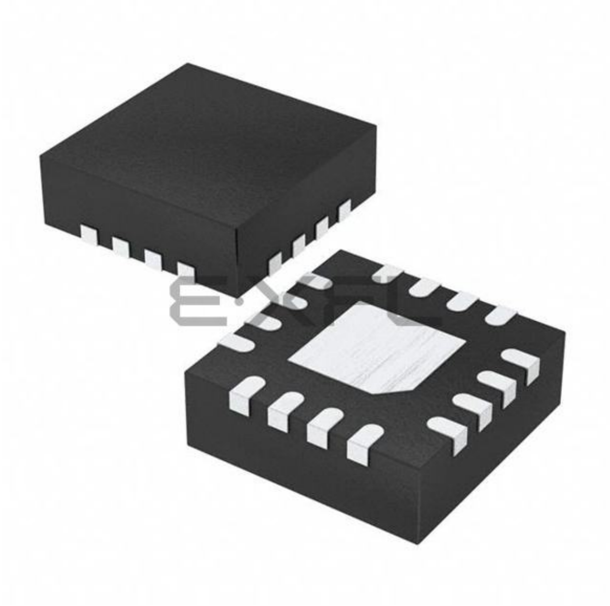
# Microchip Technology - PIC16F676T-I/ML Datasheet



| **Details** | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 20MHz |
| Connectivity | - |
| Peripherals | Brown-out Detect/Reset, POR, WDT |
| Number of I/O | 12 |
| Program Memory Size | 1.75KB (1K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 128 x 8 |
| RAM Size | 64 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 5.5V |
| Data Converters | A/D 8x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 16-VQFN Exposed Pad |
| Supplier Device Package | 16-QFN (4x4) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16f676t-i-ml |

**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in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates.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.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==

**Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PIC$^{32}$ logo, rfPIC 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, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Octopus, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rfLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA 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.

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

© 2010, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

♻ Printed on recycled paper.

ISBN: 978-1-60932-173-4

*Microchip received ISO/TS-16949:2002 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.*

### TABLE 2-1: PIC16F630/676 SPECIAL REGISTERS SUMMARY BANK 0

| Addr | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOD | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Bank 0 | | | | | | | | | | | |
| 00h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | `xxxx xxxx` | 20,63 |
| 01h | TMR0 | Timer0 Module's Register | | | | | | | | `xxxx xxxx` | 31 |
| 02h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | `0000 0000` | 19 |
| 03h | STATUS | IRP[(2)] | RP1[(2)] | RP0 | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | Z | DC | C | `0001 1xxx` | 13 |
| 04h | FSR | Indirect data memory Address Pointer | | | | | | | | `xxxx xxxx` | 20 |
| 05h | PORTA | — | — | I/O Control Registers | | | | | | `--xx xxxx` | 21 |
| 06h | — | Unimplemented | | | | | | | | — | — |
| 07h | PORTC | — | — | I/O Control Registers | | | | | | `--xx xxxx` | 28 |
| 08h | — | Unimplemented | | | | | | | | — | — |
| 09h | — | Unimplemented | | | | | | | | — | — |
| 0Ah | PCLATH | — | — | — | Write buffer for upper 5 bits of program counter | | | | | `---0 0000` | 19 |
| 0Bh | INTCON | GIE | PEIE | T0IE | INTE | RAIE | T0IF | INTF | RAIF | `0000 0000` | 15 |
| 0Ch | PIR1 | EEIF | ADIF | — | — | CMIF | — | — | TMR1IF | `00-- 0--0` | 17 |
| 0Dh | — | Unimplemented | | | | | | | | — | — |
| 0Eh | TMR1L | Holding register for the Least Significant Byte of the 16-bit TMR1 | | | | | | | | `xxxx xxxx` | 34 |
| 0Fh | TMR1H | Holding register for the Most Significant Byte of the 16-bit TMR1 | | | | | | | | `xxxx xxxx` | 34 |
| 10h | T1CON | — | T1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | TMR1CS | TMR1ON | `-000 0000` | 36 |
| 11h | — | Unimplemented | | | | | | | | — | — |
| 12h | — | Unimplemented | | | | | | | | — | — |
| 13h | — | Unimplemented | | | | | | | | — | — |
| 14h | — | Unimplemented | | | | | | | | — | — |
| 15h | — | Unimplemented | | | | | | | | — | — |
| 16h | — | Unimplemented | | | | | | | | — | — |
| 17h | — | Unimplemented | | | | | | | | — | — |
| 18h | — | Unimplemented | | | | | | | | — | — |
| 19h | CMCON | — | COUT | — | CINV | CIS | CM2 | CM1 | CM0 | `-0-0 0000` | 39 |
| 1Ah | — | Unimplemented | | | | | | | | — | — |
| 1Bh | — | Unimplemented | | | | | | | | — | — |
| 1Ch | — | Unimplemented | | | | | | | | — | — |
| 1Dh | — | Unimplemented | | | | | | | | — | — |
| 1Eh | ADRESH[(3)] | Most Significant 8 bits of the left shifted A/D result or 2 bits of right shifted result | | | | | | | | `xxxx xxxx` | 46 |
| 1Fh | ADCON0[(3)] | ADFM | VCFG | — | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON | `00-0 0000` | 47,63 |

**Legend:** – = Unimplemented locations read as '`0`', `u` = unchanged, `x` = unknown, `q` = value depends on condition shaded = unimplemented

**Note 1:** Other (non Power-up) Resets include $\overline{\text{MCLR}}$ Reset, Brown-out Detect and Watchdog Timer Reset during normal operation.

