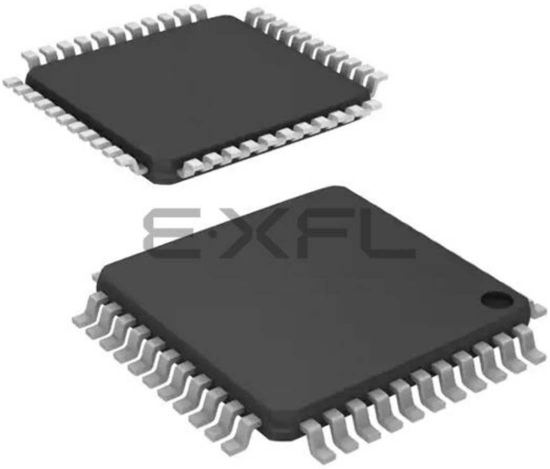


Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	36
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 28x10b; D/A 4x5b, 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1779t-i-pt

TABLE 1-3: PIC16(L)F1777/9 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RD1/AN21/PRG3IN1/PRG4IN0/ C1IN4-/C2IN4-/C3IN4-/C4IN4-/ C5IN4-/C6IN4-/C7IN4-/C8IN4-/ OPA4OUT/OPA3IN1+/OPA3IN1-	RD1	TTL/ST	CMOS	General purpose I/O.
	AN21	AN	—	ADC Channel 21 input.
	PRG3IN1	AN	—	Ramp generator 3 reference voltage input.
	PRG4IN0	AN	—	Ramp generator 4 reference voltage input.
	C1IN4-	AN	—	Comparator 1 negative input.
	C2IN4-	AN	—	Comparator 2 negative input.
	C3IN4-	AN	—	Comparator 3 negative input.
	C4IN4-	AN	—	Comparator 4 negative input.
	C5IN4-	AN	—	Comparator 5 negative input.
	C6IN4-	AN	—	Comparator 6 negative input.
	C7IN4-	AN	—	Comparator 7 negative input.
	C8IN4-	AN	—	Comparator 8 negative input.
	OPA4OUT	—	AN	Operational amplifier 4 output.
	OPA3IN1+	AN	—	Operational amplifier 3 non-inverting input.
	OPA3IN1-	AN	—	Operational amplifier 3 inverting input.
RD2/AN22/DAC4OUT1/ OPA4IN0-	RD2	TTL/ST	CMOS	General purpose I/O.
	AN22	AN	—	ADC Channel 22 input.
	DAC4OUT1	—	AN	DAC4 voltage output.
	OPA4IN0-	AN	—	Operational amplifier 4 inverting input.
RD3/AN23/C8IN2-	RD3	TTL/ST	CMOS	General purpose I/O.
	AN23	AN	—	ADC Channel 23 input.
	C8IN2-	AN	—	Comparator 8 negative input.
RD4/AN24/C7IN2-	RD4	TTL/ST	CMOS	General purpose I/O.
	AN24	AN	—	ADC Channel 24 input.
	C7IN2-	AN	—	Comparator 8 negative input.
RD5/AN25/C7IN3-/C8IN3-	RD5	TTL/ST	CMOS	General purpose I/O.
	AN25	AN	—	ADC Channel 25 input.
	C7IN3-	AN	—	Comparator 7 negative input.
	C8IN3-	AN	—	Comparator 8 negative input.
RD6/AN26/C7IN1+	RD6	TTL/ST	CMOS	General purpose I/O.
	AN26	AN	—	ADC Channel 26 input.
	C7IN1+	AN	—	Comparator 7 positive input.
RD7/AN27/C8IN1+	RD7	TTL/ST	CMOS	General purpose I/O.
	AN27	AN	—	ADC Channel 27 input.
	C8IN1+	AN	—	Comparator 8 positive input.
RE0/AN5/DAC6REF1+/ DAC8REF1+	RE0	TTL/ST	CMOS	General purpose I/O.
	AN5	AN	—	ADC Channel 5 input.
	DAC6REF1+	AN	—	DAC6 positive reference.
	DAC8REF1+	AN	—	DAC8 positive reference.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I^2C = Schmitt Trigger input with I^2C
HP = High Power XTAL = Crystal levels

Note 1: Default peripheral input. Alternate pins can be selected as the peripheral input with the PPS input selection registers.
2: All pin digital outputs default to PORT latch data. Alternate outputs can be selected as the peripheral digital output with the PPS output selection registers.
3: These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.

3.0 MEMORY ORGANIZATION

These devices contain the following types of memory:

- Program Memory
  - Configuration Words
  - Device ID
  - User ID
  - Flash Program Memory
- Data Memory
  - Core Registers
  - Special Function Registers
  - General Purpose RAM
  - Common RAM

Note 1:	The method to access Flash memory through the PMCON registers is described in **Section 10.0 "Flash Program Memory Control"**.

The following features are associated with access and control of program memory and data memory:

- PCL and PCLATH
- Stack
- Indirect Addressing

3.1 Program Memory Organization

The enhanced mid-range core has a 15-bit program counter capable of addressing a 32K x 14 program memory space. Table 3-1 shows the memory sizes implemented for the PIC16(L)F1777/8/9 family. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figures 3-1 and 3-2).

3.2 High-Endurance Flash

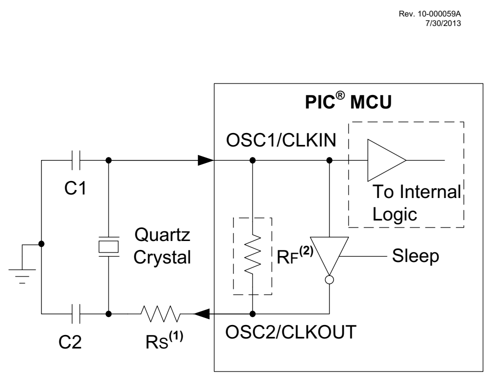
This device has a 128-byte section of high-endurance program Flash memory (PFM) in lieu of data EEPROM. This area is especially well suited for nonvolatile data storage that is expected to be updated frequently over the life of the end product. See **Section 10.2 "Flash Program Memory Overview"** for more information on writing data to PFM. See **Section 3.2.1.2 "Indirect Read with FSR"** for more information about using the FSR registers to read byte data stored in PFM.

TABLE 3-1: DEVICE SIZES AND ADDRESSES

Device	Program Memory Space (Words)	Last Program Memory Address	High-Endurance Flash Memory Address Range(1)
PIC16(L)F1777	8,192	1FFFh	1F80h-1FFFh
PIC16(L)F1778/9	16,384	3FFFh	3F80h-3FFFh

Note 1: High-endurance Flash applies to the low byte of each address in the range.

FIGURE 5-3: QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)



Note 1: A series resistor (RS) may be required for quartz crystals with low drive level.

2: The value of RF varies with the Oscillator mode selected (typically between 2 MΩ and 10 MΩ).

