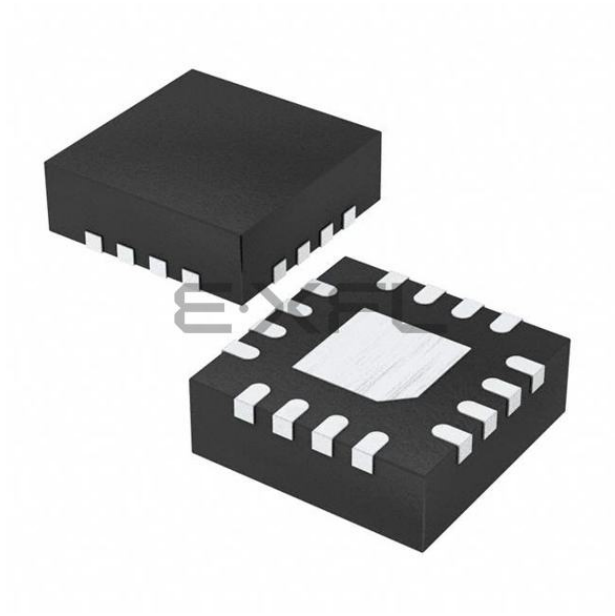


Welcome to E-XFL.COM

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	12
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	16-VQFN Exposed Pad
Supplier Device Package	16-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1823-e-ml

TABLE 3-3: PIC12(L)F1822/16(L)F1823 MEMORY MAP, BANKS 0-7

	BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	INDF0	080h	INDF0	100h	INDF0	180h	INDF0	200h	INDF0	280h	INDF0	300h	INDF0	380h	INDF0
001h	INDF1	081h	INDF1	101h	INDF1	181h	INDF1	201h	INDF1	281h	INDF1	301h	INDF1	381h	INDF1
002h	PCL	082h	PCL	102h	PCL	182h	PCL	202h	PCL	282h	PCL	302h	PCL	382h	PCL
003h	STATUS	083h	STATUS	103h	STATUS	183h	STATUS	203h	STATUS	283h	STATUS	303h	STATUS	383h	STATUS
004h	FSR0L	084h	FSR0L	104h	FSR0L	184h	FSR0L	204h	FSR0L	284h	FSR0L	304h	FSR0L	384h	FSR0L
005h	FSR0H	085h	FSR0H	105h	FSR0H	185h	FSR0H	205h	FSR0H	285h	FSR0H	305h	FSR0H	385h	FSR0H
006h	FSR1L	086h	FSR1L	106h	FSR1L	186h	FSR1L	206h	FSR1L	286h	FSR1L	306h	FSR1L	386h	FSR1L
007h	FSR1H	087h	FSR1H	107h	FSR1H	187h	FSR1H	207h	FSR1H	287h	FSR1H	307h	FSR1H	387h	FSR1H
008h	BSR	088h	BSR	108h	BSR	188h	BSR	208h	BSR	288h	BSR	308h	BSR	388h	BSR
009h	WREG	089h	WREG	109h	WREG	189h	WREG	209h	WREG	289h	WREG	309h	WREG	389h	WREG
00Ah	PCLATH	08Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH	20Ah	PCLATH	28Ah	PCLATH	30Ah	PCLATH	38Ah	PCLATH
00Bh	INTCON	08Bh	INTCON	10Bh	INTCON	18Bh	INTCON	20Bh	INTCON	28Bh	INTCON	30Bh	INTCON	38Bh	INTCON
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	—	30Ch	—	38Ch	—
00Dh	—	08Dh	—	10Dh	—	18Dh	—	20Dh	—	28Dh	—	30Dh	—	38Dh	—
00Eh	PORTC(1)	08Eh	TRISC(1)	10Eh	LATC(1)	18Eh	ANSELC(1)	20Eh	WPUC(1)	28Eh	—	30Eh	—	38Eh	—
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	—	090h	—	110h	—	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	EEADRL	211h	SSP1BUF	291h	CCPR1L	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	EEADRH	212h	SSP1ADD	292h	CCPR1H	312h	—	392h	IOCAN
013h	—	093h	—	113h	CM2CON0(1)	193h	EEDATL	213h	SSP1MASK	293h	CCP1CON	313h	—	393h	IOCAF
014h	—	094h	—	114h	CM2CON1(1)	194h	EEDATH	214h	SSP1STAT	294h	PWM1CON	314h	—	394h	—
015h	TMR0	095h	OPTION	115h	CMOUT	195h	EECON1	215h	SSP1CON1	295h	CCP1AS	315h	—	395h	—
016h	TMR1L	096h	PCON	116h	BORCON	196h	EECON2	216h	SSP1CON2	296h	PSTR1CON	316h	—	396h	—
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	—	217h	SSP1CON3	297h	—	317h	—	397h	—
018h	T1CON	098h	OSCTUNE	118h	DACCON0	198h	—	218h	—	298h	—	318h	—	398h	—
019h	T1GCON	099h	OSCCON	119h	DACCON1	199h	RCREG	219h	—	299h	—	319h	—	399h	—
01Ah	TMR2	09Ah	OSCSTAT	11Ah	SRCON0	19Ah	TXREG	21Ah	—	29Ah	—	31Ah	—	39Ah	CLKRCON
01Bh	PR2	09Bh	ADRESL	11Bh	SRCON1	19Bh	SPBRGL	21Bh	—	29Bh	—	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch	—	29Ch	—	31Ch	—	39Ch	MDCON
01Dh	—	09Dh	ADCON0	11Dh	APFCON	19Dh	RCSTA	21Dh	—	29Dh	—	31Dh	—	39Dh	MDSRC
01Eh	CPSCON0	09Eh	ADCON1	11Eh	—	19Eh	TXSTA	21Eh	—	29Eh	—	31Eh	—	39Eh	MDCARL
01Fh	CPSCON1	09Fh	—	11Fh	—	19Fh	BAUDCON	21Fh	—	29Fh	—	31Fh	—	39Fh	MDCARH
020h	General Purpose Register 80 Bytes	0A0h	General Purpose Register 32 Bytes	120h	Unimplemented Read as '0'	1A0h	Unimplemented Read as '0'	220h	Unimplemented Read as '0'	2A0h	Unimplemented Read as '0'	320h	Unimplemented Read as '0'	3A0h	Unimplemented Read as '0'
06Fh		0BFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
070h	Common RAM	0C0h	Unimplemented Read as '0'	170h	Accesses 70h – 7Fh	1F0h	Accesses 70h – 7Fh	270h	Accesses 70h – 7Fh	2F0h	Accesses 70h – 7Fh	370h	Accesses 70h – 7Fh	3F0h	Accesses 70h – 7Fh
07Fh		0FFh	Accesses 70h – 7Fh	17Fh		1FFh		27Fh		2FFh		37Fh		3FFh	