**2:** IRP and RP1 bits are reserved, always maintain these bits clear.

**3:** PIC16F676 only.

## 2.4 Indirect Addressing, INDF and FSR Registers

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

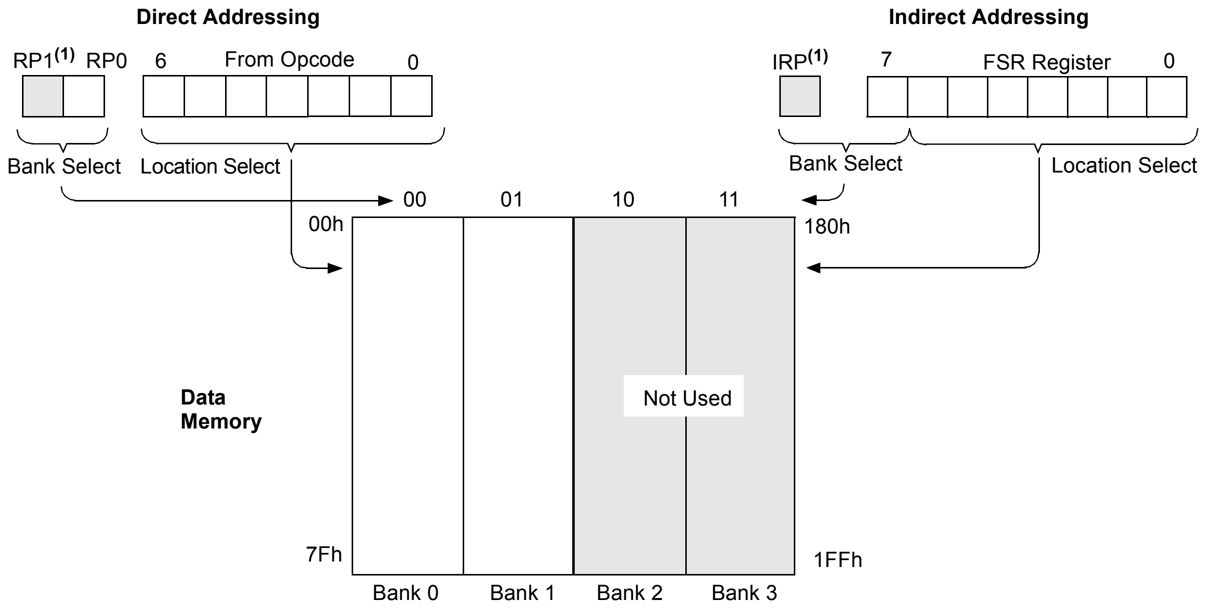
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status 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 location 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    ;inc pointer
         BTFSS   FSR,4  ;all done?
         GOTO    NEXT   ;no clear next
CONTINUE                ;yes continue
```

**FIGURE 2-4: DIRECT/INDIRECT ADDRESSING PIC16F630/676**



For memory map detail see Figure 2-2.

**Note 1:** The RP1 and IRP bits are reserved; always maintain these bits clear.

### TABLE 3-1: 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, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 05h | PORTA | — | — | RA5 | RA4 | RA3 | RA2 | RA1 | RA0 | --xx xxxx | --uu uuuu |
| 0Bh/8Bh | INTCON | GIE | PEIE | T0IE | INTE | RAIE | T0IF | INTF | RAIF | 0000 0000 | 0000 000u |
| 19h | CMCON | — | COUT | — | CINV | CIS | CM2 | CM1 | CM0 | -0-0 0000 | -0-0 0000 |
| 81h | OPTION_REG | $\overline{\text{RAPU}}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 1111 1111 |
| 85h | TRISA | — | — | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | --11 1111 | --11 1111 |
| 91h | ANSEL(1) | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | 1111 1111 | 1111 1111 |
| 95h | WPUA | — | — | WPUA5 | WPUA4 | — | WPUA2 | WPUA1 | WPUA0 | --11 -111 | --11 -111 |
| 96h | IOCA | — | — | IOCA5 | IOCA4 | IOCA3 | IOCA2 | IOCA1 | IOCA0 | --00 0000 | --00 0000 |

**Note 1:** PIC16F676 only.

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

### REGISTER 3-5: PORTC — PORTC REGISTER (ADDRESS: 07h)

| U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| — | — | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 |
| bit 7 | | | | | | | bit 0 |

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

bit 5-0: **PORTC<5:0>**: General Purpose I/O pin bits
1 = Port pin is >VIH
0 = Port pin is <VIL

**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 |

### REGISTER 3-6: TRISC — PORTC TRI-STATE REGISTER (ADDRESS: 87h)

| U-0 | U-0 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | — | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 |
| bit 7 | | | | | | | bit 0 |

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

bit 5-0: **TRISC<5:0>**: PORTC Tri-State Control bits
1 = PORTC pin configured as an input (tri-stated)
0 = PORTC pin configured as an 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 3-2: 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, BOD | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 07h | PORTC | — | — | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | `--xx xxxx` | `--uu uuuu` |
| 87h | TRISC | — | — | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | `--11 1111` | `--11 1111` |
| 91h | ANSEL[(1)] | ANS7 | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 | `1111 1111` | `1111 1111` |

**Note 1:** PIC16F676 only.

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

## 5.1 Timer1 Modes of Operation

Timer1 can operate in one of three modes:

- 16-bit timer with prescaler
- 16-bit synchronous counter
- 16-bit asynchronous counter

In Timer mode, Timer1 is incremented on every instruction cycle. In Counter mode, Timer1 is incremented on the rising edge of the external clock input T1CKI. In addition, the Counter mode clock can be synchronized to the microcontroller system clock or run asynchronously.

In counter and timer modules, the counter/timer clock can be gated by the $\overline{\text{T1G}}$ input.

If an external clock oscillator is needed (and the microcontroller is using the INTOSC w/o CLKOUT), Timer1 can use the LP oscillator as a clock source.

| **Note:** | In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge. |
|---|---|

## 5.2 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit (PIR1<0>) is set. To enable the interrupt on rollover, you must set these bits:

- Timer1 interrupt Enable bit (PIE1<0>)
- PEIE bit (INTCON<6>)
- GIE bit (INTCON<7>).

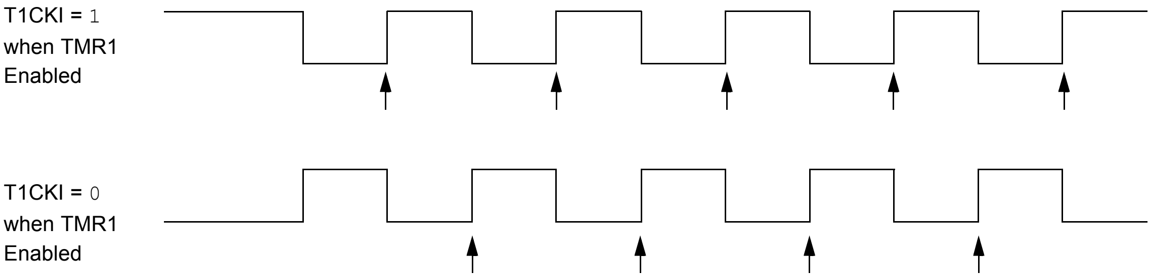
The interrupt is cleared by clearing the TMR1IF in the Interrupt Service Routine.

| **Note:** | The TMR1H:TTMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts. |
|---|---|

## 5.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4, or 8 divisions of the clock input. The T1CKPS bits (T1CON<5:4>) control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

**FIGURE 5-2: TIMER1 INCREMENTING EDGE**



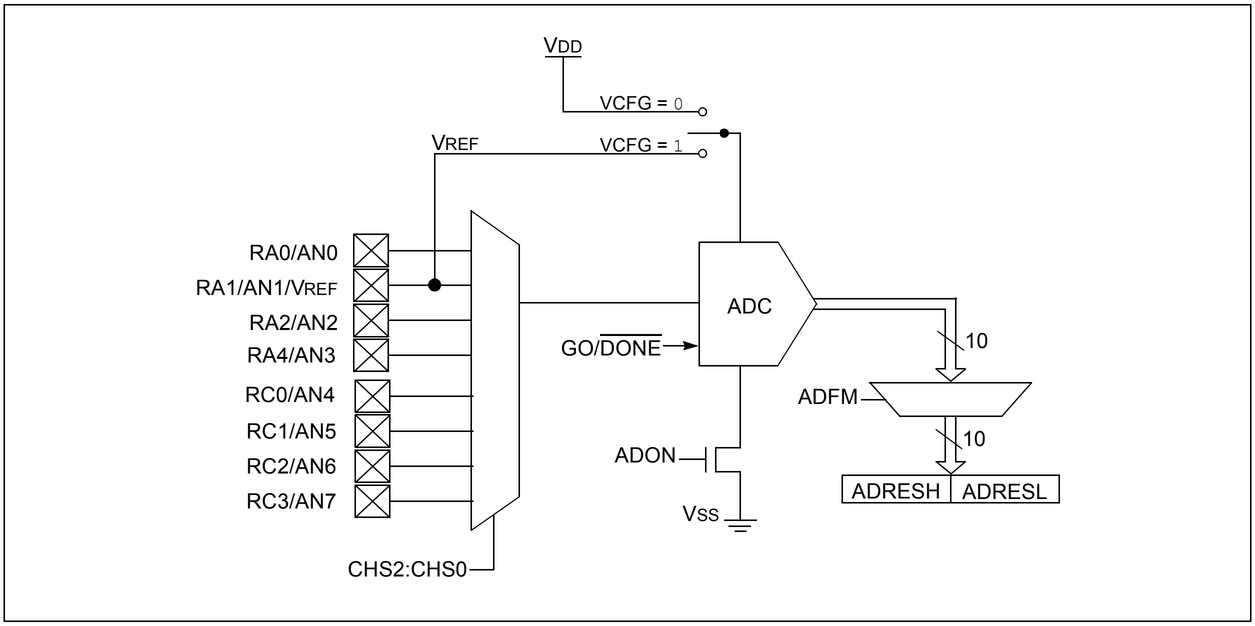