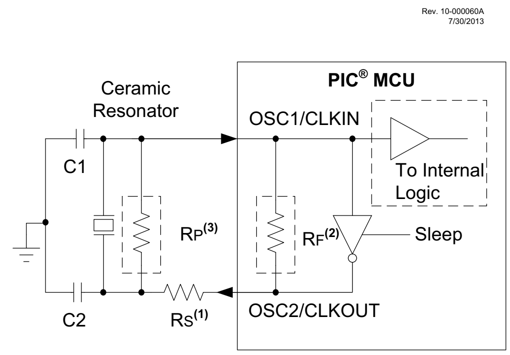
Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

2: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

3: For oscillator design assistance, reference the following Microchip Application Notes:

- AN826, "*Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices*" (DS00826)
- AN849, "*Basic PIC® Oscillator Design*" (DS00849)
- AN943, "*Practical PIC® Oscillator Analysis and Design*" (DS00943)
- AN949, "*Making Your Oscillator Work*" (DS00949)

FIGURE 5-4: CERAMIC RESONATOR OPERATION (XT OR HS MODE)



Note 1: A series resistor (RS) may be required for ceramic resonators with low drive level.

2: The value of RF varies with the Oscillator mode selected (typically between 2 MΩ and 10 MΩ).

3. An additional parallel feedback resistor (RP) may be required for proper ceramic resonator operation.

5.2.1.3 Oscillator Start-up Timer (OST)

If the oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from OSC1. This occurs following a Power-on Reset (POR) and when the Power-up Timer (PWRT) has expired (if configured), or a wake-up from Sleep. During this time, the program counter does not increment and program execution is suspended, unless either FSCM or Two-Speed Start-Up are enabled. In this case, code will continue to execute at the selected INTOSC frequency while the OST is counting. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the oscillator module.

In order to minimize latency between external oscillator start-up and code execution, the Two-Speed Clock Start-up mode can be selected (see **Section 5.4 "Two-Speed Clock Start-up Mode"**).

7.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIE1 or PIE2 registers)

The INTCON, PIR1 and PIR2 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (See **"Section 7.5 "Automatic Context Saving"**)
- PC is loaded with the interrupt vector 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1:	Individual interrupt flag bits are set, regardless of the state of any other enable bits.
2:	All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

7.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 7-2 and Figure 7-3 for more details.

EXAMPLE 10-3: WRITING TO FLASH PROGRAM MEMORY

```
; This write routine assumes the following:
; 1. 64 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
; stored in little endian format
; 3. A valid starting address (the least significant bits = 00000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)
;
        BCF        INTCON,GIE     ; Disable ints so required sequences will execute properly
        BANKSEL    PMADRH         ; Bank 3
        MOVF       ADDRH,W        ; Load initial address
        MOVWF      PMADRH         ;
        MOVF       ADDRL,W        ;
        MOVWF      PMADRL         ;
        MOVLW      LOW DATA_ADDR  ; Load initial data address
        MOVWF      FSR0L          ;
        MOVLW      HIGH DATA_ADDR ; Load initial data address
        MOVWF      FSR0H          ;
        BCF        PMCON1,CFGS    ; Not configuration space
        BSF        PMCON1,WREN    ; Enable writes
        BSF        PMCON1,LWLO    ; Only Load Write Latches
LOOP
        MOVIW      FSR0++         ; Load first data byte into lower
        MOVWF      PMDATL         ;
        MOVIW      FSR0++         ; Load second data byte into upper
        MOVWF      PMDATH         ;

        MOVF       PMADRL,W       ; Check if lower bits of address are '00000'
        XORLW      0x1F           ; Check if we're on the last of 32 addresses
        ANDLW      0x1F           ;
        BTFSC      STATUS,Z       ; Exit if last of 32 words,
        GOTO       START_WRITE    ;

Required Sequence
        MOVLW      55h            ; Start of required write sequence:
        MOVWF      PMCON2         ; Write 55h
        MOVLW      0AAh           ;
        MOVWF      PMCON2         ; Write AAh
        BSF        PMCON1,WR      ; Set WR bit to begin write
        NOP                       ; NOP instructions are forced as processor
                                  ; loads program memory write latches
        NOP                       ;

        INCF       PMADRL,F       ; Still loading latches Increment address
        GOTO       LOOP           ; Write next latches

START_WRITE
        BCF        PMCON1,LWLO    ; No more loading latches - Actually start Flash program
                                  ; memory write

Required Sequence
        MOVLW      55h            ; Start of required write sequence:
        MOVWF      PMCON2         ; Write 55h
        MOVLW      0AAh           ;
        MOVWF      PMCON2         ; Write AAh
        BSF        PMCON1,WR      ; Set WR bit to begin write
        NOP                       ; NOP instructions are forced as processor writes
                                  ; all the program memory write latches simultaneously
        NOP                       ; to program memory.
                                  ; After NOPs, the processor
                                  ; stalls until the self-write process in complete
                                  ; after write processor continues with 3rd instruction
        BCF        PMCON1,WREN    ; Disable writes
        BSF        INTCON,GIE     ; Enable interrupts
```

12.3 Bidirectional Pins

PPS selections for peripherals with bidirectional signals on a single pin must be made so that the PPS input and PPS output select the same pin. Peripherals that have bidirectional signals include:

- EUSART (synchronous operation)
- MSSP (I^2C)

Note:	The I^2C default input pins are I^2C and SMBus compatible and are the only pins on the device with this compatibility.

12.4 PPS Lock

The PPS includes a mode in which all input and output selections can be locked to prevent inadvertent changes. PPS selections are locked by setting the PPSLOCKED bit of the PPSLOCK register. Setting and clearing this bit requires a special sequence as an extra precaution against inadvertent changes. Examples of setting and clearing the PPSLOCKED bit are shown in Example 12-1.

EXAMPLE 12-1: PPS LOCK/UNLOCK SEQUENCE

```
; suspend interrupts
   bcf     INTCON,GIE
;   BANKSEL PPSLOCK    ; set bank
; required sequence, next 5 instructions
   movlw   0x55
   movwf   PPSLOCK
   movlw   0xAA
   movwf   PPSLOCK
; Set PPSLOCKED bit to disable writes or
; Clear PPSLOCKED bit to enable writes
   bsf     PPSLOCK,PPSLOCKED
; restore interrupts
   bsf     INTCON,GIE
```

12.5 PPS Permanent Lock

The PPS can be permanently locked by setting the PPS1WAY Configuration bit. When this bit is set, the PPSLOCKED bit can only be cleared and set one time after a device Reset. This allows for clearing the PPSLOCKED bit so that the input and output selections can be made during initialization. When the PPSLOCKED bit is set after all selections have been made, it will remain set and cannot be cleared until after the next device Reset event.

12.6 Operation During Sleep

PPS input and output selections are unaffected by Sleep.

12.7 Effects of a Reset

A device Power-on Reset (POR) clears all PPS input and output selections to their default values. All other Resets leave the selections unchanged. Default input selections are shown in Table 12-1.

