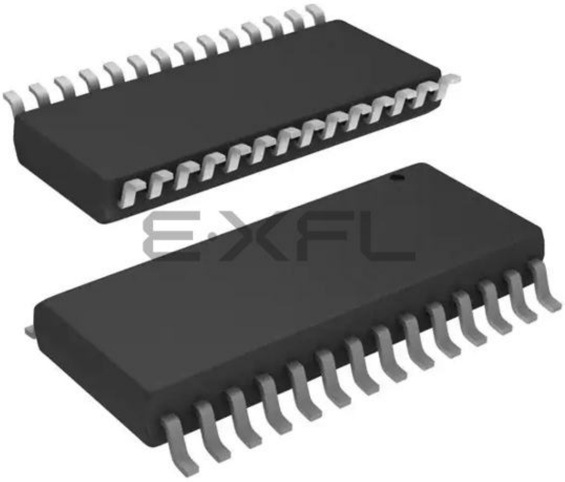
Microchip Technology - PIC16LF18854-I/SO Datasheet



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	32MHz
Connectivity	I²C, 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	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x10b; D/A 1x5b
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	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf18854-i-so

TABLE 3-11: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
Bank 8											
CPU CORE REGISTERS; see Table 3-2 for specifics											
40Ch	SCANLADRL	LADR<7:0>								`0000 0000`	`0000 0000`
40Dh	SCANLADRH	LADR<15:8>								`0000 0000`	`0000 0000`
40Eh	SCANHADRL	HADR<7:0>								`1111 1111`	`1111 1111`
40Fh	SCANHADRH	HADR<15:8>								`1111 1111`	`1111 1111`
410h	SCANCON0	EN	SCANGO	BUSY	INVALID	INTM	—	MODE<1:0>		`0000 0-00`	`0000 0-00`
411h	SCANTRIG	—	—	—	—	TSEL<3:0>				`---- 0000`	`---- 0000`
412h	—	Unimplemented								—	—
413h	—	Unimplemented								—	—
414h	—	Unimplemented								—	—
415h	—	Unimplemented								—	—
416h	CRCDATL	DATA<7:0>								`xxxx xxxx`	`xxxx xxxx`
417h	CRCDATH	DATA<15:8>								`xxxx xxxx`	`xxxx xxxx`
418h	CRCACCL	ACC<7:0>								`0000 0000`	`0000 0000`
419h	CRCACCH	ACC<15:8>								`0000 0000`	`0000 0000`
41Ah	CRCSHIFTL	SHIFT<7:0>								`0000 0000`	`0000 0000`
41Bh	CRCSHIFTH	SHIFT<15:8>								`0000 0000`	`0000 0000`
41Ch	CRCXORL	X<7:1>							—	`xxxx xxx-`	`xxxx xxx-`
41Dh	CRCXORH	X<15:8>								`xxxx xxxx`	`xxxx xxxx`
41Eh	CRCCON0	EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL	`0000 --00`	`0000 --00`
41Fh	CRCCON1	DLEN<3:0>				PLEN<3:0>				`0000 0000`	`0000 0000`

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

Note 1: Register present on PIC16F18854 devices only.

2: Unimplemented, read as ‘1’.

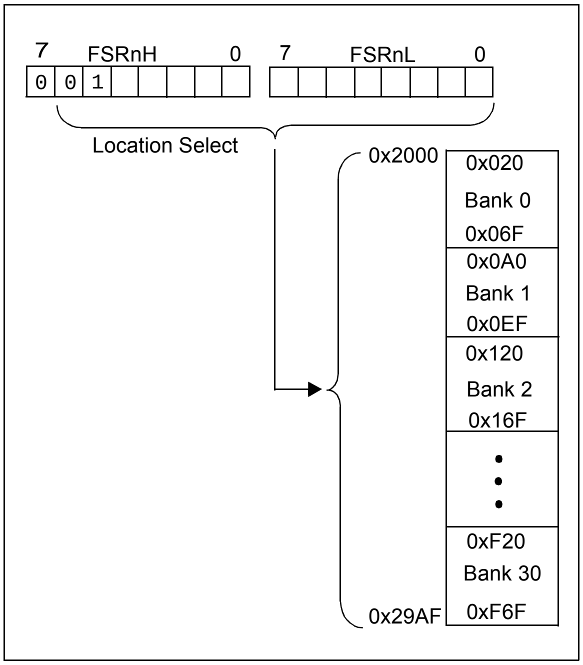
3.5.2 LINEAR DATA MEMORY

The linear data memory is the region from FSR address 0x2000 to FSR address 0x29AF. This region is a virtual region that points back to the 80-byte blocks of GPR memory in all the banks.

Unimplemented memory reads as 0x00. Use of the linear data memory region allows buffers to be larger than 80 bytes because incrementing the FSR beyond one bank will go directly to the GPR memory of the next bank.

The 16 bytes of common memory are not included in the linear data memory region.

