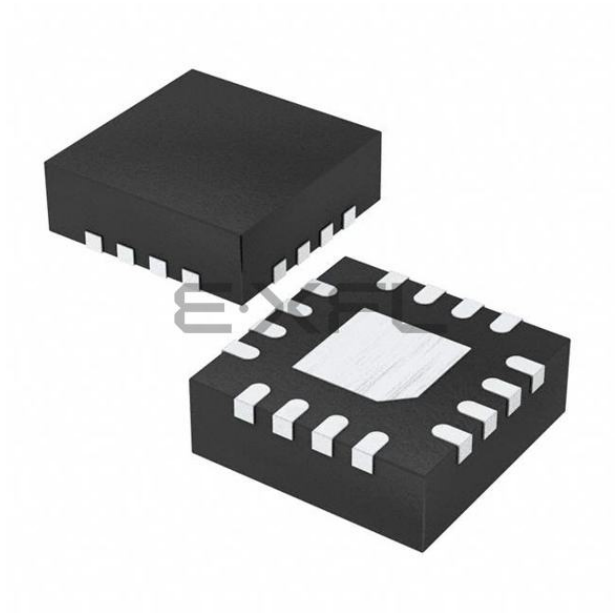


Welcome to E-XFL.COM

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	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 8x10b; D/A 1x5b, 1x10b
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/pic16f1765-e-ml

TABLE 3-10: PIC16(L)F1769 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	OPA2NCHS	593h	DAC1REFH	613h	—	693h	COG1CON0	713h	COG2CON0	793h	—
414h	T4PR	494h	TMR3H	514h	OPA2PCHS	594h	DAC2CON0	614h	—	694h	COG1CON1	714h	COG2CON1	794h	PRG1RTSS
415h	T4CON	495h	T3CON	515h	OPA2CON	595h	DAC2REFL	615h	—	695h	COG1RIS0	715h	COG2RIS0	795h	PRG1FTSS
416h	T4HLT	496h	T3GCON	516h	OPA2ORS	596h	DAC2REFH	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	DAC4CON0	619h	PWM3CON	699h	COG1FIS0	719h	COG2FIS0	799h	PRG1CON2
41Ah	T6TMR	49Ah	TMR5L	51Ah	—	59Ah	DAC4REF	61Ah	PWM4DCL	69Ah	COG1FIS1	71Ah	COG2FIS1	79Ah	PRG2RTSS
41Bh	T6PR	49Bh	TMR5H	51Bh	—	59Bh	—	61Bh	PWM4DCH	69Bh	COG1FSIM0	71Bh	COG2FSIM0	79Bh	PRG2FTSS
41Ch	T6CON	49Ch	T5CON	51Ch	—	59Ch	—	61Ch	PWM4CON	69Ch	COG1FSIM1	71Ch	COG2FSIM1	79Ch	PRG2INS
41Dh	T6HLT	49Dh	T5GCON	51Dh	—	59Dh	—	61Dh	—	69Dh	COG1ASD0	71Dh	COG2ASD0	79Dh	PRG2CON0
41Eh	T6CLKCON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	COG1ASD1	71Eh	COG2ASD1	79Eh	PRG2CON1
41Fh	T6RST	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	COG1STR	71Fh	COG2STR	79Fh	PRG2CON2
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							
								650h	Unimplemented Read as '0'						
470h	Accesses 70h-7Fh	4F0h	Accesses 70h-7Fh	570h	Accesses 70h-7Fh	5F0h	Accesses 70h-7Fh	66Fh		6EFh		76Fh		7EFh	
47Fh		4FFh		57Fh		5FFh		670h	Accesses 70h-7Fh	6F0h	Accesses 70h-7Fh	770h	Accesses 70h-7Fh	7F0h	Accesses 70h-7Fh
								67Fh		6FFh		77Fh		7FFh	

BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
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'.

