V_{REF} = V_{DD} = 5.12V$, $V_{SS} \leq V_{AIN} \leq V_{REF}$ |
| A06 | E _{OFF} | Offset error | — | — | $< \pm 1$ | LSb | $V_{REF} = V_{DD} = 5.12V$, $V_{SS} \leq V_{AIN} \leq V_{REF}$ |
| A07 | E _{GN} | Gain error | — | — | $< \pm 1$ | LSb | $V_{REF} = V_{DD} = 5.12V$, $V_{SS} \leq V_{AIN} \leq V_{REF}$ |
| A10 | — | Monotonicity ⁽³⁾ | — | guaranteed | — | — | $V_{SS} \leq V_{AIN} \leq V_{REF}$ |
| A20 | V _{REF} | Reference voltage (V _{REF+} – V _{REF-}) | 2.0V | — | V _{DD} + 0.3 | V | |
| A21 | V _{REF+} | Reference voltage High | V _{DD} – 2.5V | — | V _{DD} + 0.3V | V | Must meet spec. A20 |
| A22 | V _{REF-} | Reference voltage Low | V _{SS} – 0.3V | — | V _{REF+} – 2.0V | V | Must meet spec. A20 |
| A25 | V _{AIN} | Analog input voltage | V _{SS} – 0.3 | — | V _{REF} + 0.3 | V | |
| A30 | Z _{AIN} | Recommended impedance of analog voltage source | — | — | 10.0 | kΩ | |
| A40 | I _{AD} | A/D conversion current (V _{DD}) | Standard(F) | 220 | — | μA | Average current consumption when A/D is on (Note 1). |
| | | | Extended(LF) | 90 | — | μA | |
| A50 | I _{REF} | V _{REF} input current (Note 2) | 10 | — | 1000 | μA | During V _{AIN} acquisition. Based on differential of V _{HOLD} to V _{AIN} to charge CHOLD, see Section 10.1. During A/D Conversion cycle |
| | | | — | — | 10 | μA | |

* These parameters are characterized but not tested.

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

- Note 1:** When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.
- 2:** V_{REF} current is from RA3 pin or V_{DD} pin, whichever is selected as reference input.
- 3:** The A/D conversion result never decreases with an increase in the input voltage, and has no missing codes.

FIGURE 14-13: A/D CONVERSION TIMING

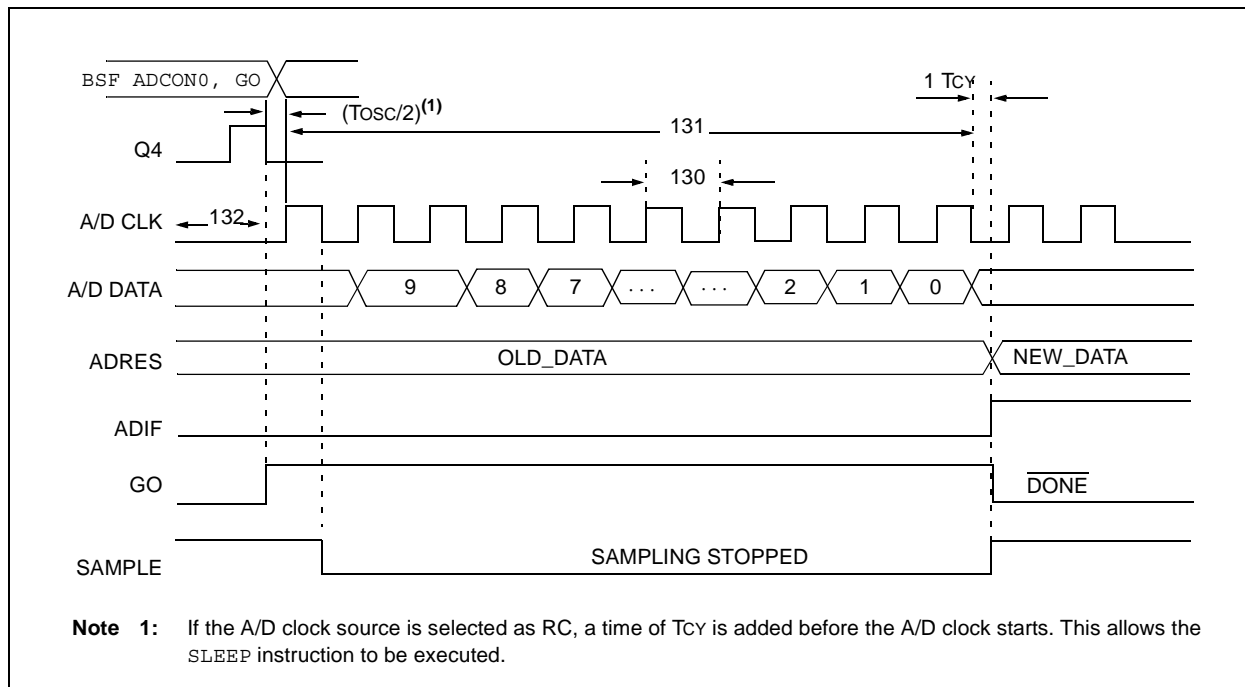


TABLE 14-10: A/D CONVERSION REQUIREMENTS

| Param No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions | |
|-----------|------|--|--------------|---------------|-----|---------------|---|--|
| 130 | TAD | A/D clock period | Standard(F) | 1.6 | — | — | μs | TOSC based, $V_{REF} \geq 3.0\text{V}$ |
| | | | Extended(LF) | 3.0 | — | — | μs | TOSC based, $V_{REF} \geq 2.0\text{V}$ |
| | | A/D RC Mode | Standard(F) | 2.0 | 4.0 | 6.0 | μs | |
| | | | Extended(LF) | 3.0 | 6.0 | 9.0 | μs | |
| 131 | TCNV | Conversion time (not including S/H time) (Note 1) | | — | 12 | TAD | | |
| 132 | TACQ | Acquisition time | (Note 2) | 40 | — | μs | The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 20.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD). | |
| | | | 10* | — | — | μs | | |
| 134 | TGO | Q4 to A/D clock start | — | $T_{osc}/2$ § | — | — | If the A/D clock source is selected as RC, a time of T_{CY} is added before the A/D clock starts. This allows the <code>SLEEP</code> instruction to be executed. | |

* These parameters are characterized but not tested.

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

§ This specification ensured by design.

Note 1: ADRES register may be read on the following T_{CY} cycle.

2: See Section 10.1 for min conditions.

PIC16F870/871

NOTES:

15.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

Note: The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

“Typical” represents the mean of the distribution at 25°C. “Maximum” or “minimum” represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over the whole temperature range.

FIGURE 15-1: TYPICAL I_{DD} vs. F_{OSC} OVER V_{DD} (HS MODE)

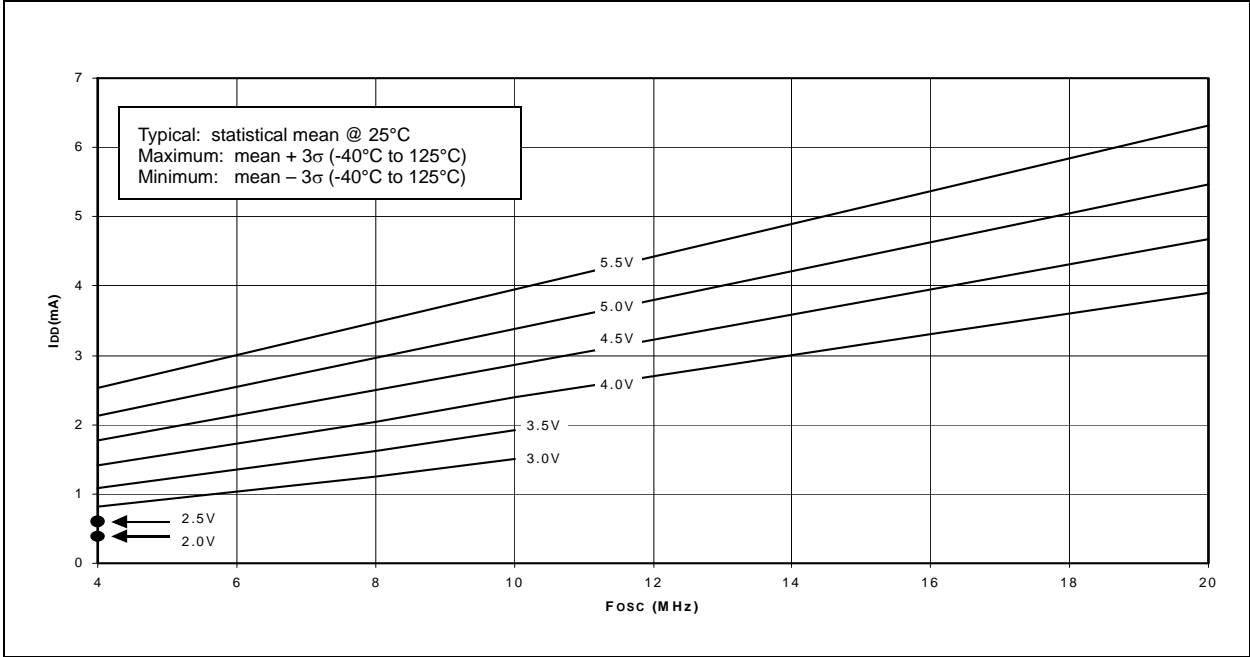
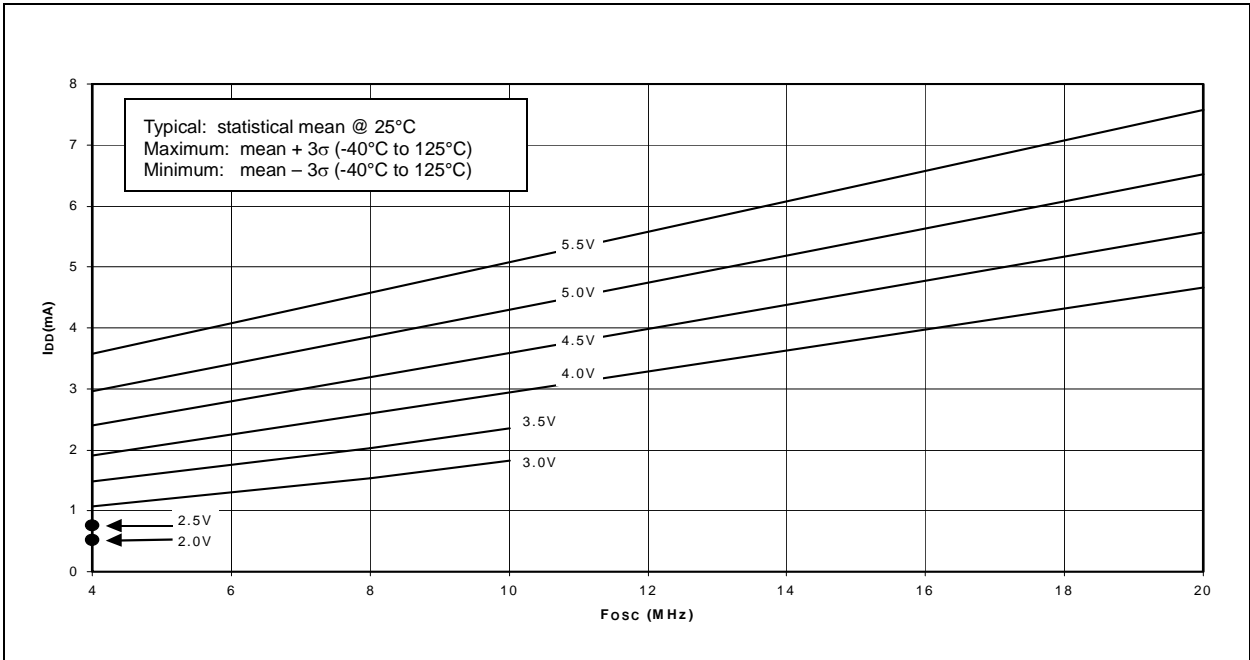


FIGURE 15-2: MAXIMUM I_{DD} vs. F_{OSC} OVER V_{DD} (HS MODE)



PIC16F870/871

FIGURE 15-3: TYPICAL I_{DD} vs. Fosc OVER V_{DD} (XT MODE)

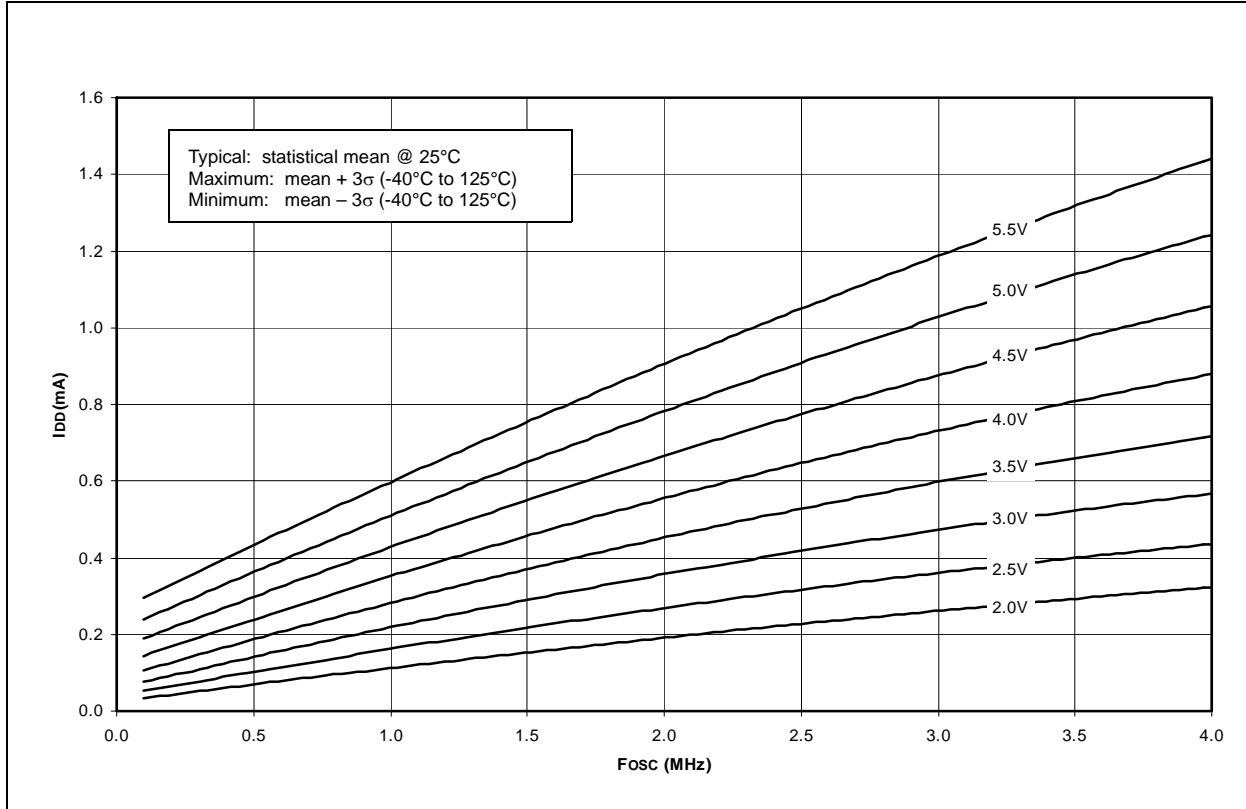


FIGURE 15-4: MAXIMUM I_{DD} vs. Fosc OVER V_{DD} (LP MODE)