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

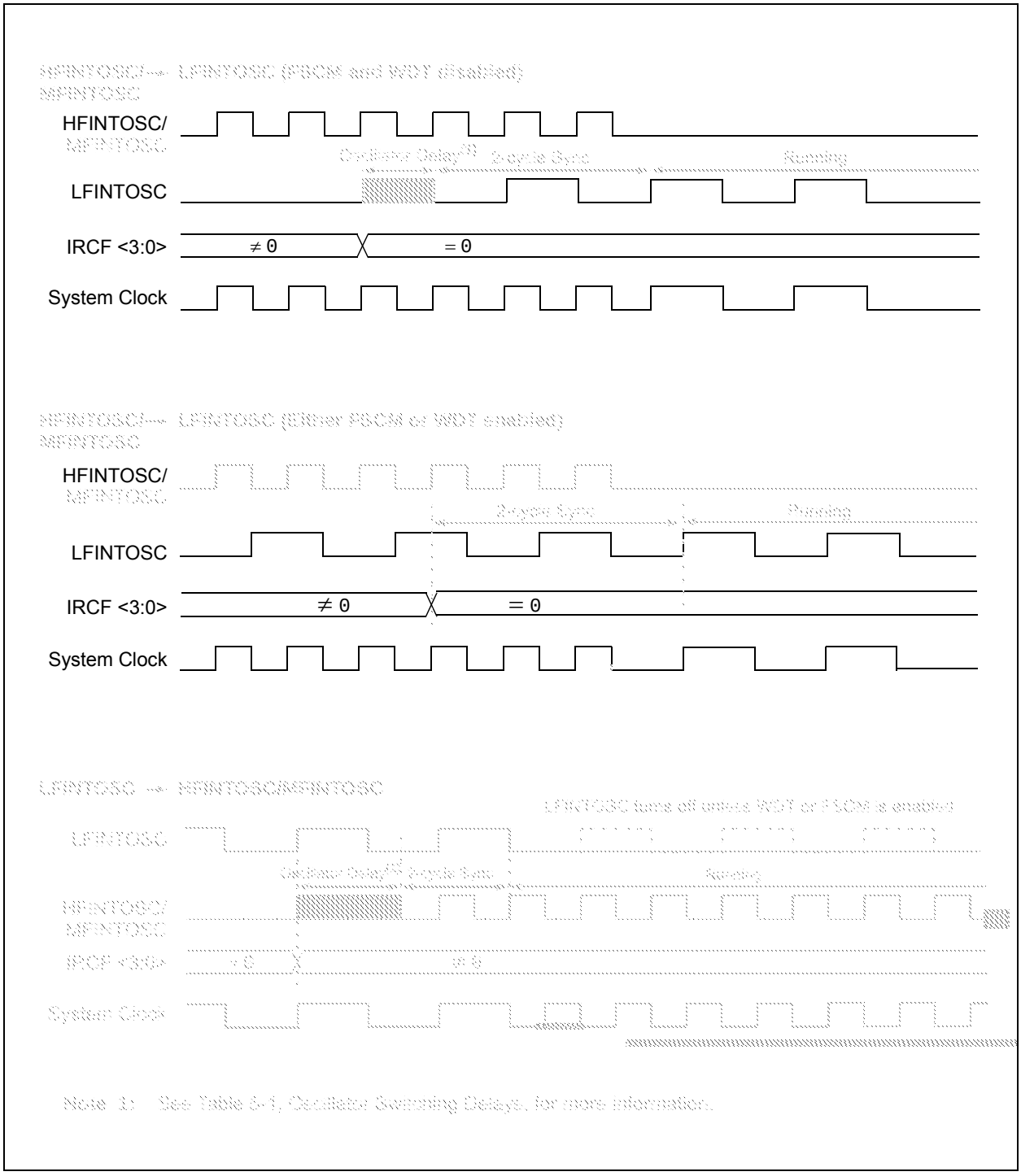
Note 1: Available only on PIC16(L)F1823.

TABLE 3-4: PIC12(L)F1822/16(L)F1823 MEMORY MAP, BANKS 8-15

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	INDF0	480h	INDF0	500h	INDF0	580h	INDF0	600h	INDF0	680h	INDF0	700h	INDF0	780h	INDF0
401h	INDF1	481h	INDF1	501h	INDF1	581h	INDF1	601h	INDF1	681h	INDF1	701h	INDF1	781h	INDF1
402h	PCL	482h	PCL	502h	PCL	582h	PCL	602h	PCL	682h	PCL	702h	PCL	782h	PCL
403h	STATUS	483h	STATUS	503h	STATUS	583h	STATUS	603h	STATUS	683h	STATUS	703h	STATUS	783h	STATUS
404h	FSR0L	484h	FSR0L	504h	FSR0L	584h	FSR0L	604h	FSR0L	684h	FSR0L	704h	FSR0L	784h	FSR0L
405h	FSR0H	485h	FSR0H	505h	FSR0H	585h	FSR0H	605h	FSR0H	685h	FSR0H	705h	FSR0H	785h	FSR0H
406h	FSR1L	486h	FSR1L	506h	FSR1L	586h	FSR1L	606h	FSR1L	686h	FSR1L	706h	FSR1L	786h	FSR1L
407h	FSR1H	487h	FSR1H	507h	FSR1H	587h	FSR1H	607h	FSR1H	687h	FSR1H	707h	FSR1H	787h	FSR1H
408h	BSR	488h	BSR	508h	BSR	588h	BSR	608h	BSR	688h	BSR	708h	BSR	788h	BSR
409h	WREG	489h	WREG	509h	WREG	589h	WREG	609h	WREG	689h	WREG	709h	WREG	789h	WREG
40Ah	PCLATH	48Ah	PCLATH	50Ah	PCLATH	58Ah	PCLATH	60Ah	PCLATH	68Ah	PCLATH	70Ah	PCLATH	78Ah	PCLATH
40Bh	INTCON	48Bh	INTCON	50Bh	INTCON	58Bh	INTCON	60Bh	INTCON	68Bh	INTCON	70Bh	INTCON	78Bh	INTCON
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—
40Eh	—	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	—	70Eh	—	78Eh	—
40Fh	—	48Fh	—	50Fh	—	58Fh	—	60Fh	—	68Fh	—	70Fh	—	78Fh	—
410h	—	490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—
411h	—	491h	—	511h	—	591h	—	611h	—	691h	—	711h	—	791h	—
412h	—	492h	—	512h	—	592h	—	612h	—	692h	—	712h	—	792h	—
413h	—	493h	—	513h	—	593h	—	613h	—	693h	—	713h	—	793h	—
414h	—	494h	—	514h	—	594h	—	614h	—	694h	—	714h	—	794h	—
415h	—	495h	—	515h	—	595h	—	615h	—	695h	—	715h	—	795h	—
416h	—	496h	—	516h	—	596h	—	616h	—	696h	—	716h	—	796h	—
417h	—	497h	—	517h	—	597h	—	617h	—	697h	—	717h	—	797h	—
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	—	798h	—
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	—	799h	—
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	—	79Ah	—
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	—	79Bh	—
41Ch	—	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	—	79Ch	—
41Dh	—	49Dh	—	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	—	79Dh	—
41Eh	—	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—
41Fh	—	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	—	71Fh	—	79Fh	—
420h	Unimplemented Read as '0'	4A0h	Unimplemented Read as '0'	520h	Unimplemented Read as '0'	5A0h	Unimplemented Read as '0'	620h	Unimplemented Read as '0'	6A0h	Unimplemented Read as '0'	720h	Unimplemented Read as '0'	7A0h	Unimplemented Read as '0'
46Fh		4EFh		56Fh		5EFh		66Fh		6EFh		76Fh		7EFh	
470h	Accesses 70h – 7Fh	4F0h	Accesses 70h – 7Fh	570h	Accesses 70h – 7Fh	5F0h	Accesses 70h – 7Fh	670h	Accesses 70h – 7Fh	6F0h	Accesses 70h – 7Fh	770h	Accesses 70h – 7Fh	7F0h	Accesses 70h – 7Fh
47Fh		4FFh		57Fh		5FFh		67Fh		6FFh		77Fh		7FFh	