7.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the interrupt enable bit of the interrupt event is contained in the PIE1 or PIE2 register)

The INTCON, PIR1 and PIR2 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual interrupt enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- Critical registers are automatically saved to the shadow registers (see **"Section 7.5 "Automatic Context Saving"**)
- PC is loaded with the interrupt vector, 0004h

The firmware within the Interrupt Service Routine (ISR) should determine the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its interrupt flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack, restoring the saved context from the shadow registers and setting the GIE bit.

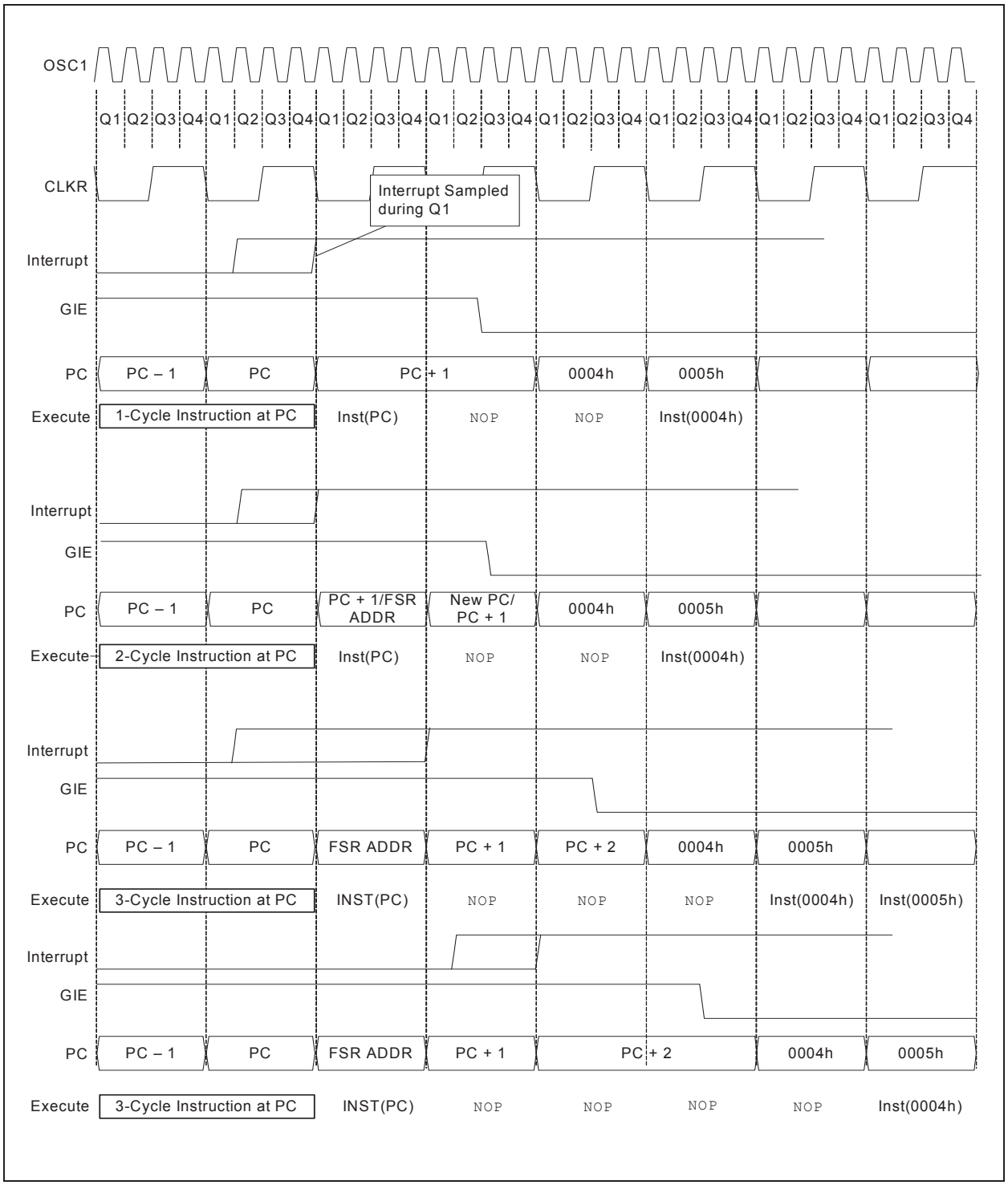
For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1:	Individual interrupt flag bits are set, regardless of the state of any other enable bits.
2:	All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

