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	18
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1619t-i-ss

TABLE 3-3: PIC16(L)F1619 MEMORY MAP, BANK 0-7

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	Core Registers (Table 3-1)	080h	Core Registers (Table 3-1)	100h	Core Registers (Table 3-1)	180h	Core Registers (Table 3-1)	200h	Core Registers (Table 3-1)	280h	Core Registers (Table 3-1)	300h	Core Registers (Table 3-1)	380h	Core Registers (Table 3-1)
00Bh		08Bh		10Bh		18Bh		20Bh		28Bh		30Bh		38Bh	
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Eh	WPUB	28Eh	ODCONB	30Eh	SLRCONB	38Eh	INLVLB
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	PIR1	090h	PIE1	110h	—	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR2	091h	PIE2	111h	CM1CON0	191h	PMADRL	211h	SSP1BUF	291h	CCP1RL	311h	—	391h	IOCAP
012h	PIR3	092h	PIE3	112h	CM1CON1	192h	PMADRH	212h	SSP1ADD	292h	CCP1RH	312h	—	392h	IOCAN
013h	PIR4	093h	PIE4	113h	CM2CON0	193h	PMDATL	213h	SSP1MSK	293h	CCP1CON	313h	—	393h	IOCAF
014h	PIR5	094h	PIE5	114h	CM2CON1	194h	PMDATH	214h	SSP1STAT	294h	CCP1CAP	314h	—	394h	IOCBP
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	PMCON1	215h	SSP1CON	295h	—	315h	—	395h	IOCBN
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—	396h	IOCBF
017h	TMR1H	097h	—	117h	FVRCON	197h	VREGCON	217h	SSP1CON3	297h	—	317h	—	397h	IOCCP
018h	T1CON	098h	OSCTUNE	118h	DAC1CON0	198h	—	218h	—	298h	CCP2RL	318h	—	398h	IOCCN
019h	T1GCON	099h	OSCCON	119h	DAC1CON1	199h	RC1REG	219h	—	299h	CCP2RH	319h	—	399h	IOCCF
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TX1REG	21Ah	—	29Ah	CCP2CON	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SP1BRGL	21Bh	—	29Bh	CCP2CAP	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	ZCD1CON	19Ch	SP1BRGH	21Ch	—	29Ch	—	31Ch	—	39Ch	—
01Dh	T2HLT	09Dh	ADCON0	11Dh	—	19Dh	RC1STA	21Dh	—	29Dh	—	31Dh	—	39Dh	—
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	—	19Eh	TX1STA	21Eh	—	29Eh	CCPTMRS	31Eh	—	39Eh	—
01Fh	T2RST	09Fh	ADCON2	11Fh	—	19Fh	BAUD1CON	21Fh	—	29Fh	—	31Fh	—	39Fh	—
020h	General Purpose Register 96 Bytes	0A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	General Purpose Register 80 Bytes	220h	General Purpose Register 80 Bytes	2A0h	General Purpose Register 80 Bytes	320h	General Purpose Register 80 Bytes	3A0h	General Purpose Register 80 Bytes
07Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
		0F0h	Common RAM (Accesses 70h – 7Fh)	170h	Common RAM (Accesses 70h – 7Fh)	1F0h	Common RAM (Accesses 70h – 7Fh)	270h	Common RAM (Accesses 70h – 7Fh)	2F0h	Common RAM (Accesses 70h – 7Fh)	370h	Common RAM (Accesses 70h – 7Fh)	3F0h	Common RAM (Accesses 70h – 7Fh)
		0FFh		17Fh		1FFh		27Fh		2FFh		37Fh		3FFh	

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

TABLE 3-14: 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 12											
60Ch	PID1Z2L	Z2<7:0>								0000 0000	0000 0000
60Dh	PID1Z2H	Z2<15:8>								0000 0000	0000 0000
60Eh	PID1Z2U	—	—	—	—	—	—	—	Z216	---- ---0	---- ---0
60Fh	PID1ACCLL	ACC<7:0>								0000 0000	0000 0000
610h	PID1ACCLH	ACC<15:8>								0000 0000	0000 0000
611h	PID1ACCHL	ACC<23:16>								0000 0000	0000 0000
612h	PID1ACCHH	ACC<31:24>								0000 0000	0000 0000
613h	PID1ACCU	—	—	—	—	—	ACC<34:32>			---- -000	---- -000
614h	PID1CON	EN	BUSY	—	—	—	MODE<2:0>			00-- 0000	00-- 0000
615h	—	Unimplemented								—	—
616h	—	Unimplemented								—	—
617h	PWM3DCL	DC<1:0>		—	—	—	—	—	—	xx-- ----	xx-- ----
618h	PWM3DCH	DC<9:2>								xxxx xxxx	xxxx xxxx
619h	PWM3CON	EN	—	OUT	POL	—	—	—	—	0-x0 ----	0-x0 ----
61Ah	PWM4DCL	DC<1:0>		—	—	—	—	—	—	xx-- ----	xx-- ----
61Bh	PWM4DCH	DC<9:2>								xxxx xxxx	xxxx xxxx
61Ch	PWM4CON	EN	—	OUT	POL	—	—	—	—	0-x0 ----	0-x0 ----
61Dh to 61Fh	—	Unimplemented								—	—