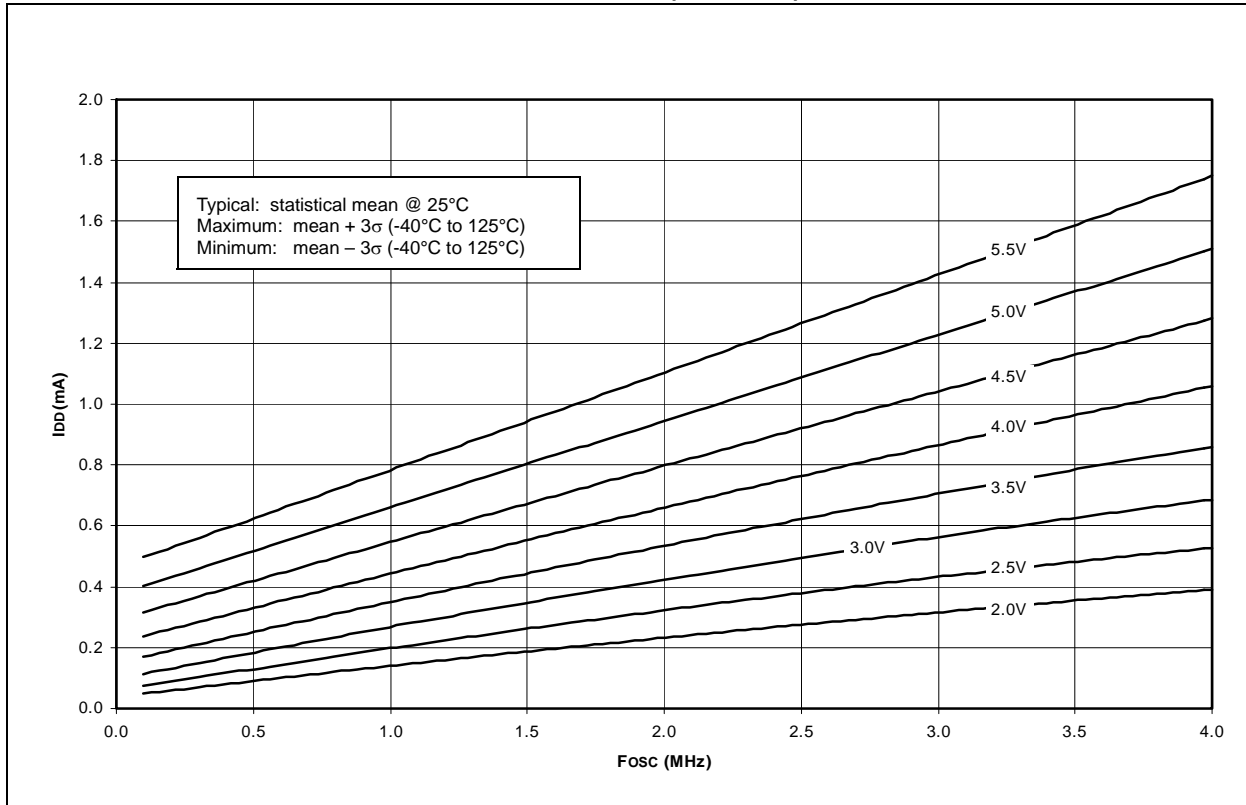


FIGURE 15-5: TYPICAL I_{DD} vs. F_{osc} OVER V_{DD} (LP MODE)

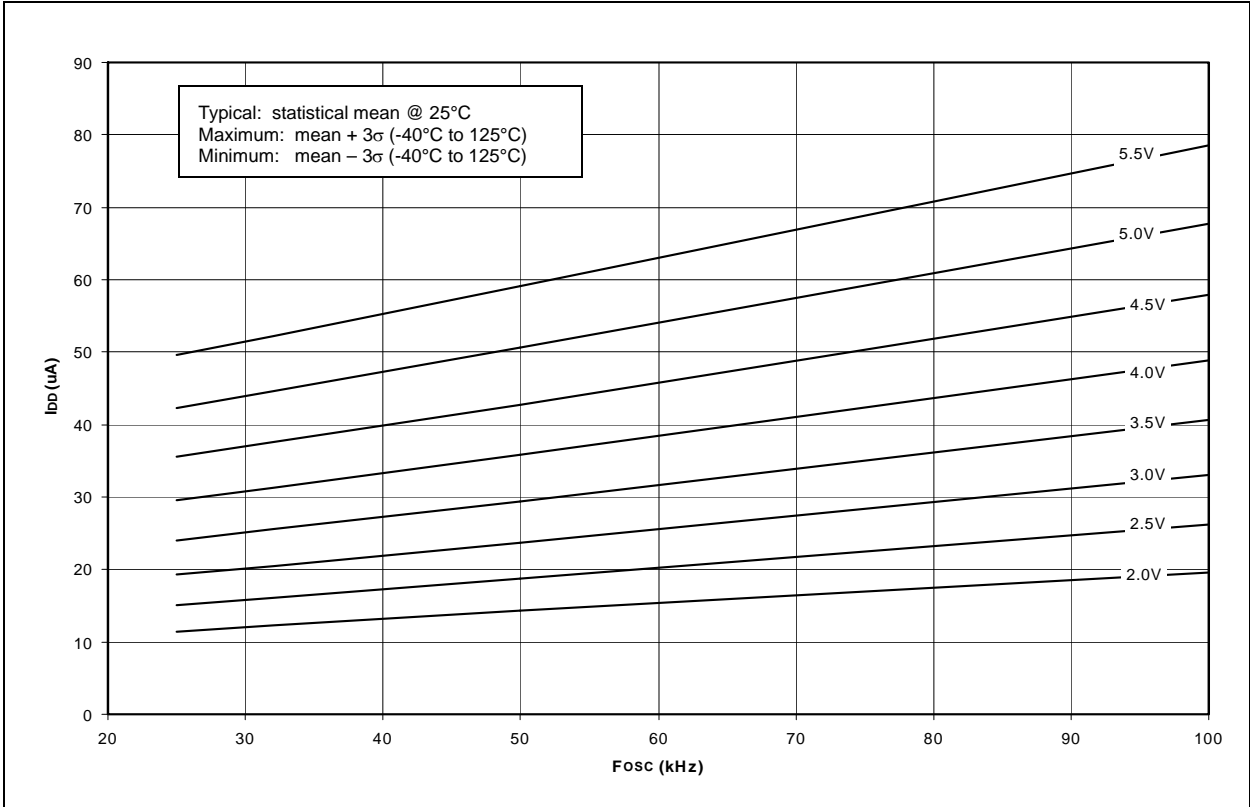
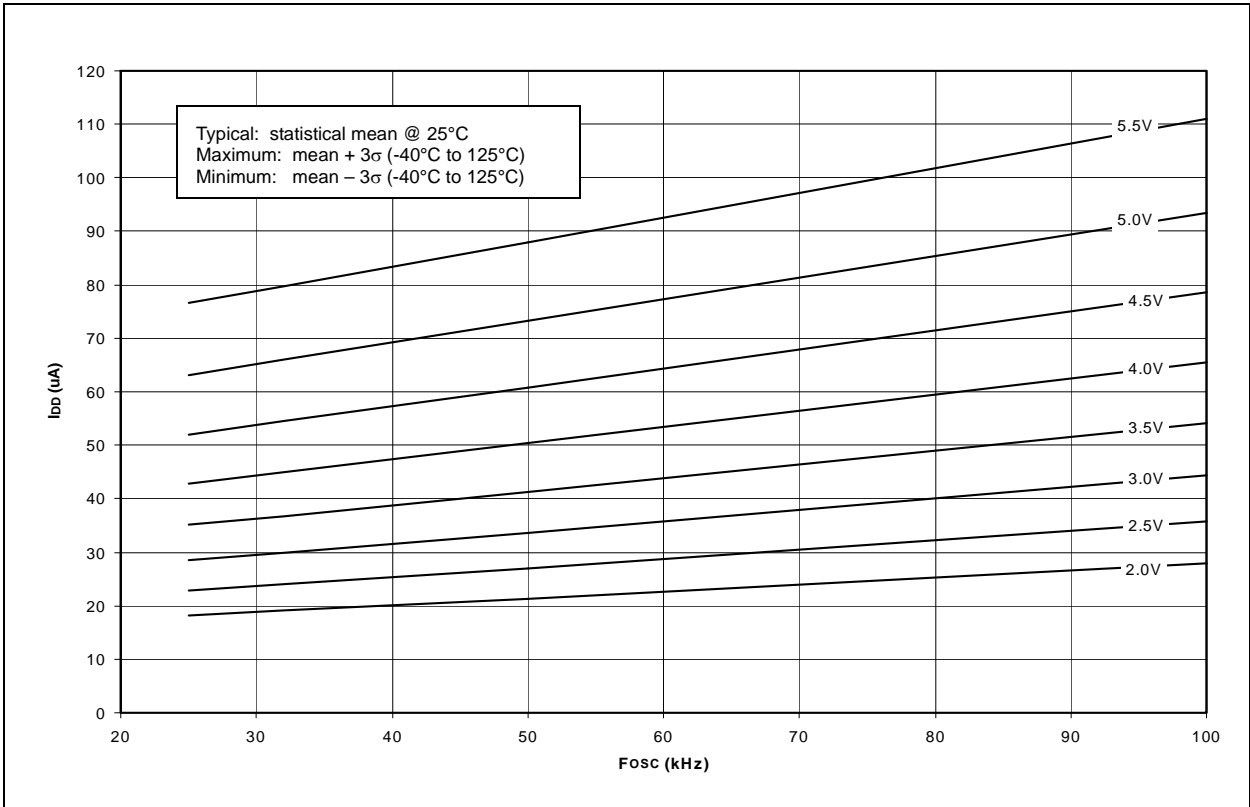


FIGURE 15-6: MAXIMUM I_{DD} vs. F_{osc} OVER V_{DD} (XT MODE)



PIC16F870/871

FIGURE 15-7: AVERAGE F_{osc} vs. V_{DD} FOR VARIOUS VALUES OF R (RC MODE, C = 20 pF, 25°C)

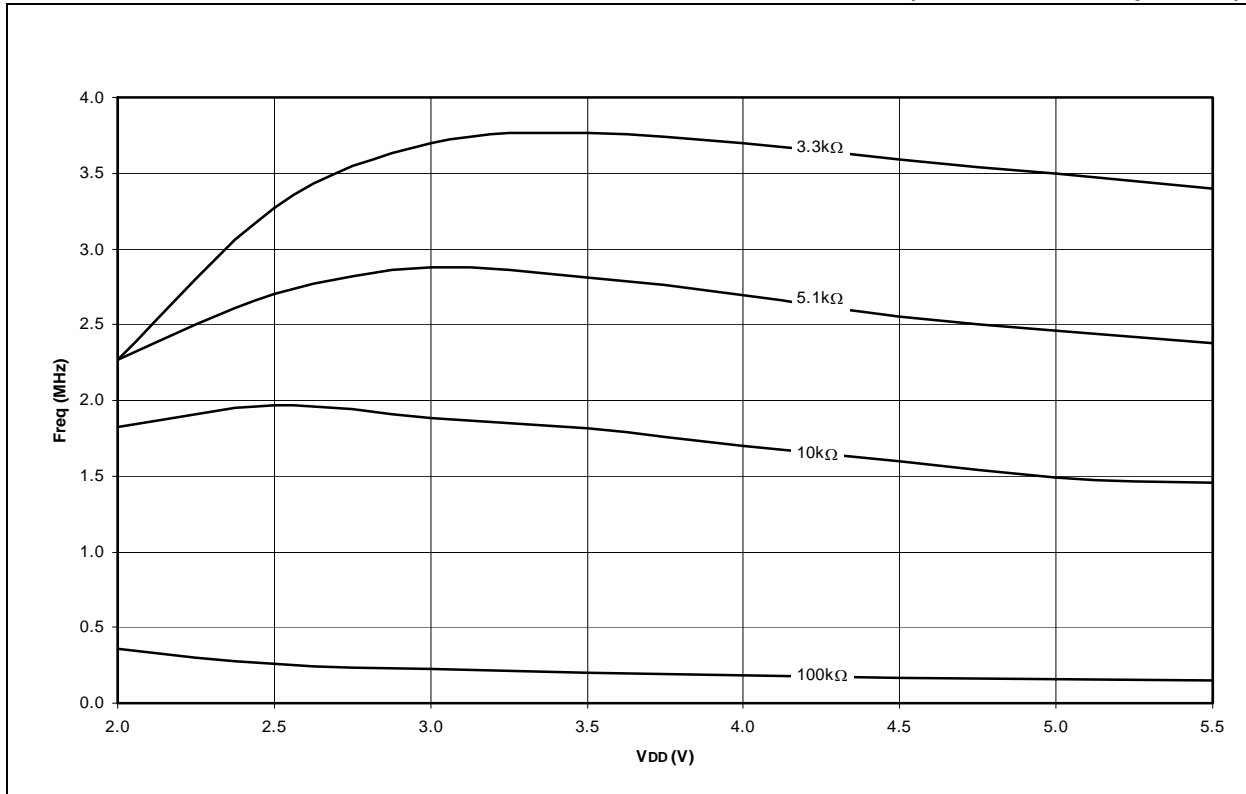


FIGURE 15-8: AVERAGE F_{osc} vs. V_{DD} FOR VARIOUS VALUES OF R (RC MODE, C = 100 pF, 25°C)

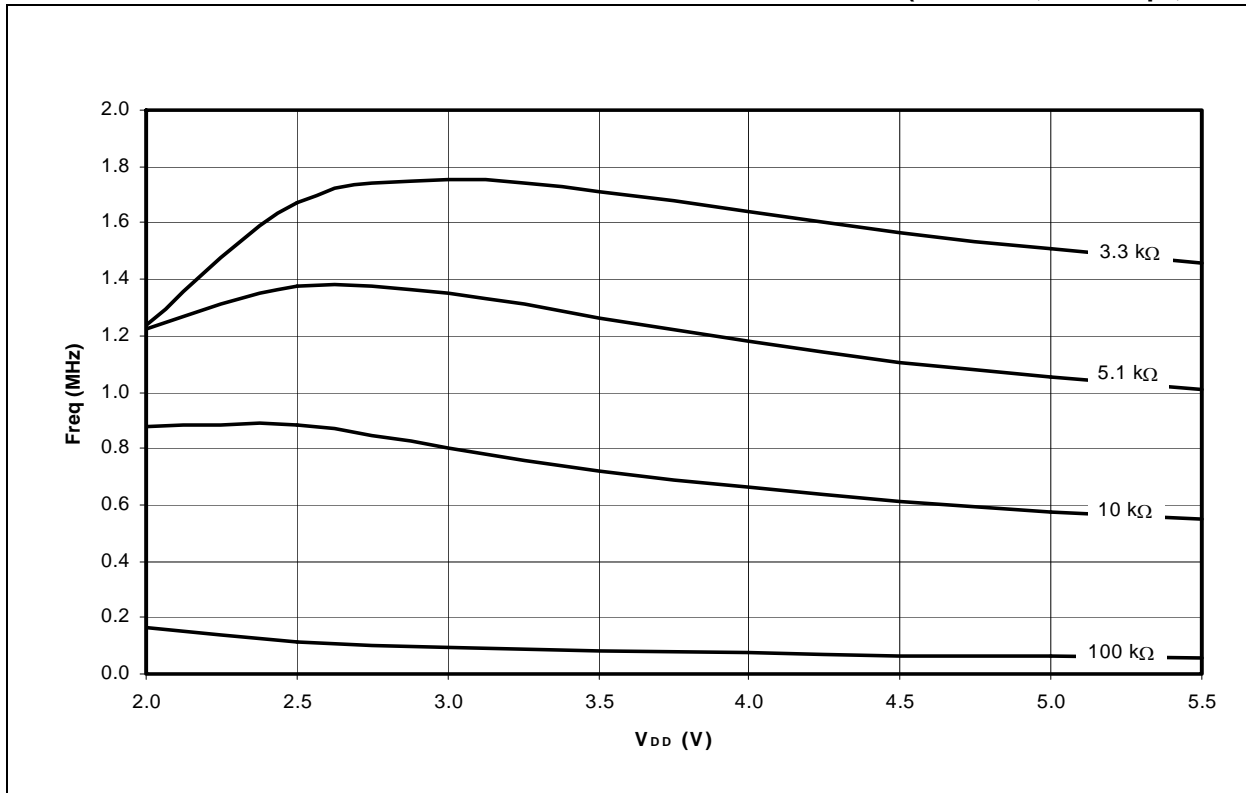


FIGURE 15-9: AVERAGE Fosc vs. VDD FOR VARIOUS VALUES OF R (RC MODE, C = 300 pF, 25°C)

