

Welcome to [E-XFL.COM](http://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	32MHz
Connectivity	I <sup>2</sup> C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 17x10b; D/A 1x5b, 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic16f1713-i-so">https://www.e-xfl.com/product-detail/microchip-technology/pic16f1713-i-so</a>

# PIC16(L)F1713/6

## 3.4.5 CORE FUNCTION REGISTERS SUMMARY

The Core Function registers listed in Table 3-10 can be addressed from any Bank.

**TABLE 3-10: CORE FUNCTION REGISTERS SUMMARY**

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets	
<b>Bank 0-31</b>												
x00h or x80h	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu	
x01h or x81h	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu	
x02h or x82h	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000	
x03h or x83h	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
x04h or x84h	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu	
x05h or x85h	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000	
x06h or x86h	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu	
x07h or x87h	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000	
x08h or x88h	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000	
x09h or x89h	WREG	Working Register								0000 0000	uuuu uuuu	
x0Ah or x8Ah	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter								-000 0000	-000 0000
x0Bh or x8Bh	INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	0000 0000	0000 0000	

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

**Note 1:** These registers can be addressed from any bank.

# PIC16(L)F1713/6

## 6.2.1.4 4x PLL

The oscillator module contains a 4x PLL that can be used with both external and internal clock sources to provide a system clock source. The input frequency for the 4x PLL must fall within specifications. See the PLL Clock Timing Specifications in Table 34-9.

The 4x PLL may be enabled for use by one of two methods:

1. Program the PLEN bit in Configuration Words to a '1'.
2. Write the SPLLEN bit in the OSCCON register to a '1'. If the PLEN bit in Configuration Words is programmed to a '1', then the value of SPLLEN is ignored.

## 6.2.1.5 Secondary Oscillator

The secondary oscillator is a separate crystal oscillator that is associated with the Timer1 peripheral. It is optimized for timekeeping operations with a 32.768 kHz crystal connected between the SOSCO and SOSCI device pins.

The secondary oscillator can be used as an alternate system clock source and can be selected during run-time using clock switching. Refer to **Section 6.3 "Clock Switching"** for more information.

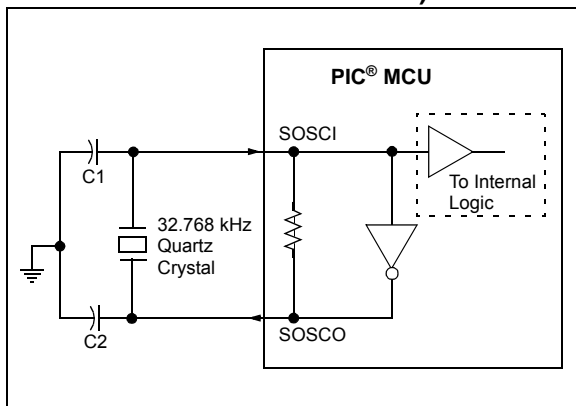
**Note 1:** Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

**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 rPIC® and PIC® Devices" (DS00826)
- AN849, "Basic PIC® Oscillator Design" (DS00849)
- AN943, "Practical PIC® Oscillator Analysis and Design" (DS00943)
- AN949, "Making Your Oscillator Work" (DS00949)
- TB097, "Interfacing a Micro Crystal MS1V-T1K 32.768 kHz Tuning Fork Crystal to a PIC16F690/SS" (DS91097)
- AN1288, "Design Practices for Low-Power External Oscillators" (DS01288)

**FIGURE 6-5: QUARTZ CRYSTAL OPERATION (SECONDARY OSCILLATOR)**



## 6.2.2.5 Internal Oscillator Frequency Selection

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits  $IRCF<3:0>$  of the  $OSCCON$  register.

The postscaled output of the 16 MHz HFINTOSC, 500 kHz MFINTOSC, and 31 kHz LFINTOSC connect to a multiplexer (see Figure 6-1). The Internal Oscillator Frequency Select bits  $IRCF<3:0>$  of the  $OSCCON$  register select the frequency output of the internal oscillators. One of the following frequencies can be selected via software:

- 32 MHz (requires 4x PLL)
- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz
- 500 kHz (default after Reset)
- 250 kHz
- 125 kHz
- 62.5 kHz
- 31.25 kHz
- 31 kHz (LFINTOSC)

**Note:** Following any Reset, the  $IRCF<3:0>$  bits of the  $OSCCON$  register are set to '0111' and the frequency selection is set to 500 kHz. The user can modify the  $IRCF$  bits to select a different frequency.

The  $IRCF<3:0>$  bits of the  $OSCCON$  register allow duplicate selections for some frequencies. These duplicate choices can offer system design trade-offs. Lower power consumption can be obtained when changing oscillator sources for a given frequency. Faster transition times can be obtained between frequency changes that use the same oscillator source.

## 6.2.2.6 32 MHz Internal Oscillator Frequency Selection

The Internal Oscillator Block can be used with the 4x PLL associated with the External Oscillator Block to produce a 32 MHz internal system clock source. The following settings are required to use the 32 MHz internal clock source:

- The  $FOSC$  bits in Configuration Words must be set to use the INTOSC source as the device system clock ( $FOSC<2:0> = 100$ ).
- The  $SCS$  bits in the  $OSCCON$  register must be cleared to use the clock determined by  $FOSC<2:0>$  in Configuration Words ( $SCS<1:0> = 00$ ).
- The  $IRCF$  bits in the  $OSCCON$  register must be set to the 8 MHz HFINTOSC set to use ( $IRCF<3:0> = 1110$ ).
- The  $SPLLEN$  bit in the  $OSCCON$  register must be set to enable the 4x PLL, or the  $PLLEN$  bit of the Configuration Words must be programmed to a '1'.

**Note:** When using the  $PLLEN$  bit of the Configuration Words, the 4x PLL cannot be disabled by software and the  $SPLLEN$  option will not be available.

The 4x PLL is not available for use with the internal oscillator when the  $SCS$  bits of the  $OSCCON$  register are set to '1x'. The  $SCS$  bits must be set to '00' to use the 4x PLL with the internal oscillator.

## 6.6 Register Definitions: Oscillator Control

### REGISTER 6-1: OSCCON: OSCILLATOR CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
SPLLEN	IRCF<3:0>			—	SCS<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      **SPLLEN:** Software PLL Enable bit  
If PLEN in Configuration Words = 1:  
 SPLLEN bit is ignored. 4x PLL is always enabled (subject to oscillator requirements)  
If PLEN in Configuration Words = 0:  
 1 = 4x PLL is enabled  
 0 = 4x PLL is disabled
- bit 6-3    **IRCF<3:0>:** Internal Oscillator Frequency Select bits  
 1111 = 16 MHz HF  
 1110 = 8 MHz or 32 MHz HF<sup>(2)</sup>  
 1101 = 4 MHz HF  
 1100 = 2 MHz HF  
 1011 = 1 MHz HF  
 1010 = 500 kHz HF<sup>(1)</sup>  
 1001 = 250 kHz HF<sup>(1)</sup>  
 1000 = 125 kHz HF<sup>(1)</sup>  
 0111 = 500 kHz MF (default upon Reset)  
 0110 = 250 kHz MF  
 0101 = 125 kHz MF  
 0100 = 62.5 kHz MF  
 0011 = 31.25 kHz HF<sup>(1)</sup>  
 0010 = 31.25 kHz MF  
 000x = 31 kHz LF
- bit 2      **Unimplemented:** Read as '0'
- bit 1-0    **SCS<1:0>:** System Clock Select bits  
 1x = Internal oscillator block  
 01 = Secondary oscillator  
 00 = Clock determined by FOSC<2:0> in Configuration Words
- Note 1:** Duplicate frequency derived from HFINTOSC.  
**Note 2:** 32 MHz when SPLLEN bit is set. Refer to **Section 6.2.2.6 “32 MHz Internal Oscillator Frequency Selection”**.