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

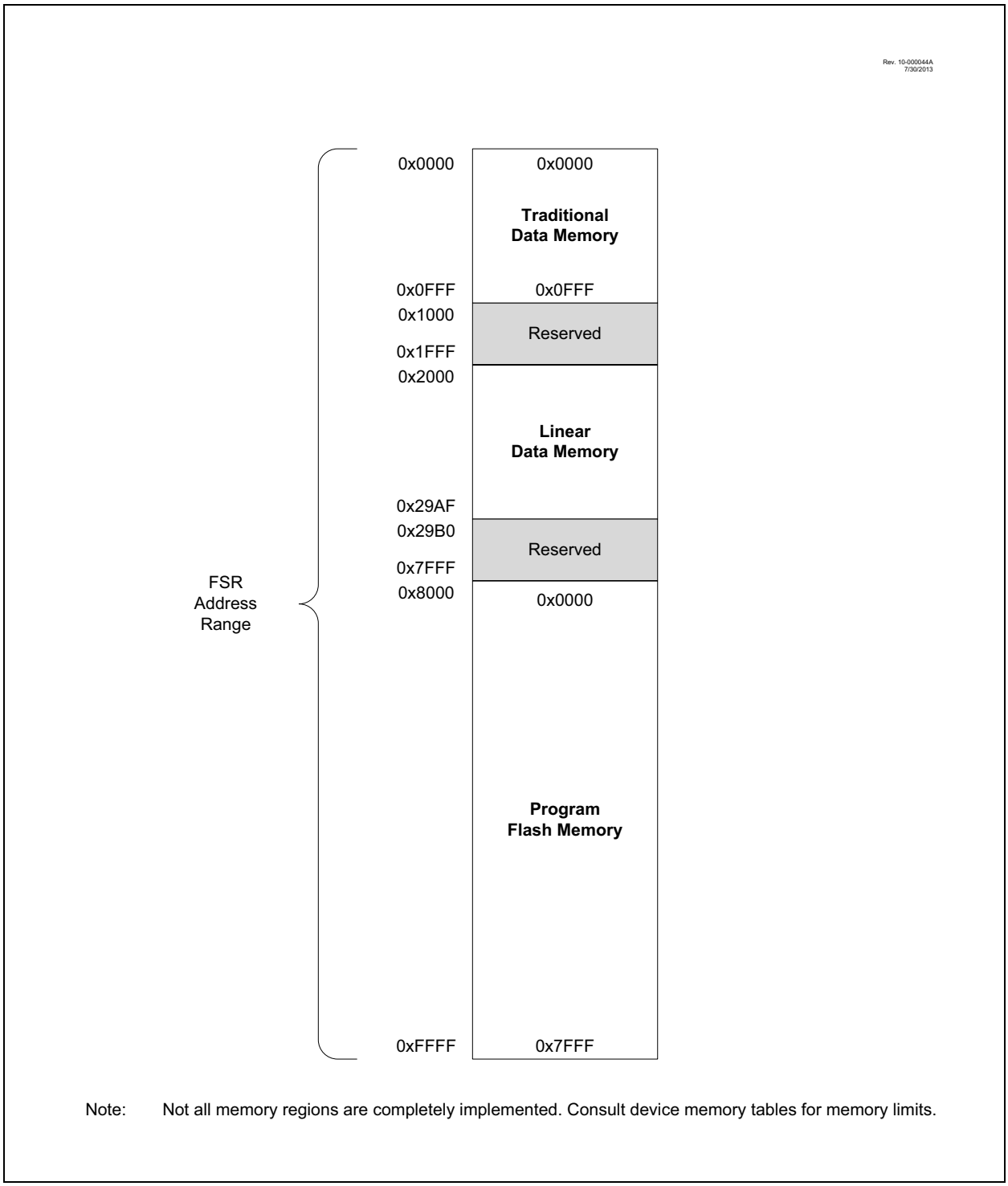
Note 1: PIC16F1615/9 only.

2: Unimplemented, read as '1'.

3: PIC16(L)F1615 only.

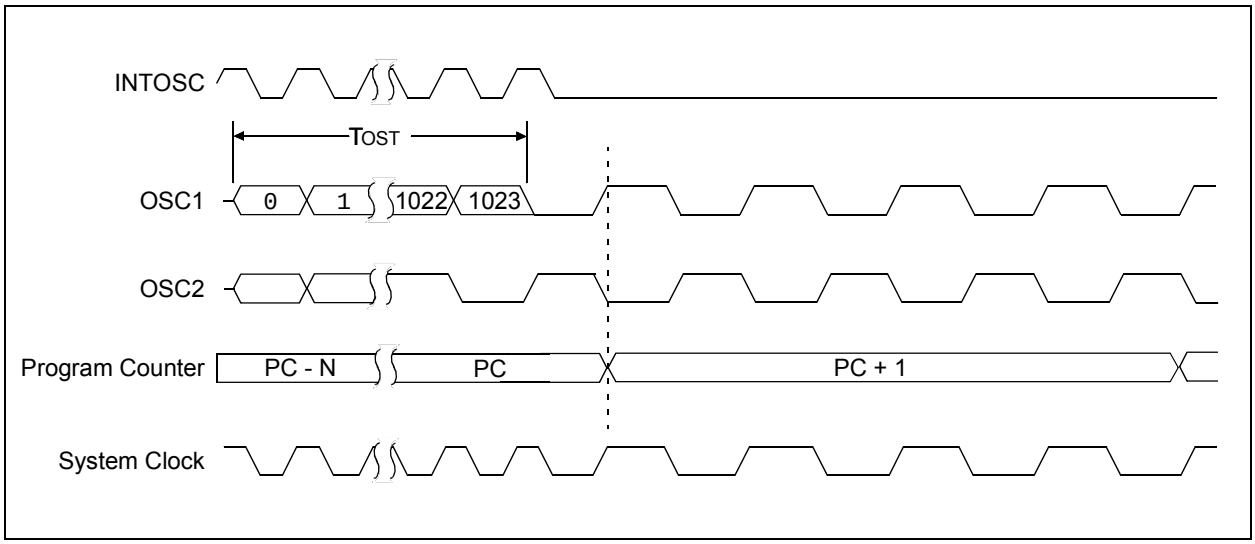
4: PIC16(L)F1619 only.

FIGURE 3-8: INDIRECT ADDRESSING



Note: Not all memory regions are completely implemented. Consult device memory tables for memory limits.

FIGURE 5-6: TWO-SPEED START-UP



REGISTER 7-3: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
OSFIE	C2IE	C1IE	—	BCL1IE	TMR6IE	TMR4IE	CCP2IE
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 **OSFIE:** Oscillator Fail Interrupt Enable bit
1 = Enables the Oscillator Fail interrupt
0 = Disables the Oscillator Fail interrupt

bit 6 **C2IE:** Comparator C2 Interrupt Enable bit
1 = Enables the Comparator C2 interrupt
0 = Disables the Comparator C2 interrupt

bit 5 **C1IE:** Comparator C1 Interrupt Enable bit
1 = Enables the Comparator C1 interrupt
0 = Disables the Comparator C1 interrupt

bit 4 **Unimplemented:** Read as '0'

bit 3 **BCL1IE:** MSSP Bus Collision Interrupt Enable bit
1 = Enables the MSSP Bus Collision Interrupt
0 = Disables the MSSP Bus Collision Interrupt

bit 2 **TMR6IE:** TMR6 to PR6 Match Interrupt Enable bit
1 = Enables the Timer6 to PR6 match interrupt
0 = Disables the Timer6 to PR6 match interrupt

bit 1 **TMR4IE:** TMR4 to PR4 Match Interrupt Enable bit
1 = Enables the Timer4 to PR4 match interrupt
0 = Disables the Timer4 to PR4 match interrupt

bit 0 **CCP2IE:** CCP2 Interrupt Enable bit
1 = The CCP2 interrupt is enabled
0 = The CCP2 interrupt is not enabled

Note 1: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 7-4: PIE3: PERIPHERAL INTERRUPT ENABLE REGISTER 3