**Note 1:** Arrows indicate counter increments.

**2:** In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge of the clock.

## 7.0 ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE (PIC16F676 ONLY)

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. The PIC16F676 has eight analog inputs, multiplexed into one sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a binary result via successive approximation and stores the result in a 10-bit register. The voltage reference used in the conversion is software selectable to either VDD or a voltage applied by the VREF pin. Figure 7-1 shows the block diagram of the A/D on the PIC16F676.

**FIGURE 7-1: A/D BLOCK DIAGRAM**



## 7.1 A/D Configuration and Operation

There are three registers available to control the functionality of the A/D module:

1. ADCON0 (Register 7-1)
2. ADCON1 (Register 7-2)
3. ANSEL (Register 7-3)

### 7.1.1 ANALOG PORT PINS

The ANS7:ANS0 bits (ANSEL<7:0>) and the TRISA bits control the operation of the A/D port pins. Set the corresponding TRISA bits to set the pin output driver to its high-impedance state. Likewise, set the corresponding ANS bit to disable the digital input buffer.

| **Note:** | Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current. |
|---|---|

### 7.1.2 CHANNEL SELECTION

There are eight analog channels on the PIC16F676, AN0 through AN7. The CHS2:CHS0 bits (ADCON0<4:2>) control which channel is connected to the sample and hold circuit.

### 7.1.3 VOLTAGE REFERENCE

There are two options for the voltage reference to the A/D converter: either VDD is used, or an analog voltage applied to VREF is used. The VCFG bit (ADCON0<6>) controls the voltage reference selection. If VCFG is set, then the voltage on the VREF pin is the reference; otherwise, VDD is the reference.

### 7.1.4 CONVERSION CLOCK

The A/D conversion cycle requires 11 TAD. The source of the conversion clock is software selectable via the ADCS bits (ADCON1<6:4>). There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC (dedicated internal oscillator)

For correct conversion, the A/D conversion clock (1/TAD) must be selected to ensure a minimum TAD of 1.6 μs. Table 7-1 shows a few TAD calculations for selected frequencies.

## REGISTER 7-1: ADCON0 — A/D CONTROL REGISTER (ADDRESS: 1Fh)

| R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ADFM | VCFG | — | CHS2 | CHS1 | CHS0 | GO/$\overline{\text{DONE}}$ | ADON |

bit 7 bit 0

bit 7 **ADFM:** A/D Result Formed Select bit
1 = Right justified
0 = Left justified

bit 6 **VCFG:** Voltage Reference bit
1 = VREF pin
0 = VDD

bit 5 **Unimplemented:** Read as zero

bit 4-2 **CHS2:CHS0:** Analog Channel Select bits
000 = Channel 00 (AN0)
001 = Channel 01 (AN1)
010 = Channel 02 (AN2)
011 = Channel 03 (AN3)
100 = Channel 04 (AN4)
101 = Channel 05 (AN5)
110 = Channel 06 (AN6)
111 = Channel 07 (AN7)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle. This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 **ADON:** A/D Conversion Status bit
1 = A/D converter module is operating
0 = A/D converter is shut-off and consumes no operating current

| **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 |

## REGISTER 7-2: ADCON1 — A/D CONTROL REGISTER 1 (ADRESS: 9Fh)

| U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | ADCS2 | ADCS1 | ADCS0 | — | — | — | — |

bit 7 bit 0

bit 7: **Unimplemented:** Read as ‘0’

bit 6-4: **ADCS<2:0>:** A/D Conversion Clock Select bits
000 = FOSC/2
001 = FOSC/8
010 = FOSC/32
x11 = FRC (clock derived from a dedicated internal oscillator = 500 kHz max)
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64

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 |

## 7.2 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor ($C_{HOLD}$) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 7-3. The source impedance ($R_S$) and the internal sampling switch ($R_{SS}$) impedance directly affect the time required to charge the capacitor $C_{HOLD}$. The sampling switch ($R_{SS}$) impedance varies over the device voltage ($V_{DD}$), see Figure 7-3. **The maximum recommended impedance for analog sources is 10 kΩ.** 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 7-1 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 Reference Manual (DS33023).

### EQUATION 7-1: ACQUISITION TIME

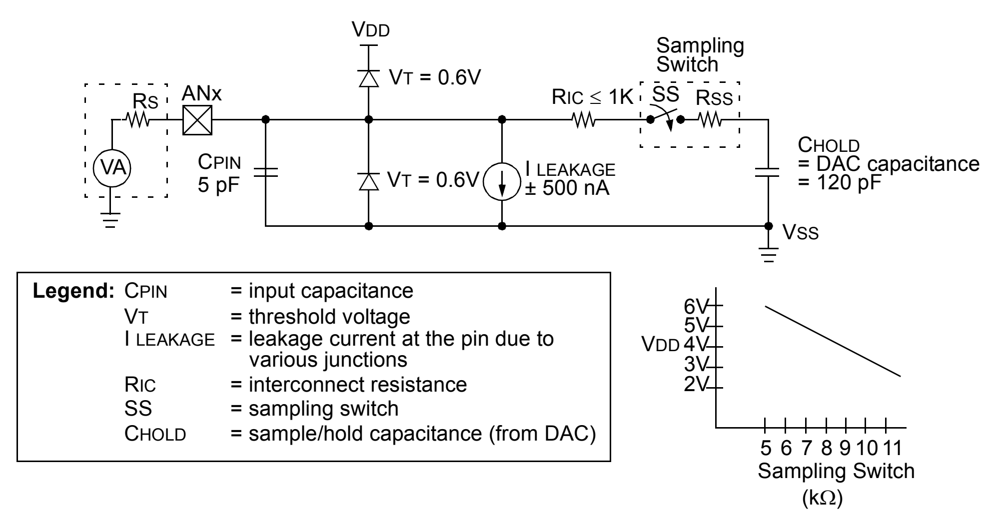
$$
\begin{aligned}
T_{ACQ} &= \text{Amplifier Settling Time} + \\
&\quad\ \text{Hold Capacitor Charging Time} + \\
&\quad\ \text{Temperature Coefficient} \\
&= T_{AMP} + T_C + T_{COFF} \\
&= 2\mu s + T_C + [(\text{Temperature} - 25^\circ C)(0.05\mu s/^\circ C)] \\
T_C &= C_{HOLD}\,(R_{IC} + R_{SS} +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

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

**2:** The charge holding capacitor ($C_{HOLD}$) is not discharged after each conversion.

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

### FIGURE 7-3: ANALOG INPUT MODEL



**Legend:** $C_{PIN}$ = input capacitance
$V_T$ = threshold voltage
$I_{LEAKAGE}$ = leakage current at the pin due to various junctions