FIGURE 3-10: LINEAR DATA MEMORY MAP



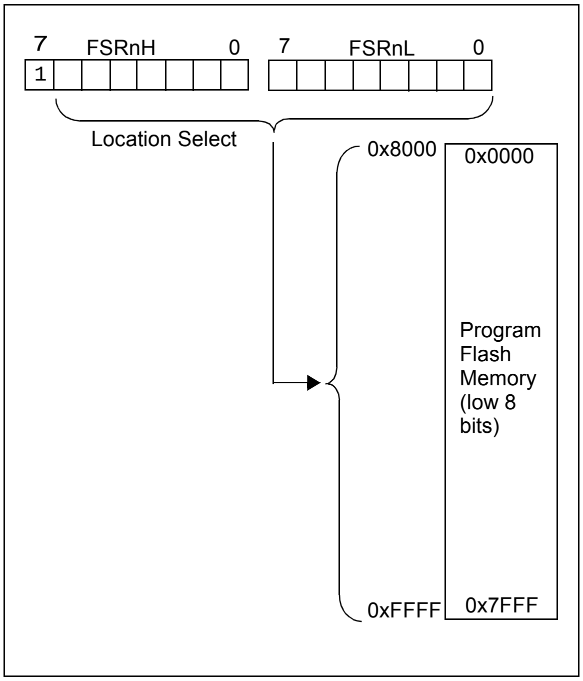
3.5.3 DATA EEPROM MEMORY

The EEPROM memory can be read or written through the NVMCONx/NVMADRx/NVMDATx register interface (see section **Section 10.2 "Data EEPROM Memory"**). However, to make access to the EEPROM memory easier, read-only access to the EEPROM contents are also available through indirect addressing by an FSR. When the MSB of the FSR (ex: FSRxH) is set to 0x70, the lower 8-bit address value (in FSRxL) determines the EEPROM location that may be read from (through the INDF register). In other words, the EEPROM address range 0x00-0xFF is mapped into the FSR address space between 0x7000-0x70FF. Writing to the EEPROM cannot be accomplished via the FSR/INDF interface. Reads from the EEPROM through the FSR/INDF interface will require one additional instruction cycle to complete.

3.5.4 PROGRAM FLASH MEMORY

To make constant data access easier, the entire Program Flash Memory is mapped to the upper half of the FSR address space. When the MSB of FSRnH is set, the lower 15 bits are the address in program memory which will be accessed through INDF. Only the lower eight bits of each memory location is accessible via INDF. Writing to the Program Flash Memory cannot be accomplished via the FSR/INDF interface. All instructions that access Program Flash Memory via the FSR/INDF interface will require one additional instruction cycle to complete.

FIGURE 3-11: PROGRAM FLASH MEMORY MAP



4.0 DEVICE CONFIGURATION

Device configuration consists of Configuration Words, Code Protection and Device ID.

4.1 Configuration Words

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as shown in Table 4-1.

TABLE 4-1: CONFIGURATION WORD LOCATIONS

Configuration Word	Location
CONFIG1	8007h
CONFIG2	8008h
CONFIG3	8009h
CONFIG4	800Ah
CONFIG5	800Bh

Note: The $\overline{\text{DEBUG}}$ bit in Configuration Words is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a ‘1’.

4.6 Device ID and Revision ID

The 14-bit device ID word is located at 8006h and the 14-bit revision ID is located at 8005h. These locations are read-only and cannot be erased or modified.

Development tools, such as device programmers and debuggers, may be used to read the Device ID, Revision ID and Configuration Words. These locations can also be read from the NVMCON register.

4.7 Register Definitions: Device and Revision

REGISTER 4-6: DEVID: DEVICE ID REGISTER

R	R	R	R	R	R
DEV<13:8>					
bit 13					bit 8

R	R	R	R	R	R	R	R
DEV<7:0>							
bit 7							bit 0

Legend:

R = Readable bit

‘1’ = Bit is set ‘0’ = Bit is cleared

bit 13-0 **DEV<13:0>:** Device ID bits

Device	DEVID<13:0> Values
PIC16F18854	`11 0000 0110 1010 (306Ah)`
PIC16LF18854	`11 0000 0110 1011 (306Bh)`

REGISTER 4-7: REVISIONID: REVISION ID REGISTER

R	R	R	R	R	R	R	R	R	R	R	R	R	R
1	0	MJRREV<5:0>						MNRREV<5:0>					
bit 13													bit 0

Legend:

R = Readable bit

‘0’ = Bit is cleared ‘1’ = Bit is set x = Bit is unknown

bit 13-12 **Fixed Value**: Read-only bits
These bits are fixed with value ‘`10`’ for all devices included in this data sheet.

bit 11-6 **MJRREV<5:0>**: Major Revision ID bits
These bits are used to identify a major revision. A major revision is indicated by an all layer revision (B0, C0, etc.)

bit 5-0 **MNRREV<5:0>**: Minor Revision ID bits
These bits are used to identify a minor revision.

REGISTER 7-8: PIE6: PERIPHERAL INTERRUPT ENABLE REGISTER 6

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	CCP5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE
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	HS = Hardware set

bit 7-5 **Unimplemented:** Read as ‘0’.

bit 4 **CCP5IE:** CCP5 Interrupt Enable bit
1 = CCP5 interrupt is enabled
0 = CCP5 interrupt is disabled

bit 3 **CCP4IE:** CCP4 Interrupt Enable bit
1 = CCP4 interrupt is enabled
0 = CCP4 interrupt is disabled

bit 2 **CCP3IE:** CCP3 Interrupt Enable bit
1 = CCP3 interrupt is enabled
0 = CCP3 interrupt is disabled

bit 1 **CCP2IE:** CCP2 Interrupt Enable bit
1 = CCP2 interrupt is enabled
0 = CCP2 interrupt is disabled

bit 0 **CCP1IE:** CCP1 Interrupt Enable bit
1 = CCP1 interrupt is enabled
0 = CCP1 interrupt is disabled

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

REGISTER 7-18: PIR6: PERIPHERAL INTERRUPT REQUEST REGISTER 6

U-0	U-0	U-0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
—	—	—	CCP5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF
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	HS = Hardware set

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4 **CCP5IF:** CCP5 Interrupt Flag bit
CCP5IF = 1:
Capture mode: Capture occurred (must be cleared in software)
Compare mode: Compare match occurred (must be cleared in software)
PWM mode: Output trailing edge occurred (must be cleared in software)
CCP5IF = 0:
Capture mode: Capture did not occur
Compare mode: Compare match did not occur
PWM mode: Output trailing edge did not occur

bit 3 **CCP4IF:** CCP4 Interrupt Flag bit
CCP4IF = 1:
Capture mode: Capture occurred (must be cleared in software)
Compare mode: Compare match occurred (must be cleared in software)
PWM mode: Output trailing edge occurred (must be cleared in software)
CCP4IF = 0:
Capture mode: Capture did not occur
Compare mode: Compare match did not occur
PWM mode: Output trailing edge did not occur