# PIC16(L)F1713/6

## REGISTER 6-2: OSCSTAT: OSCILLATOR STATUS REGISTER

R-1/q	R-0/q	R-q/q	R-0/q	R-0/q	R-q/q	R-0/0	R-0/q
SOSCR	PLL R	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS
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	q = Conditional

- bit 7      **SOSCR:** Secondary Oscillator Ready bit  
If T1OSCEN = 1:  
 1 = Secondary oscillator is ready  
 0 = Secondary oscillator is not ready  
If T1OSCEN = 0:  
 1 = Secondary clock source is always ready
- bit 6      **PLL R** 4x PLL Ready bit  
 1 = 4x PLL is ready  
 0 = 4x PLL is not ready
- bit 5      **OSTS:** Oscillator Start-up Timer Status bit  
 1 = Running from the clock defined by the FOSC<2:0> bits of the Configuration Words  
 0 = Running from an internal oscillator (FOSC<2:0> = 100)
- bit 4      **HFIOFR:** High-Frequency Internal Oscillator Ready bit  
 1 = HFINTOSC is ready  
 0 = HFINTOSC is not ready
- bit 3      **HFIOFL:** High-Frequency Internal Oscillator Locked bit  
 1 = HFINTOSC is at least 2% accurate  
 0 = HFINTOSC is not 2% accurate
- bit 2      **MFIOFR:** Medium Frequency Internal Oscillator Ready bit  
 1 = MFINTOSC is ready  
 0 = MFINTOSC is not ready
- bit 1      **LFIOFR:** Low-Frequency Internal Oscillator Ready bit  
 1 = LFINTOSC is ready  
 0 = LFINTOSC is not ready
- bit 0      **HFIOFS:** High-Frequency Internal Oscillator Stable bit  
 1 = HFINTOSC is at least 0.5% accurate  
 0 = HFINTOSC is not 0.5% accurate

# PIC16(L)F1713/6

## 12.0 PERIPHERAL PIN SELECT (PPS) MODULE

The Peripheral Pin Select (PPS) module connects peripheral inputs and outputs to the device I/O pins. Only digital signals are included in the selections. All analog inputs and outputs remain fixed to their assigned pins. Input and output selections are independent as shown in the simplified block diagram Figure 12-1.

### 12.1 PPS Inputs

Each peripheral has a PPS register with which the inputs to the peripheral are selected. Inputs include the device pins.

Multiple peripherals can operate from the same source simultaneously. Port reads always return the pin level regardless of peripheral PPS selection. If a pin also has associated analog functions, the ANSEL bit for that pin must be cleared to enable the digital input buffer.

Although every peripheral has its own PPS input selection register, the selections are identical for every peripheral as shown in Register 12-1.

**Note:** The notation “xxx” in the register name is a place holder for the peripheral identifier. For example, CLC1PPS.

### 12.2 PPS Outputs

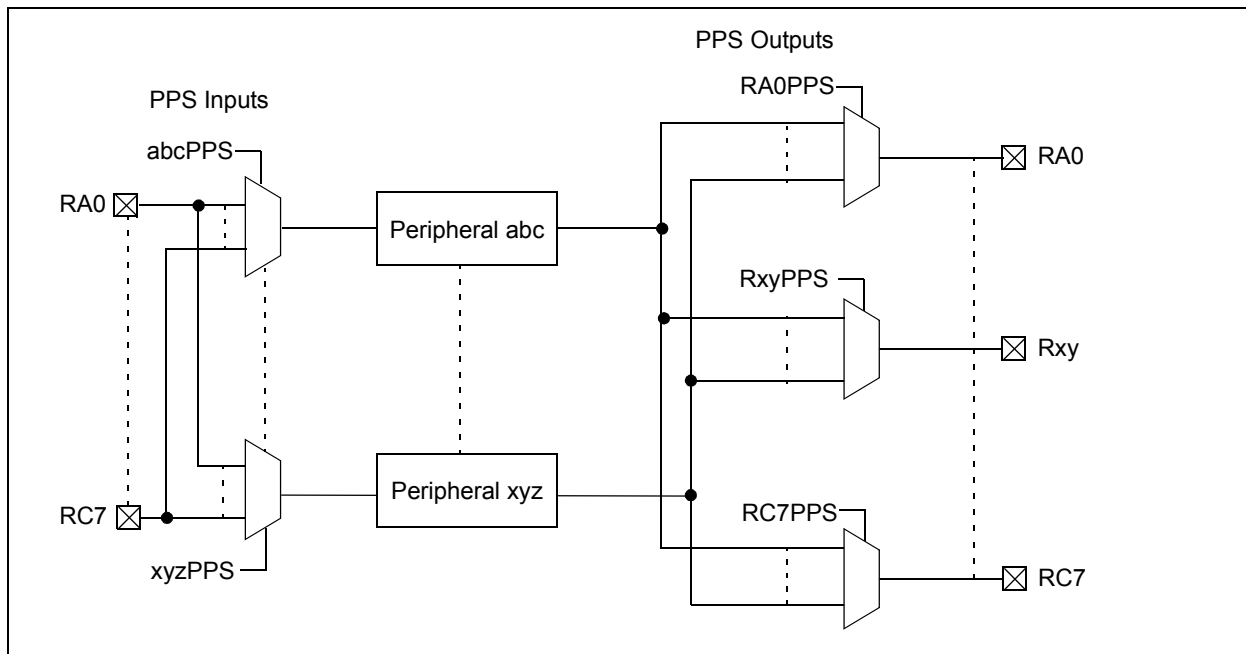
Each I/O pin has a PPS register with which the pin output source is selected. With few exceptions, the port TRIS control associated with that pin retains control over the pin output driver. Peripherals that control the pin output driver as part of the peripheral operation will override the TRIS control as needed. These peripherals include:

- EUSART (synchronous operation)
- MSSP (I<sup>2</sup>C)
- COG (auto-shutdown)

Although every pin has its own PPS peripheral selection register, the selections are identical for every pin as shown in Register 12-2.

**Note:** The notation “Rxy” is a place holder for the pin identifier. For example, RA0PPS.

FIGURE 12-1: SIMPLIFIED PPS BLOCK DIAGRAM



## 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<sup>2</sup>C)

**Note:** The I<sup>2</sup>C default input pins are I<sup>2</sup>C 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 pin allocation Table 1.