7.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three or four instruction cycles. For asynchronous interrupts, the latency is three to five instruction cycles, depending on when the interrupt occurs. See Figure 7-2 and Figure 7-3 for more details.

FIGURE 7-2: INTERRUPT LATENCY



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

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
OSFIE	C2IE	C1IE	—	BCL1IE	C4IE(1)	C3IE(1)	CCP2IE(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 **OSFIE:** Oscillator Fail Interrupt Enable bit
- 1 = Enables the Oscillator fail interrupt
- 0 = Disables the Oscillator fail interrupt

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

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

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

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

bit 2 **C4IE:** TMR6 to T6PR Match Interrupt Enable bit(1)
- 1 = Enables the Comparator C4 interrupt
- 0 = Disables the Comparator C4 interrupt

bit 1 **C3IE:** TMR4 to T4PR Match Interrupt Enable bit(1)
- 1 = Enables the Comparator C3 interrupt
- 0 = Disables the Comparator C3 interrupt

bit 0 **CCP2IE:** CCP2 Interrupt Enable bit(1)
- 1 = Enables the CCP2 interrupt
- 0 = Disables the CCP2 interrupt

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

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

8.0 POWER-DOWN MODE (SLEEP)

The Power-Down mode is entered by executing a SLEEP instruction.

Upon entering Sleep mode, the following conditions exist:

1. WDT will be cleared, but keeps running if enabled for operation during Sleep.
2. $\overline{PD}$ bit of the STATUS register is cleared.
3. $\overline{TO}$ bit of the STATUS register is set.
4. CPU clock is disabled.
5. 31 kHz LFINTOSC is unaffected and peripherals that operate from it may continue operation in Sleep.
6. Timer1 and peripherals that operate from Timer1 continue operation in Sleep when the Timer1 clock source selected is:
   - LFINTOSC
   - T1CKI
   - Secondary oscillator
7. ADC is unaffected if the dedicated FRC oscillator is selected.
8. I/O ports maintain the status they had before SLEEP was executed (driving high, low or high-impedance).
9. Resets other than WDT are not affected by Sleep mode.

Refer to individual chapters for more details on peripheral operation during Sleep.

To minimize current consumption, the following conditions should be considered:

- I/O pins should not be floating
- External circuitry sinking current from I/O pins
- Internal circuitry sourcing current from I/O pins
- Current draw from pins with internal weak pull-ups
- Modules using 31 kHz LFINTOSC
- Modules using secondary oscillator

I/O pins that are high-impedance inputs should be pulled to VDD or VSS externally to avoid switching currents caused by floating inputs.

Examples of internal circuitry that might be sourcing current include modules such as the DAC and FVR modules. See **Section 17.0 "5-Bit Digital-to-Analog Converter (DAC) Module"** and **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information on these modules.

8.1 Wake-up from Sleep

The device can wake-up from Sleep through one of the following events:

1. External Reset input on $\overline{MCLR}$ pin, if enabled.
2. Brown-out Reset (BOR), if enabled.
3. Power-on Reset (POR), if enabled.
4. Watchdog Timer, if enabled.
5. Any external interrupt.
6. Interrupts by peripherals capable of running during Sleep (see individual peripheral for more information).

The first three events will cause a device Reset. The last three events are considered a continuation of program execution. To determine whether a device Reset or wake-up event occurred, refer to **Section 6.12 "Determining the Cause of a Reset"**.

