**Welcome to E-XFL.COM**

**What is "Embedded - Microcontrollers"?**

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

**Applications of "Embedded - Microcontrollers"**

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 10MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 33 |
| Program Memory Size | 14KB (8K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 368 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 5.5V |
| Data Converters | A/D 8x10b |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-LCC (J-Lead) |
| Supplier Device Package | 44-PLCC (16.59x16.59) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16lf877at-i-l |

## 1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

Additional information may be found in the PIC® Mid-Range 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.

**TABLE 1-1:    PIC16F87XA DEVICE FEATURES**

| Key Features | PIC16F873A | PIC16F874A | PIC16F876A | PIC16F877A |
|---|---|---|---|---|
| Operating Frequency | DC – 20 MHz | DC – 20 MHz | DC – 20 MHz | DC – 20 MHz |
| Resets (and Delays) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) | POR, BOR (PWRT, OST) |
| Flash Program Memory (14-bit words) | 4K | 4K | 8K | 8K |
| Data Memory (bytes) | 192 | 192 | 368 | 368 |
| EEPROM Data Memory (bytes) | 128 | 128 | 256 | 256 |
| Interrupts | 14 | 15 | 14 | 15 |
| I/O Ports | Ports A, B, C | Ports A, B, C, D, E | Ports A, B, C | Ports A, B, C, D, E |
| Timers | 3 | 3 | 3 | 3 |
| Capture/Compare/PWM modules | 2 | 2 | 2 | 2 |
| Serial Communications | MSSP, USART | MSSP, USART | MSSP, USART | MSSP, USART |
| Parallel Communications | — | PSP | — | PSP |
| 10-bit Analog-to-Digital Module | 5 input channels | 8 input channels | 5 input channels | 8 input channels |
| Analog Comparators | 2 | 2 | 2 | 2 |
| Instruction Set | 35 Instructions | 35 Instructions | 35 Instructions | 35 Instructions |
| Packages | 28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN | 40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN | 28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN | 40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN |

**TABLE 1-3:** **PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)**

| Pin Name | PDIP Pin# | PLCC Pin# | TQFP Pin# | QFN Pin# | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|---|---|
| | | | | | | | PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. |
| RB0/INT | 33 | 36 | 8 | 9 | | TTL/ST[1] | |
| RB0 | | | | | I/O | | Digital I/O. |
| INT | | | | | I | | External interrupt. |
| RB1 | 34 | 37 | 9 | 10 | I/O | TTL | Digital I/O. |
| RB2 | 35 | 38 | 10 | 11 | I/O | TTL | Digital I/O. |
| RB3/PGM | 36 | 39 | 11 | 12 | | TTL | |
| RB3 | | | | | I/O | | Digital I/O. |
| PGM | | | | | I | | Low-voltage ICSP programming enable pin. |
| RB4 | 37 | 41 | 14 | 14 | I/O | TTL | Digital I/O. |
| RB5 | 38 | 42 | 15 | 15 | I/O | TTL | Digital I/O. |
| RB6/PGC | 39 | 43 | 16 | 16 | | TTL/ST[2] | |
| RB6 | | | | | I/O | | Digital I/O. |
| PGC | | | | | I | | In-circuit debugger and ICSP programming clock. |
| RB7/PGD | 40 | 44 | 17 | 17 | | TTL/ST[2] | |
| RB7 | | | | | I/O | | Digital I/O. |
| PGD | | | | | I/O | | In-circuit debugger and ICSP programming data. |

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

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.
      **2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.
      **3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

**TABLE 1-3: PIC16F874A/877A PINOUT DESCRIPTION (CONTINUED)**

| Pin Name | PDIP Pin# | PLCC Pin# | TQFP Pin# | QFN Pin# | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|---|---|
| | | | | | | | PORTD is a bidirectional I/O port or Parallel Slave Port when interfacing to a microprocessor bus. |
| RD0/PSP0 | 19 | 21 | 38 | 38 | | ST/TTL(3) | |
| RD0 | | | | | I/O | | Digital I/O. |
| PSP0 | | | | | I/O | | Parallel Slave Port data. |
| RD1/PSP1 | 20 | 22 | 39 | 39 | | ST/TTL(3) | |
| RD1 | | | | | I/O | | Digital I/O. |
| PSP1 | | | | | I/O | | Parallel Slave Port data. |
| RD2/PSP2 | 21 | 23 | 40 | 40 | | ST/TTL(3) | |
| RD2 | | | | | I/O | | Digital I/O. |
| PSP2 | | | | | I/O | | Parallel Slave Port data. |
| RD3/PSP3 | 22 | 24 | 41 | 41 | | ST/TTL(3) | |
| RD3 | | | | | I/O | | Digital I/O. |
| PSP3 | | | | | I/O | | Parallel Slave Port data. |
| RD4/PSP4 | 27 | 30 | 2 | 2 | | ST/TTL(3) | |
| RD4 | | | | | I/O | | Digital I/O. |
| PSP4 | | | | | I/O | | Parallel Slave Port data. |
| RD5/PSP5 | 28 | 31 | 3 | 3 | | ST/TTL(3) | |
| RD5 | | | | | I/O | | Digital I/O. |
| PSP5 | | | | | I/O | | Parallel Slave Port data. |
| RD6/PSP6 | 29 | 32 | 4 | 4 | | ST/TTL(3) | |
| RD6 | | | | | I/O | | Digital I/O. |
| PSP6 | | | | | I/O | | Parallel Slave Port data. |
| RD7/PSP7 | 30 | 33 | 5 | 5 | | ST/TTL(3) | |
| RD7 | | | | | I/O | | Digital I/O. |
| PSP7 | | | | | I/O | | Parallel Slave Port data. |
| | | | | | | | PORTE is a bidirectional I/O port. |
| RE0/$\overline{RD}$/AN5 | 8 | 9 | 25 | 25 | | ST/TTL(3) | |
| RE0 | | | | | I/O | | Digital I/O. |
| $\overline{RD}$ | | | | | I | | Read control for Parallel Slave Port. |
| AN5 | | | | | I | | Analog input 5. |
| RE1/$\overline{WR}$/AN6 | 9 | 10 | 26 | 26 | | ST/TTL(3) | |
| RE1 | | | | | I/O | | Digital I/O. |
| $\overline{WR}$ | | | | | I | | Write control for Parallel Slave Port. |
| AN6 | | | | | I | | Analog input 6. |
| RE2/$\overline{CS}$/AN7 | 10 | 11 | 27 | 27 | | ST/TTL(3) | |
| RE2 | | | | | I/O | | Digital I/O. |
| $\overline{CS}$ | | | | | I | | Chip select control for Parallel Slave Port. |
| AN7 | | | | | I | | Analog input 7. |
| VSS | 12, 31 | 13, 34 | 6, 29 | 6, 30, 31 | P | — | Ground reference for logic and I/O pins. |
| VDD | 11, 32 | 12, 35 | 7, 28 | 7, 8, 28, 29 | P | — | Positive supply for logic and I/O pins. |
| NC | — | 1, 17, 28, 40 | 12,13, 33, 34 | 13 | — | — | These pins are not internally connected. These pins should be left unconnected. |

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

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.
    **2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.
    **3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

**NOTES:**

## 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 2-5 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 2-5: LOADING OF PC IN DIFFERENT SITUATIONS
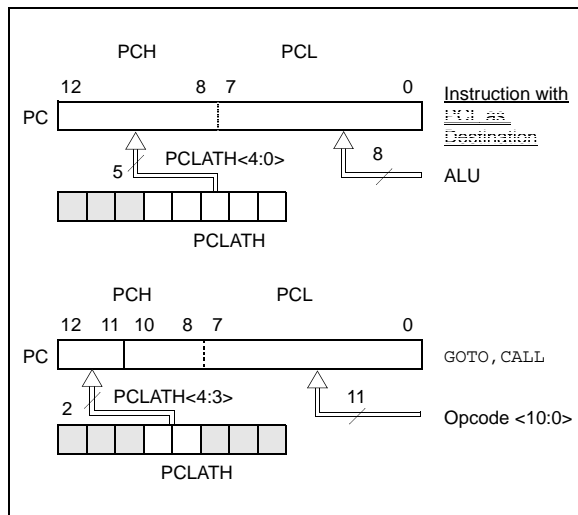


### 2.3.1 COMPUTED GOTO

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, *AN556, "Implementing a Table Read"* (DS00556).

