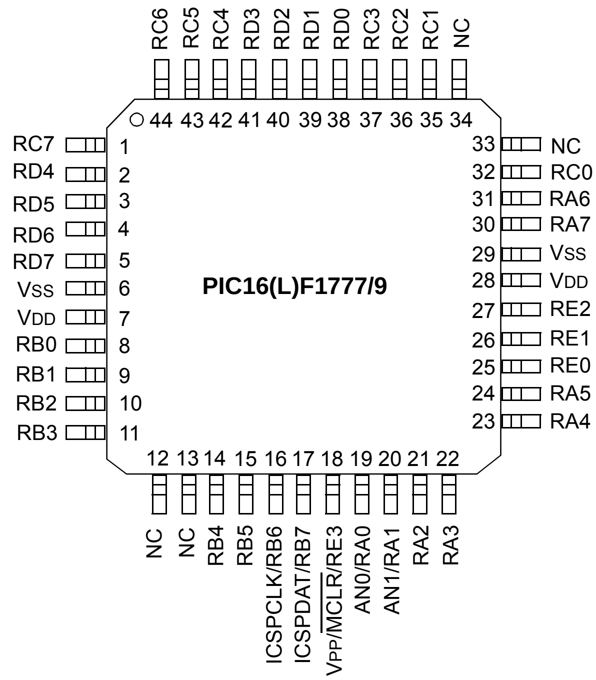
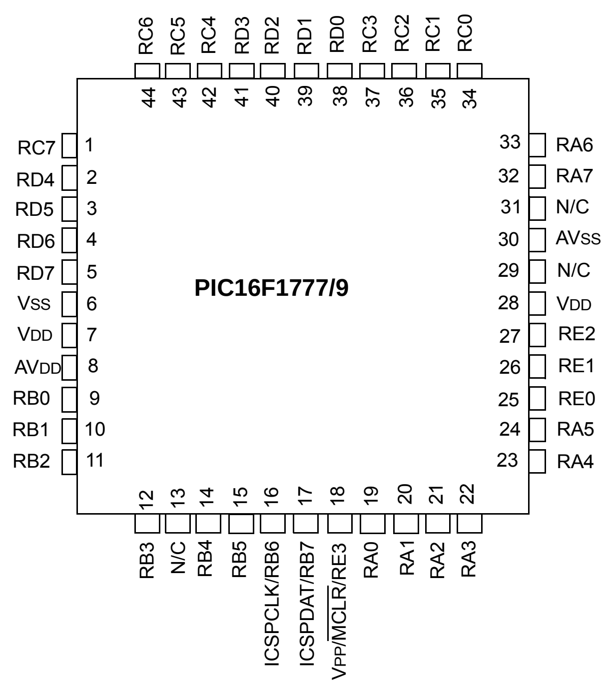
Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 17x10b; D/A 3x5b, 3x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1778-i-ss

FIGURE 5: 44-PIN TQFP (10x10 mm)



Note: See Table 4 for location of all peripheral functions.

FIGURE 6: 44-PIN QFN (8X8 mm)



Note: See Table 4 for location of all peripheral functions.

TABLE 3-4: PIC16(L)F1777/9 MEMORY MAP (BANKS 0-7)

Addr	BANK 0	Addr	BANK 1	Addr	BANK 2	Addr	BANK 3	Addr	BANK 4	Addr	BANK 5	Addr	BANK 6	Addr	BANK 7
000h–00Bh	Core Registers (Table 3-2)	080h–08Bh	Core Registers (Table 3-2)	100h–10Bh	Core Registers (Table 3-2)	180h–18Bh	Core Registers (Table 3-2)	200h–20Bh	Core Registers (Table 3-2)	280h–28Bh	Core Registers (Table 3-2)	300h–30Bh	Core Registers (Table 3-2)	380h–38Bh	Core Registers (Table 3-2)
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	ODCONB	30Dh	SLRCONB	38Dh	INLVLB
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC
00Fh	PORTD	08Fh	TRISD	10Fh	LATD	18Fh	ANSELD	20Fh	WPUD	28Fh	ODCOND	30Fh	SLRCOND	38Fh	INLVLD
010h	PORTE	090h	TRISE	110h	LATE	190h	ANSELE	210h	WPUE	290h	ODCONE	310h	SLRCONE	390h	INLVLE
011h	PIR1	091h	PIE1	111h	CMOUT	191h	PMADRL	211h	SSP1BUF	291h	CCPR1L	311h	CCPR8L	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1CON0	192h	PMADRH	212h	SSP1ADD	292h	CCPR1H	312h	CCPR8H	392h	IOCAN
013h	PIR3	093h	PIE3	113h	CM1CON1	193h	PMDATL	213h	SSP1MSK	293h	CCP1CON	313h	CCP8CON	393h	IOCAF
014h	PIR4	094h	PIE4	114h	CM1NSEL	194h	PMDATH	214h	SSP1STAT	294h	CCP1CAP	314h	CCP8CAP	394h	IOCBP
015h	PIR5	095h	PIE5	115h	CM1PSEL	195h	PMCON1	215h	SSP1CON1	295h	CCPR2L	315h	MD1CON0	395h	IOCBN
016h	PIR6	096h	PIE6	116h	CM2CON0	196h	PMCON2	216h	SSP1CON2	296h	CCPR2H	316h	MD1CON1	396h	IOCBF
017h	TMR0	097h	OPTION_REG	117h	CM2CON1	197h	VREGCON(1)	217h	SSP1CON3	297h	CCP2CON	317h	MD1SRC	397h	IOCCP
018h	TMR1L	098h	PCON	118h	CM2NSEL	198h	—	218h	—	298h	CCP2CAP	318h	MD1CARL	398h	IOCCN
019h	TMR1H	099h	WDTCON	119h	CM2PSEL	199h	RC1REG	219h	—	299h	CCPR7L	319h	MD1CARH	399h	IOCCF
01Ah	T1CON	09Ah	OSCTUNE	11Ah	CM3CON0	19Ah	TX1REG	21Ah	—	29Ah	CCPR7H	31Ah	—	39Ah	—
01Bh	T1GCON	09Bh	OSCCON	11Bh	CM3CON1	19Bh	SP1BRGL	21Bh	MD3CON0	29Bh	CCP7CON	31Bh	MD2CON0	39Bh	—
01Ch	TMR3L	09Ch	OSCSTAT	11Ch	CM3NSEL	19Ch	SP1BRGH	21Ch	MD3CON1	29Ch	CCP7CAP	31Ch	MD2CON1	39Ch	—
01Dh	TMR3H	09Dh	BORCON	11Dh	CM3PSEL	19Dh	RC1STA	21Dh	MD3SRC	29Dh	—	31Dh	MD2SRC	39Dh	IOCEP
01Eh	T3CON	09Eh	FVRCON	11Eh	—	19Eh	TX1STA	21Eh	MD3CARL	29Eh	CCPTMRS1	31Eh	MD2CARL	39Eh	IOCEN
01Fh	T3GCON	09Fh	ZCD1CON	11Fh	—	19Fh	BAUD1CON	21Fh	MD3CARH	29Fh	CCPTMRS2	31Fh	MD2CARH	39Fh	IOCEF
020h–06Fh	General Purpose Register 80 Bytes	0A0h–0EFh	General Purpose Register 80 Bytes	120h–16Fh	General Purpose Register 80 Bytes	1A0h–1EFh	General Purpose Register 80 Bytes	220h–26Fh	General Purpose Register 80 Bytes	2A0h–2EFh	General Purpose Register 80 Bytes	320h–36Fh	General Purpose Register 80 Bytes	3A0h–3EFh	General Purpose Register 80 Bytes
070h–07Fh	Common RAM 70h – 7Fh	0F0h–0FFh	Accesses 70h – 7Fh	170h–17Fh	Accesses 70h – 7Fh	1F0h–1FFh	Accesses 70h – 7Fh	270h–27Fh	Accesses 70h – 7Fh	2F0h–2FFh	Accesses 70h – 7Fh	370h–37Fh	Accesses 70h – 7Fh	3F0h–3FFh	Accesses 70h – 7Fh