U-0	U-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
—	—	CWGIE	ZCDIE	CLC4IE	CLC3IE	CLC2IE	CLC1IE
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 **CWGIE**: Complementary Waveform Generator (CWG) Interrupt Enable bit
- 1 = Enables the CWG interrupt
- 0 = Disables the CWG interrupt

bit 4 **ZCDIE**: Zero-Cross Detection (ZCD) Interrupt Enable bit
- 1 = Enables the ZCD interrupt
- 0 = Disables the ZCD interrupt

bit 3 **CLC4IE**: Configurable Logic Block 4 Interrupt Enable bit
- 1 = Enables the CLC 4 interrupt
- 0 = Disables the CLC 4 interrupt

bit 2 **CLC3IE**: Configurable Logic Block 3 Interrupt Enable bit
- 1 = Enables the CLC 3 interrupt
- 0 = Disables the CLC 3 interrupt

bit 1 **CLC2IE**: Configurable Logic Block 2 Interrupt Enable bit
- 1 = Enables the CLC 2 interrupt
- 0 = Disables the CLC 2 interrupt

bit 0 **CLC1IE**: Configurable Logic Block 1 Interrupt Enable bit
- 1 = Enables the CLC 1 interrupt
- 0 = Disables the CLC 1 interrupt

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

REGISTER 7-10: PIR4: PERIPHERAL INTERRUPT REQUEST REGISTER 4

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
SCANIF	CRCIF	SMT2PWAIF	SMT2PRAIF	SMT2IF	SMT1PWAIF	SMT1PRAIF	SMT1IF
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 **SCANIF**: Scanner Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 6 **CRCIF**: CRC Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 5 **SMT2PWAIF**: SMT2 Pulse Width Acquisition Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 4 **SMT2PRAIF**: SMT2 Period Acquisition Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 3 **SMT2IF**: SMT2 Match Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 2 **SMT1PWAIF**: SMT1 Pulse Width Acquisition Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 1 **SMT1PRAIF**: SMT1 Period Acquisition Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 0 **SMT1IF**: SMT1 Match Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

11.0 CYCLIC REDUNDANCY CHECK (CRC) MODULE

The Cyclic Redundancy Check (CRC) module provides a software-configurable hardware-implemented CRC checksum generator. This module includes the following features:

- Any standard CRC up to 16 bits can be used
- Configurable Polynomial
- Any seed value up to 16 bits can be used
- Standard and reversed bit order available
- Augmented zeros can be added automatically or by the user
- Memory scanner for fast CRC calculations on program memory user data
- Software loadable data registers for calculating CRC values not from the memory scanner

11.1 CRC Module Overview

The CRC module provides a means for calculating a check value of program memory. The CRC module is coupled with a memory scanner for faster CRC calculations. The memory scanner can automatically provide data to the CRC module. The CRC module can also be operated by directly writing data to SFRs, without using the scanner.

11.2 CRC Functional Overview

The CRC module can be used to detect bit errors in the Flash memory using the built-in memory scanner or through user input RAM. The CRC module can accept up to a 16-bit polynomial with up to a 16-bit seed value. A CRC calculated check value (or checksum) will then be generated into the CRCACC<15:0> registers for user storage. The CRC module uses an XOR shift register implementation to perform the polynomial division required for the CRC calculation.

EXAMPLE 11-1:

Rev. 10-000206A 1/8/2014

CRC-16-ANSI

$x^{16} + x^{15} + x^2 + 1$ (17 bits)

Standard 16-bit representation = 0x8005

```
CRCXORH = 0b10000000
CRCXORL = 0b0000010-
```
(1)

Data Sequence:
0x55, 0x66, 0x77, 0x88

```
DLEN = 0b0111
PLEN = 0b1111
```

Data entered into the CRC:
SHIFTM = 0:

```
01010101 01100110 01110111 10001000
```

SHIFTM = 1:

```
10101010 01100110 11101110 00010001
```

Check Value (ACCM = 1):

SHIFTM = 0: 0x32D6

```
CRCACCH = 0b00110010
CRCACCL = 0b11010110
```

SHIFTM = 1: 0x6BA2

```
CRCACCH = 0b01101011
CRCACCL = 0b10100010
```

Note 1: Bit 0 is unimplemented. The LSb of any CRC polynomial is always '1' and will always be treated as a '1' by the CRC for calculating the CRC check value. This bit will be read in software as a '0'.

11.3 CRC Polynomial Implementation

Any standard polynomial up to 17 bits can be used. The PLEN<3:0> bits are used to specify how long the polynomial used will be. For an x^n polynomial, PLEN = n-2. In an n-bit polynomial the x^n bit and the LSb will be used as a '1' in the CRC calculation because the MSb and LSb must always be a '1' for a CRC polynomial. For example, if using CRC-16-ANSI, the polynomial will look like 0x8005. This will be implemented into the CRCXOR<15:1> registers, as shown in Example 11-1.

EXAMPLE 11-2: CRC LFSR EXAMPLE

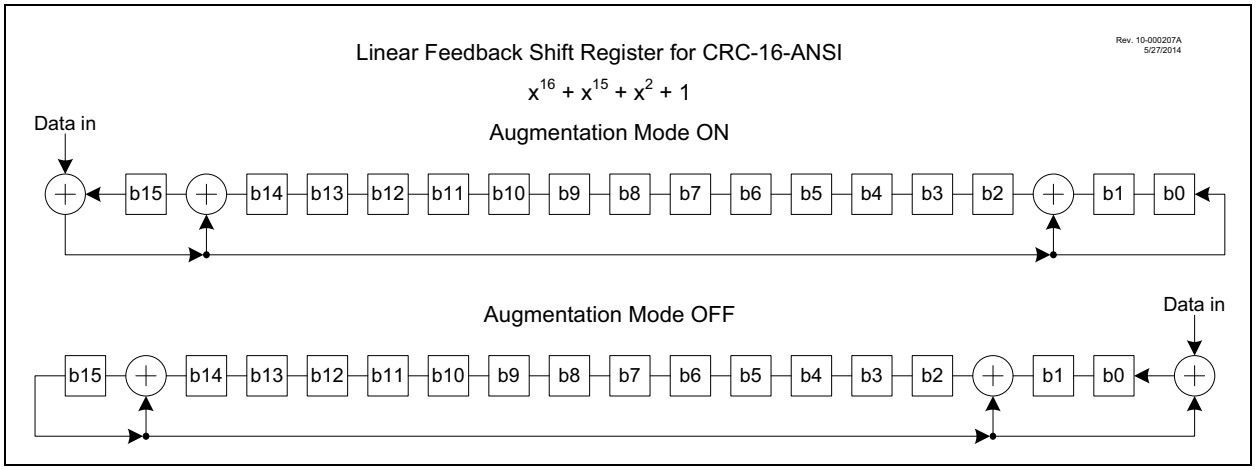


TABLE 13-2: AVAILABLE PORTS FOR OUTPUT BY PERIPHERAL(2)