bit 2 **CCP3IF:** CCP3 Interrupt Flag bit
CCP3IF = 1:
Capture mode: Capture occurred (must be cleared in software)
Compare mode: Compare match occurred (must be cleared in software)
PWM mode: Output trailing edge occurred (must be cleared in software)
CCP3IF = 0:
Capture mode: Capture did not occur
Compare mode: Compare match did not occur
PWM mode: Output trailing edge did not occur

bit 1 **CCP2IF:** CCP2 Interrupt Flag bit
CCP2IF = 1:
Capture mode: Capture occurred (must be cleared in software)
Compare mode: Compare match occurred (must be cleared in software)
PWM mode: Output trailing edge occurred (must be cleared in software)
CCP2IF = 0:
Capture mode: Capture did not occur
Compare mode: Compare match did not occur
PWM mode: Output trailing edge did not occur

bit 0 **CCP1IF:** CCP1 Interrupt Flag bit
CCP1IF = 1:
Capture mode: Capture occurred (must be cleared in software)
Compare mode: Compare match occurred (must be cleared in software)
PWM mode: Output trailing edge occurred (must be cleared in software)
CCP1IF = 0:
Capture mode: Capture did not occur
Compare mode: Compare match did not occur
PWM mode: Output trailing edge did not occur

12.6 PORTB Registers

12.6.1 DATA REGISTER

PORTB is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 12-13). Setting a TRISB bit (= `1`) will make the corresponding PORTB pin an input (i.e., disable the output driver). Clearing a TRISB bit (= `0`) will make the corresponding PORTB pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). Example 12.4.9 shows how to initialize PORTB.

Reading the PORTB register (Register 12-12) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATB).

The PORT data latch LATB (Register 12-14) holds the output port data, and contains the latest value of a LATB or PORTB write.

EXAMPLE 12-2: INITIALIZING PORTA

```
; This code example illustrates
; initializing the PORTA register. The
; other ports are initialized in the same
; manner.

BANKSEL PORTA       ;
CLRF    PORTA       ;Init PORTA
BANKSEL LATA        ;Data Latch
CLRF    LATA        ;
BANKSEL ANSELA      ;
CLRF    ANSELA      ;digital I/O
BANKSEL TRISA       ;
MOVLW   B'00111000' ;Set RA<5:3> as inputs
MOVWF   TRISA       ;and set RA<2:0> as
                    ;outputs
```

12.6.2 DIRECTION CONTROL

The TRISB register (Register 12-13) controls the PORTB pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISB register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read ‘`0`’.

12.6.3 OPEN-DRAIN CONTROL

The ODCONB register (Register 12-17) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCONB bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCONB bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

Note: It is not necessary to set open-drain control when using the pin for I^{2}C; the I^{2}C module controls the pin and makes the pin open-drain.

12.6.4 SLEW RATE CONTROL

The SLRCONB register (Register 12-18) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONB bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONB bit is cleared, The corresponding port pin drive slews at the maximum rate possible.

REGISTER 12-30: CCDPC: CURRENT CONTROLLED DRIVE POSITIVE PORTC REGISTER

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
CCDPC7	CCDPC6	CCDPC5	CCDPC4	CCDPC3	CCDPC2	CCDPC1	CCDPC0
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 7-0 **CCDPC<7:0>:** RC<7:0> Current Controlled Drive Positive Control bits[(1)]
`1` = Current-controlled source enabled
`0` = Current-controlled source disabled

Note 1: If CCDPCy is set, when CCDEN = `0` (Register 12-1), operation of the pin is undefined.

REGISTER 12-31: CCDNC: CURRENT CONTROLLED DRIVE NEGATIVE PORTC REGISTER

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
CCDNC7	CCDNC6	CCDNC5	CCDNC4	CCDNC3	CCDNC2	CCDNC1	CCDNC0
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 7-0 **CCDNC<7:0>:** RC<7:0> Current Controlled Drive Negative Control bits[(1)]
`1` = Current-controlled source enabled
`0` = Current-controlled source disabled

Note 1: If CCDNCy is set, when CCDEN = `0` (Register 12-1), operation of the pin is undefined.

TABLE 13-1: PPS INPUT SIGNAL ROUTING OPTIONS (CONTINUED)

Input Signal Name	Input Register Name	Default Location at POR	Remappable to Pins of PORTx		
			PIC16F18854		
			PORTA	PORTB	PORTC
CLCIN3	CLCIN3PPS	RB7		●	●
ADCACT	ADCACTPPS	RB4		●	●
SCK1/SCL1	SSP1CLKPPS	RC3		●	●
SDI1/SDA1	SSP1DATPPS	RC4		●	●
SS1	SSPSS1PPS	RA5	●		●
SCK2/SCL2	SSP2CLKPPS	RB1		●	●
SDI2/SDA2	SSP2DATPPS	RB2		●	●
SS2	SSP2SSPPS	RB0		●	●
RX/DT	RXPPS	RC7		●	●
CK	TXPPS	RC6		●	●