Legend: ▒ = Unimplemented data memory locations, read as '0'.

Note 1: Unimplemented on PIC16LF1777/9.

TABLE 3-18: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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
Bank 3											
18Ch	ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	`--11 1111`	`--11 1111`
18Dh	ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	`--11 1111`	`--11 1111`
18Eh	ANSELC	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—	`1111 11--`	`1111 11--`
18Fh	ANSELD(3)	ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0	`1111 1111`	`1111 1111`
190h	ANSELE(3)	—	—	—	—	—	ANSE2	ANSE1	ANSE0	`---- -111`	`---- -111`
191h	PMADRL	Program Memory Address Register Low Byte								`0000 0000`	`0000 0000`
192h	PMADRH	—(1)	Program Memory Address Register High Byte							`1000 0000`	`1000 0000`
193h	PMDATL	Program Memory Read Data Register Low Byte								`xxxx xxxx`	`uuuu uuuu`
194h	PMDATH	—	—	Program Memory Read Data Register High Byte						`--xx xxxx`	`--uu uuuu`
195h	PMCON1	—(1)	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	`1000 x000`	`1000 q000`
196h	PMCON2	Program Memory Control Register 2								`0000 0000`	`0000 0000`
197h	VREGCON	—	—	—	—	—	—	VREGPM(2)	Reserved	`---- --01`	`---- --01`
198h	—	Unimplemented								—	—
198h	—	Unimplemented								—	—
199h	RC1REG	EUSART Receive Data Register								`0000 0000`	`0000 0000`
19Ah	TX1REG	EUSART Transmit Data Register								`0000 0000`	`0000 0000`
19Bh	SP1BRGL	SP1BRG<7:0>								`0000 0000`	`0000 0000`
19Ch	SP1BRGH	SP1BRG<15:8>								`0000 0000`	`0000 0000`
19Dh	RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	`0000 0000`	`0000 0000`
19Eh	TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	`0000 0010`	`0000 0010`
19Fh	BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	`01-0 0-00`	`01-0 0-00`
199h — 19Fh	—	Unimplemented								—	—

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: Unimplemented, read as '1'.
2: Unimplemented on PIC16LF1777/8/9.
3: Unimplemented on PIC16(L)F1778.