# PIC16(L)F1713/6

---

## 13.0 INTERRUPT-ON-CHANGE

All pins on all ports can be configured to operate as Interrupt-on-Change (IOC) pins. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual pin, or combination of pins, can be configured to generate an interrupt. The interrupt-on-change module has the following features:

- Interrupt-on-Change enable (Master Switch)
- Individual pin configuration
- Rising and falling edge detection
- Individual pin interrupt flags

Figure 13-1 is a block diagram of the IOC module.

### 13.1 Enabling the Module

To allow individual pins to generate an interrupt, the IOCIE bit of the INTCON register must be set. If the IOCIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

### 13.2 Individual Pin Configuration

For each pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated bit of the IOCxP register is set. To enable a pin to detect a falling edge, the associated bit of the IOCxN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting the associated bits in both of the IOCxP and IOCxN registers.

## 13.3 Interrupt Flags

The bits located in the IOCxF registers are status flags that correspond to the interrupt-on-change pins of each port. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the INTCON register reflects the status of all IOCxF bits.

### 13.4 Clearing Interrupt Flags

The individual status flags, (IOCxF register bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

#### EXAMPLE 13-1: CLEARING INTERRUPT FLAGS (PORTA EXAMPLE)

```
MOVLW  0xff
XORWF  IOCAF, W
ANDWF  IOCAF, F
```

### 13.5 Operation in Sleep

The interrupt-on-change interrupt sequence will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the affected IOCxF register will be updated prior to the first instruction executed out of Sleep.

# PIC16(L)F1713/6

**REGISTER 16-3: CMOUT: COMPARATOR OUTPUT REGISTER**

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	MC2OUT	MC1OUT
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

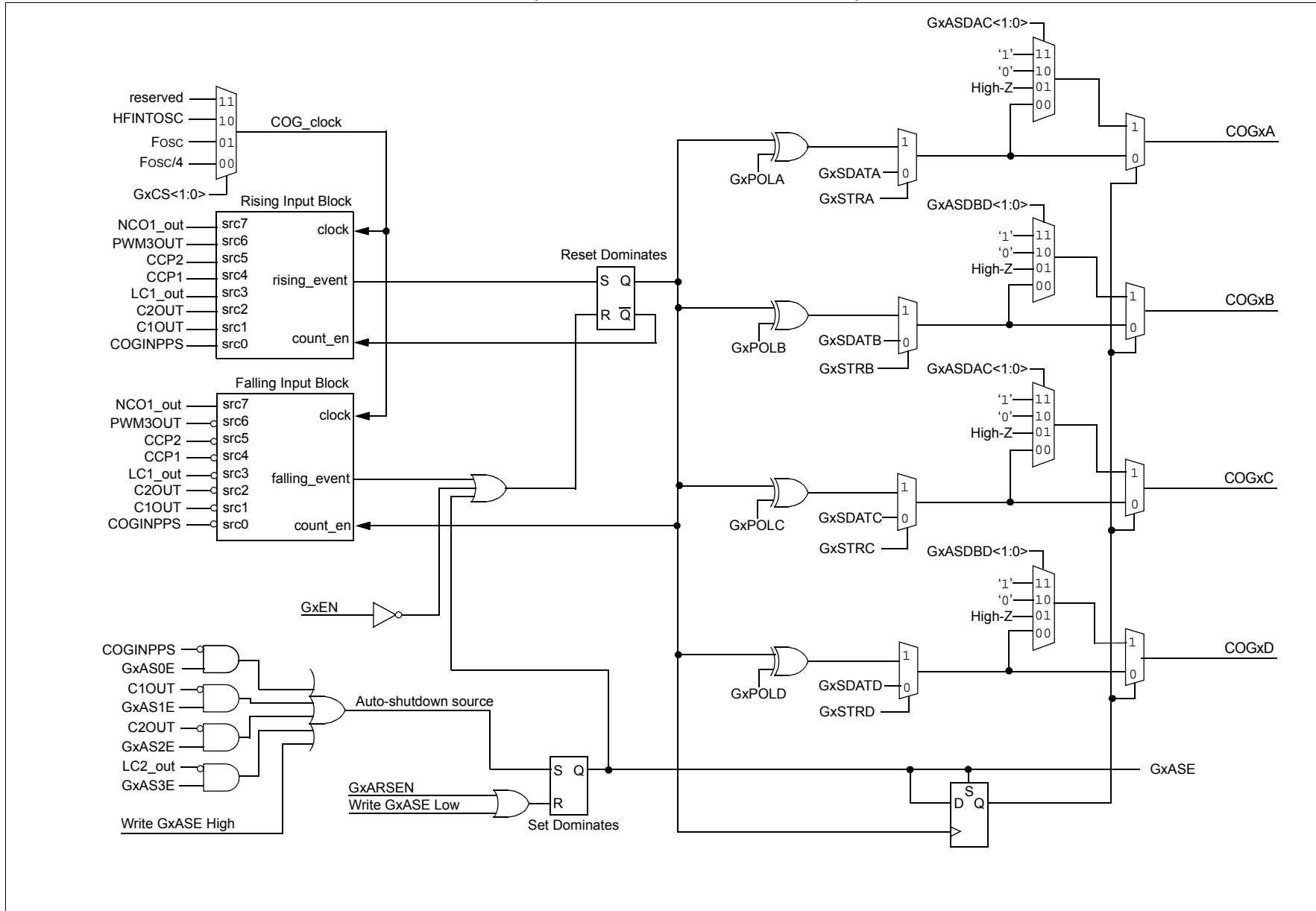
bit 7-2            **Unimplemented:** Read as '0'  
bit 1              **MC2OUT:** Mirror Copy of C2OUT bit  
bit 0              **MC1OUT:** Mirror Copy of C1OUT bit