### 2.3.2 STACK

The PIC16F87XA 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 POP'ed 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

All PIC16F87XA devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper 2 bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is popped off the stack. Therefore, manipulation of the PCLATH<4:3> bits is not required for the RETURN instructions (which POPs the address from the stack).

| Note: | The contents of the PCLATH register are unchanged after a RETURN or RETFIE instruction is executed. The user must rewrite the contents of the PCLATH register for any subsequent subroutine calls or GOTO instructions. |
| --- | --- |

Example 2-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that PCLATH is saved and restored by the Interrupt Service Routine (if interrupts are used).

### EXAMPLE 2-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

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

# PIC16F87XA

**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
         Reads '0' after a POR; this bit cannot be changed while a 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 $\overline{MCLR}$ 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 |

                          

## 4.6 Parallel Slave Port

The Parallel Slave Port (PSP) is not implemented on the PIC16F873A or PIC16F876A.

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

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PSPMODE enables port pin RE0/$\overline{RD}$/AN5 to be the $\overline{RD}$ input, RE1/$\overline{WR}$/AN6 to be the $\overline{WR}$ input and RE2/$\overline{CS}$/AN7 to be the $\overline{CS}$ (Chip Select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set). 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 external device is controlling the direction of data flow.

A write to the PSP occurs when both the $\overline{CS}$ and $\overline{WR}$ lines are first detected low. When either the $\overline{CS}$ or $\overline{WR}$ lines become high (level triggered), 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 4-11). 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 4-12), 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 4-10:     PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)**



**Note 1:** I/O pins have protection diodes to VDD and VSS.
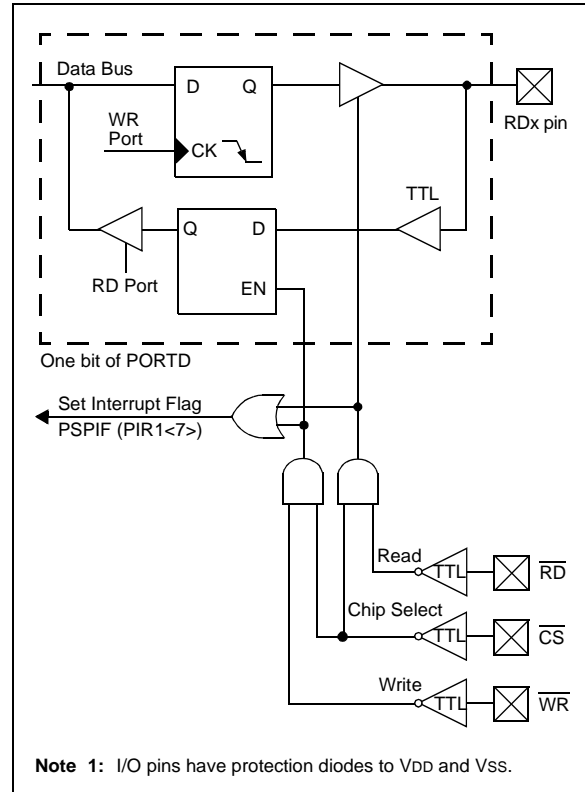
## 9.4.10    I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full flag bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time specification, parameter #106). SCL is held low for one Baud Rate Generator rollover count (T$_{BRG}$). Data should be valid before SCL is released high (see data setup time specification, parameter #107). When the SCL pin is released high, it is held that way for T$_{BRG}$. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an $\overline{ACK}$ bit during the ninth bit time, if an address match occurred or if data was received properly. The status of $\overline{ACK}$ is written into the ACKDT bit on the falling edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 9-21).

After the write to the SSPBUF, each bit of address will be shifted out on the falling edge of SCL, until all seven address bits and the R/$\overline{W}$ bit are completed. On the falling edge of the eighth clock, the master will deassert the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT status bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

### 9.4.10.1    BF Status Flag

In Transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all eight bits are shifted out.

### 9.4.10.2    WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

WCOL must be cleared in software.

### 9.4.10.3    ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an Acknowledge ($\overline{ACK}$ = 0) and is set when the slave does Not Acknowledge ($\overline{ACK}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call) or when the slave has properly received its data.

## 9.4.11    I²C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (SSPCON2<3>).

> **Note:** The MSSP module must be in an Idle state before the RCEN bit is set or the RCEN bit will be disregarded.

The Baud Rate Generator begins counting and on each rollover, the state of the SCL pin changes (high to low/low to high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state, awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>).

### 9.4.11.1    BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when the SSPBUF register is read.

### 9.4.11.2    SSPOV Status Flag

In receive operation, the SSPOV bit is set when 8 bits are received into the SSPSR and the BF flag bit is already set from a previous reception.
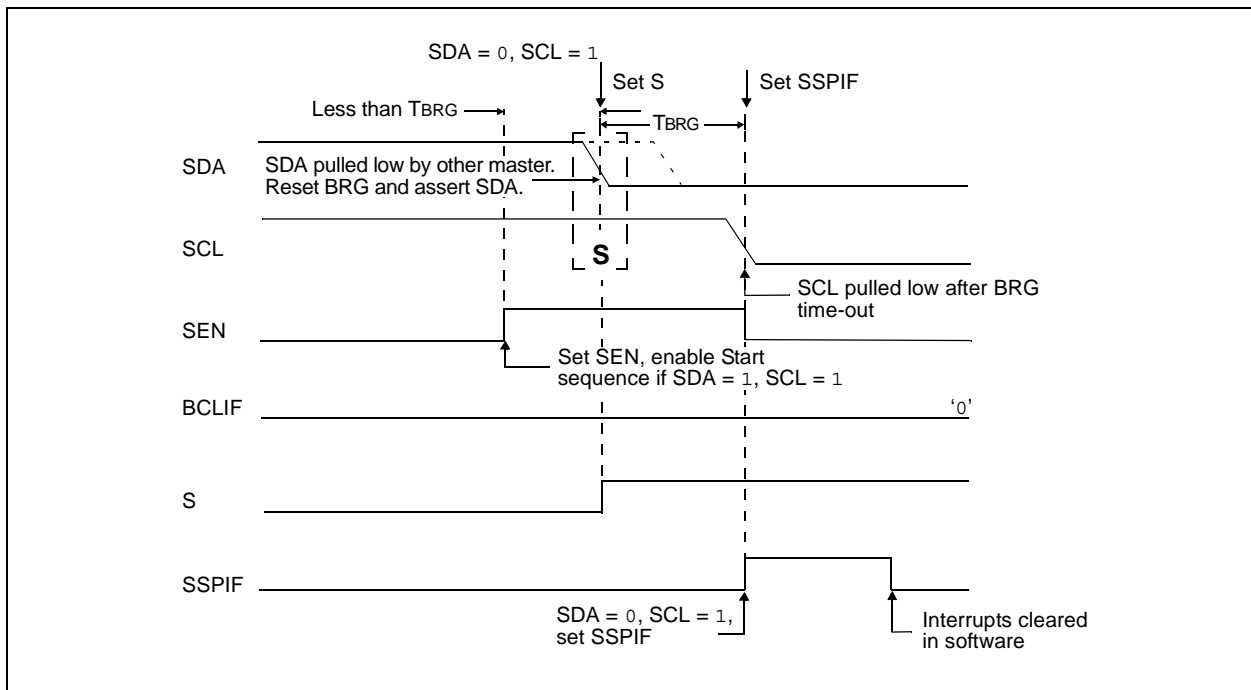
### 9.4.11.3    WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