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

FIGURE 5-7: INTERNAL OSCILLATOR SWITCH TIMING



REGISTER 5-3: OSCTUNE: OSCILLATOR TUNING REGISTER

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
—	—	TUN<5:0>					
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 **Unimplemented:** Read as '0'

bit 5-0 **TUN<5:0>:** Frequency Tuning bits

011111 = Maximum frequency
011110 =
•
•
•
000001 =
000000 = Oscillator module is running at the factory-calibrated frequency.
111111 =
•
•
•
100000 = Minimum frequency

TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		65
OSCSTAT	T1OSCR	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	66
OSCTUNE	—	—	TUN<5:0>						67
PIE2	OSFIE	C2IE(1)	C1IE	EEIE	BCL1IE	—	—	—	88
PIR2	OSFIF	C2IF(1)	C1IF	EEIF	BCL1IF	—	—	—	90
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	173

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.
Note 1: PIC16(L)F1823 only.

TABLE 5-3: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	FCMEN	IESO	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>		$\overline{\text{CPD}}$	46
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		FOSC<2:0>			

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.
Note 1: PIC12F1822/16F1823 only.

8.3 Interrupts During Sleep

Some interrupts can be used to wake from Sleep. To wake from Sleep, the peripheral must be able to operate without the system clock. The interrupt source must have the appropriate Interrupt Enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the SLEEP instruction. The instruction directly after the SLEEP instruction will always be executed before branching to the ISR. Refer to the **Section 9.0 "Power-Down Mode (Sleep)"** for more details.

8.4 INT Pin

The INT pin can be used to generate an asynchronous edge-triggered interrupt. This interrupt is enabled by setting the INTE bit of the INTCON register. The INTEDG bit of the OPTION register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the INTCON register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector.

8.5 Automatic Context Saving

Upon entering an interrupt, the return PC address is saved on the stack. Additionally, the following registers are automatically saved in the shadow registers:

- W register
- STATUS register (except for $\overline{TO}$ and $\overline{PD}$)
- BSR register
- FSR registers
- PCLATH register

Upon exiting the Interrupt Service Routine, these registers are automatically restored. Any modifications to these registers during the ISR will be lost. If modifications to any of these registers are desired, the corresponding shadow register should be modified and the value will be restored when exiting the ISR. The shadow registers are available in Bank 31 and are readable and writable. Depending on the user's application, other registers may also need to be saved.

14.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of VDD, 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)
- Capacitive Sensing (CPS) module

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, DAC and CPS module is routed through two independent programmable gain amplifiers. Each amplifier can be configured to amplify the reference voltage by 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 Comparators, DAC and CPS module. Reference **Section 17.0 "Digital-to-Analog Converter (DAC) Module"**, **Section 19.0 "Comparator Module"** and **Section 27.0 "Capacitive Sensing (CPS) 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 **Section 30.0 "Electrical Specifications"** for the minimum delay requirement.

FIGURE 14-1: VOLTAGE REFERENCE BLOCK DIAGRAM

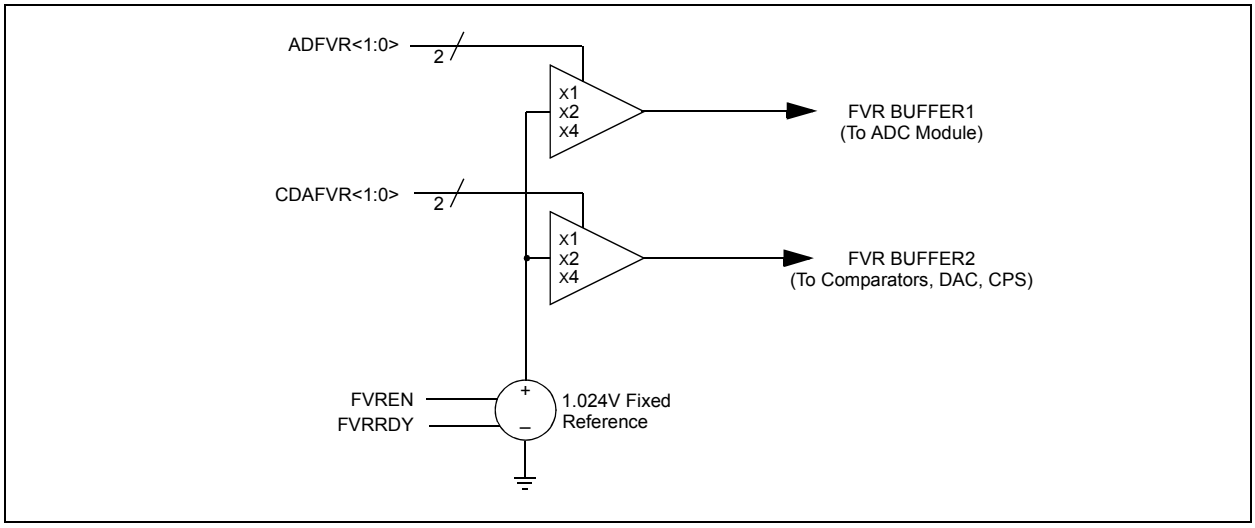


TABLE 18-1: SRCLK FREQUENCY TABLE

SRCLK	Divider	Fosc = 32 MHz	Fosc = 20 MHz	Fosc = 16 MHz	Fosc = 4 MHz	Fosc = 1 MHz
111	512	62.5 kHz	39.0 kHz	31.3 kHz	7.81 kHz	1.95 kHz
110	256	125 kHz	78.1 kHz	62.5 kHz	15.6 kHz	3.90 kHz
101	128	250 kHz	156 kHz	125 kHz	31.25 kHz	7.81 kHz
100	64	500 kHz	313 kHz	250 kHz	62.5 kHz	15.6 kHz
011	32	1 MHz	625 kHz	500 kHz	125 kHz	31.3 kHz
010	16	2 MHz	1.25 MHz	1 MHz	250 kHz	62.5 kHz
001	8	4 MHz	2.5 MHz	2 MHz	500 kHz	125 kHz
000	4	8 MHz	5 MHz	4 MHz	1 MHz	250 kHz