RxyPPS<4:0>	Output Signal	PIC16(L)F1619			PIC16(L)F1615	
		PORTA	PORTB	PORTC	PORTA	PORTC
11xxx	Reserved	●	●	●	●	●
10111	Reserved	●	●	●	●	●
10110	Reserved	●	●	●	●	●
10101	Reserved	●	●	●	●	●
10100	Reserved	●	●	●	●	●
10011	DT	●	●	●	●	●
10010	TX/CK	●	●	●	●	●
10001	SDO/SDA(1)	●	●	●	●	●
10000	SCK/SCL(1)	●	●	●	●	●
01111	PWM4_out	●	●	●	●	●
01110	PWM3_out	●	●	●	●	●
01101	CCP2_out	●	●	●	●	●
01100	CCP1_out	●	●	●	●	●
01011	CWG1OUTD(1)	●	●	●	●	●
01010	CWG1OUTC(1)	●	●	●	●	●
01001	CWG1OUTB(1)	●	●	●	●	●
01000	CWG1OUTA(1)	●	●	●	●	●
00111	LC4_out	●	●	●	●	●
00110	LC3_out	●	●	●	●	●
00101	LC2_out	●	●	●	●	●
00100	LC1_out	●	●	●	●	●
00011	ZCD1_out	●	●	●	●	●
00010	sync_C2OUT	●	●	●	●	●
00001	sync_C1OUT	●	●	●	●	●
00000	LATxy	●	●	●	●	●

Note 1: TRIS control is overridden by the peripheral as required.

2: Unsupported peripherals will output a '0'.

15.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference (FVR) is a stable voltage reference, independent of V_{DD}, with a nominal output level (V_{FVR}) of 1.024V. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- Comparator positive input
- Comparator negative input

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

15.1 Independent Gain Amplifier

The output of the FVR supplied to the peripherals, (listed above), is routed through a programmable gain amplifier. 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 **Section17.0 "Analog-to-Digital Converter (ADC) Module"** for additional information.

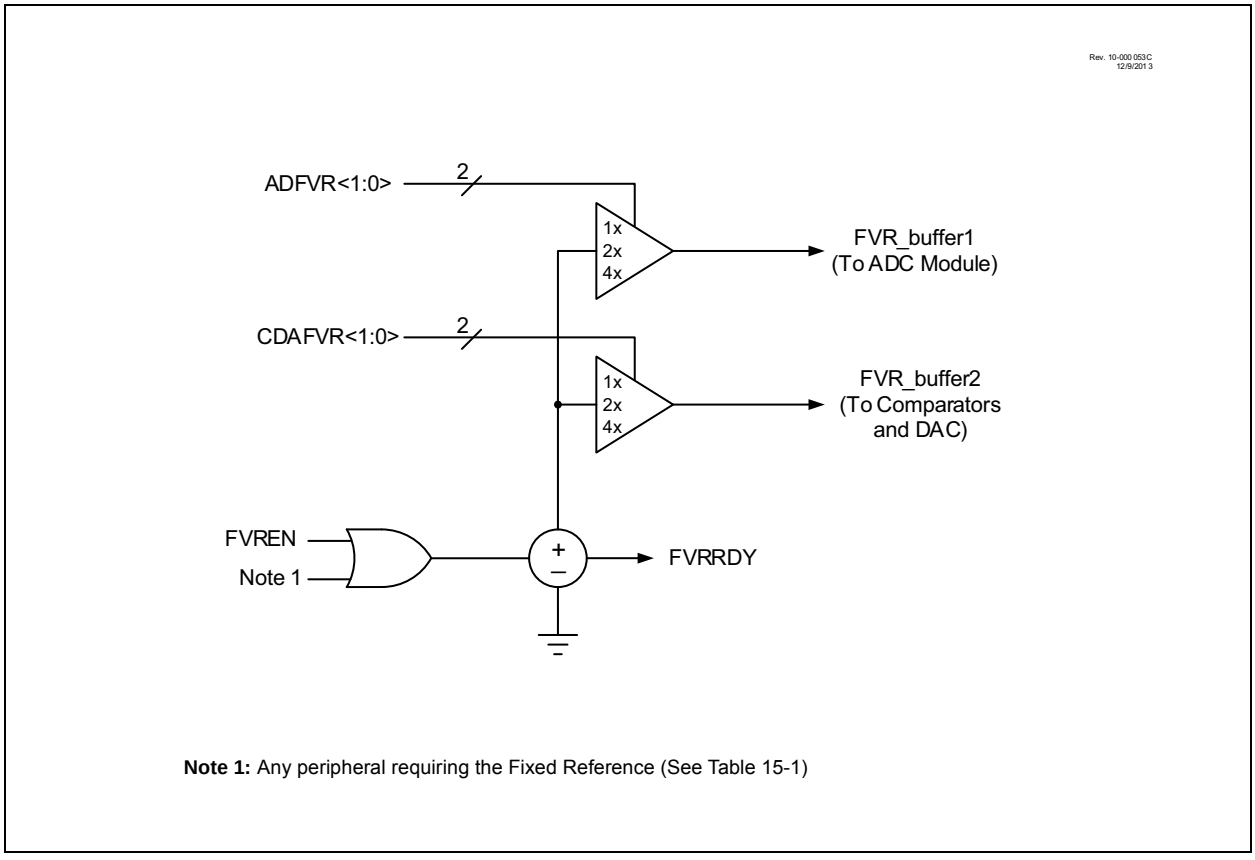
The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the comparator modules. Reference **Section19.0 "Comparator Module"** for additional information.

To minimize current consumption when the FVR is disabled, the FVR buffers should be turned off by clearing the Buffer Gain Selection bits.

15.2 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 36-64: FVR Stabilization Period, PIC16LF1614/8 Only.