TABLE 13-4: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	215
INTPPS	—	—	—	—	INTPPS<3:0>				214
T0CKIPPS	—	—	—	—	T0CKIPPS<3:0>				214
T1CKIPPS	—	—	—	T1CKIPPS<4:0>					214
T1GPPS	—	—	—	T1GPPS<4:0>					214
T3CKIPPS	—	—	—	T3CKIPPS<4:0>					214
T3GPPS	—	—	—	T3GPPS<4:0>					214
T5CKIPPS	—	—	—	T5CKIPPS<4:0>					214
T5GPPS	—	—	—	T5GPPS<4:0>					214
T5GPPS	—	—	—	T5GPPS<4:0>					214
T2AINPPS				T2AINPPS<4:0>					214
T4AINPPS				T5AINPPS<4:0>					214
T6AINPPS				T6AINPPS<4:0>					214
CCP1PPS	—	—	—	CCP1PPS<4:0>					214
CCP2PPS	—	—	—	CCP2PPS<4:0>					214
CCP3PPS	—	—	—	CCP3PPS<4:0>					214
CCP4PPS	—	—	—	CCP4PPS<4:0>					214
CCP5PPS	—	—	—	CCP5PPS<4:0>					214
CWG1PPS	—	—	—	CWG1PPS<4:0>					214
CWG2PPS	—	—	—	CWG2PPS<4:0>					214
CWG3PPS	—	—	—	CWG3PPS<4:0>					214
MDCARLPPS	—	—	—	MDCARLPPS<4:0>					214
MDCARHPPS	—	—	—	MDCARHPPS<4:0>					214
MDSRCPPS	—	—	—	MDSRCPPS<4:0>					214
SSP1CLKPPS	—	—	—	SSP1CLKPPS<4:0>					214
SSP1DATPPS	—	—	—	SSP1DATPPS<4:0>					214
SSP1SSPPS	—	—	—	SSP1SSPPS<4:0>					214
SSP2CLKPPS	—	—	—	SSP2CLKPPS<4:0>					214
SSP2DATPPS	—	—	—	SSP2DATPPS<4:0>					214
SSP2SSPPS	—	—	—	SSP2SSPPS<4:0>					214
RXPPS	—	—	—	RXPPS<4:0>					215
TXPPS	—	—	—	TXPPS<4:0>					214
CLCIN0PPS	—	—	—	CLCIN0PPS<4:0>					214
CLCIN1PPS	—	—	—	CLCIN1PPS<4:0>					214
CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>					214
CLCIN3PPS	—	—	—	CLCIN3PPS<4:0>					214
SMT1WINPPS	—	—	—	SMT1WINPPS<4:0>					214
SMT1SIGPPS	—	—	—	SMT1SIGPPS<4:0>					214
SMT2WINPPS	—	—	—	SMT2WINPPS<4:0>					214
SMT2SIGPPS	—	—	—	SMT2SIGPPS<4:0>					214
ADCACTPPS	—	—	—	ADCACTPPS<4:0>					
RA0PPS	—	—	RA0PPS<5:0>						215
RA1PPS	—	—	RA1PPS<5:0>						215
RA2PPS	—	—	RA2PPS<5:0>						215
RA3PPS	—	—	RA3PPS<5:0>						215

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the PPS module.
Note 1: PIC16F18875 only.

18.8 Comparator Response Time

The comparator output is indeterminate for a period of time after the change of an input source or the selection of a new reference voltage. This period is referred to as the response time. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response time to a comparator input change. See the Comparator and Voltage Reference Specifications in Table 37-14 for more details.

18.9 Analog Input Connection Considerations

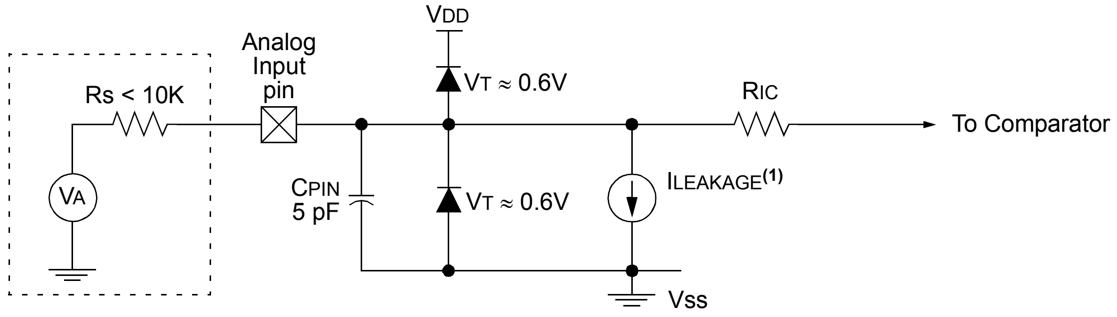
A simplified circuit for an analog input is shown in Figure 18-3. Since the analog input pins share their connection with a digital input, they have reverse biased ESD protection diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Also, any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current to minimize inaccuracies introduced.

Note 1: When reading a PORT register, all pins configured as analog inputs will read as a ‘0’. Pins configured as digital inputs will convert as an analog input, according to the input specification.

2: Analog levels on any pin defined as a digital input, may cause the input buffer to consume more current than is specified.

FIGURE 18-3: ANALOG INPUT MODEL



Legend:

CPIN	= Input Capacitance
ILEAKAGE	= Leakage Current at the pin due to various junctions
RIC	= Interconnect Resistance
RS	= Source Impedance
VA	= Analog Voltage
VT	= Threshold Voltage

Note 1: See I/O Ports in Table 37-4.

20.9 CWG Steering Mode

In Steering mode (MODE = 00x), the CWG allows any combination of the CWGxx pins to be the modulated signal. The same signal can be simultaneously available on multiple pins, or a fixed-value output can be presented.

When the respective STRx bit of CWGxOCON0 is '0', the corresponding pin is held at the level defined. When the respective STRx bit of CWGxOCON0 is '1', the pin is driven by the input data signal. The user can assign the input data signal to one, two, three, or all four output pins.

The POLx bits of the CWGxCON1 register control the signal polarity only when STRx = 1.

The CWG auto-shutdown operation also applies in Steering modes as described in **Section 20.10 "Auto-Shutdown"**. An auto-shutdown event will only affect pins that have STRx = 1.

20.9.1 STEERING SYNCHRONIZATION

Changing the MODE bits allows for two modes of steering, synchronous and asynchronous.

When MODE = 000, the steering event is asynchronous and will happen at the end of the instruction that writes to STRx (that is, immediately). In this case, the output signal at the output pin may be an incomplete waveform. This can be useful for immediately removing a signal from the pin.

When MODE = 001, the steering update is synchronous and occurs at the beginning of the next rising edge of the input data signal. In this case, steering the output on/off will always produce a complete waveform.

Figure 20-10 and Figure 20-11 illustrate the timing of asynchronous and synchronous steering, respectively.