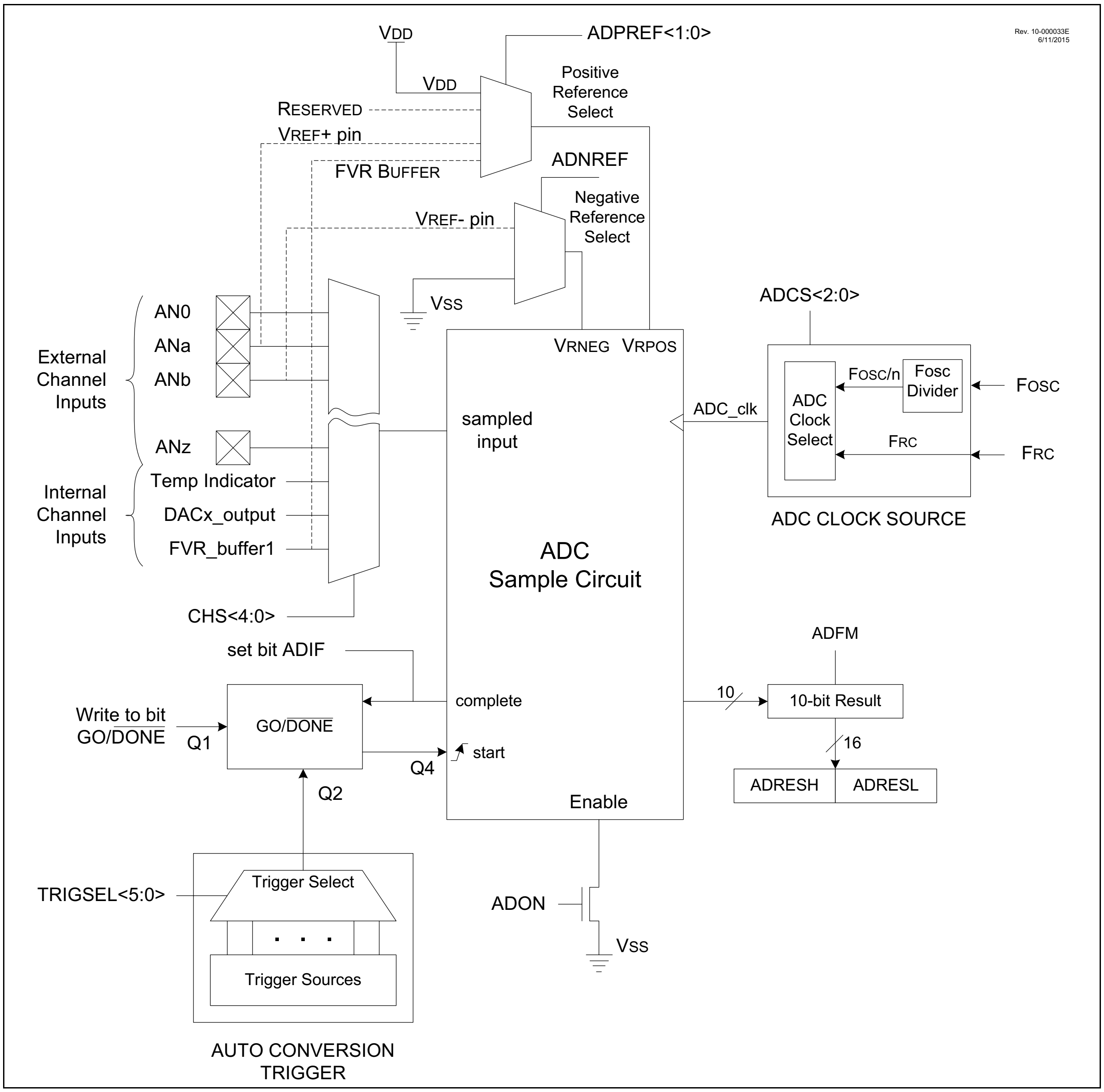
16.0 ANALOG-TO-DIGITAL CONVERTER (ADC) MODULE

The Analog-to-Digital Converter (ADC) allows conversion of an analog input signal to a 10-bit binary representation of that signal. This device uses analog inputs, which are multiplexed into a single sample and hold circuit. The output of the sample and hold is connected to the input of the converter. The converter generates a 10-bit binary result via successive approximation and stores the conversion result into the ADC result registers (ADRESH:ADRESL register pair). Figure 16-1 shows the block diagram of the ADC.

The ADC voltage reference is software selectable to be either internally generated or externally supplied.

The ADC can generate an interrupt upon completion of a conversion. This interrupt can be used to wake-up the device from Sleep.

FIGURE 16-1: ADC BLOCK DIAGRAM



18.7 Register Definitions: DAC Control

Long bit name prefixes for the 10-bit DAC peripherals are shown in Table 18-2. Refer to **Section 1.1 "Register and Bit naming conventions"** for more information

TABLE 18-2:

Peripheral	Bit Name Prefix
DAC1	DAC1
DAC2	DAC2
DAC5	DAC5
DAC6(1)	DAC6

Note 1: PIC16(L)F1777/9 only.

REGISTER 18-1: DACxCON0: DAC CONTROL REGISTER 0

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	FM	OE1	OE2	PSS<1:0>		NSS<1:0>	
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **EN:** DAC Enable bit
- 1 = DACx is enabled
- 0 = DACx is disabled

bit 6 **FM:** DAC Reference Format bit
- 1 = DACx reference selection is left justified
- 0 = DACx reference selection is right justified

bit 5 **OE1:** DAC Voltage Output Enable bit
- 1 = DACx voltage level is also an output on the DACxOUT1 pin
- 0 = DACx voltage level is disconnected from the DACxOUT1 pin

bit 4 **OE2:** DAC Voltage Output Enable bit
- 1 = DACx voltage level is also an output on the DACxOUT2 pin
- 0 = DACx voltage level is disconnected from the DACxOUT2 pin

bit 3-2 **PSS<1:0>:** DAC Positive Source Select bits
- 11 = DACxREF1+ (DAC5/6) or Reserved (DAC1/2)
- 10 = FVR_buffer2
- 01 = DACxREF0+
- 00 = VDD

bit 1-0 **NSS<1:0>:** DAC Negative Source Select bit
- 11 = Reserved. Do not use.
- 10 = DACxREF1- (DAC5/6) or Reserved (DAC1/2)
- 01 = DACxREF0-
- 00 = AGND (AVSS)

23.6 Operation Examples

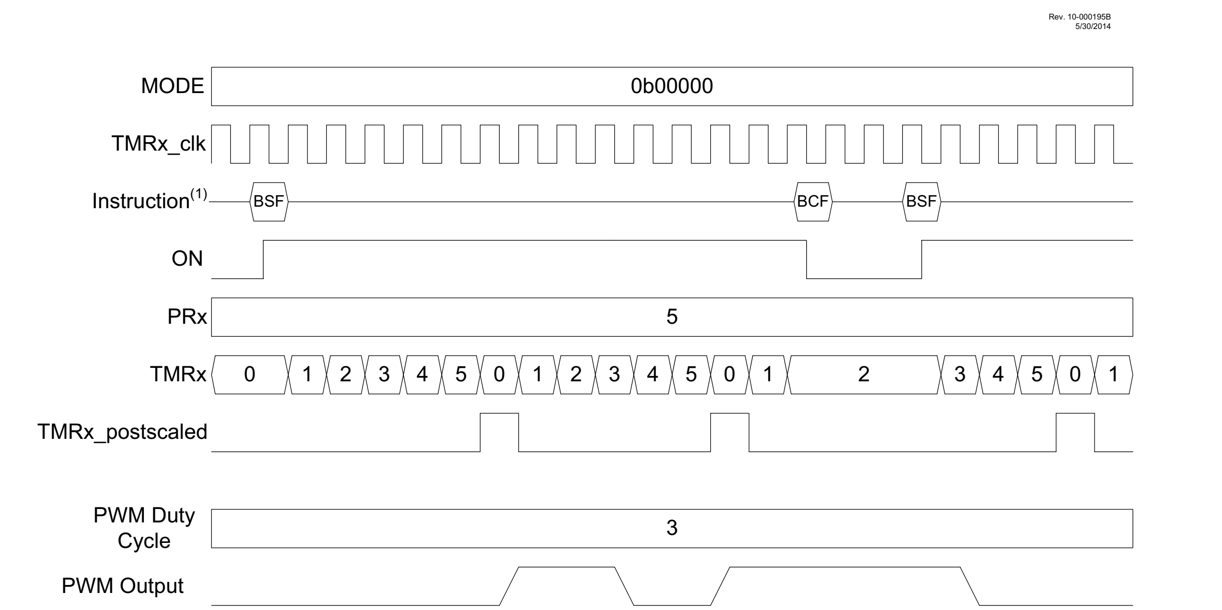
Unless otherwise specified, the following notes apply to the following timing diagrams:

- Both the prescaler and postscaler are set to 1:1 (both the CKPS and OUTPS bits in the TxCON register are cleared).
- The diagrams illustrate any clock except Fosc/4 and show clock-sync delays of at least two full cycles for both ON and Timer2_ers. When using Fosc/4, the clock-sync delay is at least one instruction period for Timer2_ers; ON applies in the next instruction period.
- The PWM Duty Cycle and PWM output are illustrated assuming that the timer is used for the PWM function of the CCP module as described in **Section 24.6 “CCP/PWM Clock Selection”**. The signals are not a part of the Timer2 module.

23.6.1 SOFTWARE GATE MODE

This mode corresponds to legacy Timer2 operation. The timer increments with each clock input when ON = `1` and does not increment when ON = `0`. When the TMRx count equals the PRx period count the timer resets on the next clock and continues counting from 0. Operation with the ON bit software controlled is illustrated in Figure 23-4. With PRx = 5, the counter advances until TMRx = 5, and goes to zero with the next clock.

FIGURE 23-4: SOFTWARE GATE MODE TIMING DIAGRAM (MODE = `00000`)



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

23.6.8 LEVEL RESET, EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODES

In Level -Triggered One-Shot mode the timer count is reset on the external signal level and starts counting on the rising/falling edge of the transition from Reset level to the active level while the ON bit is set. Reset levels are selected as follows:

- Low Reset level (MODE<4:0> = `01110`)
- High Reset level (MODE<4:0> = `01111`)

When the timer count matches the PRx period count, the timer is reset and the ON bit is cleared. When the ON bit is cleared by either a PRx match or by software control a new external signal edge is required after the ON bit is set to start the counter.

When Level-Triggered Reset One-Shot mode is used in conjunction with the CCP PWM operation the PWM drive goes active with the external signal edge that starts the timer. The PWM drive goes inactive when the timer count equals the CCPRx pulse width count. The PWM drive does not go active when the timer count clears at the PRx period count match.

24.4 CCP/PWM Clock Selection

The PIC16(L)F1777/8/9 allows each individual CCP and PWM module to select the timer source that controls the module. Each module has an independent selection.

As there are up to four 8-bit timers with auto-reload (Timer2/4/6/8). The PWM mode on the CCP and PWM modules can use any of these timers.

The CCPTMRS register is used to select which timer is used.

24.4.1 USING THE TMR2/4/6/8 WITH THE CCP MODULE

This device has a new version of the TMR2 module that has many new modes, which allow for greater customization and control of the PWM signals than older parts. Refer to **Section 23.6 "Operation Examples"** for examples of PWM signal generation using the different modes of Timer2. The CCP operation requires that the timer used as the PWM time base has the FOSC/4 clock source selected.

24.4.2 PWM PERIOD

The PWM period is specified by the T2PR/T4PR/T6PR/T8PR register of Timer2/4/6/8. The PWM period can be calculated using the formula of Equation 24-1.

EQUATION 24-1: PWM PERIOD

$$PWM\ Period = [(PR2) + 1] \bullet 4 \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

Note 1: TOSC = 1/FOSC

When TMR2/4/6/8 is equal to its respective T2PR/T4PR/T6PR/T8PR register, the following three events occur on the next increment cycle:

- TMR2/4/6/8 is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from the CCPRxH:CCPRxL pair into the internal 10-bit latch.

Note:	The Timer postscaler (see Figure 24-1) is not used in the determination of the PWM frequency.

24.4.3 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to two registers: the CCPRxH:CCPRxL register pair. Where the particular bits go is determined by the FMT bit of the CCPxCON register. If FMT = 0, the two Most Significant bits of the duty cycle value should be written to bits <1:0> of the CCPRxH register and the remaining eight bits to the CCPRxL register. If FMT = 1, the Least Significant two bits of the duty cycle should be written to bits <7:6> of the CCPRxL register and the Most Significant eight bits to the CCPRxH register. This is illustrated in Figure 24-4. These bits can be written at any time. The duty cycle value is not latched into the internal latch until after the period completes (i.e., a match between T2PR/T4PR/T6PR/T8PR and TMR2/4/6/8 registers occurs).

Equation 24-2 is used to calculate the PWM pulse width. Equation 24-3 is used to calculate the PWM duty cycle ratio.

EQUATION 24-2: PULSE WIDTH

$$Pulse\ Width = CCPRxH:CCPRxL \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

EQUATION 24-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPRxH:CCPRxL)}{4(PRx + 1)}$$

The PWM duty cycle registers are double buffered for glitchless PWM operation.

The 8-bit timer TMR2/4/6/8 register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2/4/6/8 prescaler is set to 1:1.

When the 10-bit time base matches the internal buffer register, then the CCPx pin is cleared (see Figure 24-3).

FIGURE 24-4: CCPx DUTY CYCLE ALIGNMENT

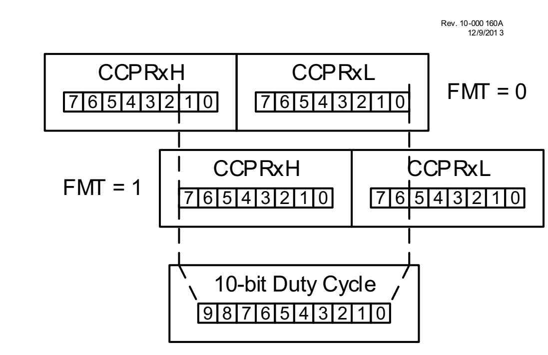


FIGURE 30-3: ALTERNATING RISING/FALLING RAMP GENERATION TIMING DIAGRAM (OS = 0, MODE = 01)