**TABLE 16-3: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARATOR MODULE**

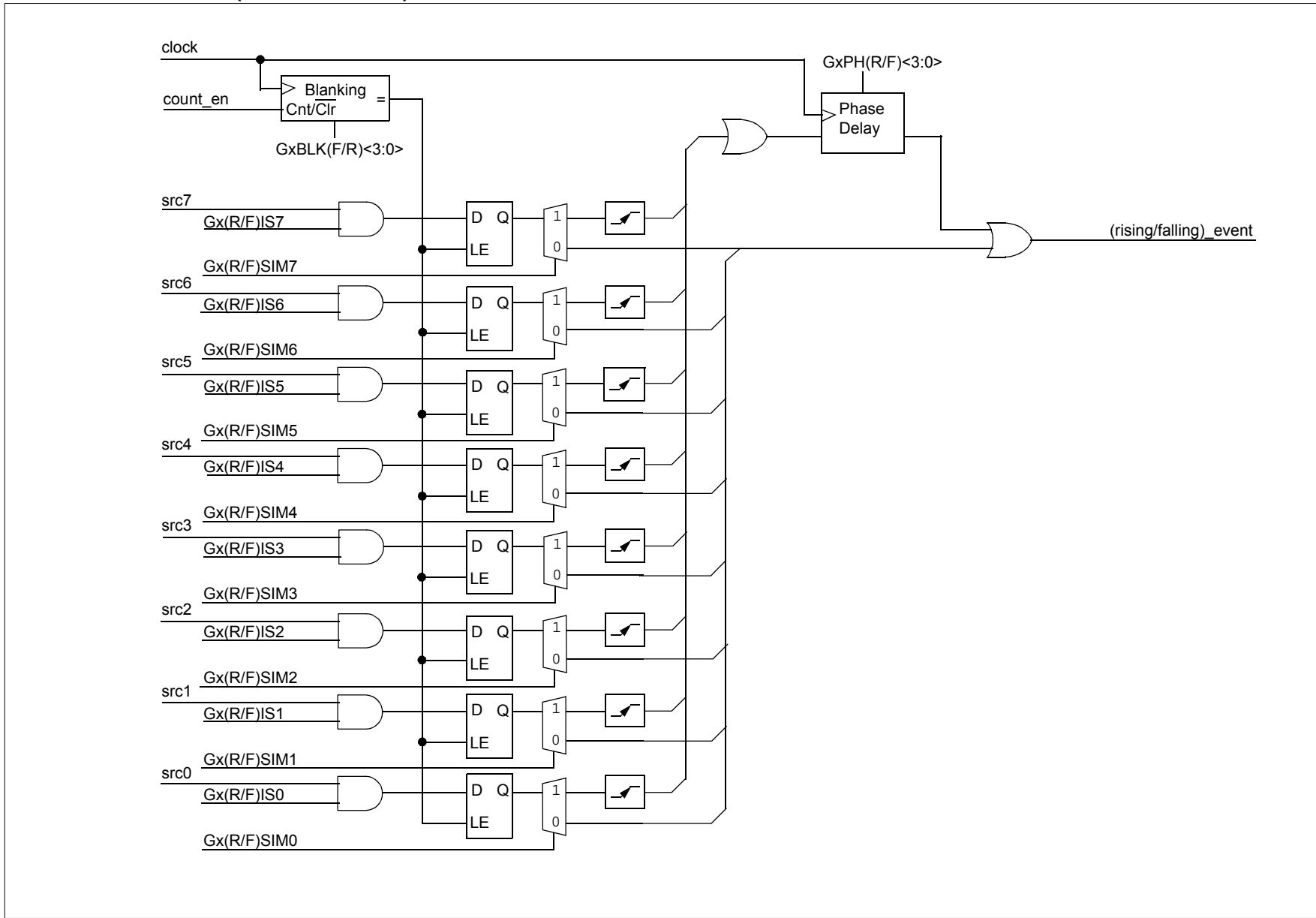
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	120
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	126
CM1CON0	C1ON	C1OUT	—	C1POL	C1ZLF	C1SP	C1HYS	C1SYNC	160
CM2CON0	C2ON	C2OUT	—	C2POL	C2ZLF	C2SP	C2HYS	C2SYNC	160
CM1CON1	C1NTP	C1INTN	C1PCH<2:0>			C1NCH<2:0>			161
CM2CON1	C2NTP	C2INTN	C2PCH<2:0>			C2NCH<2:0>			161
CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	162
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		151
DAC1CON0	DAC1EN	—	DAC1OE1	DAC1OE2	DAC1PSS<1:0>		—	DAC1NSS	249
DAC1CON1	DAC1R<7:0>								249
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	83
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	TMR6IE	TMR4IE	CCP2IE	85
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	TMR6IF	TMR4IF	CCP2IF	88
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	119
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	125
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	130
RxyPPS	—	—	—	RxyPPS<4:0>					137

**Legend:** — = unimplemented location, read as '0'. Shaded cells are unused by the comparator module.

**FIGURE 18-2: SIMPLIFIED COG BLOCK DIAGRAM (STEERED PWM MODE, GXMD = 0)**



**FIGURE 18-7: COG (RISING/FALLING) INPUT BLOCK**



## REGISTER 21-3: ADCON2: ADC CONTROL REGISTER 2

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0
TRIGSEL<3:0> <sup>(1)</sup>				—	—	—	—
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 7-4      **TRIGSEL<3:0>**: Auto-Conversion Trigger Selection bits<sup>(1)</sup>

0000	= No auto-conversion trigger selected
0001	= CCP1
0010	= CCP2
0011	= Timer0 – T0_overflow <sup>(2)</sup>
0100	= Timer1 – T1_overflow <sup>(2)</sup>
0101	= Timer2 – T2_match
0110	= Comparator C1 – sync_C1OUT
0111	= Comparator C2 – sync_C2OUT
1000	= CLC1 – LC1_out
1001	= CLC2 – LC2_out
1010	= CLC3 – LC3_out
1011	= CLC4 – LC4_out
1100	= Timer4 – T4_match
1101	= Timer6 – T6_match
1110	= Reserved
1111	= Reserved

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

**Note 1:** This is a rising edge sensitive input for all sources.

**2:** Signal also sets its corresponding interrupt flag.

## 30.2.5 SLAVE SELECT SYNCHRONIZATION

The Slave Select can also be used to synchronize communication. The Slave Select line is held high until the master device is ready to communicate. When the Slave Select line is pulled low, the slave knows that a new transmission is starting.

If the slave fails to receive the communication properly, it will be reset at the end of the transmission, when the Slave Select line returns to a high state. The slave is then ready to receive a new transmission when the Slave Select line is pulled low again. If the Slave Select line is not used, there is a risk that the slave will eventually become out of sync with the master. If the slave misses a bit, it will always be one bit off in future transmissions. Use of the Slave Select line allows the slave and master to align themselves at the beginning of each transmission.

The  $\overline{SS}$  pin allows a Synchronous Slave mode. The SPI must be in Slave mode with  $\overline{SS}$  pin control enabled ( $SSPCON1<3:0> = 0100$ ).

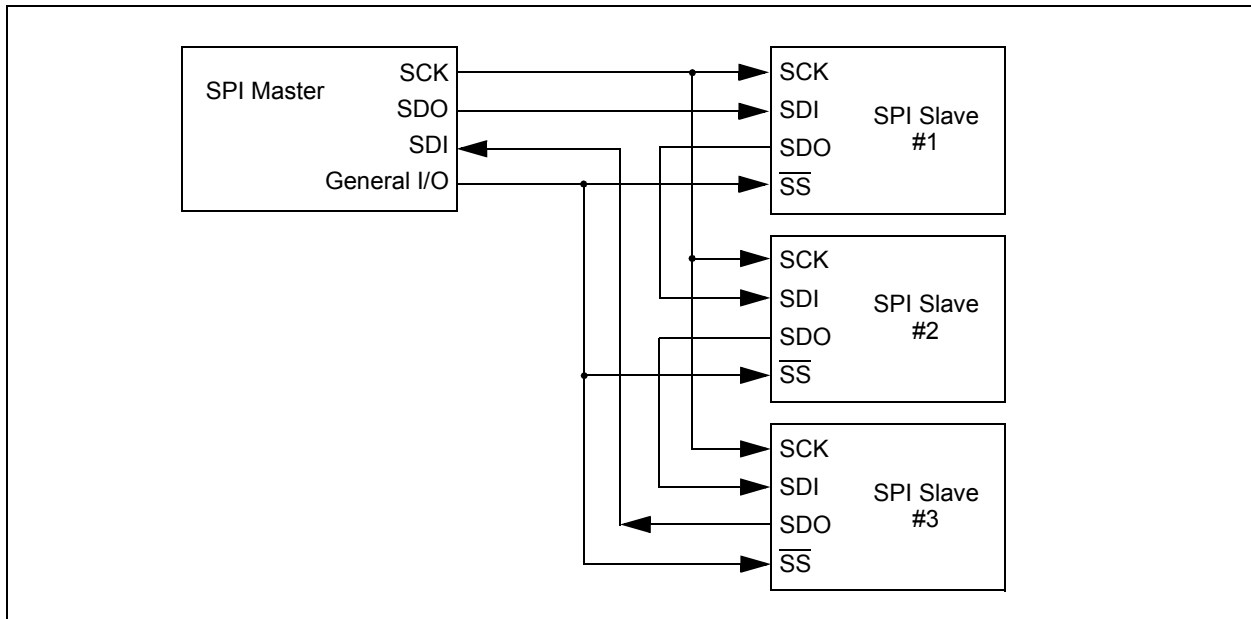
When the  $\overline{SS}$  pin is low, transmission and reception are enabled and the SDO pin is driven.

