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	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 12x10b; D/A 2x5b, 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1769-e-p

TABLE 3-8: PIC16(L)F1765 MEMORY MAP (BANKS 8-23)

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	Core Registers (Table 3-2)	480h	Core Registers (Table 3-2)	500h	Core Registers (Table 3-2)	580h	Core Registers (Table 3-2)	600h	Core Registers (Table 3-2)	680h	Core Registers (Table 3-2)	700h	Core Registers (Table 3-2)	780h	Core Registers (Table 3-2)
40Bh		48Bh		50Bh		58Bh		60Bh		68Bh		70Bh		78Bh	
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	COG1PHR	70Dh	COG2PHR	78Dh	—
40Eh	HIDRVC	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	COG1PHF	70Eh	COG2PHF	78Eh	—
40Fh	—	48Fh	—	50Fh	OPA1NCHS	58Fh	—	60Fh	—	68Fh	COG1BLKR	70Fh	COG2BLKR	78Fh	—
410h	—	490h	—	510h	OPA1PCHS	590h	DACLD	610h	—	690h	COG1BLKF	710h	COG2BLKF	790h	—
411h	—	491h	—	511h	OPA1CON	591h	DAC1CON0	611h	—	691h	COG1DBR	711h	COG2DBR	791h	—
412h	—	492h	—	512h	OPA1ORS	592h	DAC1REFL	612h	—	692h	COG1DBF	712h	COG2DBF	792h	—
413h	T4TMR	493h	TMR3L	513h	—	593h	DAC1REFH	613h	—	693h	COG1CON0	713h	COG2CON0	793h	—
414h	T4PR	494h	TMR3H	514h	—	594h	—	614h	—	694h	COG1CON1	714h	COG2CON1	794h	PRG1RTSS
415h	T4CON	495h	T3CON	515h	—	595h	—	615h	—	695h	COG1RIS0	715h	COG2RIS0	795h	PRG1FTSS
416h	T4HLT	496h	T3GCON	516h	—	596h	—	616h	—	696h	COG1RIS1	716h	COG2RIS1	796h	PRG1INS
417h	T4CLKCON	497h	—	517h	—	597h	DAC3CON0	617h	PWM3DCL	697h	COG1RSIM0	717h	COG2RSIM0	797h	PRG1CON0
418h	T4RST	498h	—	518h	—	598h	DAC3REF	618h	PWM3DCH	698h	COG1RSIM1	718h	COG2RSIM1	798h	PRG1CON1
419h	—	499h	—	519h	—	599h	—	619h	PWM3CON	699h	COG1FIS0	719h	COG2FIS0	799h	PRG1CON2
41Ah	T6TMR	49Ah	TMR5L	51Ah	—	59Ah	—	61Ah	—	69Ah	COG1FIS1	71Ah	COG2FIS1	79Ah	—
41Bh	T6PR	49Bh	TMR5H	51Bh	—	59Bh	—	61Bh	—	69Bh	COG1FSIM0	71Bh	COG2FSIM0	79Bh	—
41Ch	T6CON	49Ch	T5CON	51Ch	—	59Ch	—	61Ch	—	69Ch	COG1FSIM1	71Ch	COG2FSIM1	79Ch	—
41Dh	T6HLT	49Dh	T5GCON	51Dh	—	59Dh	—	61Dh	—	69Dh	COG1ASD0	71Dh	COG2ASD0	79Dh	—
41Eh	T6CLKCON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	COG1ASD1	71Eh	COG2ASD1	79Eh	—
41Fh	T6RST	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	COG1STR	71Fh	COG2STR	79Fh	—
420h	General Purpose Register 80 Bytes	4A0h	General Purpose Register 80 Bytes	520h	General Purpose Register 80 Bytes	5A0h	General Purpose Register 80 Bytes	620h	General Purpose Register 48 Bytes	6A0h	Unimplemented Read as '0'	720h	Unimplemented Read as '0'	7A0h	Unimplemented Read as '0'
46Fh		4EFh		56Fh		5EFh		64Fh		6EFh		76Fh		7EFh	
470h	Accesses 70h-7Fh	4F0h	Accesses 70h-7Fh	570h	Accesses 70h-7Fh	5F0h	Accesses 70h-7Fh	650h	Unimplemented Read as '0'	6F0h	Accesses 70h-7Fh	770h	Accesses 70h-7Fh	7F0h	Accesses 70h-7Fh
47Fh		4FFh		57Fh		5FFh		66Fh		6FFh		77Fh		7FFh	
800h	Core Registers (Table 3-2)	880h	Core Registers (Table 3-2)	900h	Core Registers (Table 3-2)	980h	Core Registers (Table 3-2)	A00h	Core Registers (Table 3-2)	A80h	Core Registers (Table 3-2)	B00h	Core Registers (Table 3-2)	B80h	Core Registers (Table 3-2)
80Bh		88Bh		90Bh		98Bh		A0Bh		A8Bh		B0Bh		B8Bh	
80Ch	Unimplemented Read as '0'	88Ch	Unimplemented Read as '0'	90Ch	Unimplemented Read as '0'	98Ch	Unimplemented Read as '0'	A0Ch	Unimplemented Read as '0'	A8Ch	Unimplemented Read as '0'	B0Ch	Unimplemented Read as '0'	B8Ch	Unimplemented Read as '0'
86Fh		8EFh		96Fh		9EFh		A6Fh		AEFh		B6Fh		BEFh	
870h	Accesses 70h-7Fh	8F0h	Accesses 70h-7Fh	970h	Accesses 70h-7Fh	9F0h	Accesses 70h-7Fh	A70h	Accesses 70h-7Fh	AF0h	Accesses 70h-7Fh	B70h	Accesses 70h-7Fh	BF0h	Accesses 70h-7Fh
87Fh		8FFh		97Fh		9FFh		A7Fh		AFFh		B7Fh		BFFh	

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

TABLE 3-16: SPECIAL FUNCTION REGISTER SUMMARY

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on All Other Resets
Bank 0											
00Ch	PORTA	—	—	RA<5:0>						--xx xxxx	--uu uuuu
00Dh	PORTB(2)	RB<7:4>				—	—	—	—	xxxx ----	uuuu ----
00Eh	PORTC	RC<7:6>(2)		RC<5:0>						xxxx xxxx	uuuu uuuu
00Fh	—	Unimplemented								—	—
010h	—	Unimplemented								—	—
011h	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
012h	PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF(2)	C3IF(2)	CCP2IF(2)	000- 0000	000- 0000
013h	PIR3	PWM6IF(2)	PWM5IF	COG1IF	ZCDIF	COG2IF(2)	CLC3IF	CLC2IF	CLC1IF	0000 0000	0000 0000
014h	PIR4	—	—	TMR5GIF	TMR5IF	TMR3GIF	TMR3IF	TMR6IF	TRM4IF	--00 0000	--00 0000
015h	TMR0	Timer0 Module Register								0000 0000	0000 0000
016h	TMR1L	Holding Register for the Least Significant Byte of the 16-Bit TMR1 Register								xxxx xxxx	uuuu uuuu
017h	TMR1H	Holding Register for the Most Significant Byte of the 16-Bit TMR1 Register								xxxx xxxx	uuuu uuuu
018h	T1CON	CS<1:0>		CKPS<1:0>		OSCEN	$\overline{SYNC}$	—	ON	0000 00-0	uuuu uu-u
019h	T1GCON	GE	GPOL	GTM	GSPM	GGO/$\overline{DONE}$	GVAL	GSS<1:0>		0000 0x00	uuuu uxuu
01Ah	T2TMR	Holding Register for the 8-Bit TMR2 Register								0000 0000	0000 0000
01Bh	T2PR	TMR2 Period Register								1111 1111	1111 1111
01Ch	T2CON	ON	CKPS<2:0>			OUTPS<3:0>				0000 0000	0000 0000
01Dh	T2HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					0000 0000	0000 0000
01Eh	T2CLKCON	—	—	—	—	CS<3:0>				---- 0000	---- 0000
01Fh	T2RST	—	—	—	—	RSEL<3:0>				---- 0000	---- 0000
Bank 1											
08Ch	TRISA	—	—	TRISA<5:4>		—(1)	TRISA<2:0>			--11 1111	--11 1111
08Dh	TRISB(2)	TRISB<7:4>				—	—	—	—	1111 ----	1111 ----
08Eh	TRISC	TRISC<7:6>(2)		TRISC<5:0>						1111 1111	1111 1111
08Fh	—	Unimplemented								—	—
090h	—	Unimplemented								—	—
091h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
092h	PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE(2)	C3IE(2)	CCP2IE(2)	000- 0000	000- 0000
093h	PIE3	PWM6IE(2)	PWM5IE	COG1IE	ZCDIE	COG2IE(2)	CLC3IE	CLC2IE	CLC1IE	0000 0000	0000 0000
094h	PIE4	—	—	TMR5GIE	TMR5IE	TMR3GIE	TMR3IE	TMR6IE	TRM4IE	--00 0000	--00 0000
095h	OPTION_REG	$\overline{WPUEN}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			1111 1111	1111 1111
096h	PCON	STKOVF	STKUNF	—	$\overline{RWDT}$	$\overline{RMCLR}$	$\overline{RI}$	$\overline{POR}$	$\overline{BOR}$	00-1 11qq	qq-q qquu
097h	WDTCON	—	—	WDTPS<4:0>					SWDTEN	--01 0110	--01 0110
098h	OSCTUNE	—	—	TUN<5:0>						--00 0000	--00 0000
099h	OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		0011 1-00	0011 1-00
09Ah	OSCSTAT	SOSCR	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	00q0 0q0q	qqqq qq0q
09Bh	ADRESL	ADC Result Register Low								xxxx xxxx	uuuu uuuu
09Ch	ADRESH	ADC Result Register High								xxxx xxxx	uuuu uuuu
09Dh	ADCON0	—	CHS<4:0>					GO/$\overline{DONE}$	ADON	-000 0000	-000 0000
09Eh	ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		0000 -000	0000 -000
09Fh	ADCON2	TRIGSEL<4:0>					—	—	—	0000 0---	0000 0---