TABLE 3-18: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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
Bank 11											
58Ch	—	Unimplemented								—	—
58Dh	DACLD	—	—	DAC6LD(3)	DAC5LD	—	—	DAC2LD	DAC1LD	--00 --00	--00 --00
58Eh	DAC1CON0	EN	FM	OE1	OE2	PSS<1:0>		NSS<1:0>		0000 0000	0000 0000
58Fh	DAC1REFL	REF<7:0>								00000 0000	0000 0000
590h	DAC1REFH	REF<15:8>								00000 0000	0000 0000
591h	DAC2CON0	EN	FM	OE1	OE2	PSS<1:0>		NSS<1:0>		0000 0000	0000 0000
592h	DAC2REFL	REF<7:0>								00000 0000	0000 0000
593h	DAC2REFH	REF<15:8>								00000 0000	0000 0000
594h	DAC3CON0	EN	—	OE1	OE2	PSS<1:0>		NSS<1:0>		0-00 0000	0-00 0000
595h	DAC3REF	—	—	—	REF<4:0>					---0 0000	---0 0000
596h	DAC4CON0	EN	—	OE1	OE2	PSS<1:0>		NSS<1:0>		0-00 0000	0-00 0000
597h	DAC4REF	—	—	—	REF<4:0>					---0 0000	0000 0000
598h	DAC5CON0	EN	FM	OE1	OE2	PSS<1:0>		NSS<1:0>		0000 0000	0000 0000
599h	DAC5REFL	REF<7:0>								00000 0000	0000 0000
59Ah	DAC5REFH	REF<15:8>								00000 0000	0000 0000
59Bh	DAC6CON0(3)	EN	FM	OE1	OE2	PSS<1:0>		NSS<1:0>		0000 0000	0000 0000
59Ch	DAC6REFL(3)	REF<7:0>								00000 0000	0000 0000
59Dh	DAC6REFH(3)	REF<15:8>								00000 0000	0000 0000
59Eh	DAC7CON0	EN	—	OE1	OE2	PSS<1:0>		NSS<1:0>		0-00 0000	0-00 0000
59Fh	DAC7REF	—	—	—	REF<4:0>					---0 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: Unimplemented, read as '1'.
2: Unimplemented on PIC16LF1777/8/9.
3: Unimplemented on PIC16(L)F1778.

10.2.3 ERASING FLASH PROGRAM MEMORY

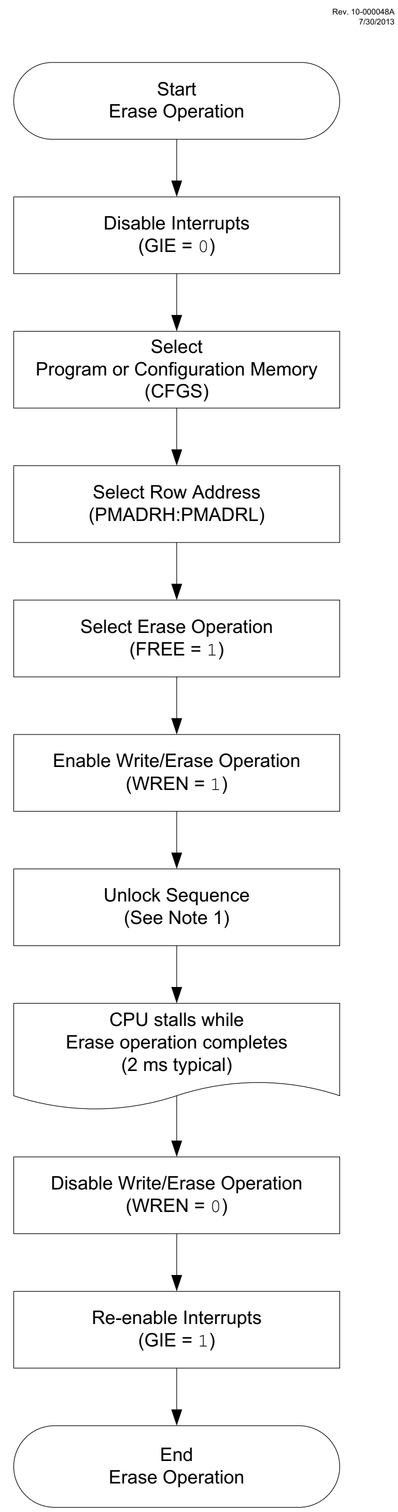
While executing code, program memory can only be erased by rows. To erase a row:

1. Load the PMADRH:PMADRL register pair with any address within the row to be erased.
2. Clear the CFGS bit of the PMCON1 register.
3. Set the FREE and WREN bits of the PMCON1 register.
4. Write 55h, then AAh, to PMCON2 (Flash programming unlock sequence).
5. Set control bit WR of the PMCON1 register to begin the erase operation.

See Example 10-2.

After the "`BSF PMCON1,WR`" instruction, the processor requires two cycles to set up the erase operation. The user must place two `NOP` instructions immediately following the WR bit set instruction. The processor will halt internal operations for the typical 2 ms erase time. This is not Sleep mode as the clocks and peripherals will continue to run. After the erase cycle, the processor will resume operation with the third instruction after the PMCON1 write instruction.

FIGURE 10-4: FLASH PROGRAM MEMORY ERASE FLOWCHART



Note 1: See Figure 10-3.

10.6 Register Definitions: Flash Program Memory Control

REGISTER 10-1: PMDATL: PROGRAM MEMORY DATA LOW BYTE REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PMDAT<7:0>							
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 7-0 **PMDAT<7:0>**: Read/write value for Least Significant bits of program memory

REGISTER 10-2: PMDATH: PROGRAM MEMORY DATA HIGH BYTE REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	PMDAT<13:8>					
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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **PMDAT<13:8>**: Read/write value for Most Significant bits of program memory

REGISTER 10-3: PMADRL: PROGRAM MEMORY ADDRESS LOW BYTE 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
PMADR<7:0>							
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 7-0 **PMADR<7:0>**: Specifies the Least Significant bits for program memory address

11.0 I/O PORTS

Each port has six standard registers for its operation. These registers are:

- TRISx registers (data direction)
- PORTx registers (reads the levels on the pins of the device)
- LATx registers (output latch)
- INLVLx (input level control)
- ODCONx registers (open-drain)
- SLRCONx registers (slew rate

Some ports may have one or more of the following additional registers. These registers are:

- ANSELx (analog select)
- WPUx (weak pull-up)

In general, when a peripheral is enabled on a port pin, that pin cannot be used as a general purpose output. However, the pin can still be read.