When the  $\overline{SS}$  pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

- Note 1:** When the SPI is in Slave mode with  $\overline{SS}$  pin control enabled ( $SSPCON1<3:0> = 0100$ ), the SPI module will reset if the  $\overline{SS}$  pin is set to VDD.
- 2:** When the SPI is used in Slave mode with CKE set; the user must enable  $\overline{SS}$  pin control.
- 3:** While operated in SPI Slave mode the SMP bit of the SSPSTAT register must remain clear.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the  $\overline{SS}$  pin to a high level or clearing the SSPEN bit.

**FIGURE 30-7: SPI DAISY-CHAIN CONNECTION**



## 30.4 I<sup>2</sup>C MODE OPERATION

All MSSP I<sup>2</sup>C communication is byte oriented and shifted out MSb first. Six SFR registers and two interrupt flags interface the module with the PIC<sup>®</sup> microcontroller and user software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I<sup>2</sup>C devices.

### 30.4.1 BYTE FORMAT

All communication in I<sup>2</sup>C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent back. After the eighth falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

### 30.4.2 DEFINITION OF I<sup>2</sup>C TERMINOLOGY

There is language and terminology in the description of I<sup>2</sup>C communication that have definitions specific to I<sup>2</sup>C. That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Philips I<sup>2</sup>C specification.

### 30.4.3 SDA AND SCL PINS

Selection of any I<sup>2</sup>C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRIS bits.

**Note 1:** Data is tied to output zero when an I<sup>2</sup>C mode is enabled.

**2:** Any device pin can be selected for SDA and SCL functions with the PPS peripheral. These functions are bidirectional. The SDA input is selected with the SSPDATPPS registers. The SCL input is selected with the SSPCLKPPS registers. Outputs are selected with the RxyPPS registers. It is the user's responsibility to make the selections so that both the input and the output for each function is on the same pin.

### 30.4.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit of the SSPCON3 register. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

TABLE 30-2: I<sup>2</sup>C BUS TERMS

TERM	Description
Transmitter	The device which shifts data out onto the bus.
Receiver	The device which shifts data in from the bus.
Master	The device that initiates a transfer, generates clock signals and terminates a transfer.
Slave	The device addressed by the master.
Multi-master	A bus with more than one device that can initiate data transfers.
Arbitration	Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted.
Synchronization	Procedure to synchronize the clocks of two or more devices on the bus.
Idle	No master is controlling the bus, and both SDA and SCL lines are high.
Active	Any time one or more master devices are controlling the bus.
Addressed Slave	Slave device that has received a matching address and is actively being clocked by a master.
Matching Address	Address byte that is clocked into a slave that matches the value stored in SSPADD.
Write Request	Slave receives a matching address with R/W bit clear, and is ready to clock in data.
Read Request	Master sends an address byte with the R/W bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop.
Clock Stretching	When a device on the bus hold SCL low to stall communication.
Bus Collision	Any time the SDA line is sampled low by the module while it is outputting and expected high state.

# PIC16(L)F1713/6

## 30.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSP module configured as an I<sup>2</sup>C slave in 7-bit Addressing mode. Figure 30-14 and Figure 30-15 is used as a visual reference for this description.

This is a step by step process of what typically must be done to accomplish I<sup>2</sup>C communication.

1. Start bit detected.
2. S bit of SSPSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
3. Matching address with R/W bit clear is received.
4. The slave pulls SDA low sending an  $\overline{\text{ACK}}$  to the master, and sets SSPIF bit.
5. Software clears the SSPIF bit.
6. Software reads received address from SSPBUF clearing the BF flag.
7. If SEN = 1; Slave software sets CKP bit to release the SCL line.
8. The master clocks out a data byte.
9. Slave drives SDA low sending an  $\overline{\text{ACK}}$  to the master, and sets SSPIF bit.
10. Software clears SSPIF.
11. Software reads the received byte from SSPBUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the master.
13. Master sends Stop condition, setting P bit of SSPSTAT, and the bus goes idle.

## 30.5.2.2 7-bit Reception with AHEN and DHEN

Slave device reception with AHEN and DHEN set operate the same as without these options with extra interrupts and clock stretching added after the eighth falling edge of SCL. These additional interrupts allow the slave software to decide whether it wants to  $\overline{\text{ACK}}$  the receive address or data byte, rather than the hardware. This functionality adds support for PMBus™ that was not present on previous versions of this module.

This list describes the steps that need to be taken by slave software to use these options for I<sup>2</sup>C communication. Figure 30-16 displays a module using both address and data holding. Figure 30-17 includes the operation with the SEN bit of the SSPCON2 register set.