REGISTER 18-1: SRCON0: SR LATCH CONTROL 0 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/S-0/0	R/S-0/0
SRLEN	SRCLK2	SRCLK1	SRCLK0	SRQEN	SRNQEN	SRPS	SRPR
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	S = Bit is set only

bit 7 **SRLEN:** SR Latch Enable bit
1 = SR latch is enabled
0 = SR latch is disabled

bit 6-4 **SRCLK<2:0>:** SR Latch Clock Divider bits
000 = Generates a 1 FOSC wide pulse every 4th FOSC cycle clock
001 = Generates a 1 FOSC wide pulse every 8th FOSC cycle clock
010 = Generates a 1 FOSC wide pulse every 16th FOSC cycle clock
011 = Generates a 1 FOSC wide pulse every 32nd FOSC cycle clock
100 = Generates a 1 FOSC wide pulse every 64th FOSC cycle clock
101 = Generates a 1 FOSC wide pulse every 128th FOSC cycle clock
110 = Generates a 1 FOSC wide pulse every 256th FOSC cycle clock
111 = Generates a 1 FOSC wide pulse every 512th FOSC cycle clock

bit 3 **SRQEN:** SR Latch Q Output Enable bit
If SRLEN = 1:
1 = Q is present on the SRQ pin
0 = External Q output is disabled
If SRLEN = 0:
SR latch is disabled

bit 2 **SRNQEN:** SR Latch $\overline{Q}$ Output Enable bit
If SRLEN = 1:
1 = $\overline{Q}$ is present on the SRnQ pin
0 = External $\overline{Q}$ output is disabled
If SRLEN = 0:
SR latch is disabled

bit 1 **SRPS:** Pulse Set Input of the SR Latch bit(1)
1 = Pulse set input for 1 Q-clock period
0 = No effect on set input.

bit 0 **SRPR:** Pulse Reset Input of the SR Latch bit(1)
1 = Pulse reset input for 1 Q-clock period
0 = No effect on reset input.

Note 1: Set only, always reads back ‘0’.

20.0 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 20-1 is a block diagram of the Timer0 module.

20.1 Timer0 Operation

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

20.1.1 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 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.

20.1.2 8-BIT COUNTER MODE

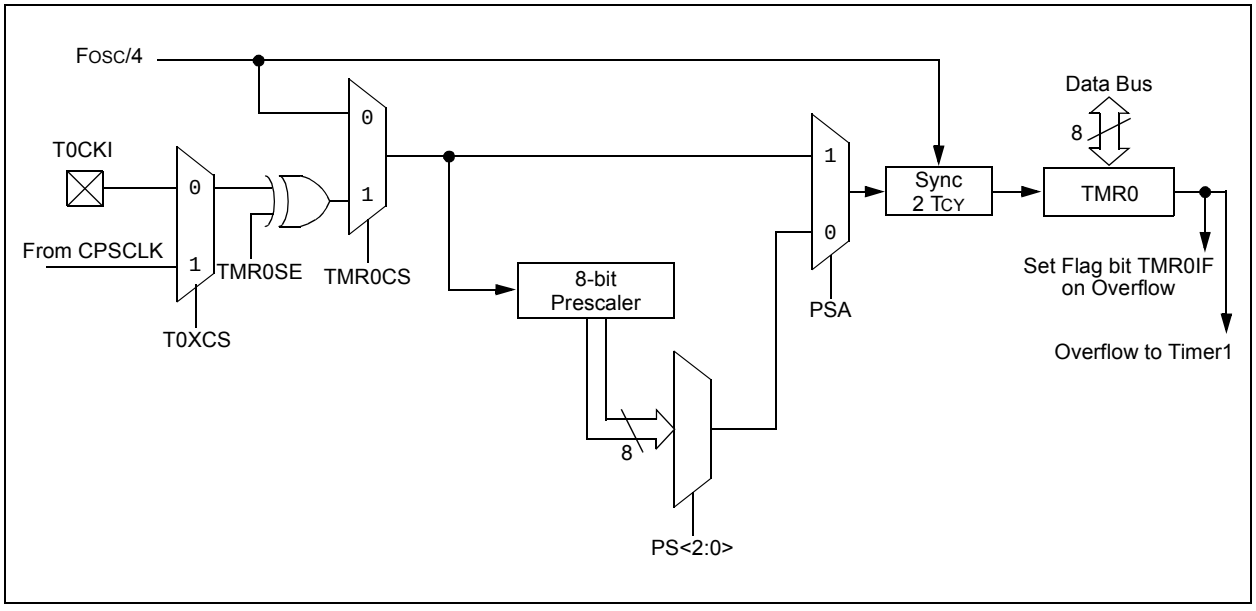
In 8-Bit Counter mode, the Timer0 module will increment on every rising or falling edge of the T0CKI pin or the Capacitive Sensing Oscillator (CPSCLK) signal.

8-Bit Counter mode using the T0CKI pin is selected by setting the TMR0CS bit in the OPTION register to '1' and resetting the T0XCS bit in the CPSCON0 register to '0'.

8-Bit Counter mode using the Capacitive Sensing Oscillator (CPSCLK) signal is selected by setting the TMR0CS bit in the OPTION register to '1' and setting the T0XCS bit in the CPSCON0 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 register.

FIGURE 20-1: BLOCK DIAGRAM OF THE TIMER0



23.0 DATA SIGNAL MODULATOR

The Data Signal Modulator (DSM) is a peripheral which 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 MDOUT 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:

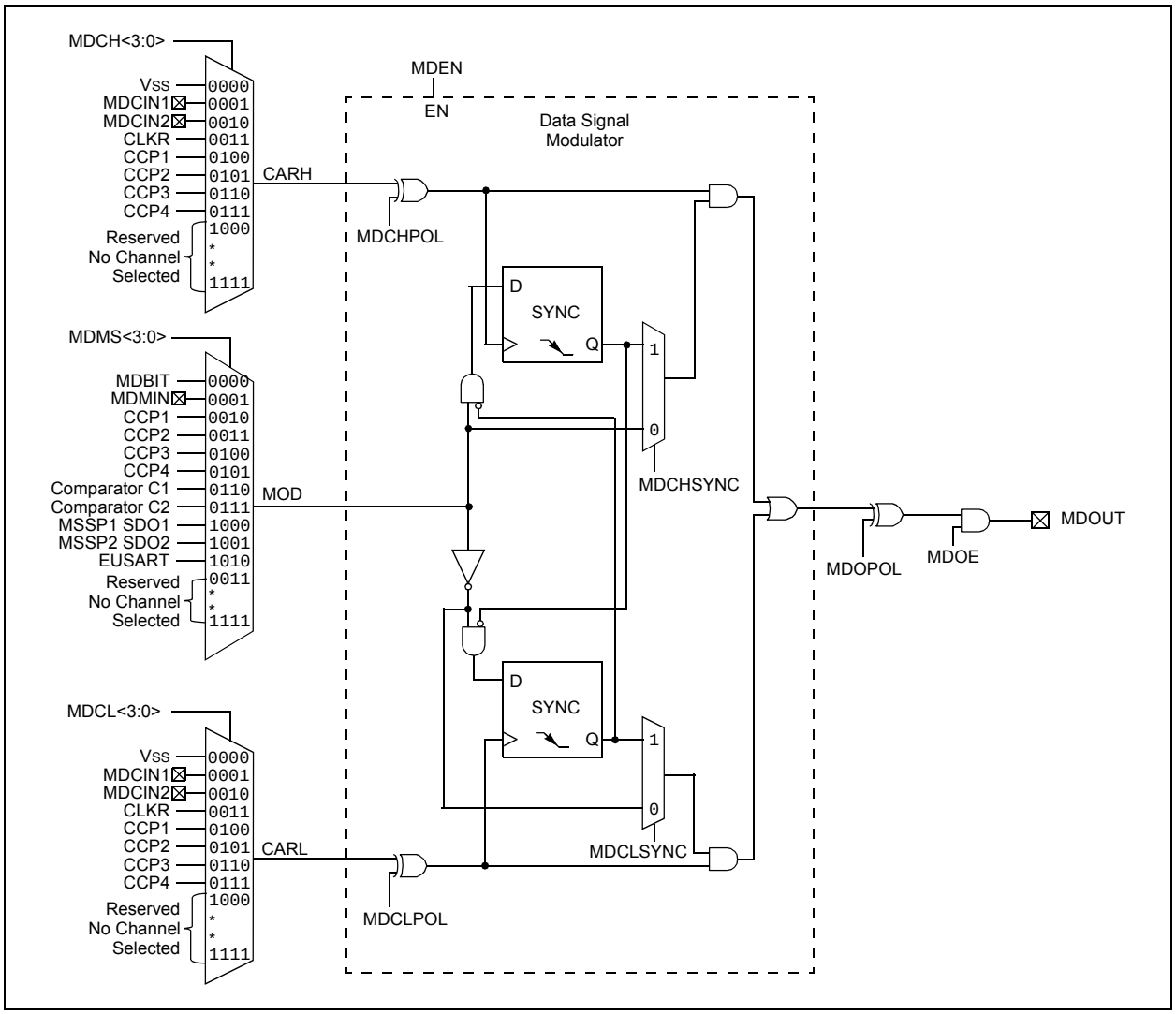
- Frequency-Shift Keying (FSK)
- Phase-Shift Keying (PSK)
- On-Off Keying (OOK)

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 23-1 shows a Simplified Block Diagram of the Data Signal Modulator peripheral.

FIGURE 23-1: SIMPLIFIED BLOCK DIAGRAM OF THE DATA SIGNAL MODULATOR



REGISTER 23-3: MDCARH: MODULATION HIGH CARRIER CONTROL REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
MDCHODIS	MDCHPOL	MDCHSYNC	—	MDCH<3:0>			
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 **MDCHODIS:** Modulator High Carrier Output Disable bit
- 1 = Output signal driving the peripheral output pin (selected by MDCH<3:0>) is disabled
- 0 = Output signal driving the peripheral output pin (selected by MDCH<3:0>) is enabled

bit 6 **MDCHPOL:** Modulator High Carrier Polarity Select bit
- 1 = Selected high carrier signal is inverted
- 0 = Selected high carrier signal is not inverted

bit 5 **MDCHSYNC:** Modulator High Carrier Synchronization Enable bit
- 1 = Modulator waits for a falling edge on the high time carrier signal before allowing a switch to the low time carrier
- 0 = Modulator Output is not synchronized to the high time carrier signal **(1)**

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

bit 3-0 **MDCH<3:0>** Modulator Data High Carrier Selection bits **(1)**
- 1111 = Reserved. No channel connected.
- •
- •
- •
- 0101 = Reserved. No channel connected.
- 0100 = CCP1 output (PWM Output mode only)
- 0011 = Reference Clock module signal (CLKR)
- 0010 = MDCIN2 port pin
- 0001 = MDCIN1 port pin
- 0000 = VSS

Note 1: Narrowed carrier pulse widths or spurs may occur in the signal stream if the carrier is not synchronized.

REGISTER 24-4: PSTR1CON: PWM STEERING CONTROL REGISTER(1)

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-1/1
—	—	—	STR1SYNC	STR1D	STR1C	STR1B	STR1A
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 7-5 **Unimplemented:** Read as '0'

bit 4 **STR1SYNC:** Steering Sync bit
1 = Output steering update occurs on next PWM period
0 = Output steering update occurs at the beginning of the instruction cycle boundary

bit 3 **STR1D:** Steering Enable bit D(2)
1 = P1D pin has the PWM waveform with polarity control from CCP1M<1:0>
0 = P1D pin is assigned to port pin

bit 2 **STR1C:** Steering Enable bit C(2)
1 = P1C pin has the PWM waveform with polarity control from CCP1M<1:0>
0 = P1C pin is assigned to port pin

bit 1 **STR1B:** Steering Enable bit B
1 = P1B pin has the PWM waveform with polarity control from CCP1M<1:0>
0 = P1B pin is assigned to port pin

bit 0 **STR1A:** Steering Enable bit A
1 = P1A pin has the PWM waveform with polarity control from CCP1M<1:0>
0 = P1A pin is assigned to port pin

Note 1: The PWM Steering mode is available only when the CCP1CON register bits CCP1M<3:2> = 11 and P1M<1:0> = 00.
2: PIC16(L)F1823 only.

25.4.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit of the SSP1CON3 register. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

TABLE 25-2: I²C BUS TERMS

TERM	Description
Transmitter	The device which shifts data out onto the bus.
Receiver	The device which shifts data in from the bus.
Master	The device that initiates a transfer, generates clock signals and terminates a transfer.
Slave	The device addressed by the master.
Multi-master	A bus with more than one device that can initiate data transfers.
Arbitration	Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted.
Synchronization	Procedure to synchronize the clocks of two or more devices on the bus.
Idle	No master is controlling the bus, and both SDA and SCL lines are high.
Active	Any time one or more master devices are controlling the bus.
Addressed Slave	Slave device that has received a matching address and is actively being clocked by a master.
Matching Address	Address byte that is clocked into a slave that matches the value stored in SSP1ADD.
Write Request	Slave receives a matching address with R/$\overline{W}$ bit clear, and is ready to clock in data.
Read Request	Master sends an address byte with the R/$\overline{W}$ bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop.
Clock Stretching	When a device on the bus hold SCL low to stall communication.
Bus Collision	Any time the SDA line is sampled low by the module while it is outputting and expected high state.