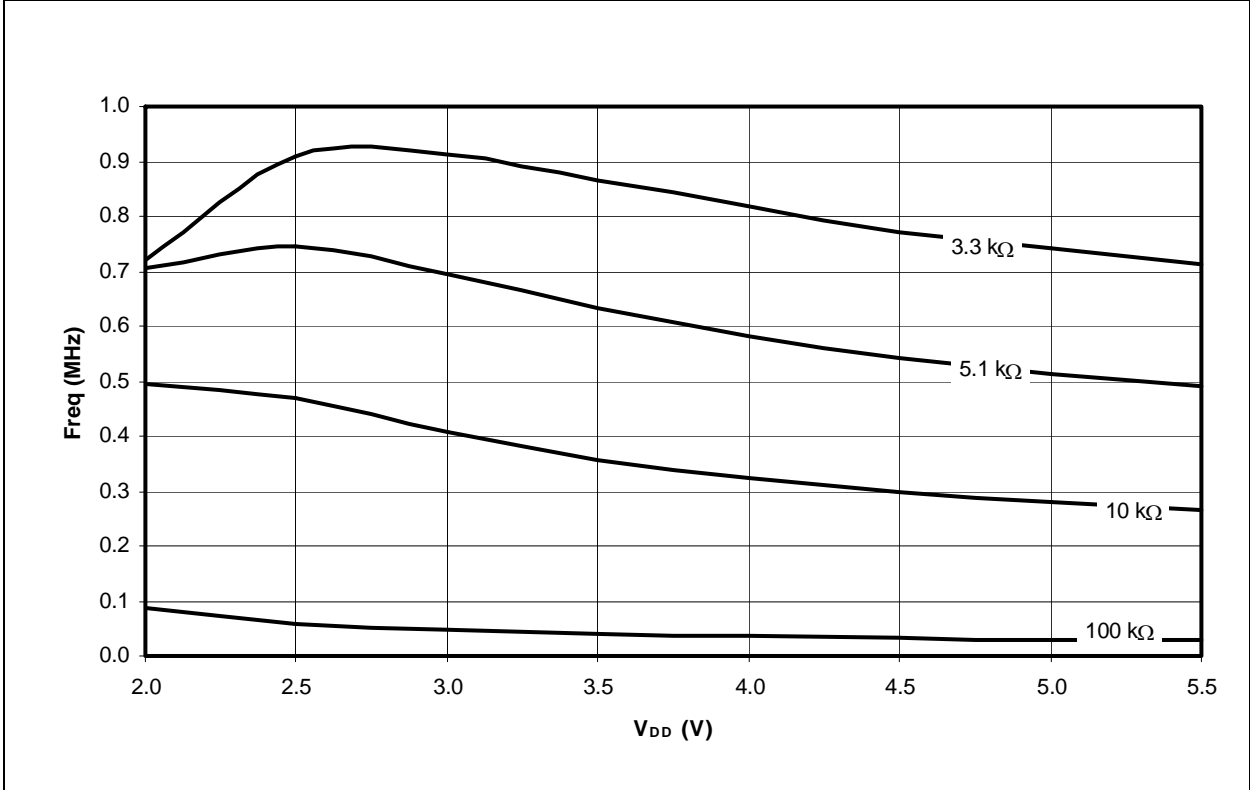
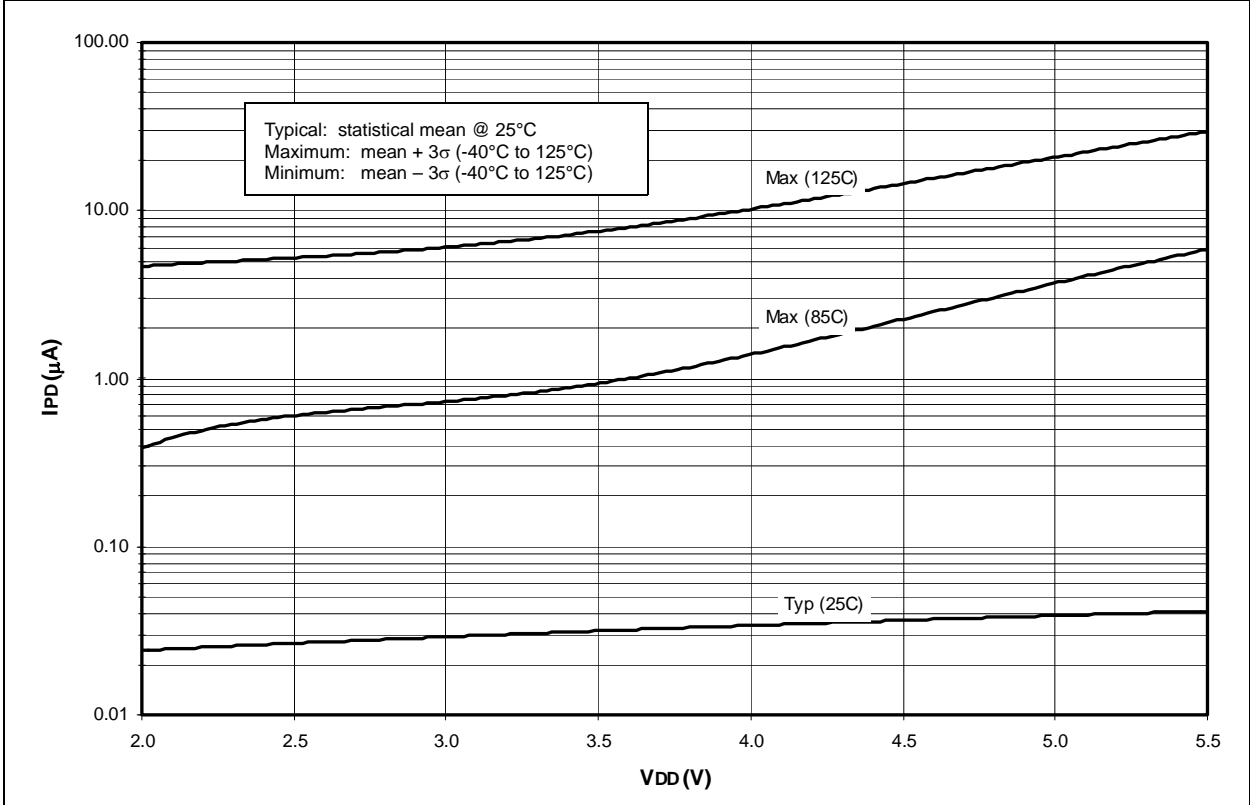


FIGURE 15-10: IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)



PIC16F870/871

FIGURE 15-11: ΔI_{BOR} vs. V_{DD} OVER TEMPERATURE

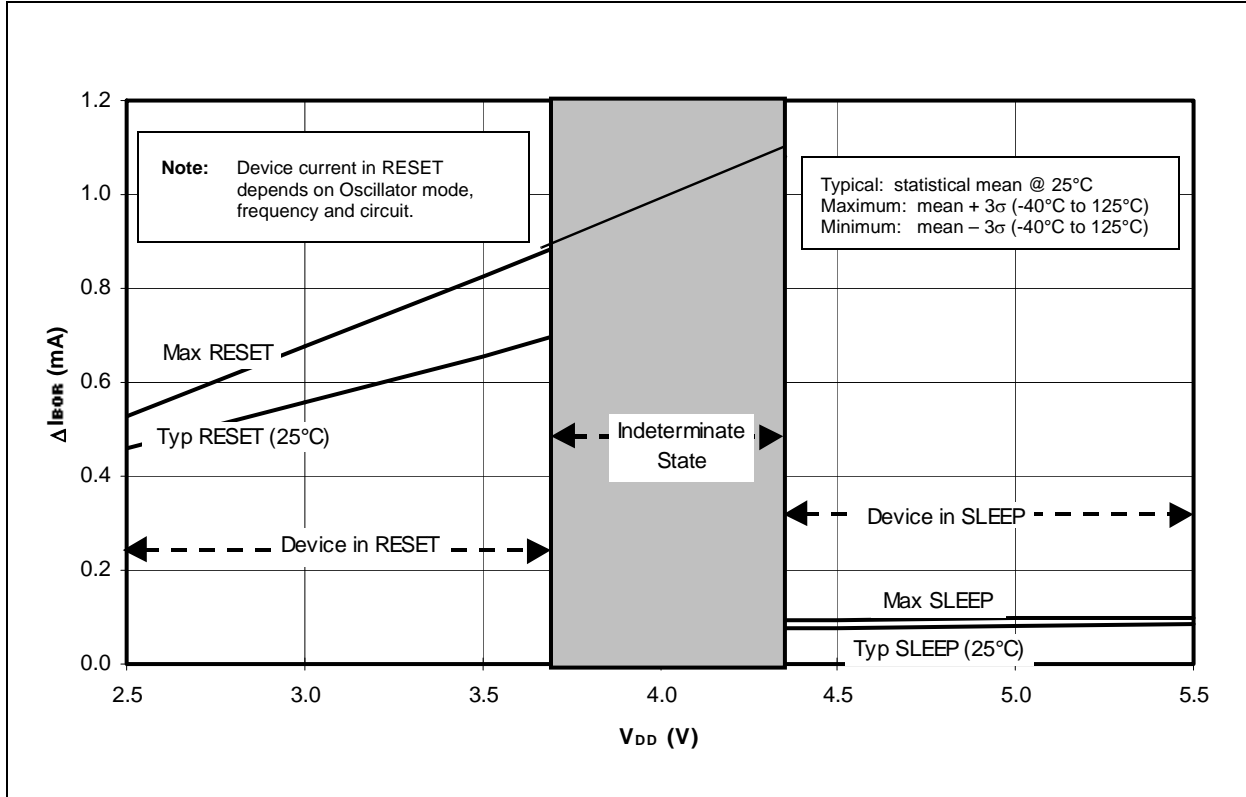


FIGURE 15-12: TYPICAL AND MAXIMUM ΔI_{TMR1} vs. V_{DD} OVER TEMPERATURE (-10°C TO 70°C, TIMER1 WITH OSCILLATOR, XTAL=32 KHZ, C1 AND C2=50 pF)

