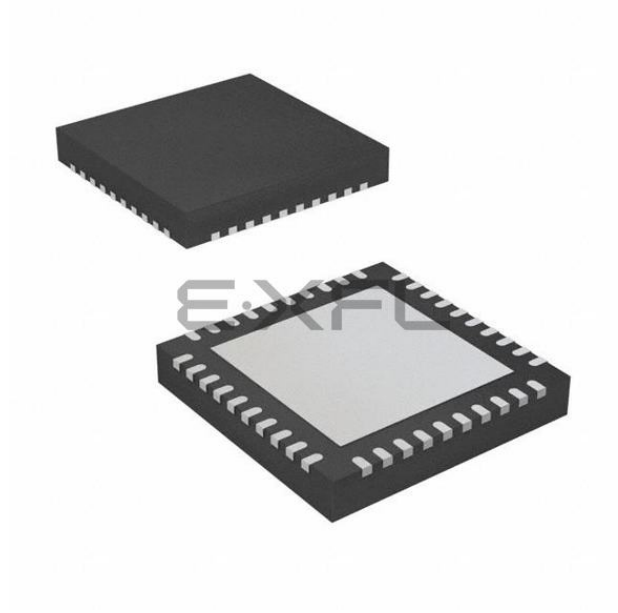


Welcome to E-XFL.COM

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	36
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 28x10b; D/A 4x5b, 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	40-UFQFN Exposed Pad
Supplier Device Package	40-UQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1779t-i-mv

FIGURE 3-6: ACCESSING THE STACK EXAMPLE 2

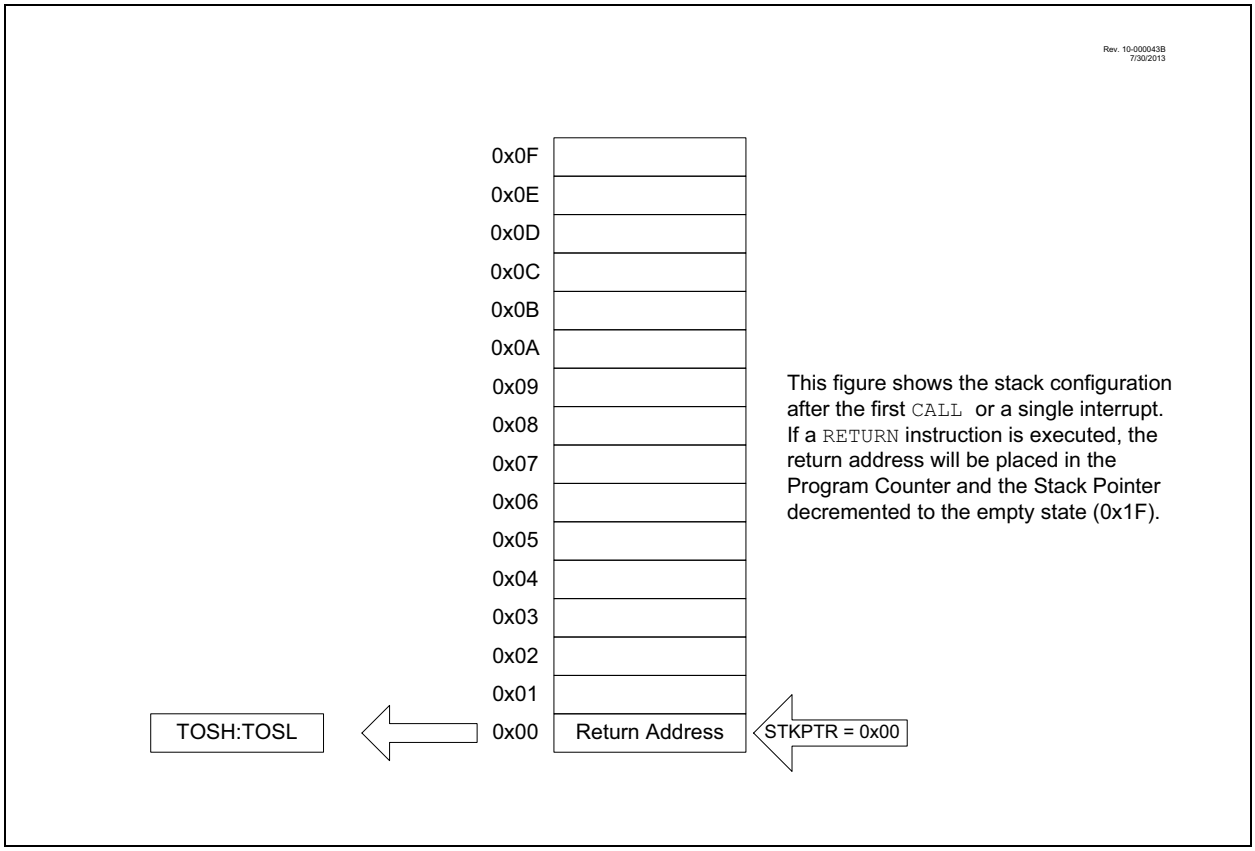
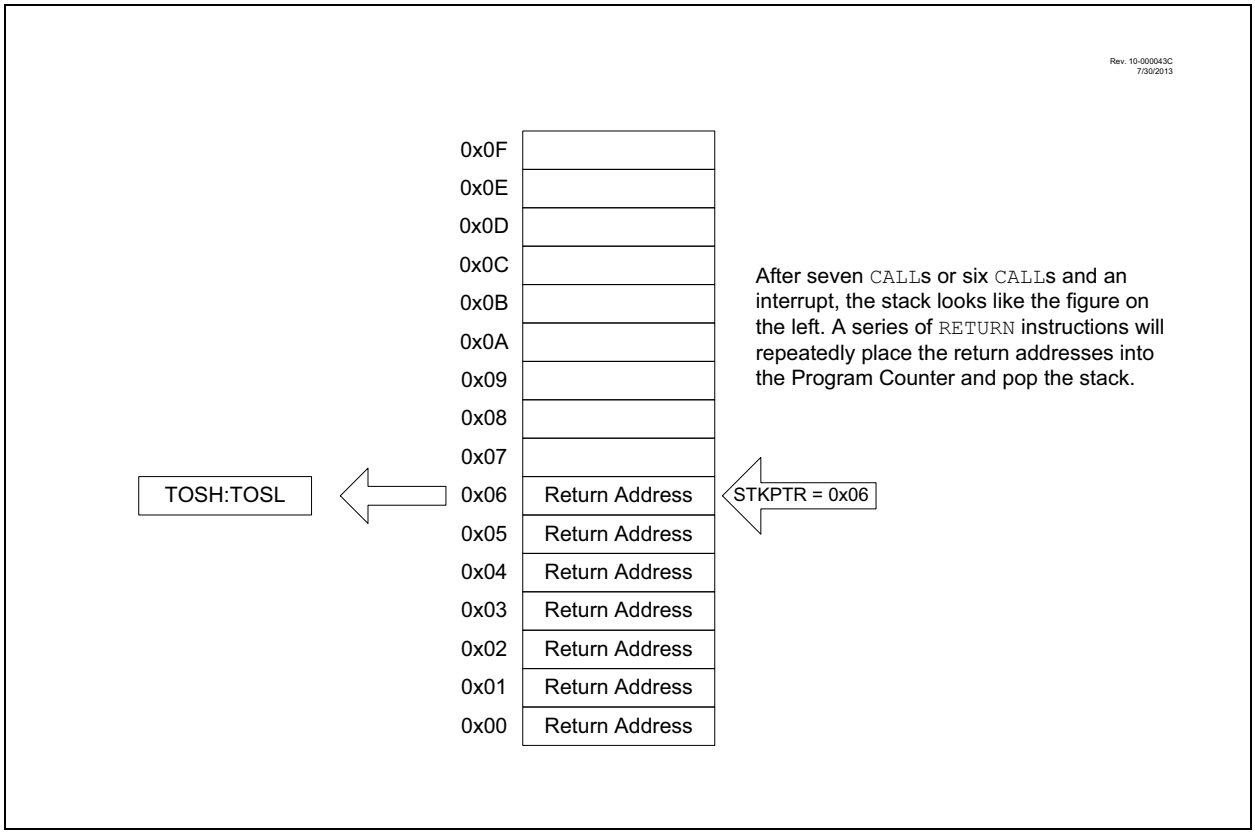