FIGURE 15-1: VOLTAGE REFERENCE BLOCK DIAGRAM

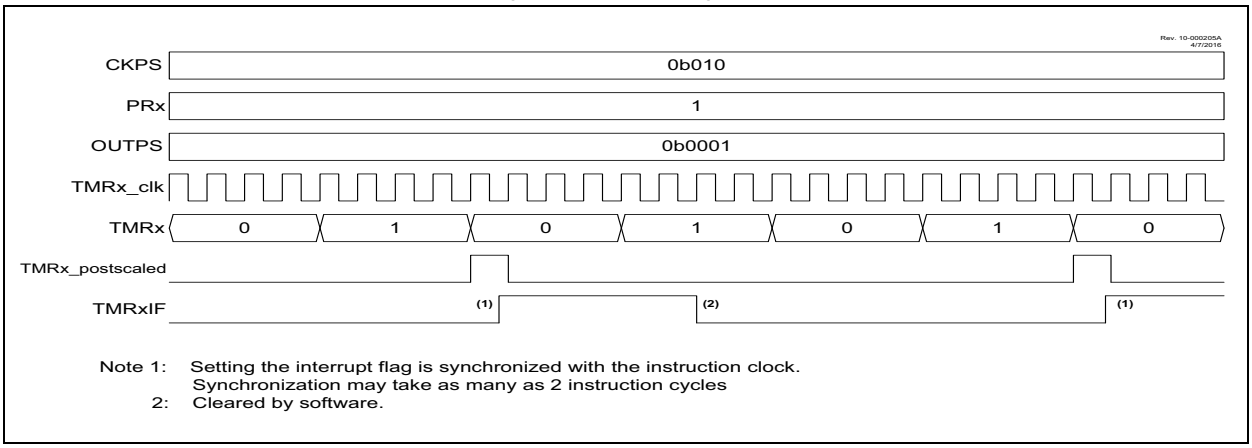


Note 1: Any peripheral requiring the Fixed Reference (See Table 15-1)

23.4 Timer2 Interrupt

Timer2 can also generate a device interrupt. The interrupt is generated when the postscaler counter matches one of 16 postscale options (from 1:1 through 1:16), which are selected with the postscaler control bits, OUTPS<3:0> of the T2CON register. The interrupt is enabled by setting the TMR2IE interrupt enable bit of the PIE1 register. Interrupt timing is illustrated in Figure 23-3.

FIGURE 23-3: TIMER2 PRESCALER, POSTSCALER, AND INTERRUPT TIMING DIAGRAM



24.2.5 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 (SSPxCON1<3:0> = 0100).

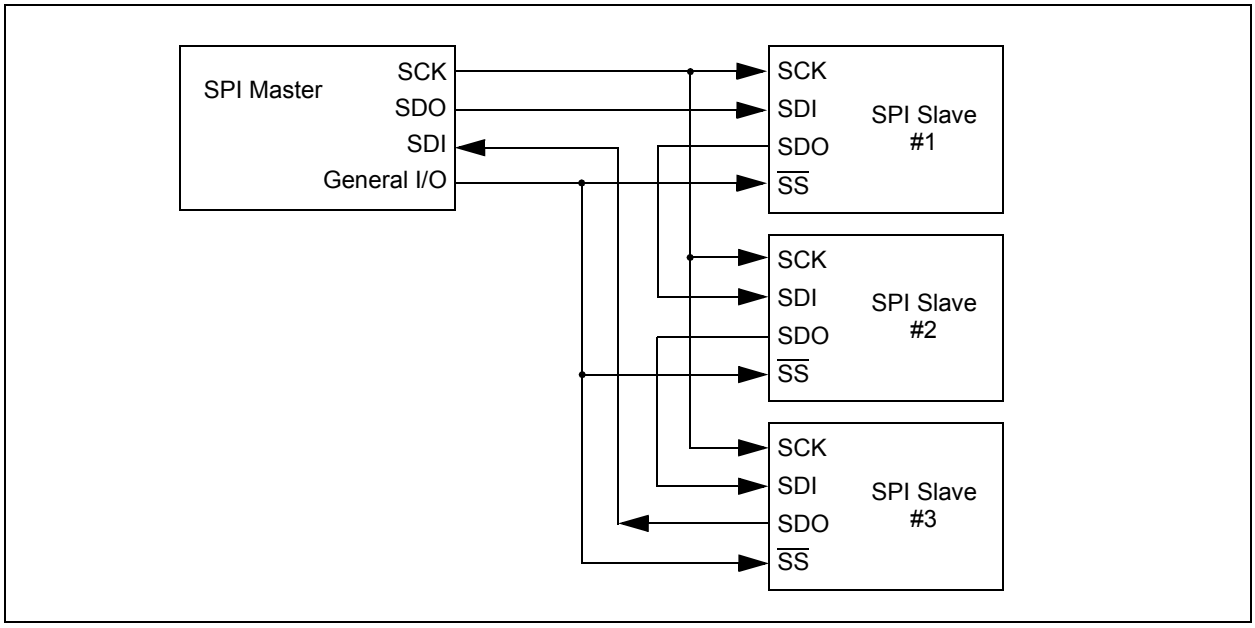
When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven.

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 (SSPxCON1<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 SSPxSTAT 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 24-7: SPI DAISY-CHAIN CONNECTION



24.7 BAUD RATE GENERATOR

The MSSP module has a Baud Rate Generator available for clock generation in both I^2C and SPI Master modes. The Baud Rate Generator (BRG) reload value is placed in the SSPxADD register (Register 24-6). When a write occurs to SSPxBUF, the Baud Rate Generator will automatically begin counting down.

Once the given operation is complete, the internal clock will automatically stop counting and the clock pin will remain in its last state.

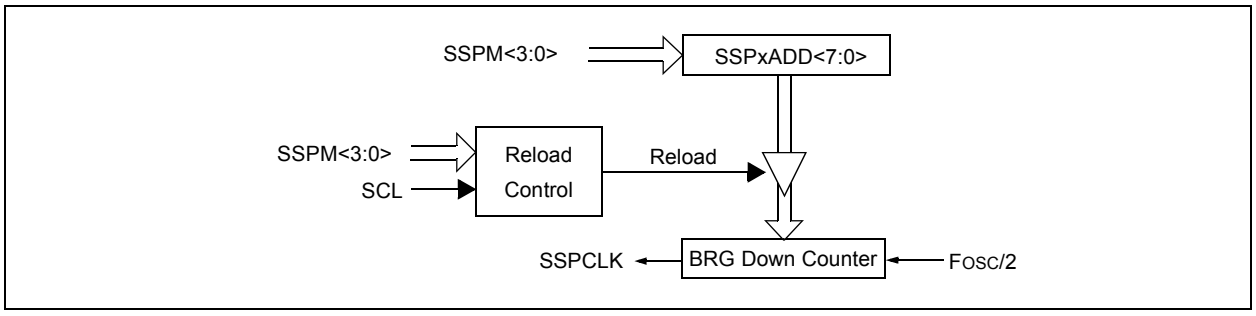
An internal signal “Reload” in Figure 24-40 triggers the value from SSPxADD to be loaded into the BRG counter. This occurs twice for each oscillation of the module clock line. The logic dictating when the reload signal is asserted depends on the mode the MSSP is being operated in.

Table 24-4 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPxADD.