When the SLEEP instruction is being executed, the next instruction (PC + 1) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be enabled. Wake-up will occur regardless of the state of the GIE bit. If the GIE bit is disabled, the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is enabled, the device executes the instruction after the SLEEP instruction, the device will then call the Interrupt Service Routine. In cases where the execution of the instruction following SLEEP is not desirable, the user should insert a NOP after the SLEEP instruction.

The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

TABLE 10-1: FLASH MEMORY ORGANIZATION BY DEVICE

Device	Row Erase (words)	Write Latches (words)
PIC16(L)F1764	32	32
PIC16(L)F1765		
PIC16(L)F1768		
PIC16(L)F1769		

10.2.1 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must:

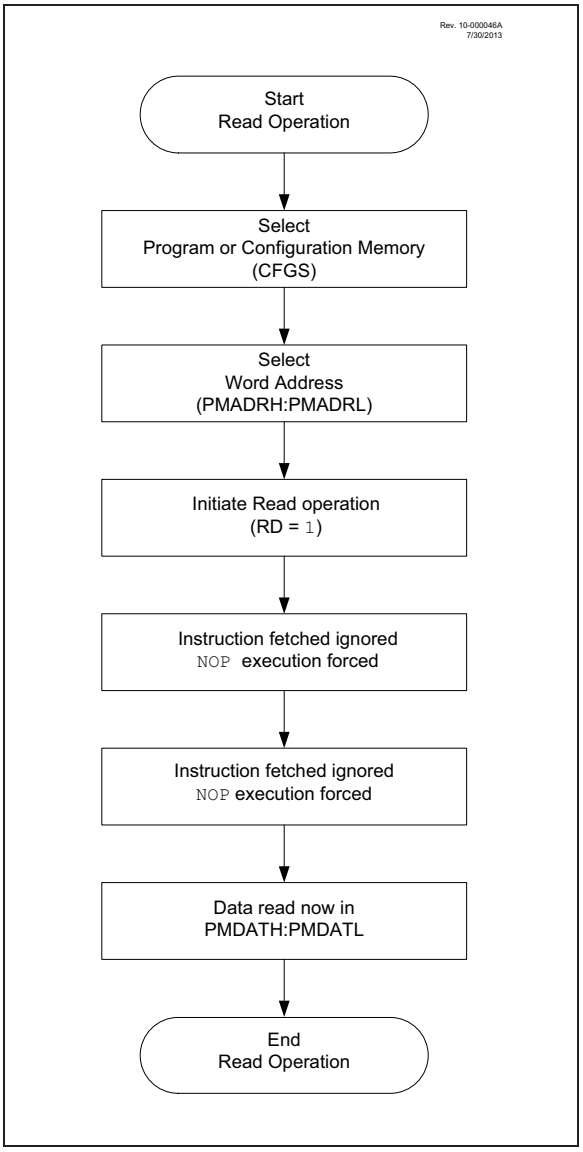
1. Write the desired address to the PMADRH:PMADRL register pair.
2. Clear the CFGS bit of the PMCON1 register.
3. Then, set control bit, RD, of the PMCON1 register.

Once the read control bit is set, the Program Flash Memory controller will use the second instruction cycle to read the data. This causes the second instruction, immediately following the "`BSF PMCON1,RD`" instruction, to be ignored. The data is available in the very next cycle, in the PMDATH:PMDATL register pair; therefore, it can be read as two bytes in the following instructions.

The PMDATH:PMDATL register pair will hold this value until another read or until it is written to by the user.

Note: The two instructions following a program memory read are required to be `NOP`s. This prevents the user from executing a 2-cycle instruction on the next instruction after the RD bit is set.