TABLE 11-1: PORT AVAILABILITY PER DEVICE

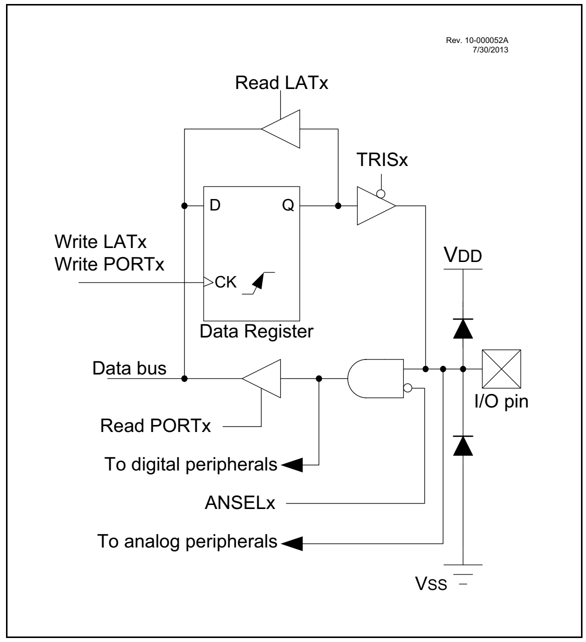
Device	PORTA	PORTB	PORTC	PORTD	PORTE
PIC16(L)F1778	●	●	●		●
PIC16(L)F1777/9	●	●	●	●	●

The Data Latch (LATx registers) is useful for read-modify-write operations on the value that the I/O pins are driving.

A write operation to the LATx register has the same effect as a write to the corresponding PORTx register. A read of the LATx register reads of the values held in the I/O PORT latches, while a read of the PORTx register reads the actual I/O pin value.

Ports that support analog inputs have an associated ANSELx register. When an ANSEL bit is set, the digital input buffer associated with that bit is disabled. Disabling the input buffer prevents analog signal levels on the pin between a logic high and low from causing excessive current in the logic input circuitry. A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 11-1.

FIGURE 11-1: GENERIC I/O PORT OPERATION



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.

TABLE 12-3: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE (CONTINUED)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
CLC1IN1PPS	—	—	CLCIN1PPS<5:0>						205
CLC1IN2PPS	—	—	CLCIN2PPS<5:0>						205
CLC1IN3PPS	—	—	CLCIN3PPS<5:0>						205
ADCACTPPS	—	—	ADCACTPPS<5:0>						205
SSPCLKPPS	—	—	SSPCLKPPS<5:0>						205
SSPDATPPS	—	—	SSPDATPPS<5:0>						205
SSPSSPPS	—	—	SSPSSPPS<5:0>						205
RXPPS	—	—	RXPPS<5:0>						205
CKPPS	—	—	CKPPS<5:0>						205

Legend: — = unimplemented, read as ‘0’. Shaded cells are unused by the PPS module.
Note 1: PIC16(L)F1777/9 only.

14.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of V_{DD}, with 1.024V, 2.048V or 4.096V selectable output levels. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- ADC positive reference
- Comparator positive input
- Digital-to-Analog Converter (DAC)

The FVR can be enabled by setting the FVREN bit of the FVRCON register.

14.1 Independent Gain Amplifiers

The output of the FVR supplied to the ADC, Comparators, and DAC is routed through two independent programmable gain amplifiers. Each amplifier can be programmed for a gain of 1x, 2x or 4x, to produce the three possible voltage levels.

The ADFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module. Reference **Section 16.0 "Analog-to-Digital Converter (ADC) Module"** for additional information.

The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the DAC and comparator module. Reference **Section 17.0 "5-Bit Digital-to-Analog Converter (DAC) Module"** and **Section 19.0 "Comparator Module"** for additional information.

14.2 FVR Stabilization Period

When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the FVRRDY bit of the FVRCON register will be set. See Figure 37-77: FVR Stabilization Period, PIC16LF1777/8/9 Only..

14.3 FVR Buffer Stabilization Period

When either FVR Buffer1 or Buffer2 is enabled then the buffer amplifier circuits require 30 μs to stabilize.This stabilization time is required even when the FVR is already operating and stable.

17.0 5-BIT DIGITAL-TO-ANALOG CONVERTER (DAC) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 32 selectable output levels.

The input of the DAC can be connected to:

- External VREF pins
- VDD supply voltage
- FVR (Fixed Voltage Reference)

The output of the DAC can be configured to supply a reference voltage to the following:

- Comparator positive input
- Operational amplifier inverting and non-inverting inputs
- ADC input channel
- DACxOUTx pin

TABLE 17-1: AVAILABLE 5-BIT DACS

Device	DAC3	DAC4	DAC7	DAC8
PIC16(L)F1778	●	●	●	
PIC16(L)F1777/9	●	●	●	●

The Digital-to-Analog Converter (DAC) is enabled by setting the EN bit of the DACxCON0 register.

17.1 Output Voltage Selection

The DAC has 32 voltage level ranges. The 32 levels are set with the REF<4:0> bits of the DACxREF register.

The DAC output voltage is determined by Equation 17-1:

EQUATION 17-1: DAC OUTPUT VOLTAGE

IF DACxEN = 1

$$V_{OUT} = \left((V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACxR[4:0]}{2^5} \right) + V_{SOURCE-}$$

$V_{SOURCE+} = V_{DD}$, V_{REF}, or FVR BUFFER 2

$V_{SOURCE-} = V_{SS}$

17.2 Ratiometric Output Level

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

The value of the individual resistors within the ladder can be found in Table 36-20: 10-bit Digital-to-Analog Converter (DAC) Specifications.

17.3 DAC Voltage Reference Output

The DAC voltage can be output to the DACxOUTx pin by setting the OEx bit of the DACxCON0 register. Selecting the DAC voltage for output on the DACxOUTx pin automatically overrides the digital output buffer and digital input threshold detector functions of that pin. Reading the DACxOUTx pin when it has been configured for DAC voltage output will always return a ‘0’.