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: PIC16(L)F1768/9 only.

3: PIC16(L)F1764/5 only.

4: Unimplemented on PIC16LF1764/5/8/9.

3.5 Stack

All devices have a 16-level x 15-bit wide hardware stack (refer to Figure 3-1). The stack space is not part of either program or data space. The PC is PUSHed onto the stack when CALL or CALLW instructions are executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer if the STVREN bit is programmed to '0' (Configuration Words). This means that after the stack has been PUSHed sixteen times, the seventeenth PUSH overwrites the value that was stored from the first PUSH. The eighteenth PUSH overwrites the second PUSH (and so on). The STKOVF and STKUNF flag bits will be set on an Overflow/Underflow Reset, regardless of whether the Reset is enabled.

Note:	There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, CALLW, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

3.5.1 ACCESSING THE STACK

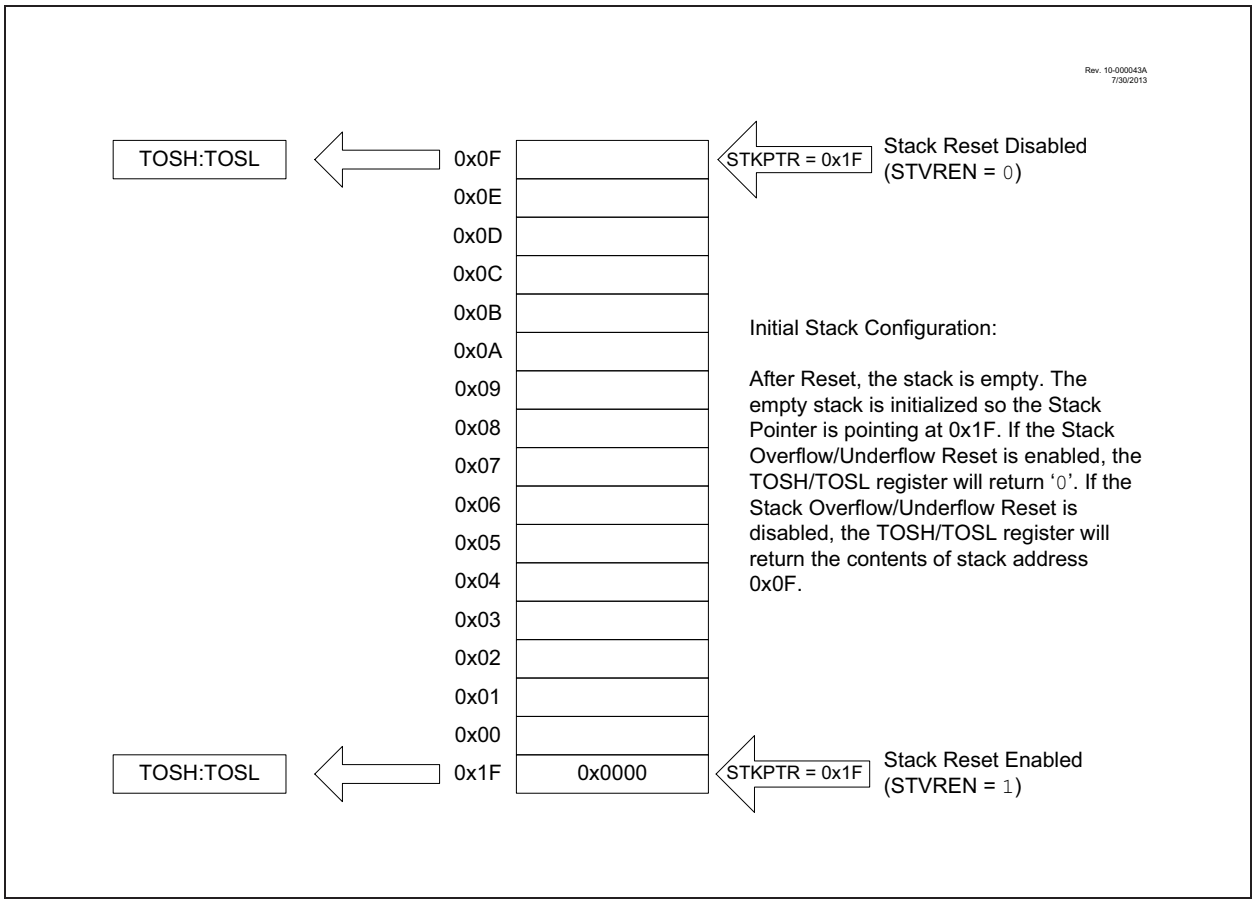
The stack is available through the TOSH, TOSL and STKPTR registers. STKPTR is the current value of the Stack Pointer. TOSH:TOSL register pair points to the TOP of the stack. Both registers are read/writable. TOS is split into TOSH and TOSL due to the 15-bit size of the PC. To access the stack, adjust the value of STKPTR, which will position TOSH:TOSL, then read/write to TOSH:TOSL. STKPTR is five bits to allow detection of overflow and underflow.

Note:	Care should be taken when modifying the STKPTR while interrupts are enabled.

During normal program operation, CALL, CALLW and Interrupts will increment STKPTR while RETLW, RETURN, and RETFIE will decrement STKPTR. At any time, STKPTR can be inspected to see how much stack is left. The STKPTR always points at the currently used place on the stack. Therefore, a CALL or CALLW will increment the STKPTR and then write the PC, and a return will unload the PC and then decrement the STKPTR.

Reference Figure 3-4 through Figure 3-7 for examples of accessing the stack.

FIGURE 3-4: ACCESSING THE STACK EXAMPLE 1



5.0 OSCILLATOR MODULE (WITH FAIL-SAFE CLOCK MONITOR)

5.1 Overview

The oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 5-1 illustrates a block diagram of the oscillator module.

Clock sources can be supplied from external oscillators, quartz crystal resonators, ceramic resonators and Resistor-Capacitor (RC) circuits. In addition, the system clock source can be supplied from one of two internal oscillators and PLL circuits, with a choice of speeds selectable via software. Additional clock features include:

- Selectable system clock source between external or internal sources via software.
- Two-Speed Start-up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, ECH, ECM, ECL or EXTRC modes) and switch automatically to the internal oscillator.
- Oscillator Start-up Timer (OST) ensures stability of crystal oscillator sources.

The oscillator module can be configured in one of the following clock modes.

1. ECL – External Clock Low-Power mode (0 MHz to 0.5 MHz).
2. ECM – External Clock Medium Power mode (0.5 MHz to 4 MHz).
3. ECH – External Clock High-Power mode (4 MHz to 32 MHz).
4. LP – 32 kHz Low-Power Crystal mode.
5. XT – Medium Gain Crystal or Ceramic Resonator Oscillator mode (up to 4 MHz).
6. HS – High Gain Crystal or Ceramic Resonator mode (4 MHz to 20 MHz).
7. EXTRC – External Resistor-Capacitor.
8. INTOSC – Internal Oscillator (31 kHz to 32 MHz).