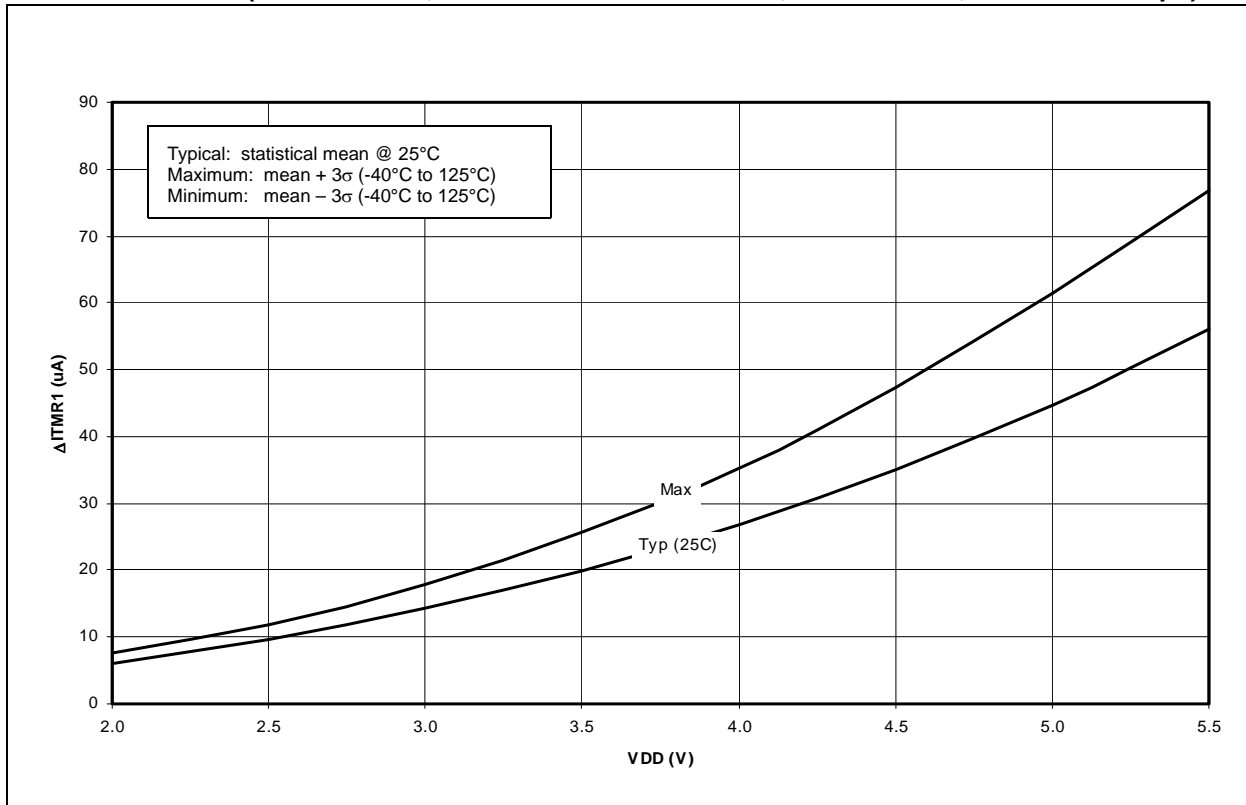


FIGURE 15-13: TYPICAL AND MAXIMUM ΔI_{WDT} vs. V_{DD} OVER TEMPERATURE

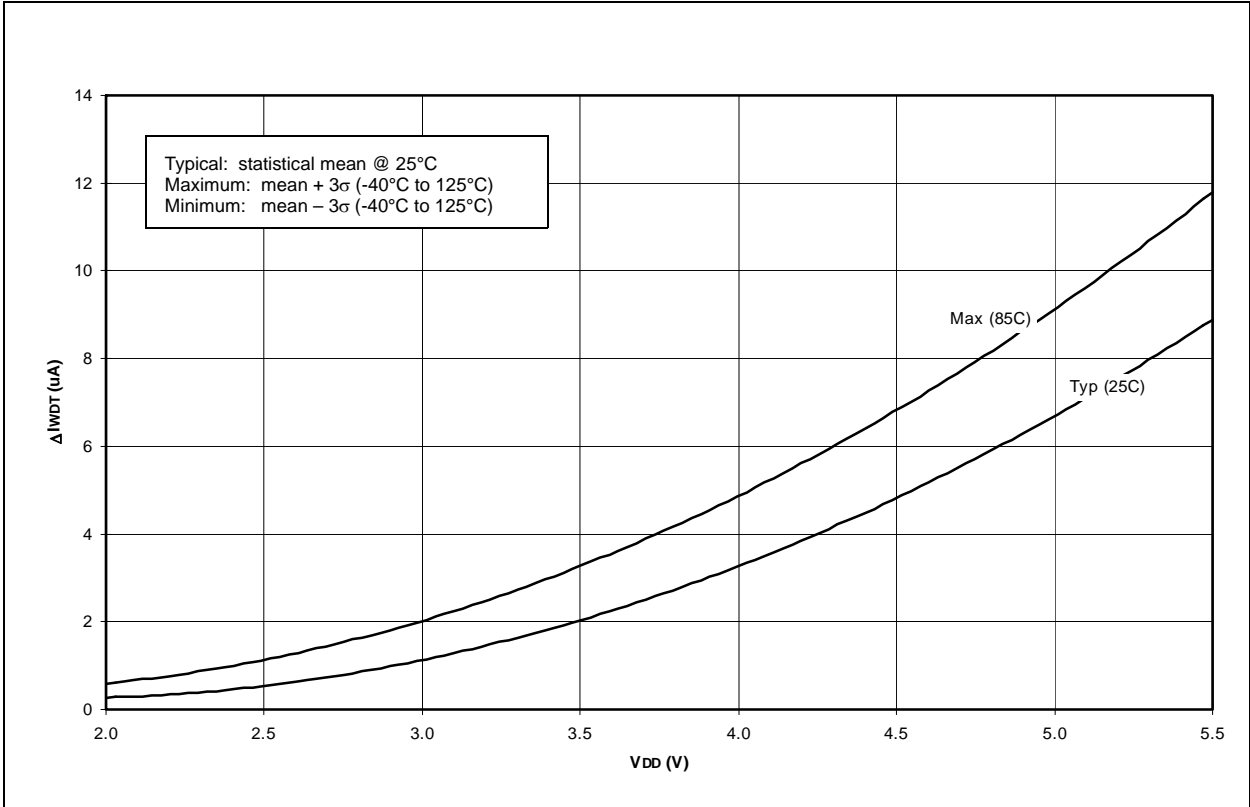
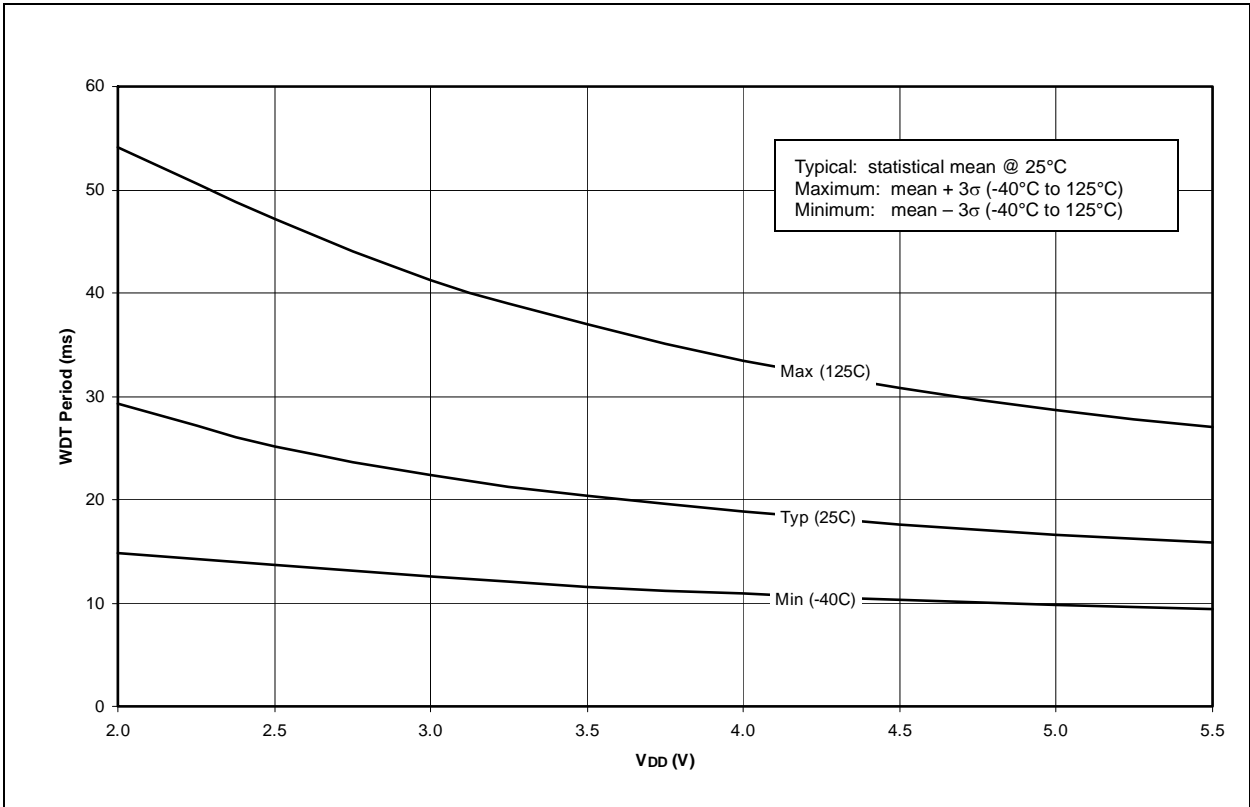


FIGURE 15-14: TYPICAL, MINIMUM AND MAXIMUM WDT PERIOD vs. V_{DD} (-40°C TO 125°C)



PIC16F870/871

FIGURE 15-15: AVERAGE WDT PERIOD vs. VDD OVER TEMPERATURE (-40°C TO 125°C)

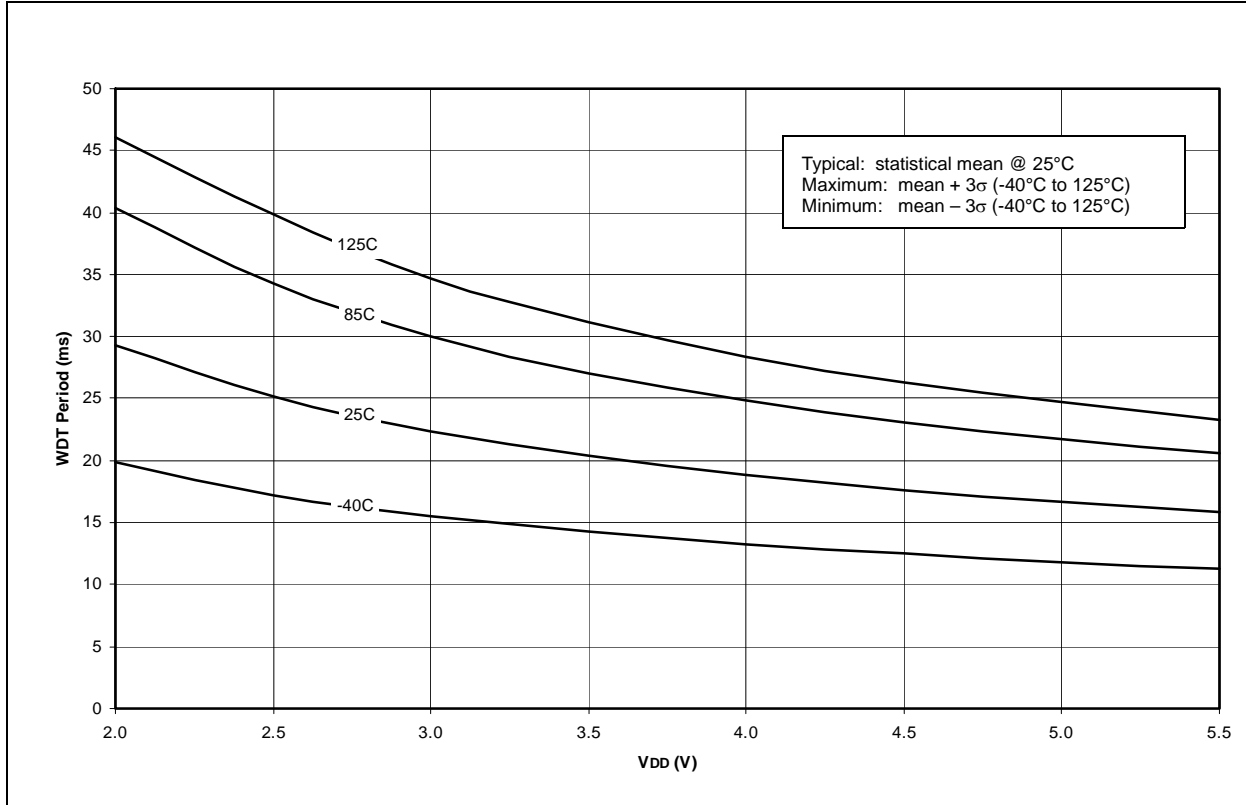


FIGURE 15-16: TYPICAL, MINIMUM AND MAXIMUM VOH vs. IOH (VDD = 5V, -40°C TO 125°C)

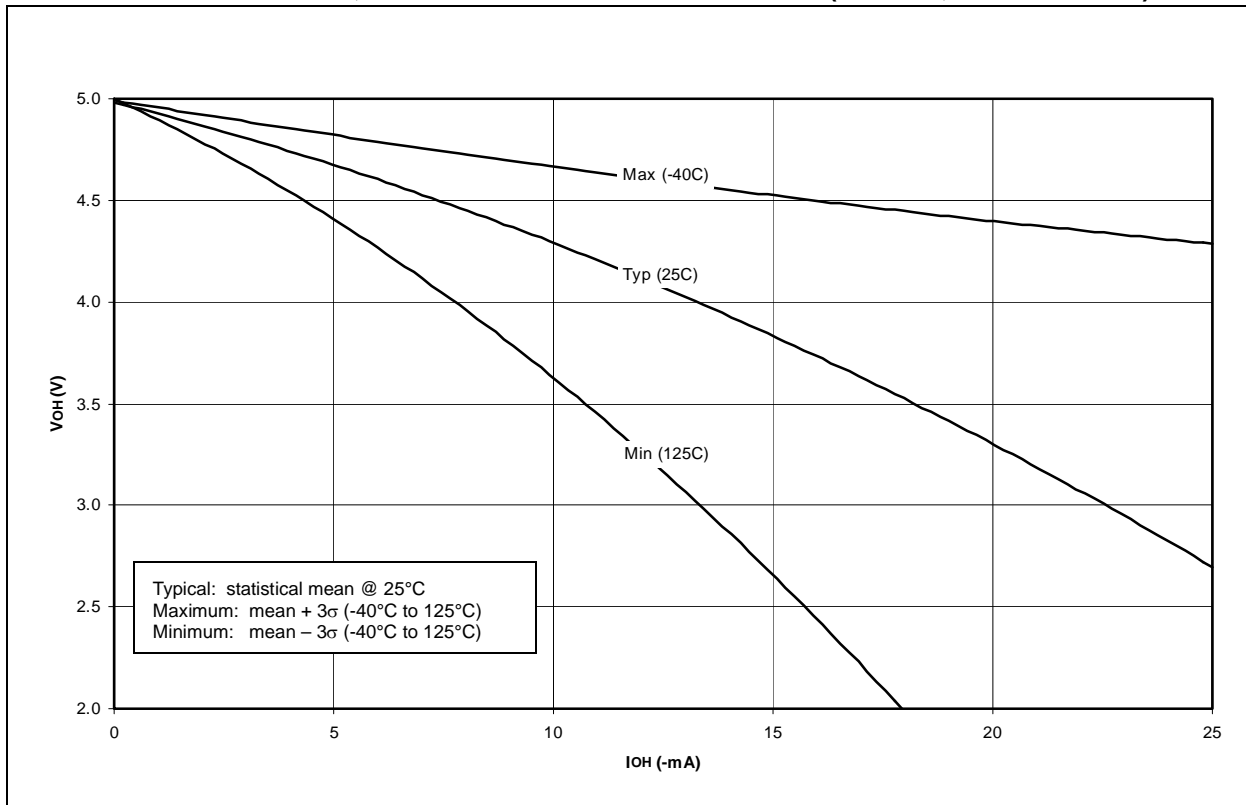


FIGURE 15-17: TYPICAL, MINIMUM AND MAXIMUM V_{OH} vs. I_{OH} ($V_{DD} = 3V$, $-40^{\circ}C$ TO $125^{\circ}C$)

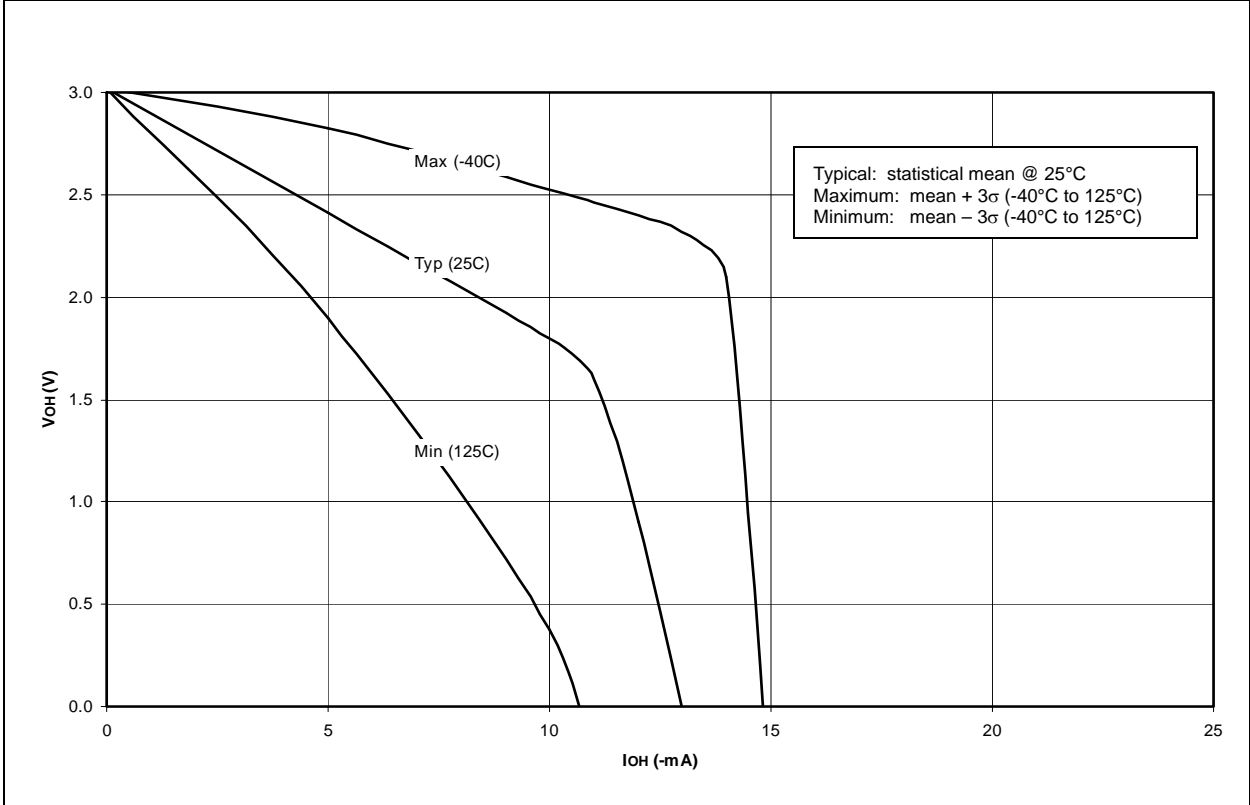
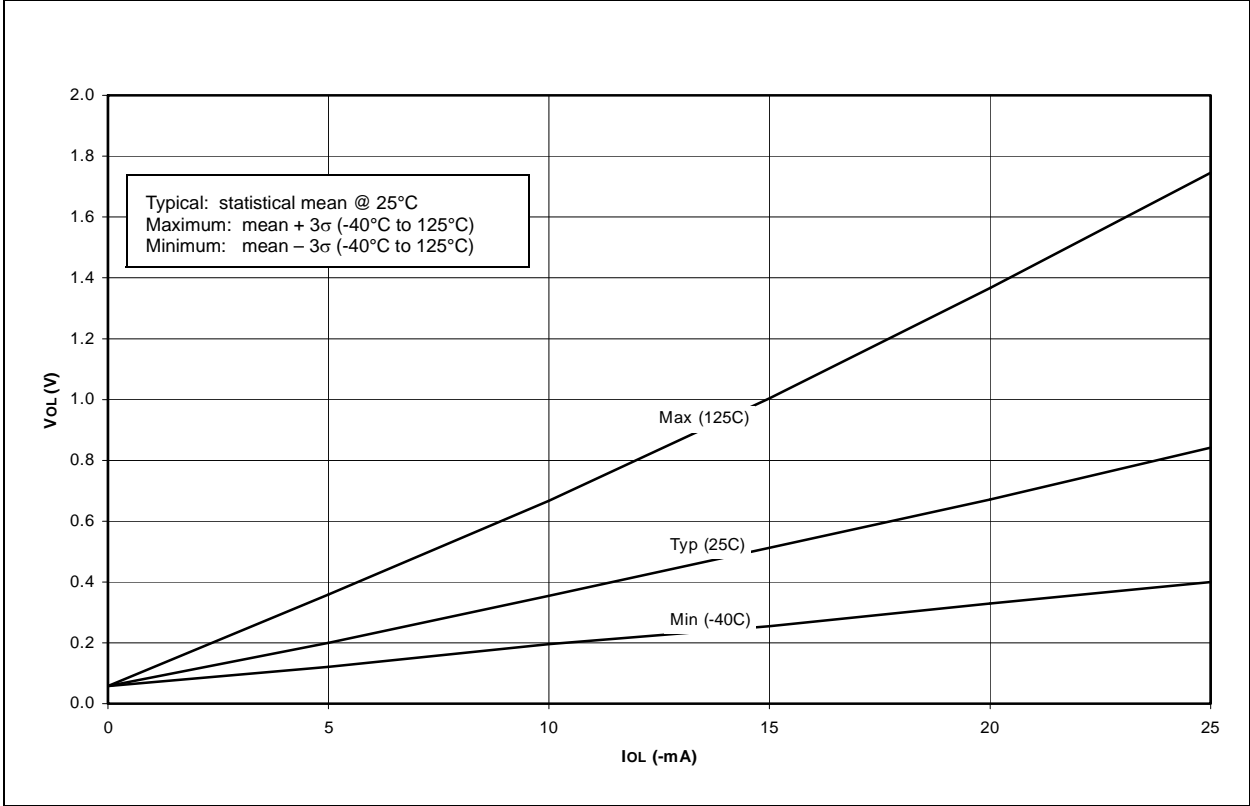