Due to the limited current drive capability, a buffer must be used on the DAC voltage output for external connections to the DACxOUTx pin. Figure 17-2 shows an example buffering technique.

21.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 8-bit prescaler (independent of Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt-on-overflow
- TMR0 can be used to gate Timer1

Figure 21-1 is a block diagram of the Timer0 module.

21.1 Timer0 Operation

The Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

21.1.1 8-BIT TIMER MODE

The Timer0 module will increment every instruction cycle, if used without a prescaler. 8-bit Timer mode is selected by clearing the TMR0CS bit of the OPTION_REG register.

When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

Note:	The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

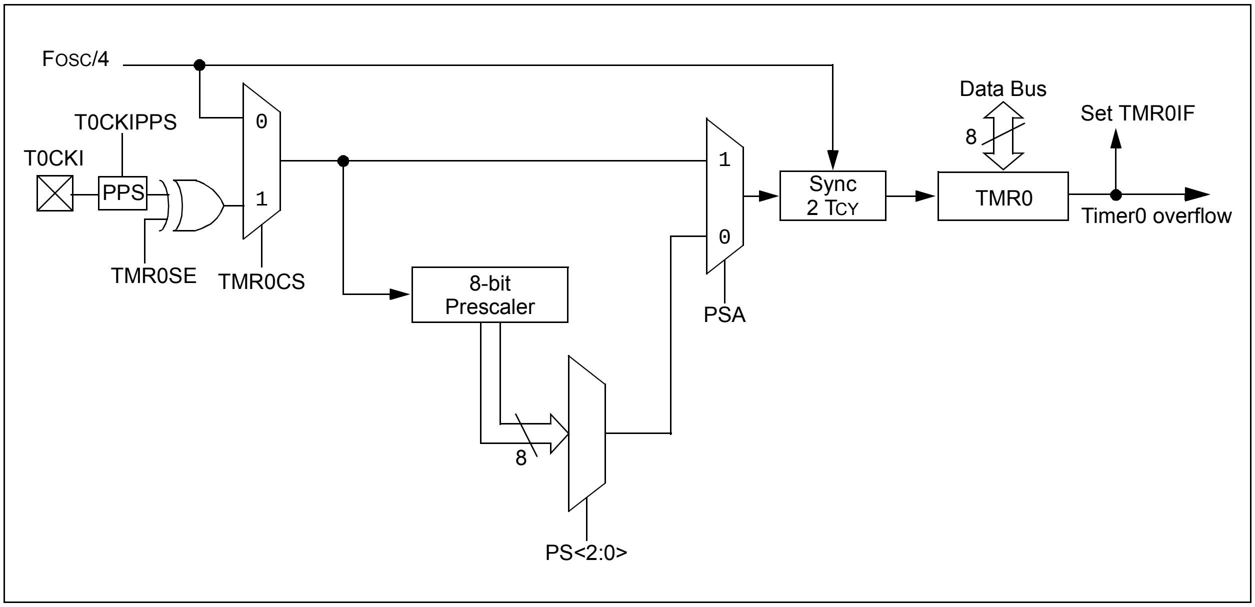
21.1.2 8-BIT COUNTER MODE

In 8-Bit Counter mode, the Timer0 module will increment on every rising or falling edge of the T0CKI pin.

8-Bit Counter mode using the T0CKI pin is selected by setting the TMR0CS bit in the OPTION_REG register to '1'.

The rising or falling transition of the incrementing edge for either input source is determined by the TMR0SE bit in the OPTION_REG register.

FIGURE 21-1: BLOCK DIAGRAM OF THE TIMER0



23.6.7 EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODE

In Edge-Triggered Hardware Limit One-Shot modes the timer starts on the first external signal edge after the ON bit is set and resets on all subsequent edges. Only the first edge after the ON bit is set is needed to start the timer. The counter will resume counting automatically two clocks after all subsequent external Reset edges. Edge triggers are as follows:

- Rising edge start and Reset (MODE<4:0> = `01100`)
- Falling edge start and Reset (MODE<4:0> = `01101`)

The timer resets and clears the ON bit when the timer value matches the PRx period value. External signal edges will have no effect until after software sets the ON bit. Figure 23-10 illustrates the rising edge hardware limit one-shot operation.

When this mode is used in conjunction with the CCP then the first starting edge trigger, and all subsequent Reset edges, will activate the PWM drive. The PWM drive will deactivate when the timer matches the CCPRx pulse-width value and stay deactivated until the timer halts at the PRx period match unless an external signal edge resets the timer before the match occurs.

24.6 CCP/PWM Clock Selection

This device 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 four 8-bit timers with auto-reload (Timer2, Timer4, Timer6 and Timer8). The PWM mode on the CCP and 10-bit PWM modules can use any of these timers.

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

24.7 Register Definitions: CCP/PWM Timers Control

REGISTER 24-5: CCPTMRS1: PWM TIMER SELECTION CONTROL REGISTER 1

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
C8TSEL<1:0>(1)		C7TSEL<1:0>		C2TSEL<1:0>		C1TSEL<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 7-6 **C8TSEL<1:0>:** CCP8 (PWM8) Timer Selection bits(1)
- `11` = CCP8 is based off Timer8 in PWM mode
- `10` = CCP8 is based off Timer6 in PWM mode
- `01` = CCP8 is based off Timer4 in PWM mode
- `00` = CCP8 is based off Timer2 in PWM mode

bit 5-4 **C7TSEL<1:0>:** CCP7 (PWM7) Timer Selection bits
- `11` = CCP7 is based off Timer8 in PWM mode
- `10` = CCP7 is based off Timer6 in PWM mode
- `01` = CCP7 is based off Timer4 in PWM mode
- `00` = CCP7 is based off Timer2 in PWM mode

bit 3-2 **C2TSEL<1:0>:** CCP2 (PWM2) Timer Selection bits
- `11` = CCP2 is based off Timer8 in PWM mode
- `10` = CCP2 is based off Timer6 in PWM mode
- `01` = CCP2 is based off Timer4 in PWM mode
- `00` = CCP2 is based off Timer2 in PWM mode

bit 1-0 **C1TSEL<1:0>:** CCP1 (PWM1) Timer Selection bits
- `11` = CCP1 is based off Timer8 in PWM mode
- `10` = CCP1 is based off Timer6 in PWM mode
- `01` = CCP1 is based off Timer4 in PWM mode
- `00` = CCP1 is based off Timer2 in PWM mode

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

26.2 PWM Modes

PWM modes are selected with MODE<1:0> bits of the PWMxCON register (Register 26-1).