**FIGURE 9-27:** **BUS COLLISION DURING START CONDITION (SCL = 0)**
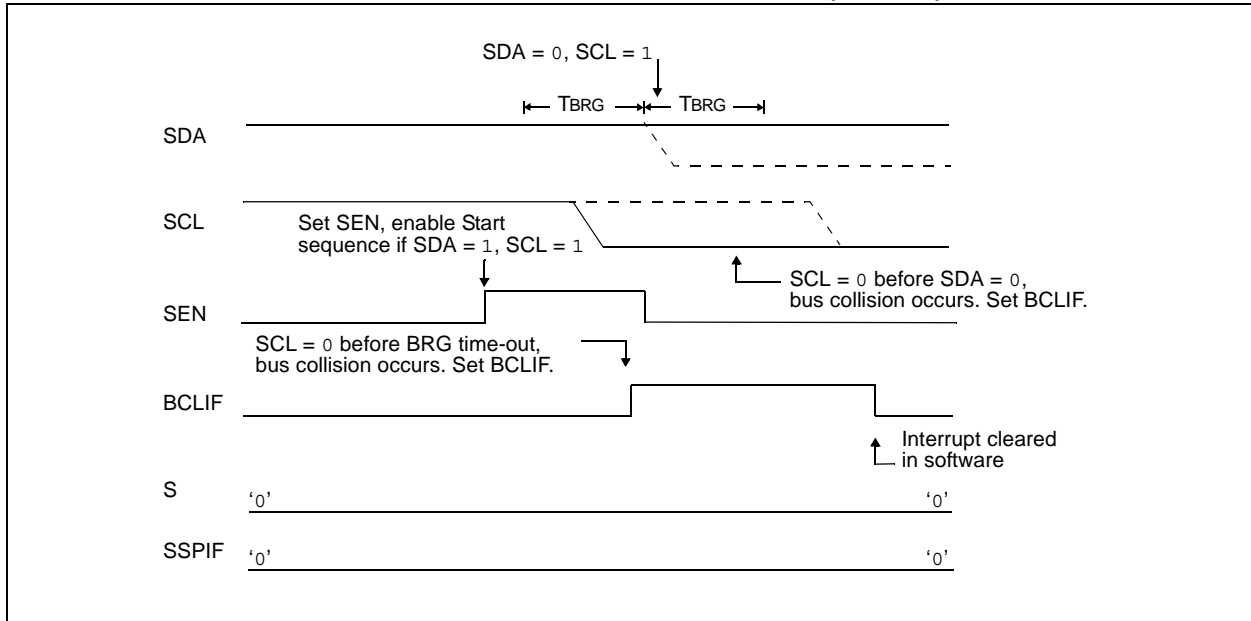


**FIGURE 9-28:** **BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION**

# PIC16F87XA

**TABLE 10-8:** **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 | TMR0IE | INTE | RBIE | TMR0IF | INTF | R0IF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF[1] | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 19h | TXREG | USART Transmit Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE[1] | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

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

**Note   1:** Bits PSPIE and PSPIF are reserved on 28-pin devices; always maintain these bits clear.

**FIGURE 10-9:** **SYNCHRONOUS TRANSMISSION**



**Note:** Sync Master mode; SPBRG = 0. Continuous transmission of two 8-bit words.
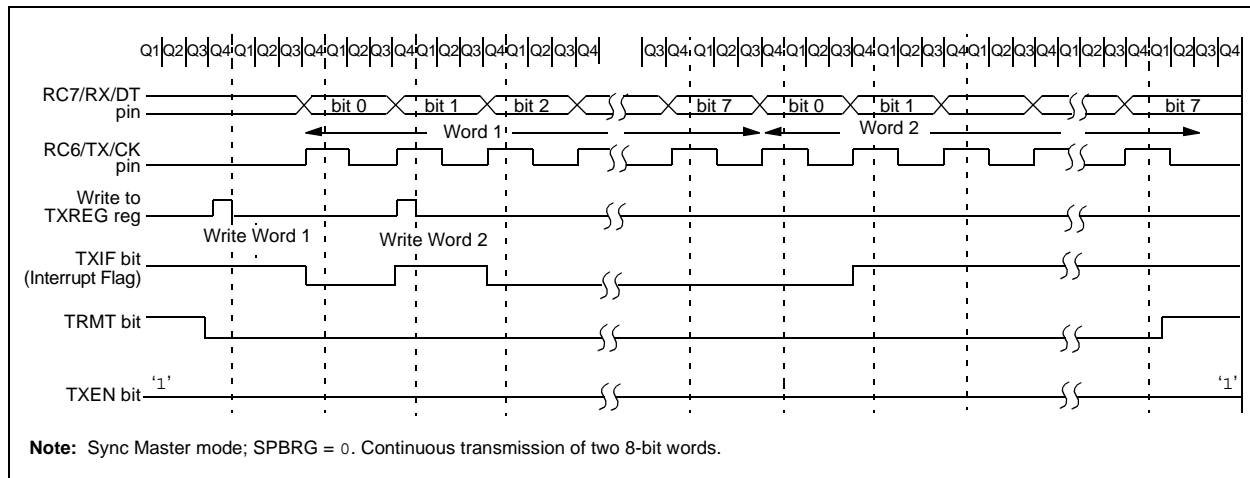
**FIGURE 10-10:** **SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**
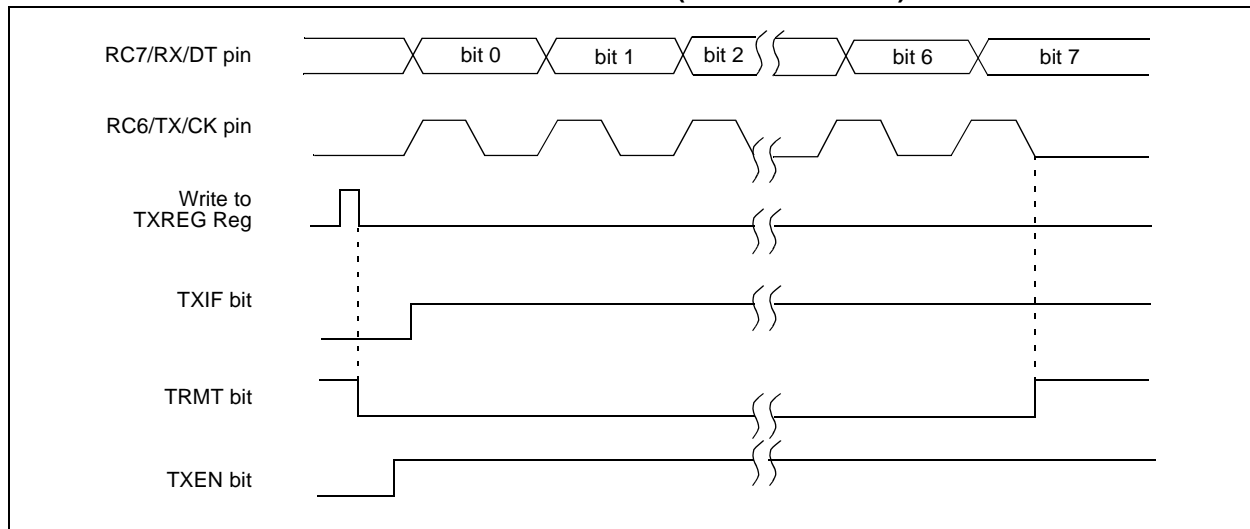
The ADRESH:ADRESL registers contain the 10-bit result of the A/D conversion. When the A/D conversion is complete, the result is loaded into this A/D Result register pair, the GO/$\overline{DONE}$ bit (ADCON0<2>) is cleared and the A/D interrupt flag bit ADIF is set. The block diagram of the A/D module is shown in Figure 11-1.