Clock source modes are selected by the FOSC<2:0> bits in the Configuration Words. The FOSCx bits determine the type of oscillator that will be used when the device is first powered.

The ECH, ECM and ECL Clock modes rely on an external logic level signal as the device clock source. The LP, XT and HS Clock modes require an external crystal or resonator to be connected to the device. Each mode is optimized for a different frequency range. The EXTRC Clock mode requires an external resistor and capacitor to set the oscillator frequency.

The Internal Oscillator Block (INTOSC) produces low, medium and high-frequency clock sources, designated LFINTOSC, MFINTOSC and HFINTOSC (see Internal Oscillator Block, Figure 5-1). A wide selection of device clock frequencies may be derived from these three clock sources.

6.2.1 BOR IS ALWAYS ON

When the BOREN<1:0> bits of the Configuration Words are programmed to '11', the BOR is always on. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is active during Sleep. The BOR does not delay wake-up from Sleep.

6.2.2 BOR IS OFF IN SLEEP

When the BORENx bits of the Configuration Words are programmed to '10', the BOR is on, except in Sleep. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is not active during Sleep. The device wake-up will be delayed until the BOR is ready.

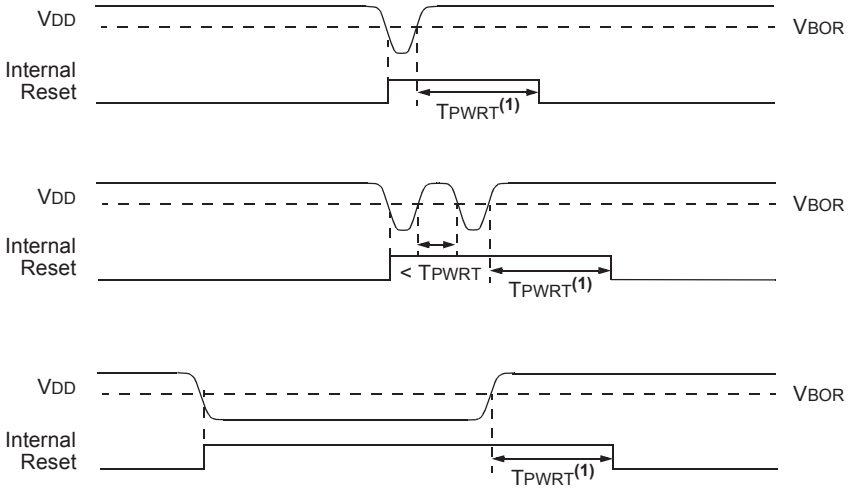
6.2.3 BOR CONTROLLED BY SOFTWARE

When the BORENx bits of the Configuration Words are programmed to '01', the BOR is controlled by the SBOREN bit of the BORCON register. The device start-up is not delayed by the BOR ready condition or the VDD level.

BOR protection begins as soon as the BOR circuit is ready. The status of the BOR circuit is reflected in the BORRDY bit of the BORCON register.

BOR protection is unchanged by Sleep.

FIGURE 6-2: BROWN-OUT SITUATIONS



Note 1: TPWRT delay only if the $\overline{\text{PWRTE}}$ bit is programmed to '0'.

REGISTER 7-6: PIR2: PERIPHERAL INTERRUPT REQUEST 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
OSFIF	C2IF	C1IF	—	BCL1IF	C4IF(1)	C3IF(1)	CCP2IF(1)
bit 7							bit 0

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

bit 7 **OSFIF:** Oscillator Fail Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 6 **C2IF:** Comparator C2 Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 5 **C1IF:** Comparator C1 Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

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

bit 3 **BCL1IF:** MSSP Bus Collision Interrupt Flag bit
1 = Interrupt is pending
0 = Interrupt is not pending

bit 2 **C4IF:** Comparator C4 Interrupt Flag bit(1)
1 = Interrupt is pending
0 = Interrupt is not pending

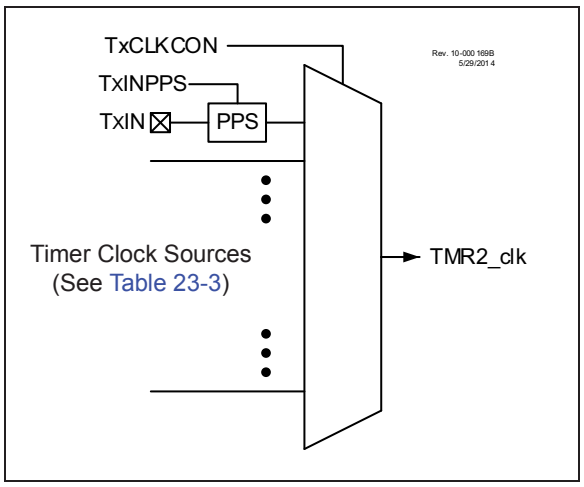
bit 1 **C3IF:** Comparator C3 Interrupt Flag bit(1)
1 = Interrupt is pending
0 = Interrupt is not pending

bit 0 **CCP2IF:** CCP2 Interrupt Flag bit(1)
1 = Interrupt is pending
0 = Interrupt is not pending

Note 1: PIC16(L)F1768/9 only.

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.

FIGURE 23-2: TIMER2 CLOCK SOURCE BLOCK DIAGRAM



23.1 Timer2 Operation

Timer2 operates in three major modes:

- Free-Running Period
- One-Shot
- Monostable

Within each mode, there are several options for starting, stopping and resetting. Table 23-1 lists the options.

In all modes, the TMR2 Count register is incremented on the rising edge of the clock signal from the programmable prescaler. When TMR2 equals T2PR, a high level is output to the postscaler counter. TMR2 is cleared on the next clock input.

An external signal from hardware can also be configured to gate the timer operation or force a TMR2 count Reset. In Gate modes the counter stops when the gate is disabled and resumes when the gate is enabled. In Reset modes the TMR2 count is reset on either the level or edge from the external source.

The T2PR period register is double buffered. Software reads and writes the T2PR register. However, the timer uses a buffered PRx register for operation. Software does not have direct access to the buffered PRx register. The content of the PRx register is transferred to the buffer by any of the following events:

- A write to the TMR2 register
- A write to the T2CON register
- When TMR2 = PRx buffer and the prescaler rolls over
- An external Reset event

The TMR2 register is directly readable and writable. The TMR2 register is cleared and the T2PR register initializes to FFh on any device Reset. Both the prescaler and postscaler counters are cleared on the following events:

- A write to the TMR2 register
- A write to the T2CON register
- Any device Reset
- External Reset source event that resets the timer.

Note: TMR2 is not cleared when T2CON is written.

23.1.1 FREE-RUNNING PERIOD MODE

The value of TMR2 is compared to that of the Period register, T2PR, on each clock cycle. When the two values match, the comparator resets the value of TMR2 to 00h on the next cycle and increments the output postscaler counter. When the postscaler count equals the value in the OUTPS<4:0> bits of the TMRxCON1 register then a one clock period wide pulse occurs on the TMR2_postscaled output, and the postscaler count is cleared.

23.1.2 ONE-SHOT MODE

The One-Shot mode is identical to the Free-Running Period mode except that the ON bit is cleared and the timer is stopped when TMR2 matches T2PR, and will not restart until the T2ON bit is cycled off and on. Postscaler OUTPS<4:0> values other than 0 are meaningless in this mode because the timer is stopped at the first period event and the postscaler is reset when the timer is restarted.

23.1.3 MONOSTABLE MODE

Monostable modes are similar to One-Shot modes except that the ON bit is not cleared and the timer can be restarted by an external Reset event.

23.2 Timer2 Output

The Timer2 module's primary output is TMR2_postscaled, which pulses for a single TMR2_clk period when the postscaler counter matches the value in the OUTPS bits of the TMR2xCON register. The T2PR postscaler is incremented each time the TMR2 value matches the T2PR value. This signal can be selected as an input to several other input modules:

- The ADC module as an auto-conversion trigger
- COG as an auto-shutdown source

In addition, the Timer2 is also used by the CCP module for pulse generation in PWM mode. Both the actual TMR2 value as well as other internal signals are sent to the CCP module to properly clock both the period and pulse width of the PWM signal. See **Section 24.6 "CCP/PWM Clock Selection"** for more details on setting up Timer2 for use with the CCP, as well as the timing diagrams in **Section 23.6 "Operation Examples"** for examples of how the varying Timer2 modes affect CCP PWM output.