$R_{IC}$ = interconnect resistance
SS = sampling switch
$C_{HOLD}$ = sample/hold capacitance (from DAC)

## 9.2 Oscillator Configurations

### 9.2.1 OSCILLATOR TYPES

The PIC16F630/676 can be operated in eight different Oscillator Option modes. The user can program three Configuration bits (FOSC2 through FOSC0) to select one of these eight modes:

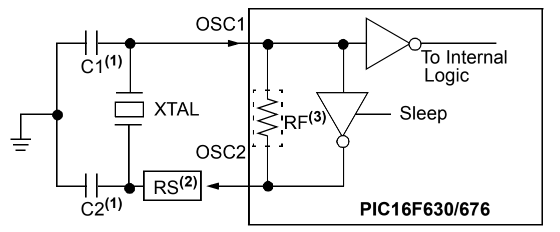
- LP Low-Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC External Resistor/Capacitor (2 modes)
- INTOSC Internal Oscillator (2 modes)
- EC External Clock In

| **Note:** | Additional information on oscillator configurations is available in the PIC® Mid-Range Reference Manual, (DS33023). |
|---|---|

### 9.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

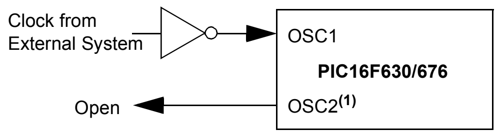
In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (see Figure 9-1). The PIC16F630/676 oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may yield a frequency outside of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1 pin (see Figure 9-2).

**FIGURE 9-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)**



**Note 1:** See Table 9-1 and Table 9-2 for recommended values of C1 and C2.
**2:** A series resistor may be required for AT strip cut crystals.
**3:** RF varies with the Oscillator mode selected (Approx. value = 10 MΩ).

**FIGURE 9-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT, EC, OR LP OSC CONFIGURATION)**



**Note 1:** Functions as RA4 in EC Osc mode.

**TABLE 9-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS**

| Ranges Characterized: | | | |
|---|---|---|---|
| **Mode** | **Freq** | **OSC1(C1)** | **OSC2(C2)** |
| XT | 455 kHz<br>2.0 MHz<br>4.0 MHz | 68-100 pF<br>15-68 pF<br>15-68 pF | 68-100 pF<br>15-68 pF<br>15-68 pF |
| HS | 8.0 MHz<br>16.0 MHz | 10-68 pF<br>10-22 pF | 10-68 pF<br>10-22 pF |

**Note 1:** Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

**TABLE 9-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR**

| Mode | Freq | OSC1(C1) | OSC2(C2) |
|---|---|---|---|
| LP | 32 kHz | 68-100 pF | 68-100 pF |
| XT | 100 kHz<br>2 MHz<br>4 MHz | 68-150 pF<br>15-30 pF<br>15-30 pF | 150-200 pF<br>15-30 pF<br>15-30 pF |
| HS | 8 MHz<br>10 MHz<br>20 MHz | 15-30 pF<br>15-30 pF<br>15-30 pF | 15-30 pF<br>15-30 pF<br>15-30 pF |

**Note 1:** Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

### 9.2.3 EXTERNAL CLOCK IN

For applications where a clock is already available elsewhere, users may directly drive the PIC16F630/676 provided that this external clock source meets the AC/DC timing requirements listed in **Section 12.0 “Electrical Specifications”**. Figure 9-2 shows how an external clock circuit should be configured.

### 9.2.4 RC OSCILLATOR

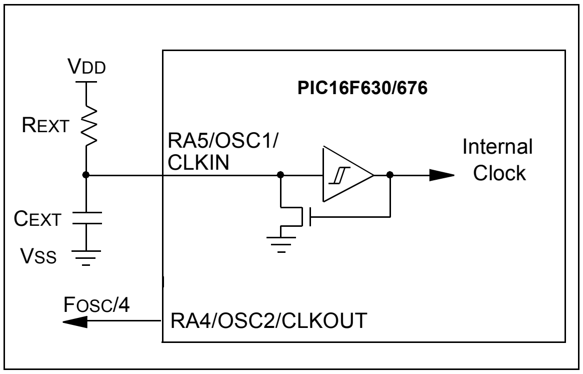
For applications where precise timing is not a requirement, the RC oscillator option is available. The operation and functionality of the RC oscillator is dependent upon a number of variables. The RC oscillator frequency is a function of:

- Supply voltage
- Resistor (REXT) and capacitor (CEXT) values
- Operating temperature

The oscillator frequency will vary from unit to unit due to normal process parameter variation. The difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low CEXT values. The user also needs to account for the tolerance of the external R and C components. Figure 9-3 shows how the R/C combination is connected.

Two options are available for this Oscillator mode which allow RA4 to be used as a general purpose I/O or to output FOSC/4.

**FIGURE 9-3: RC OSCILLATOR MODE**



### 9.2.5 INTERNAL 4 MHZ OSCILLATOR

When calibrated, the internal oscillator provides a fixed 4 MHz (nominal) system clock. See Electrical Specifications, **Section 12.0 “Electrical Specifications”**, for information on variation over voltage and temperature.

Two options are available for this Oscillator mode which allow RA4 to be used as a general purpose I/O or to output FOSC/4.

#### 9.2.5.1 Calibrating the Internal Oscillator

A calibration instruction is programmed into the last location of program memory. This instruction is a `RETLW XX`, where the literal is the calibration value. The literal is placed in the OSCCAL register to set the calibration of the internal oscillator. Example 9-1 demonstrates how to calibrate the internal oscillator. For best operation, decouple (with capacitance) VDD and VSS as close to the device as possible.

**Note:** Erasing the device will also erase the pre-programmed internal calibration value for the internal oscillator. The calibration value must be saved prior to erasing part as specified in the PIC16F630/676 Programming specification. Microchip Development Tools maintain all calibration bits to factory settings.

**EXAMPLE 9-1: CALIBRATING THE INTERNAL OSCILLATOR**