FIGURE 10-1: FLASH PROGRAM MEMORY READ FLOWCHART



10.6 Register Definitions: Flash Program Memory Control

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

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PMDAT<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	
u = 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 7-0 **PMDAT<7:0>**: Read/Write Value for Least Significant bits of Program Memory bits

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

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	PMDAT<13:8>					
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	
u = 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 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **PMDAT<13:8>**: Read/Write Value for Most Significant bits of Program Memory bits

16.3 Register Definitions: ADC Control

REGISTER 16-1: ADCON0: ADC CONTROL REGISTER 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	R/W-0/0
—	CHS<4:0>					GO/$\overline{\text{DONE}}$	ADON
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 **Unimplemented:** Read as '0'

bit 6-2 **CHS<4:0>:** Analog Channel Select bits

11111 = FVR (Fixed Voltage Reference) Buffer1 Output(2)
11110 = DAC1_output(1)
11101 = Temperature indicator(3)
11100 = DAC2_output(1,5)
11011 = DAC3_output(4)
11010 = DAC4_output(4,5)
11001 = Reserved; no channel connected
•
•
•
01111 = Switched AN7(5,6)
01110 = Switched AN6(6)
01101 = Reserved; no channel connected.
01100 = Reserved; no channel connected.
01011 = AN11(5)
01010 = AN10(5)
01001 = AN9(5)
01000 = AN8(5)
00111 = AN7
00110 = AN6
00101 = AN5
00100 = AN4
00011 = AN3
00010 = AN2
00001 = AN1
00000 = AN0

bit 1 **GO/$\overline{\text{DONE}}$:** ADC Conversion Status bit

1 = ADC conversion cycle in progress; setting this bit starts an ADC conversion cycle
This bit is automatically cleared by hardware when the ADC conversion has completed.
0 = ADC conversion completed/not in progress

bit 0 **ADON:** ADC Enable bit

1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

Note 1: See **Section 17.0 "5-Bit Digital-to-Analog Converter (DAC) Module"** for more information.
2: See **Section 14.0 "Fixed Voltage Reference (FVR)"** for more information.
3: See **Section 15.0 "Temperature Indicator Module"** for more information.
4: See **Section 18.0 "10-Bit Digital-to-Analog Converter (DAC) Module"** for more information.
5: PIC16(L)F1768/9 only.
6: Input source is switched off when op amp override is forced tri-state. See **Section 29.3 "Override Control"**.

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.

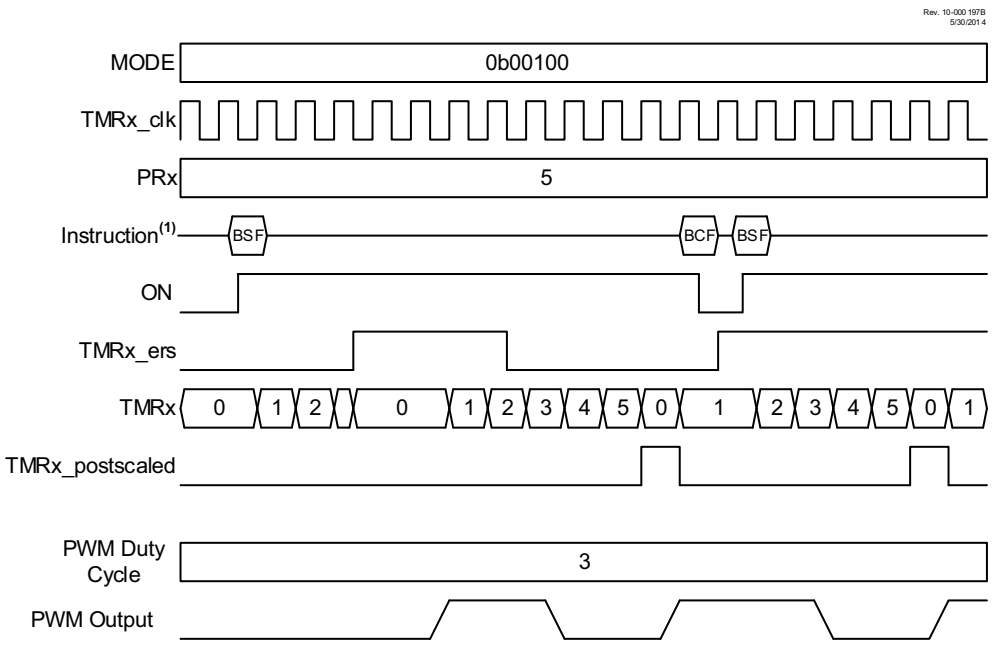
23.6.3 EDGE-TRIGGERED HARDWARE LIMIT MODE