FIGURE 20-10: EXAMPLE OF STEERING EVENT AT END OF INSTRUCTION (MODE<2:0> = 000)

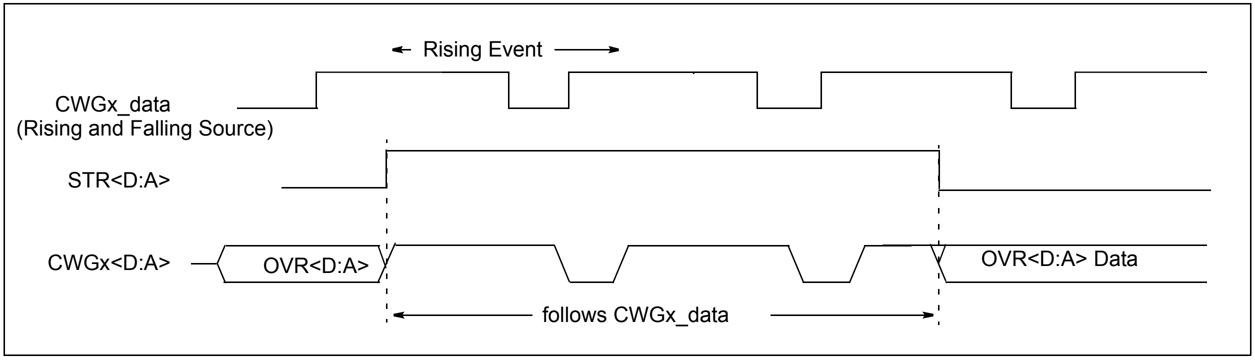


FIGURE 20-11: EXAMPLE OF STEERING EVENT AT BEGINNING OF INSTRUCTION (MODE<2:0> = 001)

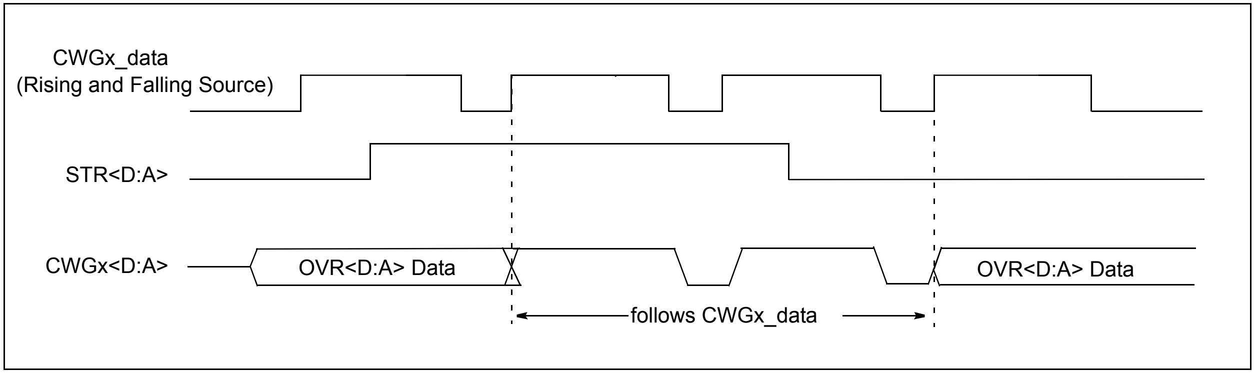
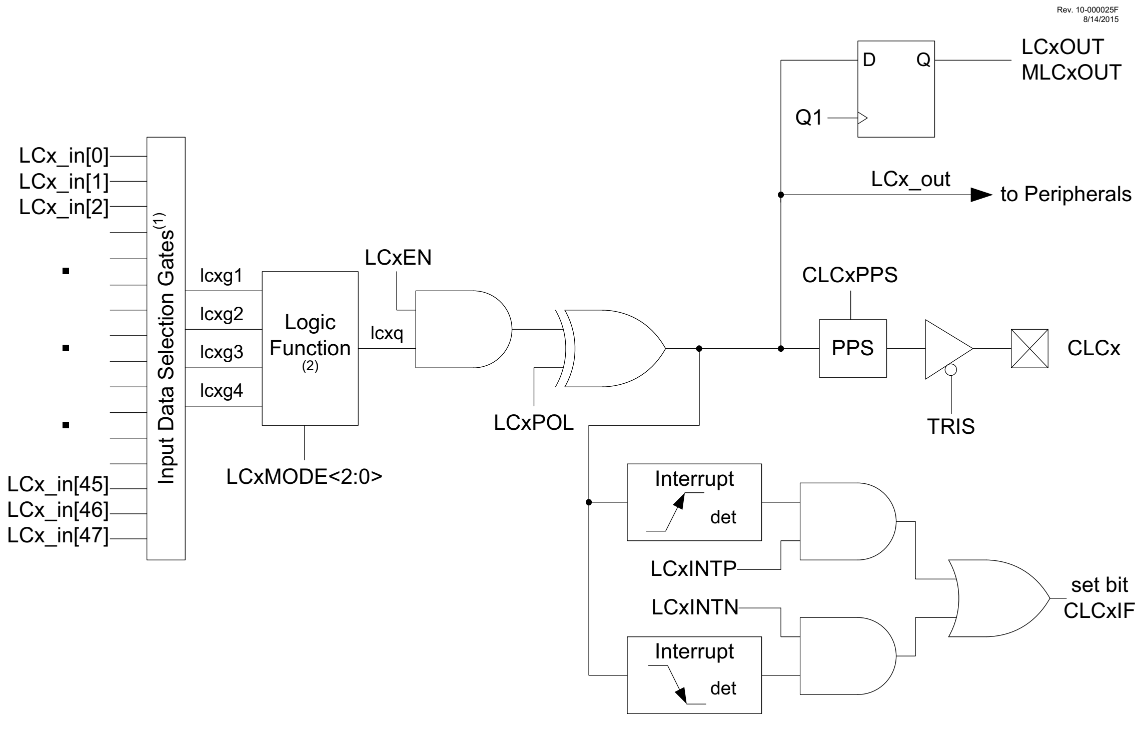


FIGURE 22-1: CLCx SIMPLIFIED BLOCK DIAGRAM



Note 1: See Figure 22-2: Input Data Selection and Gating

2: See Figure 22-3: Programmable Logic Functions.