```
BSF    STATUS, RP0    ;Bank 1
CALL   3FFh           ;Get the cal value
MOVWF  OSCCAL         ;Calibrate
BCF    STATUS, RP0    ;Bank 0
```

### 9.2.6 CLKOUT

The PIC16F630/676 devices can be configured to provide a clock out signal in the INTOSC and RC Oscillator modes. When configured, the oscillator frequency divided by four (FOSC/4) is output on the RA4/OSC2/CLKOUT pin. FOSC/4 can be used for test purposes or to synchronize other logic.

## 9.7 Power-Down Mode (Sleep)

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

If the Watchdog Timer is enabled:

- WDT will be cleared but keeps running
- $\overline{\text{PD}}$ bit in the STATUS register is cleared
- $\overline{\text{TO}}$ bit is set
- Oscillator driver is turned off
- I/O ports maintain the status they had before `SLEEP` was executed (driving high, low, or high-impedance).

For lowest current consumption in this mode, all I/O pins should be either at VDD, or VSS, with no external circuitry drawing current from the I/O pin and the comparators and CVREF should be disabled. I/O pins that are high-impedance inputs should be pulled 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 PORTA should be considered.

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

**Note:** It should be noted that a Reset generated by a WDT time-out does not drive $\overline{\text{MCLR}}$ pin low.

### 9.7.1 WAKE-UP FROM SLEEP

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

1. External Reset input on $\overline{\text{MCLR}}$ pin
2. Watchdog Timer Wake-up (if WDT was enabled)
3. Interrupt from RA2/INT pin, PORTA change, or a peripheral interrupt.

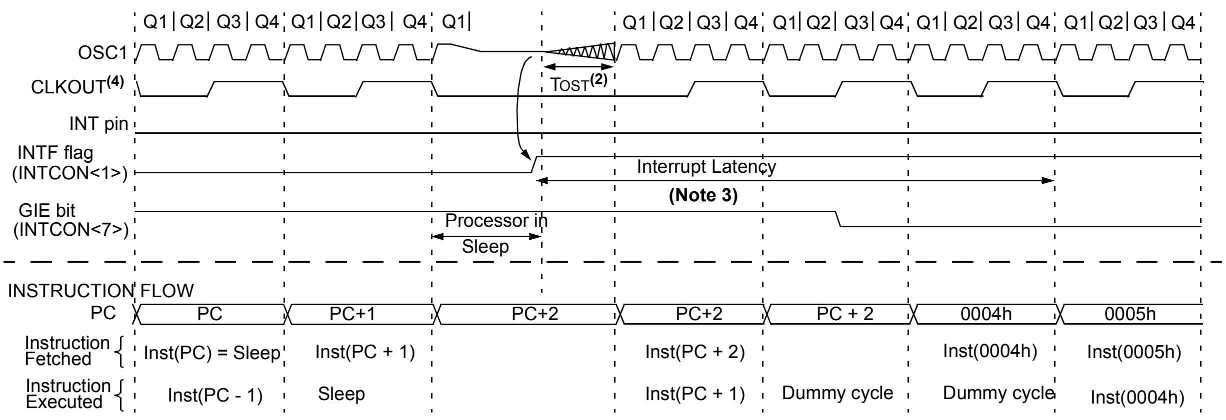
The first event will cause a device Reset. The two latter events are considered a continuation of program execution. The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits in the STATUS register can be used to determine the cause of device Reset. The $\overline{\text{PD}}$ bit, which is set on power-up, is cleared when Sleep is invoked. $\overline{\text{TO}}$ bit is cleared if WDT Wake-up occurred.

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, then branches to the interrupt address (0004h). In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have an `NOP` after the `SLEEP` instruction.

**Note:** If the global interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from Sleep. The `SLEEP` instruction is completely executed.

The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

**FIGURE 9-13: WAKE-UP FROM SLEEP THROUGH INTERRUPT**



**Note 1:** XT, HS or LP Oscillator mode assumed.
**2:** TOST = 1024TOSC (drawing not to scale). Approximately 1 μs delay for RC Oscillator mode. See Section 12 for wake-up from Sleep delay in INTOSC mode.
**3:** GIE = `1` assumed. In this case after wake-up, the processor jumps to the interrupt routine. If GIE = `0`, execution will continue in-line.
**4:** CLKOUT is not available in XT, HS, LP or EC Osc modes, but shown here for timing reference.

## 9.8 Code Protection

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

**Note:** The entire data EEPROM and Flash program memory will be erased when the code protection is turned off. The INTOSC calibration data is also erased. See PIC16F630/676 Programming Specification for more information.

## 9.9 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. Only the Least Significant 7 bits of the ID locations are used.

## 9.10 In-Circuit Serial Programming

The PIC16F630/676 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
- programming voltage

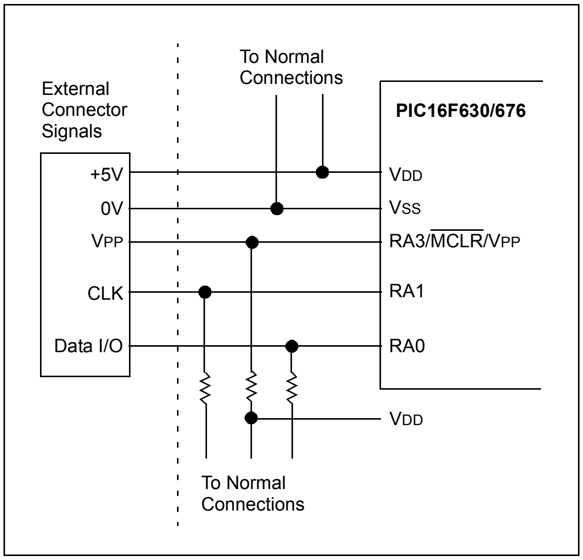