FIGURE 3-7: ACCESSING THE STACK EXAMPLE 3



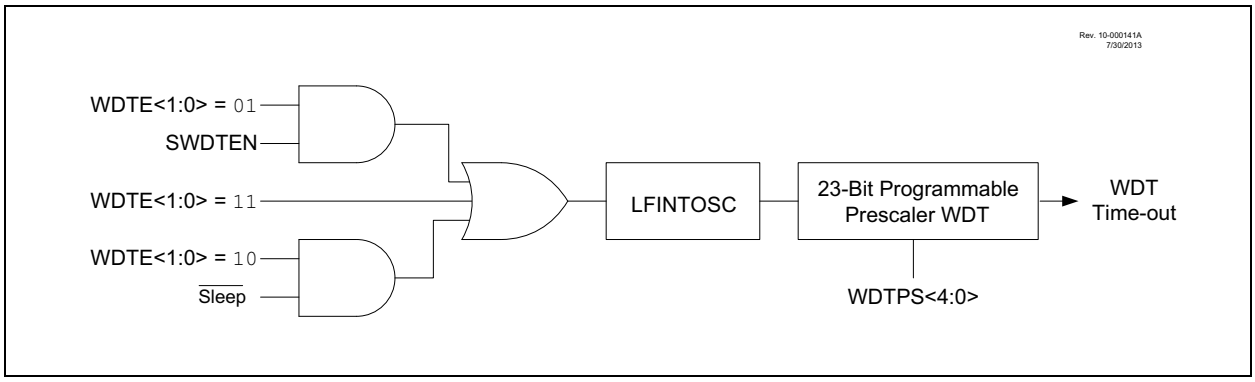
9.0 WATCHDOG TIMER (WDT)

The Watchdog Timer is a system timer that generates a Reset if the firmware does not issue a CLRWDT instruction within the time-out period. The Watchdog Timer is typically used to recover the system from unexpected events.