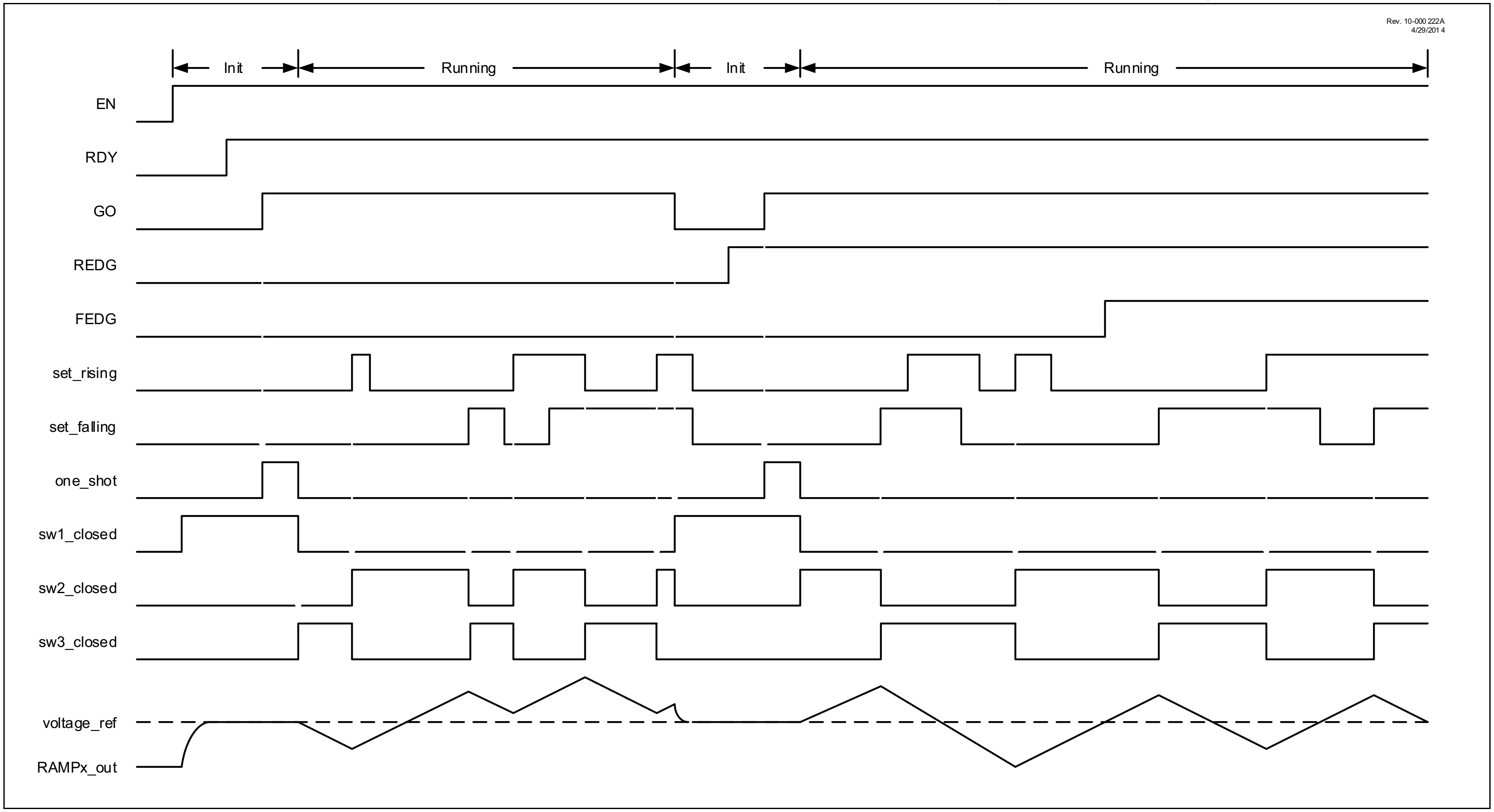


FIGURE 32-9: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

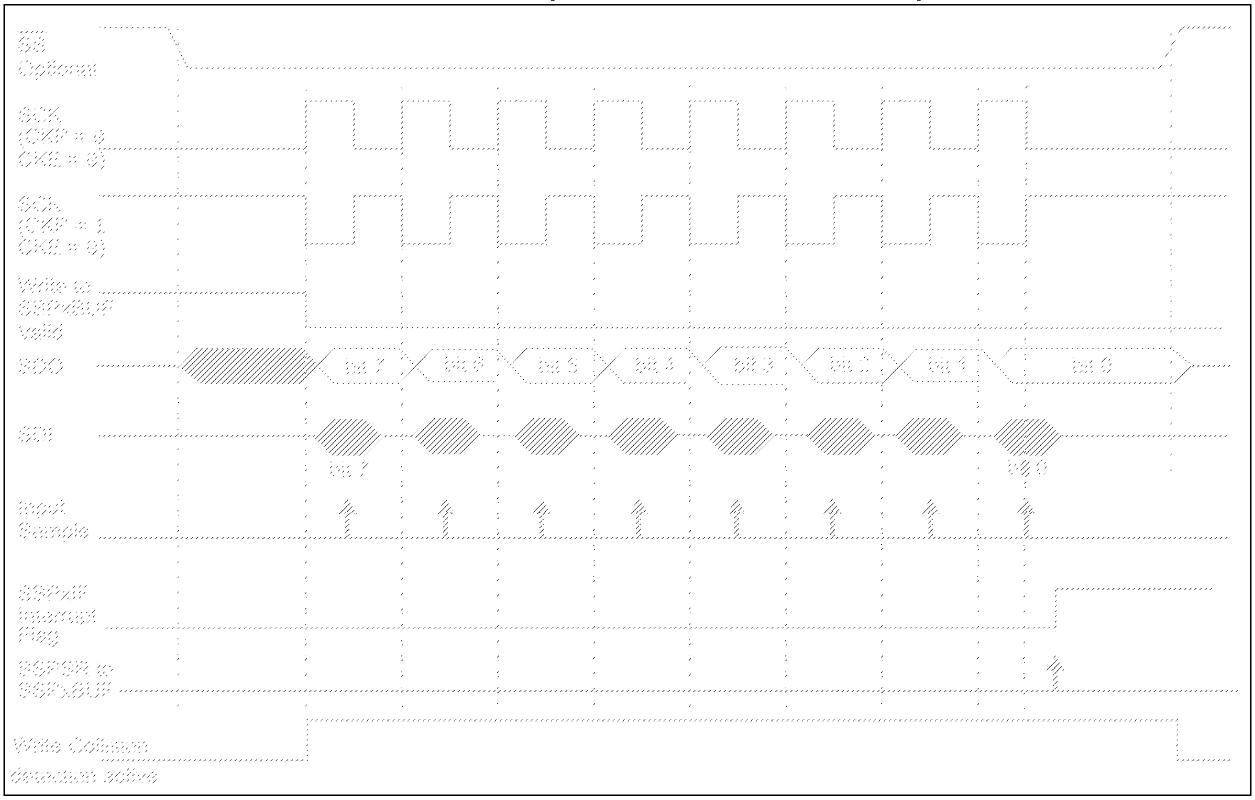
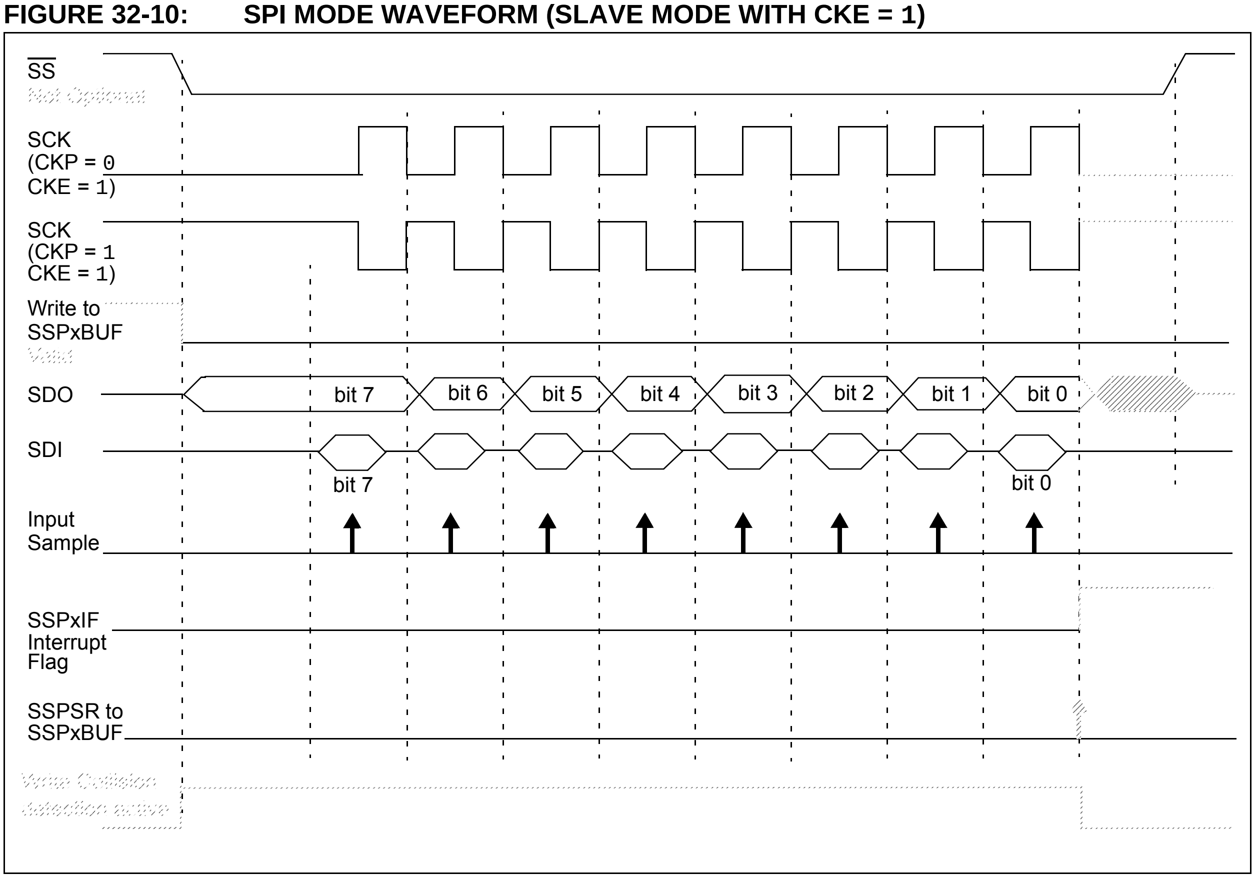