This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the RA0 and RA1 pins low, while raising the $\overline{\text{MCLR}}$ (VPP) pin from VIL to VIHH (see Programming Specification). RA0 becomes the programming data and RA1 becomes the programming clock. Both RA0 and RA1 are Schmitt Trigger inputs in this mode.

After Reset, to place the device into Programming/Verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14 bits of program data are then supplied to or from the device, depending on whether the command was a load or a read. For complete details of serial programming, please refer to the PIC16F630/676 Programming Specification.

A typical In-Circuit Serial Programming connection is shown in Figure 9-14.

**FIGURE 9-14: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**



## 9.11 In-Circuit Debugger

Since in-circuit debugging requires the loss of clock, data and $\overline{\text{MCLR}}$ pins, MPLAB® ICD 2 development with an 14-pin device is not practical. A special 20-pin PIC16F676-ICD device is used with MPLAB ICD 2 to provide separate clock, data and $\overline{\text{MCLR}}$ pins and frees all normally available pins to the user.

This special ICD device is mounted on the top of the header and its signals are routed to the MPLAB ICD 2 connector. On the bottom of the header is an 14-pin socket that plugs into the user's target via the 14-pin stand-off connector.

When the $\overline{\text{ICD}}$ pin on the PIC16F676-ICD device is held low, the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB ICD 2. When the microcontroller has this feature enabled, some of the resources are not available for general use. Table 9-10 shows which features are consumed by the background debugger:

**TABLE 9-10: DEBUGGER RESOURCES**

| | |
|---|---|
| I/O pins | ICDCLK, ICDDATA |
| Stack | 1 level |
| Program Memory | Address 0h must be `NOP`<br>300h-3FEh |

For more information, see 14-Pin MPLAB ICD 2 Header Information Sheet (DS51292) available on Microchip's web site (www.microchip.com).

## 10.2 Instruction Descriptions

| **ADDLW** | **Add Literal and W** |
|---|---|
| Syntax: | [*label*] ADDLW k |
| Operands: | $0 \leq k \leq 255$ |
| Operation: | (W) + k → (W) |
| Status Affected: | C, DC, Z |
| Description: | The contents of the W register are added to the eight-bit literal ‘k’ and the result is placed in the W register. |

| **ADDWF** | **Add W and f** |
|---|---|
| Syntax: | [*label*] ADDWF f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| Operation: | (W) + (f) → (destination) |
| Status Affected: | C, DC, Z |
| Description: | Add the contents of the W register with register ‘f’. If ‘d’ is `0`, the result is stored in the W register. If ‘d’ is `1`, the result is stored back in register ‘f’. |

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

| **ANDWF** | **AND W with f** |
|---|---|
| Syntax: | [*label*] ANDWF f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| Operation: | (W) .AND. (f) → (destination) |
| Status Affected: | Z |
| Description: | AND the W register with register ‘f’. If ‘d’ is `0`, the result is stored in the W register. If ‘d’ is `1`, the result is stored back in register ‘f’. |

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

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

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

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

## 12.2 DC Characteristics: PIC16F630/676-I (Industrial)

| | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Device Characteristics** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** | |
| | | | | | | **VDD** | **Note** |
| D010 | **Supply Current (IDD)** | — | 9 | 16 | μA | 2.0 | FOSC = 32 kHz<br>LP Oscillator Mode |
| | | — | 18 | 28 | μA | 3.0 | |
| | | — | 35 | 54 | μA | 5.0 | |
| D011 | | — | 110 | 150 | μA | 2.0 | FOSC = 1 MHz<br>XT Oscillator Mode |