In all PWM modes an offset match event can also be used to synchronize the PWMxTMR in three offset modes. See **Section 26.3 "Offset Modes"** for more information.

26.2.1 STANDARD MODE

The Standard mode (MODE = `00`) selects a single phase PWM output. The PWM output in this mode is determined by when the period, duty cycle, and phase counts match the PWMxTMR value. The start of the duty cycle occurs on the phase match and the end of the duty cycle occurs on the duty cycle match. The period match resets the timer. The offset match can also be used to synchronize the PWMxTMR in the offset modes. See **Section 26.3 "Offset Modes"** for more information.

Equation 26-1 is used to calculate the PWM period in Standard mode.

Equation 26-2 is used to calculate the PWM duty cycle ratio in Standard mode.

EQUATION 26-1: PWM PERIOD IN STANDARD MODE

$$Period = \frac{(PWMxPR + 1) \cdot Prescale}{PWM\_clock}$$

EQUATION 26-2: PWM DUTY CYCLE IN STANDARD MODE

$$Duty\ Cycle = \frac{(PWMxDC - PWMxPH)}{PWMxPR + 1}$$

A detailed timing diagram for Standard mode is shown in Figure 26-4.

26.2.2 SET ON MATCH MODE

The Set On Match mode (MODE = `01`) generates an active output when the phase count matches the PWMxTMR value. The output stays active until the OUT bit of the PWMxCON register is cleared or the PWM module is disabled. The duty cycle count has no effect in this mode. The period count only determines the maximum PWMxTMR value above which no phase matches can occur.

The PWMxOUT bit can be used to set or clear the output of the PWM in this mode. Writes to this bit will take place on the next rising edge of the PWM_clock after the bit is written.

A detailed timing diagram for Set On Match is shown in Figure 26-5.

26.2.3 TOGGLE ON MATCH MODE

The Toggle On Match mode (MODE = `10`) generates a 50% duty cycle PWM with a period twice as long as that computed for the standard PWM mode. Duty cycle count has no effect in this mode. The phase count determines how many PWMxTMR periods after a period event the output will toggle.

Writes to the OUT bit of the PWMxCON register will have no effect in this mode.

A detailed timing diagram for Toggle On Match is shown in Figure 26-6.

26.2.4 CENTER ALIGNED MODE

The Center Aligned mode (MODE = `11`) generates a PWM waveform that is centered in the period. In this mode the period is two times the PWMxPR count. The PWMxTMR counts up to the period value then counts back down to 0. The duty cycle count determines both the start and end of the active PWM output. The start of the duty cycle occurs at the match event when PWMxTMR is incrementing and the duty cycle ends at the match event when PWMxTMR is decrementing. The incrementing match value is the period count minus the duty cycle count. The decrementing match value is the incrementing match value plus 1.

Equation 26-3 is used to calculate the PWM period in Center Aligned mode.

EQUATION 26-3: PWM PERIOD IN CENTER ALIGNED MODE

$$Period = \frac{(PWMxPR + 1) \cdot 2 \cdot Prescale}{PWM\_clock}$$

Equation 26-4 is used to calculate the PWM duty cycle ratio in Center Aligned mode