The WDT has the following features:

- Independent clock source
- Multiple operating modes
  - WDT is always on
  - WDT is off when in Sleep
  - WDT is controlled by software
  - WDT is always off
- Configurable time-out period is from 1 ms to 256 seconds (nominal)
- Multiple Reset conditions
- Operation during Sleep

FIGURE 9-1: WATCHDOG TIMER BLOCK DIAGRAM



11.10 Register Definitions: PORTE

REGISTER 11-34: PORTE: PORTE REGISTER

U-0	U-0	U-0	U-0	R-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	RE3	RE2(1)	RE1(1)	RE0(1)
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 7-4 **Unimplemented**: Read as '0'

bit 3-0 **RE<3:0>**: PORTE I/O Pin bits(1)
1 = Port pin is > VIH
0 = Port pin is < VIL

Note 1: RE<2:0> are not implemented on the PIC16(L)F1778. Read as '0'. Writes to RE<2:0> are actually written to corresponding LATE register. Reads from PORTE register is the return of actual I/O pin values.

REGISTER 11-35: TRISE: PORTE TRI-STATE REGISTER

U-0	U-0	U-0	U-0	U-1(2)	R/W-1	R/W-1	R/W-1
—	—	—	—	—	TRISE2(1)	TRISE1(1)	TRISE0(1)
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 7-4 **Unimplemented**: Read as '0'

bit 3 **Unimplemented**: Read as '1'

bit 2-0 **TRISE<2:0>:** RE<2:0> Tri-State Control bits(1)
1 = PORTE pin configured as an input (tri-stated)
0 = PORTE pin configured as an output

Note 1: TRISE<2:0> are not implemented on the PIC16(L)F1778.
2: Unimplemented, read as '1'.

REGISTER 11-40: SLRCONE: PORTE SLEW RATE CONTROL REGISTER(1)

U-0	U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	—	SLRE2	SLRE1	SLRE0
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 7-3 **Unimplemented:** Read as '0'

bit 2-0 **SLRE<2:0>:** PORTE Slew Rate Enable bits
For RE<2:0> pins
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

Note 1: The SLRCONE register is not implemented on the PIC16(L)F1778.

REGISTER 11-41: INLVLE: PORTE INPUT LEVEL CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	INLVLE3	INLVLE2(1)	INLVLE1(1)	INLVLE0(1)
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 7-4 **Unimplemented:** Read as '0'

bit 3-0 **INLVLE<3:0>:** PORTE Input Level Select bit
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

Note 1: INLVLE<2:0> are not implemented on the PIC16(L)F1778.

TABLE 11-7: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELE(2)	—	—	—	—	—	ANSELE2	ANSELE1	ANSELE0	200
INLVLE	—	—	—	—	INLVLE3	INLVLE2(2)	INLVLE1(2)	INLVLE0(2)	202
LATE(2)	—	—	—	—	—	LATE2	LATE1	LATE0	200
ODCONE(2)	—	—	—	—	—	ODE2	ODE1	ODE0	201
PORTE	—	—	—	—	RE3	RE2(2)	RE1(2)	RE0(2)	199
SLRCONE(2)	—	—	—	—	—	SLRE2	SLRE1	SLRE0	202
TRISE	—	—	—	—	—(1)	TRISE2(2)	TRISE1(2)	TRISE0(2)	199
WPUE	—	—	—	—	WPUE3	WPUE2(2)	WPUE1(2)	WPUE0(2)	201

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by PORTE.
Note 1: Unimplemented, read as '1'.
2: PIC16(L)F1777/9 only.