| | | — | 190 | 280 | μA | 3.0 | |
| | | — | 330 | 450 | μA | 5.0 | |
| D012 | | — | 220 | 280 | μA | 2.0 | FOSC = 4 MHz<br>XT Oscillator Mode |
| | | — | 370 | 650 | μA | 3.0 | |
| | | — | 0.6 | 1.4 | mA | 5.0 | |
| D013 | | — | 70 | 110 | μA | 2.0 | FOSC = 1 MHz<br>EC Oscillator Mode |
| | | — | 140 | 250 | μA | 3.0 | |
| | | — | 260 | 390 | μA | 5.0 | |
| D014 | | — | 180 | 250 | μA | 2.0 | FOSC = 4 MHz<br>EC Oscillator Mode |
| | | — | 320 | 470 | μA | 3.0 | |
| | | — | 580 | 850 | μA | 5.0 | |
| D015 | | — | 340 | 450 | μA | 2.0 | FOSC = 4 MHz<br>INTOSC Mode |
| | | — | 500 | 780 | μA | 3.0 | |
| | | — | 0.8 | 1.1 | mA | 5.0 | |
| D016 | | — | 180 | 250 | μA | 2.0 | FOSC = 4 MHz<br>EXTRC Mode |
| | | — | 320 | 450 | μA | 3.0 | |
| | | — | 580 | 800 | μA | 5.0 | |
| D017 | | — | 2.1 | 2.95 | mA | 4.5 | FOSC = 20 MHz<br>HS Oscillator Mode |
| | | — | 2.4 | 3.0 | mA | 5.0 | |

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The test conditions for all IDD measurements in Active Operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

**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.

## 12.4 DC Characteristics: PIC16F630/676-E (Extended)

| Param No. | Device Characteristics | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +125°C for extended | | | | | |
|---|---|---|---|---|---|---|---|
| | | Min | Typ† | Max | Units | Conditions | |
| | | | | | | VDD | Note |
| D010E | **Supply Current (IDD)** | — | 9 | 16 | μA | 2.0 | FOSC = 32 kHz<br>LP Oscillator Mode |
| | | — | 18 | 28 | μA | 3.0 | |
| | | — | 35 | 54 | μA | 5.0 | |
| D011E | | — | 110 | 150 | μA | 2.0 | FOSC = 1 MHz<br>XT Oscillator Mode |
| | | — | 190 | 280 | μA | 3.0 | |
| | | — | 330 | 450 | μA | 5.0 | |
| D012E | | — | 220 | 280 | μA | 2.0 | FOSC = 4 MHz<br>XT Oscillator Mode |
| | | — | 370 | 650 | μA | 3.0 | |
| | | — | 0.6 | 1.4 | mA | 5.0 | |
| D013E | | — | 70 | 110 | μA | 2.0 | FOSC = 1 MHz<br>EC Oscillator Mode |
| | | — | 140 | 250 | μA | 3.0 | |
| | | — | 260 | 390 | μA | 5.0 | |
| D014E | | — | 180 | 250 | μA | 2.0 | FOSC = 4 MHz<br>EC Oscillator Mode |
| | | — | 320 | 470 | μA | 3.0 | |
| | | — | 580 | 850 | μA | 5.0 | |
| D015E | | — | 340 | 450 | μA | 2.0 | FOSC = 4 MHz<br>INTOSC Mode |
| | | — | 500 | 780 | μA | 3.0 | |
| | | — | 0.8 | 1.1 | mA | 5.0 | |
| D016E | | — | 180 | 250 | μA | 2.0 | FOSC = 4 MHz<br>EXTRC Mode |
| | | — | 320 | 450 | μA | 3.0 | |
| | | — | 580 | 800 | μA | 5.0 | |
| D017E | | — | 2.1 | 2.95 | mA | 4.5 | FOSC = 20 MHz<br>HS Oscillator Mode |
| | | — | 2.4 | 3.0 | mA | 5.0 | |

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The test conditions for all IDD measurements in Active Operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{\text{MCLR}}$ = VDD; WDT disabled.

**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.

## 12.6 DC Characteristics: PIC16F630/676-I (Industrial), PIC16F630/676-E (Extended)

| **DC CHARACTERISTICS** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial<br>-40°C ≤ TA ≤ +125°C for extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| | | **Input Low Voltage** | | | | | |
| | VIL | I/O ports | | | | | |
| D030 | | with TTL buffer | VSS | — | 0.8 | V | 4.5V ≤ VDD ≤ 5.5V |
| D030A | | | VSS | — | 0.15 VDD | V | Otherwise |
| D031 | | with Schmitt Trigger buffer | VSS | — | 0.2 VDD | V | Entire range |
| D032 | | $\overline{\text{MCLR}}$, OSC1 (RC mode) | VSS | — | 0.2 VDD | V | |
| D033 | | OSC1 (XT and LP modes) | VSS | — | 0.3 | V | **(Note 1)** |
| D033A | | OSC1 (HS mode) | VSS | — | 0.3 VDD | V | **(Note 1)** |
| | | **Input High Voltage** | | | | | |
| | VIH | I/O ports | | — | | | |
| D040 | | with TTL buffer | 2.0 | — | VDD | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | (0.25 VDD+0.8) | — | VDD | V | otherwise |