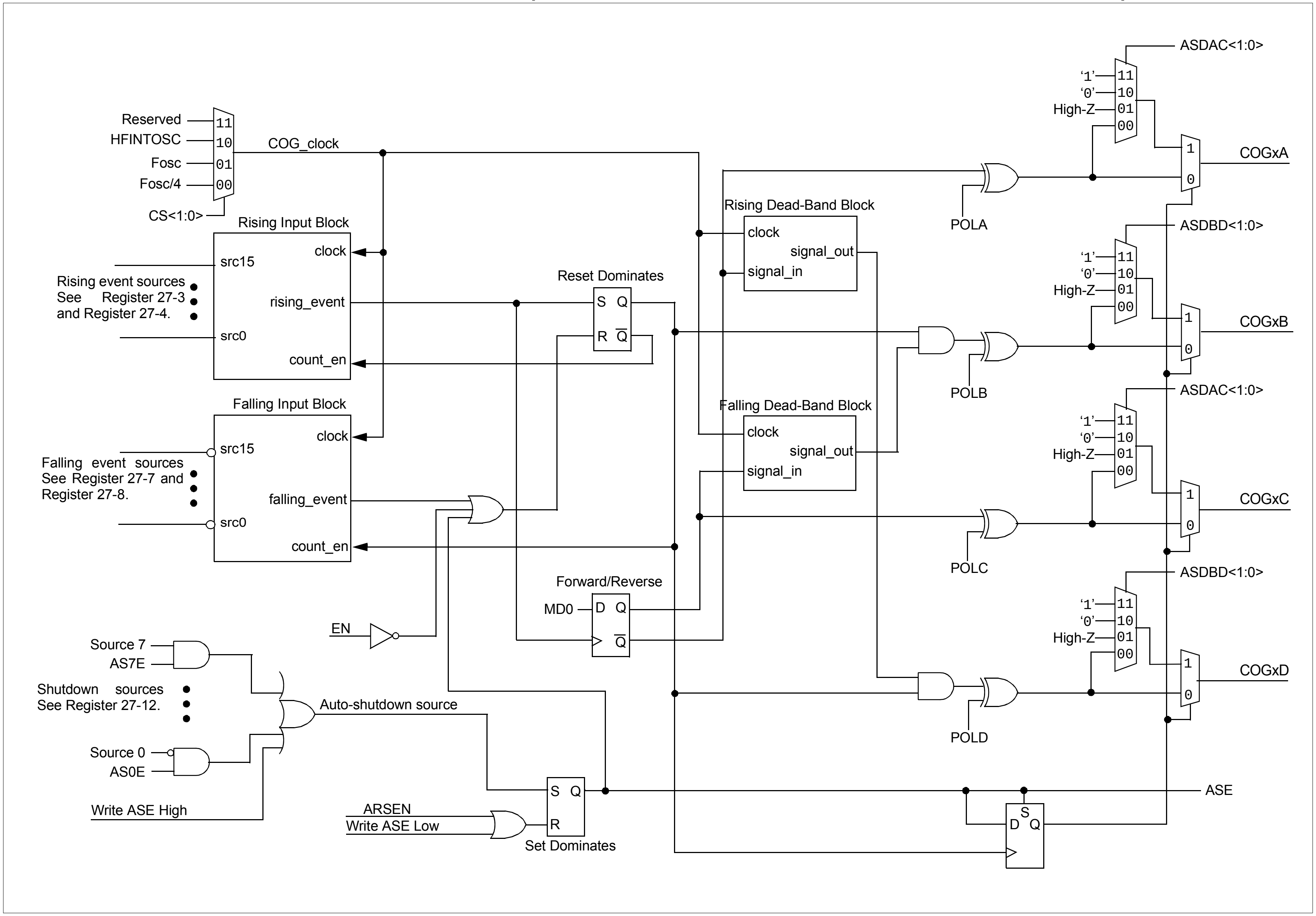
EQUATION 26-4: PWM DUTY CYCLE IN CENTER ALIGNED MODE

$$Duty\ Cycle = \frac{PWMxDC \cdot 2}{(PWMxPR + 1) \cdot 2}$$

Writes to PWMxOUT will have no effect in this mode.

A detailed timing diagram for Center Aligned mode is shown in Figure 26-7.

FIGURE 27-4: SIMPLIFIED COG BLOCK DIAGRAM (FULL-BRIDGE MODES, FORWARD: MD = 2, REVERSE: MD = 3)



32.2 SPI Mode Overview

The Serial Peripheral Interface (SPI) bus is a synchronous serial data communication bus that operates in Full-Duplex mode. Devices communicate in a master/slave environment where the master device initiates the communication. A slave device is controlled through a Chip Select known as Slave Select.

The SPI bus specifies four signal connections:

- Serial Clock (SCK)
- Serial Data Out (SDO)
- Serial Data In (SDI)
- Slave Select ($\overline{SS}$)

Figure 32-1 shows the block diagram of the MSSP module when operating in SPI mode.

The SPI bus operates with a single master device and one or more slave devices. When multiple slave devices are used, an independent Slave Select connection is required from the master device to each slave device.

Figure 32-4 shows a typical connection between a master device and multiple slave devices.

The master selects only one slave at a time. Most slave devices have tri-state outputs so their output signal appears disconnected from the bus when they are not selected.

Transmissions involve two shift registers, eight bits in size, one in the master and one in the slave. With either the master or the slave device, data is always shifted out one bit at a time, with the Most Significant bit (MSb) shifted out first. At the same time, a new Least Significant bit (LSb) is shifted into the same register.

Figure 32-5 shows a typical connection between two processors configured as master and slave devices.

Data is shifted out of both shift registers on the programmed clock edge and latched on the opposite edge of the clock.

The master device transmits information out on its SDO output pin which is connected to, and received by, the slave's SDI input pin. The slave device transmits information out on its SDO output pin, which is connected to, and received by, the master's SDI input pin.

To begin communication, the master device first sends out the clock signal. Both the master and the slave devices should be configured for the same clock polarity.

The master device starts a transmission by sending out the MSb from its shift register. The slave device reads this bit from that same line and saves it into the LSb position of its shift register.

During each SPI clock cycle, a full-duplex data transmission occurs. This means that while the master device is sending out the MSb from its shift register (on its SDO pin) and the slave device is reading this bit and saving it as the LSb of its shift register, that the slave device is also sending out the MSb from its shift register (on its SDO pin) and the master device is reading this bit and saving it as the LSb of its shift register.

After eight bits have been shifted out, the master and slave have exchanged register values.

If there is more data to exchange, the shift registers are loaded with new data and the process repeats itself.

Whether the data is meaningful or not (dummy data), depends on the application software. This leads to three scenarios for data transmission:

- Master sends useful data and slave sends dummy data.
- Master sends useful data and slave sends useful data.
- Master sends dummy data and slave sends useful data.

Transmissions may involve any number of clock cycles. When there is no more data to be transmitted, the master stops sending the clock signal and it deselects the slave.

Every slave device connected to the bus that has not been selected through its slave select line must disregard the clock and transmission signals and must not transmit out any data of its own.

32.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSPxCON2 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 32-30).