TABLE 22-4: SUMMARY OF REGISTERS ASSOCIATED WITH CLCx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	114
PIR5	CLC4IF	CLC3IF	CLC2IF	CLC1IF	—	TMR5GIF	TMR3GIF	TMR1GIF	129
PIE5	CLC4IE	CLC4IE	CLC2IE	CLC1IE	—	TMR5GIE	TMR3GIE	TMR1GIE	120
CLC1CON	LC1EN	—	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			292
CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	293
CLC1SEL0	—	—	LC1D1S<5:0>						294
CLC1SEL1	—	—	LC1D2S<5:0>						294
CLC1SEL2	—	—	LC1D3S<5:0>						294
CLC1SEL3	—	—	LC1D4S<5:0>						294
CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	295
CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	296
CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	297
CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	298
CLC2CON	LC2EN	—	LC2OUT	LC2INTP	LC2INTN	LC2MODE<2:0>			292
CLC2POL	LC2POL	—	—	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1POL	293
CLC2SEL0	—	—	LC2D1S<5:0>						294
CLC2SEL1	—	—	LC2D2S<5:0>						294
CLC2SEL2	—	—	LC2D3S<5:0>						294
CLC2SEL3	—	—	LC2D4S<5:0>						294
CLC2GLS0	LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N	295
CLC2GLS1	LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N	296
CLC2GLS2	LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	297
CLC2GLS3	LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	298
CLC3CON	LC3EN	—	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>			292
CLC3POL	LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	293
CLC3SEL0	—	—	LC3D1S<5:0>						294
CLC3SEL1	—	—	LC3D2S<5:0>						294
CLC3SEL2	—	—	LC3D3S<5:0>						294
CLC3SEL3	—	—	LC3D4S<5:0>						294
CLC3GLS0	LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	295
CLC3GLS1	LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	296
CLC3GLS2	LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	297
CLC3GLS3	LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	298
CLC4CON	LC4EN	—	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			292
CLC4POL	LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	293
CLC4SEL0	—	—	LC4D1S<5:0>						294
CLC4SEL1	—	—	LC4D2S<5:0>						294
CLC4SEL2	—	—	LC4D3S<5:0>						294
CLC4SEL3	—	—	LC4D4S<5:0>						294
CLC4GLS0	LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	295

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the CLCx modules.

24.0 NUMERICALLY CONTROLLED OSCILLATOR (NCO) MODULE

The Numerically Controlled Oscillator (NCO) module is a timer that uses overflow from the addition of an increment value to divide the input frequency. The advantage of the addition method over simple counter driven timer is that the output frequency resolution does not vary with the divider value. The NCO is most useful for application that requires frequency accuracy and fine resolution at a fixed duty cycle.

Features of the NCO include:

- 20-bit Increment Function
- Fixed Duty Cycle mode (FDC) mode
- Pulse Frequency (PF) mode
- Output Pulse Width Control
- Multiple Clock Input Sources
- Output Polarity Control
- Interrupt Capability

Figure 24-1 is a simplified block diagram of the NCO module.

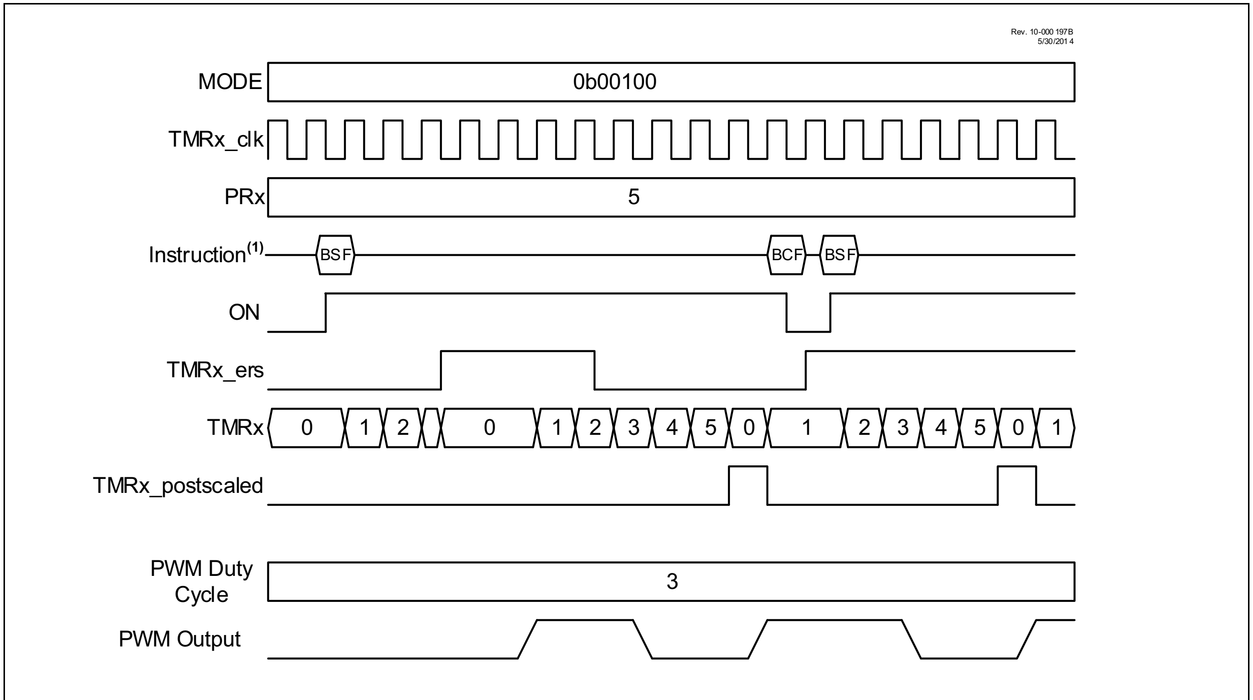
29.5.3 EDGE-TRIGGERED HARDWARE LIMIT MODE

In Hardware Limit mode the timer can be reset by the TMRx_ers external signal before the timer reaches the period count. Three types of Resets are possible:

- Reset on rising or falling edge (MODE<4:0>= `00011`)
- Reset on rising edge (MODE<4:0> = `00100`)
- Reset on falling edge (MODE<4:0> = `00101`)

When the timer is used in conjunction with the CCP in PWM mode then an early Reset shortens the period and restarts the PWM pulse after a two clock delay. Refer to Figure 29-6.