| D041 | | with Schmitt Trigger buffer | 0.8 VDD | — | VDD | | entire range |
| D042 | | $\overline{\text{MCLR}}$ | 0.8 VDD | — | VDD | V | |
| D043 | | OSC1 (XT and LP modes) | 1.6 | — | VDD | V | **(Note 1)** |
| D043A | | OSC1 (HS mode) | 0.7 VDD | — | VDD | V | **(Note 1)** |
| D043B | | OSC1 (RC mode) | 0.9 VDD | — | VDD | V | |
| D070 | IPUR | **PORTA Weak Pull-up Current** | 50* | 250 | 400* | μA | VDD = 5.0V, VPIN = VSS |
| | | **Input Leakage Current**$^{(3)}$ | | | | | |
| D060 | IIL | I/O ports | — | ± 0.1 | ± 1 | μA | VSS ≤ VPIN ≤ VDD,<br>Pin at high-impedance |
| D060A | | Analog inputs | — | ± 0.1 | ± 1 | μA | VSS ≤ VPIN ≤ VDD |
| D060B | | VREF | — | ± 0.1 | ± 1 | μA | VSS ≤ VPIN ≤ VDD |
| D061 | | $\overline{\text{MCLR}}^{(2)}$ | — | ± 0.1 | ± 5 | μA | VSS ≤ VPIN ≤ VDD |
| D063 | | OSC1 | — | ± 0.1 | ± 5 | μA | VSS ≤ VPIN ≤ VDD, XT, HS and LP osc configuration |
| | | **Output Low Voltage** | | | | | |
| D080 | VOL | I/O ports | — | — | 0.6 | V | IOL = 8.5 mA, VDD = 4.5V (Ind.) |
| D083 | | OSC2/CLKOUT (RC mode) | — | — | 0.6 | V | IOL = 1.6 mA, VDD = 4.5V (Ind.)<br>IOL = 1.2 mA, VDD = 4.5V (Ext.) |
| | | **Output High Voltage** | | | | | |
| D090 | VOH | I/O ports | VDD - 0.7 | — | — | V | IOH = -3.0 mA, VDD = 4.5V (Ind.) |
| D092 | | OSC2/CLKOUT (RC mode) | VDD - 0.7 | — | — | V | IOH = -1.3 mA, VDD = 4.5V (Ind.)<br>IOH = -1.0 mA, VDD = 4.5V (Ext.) |

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended to use an 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.

## 12.8 TIMING PARAMETER SYMBOLOGY

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

1. TppS2ppS
2. TppS

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

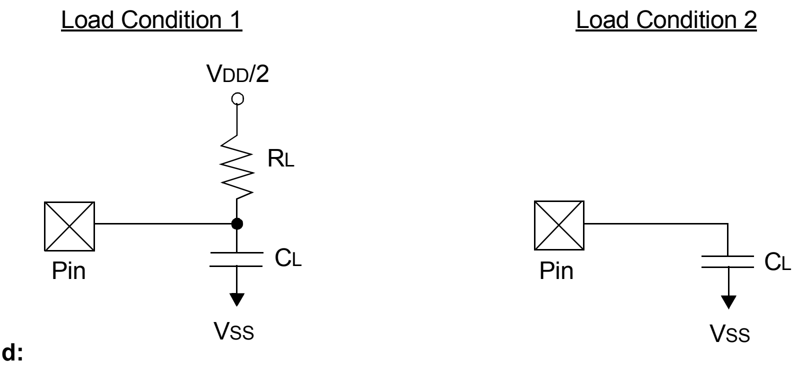
Lowercase letters (pp) and their meanings:

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

Uppercase letters and their meanings:

| **S** | | | |
|---|---|---|---|
| F | Fall | P | Period |
| H | High | R | Rise |
| I | Invalid (High-impedance) | V | Valid |
| L | Low | Z | High-impedance |

**FIGURE 12-4: LOAD CONDITIONS**



**Legend:**

$R_L$ = 464Ω

$C_L$ = 50 pF for all pins

15 pF for OSC2 output

**FIGURE 13-3: TYPICAL IPD vs. VDD OVER TEMP (+125°C)**

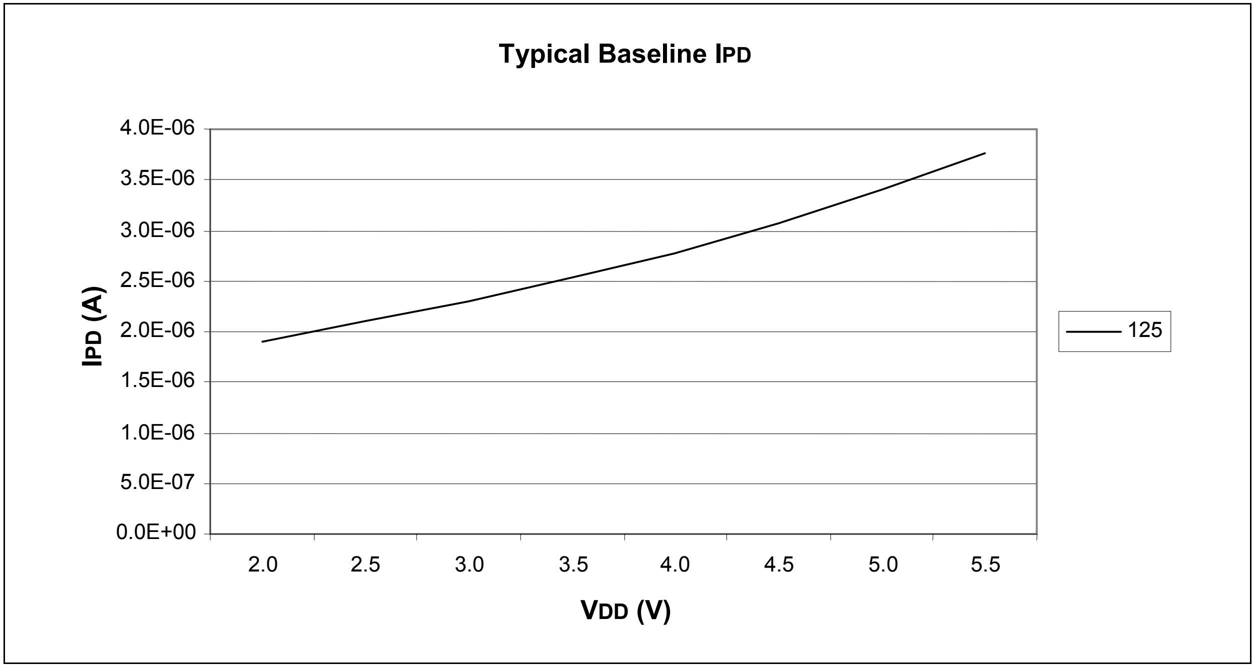
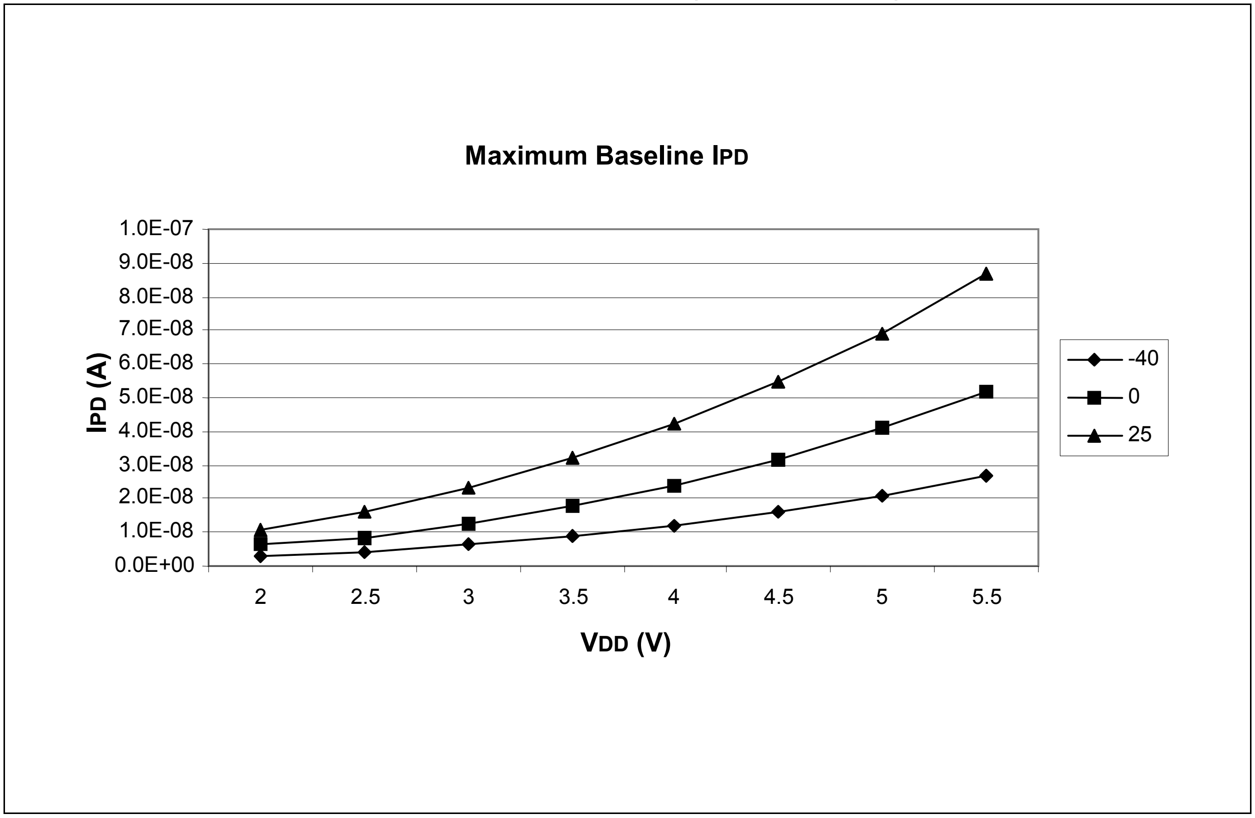


**FIGURE 13-4: MAXIMUM IPD vs. VDD OVER TEMP (-40°C TO +25°C)**



## APPENDIX C: DEVICE MIGRATIONS

This section is intended to describe the functional and electrical specification differences when migrating between functionally similar devices (such as from a PIC16C74A to a PIC16C74B).

**Not Applicable**

## APPENDIX D: MIGRATING FROM OTHER PIC® DEVICES

This discusses some of the issues in migrating from other PIC devices to the PIC16F6XX family of devices.

### D.1 PIC12C67X to PIC12F6XX

**TABLE 1: FEATURE COMPARISON**

| Feature | PIC12C67X | PIC16F6XX |
|---|---|---|
| Max Operating Speed | 10 MHz | 20 MHz |
| Max Program Memory | 2048 bytes | 1024 bytes |
| A/D Resolution | 8-bit | 10-bit |
| Data EEPROM | 16 bytes | 64 bytes |
| Oscillator Modes | 5 | 8 |
| Brown-out Detect | N | Y |
| Internal Pull-ups | RA0/1/3 | RA0/1/2/4/5 |
| Interrupt-on-change | RA0/1/3 | RA0/1/2/3/4/5 |
| Comparator | N | Y |

**Note:** This device has been designed to perform to the parameters of its data sheet. It has been tested to an electrical specification designed to determine its conformance with these parameters. Due to process differences in the manufacture of this device, this device may have different performance characteristics than its earlier version. These differences may cause this device to perform differently in your application than the earlier version of this device.