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 11.1 "A/D Acquisition Requirements"**. After this acquisition time has elapsed, the A/D conversion can be started.

To do an A/D Conversion, follow these steps:

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/$\overline{DONE}$ bit (ADCON0)

5. Wait for A/D conversion to complete by either:
   • Polling for the GO/$\overline{DONE}$ bit to be cleared (interrupts disabled); 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 $T_{AD}$.

**FIGURE 11-1:     A/D BLOCK DIAGRAM**



**Note   1:** Not available on 28-pin devices.
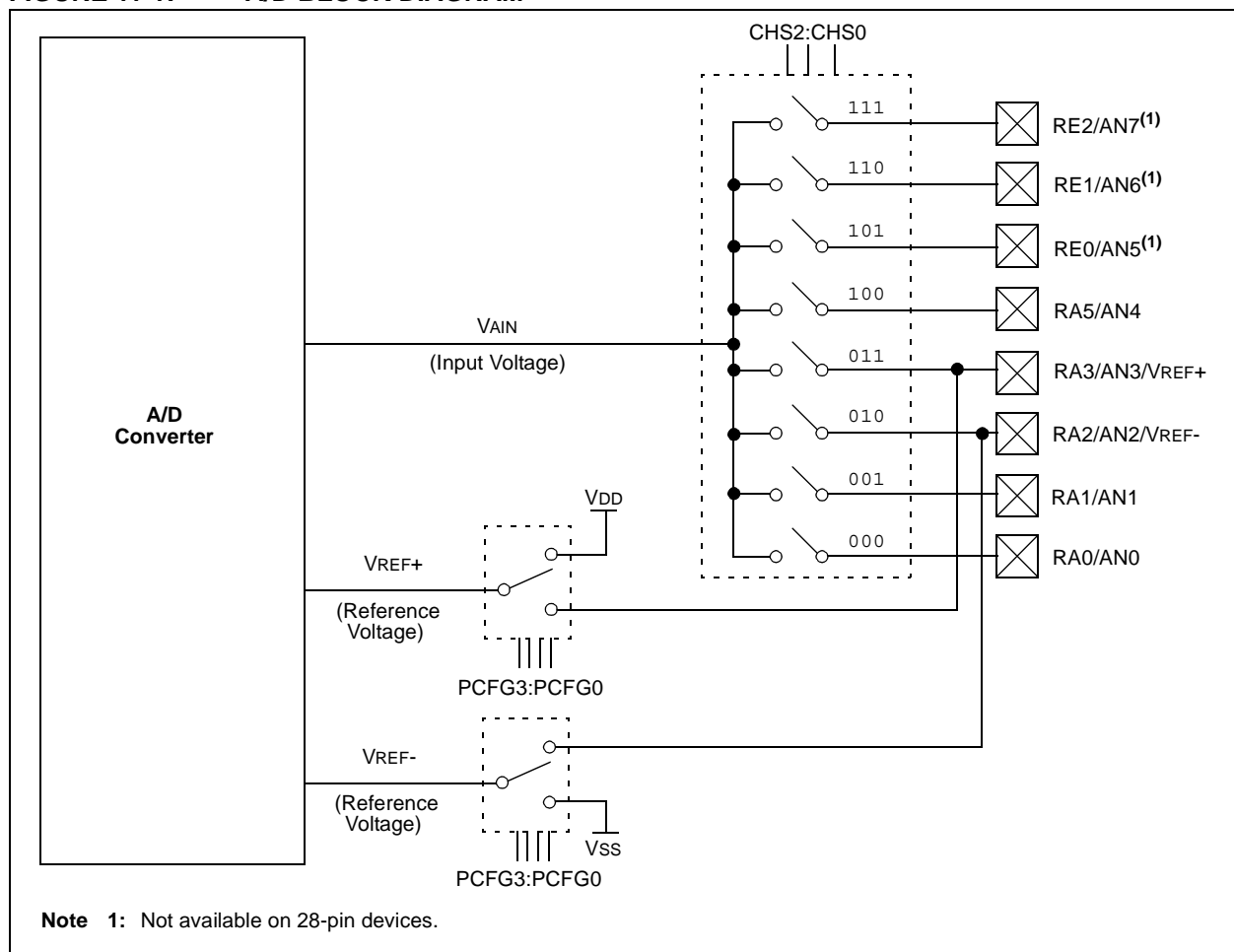
## 12.0   COMPARATOR MODULE

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with I/O port pins RA0 through RA3, while the outputs are multiplexed to pins RA4 and RA5. The on-chip voltage reference (**Section 13.0 "Comparator Voltage Reference Module"**) can also be an input to the comparators.

The CMCON register (Register 12-1) controls the comparator input and output multiplexers. A block diagram of the various comparator configurations is shown in Figure 12-1.

**REGISTER 12-1:**   **CMCON REGISTER**

| R-0 | R-0 | R/W-0 | R/W-0 | R/W-0 | R/W-1 | R/W-1 | R/W-1 |
|-----|-----|-------|-------|-------|-------|-------|-------|
| C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 |

bit 7                                                              bit 0

bit 7      **C2OUT**: Comparator 2 Output bit

When C2INV = 0:
1 = C2 $V_{IN}+$ > C2 $V_{IN}-$
0 = C2 $V_{IN}+$ < C2 $V_{IN}-$
When C2INV = 1:
1 = C2 $V_{IN}+$ < C2 $V_{IN}-$
0 = C2 $V_{IN}+$ > C2 $V_{IN}-$

bit 6      **C1OUT**: Comparator 1 Output bit

When C1INV = 0:
1 = C1 $V_{IN}+$ > C1 $V_{IN}-$
0 = C1 $V_{IN}+$ < C1 $V_{IN}-$
When C1INV = 1:
1 = C1 $V_{IN}+$ < C1 $V_{IN}-$
0 = C1 $V_{IN}+$ > C1 $V_{IN}-$

bit 5      **C2INV**: Comparator 2 Output Inversion bit

1 = C2 output inverted
0 = C2 output not inverted

bit 4      **C1INV**: Comparator 1 Output Inversion bit

1 = C1 output inverted
0 = C1 output not inverted

bit 3      **CIS**: Comparator Input Switch bit

When CM2:CM0 = 110:
1 = C1 $V_{IN}-$ connects to RA3/AN3
       C2 $V_{IN}-$ connects to RA2/AN2
0 = C1 $V_{IN}-$ connects to RA0/AN0
       C2 $V_{IN}-$ connects to RA1/AN1

bit 2      **CM2:CM0**: Comparator Mode bits

Figure 12-1 shows the Comparator modes and CM2:CM0 bit settings.