FIGURE 32-10: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



33.4 EUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit or 16-bit timer that is dedicated to the support of both the asynchronous and synchronous EUSART operation. By default, the BRG operates in 8-bit mode. Setting the BRG16 bit of the BAUDxCON register selects 16-bit mode.

The SPxBRGH:SPxBRGL register pair determines the period of the free running baud rate timer. In Asynchronous mode the multiplier of the baud rate period is determined by both the BRGH bit of the TXxSTA register and the BRG16 bit of the BAUDxCON register. In Synchronous mode, the BRGH bit is ignored.

Table 33-3 contains the formulas for determining the baud rate. Example 33-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various Asynchronous modes have been computed for your convenience and are shown in Table 33-5. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG (BRG16 = 1) to reduce the baud rate error. The 16-bit BRG mode is used to achieve slow baud rates for fast oscillator frequencies.

Writing a new value to the SPxBRGH:SPxBRGL register pair causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

If the system clock is changed during an active receive operation, a receive error or data loss may result. To avoid this problem, check the status of the RCIDL bit to make sure that the receive operation is idle before changing the system clock.

EXAMPLE 33-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

$$\text{Desired Baud Rate} = \frac{F_{OSC}}{64([SPBRGH:SPBRGL] + 1)}$$

Solving for SPxBRGH:SPxBRGL:

$$X = \frac{\frac{F_{OSC}}{\text{Desired Baud Rate}}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25$$

$$\text{Calculated Baud Rate} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\text{Error} = \frac{\text{Calc. Baud Rate} - \text{Desired Baud Rate}}{\text{Desired Baud Rate}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

35.0 INSTRUCTION SET SUMMARY

Each instruction is a 14-bit word containing the operation code (opcode) and all required operands. The opcodes are broken into three broad categories.

- Byte Oriented
- Bit Oriented
- Literal and Control

The literal and control category contains the most varied instruction word format.

Table 35-3 lists the instructions recognized by the MPASM™ assembler.

All instructions are executed within a single instruction cycle, with the following exceptions, which may take two or three cycles:

- Subroutine takes two cycles (`CALL`, `CALLW`)
- Returns from interrupts or subroutines take two cycles (`RETURN`, `RETLW`, `RETFIE`)
- Program branching takes two cycles (`GOTO`, `BRA`, `BRW`, `BTFSS`, `BTFSC`, `DECFSZ`, `INCSFZ`)
- One additional instruction cycle will be used when any instruction references an indirect file register and the file select register is pointing to program memory.

One instruction cycle consists of four oscillator cycles; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution rate of 1 MHz.

All instruction examples use the format '`0xhh`' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

35.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

TABLE 35-1: 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`.
n	FSR or INDF number. (0-1)
mm	Pre-post increment-decrement mode selection

TABLE 35-2: ABBREVIATION DESCRIPTIONS

Field	Description
PC	Program Counter
$\overline{\text{TO}}$	Time-Out bit
C	Carry bit
DC	Digit Carry bit
Z	Zero bit
$\overline{\text{PD}}$	Power-Down bit

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

BRA	Relative Branch
Syntax:	[*label*] BRA label [*label*] BRA $+k
Operands:	$-256 \leq$ label - PC + 1 ≤ 255 $-256 \leq k \leq 255$
Operation:	(PC) + 1 + k $\rightarrow$ PC
Status Affected:	None
Description:	Add the signed 9-bit literal ‘k’ to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + k. This instruction is a 2-cycle instruction. This branch has a limited range.

BRW	Relative Branch with W
Syntax:	[*label*] BRW
Operands:	None
Operation:	(PC) + (W) $\rightarrow$ PC
Status Affected:	None
Description:	Add the contents of W (unsigned) to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + (W). This instruction is a 2-cycle instruction.

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

BTFSC	Bit Test f, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	$0 \leq f \leq 127$ $0 \leq b \leq 7$
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit ‘b’ in register ‘f’ is ‘1’, the next instruction is executed. 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.

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<b>) = 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-cycle instruction.

SWAPF — Swap Nibbles in f

Syntax:	[*label*] SWAPF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f<3:0>) $\rightarrow$ (destination<7:4>), (f<7:4>) $\rightarrow$ (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'.