FIGURE 15-18: TYPICAL, MINIMUM AND MAXIMUM V_{OL} vs. I_{OL} ($V_{DD} = 5V$, $-40^{\circ}C$ TO $125^{\circ}C$)



PIC16F870/871

FIGURE 15-19: TYPICAL, MINIMUM AND MAXIMUM V_{OL} vs. I_{OL} ($V_{DD} = 3V$, $-40^{\circ}C$ TO $125^{\circ}C$)

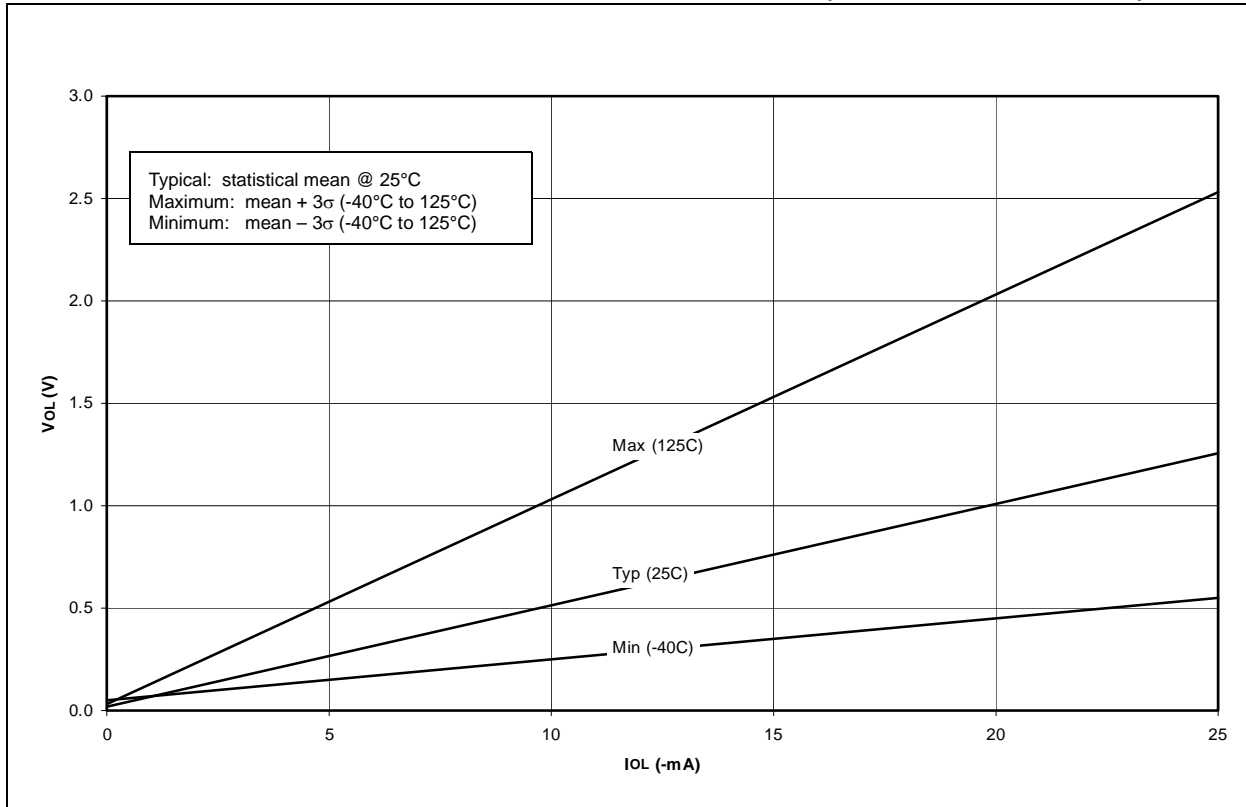


FIGURE 15-20: MINIMUM AND MAXIMUM V_{IN} vs. V_{DD} , (TTL INPUT, $-40^{\circ}C$ TO $125^{\circ}C$)

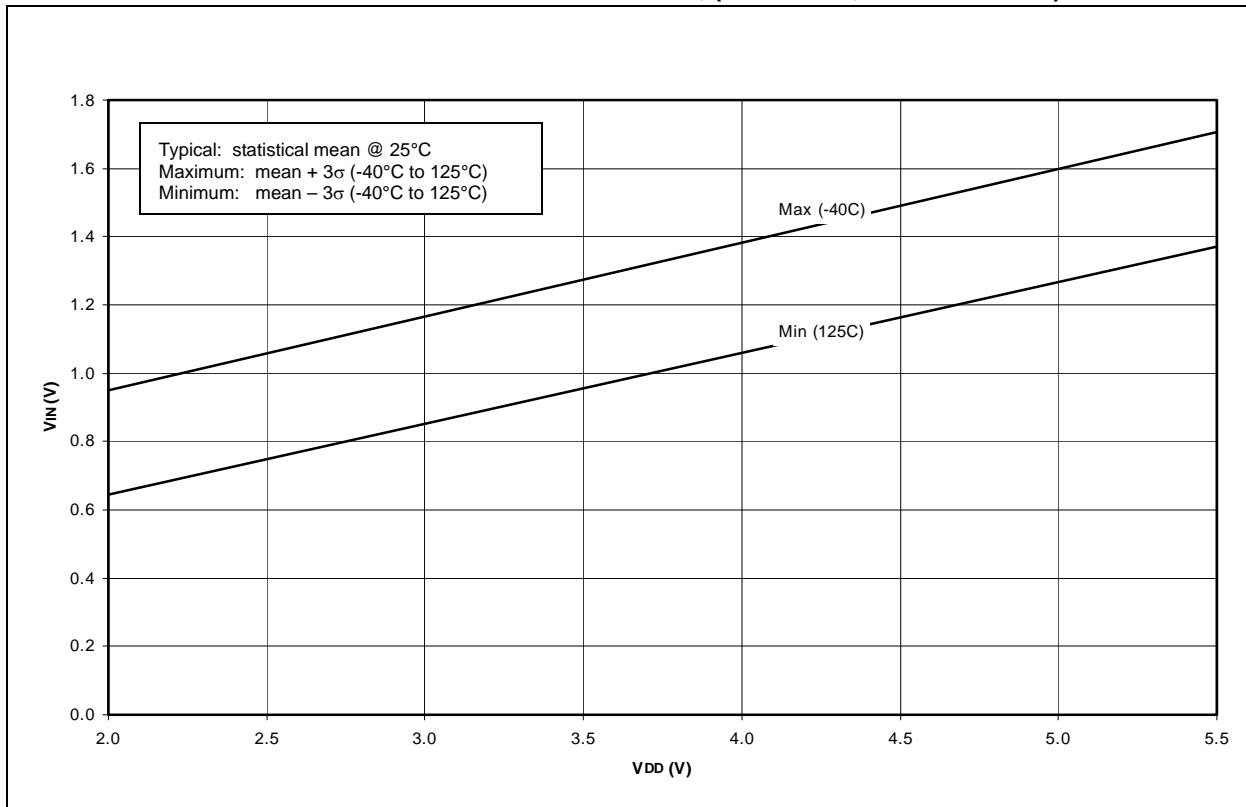


FIGURE 15-21: MINIMUM AND MAXIMUM V_{IN} vs. V_{DD} (ST INPUT, -40°C TO 125°C)

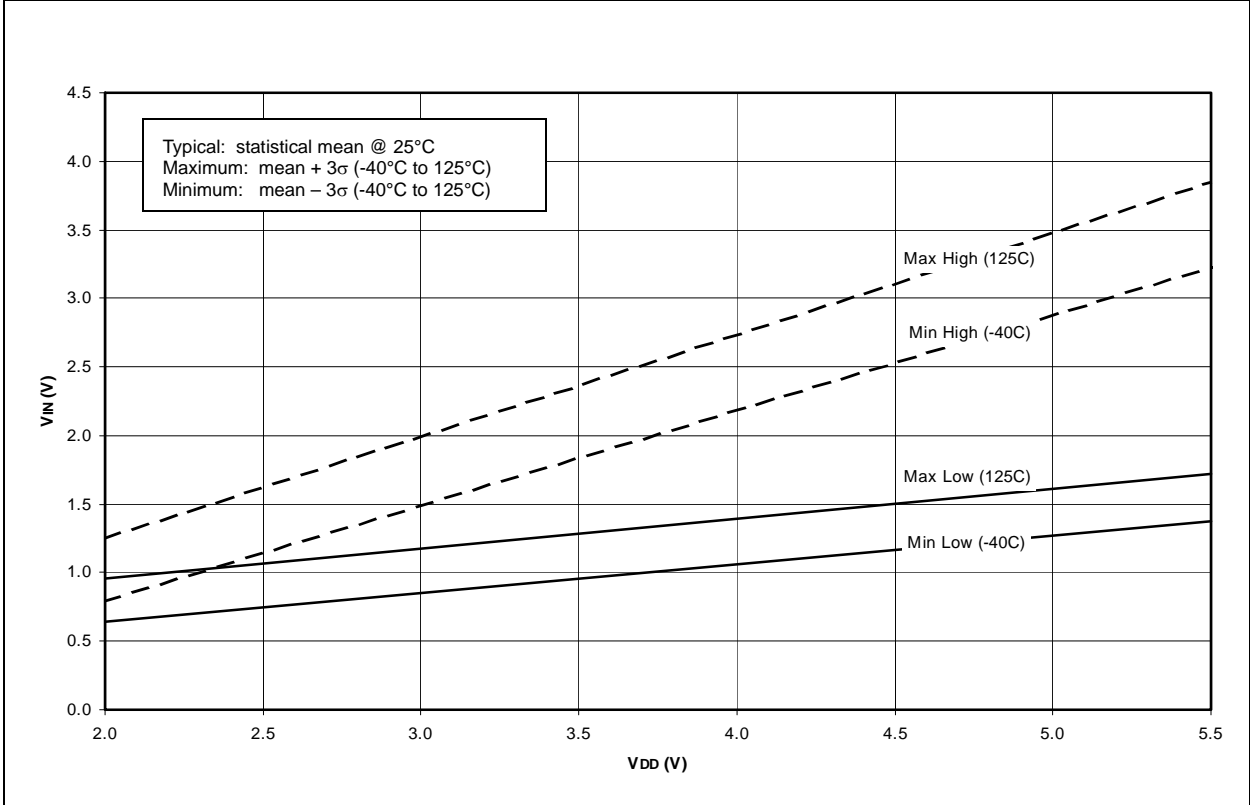
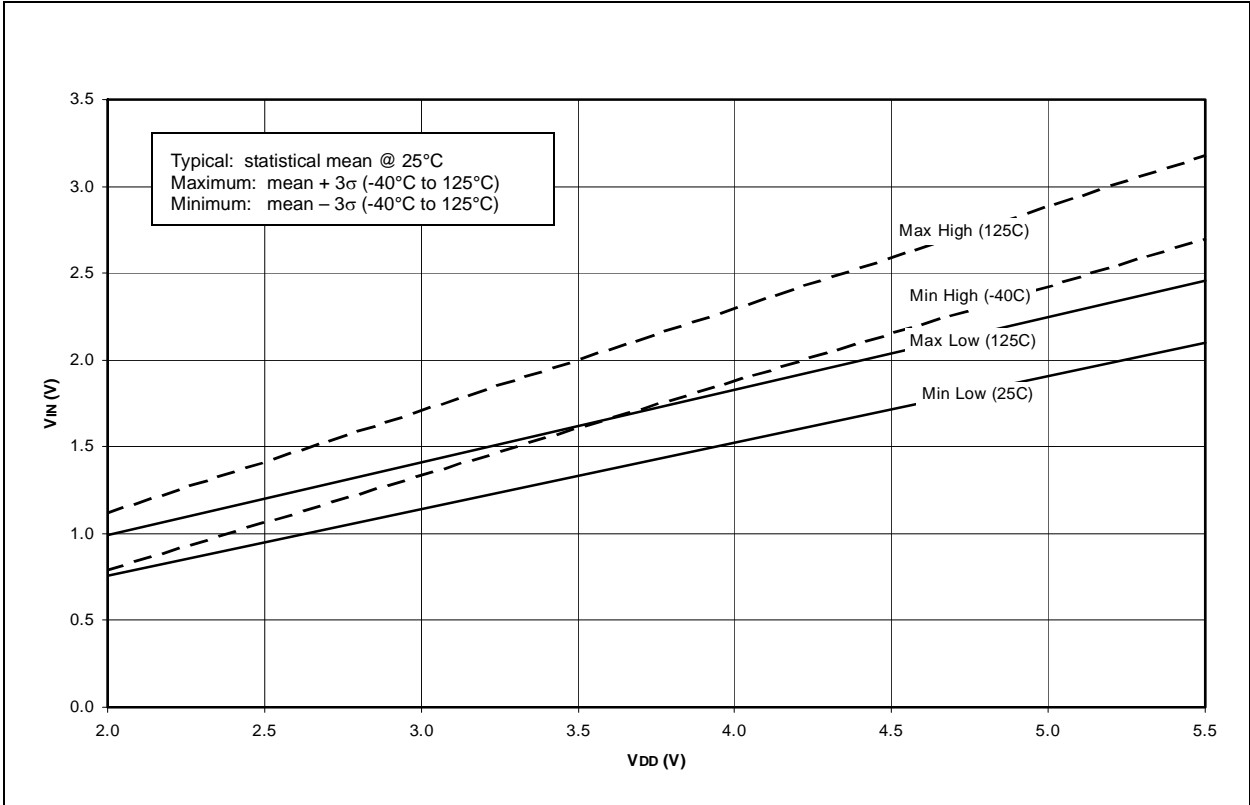


FIGURE 15-22: MINIMUM AND MAXIMUM V_{IN} vs. V_{DD} (I²C INPUT, -40°C TO 125°C)



PIC16F870/871

NOTES:

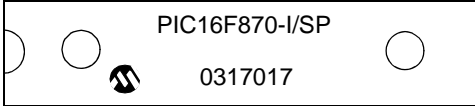
16.0 PACKAGING INFORMATION

16.1 Package Marking Information

28-Lead PDIP (Skinny DIP)



Example



28-Lead SOIC



Example



28-Lead SSOP



Example



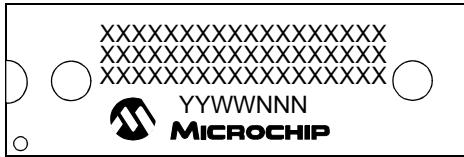
Legend: XX...X Customer-specific information
Y Year code (last digit of calendar year)
YY Year code (last 2 digits of calendar year)
WW Week code (week of January 1 is week '01')
NNN Alphanumeric traceability code
e3 Pb-free JEDEC designator for Matte Tin (Sn)
* This package is Pb-free. The Pb-free JEDEC 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.