| 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 |

# PIC16F87XA

## TABLE 15-2: PIC16F87XA INSTRUCTION SET

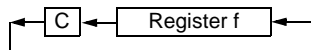| Mnemonic, Operands | | Description | Cycles | 14-Bit Opcode MSb        LSb | Status Affected | Notes |
|---|---|---|---|---|---|---|
| **BYTE-ORIENTED FILE REGISTER OPERATIONS** | | | | | | |
| ADDWF | f, d | Add W and f | 1 | 00  0111  dfff  ffff | C,DC,Z | 1,2 |
| ANDWF | f, d | AND W with f | 1 | 00  0101  dfff  ffff | Z | 1,2 |
| CLRF | f | Clear f | 1 | 00  0001  lfff  ffff | Z | 2 |
| CLRW | - | Clear W | 1 | 00  0001  0xxx  xxxx | Z | |
| COMF | f, d | Complement f | 1 | 00  1001  dfff  ffff | Z | 1,2 |
| DECF | f, d | Decrement f | 1 | 00  0011  dfff  ffff | Z | 1,2 |
| DECFSZ | f, d | Decrement f, Skip if 0 | 1(2) | 00  1011  dfff  ffff | | 1,2,3 |
| INCF | f, d | Increment f | 1 | 00  1010  dfff  ffff | Z | 1,2 |
| INCFSZ | f, d | Increment f, Skip if 0 | 1(2) | 00  1111  dfff  ffff | | 1,2,3 |
| IORWF | f, d | Inclusive OR W with f | 1 | 00  0100  dfff  ffff | Z | 1,2 |
| MOVF | f, d | Move f | 1 | 00  1000  dfff  ffff | Z | 1,2 |
| MOVWF | f | Move W to f | 1 | 00  0000  lfff  ffff | | |
| NOP | - | No Operation | 1 | 00  0000  0xx0  0000 | | |
| RLF | f, d | Rotate Left f through Carry | 1 | 00  1101  dfff  ffff | C | 1,2 |
| RRF | f, d | Rotate Right f through Carry | 1 | 00  1100  dfff  ffff | C | 1,2 |
| SUBWF | f, d | Subtract W from f | 1 | 00  0010  dfff  ffff | C,DC,Z | 1,2 |
| SWAPF | f, d | Swap nibbles in f | 1 | 00  1110  dfff  ffff | | 1,2 |
| XORWF | f, d | Exclusive OR W with f | 1 | 00  0110  dfff  ffff | Z | 1,2 |
| **BIT-ORIENTED FILE REGISTER OPERATIONS** | | | | | | |
| BCF | f, b | Bit Clear f | 1 | 01  00bb  bfff  ffff | | 1,2 |
| BSF | f, b | Bit Set f | 1 | 01  01bb  bfff  ffff | | 1,2 |
| BTFSC | f, b | Bit Test f, Skip if Clear | 1 (2) | 01  10bb  bfff  ffff | | 3 |
| BTFSS | f, b | Bit Test f, Skip if Set | 1 (2) | 01  11bb  bfff  ffff | | 3 |
| **LITERAL AND CONTROL OPERATIONS** | | | | | | |
| ADDLW | k | Add Literal and W | 1 | 11  111x  kkkk  kkkk | C,DC,Z | |
| ANDLW | k | AND Literal with W | 1 | 11  1001  kkkk  kkkk | Z | |
| CALL | k | Call Subroutine | 2 | 10  0kkk  kkkk  kkkk | | |
| CLRWDT | - | Clear Watchdog Timer | 1 | 00  0000  0110  0100 | $\overline{TO},\overline{PD}$ | |
| GOTO | k | Go to Address | 2 | 10  1kkk  kkkk  kkkk | | |
| IORLW | k | Inclusive OR Literal with W | 1 | 11  1000  kkkk  kkkk | Z | |
| MOVLW | k | Move Literal to W | 1 | 11  00xx  kkkk  kkkk | | |
| RETFIE | - | Return from Interrupt | 2 | 00  0000  0000  1001 | | |
| RETLW | k | Return with Literal in W | 2 | 11  01xx  kkkk  kkkk | | |
| RETURN | - | Return from Subroutine | 2 | 00  0000  0000  1000 | | |
| SLEEP | - | Go into Standby mode | 1 | 00  0000  0110  0011 | $\overline{TO},\overline{PD}$ | |
| SUBLW | k | Subtract W from Literal | 1 | 11  110x  kkkk  kkkk | C,DC,Z | |
| XORLW | k | Exclusive OR Literal with W | 1 | 11  1010  kkkk  kkkk | Z | |

**Note 1:** When an I/O register is modified as a function of itself ( e.g., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

**2:** If this instruction is executed on the TMR0 register (and where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.

**3:** If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

---

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

---

# PIC16F87XA

## RLF — Rotate Left f through Carry

| | |
|---|---|
| Syntax: | [ *label* ]  RLF   f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| 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'. |

```
┌──────────────────────────┐
│  ┌─┐   ┌──────────────┐   │
└─→│C│←──│  Register f   │←──┘
   └─┘   └──────────────┘
```

## RETURN — Return from Subroutine

| | |
|---|---|
| Syntax: | [ *label* ]   RETURN |
| Operands: | None |
| Operation: | TOS $\rightarrow$ 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. |

## RRF — Rotate Right f through Carry

| | |
|---|---|
| Syntax: | [ *label* ]   RRF   f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| 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'. |

```
┌──────────────────────────┐
│  ┌─┐   ┌──────────────┐   │
└──│C│←──│  Register f   │←──┘
   └─┘   └──────────────┘
```

## SLEEP

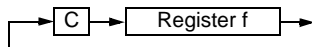| | |
|---|---|
| Syntax: | [ *label* ]  SLEEP |
| Operands: | None |
| Operation: | 00h $\rightarrow$ WDT,<br>0 $\rightarrow$ WDT prescaler,<br>1 $\rightarrow$ $\overline{\text{TO}}$,<br>0 $\rightarrow$ $\overline{\text{PD}}$ |
| Status Affected: | $\overline{\text{TO}}$, $\overline{\text{PD}}$ |
| Description: | The power-down status bit, $\overline{\text{PD}}$, is cleared. Time-out status bit, $\overline{\text{TO}}$, is set. Watchdog Timer and its prescaler are cleared. The processor is put into Sleep mode with the oscillator stopped. |

## 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. |

## SUBWF — Subtract W from f

| | |
|---|---|
| Syntax: | [ *label* ]  SUBWF  f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| Operation: | (f) - (W) $\rightarrow$ (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'. |

## 16.0   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

## 16.1   MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit micro-controller 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.