TRIS — Load TRIS Register with W

Syntax:	[*label*] TRIS f
Operands:	$5 \leq f \leq 7$
Operation:	(W) $\rightarrow$ TRIS register 'f'
Status Affected:	None
Description:	Move data from W register to TRIS register. When 'f' = 5, TRISA is loaded. When 'f' = 6, TRISB is loaded. When 'f' = 7, TRISC is loaded.

XORLW — Exclusive OR literal with W

Syntax:	[*label*] XORLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) .XOR. k $\rightarrow$ (W)
Status Affected:	Z
Description:	The contents of the W register are XOR'ed with the 8-bit literal 'k'. The result is placed in the W register.

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) $\rightarrow$ (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'.

Note: Unless otherwise noted, VIN = 5V, FOSC = 300 kHz, CIN = 0.1 µF, TA = 25°C.

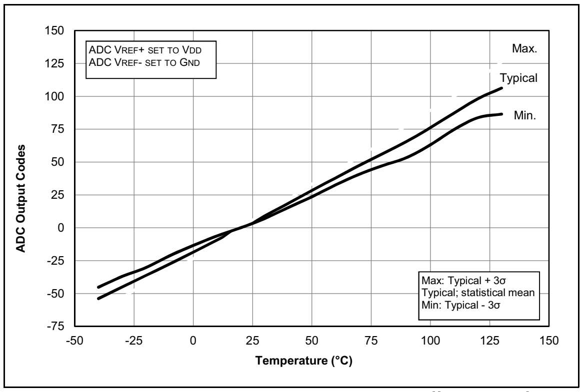


FIGURE 37-91: *Temperature Indicator Slope Normalized to 20°C, Low Range, VDD = 3.0V, PIC16F1777/8/9 only.*

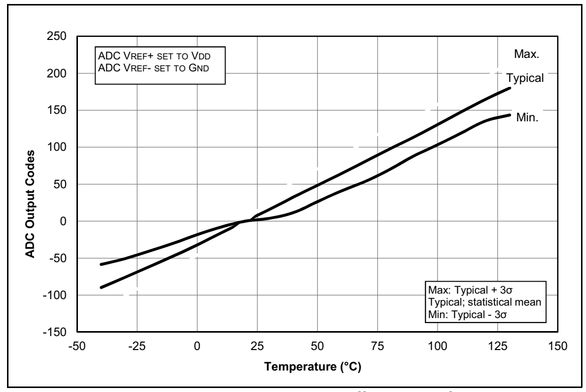


FIGURE 37-92: *Temp. Indicator Slope Normalized to 20°C, Low Range, VDD = 1.8V, PIC16LF1773/6 Only.*

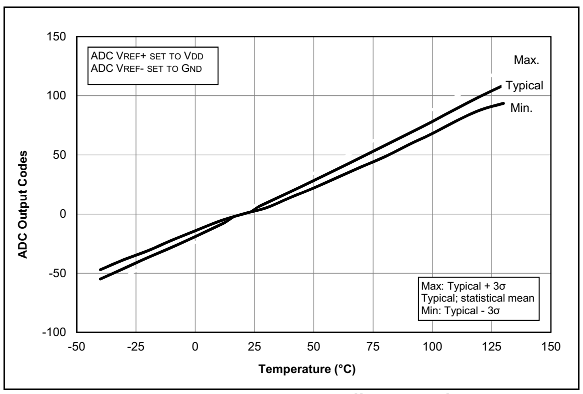


FIGURE 37-93: *Temp. Indicator Slope Normalized to 20°C, Low Range, VDD = 3.0V, PIC16LF1773/6 Only.*

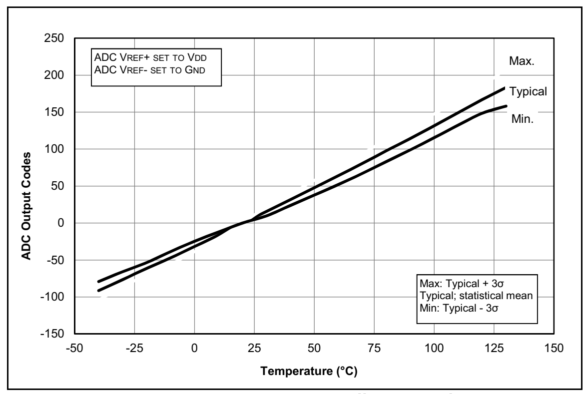


FIGURE 37-94: *Temp. Indicator Slope Normalized to 20°C, High Range, VDD = 3.6V, PIC16LF1773/6 Only.*

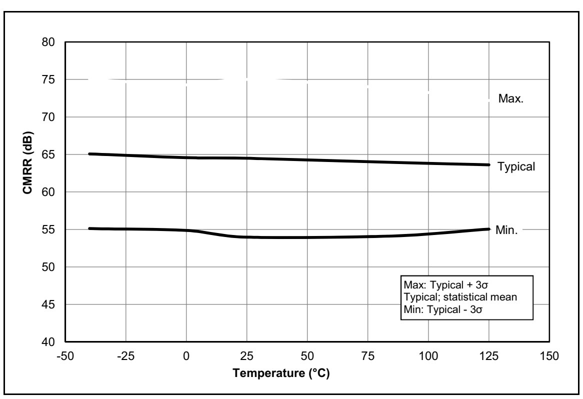


FIGURE 37-95: *Op Amp, Common Mode Rejection Ratio (CMRR), VDD = 3.0V.*

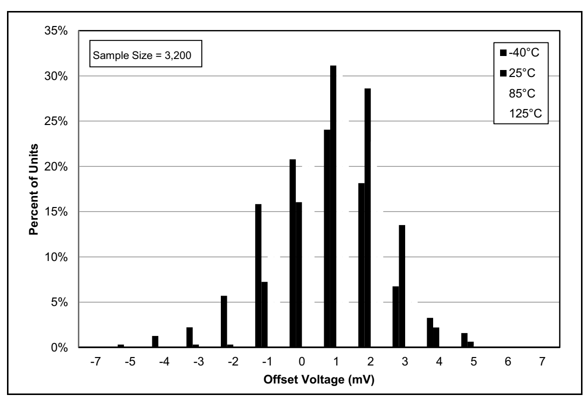


FIGURE 37-96: *Op Amp, Offset Voltage Histogram, VDD = 3.0V, VCM = VDD/2.*

Note: Unless otherwise noted, V_{IN} = 5V, F_{OSC} = 300 kHz, C_{IN} = 0.1 µF, T_A = 25°C.