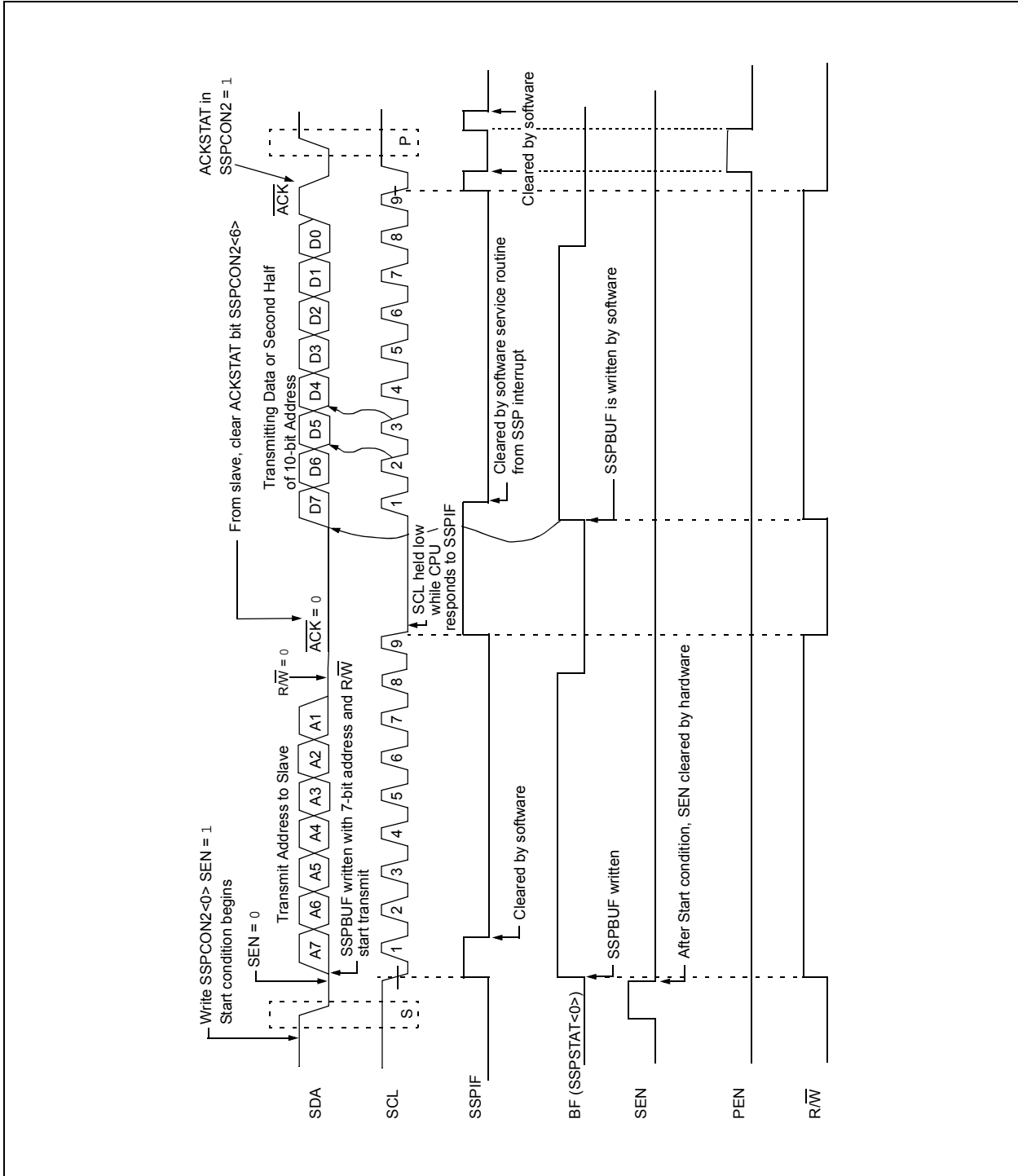
1. S bit of SSPSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
2. Matching address with R/W bit clear is clocked in. SSPIF is set and CKP cleared after the eighth falling edge of SCL.
3. Slave clears the SSPIF.
4. Slave can look at the ACKTIM bit of the SSPCON3 register to determine if the SSPIF was after or before the  $\overline{\text{ACK}}$ .
5. Slave reads the address value from SSPBUF, clearing the BF flag.
6. Slave sets  $\overline{\text{ACK}}$  value clocked out to the master by setting ACKDT.
7. Slave releases the clock by setting CKP.
8. SSPIF is set after an  $\overline{\text{ACK}}$ , not after a NACK.
9. If SEN = 1 the slave hardware will stretch the clock after the  $\overline{\text{ACK}}$ .
10. Slave clears SSPIF.

**Note:** SSPIF is still set after the 9th falling edge of SCL even if there is no clock stretching and BF has been cleared. Only if NACK is sent to master is SSPIF not set

11. SSPIF set and CKP cleared after eighth falling edge of SCL for a received data byte.
12. Slave looks at ACKTIM bit of SSPCON3 to determine the source of the interrupt.
13. Slave reads the received data from SSPBUF clearing BF.
14. Steps 7-14 are the same for each received data byte.
15. Communication is ended by either the slave sending an  $\overline{\text{ACK}} = 1$ , or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop Detect is disabled, the slave will only know by polling the P bit of the SSTSTAT register.



**FIGURE 30-28: I<sup>2</sup>C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)**



# PIC16(L)F1713/6

## 30.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSPCON2 register. When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 30-30).

### 30.6.8.1 WCOL Status Flag

If the user writes the SSPBUF when an Acknowledge sequence is in progress, then WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

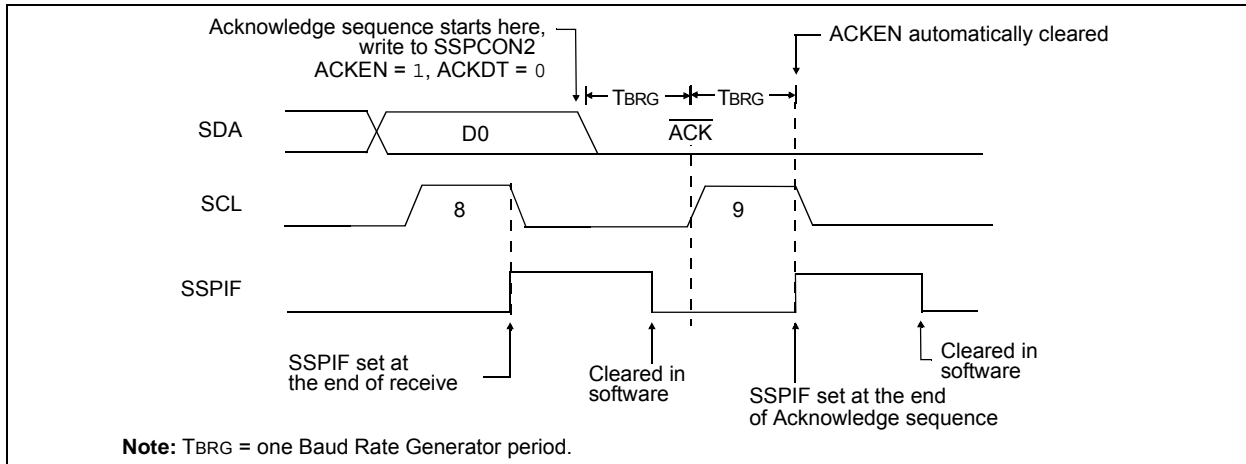
## 30.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN bit of the SSPCON2 register. At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit of the SSPSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 30-31).