TABLE 24-4: SUMMARY OF REGISTERS ASSOCIATED WITH STANDARD PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCPxCAP	—	—	—	—	—	CTS<2:0>			259
CCPxCON	EN	OE	OUT	FMT	MODE<3:0>				256
CCPRxL	Capture/Compare/PWM Register x (LSB)								258
CCPRxH	Capture/Compare/PWM Register x (MSB)								258
CCPTMRS	P4TSEL<1:0>		P3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>		257
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	101
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	102
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE(1)	C3IE(1)	CCP2IE(1)	103
T2PR	Timer2 Period Register								227*
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				244
TMR2	Timer2 Module Register								227
T4PR	Timer4 Period Register								227*
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				244
TMR4	Timer4 Module Register								227
T6PR	Timer6 Period Register								227*
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				244
TMR6	Timer6 Module Register								227

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the PWM.

* Page provides register information.

Note 1: PIC16(L)F1768/9 only.

FIGURE 27-9: TYPICAL HALF-BRIDGE MODE COG OPERATION WITH CCP1

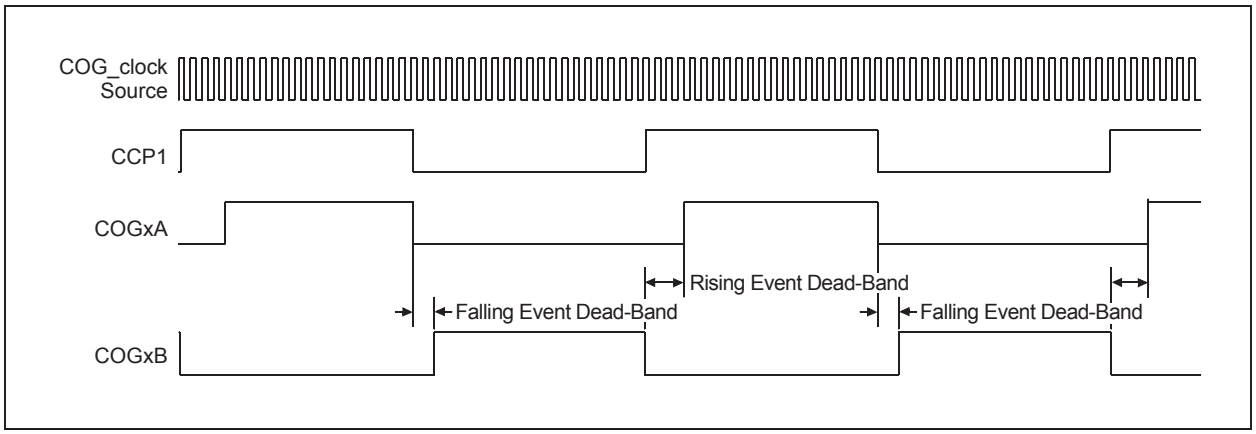


FIGURE 27-10: HALF-BRIDGE MODE COG OPERATION WITH CCP1 AND PHASE DELAY

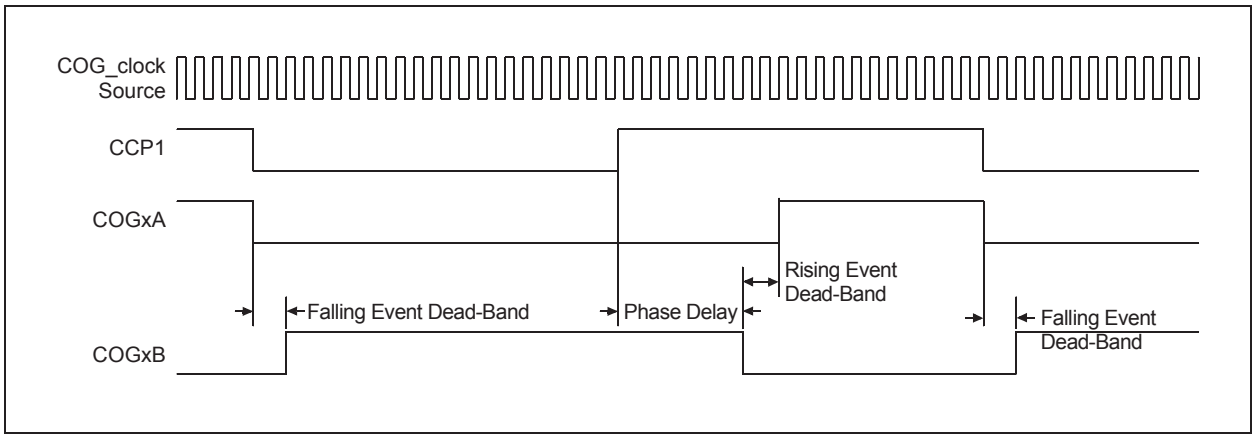
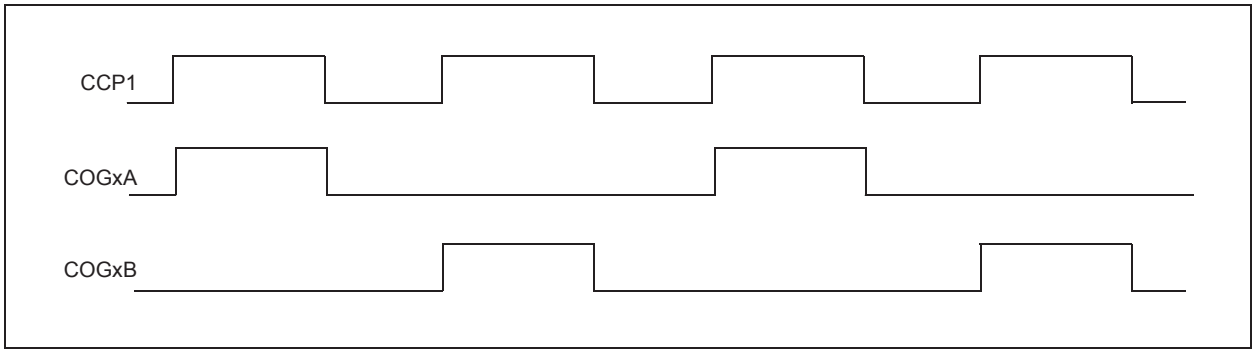


FIGURE 27-11: PUSH-PULL MODE COG OPERATION WITH CCP1



REGISTER 27-4: COGxRIS1: COGx RISING EVENT INPUT SELECTION 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
RIS15(1)	RIS14	RIS13	RIS12	RIS11	RIS10	RIS9	RIS8
bit 7							bit 0

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

bit 7 **RIS15:** COGx Rising Event Input Source 15 Enable bit(1)
- 1 = DSM2 MD2_out is enabled as a rising event input
- 0 = DSM2 MD2_out has no effect on the rising event

bit 6 **RIS14:** COGx Rising Event Input Source 14 Enable bit
- 1 = DSM1 MD1_out output is enabled as a rising event input
- 0 = DSM1 MD1_out has no effect on the rising event

bit 5 **RIS13:** COGx Rising Event Input Source 13 Enable bit
- 1 = CLC3 output is enabled as a rising event input
- 0 = CLC3 output has no effect on the rising event

bit 4 **RIS12:** COGx Rising Event Input Source 12 Enable bit
- 1 = CLC2 output is enabled as a rising event input
- 0 = CLC2 output has no effect on the rising event

bit 3 **RIS11:** COGx Rising Event Input Source 11 Enable bit
- 1 = CLC1 output is enabled as a rising event input
- 0 = CLC1 output has no effect on the rising event

bit 2 **RIS10:** COGx Rising Event Input Source 10 Enable bit
- 1 = PWM6 output is enabled as a rising event input
- 0 = PWM6 output has no effect on the rising event

bit 1 **RIS9:** COGx Rising Event Input Source 9 Enable bit
- 1 = PWM5 output is enabled as a rising event input
- 0 = PWM5 output has no effect on the rising event

bit 0 **RIS8:** COGx Rising Event Input Source 8 Enable bit
- 1 = PWM4 output is enabled as rising event input
- 0 = PWM4 output has no effect on the rising event

Note 1: PIC16(L)F1768/9 only. Otherwise unimplemented, read as '0'.

28.1 CLCx Setup

Programming the CLCx module is performed by configuring the four stages in the logic signal flow. The four stages are:

- Data selection
- Data gating
- Logic function selection
- Output polarity

Each stage is set up at run time by writing to the corresponding CLCx Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

28.1.1 DATA SELECTION

There are 32 signals available as inputs to the configurable logic. Four 32-input multiplexers are used to select the inputs to pass on to the next stage.