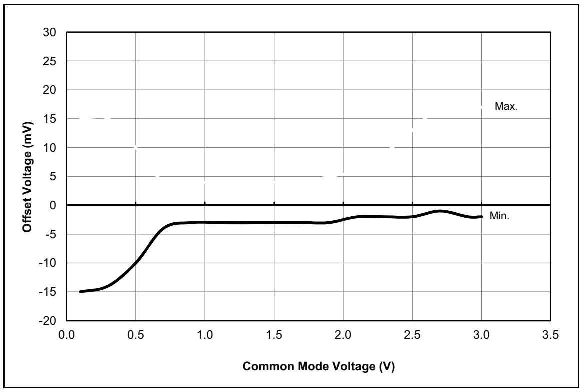


FIGURE 37-103: *Comparator Offset, NP Mode (CxSP = 1), V_{DD} = 3.0V, Typical Measured Values at 25°C.*

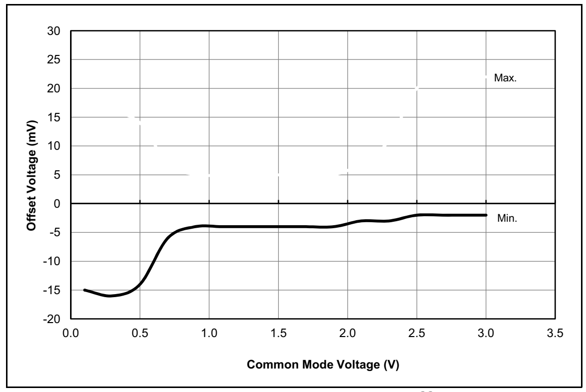


FIGURE 37-104: *Comparator Offset, NP Mode (CxSP = 1), V_{DD} = 3.0V, Typical Measured Values from -40°C to 125°C.*

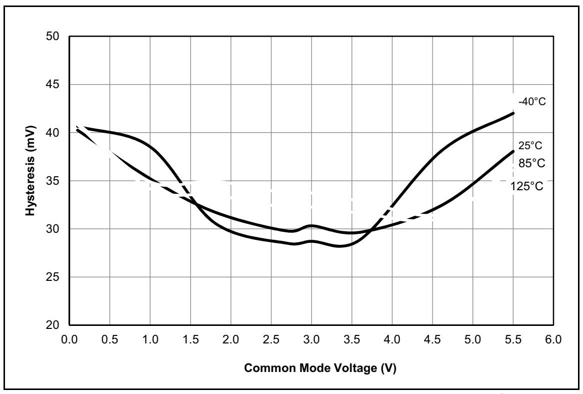


FIGURE 37-105: *Comparator Hysteresis, NP Mode (CxSP = 1), V_{DD} = 5.5V, Typical Measured Values, PIC16F1777/8/9 only.*

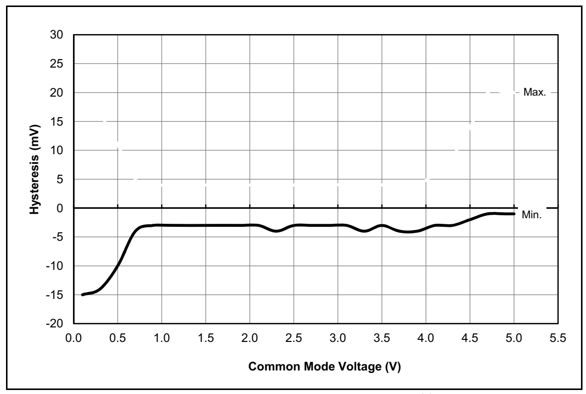


FIGURE 37-106: *Comparator Offset, NP Mode (CxSP = 1), V_{DD} = 5.0V, Typical Measured Values at 25°C, PIC16F1777/8/9 only.*

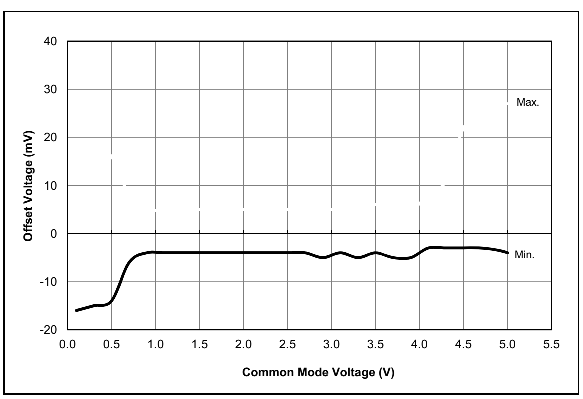


FIGURE 37-107: *Comparator Offset, NP Mode (CxSP = 1), V_{DD} = 5.5V, Typical Measured Values from -40°C to 125°C, PIC16F1777/8/9 only.*

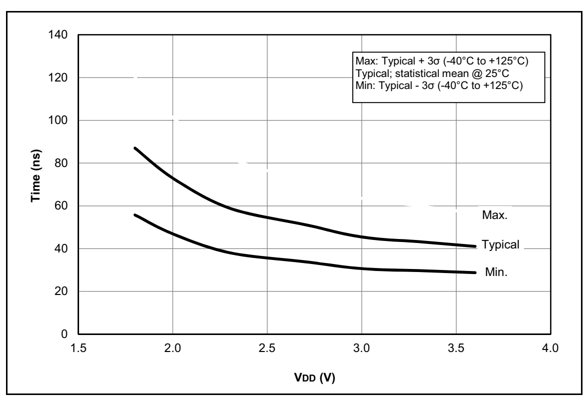
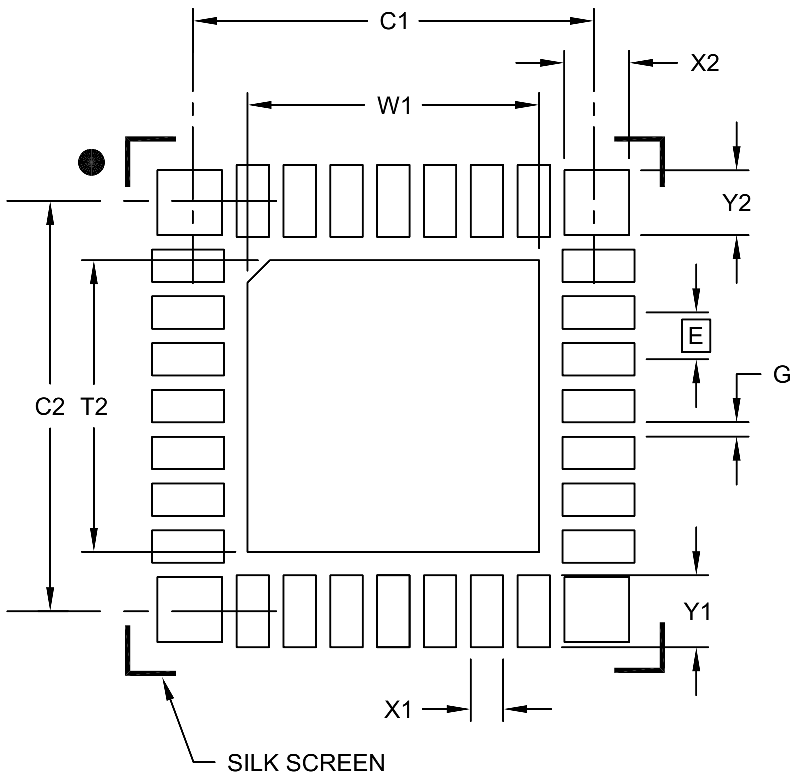


FIGURE 37-108: *Comparator Response Time Over Voltage, NP Mode (CxSP = 1), Typical Measured Values.*

28-Lead Plastic Quad Flat, No Lead Package (MX) - 6x6 mm Body [UQFN] With 0.60mm Contact Length And Corner Anchors

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



RECOMMENDED LAND PATTERN

	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	W1			4.05
Optional Center Pad Length	T2			4.05
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.00
Corner Pad Width (X4)	X2			0.90
Corner Pad Length (X4)	Y2			0.90
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2209B