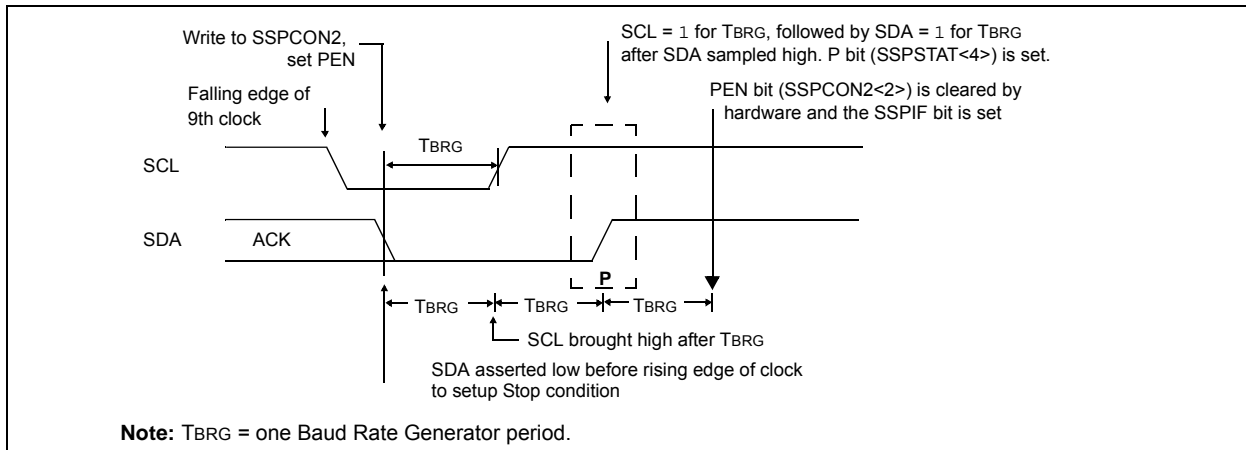
### 30.6.9.1 WCOL Status Flag

If the user writes the SSPBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

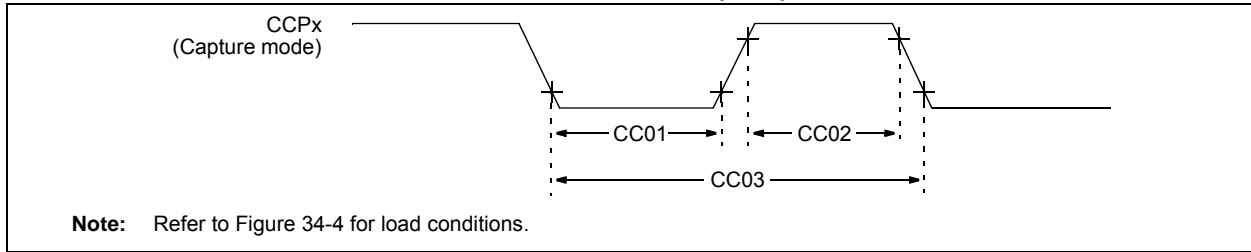
**FIGURE 30-30: ACKNOWLEDGE SEQUENCE WAVEFORM**



**FIGURE 30-31: STOP CONDITION RECEIVE OR TRANSMIT MODE**



**FIGURE 34-11: CAPTURE/COMPARE/PWM TIMINGS (CCP)**



**TABLE 34-13: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP)**

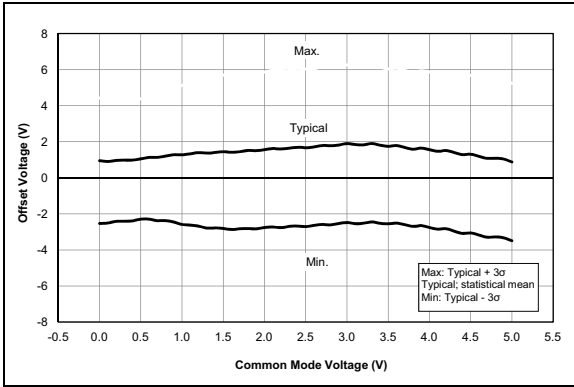
Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym.	Characteristic		Min.	Typ†	Max.	Units	Conditions
CC01*	TccL	CCPx Input Low Time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	20	—	—	ns	
CC02*	TccH	CCPx Input High Time	No Prescaler	$0.5T_{CY} + 20$	—	—	ns	
			With Prescaler	20	—	—	ns	
CC03*	TccP	CCPx Input Period		$\frac{3T_{CY} + 40}{*N}$	—	—	ns	N = prescale value

\* These parameters are characterized but not tested.

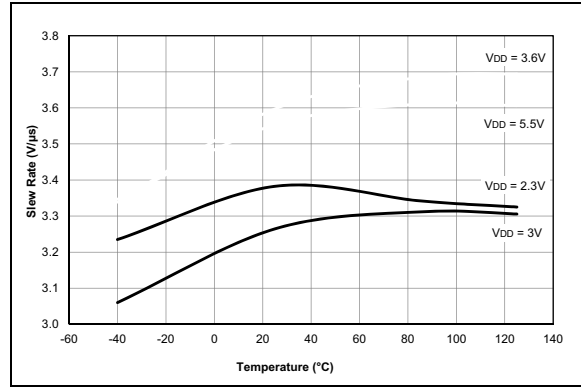
† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

# PIC16(L)F1713/6

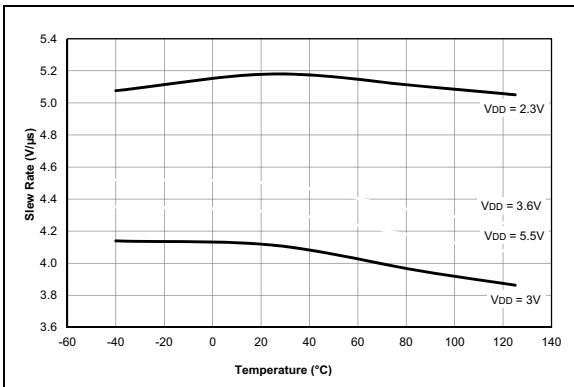
Note: Unless otherwise noted,  $V_{IN} = 5V$ ,  $F_{OSC} = 300\text{ kHz}$ ,  $C_{IN} = 0.1\ \mu\text{F}$ ,  $T_A = 25^\circ\text{C}$ .



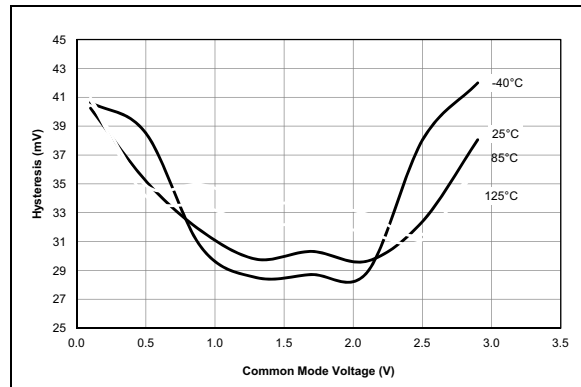
**FIGURE 35-97:** Op Amp, Offset Over Common Mode Voltage,  $V_{DD} = 5.0V$ , Temp. =  $25^\circ\text{C}$ , PIC16F1713/6 Only.



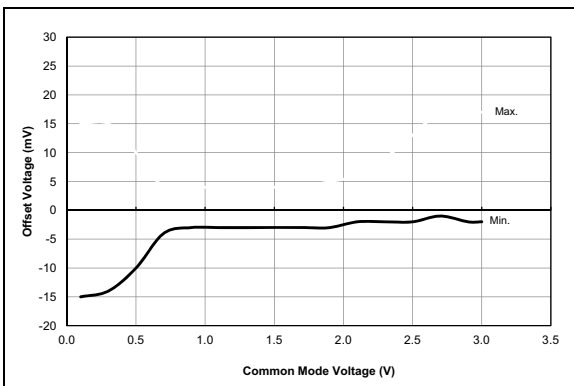
**FIGURE 35-98:** Op Amp, Output Slew Rate, Rising Edge, PIC16F1713/6 Only.