25.5.2 SLAVE RECEPTION

When the R/$\overline{W}$ bit of a matching received address byte is clear, the R/$\overline{W}$ bit of the SSP1STAT register is cleared. The received address is loaded into the SSP1BUF register and acknowledged.

When the overflow condition exists for a received address, then not Acknowledge is given. An overflow condition is defined as either bit BF of the SSP1STAT register is set, or bit SSP1OV of the SSP1CON1 register is set. The BOEN bit of the SSP1CON3 register modifies this operation. For more information see Register 25-4.

An MSSP1 interrupt is generated for each transferred data byte. Flag bit, SSP1IF, must be cleared by software.

When the SEN bit of the SSP1CON2 register is set, SCL will be held low (clock stretch) following each received byte. The clock must be released by setting the CKP bit of the SSP1CON1 register, except sometimes in 10-bit mode. See **Section 25.2.3 "SPI Master Mode"** for more detail.

25.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSP1 module configured as an I^2C Slave in 7-bit Addressing mode. All decisions made by hardware or software and their effect on reception. Figure 25-14 and Figure 25-15 is used as a visual reference for this description.

This is a step by step process of what typically must be done to accomplish I^2C communication.

1. Start bit detected.
2. S bit of SSP1STAT is set; SSP1IF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit clear is received.
4. The slave pulls SDA low sending an $\overline{ACK}$ to the master, and sets SSP1IF bit.
5. Software clears the SSP1IF bit.
6. Software reads received address from SSP1BUF clearing the BF flag.
7. If SEN = 1; Slave software sets CKP bit to release the SCL line.
8. The master clocks out a data byte.
9. Slave drives SDA low sending an $\overline{ACK}$ to the master, and sets SSP1IF bit.
10. Software clears SSP1IF.
11. Software reads the received byte from SSP1BUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the Master.
13. Master sends Stop condition, setting P bit of SSP1STAT, and the bus goes Idle.

25.5.2.2 7-bit Reception with AHEN and DHEN

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

This list describes the steps that need to be taken by slave software to use these options for I^2C communication. Figure 25-16 displays a module using both address and data holding. Figure 25-17 includes the operation with the SEN bit of the SSP1CON2 register set.

1. S bit of SSP1STAT is set; SSP1IF is set if interrupt on Start detect is enabled.
2. Matching address with R/$\overline{W}$ bit clear is clocked in. SSP1IF is set and CKP cleared after the 8th falling edge of SCL.
3. Slave clears the SSP1IF.
4. Slave can look at the ACKTIM bit of the SSP1CON3 register to determine if the SSP1IF was after or before the $\overline{ACK}$.
5. Slave reads the address value from SSP1BUF, clearing the BF flag.
6. Slave sets $\overline{ACK}$ value clocked out to the master by setting ACKDT.
7. Slave releases the clock by setting CKP.
8. SSP1IF is set after an $\overline{ACK}$, not after a NACK.
9. If SEN = 1 the slave hardware will stretch the clock after the $\overline{ACK}$.
10. Slave clears SSP1IF.

Note: SSP1IF is still set after the 9th falling edge of SCL even if there is no clock stretching and BF has been cleared. Only if NACK is sent to Master is SSP1IF not set

11. SSP1IF set and CKP cleared after eighth falling edge of SCL for a received data byte.
12. Slave looks at ACKTIM bit of SSP1CON3 to determine the source of the interrupt.
13. Slave reads the received data from SSP1BUF clearing BF.
14. Steps 7-14 are the same for each received data byte.
15. Communication is ended by either the slave sending an $\overline{ACK}$ = 1, or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop Detect is disabled, the slave will only know by polling the P bit of the SSTSTAT register.

TABLE 26-2: SUMMARY OF REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	279
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	87
RCREG	EUSART Receive Data Register								273*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	278
SPBRGL	BRG<7:0>								280*
SPBRGH	BRG<15:8>								280*
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	117
TRISC(1)	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	121
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	277

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for Asynchronous Reception.

* Page provides register information.

Note 1: PIC16(L)F1823 only.

26.4.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 26.4.1.5 "Synchronous Master Reception"**), with the following exceptions:

- Sleep
- CREN bit is always set, therefore the receiver is never Idle
- SREN bit, which is a "don't care" in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RCREG register. If the RCIE enable bit is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will branch to the interrupt vector.

26.4.2.4 Synchronous Slave Reception Set-up:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Clear the ANSEL bit for both the CK and DT pins (if applicable).
3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. If 9-bit reception is desired, set the RX9 bit.
5. Set the CREN bit to enable reception.
6. The RCIF bit will be set when reception is complete. An interrupt will be generated if the RCIE bit was set.
7. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCSTA register.
8. Retrieve the eight Least Significant bits from the receive FIFO by reading the RCREG register.
9. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

TABLE 26-10: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	279
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	89
RCREG	EUSART Receive Data Register								273*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	278
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	117
TRISC(1)	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	121
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	277

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for Synchronous Slave Reception.

* Page provides register information.

Note 1: PIC16(L)F1823 only.

27.7.3 FREQUENCY THRESHOLD

The frequency threshold should be placed midway between the value of nominal frequency and the reduced frequency of the capacitive sensing oscillator. Refer to Application Note AN1103, "*Software Handling for Capacitive Sensing*" (DS01103) for more detailed information on the software required for CPS module.

Note:	For more information on general capacitive sensing refer to Application Notes: • AN1101, "*Introduction to Capacitive Sensing*" (DS01101) • AN1102, "*Layout and Physical Design Guidelines for Capacitive Sensing*" (DS01102)

27.8 Operation during Sleep

The capacitive sensing oscillator will continue to run as long as the module is enabled, independent of the part being in Sleep. In order for the software to determine if a frequency change has occurred, the part must be awake. However, the part does not have to be awake when the timer resource is acquiring counts.

Note:	Timer0 does not operate when in Sleep, and therefore cannot be used for capacitive sense measurements in Sleep.