Data selection is through four multiplexers, as indicated on the left side of Figure 28-2. Data inputs in the figure are identified by a generic numbered input name.

Table 28-1 correlates the generic input name to the actual signal for each CLC module. The column labeled, dy, indicates the MUX selection code for the selected data input. DxS is an abbreviation for the MUX select input codes: D1S<5:0> through D4S<5:0>.

Data inputs are selected with the CLCxSEL0 through CLCxSEL3 registers (Register 28-3 through Register 28-6).

Note:	Data selections are undefined at power-up.

TABLE 28-1: CLCx DATA INPUT SELECTION

Data Input	dy DxS<5:0>	CLCx
LCx_in[38]	100110	MD1_out(1) or MD2_out(2) or Reserved(3)
LCx_in[37]	100101	FOSC
LCx_in[36]	100100	HFINTOSC
LCx_in[35]	100011	LFINTOSC
LCx_in[34]	100010	FRC (ADC RC clock)
LCx_in[33]	100001	IOCIF Set
LCx_in[32]	100000	Timer6_postscaled
LCx_in[31]	011111	Timer4_postscaled
LCx_in[30]	011110	Timer2_postscaled
LCx_in[29]	011101	Timer5 Overflow
LCx_in[28]	011100	Timer3 Overflow
LCx_in[27]	011011	Timer1 Overflow
LCx_in[26]	011010	Timer0 Overflow
LCx_in[25]	011001	EUSART RX
LCx_in[24]	011000	EUSART TX
LCx_in[23]	010111	ZCD1_output
LCx_in[22]	010110	MSSP1 SDO/SDA
LCx_in[21]	010101	MSSP1 SCL/SCK
LCx_in[20]	010100	PWM6_out
LCx_in[19]	010011	PWM5_out
LCx_in[18]	010010	PWM4_out
LCx_in[17]	010001	PWM3_out
LCx_in[16]	010000	CCP2_out
LCx_in[15]	001111	CCP1_out
LCx_in[14]	001110	COG2B
LCx_in[13]	001101	COG2A
LCx_in[12]	001100	COG1B
LCx_in[11]	001011	COG1A
LCx_in[10]	001010	sync_C4OUT
LCx_in[9]	001001	sync_C3OUT
LCx_in[8]	001000	sync_C2OUT
LCx_in[7]	000111	sync_C1OUT
LCx_in[6]	000110	LC3_out from the CLC3
LCx_in[5]	000101	LC2_out from the CLC2
LCx_in[4]	000100	LC1_out from the CLC1
LCx_in[3]	000011	CLCIN3 Pin Input Selected in CLCIN3PPS Register
LCx_in[2]	000010	CLCIN2 Pin Input Selected in CLCIN2PPS Register
LCx_in[1]	000001	CLCIN1 Pin Input Selected in CLCIN1PPS Register
LCx_in[0]	000000	CLCIN0 Pin Input Selected in CLCIN0PPS Register

Note 1: CLCxSEL0 only.
2: PIC16(L)F1768/9 CLCxSEL1 only.
3: CLCxSEL2, CLCxSEL3 and PIC16(L)F1764/5 CLCxSEL1 only.

30.1.2.2 Rising Ramp

The Rising Ramp mode is identical to the Slope Compensation mode, except that the ramps have a rising slope instead of a falling slope. One side of the internal capacitor is connected to the voltage input source and the other side is connected to the internal current source. The internal current source charges this capacitor at a programmable rate. As the capacitor charges, the capacitor voltage is added to the voltage source, producing a linear voltage rise at the required rate (see Figure 30-5). The ramp terminates and the capacitor is discharged when the set_falling timing input goes true. The next ramp starts when the set_rising timing input goes true.

Enabling the optional one-shot by setting the OS bit of the PRGxCON0 register ensures that the capacitor is fully discharged by overriding the set_rising timing input and holding the shorting switch closed for at least the one-shot period, typically 50 ns. Edge-sensitive timing inputs that occur during the one-shot period will be ignored. Level-sensitive timing inputs that occur during, and extend beyond, the one-shot period will be suspended until the end of the one-shot time.

30.2 Enable, Ready, Go

The EN bit of the PRGxCON0 register enables the analog circuitry including the current sources. This permits preparing the PRG module for use and allowing it to become stable before putting it into operation. When the EN bit is set, then the timing inputs are enabled so that initial ramp action can be determined before the GO bit is set. The capacitor shorting switch is closed when the EN bit is set and remains closed while the GO bit is zero.

The RDY bit of the PRGxCON1 register indicates that the analog circuits and current sources are stable.

The GO bit of the PRGxCON0 register enables the switch control circuits, thereby putting the PRG into operation. The GO transition, from cleared to set, triggers the one-shot, thereby extending the capacitor shorting switch closure for the one-shot period.

To ensure predictable operation, set the EN bit first, then wait for the RDY bit to go high before setting the GO bit.

30.3 Independent Set_rising and Set_falling Timing Inputs

The timing inputs determine when the ramp starts and stops. In the Alternating Rising/Falling mode, the ramp rises when the set_rising input goes true and falls when the set_falling input goes true. In the Slope Compensation and Rising Ramp modes, the capacitor is discharged when the set_falling timing input goes true and the ramp starts when the set_rising timing input goes true. The set_falling input dominates the set_rising input.

30.4 Level and Edge Timing Sensitivity

The set_rising and set_falling timing inputs can be independently configured as either level or edge-sensitive.

Level-sensitive operation is useful when it is necessary to detect a timing input true state after an overriding condition ceases. For example, level sensitivity is useful for capacitor generated timing inputs that may be suppressed by the overriding action of the one-shot. With level sensitivity, a capacitor output that changes during the one-shot period will be detected at the end of the one-shot time. With edge sensitivity, the change would be ignored.

Edge-sensitive operation is useful for periodic timing inputs, such as those generated by PWMs and clocks. The duty cycle of a level-sensitive periodic signal may interfere with the other timing input. Consider an Alternating Ramp mode with a level-sensitive 50% PWM as the set_rising timing source and a level-sensitive comparator as the set_falling timing source. If the comparator output reverses the ramp while the PWM signal is still high, then the ramp will improperly reverse again when the comparator signal goes low. That same scenario with the set_rising timing input set for edge sensitivity would properly change the ramp output to rising only on the rising edge of the PWM signal.

set_rising and set_falling timing input edge sensitivity is selected with the respective REDG and FEDG bits of the PRGxCON1 register.

30.5 One-Shot Minimum Timing

The one-shot timer ensures a minimum capacitor discharge time in the Slope Compensation and Rising Ramp modes, and a minimum rising or falling ramp duration in the Alternating Ramp mode. Setting the OS bit of the PRGxCON0 register enables the one-shot timer.

30.6 DAC Voltage Sources

When using any of the DACs as the voltage source, expect a voltage offset equal to the current setting times the DAC equivalent resistance. This will be a constant offset in the Slope Compensation and Ramp modes, and a positive/negative step offset in the Alternating mode. To avoid this limitation, feed the DAC output to the PRG input through one of the op amps set for unity gain.

30.7 Operation During Sleep

The PRG module is unaffected by Sleep.

30.8 Effects of a Reset

The PRG module resets to a disabled condition.

31.0 DATA SIGNAL MODULATOR (DSM)

The Data Signal Modulator (DSM) is a peripheral that allows the user to mix a data stream, also known as a modulator signal, with a carrier signal to produce a modulated output.

Both the carrier and the modulator signals are supplied to the DSM module either internally, from the output of a peripheral, or externally through an input pin.

The modulated output signal is generated by performing a logical “AND” operation of both the carrier and modulator signals and then provided to the MDxOUT pin.

The carrier signal is comprised of two distinct and separate signals: a Carrier High (CARH) signal and a Carrier Low (CARL) signal. During the time in which the Modulator (MOD) signal is in a logic high state, the DSM mixes the Carrier High signal with the Modulator signal. When the Modulator signal is in a logic low state, the DSM mixes the Carrier Low signal with the Modulator signal.