32.6.8.1 WCOL Status Flag

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

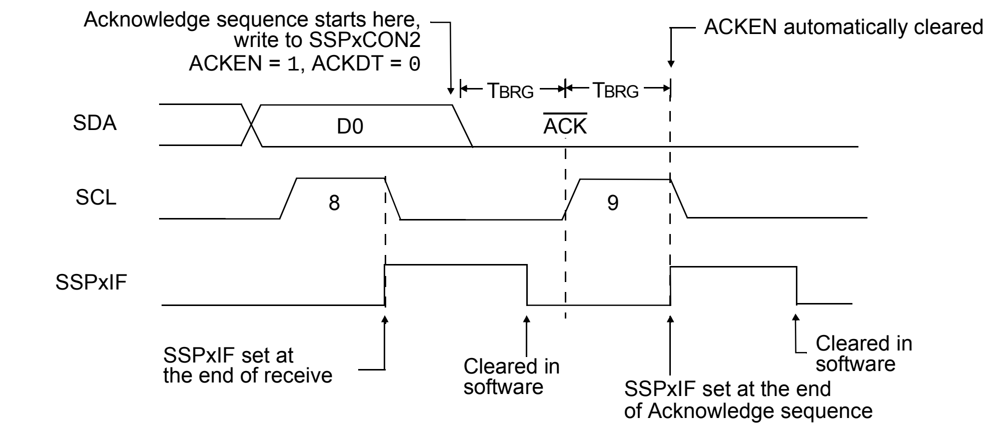
32.6.9 STOP CONDITION TIMING

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 SSPxCON2 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 SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 32-31).

32.6.9.1 WCOL Status Flag

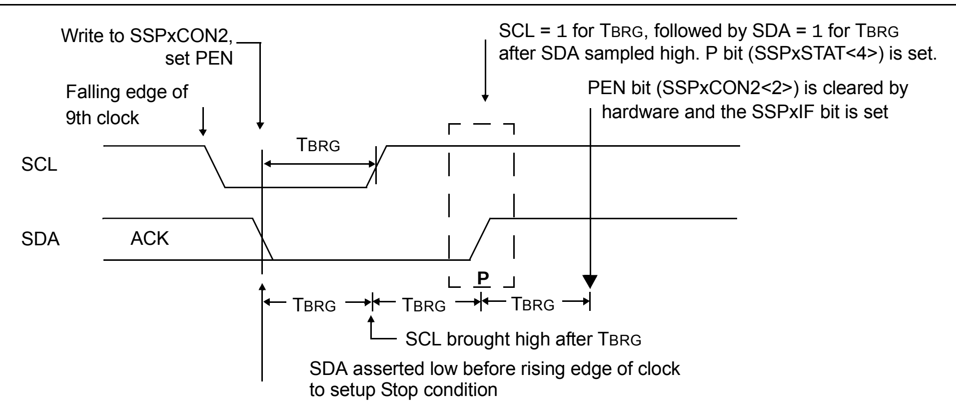
If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

FIGURE 32-30: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one Baud Rate Generator period.

FIGURE 32-31: STOP CONDITION RECEIVE OR TRANSMIT MODE



Note: TBRG = one Baud Rate Generator period.

TABLE 33-3: BAUD RATE FORMULAS

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	FOSC/[64 (n+1)]
0	0	1	8-bit/Asynchronous	FOSC/[16 (n+1)]
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	FOSC/[4 (n+1)]
1	0	x	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

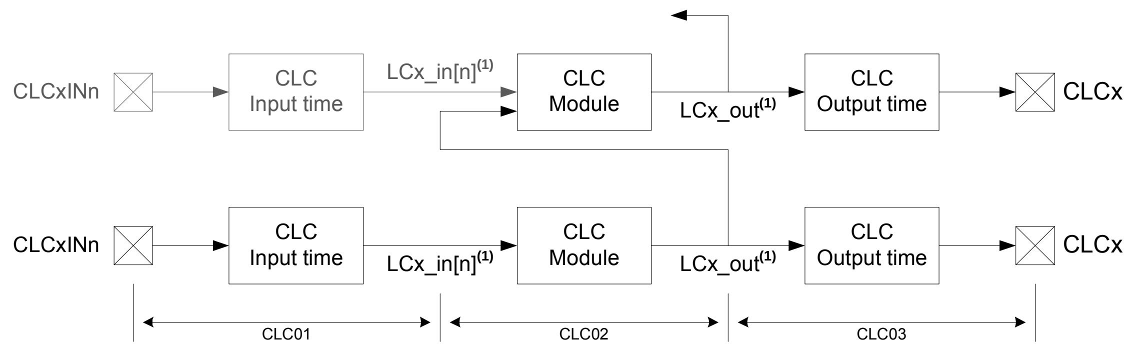
Legend: x = Don't care, n = value of SPxBRGH:SPxBRGL register pair.

TABLE 33-4: SUMMARY OF REGISTERS ASSOCIATED WITH THE BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	505
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	504
SP1BRGL	SP1BRG<7:0>								506
SP1BRGH	SP1BRG<15:8>								506
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	503

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for the Baud Rate Generator.
* Page provides register information.

FIGURE 36-12: CLC PROPAGATION TIMING



Note 1: See Figure 28-1 to identify specific CLC signals.

TABLE 36-14: CONFIGURATION LOGIC CELL (CLC) CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)
Operating temperature -40°C ≤ TA ≤ +125°C

Param. No.	Sym.	Characteristic		Min.	Typ†	Max.	Units	Conditions
CLC01*	TCLCIN	CLC input time		—	7	OS17	ns	**(Note 1)**
CLC02*	TCLC	CLC module input to output progagation time		— —	24 12	— —	ns ns	VDD = 1.8V VDD > 3.6V
CLC03*	TCLCOUT	CLC output time	Rise Time	—	OS18	—	ns	**(Note 1)**
			Fall Time	—	OS19	—	ns	**(Note 1)**
CLC04*	FCLCMAX	CLC maximum switching frequency		—	45	—	MHz	

* 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.

Note 1: See Table 36-10 for OS17, OS18 and OS19 rise and fall times.