FIGURE 29-6: EDGE-TRIGGERED HARDWARE LIMIT MODE TIMING DIAGRAM (MODE = `00100`)



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.

30.1 Capture Mode

The Capture mode function described in this section is available and identical for all CCP modules.

Capture mode makes use of the 16-bit Timer1 resource. When an event occurs on the capture source, the 16-bit CCPRxH:CCPRxL register pair captures and stores the 16-bit value of the TMR1H:TMR1L register pair, respectively. An event is defined as one of the following and is configured by the CCPxMODE<3:0> bits of the CCPxCON register:

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

When a capture is made, the Interrupt Request Flag bit CCPxIF of the PIR6 register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRxH, CCPRxL register pair is read, the old captured value is overwritten by the new captured value.

Figure 30-1 shows a simplified diagram of the capture operation.

30.1.1 CAPTURE SOURCES

In Capture mode, the CCPx pin should be configured as an input by setting the associated TRIS control bit.

Note:	If the CCPx pin is configured as an output, a write to the port can cause a capture condition.

The capture source is selected by configuring the CCPxCTS<2:0> bits of the CCPxCAP register. The following sources can be selected:

- CCPxPPS input
- C1OUT_sync
- C2OUT_sync
- IOC_interrupt
- LC1_out
- LC2_out
- LC3_out
- LC4_out

FIGURE 30-1: CAPTURE MODE OPERATION BLOCK DIAGRAM

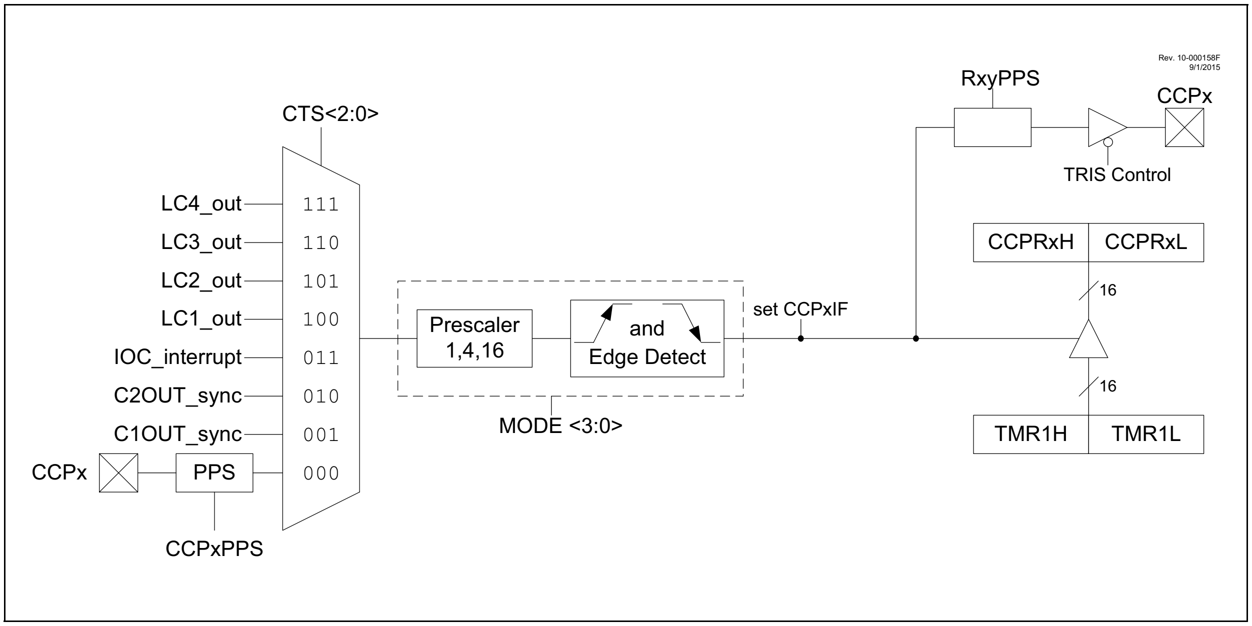
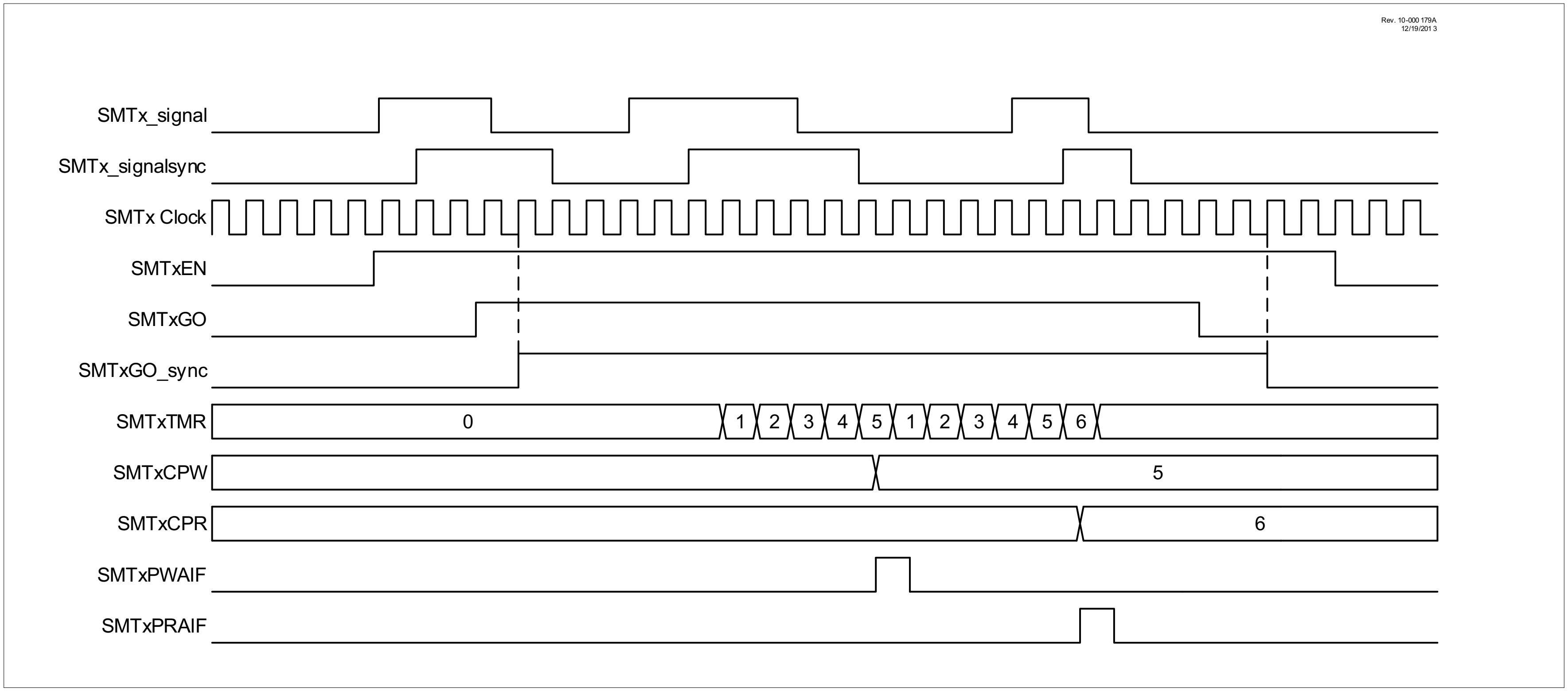