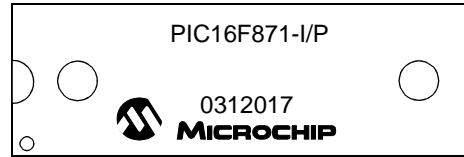
PIC16F870/871

Package Marking Information (Cont'd)

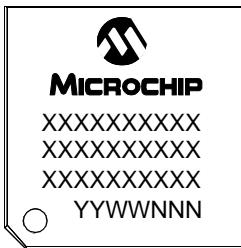
40-Lead PDIP



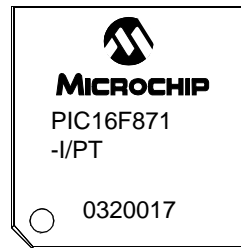
Example



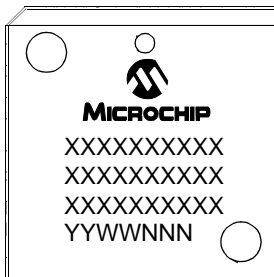
44-Lead TQFP



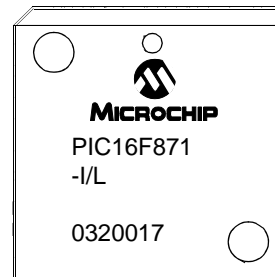
Example



44-Lead PLCC



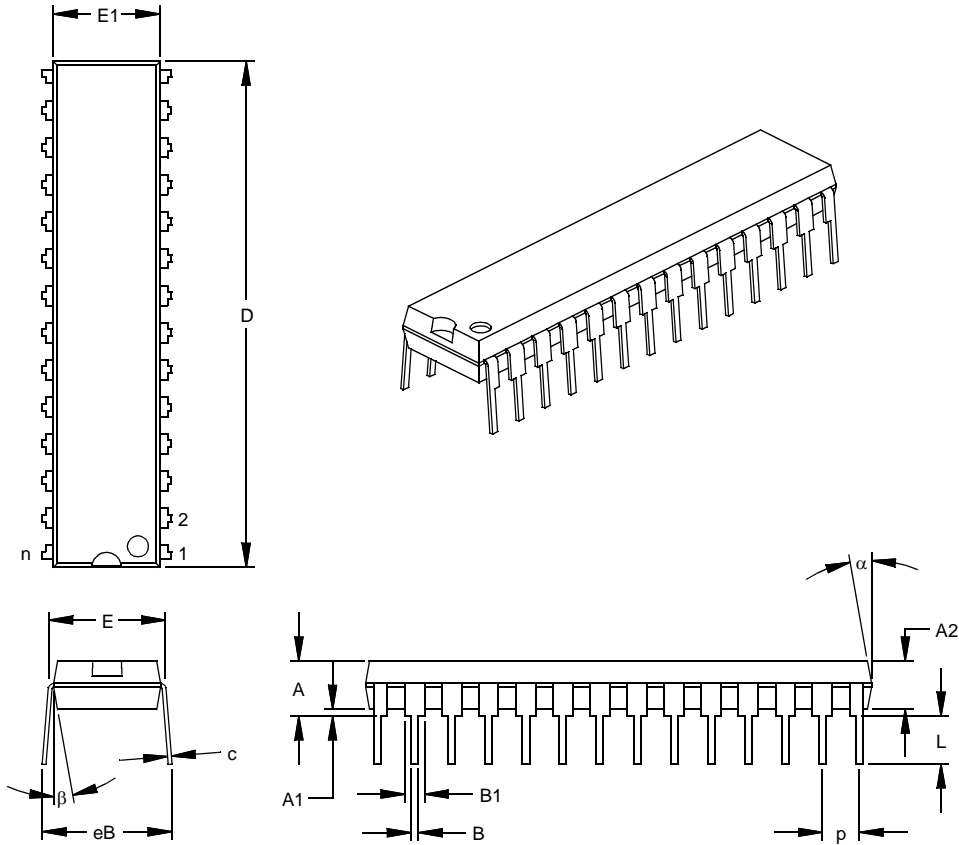
Example



PIC16F870/871

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.

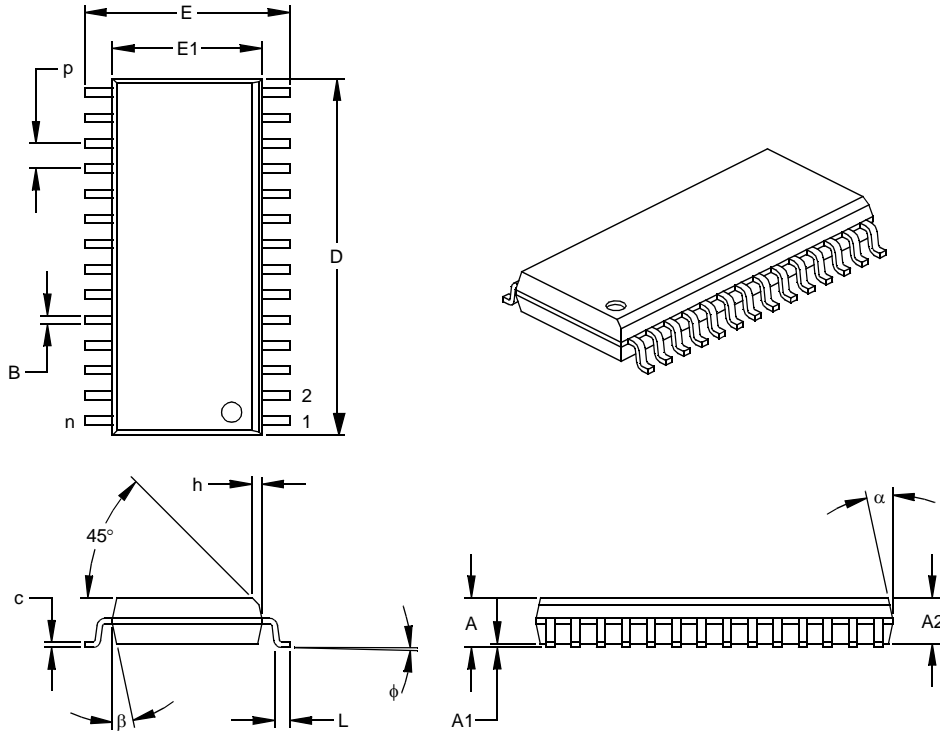
JEDEC Equivalent: MO-095

Drawing No. C04-070

PIC16F870/871

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>



| Dimension Limits | Units | INCHES* | | | MILLIMETERS | | |
|--------------------------|-------|---------|------|------|-------------|-------|-------|
| | | 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.

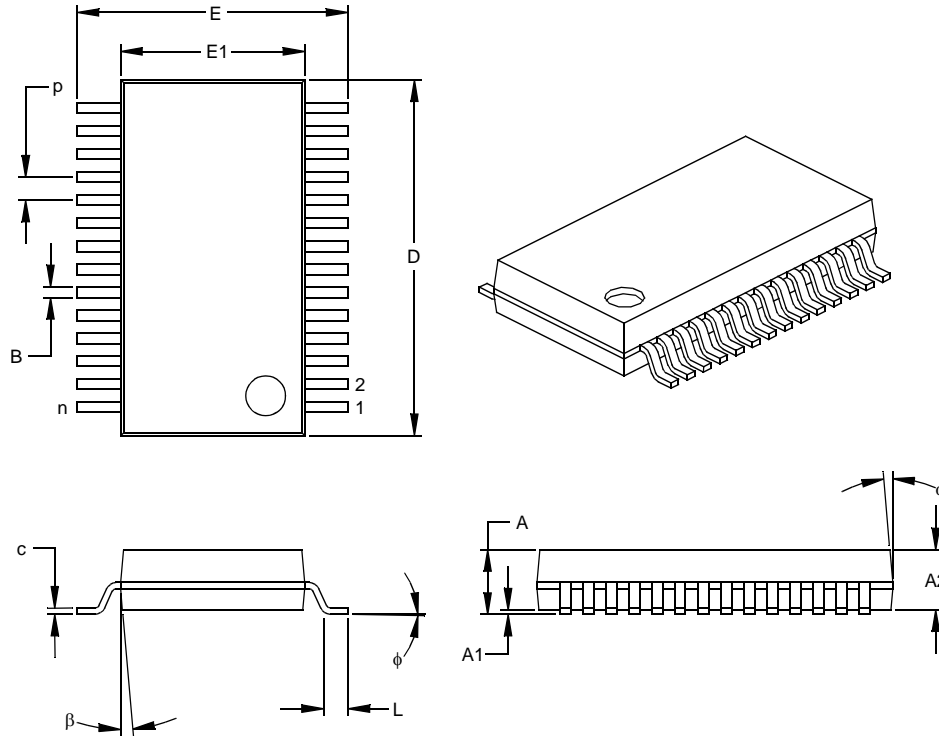
JEDEC Equivalent: MS-013

Drawing No. C04-052

PIC16F870/871

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>



| Dimension Limits | Units | INCHES | | | MILLIMETERS* | | |
|--------------------------|-------|--------|------|------|--------------|--------|--------|
| | | 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 |

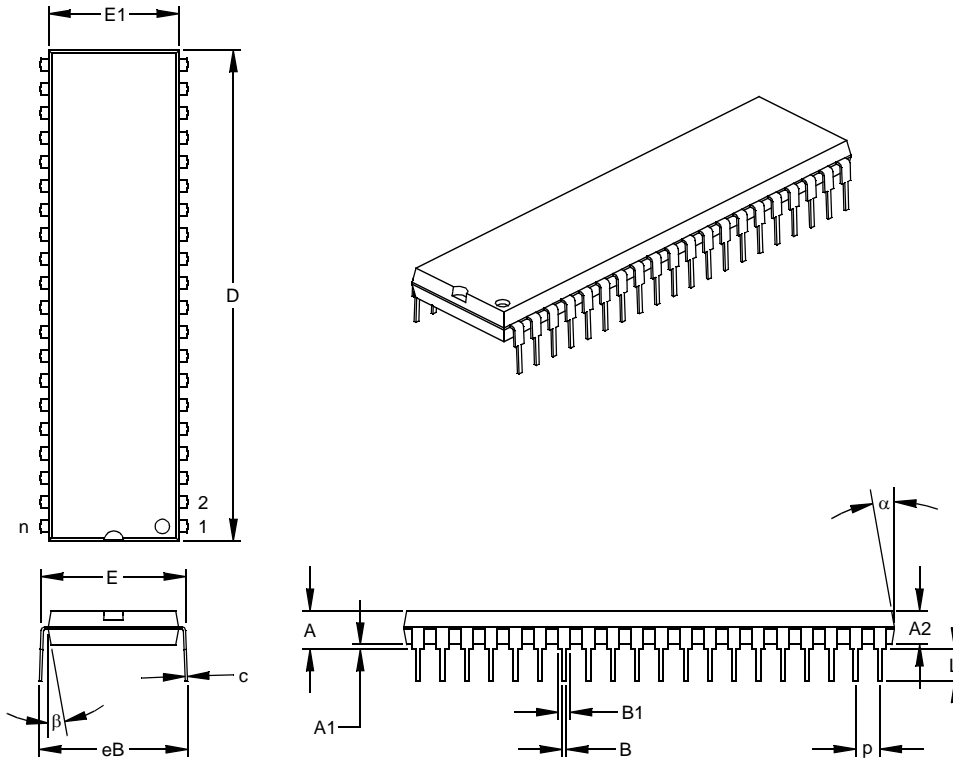
* Controlling Parameter
§ Significant Characteristic

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

PIC16F870/871

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>



| Dimension Limits | Units | INCHES* | | | MILLIMETERS | | |
|----------------------------|-------|---------|-------|-------|-------------|-------|-------|
| | | 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.

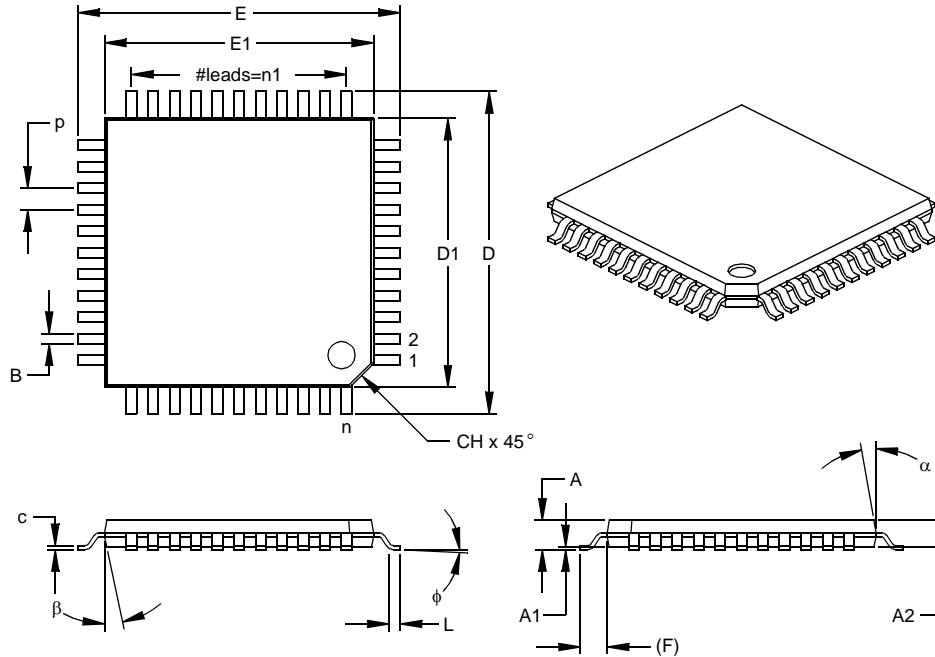
JEDEC Equivalent: MO-011

Drawing No. C04-016

PIC16F870/871

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/packageing>



| Dimension Limits | Units | INCHES | | | MILLIMETERS* | | |
|--------------------------|-------|--------|------|------|--------------|-------|-------|
| | | 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 |

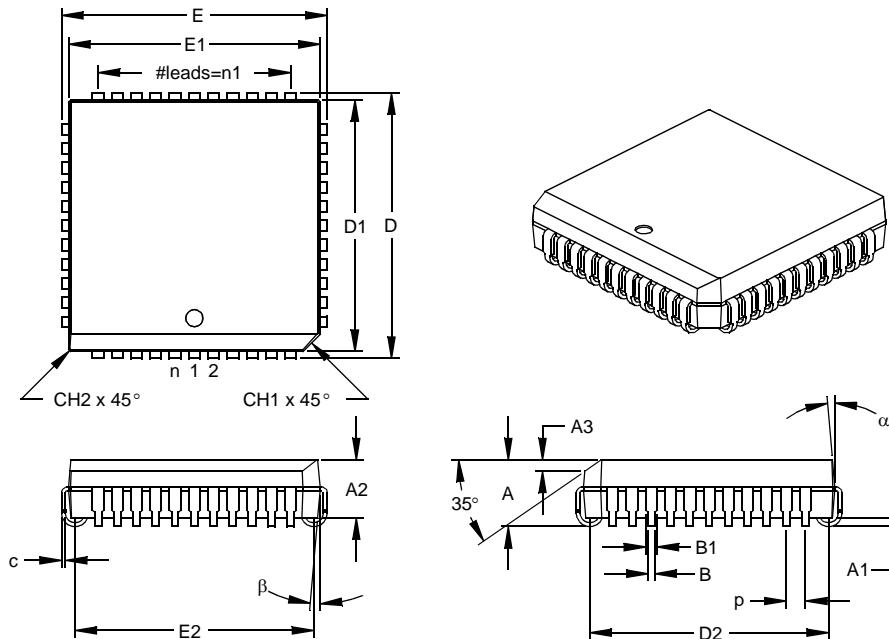
* Controlling Parameter
§ Significant Characteristic

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

PIC16F870/871

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>



| Dimension | Units | INCHES* | | | MILLIMETERS | | |
|--------------------------|-------|---------|------|------|-------------|-------|-------|
| | | 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

APPENDIX A: REVISION HISTORY

Revision A (December 1999)

Original data sheet for the PIC16F870/871 family.