30.6 Thermal Considerations

Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C					
Param No.	**Sym.**	**Characteristic**	**Typ.**	**Units**	**Conditions**
TH01	θJA	Thermal Resistance Junction to Ambient	89.3	°C/W	8-pin PDIP package
			149.5	°C/W	8-pin SOIC package
			56.7	°C/W	8-pin DFN 3X3mm package
			39.4	°C/W	8-pin UDFN 3X3mm package
			70.0	°C/W	14-pin PDIP package
			95.3	°C/W	14-pin SOIC package
			100	°C/W	14-pin TSSOP 4x4mm package
			45.7	°C/W	16-pin QFN 4X4mm package
			31.8	°C/W	16-pin UQFN 4X4mm package
TH02	θJC	Thermal Resistance Junction to Case	43.1	°C/W	8-pin PDIP package
			39.9	°C/W	8-pin SOIC package
			9.0	°C/W	8-pin DFN 3X3mm package
			40.3	°C/W	8-pin UDFN 3X3mm package
			32.0	°C/W	14-pin PDIP package
			31.0	°C/W	14-pin SOIC package
			24.4	°C/W	14-pin TSSOP 4x4mm package
			6.3	°C/W	16-pin QFN 4X4mm package
			24.4	°C/W	16-pin UQFN 4X4mm package
TH03	TJMAX	Maximum Junction Temperature	150	°C	
TH04	PD	Power Dissipation	—	W	PD = PINTERNAL + PI/O
TH05	PINTERNAL	Internal Power Dissipation	—	W	PINTERNAL = IDD x VDD(1)
TH06	PI/O	I/O Power Dissipation	—	W	PI/O = Σ (IOL * VOL) + Σ (IOH * (VDD - VOH))
TH07	PDER	Derated Power	—	W	PDER = PDMAX (TJ - TA)/θJA(2)

Legend: TBD = To Be Determined

Note 1: IDD is current to run the chip alone without driving any load on the output pins.

2: TA = Ambient Temperature. TJ = Junction Temperature.

FIGURE 30-18: SPI SLAVE MODE TIMING (CKE = 0)

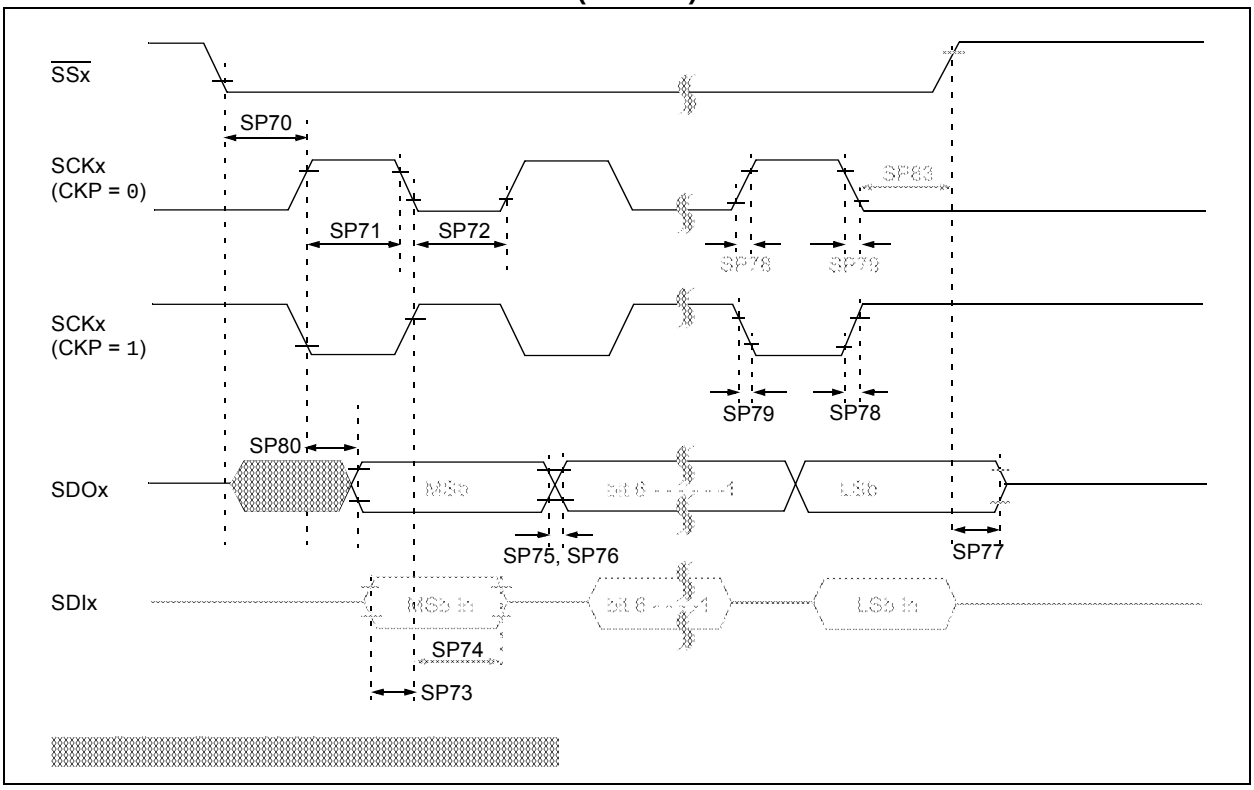


FIGURE 30-19: SPI SLAVE MODE TIMING (CKE = 1)