TABLE 12-1: PPS INPUT REGISTER RESET VALUES

Peripheral	xxxPPS Register	Default Pin Selection	Reset Value (xxxPPS<5:0>)	Port Selection PIC16(L)F1777/9					Port Selection PIC16(L)F1778		
		PIC16(L)F1777/8/9	PIC16(L)F1777/8/9	A	B	C	D	E	A	B	C
Interrupt-on-change	INTPPS	RB0	001000	•	•				•	•	
Timer0clock	T0CKIPPS	RA4	000100	•	•				•	•	
Timer1clock	T1CKIPPS	RC0	010000	•		•			•		•
Timer1 gate	T1GPPS	RB5	001101		•	•				•	•
Timer3 clock	T3CKIPPS	RC0	010000		•	•				•	•
Timer3 gate	T3GPPS	RC0	010000	•		•			•		•
Timer5 clock	T5CKIPPS	RC2	010010	•		•			•		•
Timer5 gate	T5GPPS	RB4	001100		•		•			•	•
Timer2 input	T2INPPS	RC3	010011	•		•			•		•
Timer4 input	T4INPPS	RC5	010101		•	•				•	•
Timer6 input	T6INPPS	RB7	001111		•		•			•	•
Timer8 input	T8INPPS	RC4	010100		•		•			•	•
CCP1	CCP1PPS	RC2	010010		•	•				•	•
CCP2	CCP2PPS	RC1	010001		•	•				•	•
CCP7	CCP7PPS	RB5	001101		•		•			•	•
CCP8(1)	CCP8PPS	RB0	001000		•		•			•	•
COG1	COG1INPPS	RB0	001000		•		•			•	•
COG2	COG2INPPS	RB1	001001		•		•			•	•
COG3	COG3INPPS	RB2	001010		•		•			•	•
COG4(1)	COG4INPPS	RB3	001011		•		•				
DSM1 low carrier	MD1CLPPS	RA3	000011	•			•		•		•
DSM1 high carrier	MD1CHPPS	RA4	000100	•			•		•		•
DSM1 modulation	MD1MODPPS	RA5	000101	•			•		•		•
DSM2 low carrier	MD2CLPPS	RC3	010011	•			•		•		•
DSM2 high carrier	MD2CHPPS	RC4	010100	•			•		•		•
DSM2 modulation	MD2MODPPS	RC5	010101	•			•		•		•
DSM3 low carrier	MD3CLPPS	RB3	001011		•		•			•	•
DSM3 high carrier	MD3CHPPS	RB4	001100		•		•			•	•
DSM3 modulation	MD3MODPPS	RB5	001101		•		•			•	•
DSM4 low carrier(1)	MD4CLPPS	RB0	001000		•		•				
DSM4 high carrier(1)	MD4CHPPS	RB1	001001		•		•				
DSM4 modulation(1)	MD4MODPPS	RB2	001010		•		•				
PRG1 set rising	PRG1RPPS	RA4	000100	•			•		•		•
PRG1 set falling	PRG1FPPS	RA5	000101	•			•		•		•
PRG2 set rising	PRG2RPPS	RC1	010001	•			•		•		•
PRG2 set falling	PRG2FPPS	RC2	010010	•			•		•		•
PRG3 set rising	PRG3RPPS	RC4	010100		•		•			•	•
PRG3 set falling	PRG3FPPS	RC5	010101		•		•			•	•
PRG4 set rising(1)	PRG4RPPS	RB1	010100		•		•				
PRG4set falling(1)	PRG4FPPS	RB2	010101		•		•				
ADC trigger	ADCACTPPS	RB4	001100		•		•			•	•

Example: CCP1PPS = 0x13 selects RC3 as the CCP1 input.
Note 1: PIC16(L)F1777/9 only

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

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

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

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

REGISTER 13-7: IOCCP: INTERRUPT-ON-CHANGE PORTC POSITIVE EDGE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCCP7	IOCCP6	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
bit 7							bit 0

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

bit 7-0 **IOCCP<7:0>:** Interrupt-on-Change PORTC Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive going edge. IOCCFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 13-8: IOCCN: INTERRUPT-ON-CHANGE PORTC NEGATIVE EDGE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCCN7	IOCCN6	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0
bit 7							bit 0

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

bit 7-0 **IOCCN<7:0>:** Interrupt-on-Change PORTC Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative going edge. IOCCFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 13-9: IOCCF: INTERRUPT-ON-CHANGE PORTC FLAG REGISTER

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
IOCCF7	IOCCF6	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0
bit 7							bit 0

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

bit 7-0 **IOCCF<7:0>:** Interrupt-on-Change PORTC Flag bits
1 = An enabled change was detected on the associated pin.
Set when IOCCPx = 1 and a rising edge was detected on RCx, or when IOCCNx = 1 and a falling edge was detected on RCx.
0 = No change was detected, or the user cleared the detected change.

21.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A software programmable prescaler is available for exclusive use with Timer0. The prescaler is enabled by clearing the PSA bit of the OPTION_REG register.

Note:	The Watchdog Timer (WDT) uses its own independent prescaler.

There are eight prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits of the OPTION_REG register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be disabled by setting the PSA bit of the OPTION_REG register.

The prescaler is not readable or writable. All instructions writing to the TMR0 register will clear the prescaler.

21.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The TMR0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The TMR0IF bit can only be cleared in software. The Timer0 interrupt enable is the TMR0IE bit of the INTCON register.

Note:	The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep.

21.1.5 8-BIT COUNTER MODE SYNCHRONIZATION

When in 8-Bit Counter mode, the incrementing edge on the T0CKI pin must be synchronized to the instruction clock. Synchronization can be accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the instruction clock. The high and low periods of the external clocking source must meet the timing requirements as shown in Table 36-12: Timer0 and Timer1 External Clock Requirements.

21.1.6 OPERATION DURING SLEEP

Timer0 cannot operate while the processor is in Sleep mode. The contents of the TMR0 register will remain unchanged while the processor is in Sleep mode.

TABLE 23-5: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCP1CON	EN	—	OUT	FMT	MODE<3:0>				319
CCP2CON	EN	—	OUT	FMT	MODE<3:0>				319
CCP7CON	EN	—	OUT	FMT	MODE<3:0>				319
CCP8CON(1)	EN	—	OUT	FMT	MODE<3:0>				319
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	132
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	133
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	139
T2PR	Timer2 Module Period Register								287*
TMR2	Holding Register for the 8-bit TMR2 Register								287*
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				307
T2CLKCON	—	—	—	—	CS<3:0>				306
T2RST	—	—	—	RSEL<4:0>					309
T2HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					308
T4PR	Timer4 Module Period Register								287*
TMR4	Holding Register for the 8-bit TMR4 Register								287*
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				307
T4CLKCON	—	—	—	—	CS<3:0>				306
T4RST	—	—	—	RSEL<4:0>					309
T4HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					308
T6PR	Timer6 Module Period Register								287*
TMR6	Holding Register for the 8-bit TMR6 Register								287*
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				307
T6CLKCON	—	—	—	—	CS<3:0>				306
T6RST	—	—	—	RSEL<4:0>					309
T6HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					308
T8PR	Timer6 Module Period Register								287*
TMR8	Holding Register for the 8-bit TMR6 Register								287*
T8CON	ON	CKPS<2:0>			OUTPS<3:0>				307
T8CLKCON	—	—	—	—	CS<3:0>				306
T8RST	—	—	—	RSEL<4:0>					309
T8HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					308

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for Timer2 module.