FIGURE 32-9: HIGH AND LOW MEASURE MODE SINGLE ACQUISITION TIMING DIAGRAM



33.3 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 BAUD1CON register selects 16-bit mode.

The SPBRGH, SPBRGL 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 TX1STA register and the BRG16 bit of the BAUD1CON register. In Synchronous mode, the BRGH bit is ignored.

Table 33-1 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 33-3. 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 SPBRGH, SPBRGL 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 SPBRGH:SPBRGL:

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

$$= \frac{\left(\frac{16000000}{9600}\right)}{64} - 1$$

$$= [25.042] = 25$$

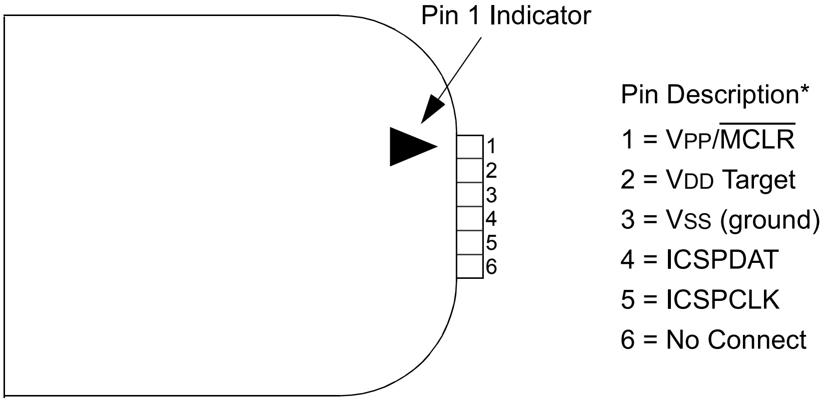
$$\text{Calculated Baud Rate} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

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

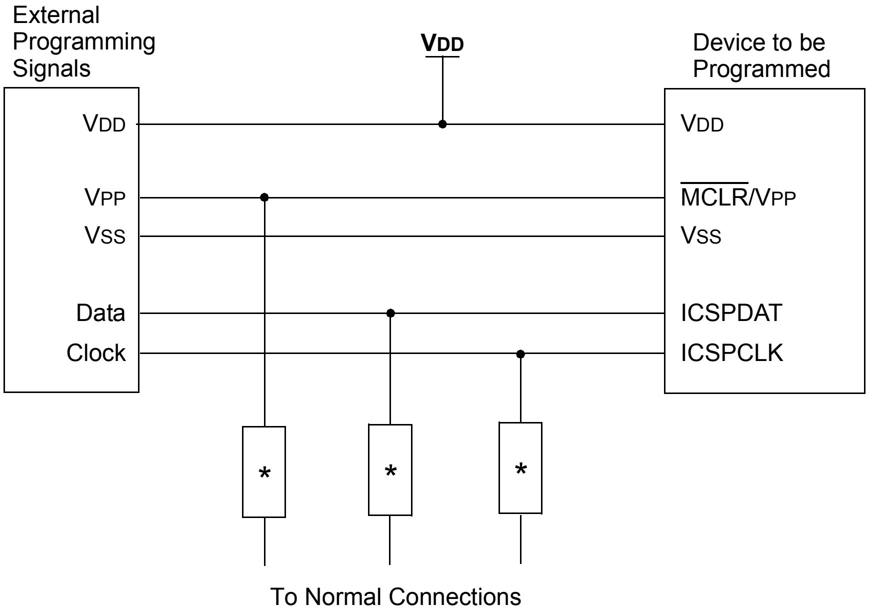
$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

FIGURE 35-2: PICkit™ PROGRAMMER STYLE CONNECTOR INTERFACE



* The 6-pin header (0.100" spacing) accepts 0.025" square pins.

FIGURE 35-3: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING



* Isolation devices (as required).