Using this method, the DSM can generate the following types of key modulation schemes:

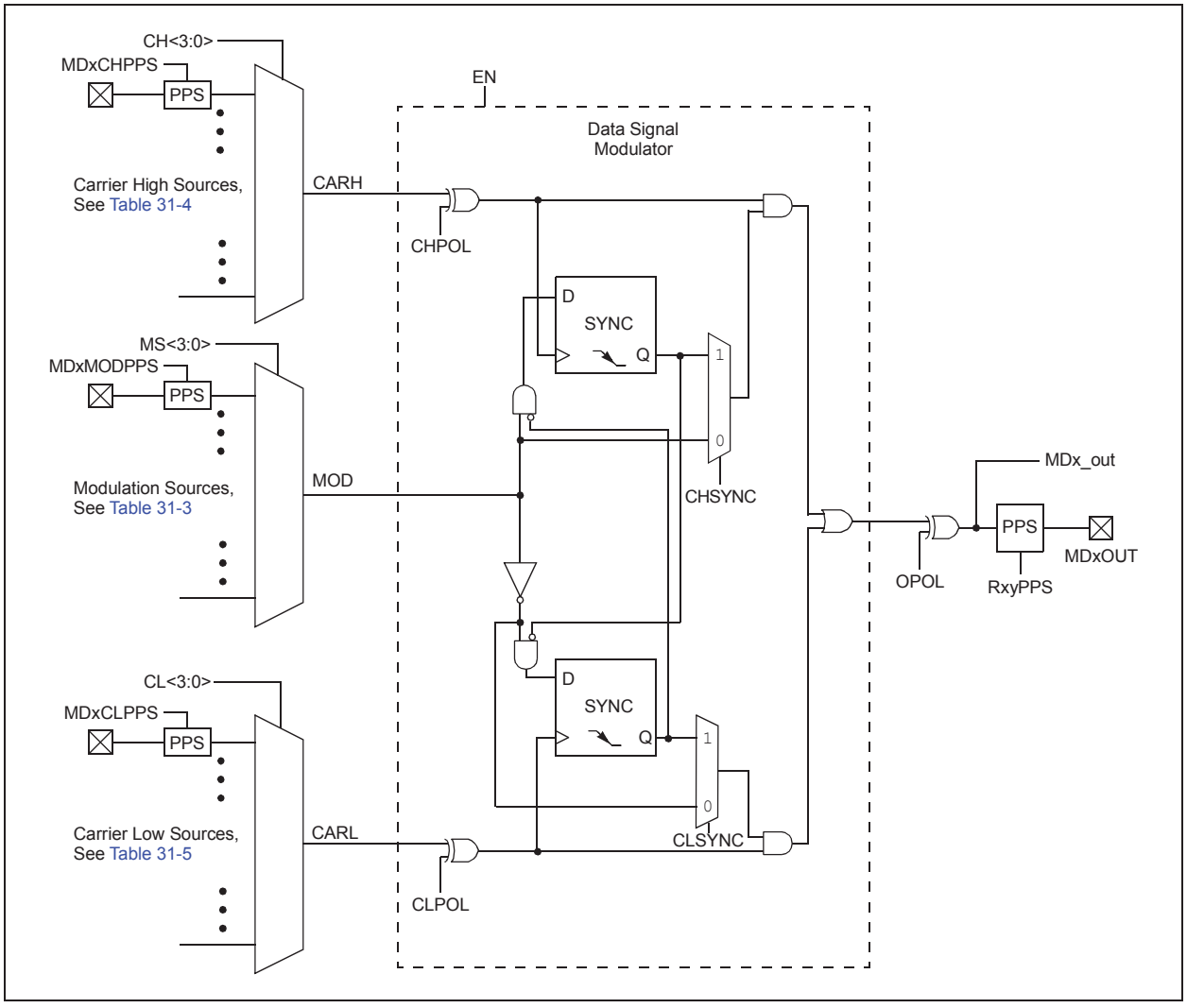
- Frequency-Shift Keying (FSK)
- Phase-Shift Keying (PSK)
- On-Off Keying (OOK)

Additionally, the following features are provided within the DSM module:

- Carrier Synchronization
- Carrier Source Polarity Select
- Carrier Source Pin Disable
- Programmable Modulator Data
- Modulator Source Pin Disable
- Modulated Output Polarity Select
- Slew Rate Control

Figure 31-1 shows a simplified block diagram of the Data Signal Modulator peripheral.

FIGURE 31-1: SIMPLIFIED BLOCK DIAGRAM OF THE DATA SIGNAL MODULATOR



32.2.5 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 V_{DD}.

2: When the SPI is used in Slave mode with CKE set; the user must enable $\overline{SS}$ pin control.

3: While operating 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 32-7: SPI DAISY-CHAIN CONNECTION