EQUATION 24-1:

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPxADD + 1)(4)}$$

FIGURE 24-40: BAUD RATE GENERATOR BLOCK DIAGRAM



Note: Values of 0x00, 0x01 and 0x02 are not valid for SSPxADD when used as a Baud Rate Generator for I^2C. This is an implementation limitation.

TABLE 24-4: MSSP CLOCK RATE W/BRG

FOSC	FCY	BRG Value	FCLOCK (2 Rollovers of BRG)
32 MHz	8 MHz	13h	400 kHz
32 MHz	8 MHz	19h	308 kHz
32 MHz	8 MHz	4Fh	100 kHz
16 MHz	4 MHz	09h	400 kHz
16 MHz	4 MHz	0Ch	308 kHz
16 MHz	4 MHz	27h	100 kHz
4 MHz	1 MHz	09h	100 kHz

Note: Refer to the I/O port electrical specifications in Table 35-4 to ensure the system is designed to support I_{OL} requirements.

26.4 CCP/PWM Clock Selection

The PIC16(L)F1615/9 allows each individual CCP and PWM module to select the timer source that controls the module. Each module has an independent selection.

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

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

26.4.1 USING THE TMR2/4/6 WITH THE CCP MODULE

This device has a new version of the TMR2 module that has many new modes, which allow for greater customization and control of the PWM signals than older parts. Refer to **Section23.5 "Operation Examples"** for examples of PWM signal generation using the different modes of Timer2. The CCP operation requires that the timer used as the PWM time base has the FOSC/4 clock source selected.

26.4.2 PWM PERIOD

The PWM period is specified by the PR2/4/6 register of Timer2/4/6. The PWM period can be calculated using the formula of Equation 26-1.

EQUATION 26-1: PWM PERIOD

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

Note 1: TOSC = 1/FOSC

When TMR2/4/6 is equal to its respective PR2/4/6 register, the following three events occur on the next increment cycle:

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

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

26.4.3 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to two registers: the CCPRxH:CCPRxL register pair. Where the particular bits go is determined by the FMT bit of the CCPxCON register. If FMT = 0, the two Most Significant bits of the duty cycle value should be written to bits <1:0> of CCPRxH register and the remaining eight bits to the CCPRxL register. If FMT = 1, the Least Significant two bits of the duty cycle should be written to bits <7:6> of the CCPRxL register and the Most Significant eight bits to the CCPRxH register. This is illustrated in Figure 26-4. These bits can be written at any time. The duty cycle value is not latched into the internal latch until after the period completes (i.e., a match between PR2/4/6 and TMR2/4/6 registers occurs).

Equation 26-2 is used to calculate the PWM pulse width. Equation 26-3 is used to calculate the PWM duty cycle ratio.

EQUATION 26-2: PULSE WIDTH

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

EQUATION 26-3: DUTY CYCLE RATIO

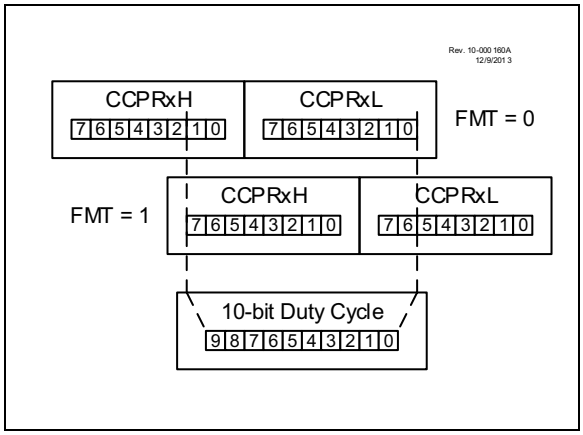
$$Duty\ Cycle\ Ratio = \frac{(CCPRxH{:}CCPRxL)}{4(PRx + 1)}$$

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

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

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

FIGURE 26-4: CCPx DUTY-CYCLE ALIGNMENT



26.4.4 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

27.1.9 SETUP FOR PWM OPERATION USING PWMx PINS

The following steps should be taken when configuring the module for PWM operation using the PWMx pins:

1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
2. Clear the PWMxCON register.
3. Load the PR2 register with the PWM period value.
4. Clear the PWMxDCH register and bits <7:6> of the PWMxDCL register.
5. Configure and start Timer2:
   - Clear the TMR2IF interrupt flag bit of the PIR1 register. See note below.
   - Configure the CKPS bits of the T2CON register with the Timer2 prescale value.
   - Enable Timer2 by setting the ON bit of the T2CON register.
6. Enable PWM output pin and wait until Timer2 overflows, TMR2IF bit of the PIR1 register is set. See note below.
7. Enable the PWMx pin output driver(s) by clearing the associated TRIS bit(s) and setting the PWMxOE bit of the PWMxCON register.
8. Configure the PWM module by loading the PWMxCON register with the appropriate values.

Note 1: In order to send a complete duty cycle and period on the first PWM output, the above steps must be followed in the order given. If it is not critical to start with a complete PWM signal, then move Step 8 to replace Step 4.

2: For operation with other peripherals only, disable PWMx pin outputs.