* Page provides register information.

Note 1: PIC16LF1777/9 only.

24.6 CCP/PWM Clock Selection

This device allows each individual CCP and PWM module to select the timer source that controls the module. Each module has an independent selection.

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

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

24.7 Register Definitions: CCP/PWM Timers Control

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

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
C8TSEL<1:0>(1)		C7TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>	
bit 7							bit 0

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

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

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

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

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

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

27.7.1 ASYNCHRONOUS DELAY CHAIN DEAD-BAND DELAY

Asynchronous dead-band delay is determined by the time it takes the input to propagate through a series of delay elements. Each delay element is a nominal five nanoseconds.

For rising event asynchronous dead-band delay set the RDBS bit of the COGxCON0 register and set the COGxDBR register (Register 27-14) value to the desired number of delay elements in the rising event dead-band time.

For falling event asynchronous dead-band delay set the FDBS bit of the COGxCON0 register and set the COGxDBF register (Register 27-15) value to the desired number of delay elements in the falling event dead-band time.

Setting the value to zero disables dead-band delay.

27.7.2 SYNCHRONOUS COUNTER DEAD-BAND DELAY

Synchronous counter dead band is timed by counting COG_clock periods from zero up to the value in the dead-band count register. Use Equation 27-1 to calculate dead-band times.

For rising event synchronous dead-band delay clear the RDBS bit of the COGxCON0 register and set the COGxDBR count register value to the number of COG_clock periods in the rising event dead-band time.

For falling event synchronous dead-band delay clear the FDBS bit of the COGxCON0 register and set the COGxDBF count register value to the number of COG_clock periods in the falling event dead-band time.

When the value is zero, dead-band delay is disabled.

27.7.3 SYNCHRONOUS COUNTER DEAD-BAND TIME UNCERTAINTY

When the rising and falling events that trigger the dead-band counters come from asynchronous inputs, it creates uncertainty in the synchronous counter dead-band time. The maximum uncertainty is equal to one COG_clock period. Refer to Example 27-1 for more detail.

When event input sources are asynchronous with no phase delay, use the Asynchronous Delay Chain Dead-Band mode to avoid the dead-band time uncertainty.

27.7.4 RISING EVENT DEAD BAND

Rising event dead band delays the turn-on of the primary outputs from when complementary outputs are turned off. The rising event dead-band time starts when the rising_ event output goes true.

See **Section 27.7.1 “Asynchronous Delay Chain Dead-Band Delay”** and **Section 27.7.2 “Synchronous Counter Dead-Band Delay”** for more information on setting the rising edge dead-band time.

27.7.5 FALLING EVENT DEAD BAND

Falling event dead band delays the turn-on of complementary outputs from when the primary outputs are turned off. The falling event dead-band time starts when the falling_event output goes true.

See **Section 27.7.1 “Asynchronous Delay Chain Dead-Band Delay”** and **Section 27.7.2 “Synchronous Counter Dead-Band Delay”** for more information on setting the rising edge dead-band time.

27.7.6 DEAD-BAND OVERLAP

There are two cases of potential dead-band overlap:

- Rising-to-falling
- Falling-to-rising

27.7.6.1 Rising-to-Falling Overlap

In this case, the falling event occurs while the rising event dead-band counter is still counting. When this happens, the primary drives are suppressed and the dead band extends by the falling event dead-band time. At the termination of the extended dead-band time, the complementary drive goes true.

27.7.6.2 Falling-to-Rising Overlap

In this case, the rising event occurs while the falling event dead-band counter is still counting. When this happens, the complementary drive is suppressed and the dead band extends by the rising event dead-band time. At the termination of the extended dead-band time, the primary drive goes true.

27.8 Blanking Control

Input blanking is a function whereby the event inputs can be masked or blanked for a short period of time. This is to prevent electrical transients caused by the turn-on/off of power components from generating a false input event.

The COG contains two blanking counters: one triggered by the rising event and the other triggered by the falling_event. The counters are cross coupled with the events they are blanking. The falling event blanking counter is used to blank rising input events and the rising event blanking counter is used to blank falling input events. Once started, blanking extends for the time specified by the corresponding blanking counter.

Blanking is timed by counting COG_clock periods from zero up to the value in the blanking count register. Use Equation 27-1 to calculate blanking times.

REGISTER 27-12: COGxASD1: COG AUTO-SHUTDOWN CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
AS7E	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E
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 7-0 **AS<7:0>E:** Auto-shutdown Source <n> Enable bits(1). See Table 27-6.
1 = COGx is shutdown when source <n> output is low
0 = Source <n> has no effect on shutdown

Note 1: Any combination of <n> bits can be selected.