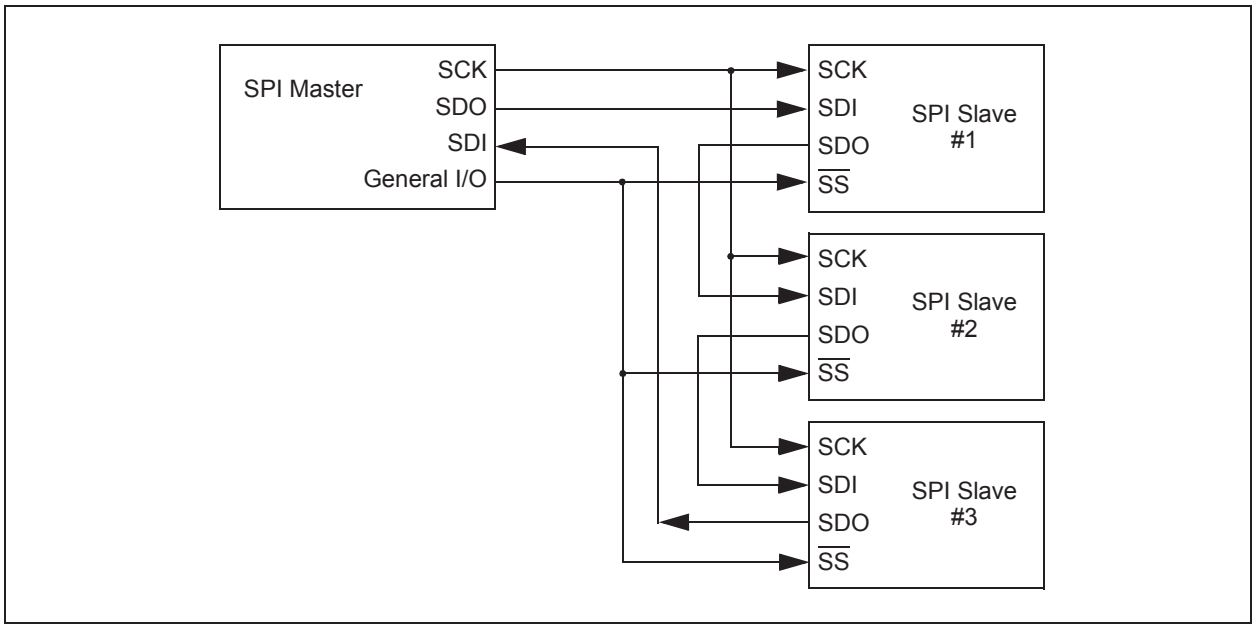


FIGURE 32-8: SLAVE SELECT SYNCHRONOUS WAVEFORM

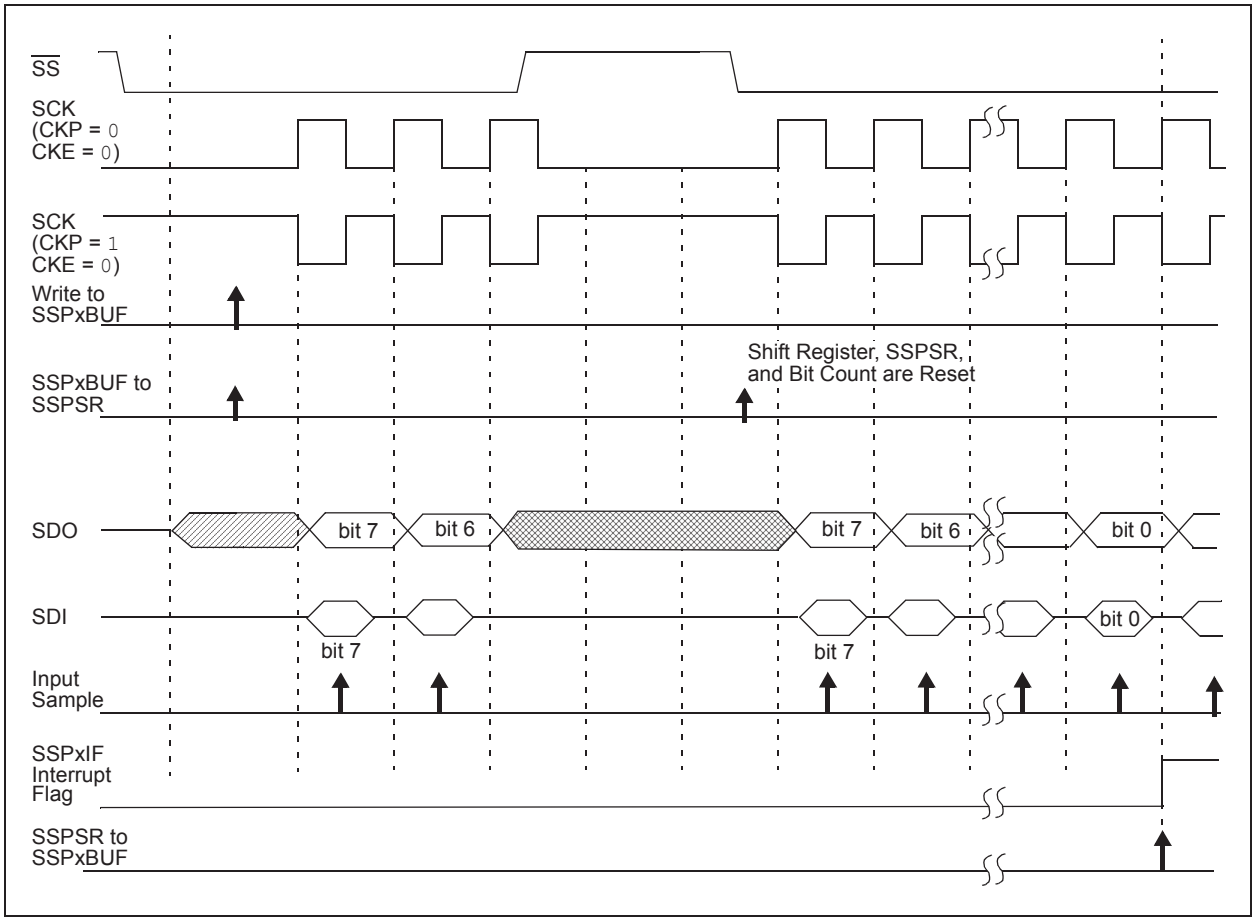


TABLE 33-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPxBRG value (decimal)	Actual Rate	% Error	SPxBRG value (decimal)	Actual Rate	% Error	SPxBRG value (decimal)	Actual Rate	% Error	SPxBRG value (decimal)
300	300.0	0.00	26666	300.0	0.00	16665	300.0	0.00	15359	300.0	0.00	9215
1200	1200	0.00	6666	1200	-0.01	4166	1200	0.00	3839	1200	0.00	2303
2400	2400	0.01	3332	2400	0.02	2082	2400	0.00	1919	2400	0.00	1151
9600	9604	0.04	832	9597	-0.03	520	9600	0.00	479	9600	0.00	287
10417	10417	0.00	767	10417	0.00	479	10425	0.08	441	10433	0.16	264
19.2k	19.18k	-0.08	416	19.23k	0.16	259	19.20k	0.00	239	19.20k	0.00	143
57.6k	57.55k	-0.08	138	57.47k	-0.22	86	57.60k	0.00	79	57.60k	0.00	47
115.2k	115.9k	0.64	68	116.3k	0.94	42	115.2k	0.00	39	115.2k	0.00	23

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPxBRG value (decimal)	Actual Rate	% Error	SPxBRG value (decimal)	Actual Rate	% Error	SPxBRG value (decimal)	Actual Rate	% Error	SPxBRG value (decimal)
300	300.0	0.00	6666	300.0	0.01	3332	300.0	0.00	3071	300.1	0.04	832
1200	1200	-0.02	1666	1200	0.04	832	1200	0.00	767	1202	0.16	207
2400	2401	0.04	832	2398	0.08	416	2400	0.00	383	2404	0.16	103
9600	9615	0.16	207	9615	0.16	103	9600	0.00	95	9615	0.16	25
10417	10417	0	191	10417	0.00	95	10473	0.53	87	10417	0.00	23
19.2k	19.23k	0.16	103	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12
57.6k	57.14k	-0.79	34	58.82k	2.12	16	57.60k	0.00	15	—	—	—
115.2k	117.6k	2.12	16	111.1k	-3.55	8	115.2k	0.00	7	—	—	—

33.5.1.5 Synchronous Master Reception

Data is received at the RX/DT pin. The RX/DT pin output driver is automatically disabled when the EUSART is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RCxSTA register) or the Continuous Receive Enable bit (CREN of the RCxSTA register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character, the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RX/DT pin on the trailing edge of the TX/CK clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCIF bit is set and the character is automatically transferred to the two-character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RCxREG. The RCIF bit remains set as long as there are unread characters in the receive FIFO.

Note:	If the RX/DT function is on an analog pin, the corresponding ANSELx bit must be cleared for the receiver to function.

33.5.1.6 Slave Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a slave receives the clock on the TX/CK line. The TX/CK pin output driver is automatically disabled when the device is configured for synchronous slave transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One data bit is transferred for each clock cycle. Only as many clock cycles should be received as there are data bits.

Note:	If the device is configured as a slave and the TX/CK function is on an analog pin, the corresponding ANSELx bit must be cleared.

33.5.1.7 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCxREG is read to access the FIFO. When this happens, the OERR bit of the RCxSTA register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear, then the error is cleared by reading RCxREG. If the overrun occurred when the CREN bit is set, then the error condition is cleared by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

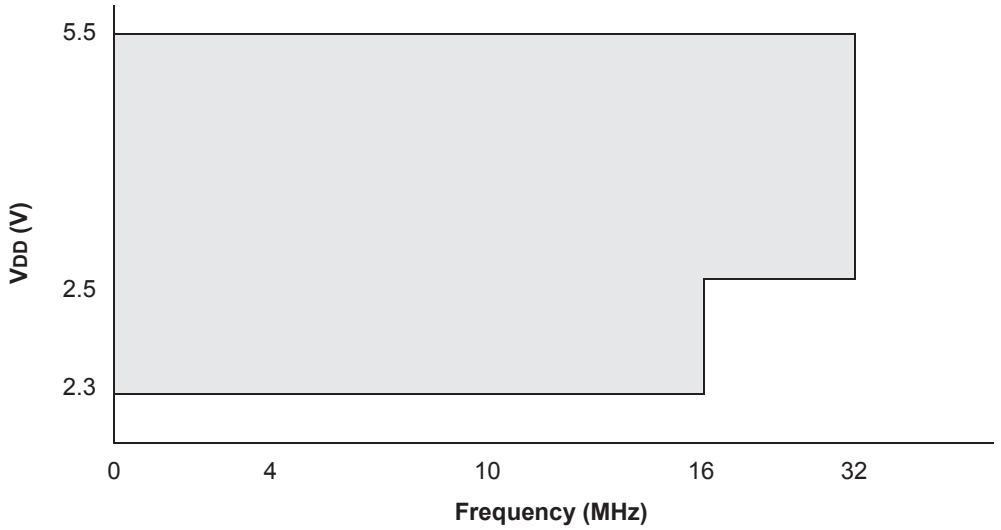
33.5.1.8 Receiving 9-Bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCxSTA register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCxREG.