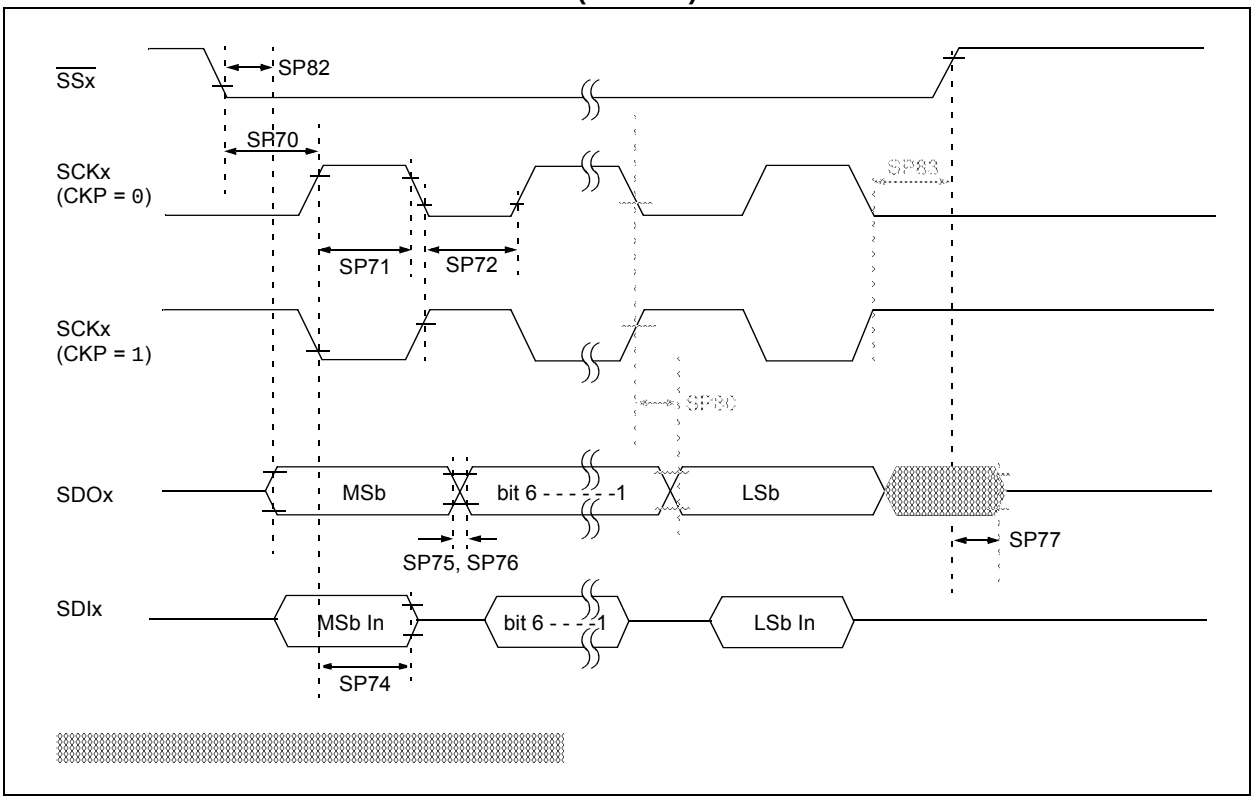


FIGURE 31-21: IDD, MFINTOSC MODE (FOSC = 500 kHz), PIC12LF1822 AND PIC16LF1823 ONLY

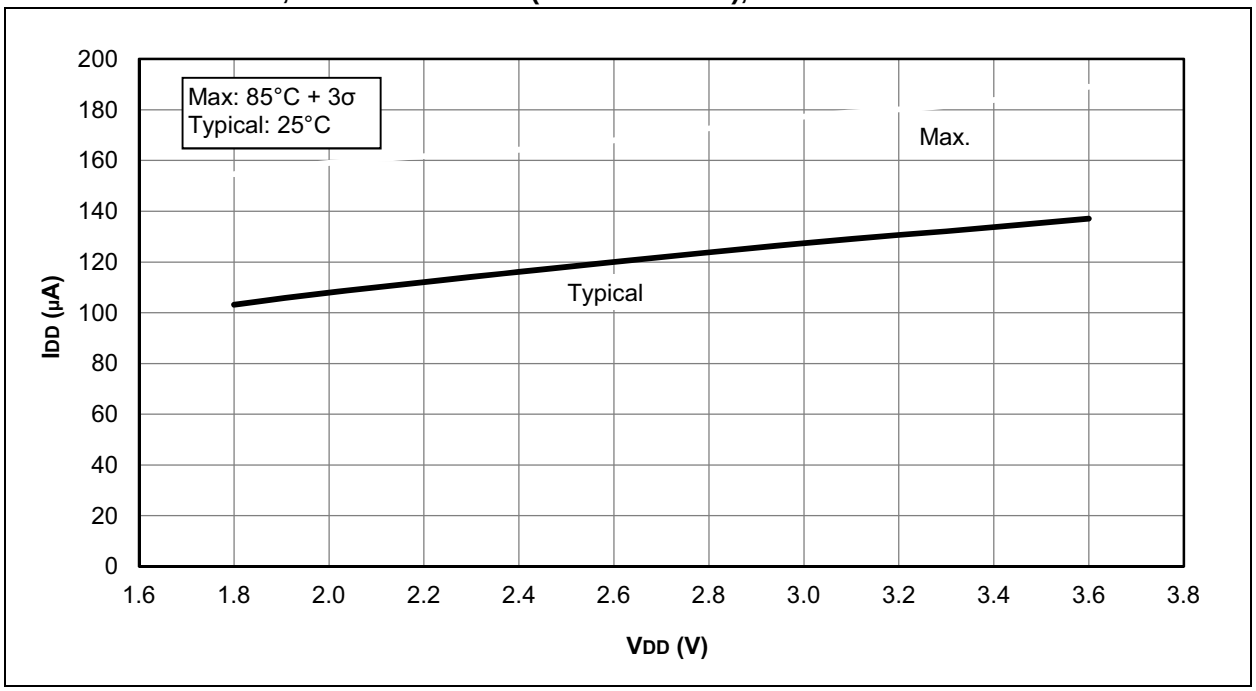
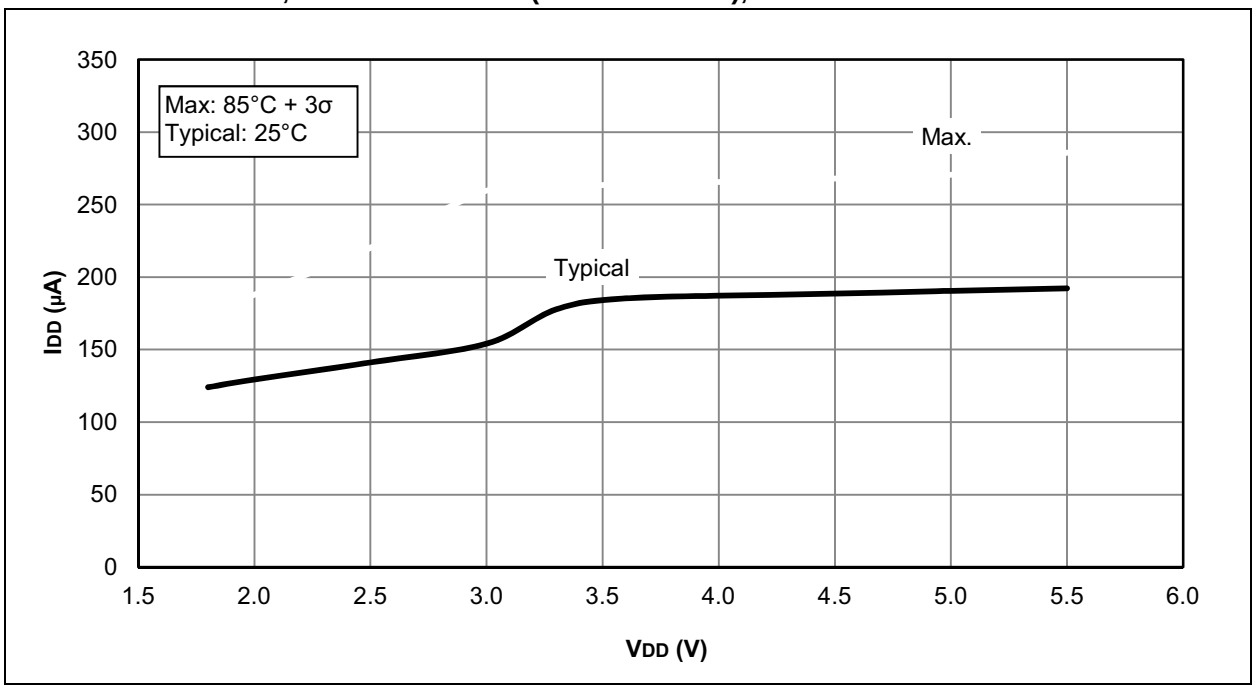


FIGURE 31-22: IDD, MFINTOSC MODE (FOSC = 500 kHz), PIC12F1822 AND PIC16F1823 ONLY



Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

ISBN: 978-1-63277-253-4

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.