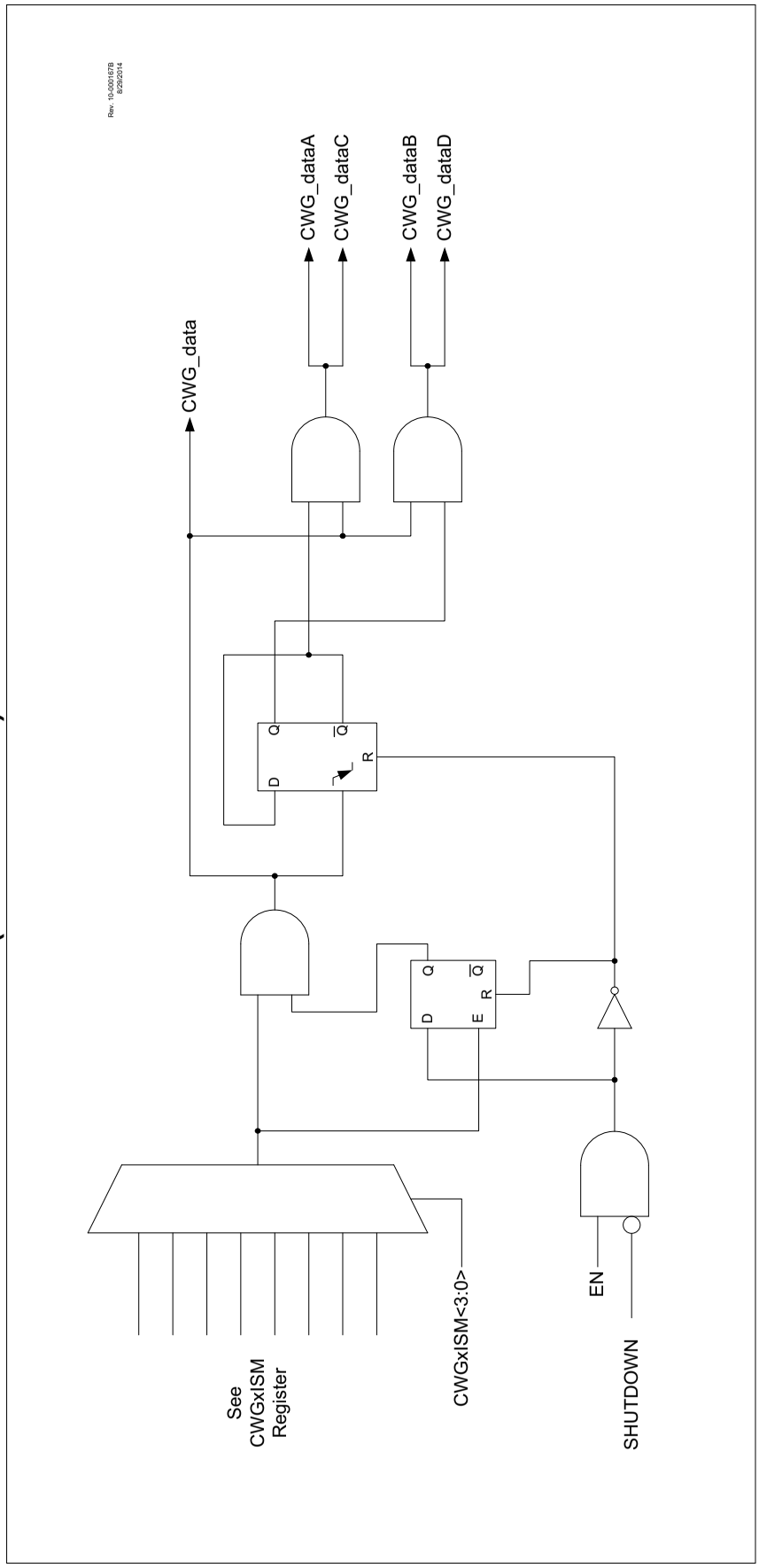


FIGURE 28-2: SIMPLIFIED CWG BLOCK DIAGRAM (PUSH-PULL MODE)

REGISTER 30-4: SMTxCLK: SMT CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CSEL<2:0>		
bit 7							bit 0

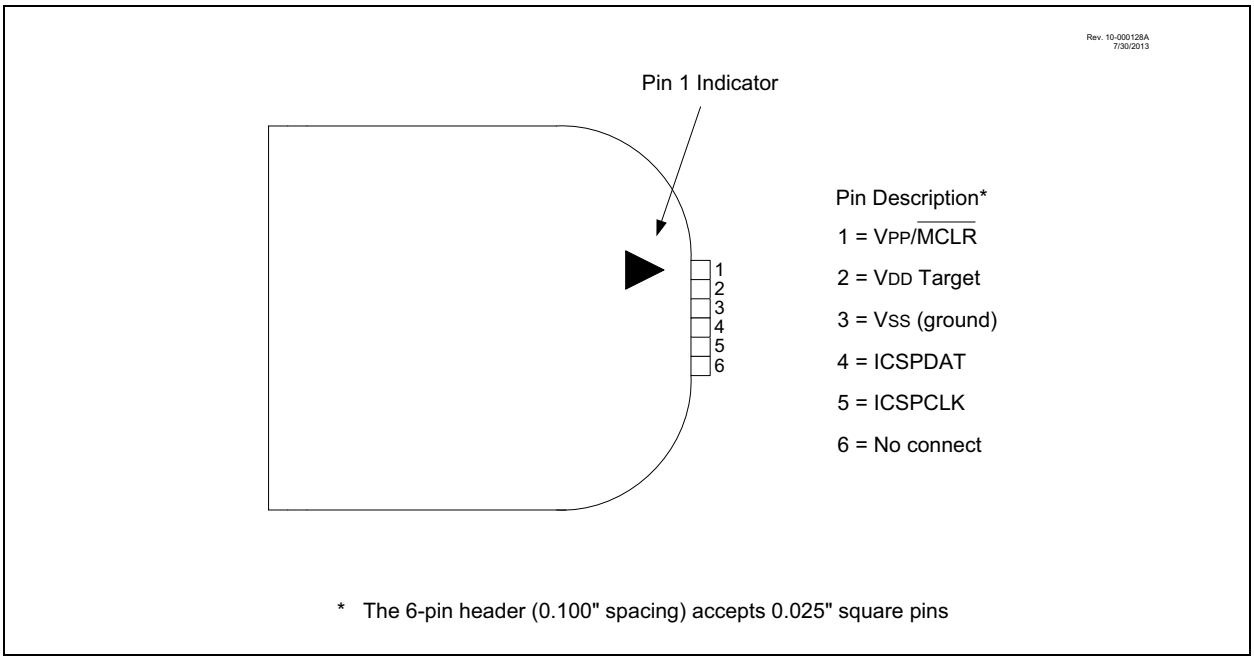
Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	q = Value depends on condition

bit 7-3 **Unimplemented**: Read as ‘0’

bit 2-0 **CSEL<2:0>:** SMT Clock Selection bits

111 = Reserved
110 = AT1_perclk
101 = MFINTOSC
100 = MFINTOSC/16
011 = LFINTOSC
010 = HFINTOSC 16 MHz
001 = FOSC/4
000 = FOSC