## 16.2   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.
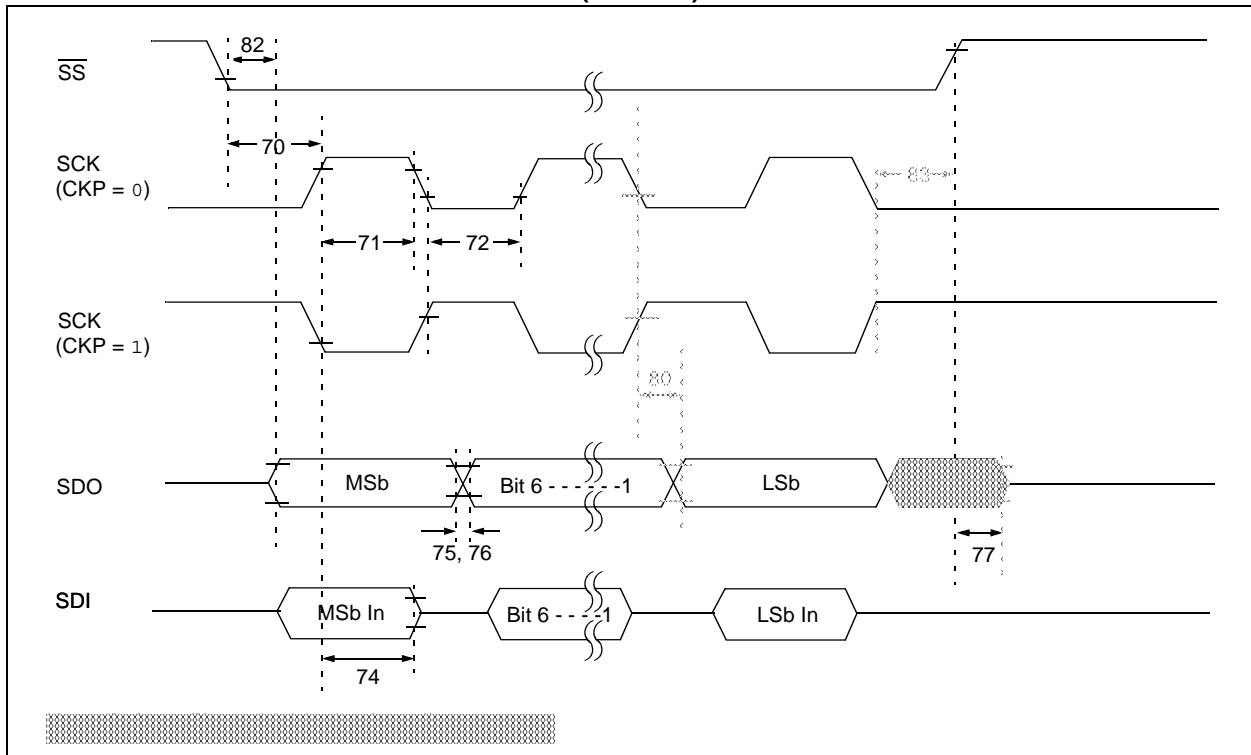
The MPASM assembler features include:

- 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

# PIC16F87XA

**FIGURE 17-13: SPI SLAVE MODE TIMING (CKE = 0)**
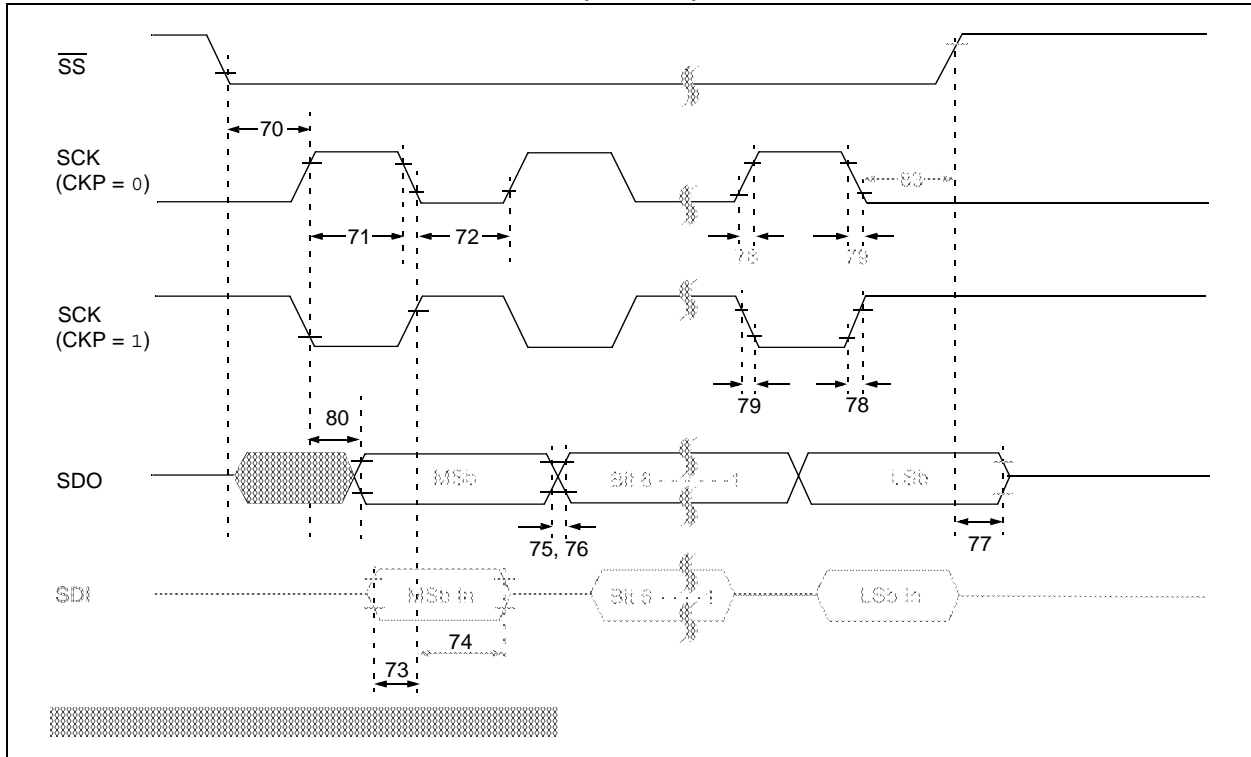


**FIGURE 17-14: SPI SLAVE MODE TIMING (CKE = 1)**

# PIC16F87XA

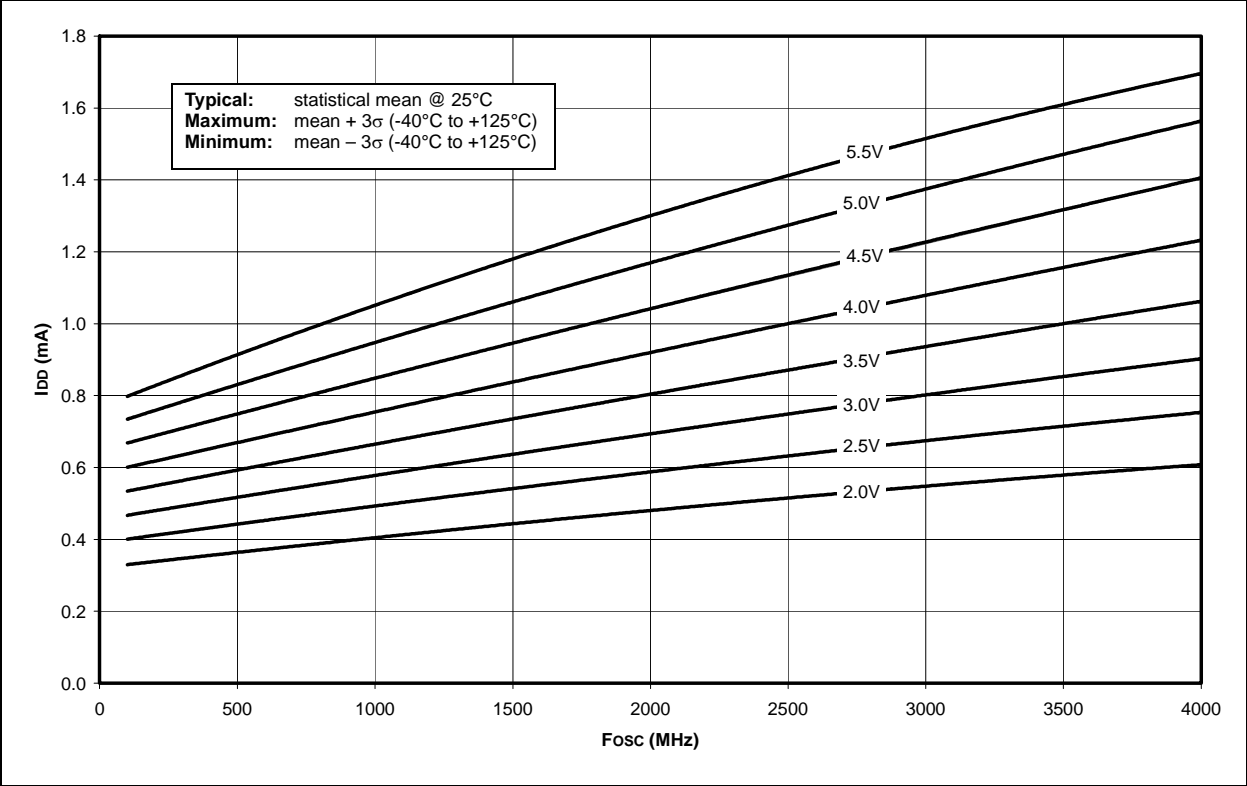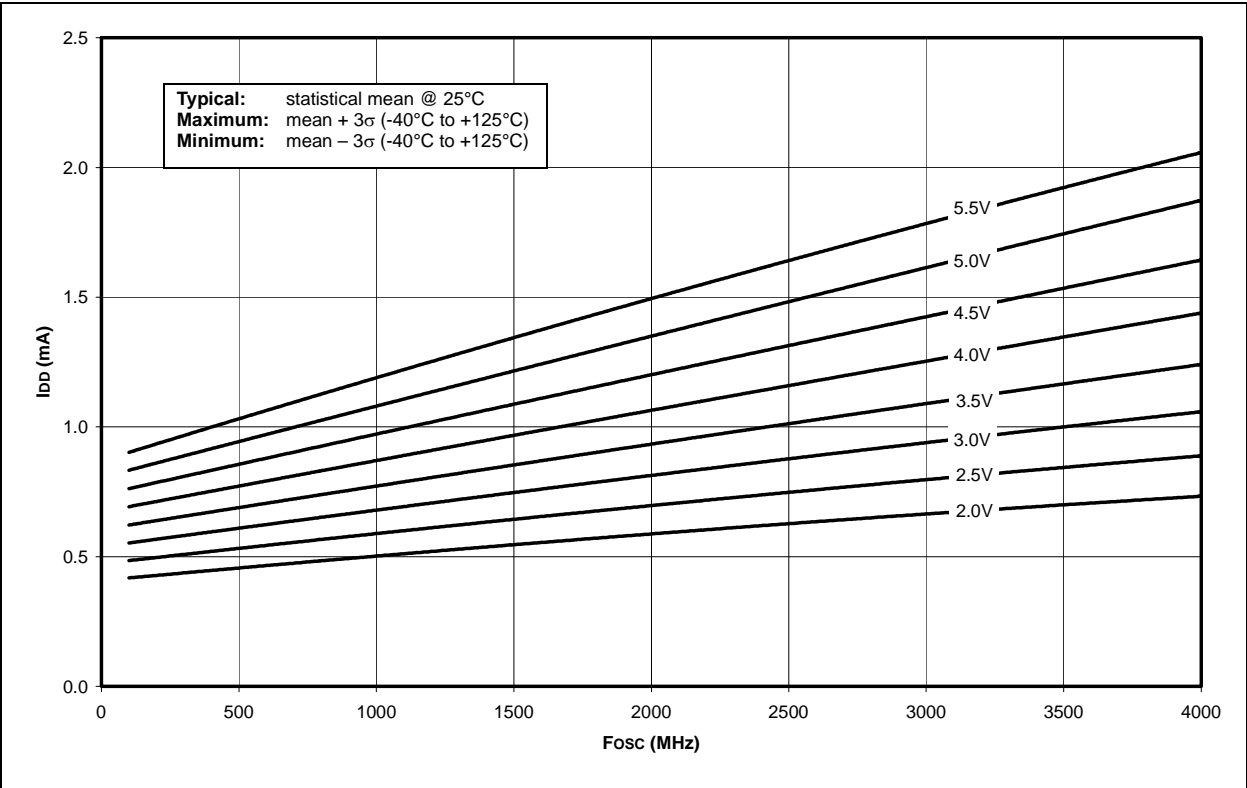**FIGURE 18-3:** TYPICAL I_DD vs. F_OSC OVER V_DD (XT MODE)