Revision B (April 2003)

This revision includes the DC and AC Characteristics Graphs and Tables. The Electrical Specifications in Section 14.0 have been updated and there have been minor corrections to the data sheet text.

Revision C (January 2013)

Added a note to each package outline drawing.

APPENDIX B: DEVICE DIFFERENCES

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

TABLE B-1: DEVICE DIFFERENCES

| Feature | PIC16F870 | PIC16F871 |
|---------------------------------|------------------------|--------------------------------|
| On-chip Program Memory (Kbytes) | 2K | 2K |
| Data Memory (bytes) | 128 | 128 |
| Boot Block (bytes) | 2048 | 512 |
| Timer1 Low Power Option | Yes | No |
| I/O Ports | Ports A, B, C | Ports A, B, C, D, E |
| A/D Channels | 5 | 8 |
| External Memory Interface | No | No |
| Package Types | 28-pin DIP, SOIC, SSOP | 40-pin PDIP, 44-pin PLCC, TQFP |

PIC16F870/871

APPENDIX C: CONVERSION CONSIDERATIONS

This appendix discusses the considerations for converting from previous versions of a device to the ones listed in this data sheet. Typically, these changes are due to the differences in the process technology used. An example of this type of conversion is from a PIC17C756 to a PIC18F8720.

Not Applicable

APPENDIX D: MIGRATION FROM MID-RANGE TO ENHANCED DEVICES

A detailed discussion of the differences between the mid-range MCU devices (i.e., PIC16CXXX) and the enhanced devices (i.e., PIC18FXXX) is provided in AN716, "Migrating Designs from PIC16C74A/74B to PIC18C442." The changes discussed, while device specific, are generally applicable to all mid-range to enhanced device migrations.

This Application Note is available as Literature Number DS00716.

APPENDIX E: MIGRATION FROM HIGH-END TO ENHANCED DEVICES

A detailed discussion of the migration pathway and differences between the high-end MCU devices (i.e., PIC17CXXX) and the enhanced devices (i.e., PIC18FXXX) is provided in AN726, "PIC17CXXX to PIC18CXXX Migration." This Application Note is available as Literature Number DS00726.

PIC16F870/871

NOTES:

INDEX

A

| | |
|--|--------|
| A/D | 79 |
| Acquisition Requirements | 82 |
| ADCON0 Register | 79 |
| ADCON1 Register | 79 |
| ADIF Bit | 80 |
| ADRESH Register | 79 |
| ADRESL Register | 79 |
| Analog Port Pins | 41, 42 |
| Associated Registers and Bits | 85 |
| Calculating Acquisition Time | 82 |
| Configuring Analog Port Pins | 83 |
| Configuring the Interrupt | 81 |
| Configuring the Module | 81 |
| Conversion Clock | 83 |
| Conversions | 84 |
| Delays | 82 |
| Effects of a RESET | 85 |
| GO/DONE Bit | 80 |
| Internal Sampling Switch (Rss) Impedance | 82 |
| Operation During SLEEP | 85 |
| Result Registers | 84 |
| Source Impedance | 82 |
| Time Delays | 82 |
| Absolute Maximum Ratings | 117 |
| ADCON0 Register | 13 |
| ADCON1 Register | 14 |
| ADRESH Registers | 13 |
| ADRESL Register | 14 |
| Analog-to-Digital Converter. <i>See</i> A/D. | |
| Application Notes | |
| AN552 (Implementing Wake-up on Key Stroke) | 35 |
| AN556 (Implementing a Table Read) | 24 |
| Assembler | |
| MPASM Assembler | 111 |
| Asynchronous Reception | |
| Associated Registers | 69 |
| Asynchronous Reception (9-bit Mode) | |
| Associated Registers | 71 |

B

| | |
|---------------------------------------|----|
| Banking, Data Memory | 11 |
| Baud Rate Generator (BRG) | |
| Associated Registers | 63 |
| Block Diagrams | |
| A/D | 81 |
| Analog Input Model | 82 |
| Capture Mode Operation | 56 |
| Compare Mode Operation | 57 |
| Interrupt Logic | 97 |
| On-Chip RESET Circuit | 91 |
| PIC16F870 | 5 |
| PIC16F871 | 6 |
| PORTC (Peripheral Output Override) | 37 |
| PORTD (In I/O Port Mode) | 38 |
| PORTD and PORTE (Parallel Slave Port) | 42 |
| PORTE (In I/O Port Mode) | 39 |
| PWM Mode | 58 |
| RA3:RA0 and RA5 Pins | 33 |
| RA4/T0CKI Pin | 33 |
| RB3:RB0 Pins | 35 |
| RB7:RB4 Pins | 35 |
| Timer0/WDT Prescaler | 45 |

| | |
|---|----------------|
| Timer1 | 50 |
| Timer2 | 53 |
| USART Asynchronous Receive | 68 |
| USART Asynchronous Receive (9-bit Mode) | 70 |
| USART Transmit | 66 |
| Watchdog Timer | 99 |
| BOR. <i>See</i> Brown-out Reset. | |
| BRGH Bit | 63 |
| Brown-out Reset (BOR) | 87, 91, 92, 93 |
| BOR Status (BOR Bit) | 23 |

C

| | |
|--|---------|
| C Compilers | |
| MPLAB C17 | 112 |
| MPLAB C18 | 112 |
| MPLAB C30 | 112 |
| Capture/Compare/PWM (CCP) | 55 |
| Associated Registers | |
| Capture, Compare and Timer1 | 59 |
| PWM and Timer2 | 60 |
| Capture Mode | 56 |
| CCP1IF | 56 |
| Prescaler | 56 |
| CCP Timer Resources | 55 |
| Compare | |
| Special Trigger Output of CCP1 | 57 |
| Compare Mode | 57 |
| Software Interrupt Mode | 57 |
| Special Event Trigger | 57 |
| PWM Mode | 58 |
| Duty Cycle | 58 |
| Example Frequencies/Resolutions (table) | 59 |
| PWM Period | 58 |
| Setup for PWM Operation | 59 |
| Special Event Trigger and A/D Conversions | 57 |
| CCP. <i>See</i> Capture/Compare/PWM. | |
| CCP1CON Register | 13 |
| CCP1M0 Bit | 55 |
| CCP1M1 Bit | 55 |
| CCP1M2 Bit | 55 |
| CCP1M3 Bit | 55 |
| CCP1X Bit | 55 |
| CCP1Y Bit | 55 |
| CCPR1H Register | 13, 55 |
| CCPR1L Register | 13, 55 |
| Code Examples | |
| Changing Between Capture Prescalers | 56 |
| EEPROM Data Read | 29 |
| EEPROM Data Write | 29 |
| FLASH Program Read | 30 |
| FLASH Program Write | 31 |
| Indirect Addressing | 24 |
| Initializing PORTA | 33 |
| Saving STATUS, W and PCLATH Registers in RAM | 98 |
| Code Protected Operation | |
| Data EEPROM and FLASH Program Memory | 31 |
| Code Protection | 87, 101 |
| Computed GOTO | 24 |
| Configuration Bits | 87 |
| Configuration Word | 88 |
| Conversion Considerations | 158 |

PIC16F870/871

| | | | |
|---|------------|---|--------|
| D | | | |
| Data EEPROM | 27 | Indirect Addressing | 25 |
| Associated Registers | 32 | FSR Register | 11, 24 |
| Code Protection | 31 | INDF Register | 24 |
| Reading | 29 | Instruction Descriptions | 105 |
| Spurious Write Protection | 31 | Instruction Format | 103 |
| Write Verify | 31 | Instruction Set | 103 |
| Writing to | 29 | ADDLW | 105 |
| Data Memory | 11 | ADDWF | 105 |
| Bank Select (RP1:RP0 Bits) | 11 | ANDLW | 105 |
| General Purpose Registers | 11 | ANDWF | 105 |
| Register File Map | 12 | BCF | 105 |
| Special Function Registers | 13 | BSF | 105 |
| DC and AC Characteristics Graphs and Tables | 137 | BTFSC | 105 |
| DC Characteristics | | BTFSS | 105 |
| PIC16F870/871 (Extended) | 123 | CALL | 106 |
| PIC16F870/871 (Industrial) | 121 | CLRf | 106 |
| PIC16F870/871 (Industrial, Extended) and | | CLRw | 106 |
| PIC16LF870/871 (Commercial, Industrial) | 119 | CLRWDT | 106 |
| Demonstration Boards | | COMF | 106 |
| PICDEM 1 | 114 | DECF | 106 |
| PICDEM 17 | 114 | DECFSZ | 107 |
| PICDEM 18R PIC18C601/801 | 115 | GOTO | 107 |
| PICDEM 2 Plus | 114 | INCF | 107 |
| PICDEM 3 PIC16C92X | 114 | INCFSZ | 107 |
| PICDEM 4 | 114 | IORLW | 107 |
| PICDEM LIN PIC16C43X | 115 | IORWF | 107 |
| PICDEM USB PIC16C7X5 | 115 | MOVF | 108 |
| PICDEM.net Internet/Ethernet | 114 | MOVLW | 108 |
| Development Support | 111 | MOVWF | 108 |
| Device Differences | 157 | NOP | 108 |
| Device Overview | 5 | RETFIE | 108 |
| Direct Addressing | 25 | RETLW | 108 |
| E | | RETURN | 109 |
| EEADR Register | 15 | RLF | 109 |
| EEADRH Register | 15 | RRF | 109 |
| EECON1 Register | 15, 27 | SLEEP | 109 |
| EECON2 Register | 15, 27 | SUBLW | 109 |
| EEDATA Register | 15 | SUBWF | 109 |
| EEDATH Register | 15 | SWAPF | 110 |
| Electrical Characteristics | 117 | XORLW | 110 |
| Errata | 4 | XORWF | 110 |
| Evaluation and Programming Tools | 115 | Summary Table | 104 |
| F | | INT Interrupt (RB0/INT). See Interrupt Sources. | |
| Firmware Instructions | 103 | INTCON | 15 |
| FLASH Program Memory | 27 | INTCON Register | 13, 15 |
| Associated Registers | 32 | GIE Bit | 18 |
| Code Protection | 31 | INTE Bit | 18 |
| Configuration Bits and Read/Write State | 32 | INTF Bit | 18 |
| Reading | 30 | PEIE Bit | 18 |
| Spurious Write Protection | 31 | RBIE Bit | 18 |
| Write Protection | 32 | RBIF Bit | 18, 35 |
| Write Verify | 31 | TOIE Bit | 18 |
| Writing to | 30 | Internal Sampling Switch (R _{ss}) Impedance | 82 |
| FSR Register | 13, 14, 15 | Interrupt Sources | 87, 96 |
| I | | Interrupt-on-Change (RB7:RB4) | 35 |
| I/O Ports | 33 | RB0/INT Pin, External | 97 |
| ID Locations | 87, 101 | TMR0 Overflow | 97 |
| In-Circuit Debugger | 87, 101 | USART Receive/Transmit Complete | 61 |
| In-Circuit Serial Programming (ICSP) | 87, 102 | Interrupts, Context Saving During | 98 |
| INDF Register | 13, 14, 15 | | |

| | |
|---|------------|
| Interrupts, Enable Bits | |
| Global Interrupt Enable (GIE Bit) | 18, 96 |
| Interrupt-on-Change (RB7:RB4) Enable (RBIE Bit) | 18, 97 |
| Peripheral Interrupt Enable (PEIE Bit) | 18 |
| RB0/INT Enable (INTE Bit) | 18 |
| TMR0 Overflow Enable (TOIE Bit) | 18 |
| Interrupts, Flag Bits | |
| Interrupt-on-Change (RB7:RB4) Flag (RBIF Bit) | 18, 35, 97 |
| RB0/INT Flag (INTF Bit) | 18 |
| TMR0 Overflow Flag (TOIF Bit) | 97 |
| L | |
| Loading of PC | 24 |
| Low Voltage In-Circuit Serial Programming | 87, 102 |
| M | |
| Master Clear (<u>MCLR</u>) | |
| <u>MCLR</u> Reset, Normal Operation | 91, 93 |
| <u>MCLR</u> Reset, SLEEP | 91, 93 |
| <u>MCLR</u> /VPP/THV Pin | 7, 8 |
| Memory Organization | |
| Data Memory | 11 |
| Program Memory | 11 |
| Migration from High-End to Enhanced Devices | 159 |
| Migration from Mid-Range to Enhanced Devices | 158 |
| MPLAB ASM30 Assembler, Linker, Librarian | 112 |
| MPLAB ICD 2 In-Circuit Debugger | 113 |
| MPLAB ICE 2000 High Performance Universal In-Circuit Emulator | 113 |
| MPLAB ICE 4000 High Performance Universal In-Circuit Emulator | 113 |
| MPLAB Integrated Development Environment Software | 111 |
| MPLINK Object Linker/MPLIB Object Librarian | 112 |
| O | |
| OPCODE Field Descriptions | 103 |
| OPTION | 15 |
| OPTION Register | 15 |
| OPTION_REG Register | 14, 17 |
| INTEDG Bit | 17 |
| PSA Bit | 17 |
| RBPU Bit | 17 |
| T0CS Bit | 17 |
| T0SE Bit | 17 |
| OSC1/CLKI Pin | 7, 8 |
| OSC2/CLKO Pin | 7, 8 |
| Oscillator Configuration | 87 |
| HS | 89, 92 |
| LP | 89, 92 |
| RC | 89, 90, 92 |
| XT | 89, 92 |
| Oscillator, WDT | 99 |
| Oscillators | |
| Capacitor Selection | 90 |
| Crystal and Ceramic Resonators | 89 |
| RC | 90 |

| | |
|---|----------------|
| P | |
| Packaging | 149 |
| Marking Information | 149 |
| Parallel Slave Port (PSP) | 9, 38, 42 |
| Associated Registers | 43 |
| RE0/RD/AN5 Pin | 41, 42 |
| RE1/WR/AN6 Pin | 41, 42 |
| RE2/CS/AN7 Pin | 41, 42 |
| Select (PSPMODE Bit) | 38, 39, 40, 42 |
| PCL Register | 13, 14, 15, 24 |
| PCLATH Register | 13, 14, 15, 24 |
| PCON Register | 14, 15, 92 |
| BOR Bit | 23 |
| POR Bit | 23 |
| PICKIT 1 FLASH Starter Kit | 115 |
| PICSTART Plus Development Programmer | 113 |
| PIE1 Register | 14, 15 |
| PIE2 Register | 14, 15 |
| Pinout Descriptions | |
| PIC16F870 | 7 |
| PIC16F871 | 8 |
| PIR1 Register | 13 |
| PIR2 Register | 13 |
| POP | 24 |
| POR. See Power-on Reset. | |
| PORTA | 7, 8 |
| Associated Registers | 34 |
| PORTA Register | 33 |
| RA0/AN0 Pin | 7, 8 |
| RA1/AN1 Pin | 7, 8 |
| RA2/AN2/VREF- Pin | 7 |
| RA2/AN2/VREF- Pin | 8 |
| RA3/AN3/VREF+ Pin | 7 |
| RA3/AN3/VREF+ Pin | 8 |
| RA4/T0CKI Pin | 7, 8 |
| RA5/AN4 Pin | 7, 8 |
| TRISA Register | 33 |
| PORTA Register | 13 |
| PORTB | 7, 8 |
| PORTB Register | 35 |
| Pull-up Enable (RBPU Bit) | 17 |
| RB0/INT Edge Select (INTEDG Bit) | 17 |
| RB0/INT Pin | 7, 8 |
| RB0/INT Pin, External | 97 |
| RB1 Pin | 7, 8 |
| RB2 Pin | 7, 8 |
| RB3/PGM Pin | 7, 8 |
| RB4 Pin | 7, 8 |
| RB5 Pin | 7, 8 |
| RB6/PGC Pin | 7, 8 |
| RB7/PGD Pin | 7, 8 |
| RB7:RB4 Interrupt-on-Change | 97 |
| RB7:RB4 Interrupt-on-Change Enable (RBIE Bit) | 18, 97 |
| RB7:RB4 Interrupt-on-Change Flag (RBIF Bit) | 18, 35, 97 |
| TRISB Register | 35 |
| PORTB Register | 13, 15 |