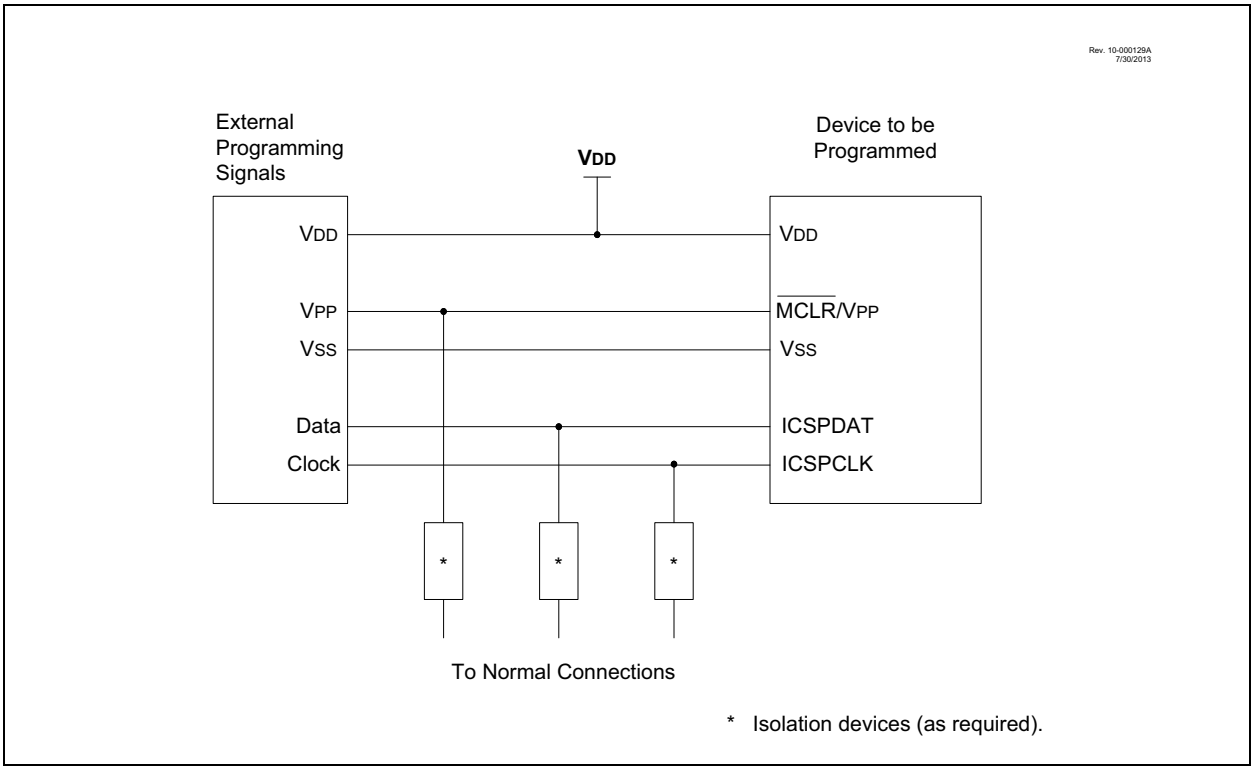
FIGURE 33-2: PICkit™ PROGRAMMER STYLE CONNECTOR INTERFACE



For additional interface recommendations, refer to your specific device programmer manual prior to PCB design.

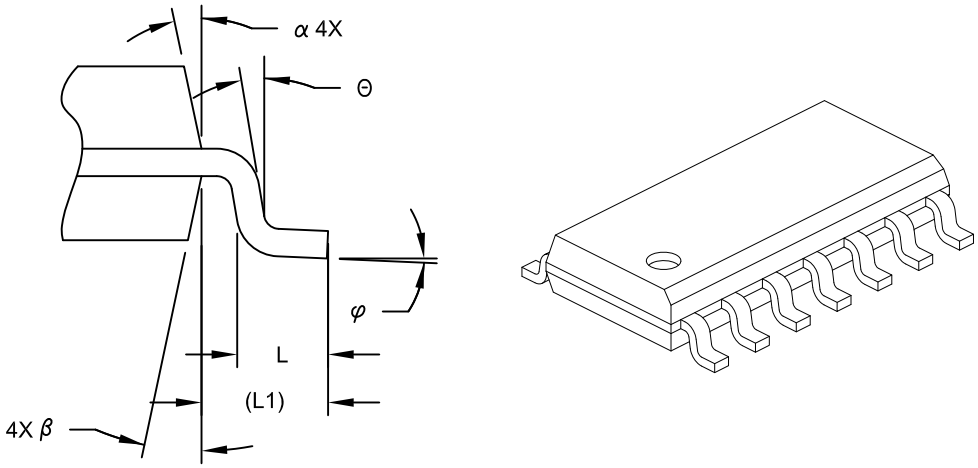
It is recommended that isolation devices be used to separate the programming pins from other circuitry. The type of isolation is highly dependent on the specific application and may include devices such as resistors, diodes, or even jumpers. See Figure 33-3 for more information.

FIGURE 33-3: TYPICAL CONNECTION FOR ICSP™ PROGRAMMING



14-Lead Plastic Small Outline (SL) - Narrow, 3.90 mm Body [SOIC]

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



VIEW C

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		14	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	1.75
Molded Package Thickness	A2	1.25	-	-
Standoff §	A1	0.10	-	0.25
Overall Width	E		6.00 BSC	
Molded Package Width	E1		3.90 BSC	
Overall Length	D		8.65 BSC	
Chamfer (Optional)	h	0.25	-	0.50
Foot Length	L	0.40	-	1.27
Footprint	L1		1.04 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.10	-	0.25
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

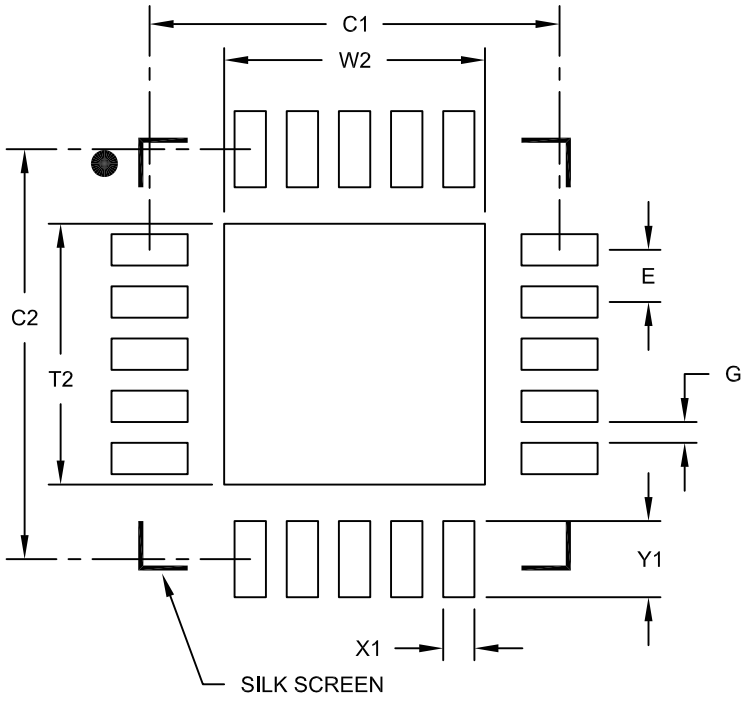
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-065C Sheet 2 of 2

20-Lead Plastic Quad Flat, No Lead Package (ML) - 4x4 mm Body [QFN]
With 0.40 mm Contact Length

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	W2			2.50
Optional Center Pad Length	T2			2.50
Contact Pad Spacing	C1		3.93	
Contact Pad Spacing	C2		3.93	
Contact Pad Width	X1			0.30
Contact Pad Length	Y1			0.73
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2126A