**FIGURE 18-4:** MAXIMUM I_DD vs. F_OSC OVER V_DD (XT MODE)

# PIC16F87XA

**FIGURE 18-23: A/D NONLINEARITY vs. VREFH (VDD = VREFH, -40°C TO +125°C)**
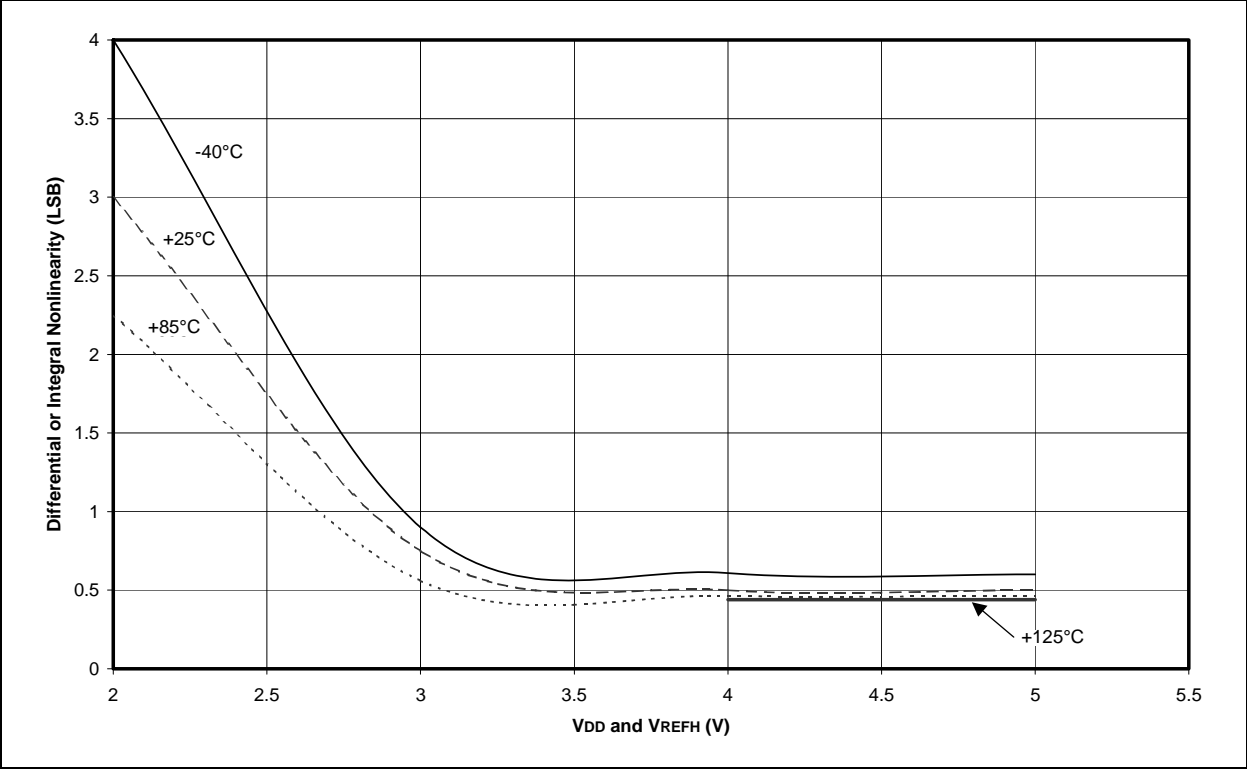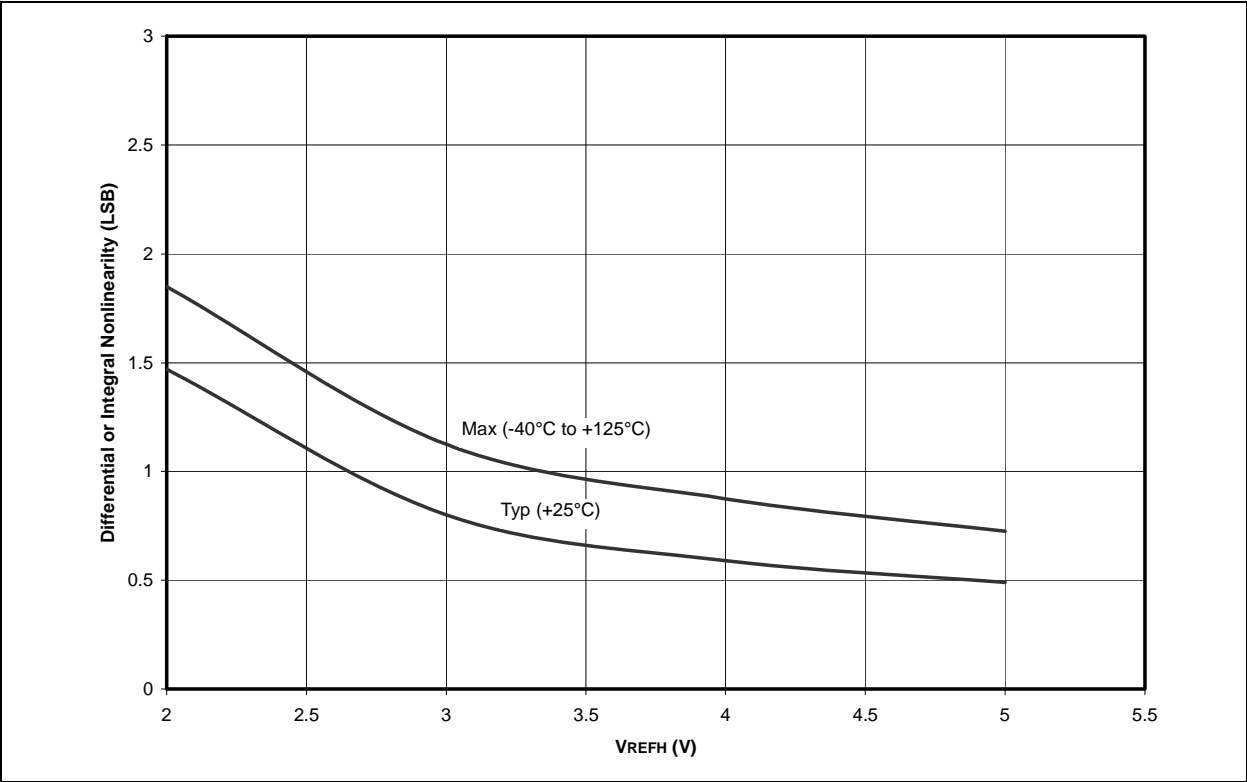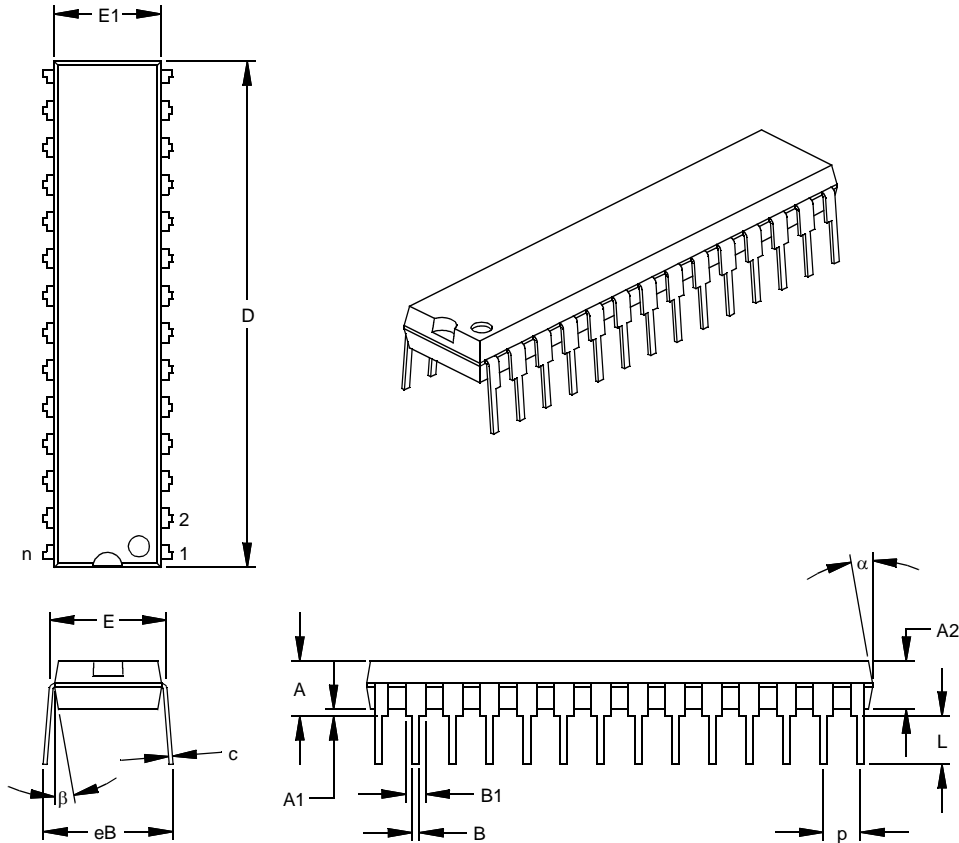


**FIGURE 18-24: A/D NONLINEARITY vs. VREFH (VDD = 5V, -40°C TO +125°C)**

## 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

**NOTES:**