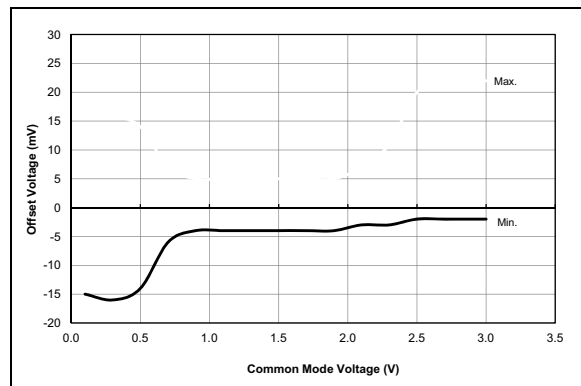
**FIGURE 35-99:** Op Amp, Output Slew Rate, Falling Edge, PIC16F1713/6 Only.



**FIGURE 35-100:** Comparator Hysteresis, NP Mode ( $CxSP = 1$ ),  $V_{DD} = 3.0V$ , Typical Measured Values.



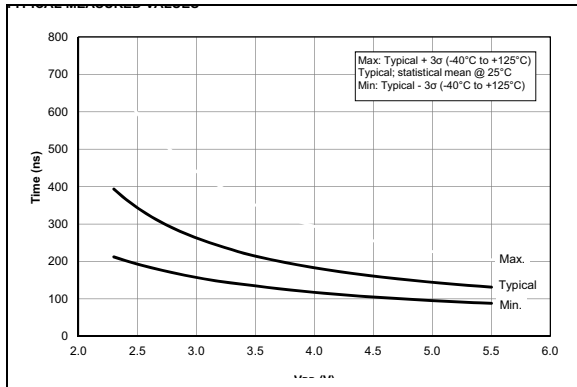
**FIGURE 35-101:** Comparator Offset, NP Mode ( $CxSP = 1$ ),  $V_{DD} = 3.0V$ , Typical Measured Values at  $25^\circ\text{C}$ .



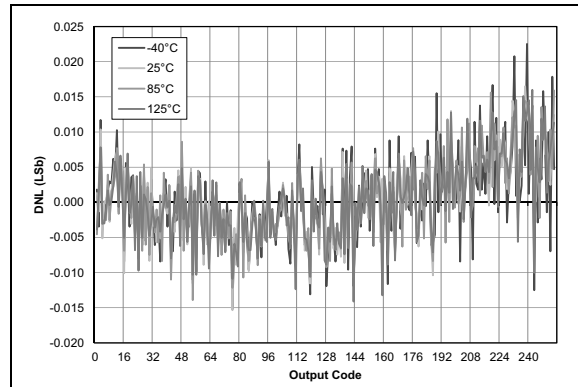
**FIGURE 35-102:** Comparator Offset, NP Mode ( $CxSP = 1$ ),  $V_{DD} = 3.0V$ , Typical Measured Values From  $-40^\circ\text{C}$  to  $125^\circ\text{C}$ .

# PIC16(L)F1713/6

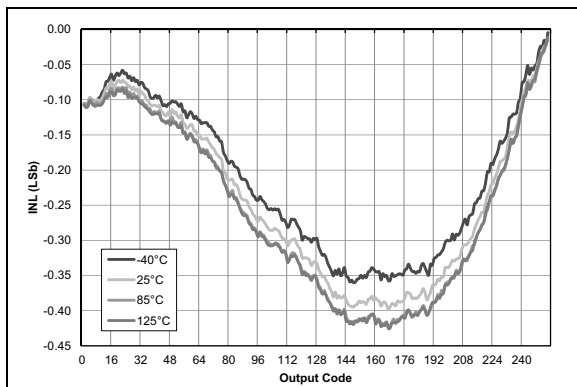
Note: Unless otherwise noted,  $V_{IN} = 5V$ ,  $F_{OSC} = 300\text{ kHz}$ ,  $C_{IN} = 0.1\ \mu\text{F}$ ,  $T_A = 25^\circ\text{C}$ .



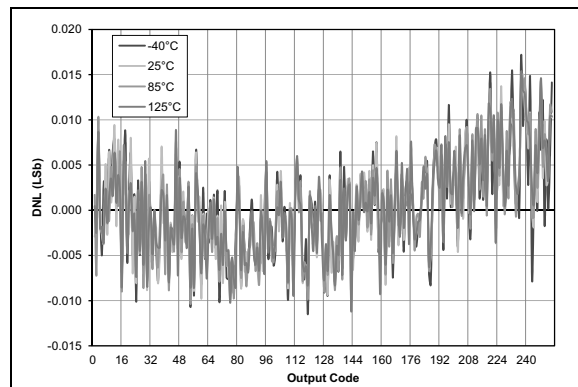
**FIGURE 35-109:** Comparator Output Filter Delay Time Over Temp., NP Mode ( $CxSP = 1$ ), Typical Measured Values, PIC16F1713/6 Only.



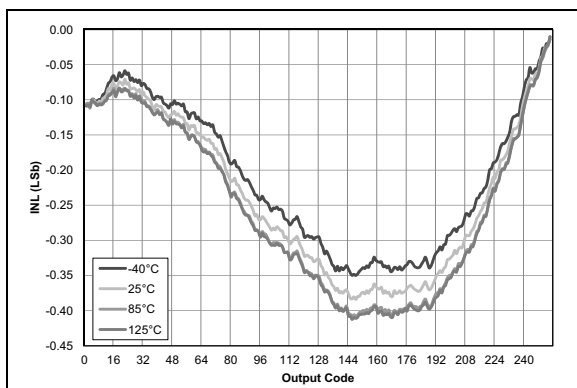
**FIGURE 35-110:** Typical DAC DNL Error,  $V_{DD} = 3.0V$ ,  $V_{REF} = \text{External } 3V$ .



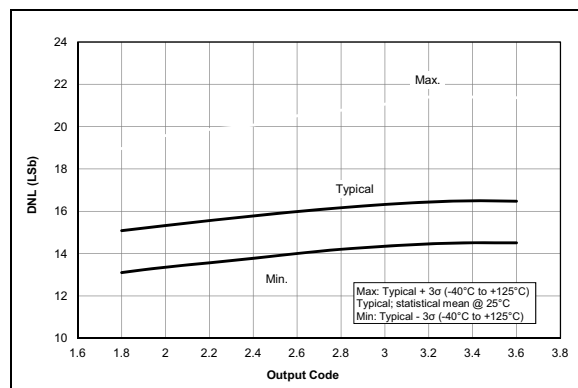
**FIGURE 35-111:** Typical DAC INL Error,  $V_{DD} = 3.0V$ ,  $V_{REF} = \text{External } 3V$ .



**FIGURE 35-112:** Typical DAC INL Error,  $V_{DD} = 5.0V$ ,  $V_{REF} = \text{External } 5V$ , PIC16F1713/6 Only.



**FIGURE 35-113:** Typical DAC INL Error,  $V_{DD} = 5.0V$ ,  $V_{REF} = \text{External } 5V$ , PIC16F1713/6 Only.



**FIGURE 35-114:** DAC INL Error,  $V_{DD} = 3.0V$ , PIC16LF1713/6 Only.