PIC16F870/871

| | | | |
|---|----------------|---|-------------|
| PORTC..... | 7, 8 | Pulse Width Modulation. | |
| Associated Registers | 37 | See Capture/Compare/PWM, PWM Mode. | |
| PORTC Register | 37 | PUSH..... | 24 |
| RC0/T1OSO/T1CKI Pin | 7, 8 | R | |
| RC1/T1OSI Pin | 7, 8 | RAM. See Data Memory. | |
| RC2/CCP1 Pin | 7, 8 | RCREG Register | 13 |
| RC3 Pin..... | 7, 8 | RCSTA Register | 13 |
| RC4 Pin..... | 7, 8 | ADDEN Bit | 62 |
| RC5 Pin..... | 7, 8 | CREN Bit | 62 |
| RC6/TX/CK Pin | 7, 8, 62 | FERR Bit..... | 62 |
| RC7/RX/DT Pin..... | 7, 8, 62, 63 | OERR Bit | 62 |
| TRISC Register..... | 37, 61 | RX9 Bit | 62 |
| PORTC Register | 13 | RX9D Bit..... | 62 |
| PORTD..... | 9, 42 | SPEN Bit..... | 61, 62 |
| Associated Registers | 38 | SREN Bit | 62 |
| Parallel Slave Port (PSP) Function | 38 | Register File..... | 11 |
| PORTD Register | 38 | Register File Map..... | 12 |
| RD0/PSP0 Pin | 9 | Registers | |
| RD1/PSP1 Pin | 9 | ADCON0 (A/D Control 0)..... | 79 |
| RD2/PSP2 Pin | 9 | ADCON1 (A/D Control 1)..... | 80 |
| RD3/PSP3 Pin | 9 | CCP1CON (CCP Control 1)..... | 55 |
| RD4/PSP4 Pin | 9 | EECON1 (EEPROM Control 1) | 28 |
| RD5/PSP5 Pin | 9 | INTCON..... | 18 |
| RD6/PSP6 Pin | 9 | OPTION_REG | 17, 46 |
| RD7/PSP7 Pin | 9 | PCON (Power Control) | 23 |
| TRISD Register..... | 38 | PIE1 (Peripheral Interrupt Enable 1)..... | 19 |
| PORTD Register | 13 | PIE2 (Peripheral Interrupt Enable 2)..... | 21 |
| PORTE..... | 9 | PIR1 (Peripheral Interrupt Request Flag 1) | 20 |
| Analog Port Pins | 41, 42 | PIR2 (Peripheral Interrupt Request Flag 2) | 22 |
| Associated Registers | 41 | RCSTA (Receive Status and Control) | 62 |
| Input Buffer Full Status (IBF Bit) | 40 | STATUS | 16 |
| Input Buffer Overflow (IBOV Bit) | 40 | T1CON (Timer1 Control) | 49 |
| Output Buffer Full Status (OBF Bit)..... | 40 | T2CON (Timer 2 Control) | 53 |
| PORTE Register | 39 | TRISE | 40 |
| PSP Mode Select (PSPMODE Bit) | 38, 39, 40, 42 | TXSTA (Transmit Status and Control)..... | 61 |
| RE0/RD/AN5 Pin..... | 9, 41, 42 | Reset | 87, 91 |
| RE1/WR/AN6 Pin..... | 9, 41, 42 | MCLR Reset. See MCLR. | |
| RE2/CS/AN7 Pin..... | 9, 41, 42 | RESET Conditions for All Registers | 93 |
| TRISE Register | 39 | RESET Conditions for PCON Register..... | 93 |
| PORTE Register | 13 | RESET Conditions for Program Counter..... | 93 |
| Postscaler, WDT | | RESET Conditions for STATUS Register..... | 93 |
| Assignment (PSA Bit) | 17 | Reset | |
| Power-down Mode. See SLEEP. | | Brown-out Reset (BOR). See Brown-out Reset (BOR). | |
| Power-on Reset (POR) | 87, 91, 92, 93 | Power-on Reset (POR). See Power-on Reset (POR). | |
| Oscillator Start-up Timer (OST) | 87, 92 | WDT Reset. See Watchdog Timer (WDT). | |
| POR Status (POR Bit)..... | 23 | Revision History..... | 157 |
| Power Control (PCON) Register | 92 | S | |
| Power-down (PD Bit) | 91 | SCI. See USART | |
| Power-up Timer (PWRT) | 87, 92 | Serial Communication Interface. See USART. | |
| Time-out (TO Bit) | 91 | SLEEP | 87, 91, 100 |
| PR2 | 15 | Software Simulator (MPLAB SIM) | 112 |
| PR2 Register..... | 14, 53 | Software Simulator (MPLAB SIM30) | 112 |
| Prescaler, Timer0 | | SPBRG | 15 |
| Assignment (PSA Bit) | 17 | SPBRG Register..... | 14 |
| PRO MATE II Universal Device Programmer | 113 | Special Features of the CPU | 87 |
| Product Identification System..... | 169 | Special Function Registers | 13 |
| Program Counter | | Special Function Register Summary | 13 |
| RESET Conditions | 93 | Speed, Operating..... | 1 |
| Program Memory | 11 | SSPAD Register..... | 15 |
| Interrupt Vector | 11 | SSPSTAT Register | 15 |
| Map and Stack | 11 | Stack..... | 24 |
| Paging..... | 24 | Overflows..... | 24 |
| RESET Vector..... | 11 | Underflow | 24 |
| Program Verification..... | 101 | | |
| Programming, Device Instructions | 103 | | |

| | | | |
|---|--------|--|----------------|
| STATUS Register | 13, 15 | Asynchronous Reception with | |
| PD Bit | 91 | Address Detect | 71 |
| TO Bit | 91 | Brown-out Reset | 129 |
| Synchronous Master Reception | | Capture/Compare/PWM (CCP1) | 131 |
| Associated Registers | 75 | CLKO and I/O | 128 |
| Synchronous Master Transmission | | External Clock | 126 |
| Associated Registers | 73 | Parallel Slave Port (PSP) Read | 43 |
| Synchronous Slave Reception | | Parallel Slave Port (PSP) Write | 43 |
| Associated Registers | 77 | RESET, Watchdog Timer, Oscillator | |
| Synchronous Slave Transmission | | Start-up Timer and Power-up Timer | 129 |
| Associated Registers | 76 | Slow Rise Time (MCLR Tied to VDD) | 96 |
| T | | Time-out Sequence on Power-up | |
| T1CKPS0 Bit | 49 | (MCLR Not Tied to VDD) | |
| T1CKPS1 Bit | 49 | Case 1 | 95 |
| T1CON Register | 13 | Case 2 | 95 |
| T1OSCEN Bit | 49 | Time-out Sequence on Power-up | |
| T1SYNC Bit | 49 | (MCLR Tied to VDD) | 95 |
| T2CKPS0 Bit | 53 | Timer0 and Timer1 External Clock | 130 |
| T2CKPS1 Bit | 53 | USART Asynchronous Reception | 68 |
| T2CON Register | 13 | USART Synchronous Receive (Master/Slave) | 133 |
| TAD | 83 | USART Synchronous Reception | |
| Time-out Sequence | 92 | (Master Mode, SREN) | 75 |
| Timer0 | 45 | USART Synchronous Transmission | 73 |
| Associated Registers | 47 | USART Synchronous Transmission | |
| Clock Source Edge Select (T0SE Bit) | 17 | (Master/Slave) | 133 |
| Clock Source Select (T0CS Bit) | 17 | Wake-up from SLEEP via Interrupt | 101 |
| External Clock | 46 | Timing Parameter Symbology | 125 |
| Interrupt | 45 | TMR0 Register | 13, 15 |
| Overflow Enable (T0IE Bit) | 18 | TMR1CS Bit | 49 |
| Overflow Flag (T0IF Bit) | 97 | TMR1H Register | 13 |
| Overflow Interrupt | 97 | TMR1L Register | 13 |
| Prescaler | 46 | TMR1ON Bit | 49 |
| T0CKI | 46 | TMR2 Register | 13 |
| Timer1 | 49 | TMR2ON Bit | 53 |
| Associated Registers | 52 | TOUTPS0 Bit | 53 |
| Asynchronous Counter Mode | | TOUTPS1 Bit | 53 |
| Reading and Writing to | 51 | TOUTPS2 Bit | 53 |
| Counter Operation | 50 | TOUTPS3 Bit | 53 |
| Incrementing Edge (figure) | 50 | TRISA | 15 |
| Operation in Asynchronous Counter Mode | 51 | TRISA Register | 14 |
| Operation in Synchronized Counter Mode | 50 | TRISB | 15 |
| Operation in Timer Mode | 50 | TRISB Register | 14, 15 |
| Oscillator | 51 | TRISC | 15 |
| Capacitor Selection | 51 | TRISC Register | 14 |
| Prescaler | 52 | TRISD | 15 |
| Resetting of Timer1 Register Pair | | TRISD Register | 14 |
| (TMR1H, TMR1L) | 52 | TRISE | 15 |
| Resetting Timer1 Using a CCP Trigger Output | 51 | TRISE Register | 14 |
| TMR1H | 51 | IBF Bit | 40 |
| TMR1L | 51 | IBOV Bit | 40 |
| Timer2 | 53 | OBF Bit | 40 |
| Associated Registers | 54 | PSPMODE Bit | 38, 39, 40, 42 |
| Output | 54 | TXREG Register | 13 |
| Postscaler | 53 | TXSTA | 15 |
| Prescaler | 53 | TXSTA Register | 14 |
| Prescaler and Postscaler | 54 | BRGH Bit | 61 |
| Timing Diagrams | | CSRC Bit | 61 |
| A/D Conversion | 135 | TRMT Bit | 61 |
| Asynchronous Master Transmission | 67 | TX9 Bit | 61 |
| Asynchronous Master Transmission | | TX9D Bit | 61 |
| (Back to Back) | 67 | TXEN Bit | 61 |
| Asynchronous Reception with | | | |
| Address Byte First | 71 | | |

PIC16F870/871

U

| | |
|---|--------|
| Universal Synchronous Asynchronous Receiver Transmitter. See USART | |
| USART | 61 |
| Address Detect Enable (ADDEN Bit) | 62 |
| Asynchronous Mode | 66 |
| Asynchronous Receive | 68 |
| Asynchronous Receive (9-bit Mode) | 70 |
| Asynchronous Receive with Address Detect. See Asynchronous Receive (9-bit Mode). | |
| Asynchronous Reception | 69 |
| Asynchronous Transmitter | 66 |
| Baud Rate Generator (BRG) | 63 |
| Baud Rate Formula | 63 |
| Baud Rates, Asynchronous Mode (BRGH = 0) | 64 |
| Baud Rates, Asynchronous Mode (BRGH = 1) | 65 |
| High Baud Rate Select (BRGH Bit) | 61 |
| Sampling | 63 |
| Clock Source Select (CSRC Bit) | 61 |
| Continuous Receive Enable (CREN Bit) | 62 |
| Framing Error (FERR Bit) | 62 |
| Overrun Error (OERR Bit) | 62 |
| Receive Data, 9th bit (RX9D Bit) | 62 |
| Receive Enable, 9-bit (RX9 Bit) | 62 |
| Serial Port Enable (SPEN Bit) | 61, 62 |
| Single Receive Enable (SREN Bit) | 62 |
| Synchronous Master Mode | 72 |
| Synchronous Master Reception | 74 |
| Synchronous Master Transmission | 72 |
| Synchronous Slave Mode | 76 |
| Synchronous Slave Reception | 77 |
| Synchronous Slave Transmit | 76 |
| Transmit Data, 9th Bit (TX9D) | 61 |
| Transmit Enable (TXEN Bit) | 61 |
| Transmit Enable, Nine-bit (TX9 Bit) | 61 |
| Transmit Shift Register Status (TRMT Bit) | 61 |

W

| | |
|--------------------------------------|---------|
| Wake-up from SLEEP | 87, 100 |
| Interrupts | 93 |
| MCLR Reset | 93 |
| Timing Diagram | 101 |
| WDT Reset | 93 |
| Watchdog Timer | |
| Register Summary | 99 |
| Watchdog Timer (WDT) | 87, 99 |
| Enable (WDTEN Bit) | 99 |
| Postscaler. See Postscaler, WDT. | |
| Programming Considerations | 99 |
| RC Oscillator | 99 |
| Time-out Period | 99 |
| WDT Reset, Normal Operation | 91, 93 |
| WDT Reset, SLEEP | 91, 93 |
| Write Verify | |
| Data EEPROM and FLASH Program Memory | 31 |
| WWW, On-Line Support | 4 |

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>

PIC16F870/871

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: PIC16F870/871

Literature Number: DS30569C

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?

PIC16F870/871 PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

| <u>PART NO.</u> | <u>-X</u> | <u>/XX</u> | <u>XXX</u> | |
|--|-------------------|------------|------------|--|
| Device | Temperature Range | Package | Pattern | Examples: |
| <p>Device</p> <p>PIC16F870, PIC16F870T; VDD range 4.0V to 5.5V PIC16F871, PIC16F871T; VDD range 4.0V to 5.5V PIC16LF870X, PIC16LF870T; VDD range 2.0V to 5.5V PIC16LF871X, PIC16LF871T; VDD range 2.0V to 5.5V</p> <p>F = Normal VDD limits LP = Extended VDD limits T = In Tape and Reel - SOIC, SSOP, TQFP and PLCC packages only.</p> <p>Temperature Range</p> <p>blank(3) = 0°C to +70°C (Commercial) I = -40°C to +85°C (Industrial)</p> <p>Package</p> <p>PQ = MQFP (Metric PQFP) PT = TQFP (Thin Quad Flatpack) SO = SOIC SP = Skinny Plastic Dip SS = SSOP P = PDIP L = PLCC</p> <p>Pattern</p> <p>QTP, Code or Special Requirements (blank otherwise)</p> | | | | <p>a) PIC16F870-I/SP 301 = Industrial temp., PDIP package, 20 MHz, normal VDD limits, QTP pattern #301.</p> <p>b) PIC16F871-I/PT = Industrial temp., TQFP package, 20 MHz, Extended VDD limits.</p> <p>c) PIC16F871-I/P = Industrial temp., PDIP package, 20 MHz, normal VDD limits.</p> <p>d) PIC16LF870-I/SS = Industrial temp., SSOP package, DC - 20 MHz, extended VDD limits.</p> |

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)

NOTES:

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, PIC³² logo, rPIC, 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, Omniclient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rLAB, 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.

© 2000-2013, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

ISBN: 9781620769379

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.

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



MICROCHIP

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