TABLE 27-6: AUTO-SHUTDOWN SOURCES

Bit <n>	COG1	COG2	COG3(2)	COG3(3)	COG4(2)
7	TMR4_postscaled(1)	TMR4_postscaled(1)	TMR8_postscaled(1)	TMR8_postscaled(1)	TMR8_postscaled(1)
6	TMR2_postscaled(1)	TMR2_postscaled(1)	TMR6_postscaled(1)	TMR6_postscaled(1)	TMR6_postscaled(1)
5	LC2_out	LC2_out	LC4_out	LC4_out	LC4_out
4	sync_CM4_out	sync_CM4_out	sync_CM8_out	sync_CM6_out	sync_CM8_out
3	sync_CM3_out	sync_CM3_out	sync_CM7_out	sync_CM5_out	sync_CM7_out
2	sync_CM2_out	sync_CM2_out	sync_CM6_out	sync_CM2_out	sync_CM6_out
1	sync_CM1_out	sync_CM1_out	sync_CM5_out	sync_CM1_out	sync_CM5_out
0	Pin selected by COG1PPS	Pin selected by COG2PPS	Pin selected by COG3PPS	Pin selected by COG3PPS	Pin selected by COG4PPS

Note 1: Shutdown when source is high.
2: PIC16(L)F1777/9 only.
3: PIC16(L)F1778 only.

31.1 DSM Operation

The DSM module is enabled by setting the EN bit in the MDxCON register. Clearing the EN bit in the MDxCON register disables the DSM module by automatically switching the carrier high and carrier low signals to the VSS signal source. The modulator signal source is also switched to the BIT bit in the MDxCON0 register. This not only assures that the DSM module is inactive, but that it is also consuming the least amount of current.

The values used to select the carrier high, carrier low, and modulator sources held by the Modulation Source, Modulation High Carrier, and Modulation Low Carrier control registers are not affected when the EN bit is cleared and the DSM module is disabled. The values inside these registers remain unchanged while the DSM is inactive. The sources for the carrier high, carrier low and modulator signals will once again be selected when the EN bit is set and the DSM module is enabled and active.

The modulated output signal can be output on any device I/O pin by selecting the desired DSM module in the pin's PPS control register (see Register 12-2). If the output is not directed to any I/O pin then the DSM module will remain active and continue to mix signals, but the output value will not be sent to any pin.

31.2 Modulator Signal Sources

The modulator signal is selected by configuring the MS<4:0> bits of the MDxSRC register. Selections are shown in Table 31-6.

31.3 Carrier Signal Sources

The carrier high signal is selected by configuring the CH<4:0> bits of the MDxCARH register. Selections are shown in Table 31-6.

The carrier low signal is selected by configuring the CL<4:0> bits of the MDxCARL register. Selections are shown in Table 31-7.

31.4 Carrier Synchronization

During the time when the DSM switches between carrier high and carrier low signal sources, the carrier data in the modulated output signal can become truncated. To prevent this, the carrier signal can be synchronized to the modulator signal. When synchronization is enabled, the carrier pulse that is being mixed at the time of the transition is allowed to transition low before the DSM switches over to the next carrier source.

Synchronization is enabled separately for the carrier high and carrier low signal sources. Synchronization for the carrier high signal is enabled by setting the CHSYNC bit of the MDxCON1 register. Synchronization for the carrier low signal is enabled by setting the CLSYNC bit of the MDxCON1 register.

Figure 31-1 through Figure 31-6 show timing diagrams of using various synchronization methods.

32.2 SPI Mode Overview

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

The SPI bus specifies four signal connections:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

FIGURE 32-34: BUS COLLISION DURING START CONDITION (SCL = 0)