33.5.1.9 Synchronous Master Reception Setup

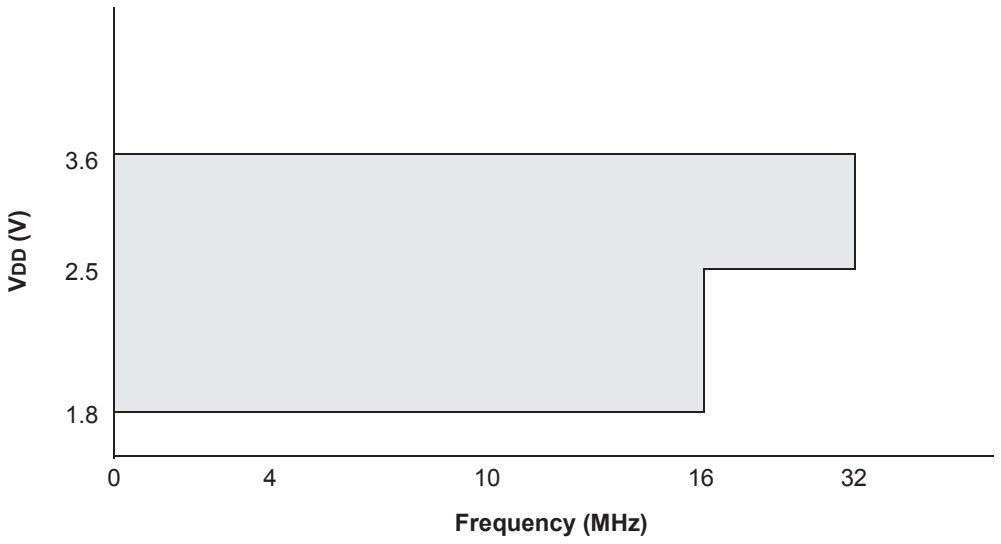
1. Initialize the SPxBRGH, SPxBRGL register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Clear the ANSELx bit for the RX pin (if applicable).
3. Enable the synchronous master serial port by setting bits, SYNC, SPEN and CSRC.
4. Ensure bits, CREN and SREN, are clear.
5. If interrupts are desired, set the RCIE bit of the PIE1 register, and the GIE and PEIE bits of the INTCON register.
6. If 9-bit reception is desired, set bit, RX9.
7. Start reception by setting the SREN bit, or for continuous reception, set the CREN bit.
8. Interrupt flag bit, RCIF, will be set when reception of a character is complete. An interrupt will be generated if the enable bit, RCIE, was set.
9. Read the RCxSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RCxREG register.
11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCxSTA register or by clearing the SPEN bit which resets the EUSART.

FIGURE 36-1: VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C, PIC16F1764/5/8/9 ONLY



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.
2: Refer to Table 36-7 for each Oscillator mode's supported frequencies.

FIGURE 36-2: VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C, PIC16LF1764/5/8/9 ONLY



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.
2: Refer to Table 36-7 for each Oscillator mode's supported frequencies.

TABLE 36-8: OSCILLATOR PARAMETERS

Standard Operating Conditions (unless otherwise stated)								
Param No.	Sym.	Characteristic	Freq. Tolerance	Min.	Typ†	Max.	Units	Conditions
OS08	HFOSC	Internal Calibrated HFINTOSC Frequency(1)	±2%	—	16.0	—	MHz	VDD = 3.0V, TA = 25°C **(Note 2)**
OS08A	MFOSC	Internal Calibrated MFINTOSC Frequency(1)	±2%	—	500	—	kHz	VDD = 3.0V, TA = 25°C **(Note 3)**
OS09	LFOSC	Internal LFINTOSC Frequency	—	—	31	—	kHz	-40°C ≤ TA ≤ +125°C
OS10*	TWARM	HFINTOSC Wake-up from Sleep Start-up Time	—	—	3.2	8	µs	
		MFINTOSC Wake-up from Sleep Start-up Time	—	—	24	35	µs	
		LFINTOSC Wake-up from Sleep Start-up Time	—	—	0.5	—	ms	

* These parameters are characterized but not tested.

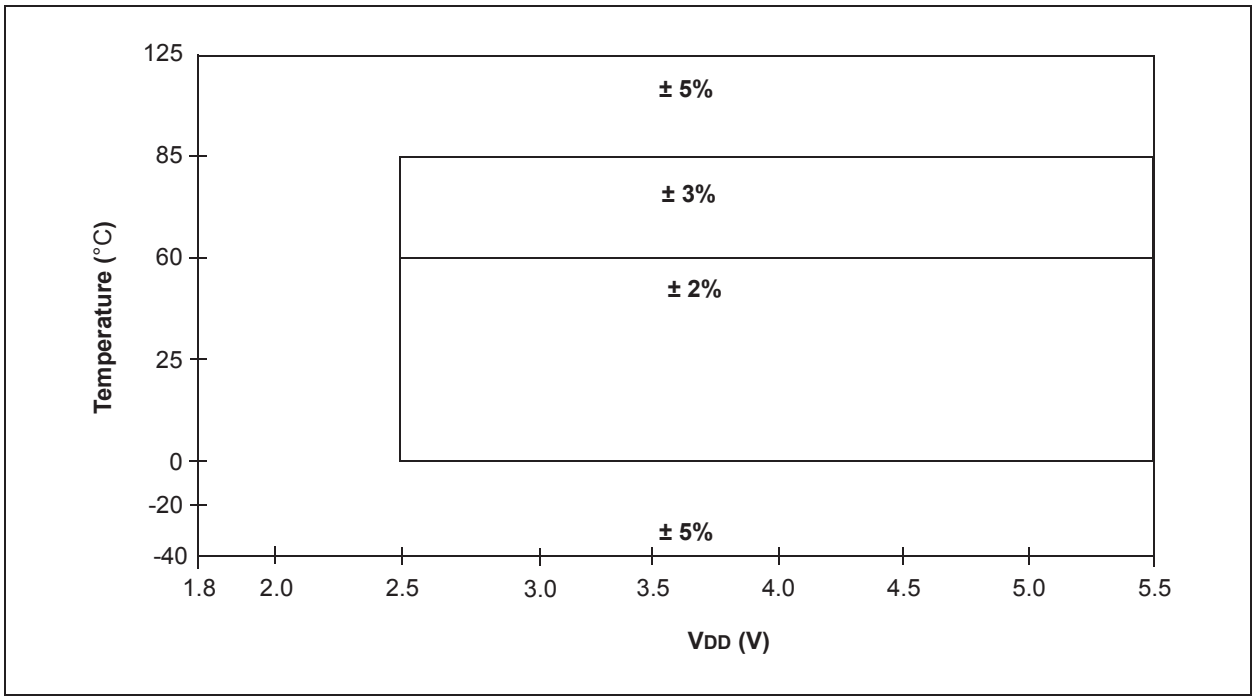
† Data in "Typ" column is at 3.0V, +25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 µF and 0.01 µF values in parallel are recommended.

2: See Figure 37-74: Wake From Sleep, VREGPM = 0. and Figure 37-75: Wake From Sleep, VREGPM = 1.

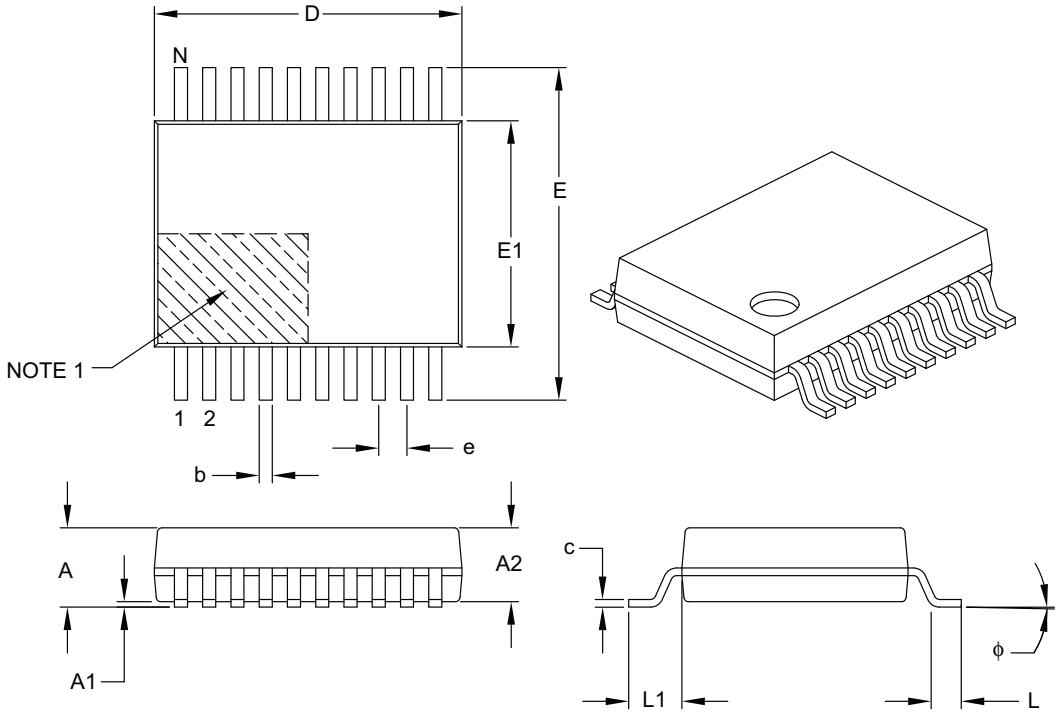
3: See Figure 37-57: LFINTOSC Frequency, PIC16LF1764/5/8/9 Only. and Figure 37-58: LFINTOSC Frequency, PIC16F1764/5/8/9 Only.

FIGURE 36-6: HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE



20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. 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.

Microchip Technology Drawing C04-072B