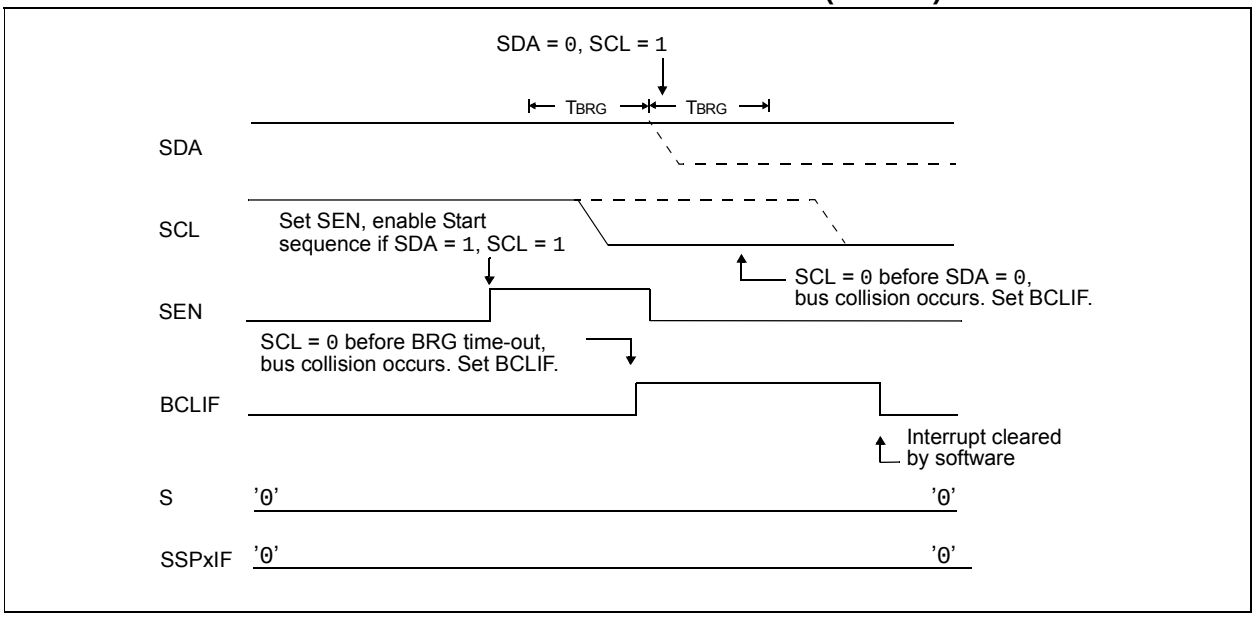


FIGURE 32-35: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION

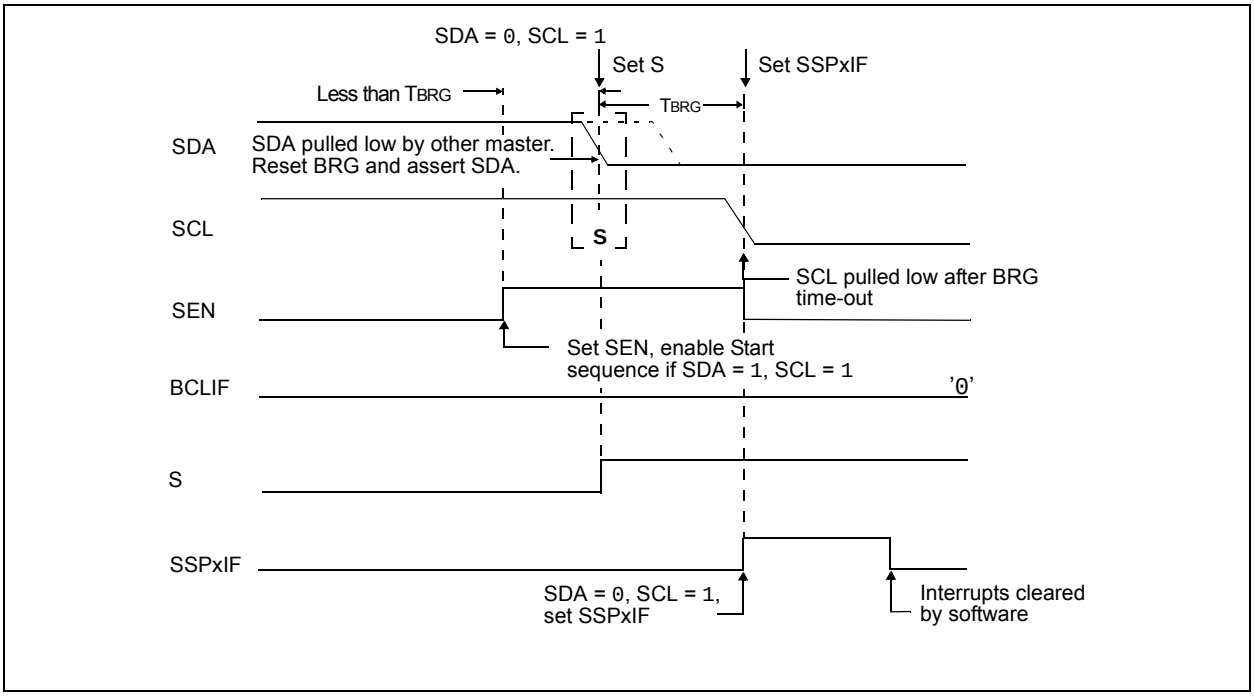


TABLE 33-7: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	177
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	182
ANSELC	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—	187
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	505
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	132
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	133
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	139
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	504
RxyPPS	—	—	RxyPPS<5:0>						205
SP1BRGL	SP1BRG<7:0>								506
SP1BRGH	SP1BRG<15:8>								506
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	176
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	181
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	186
TX1REG	EUSART Transmit Data Register								495*
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	503

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous master transmission.
* Page provides register information.

LSLF **Logical Left Shift**

Syntax: [*label*] LSLF f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f<7>) → C
(f<6:0>) → dest<7:1>
0 → dest<0>

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the left through the Carry flag. A '0' is shifted into the LSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

LSRF **Logical Right Shift**

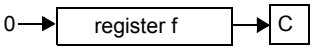
Syntax: [*label*] LSRF f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: 0 → dest<7>
(f<7:1>) → dest<6:0>,
(f<0>) → C,

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the right through the Carry flag. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Description: The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example:

```
MOVF    FSR, 0
```

After Instruction
W = value in FSR register
Z = 1

MOVIW **Move INDFn to W**

Syntax: [*label*] MOVIW ++FSRn
[*label*] MOVIW --FSRn
[*label*] MOVIW FSRn++
[*label*] MOVIW FSRn--
[*label*] MOVIW k[FSRn]

Operands: n ∈ [0,1]
mm ∈ [00,01, 10, 11]
-32 ≤ k ≤ 31

Operation: INDFn → W
Effective address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)
- Unchanged

Status Affected: Z

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

MOVLB **Move literal to BSR**

Syntax: [*label*] MOVLB k

Operands: 0 ≤ k ≤ 31

Operation: k → BSR

Status Affected: None

Description: The 5-bit literal 'k' is loaded into the Bank Select Register (BSR).

MOVLP **Move literal to PCLATH**

Syntax: [*label*] MOVLP k

Operands: 0 ≤ k ≤ 127

Operation: k → PCLATH

Status Affected: None

Description: The 7-bit literal 'k' is loaded into the PCLATH register.

MOVLW **Move literal to W**

Syntax: [*label*] MOVLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W)

Status Affected: None

Description: The 8-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.

Words: 1

Cycles: 1

Example:
```
MOVLW   0x5A
```
After Instruction
W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: 0 ≤ f ≤ 127

Operation: (W) → (f)

Status Affected: None

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example:
```
MOVWF   OPTION_REG
```
Before Instruction
OPTION_REG = 0xFF
W = 0x4F
After Instruction
OPTION_REG = 0x4F
W = 0x4F

TABLE 36-9: PLL CLOCK TIMING SPECIFICATIONS

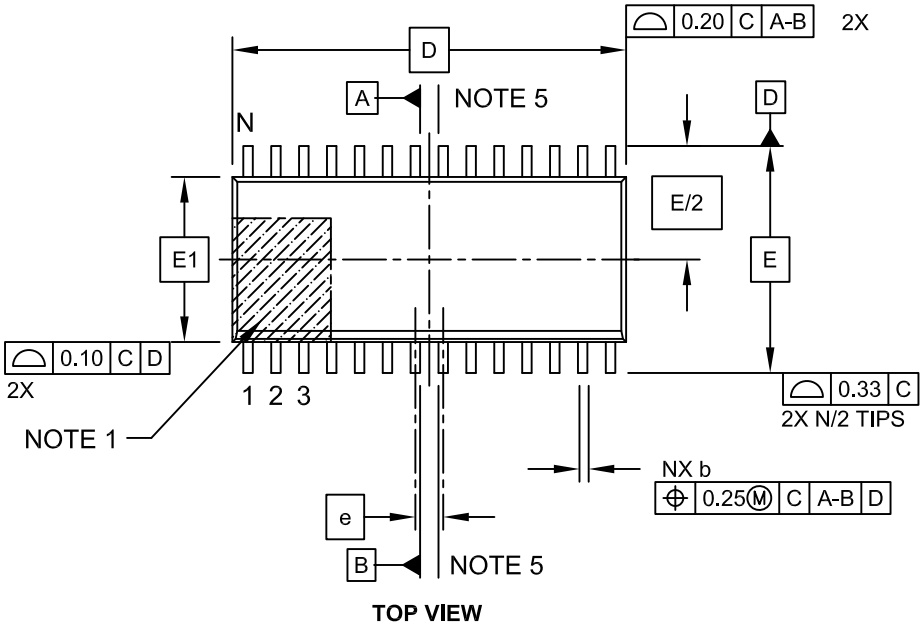
Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
F10	FOSC	Oscillator Frequency Range	4	—	8	MHz	
F11	FSYS	On-Chip VCO System Frequency	16	—	32	MHz	
F12	TRC	PLL Start-up Time (Lock Time)	—	—	2	ms	
F13*	ΔCLK	CLKOUT Stability (Jitter)	-0.25%	—	+0.25%	%	

* These parameters are characterized but not tested.

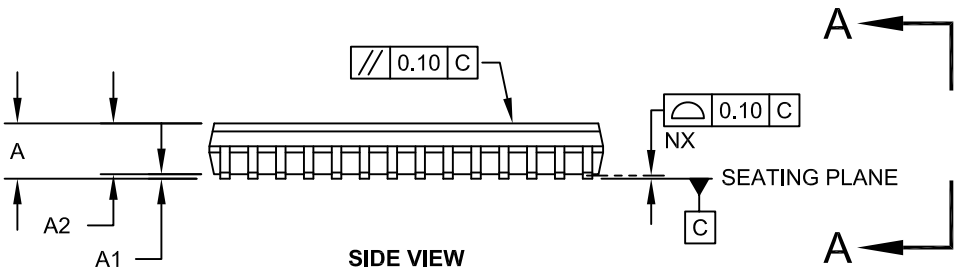
† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

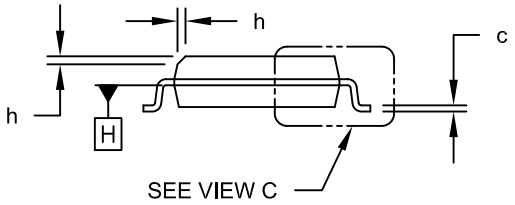
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW



VIEW A-A

Microchip Technology Drawing C04-052C Sheet 1 of 2