In Hardware Limit mode, the timer can be reset by the TMRx_ers external signal before the timer reaches the period count. Three types of Resets are possible:

- Reset on rising or falling edge (MODE<4:0>= 00011)
- Reset on rising edge (MODE<4:0> = 00100)
- Reset on falling edge (MODE<4:0> = 00101)

When the timer is used in conjunction with the CCP in PWM mode, then an early Reset shortens the period and restarts the PWM pulse after a two-clock delay. Refer to Figure 23-6.

FIGURE 23-6: EDGE-TRIGGERED HARDWARE LIMIT MODE TIMING DIAGRAM (MODE<4:0> = 00100)



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

24.4.4 PWM RESOLUTION

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

The maximum PWM resolution is 10 bits when T2PR/T4PR/T6PR is 255. The resolution is a function of the T2PR/T4PR/T6PR register value as shown by Equation 24-4.

EQUATION 24-4: PWM RESOLUTION

$$Resolution = \frac{log[4(PR2 + 1)]}{log(2)} \ bits$$

Note: If the pulse-width value is greater than the period, the assigned PWM pin(s) will remain unchanged.

TABLE 24-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 20 MHz)

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	16	4	1	1	1	1
T2PR Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6

TABLE 24-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 8 MHz)

PWM Frequency	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	16	4	1	1	1	1
T2PR Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

24.4.5 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 5.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for additional details.

24.4.6 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the CCP registers to their Reset states.

24.4.7 PWM OUTPUT

The output of the CCP in PWM mode is the PWM signal generated by the module and described above. This output is available to the following peripherals:

- ADC trigger
- COG
- PRG
- DSM
- CLC
- Op amp override
- Timer2/4/6 Reset
- Any device pins

REGISTER 26-2: PWMxINTE: PWMx INTERRUPT ENABLE REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	OFIE	PHIE	DCIE	PRIE
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 7-4 **Unimplemented:** Read as '0'

bit 3 **OFIE**: Offset Interrupt Enable bit
1 = Interrupts CPU on offset match
0 = Does not interrupt CPU on offset match

bit 2 **PHIE:** Phase Interrupt Enable bit
1 = Interrupts CPU on phase match
0 = Does not interrupt CPU on phase match

bit 1 **DCIE:** Duty Cycle Interrupt Enable bit
1 = Interrupts CPU on duty cycle match
0 = Does not interrupt CPU on duty cycle match

bit 0 **PRIE:** Period Interrupt Enable bit
1 = Interrupts CPU on period match
0 = Does not interrupt CPU on period match

REGISTER 27-7: COGxFIS0: COGx FALLING EVENT INPUT SELECTION REGISTER 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	R/W-0/0
FIS7	FIS6	FIS5	FIS4	FIS3	FIS2	FIS1	FIS0
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 **FIS7:** COGx Falling Event Input Source 7 Enable bit
1 = PWM3 output is enabled as a falling event input
0 = PWM3 output has no effect on the falling event

bit 6 **FIS6:** COGx Falling Event Input Source 6 Enable bit
1 = CCP2 output is enabled as a falling event input
0 = CCP2 output has no effect on the falling event

bit 5 **FIS5:** COGx Falling Event Input Source 5 Enable bit
1 = CCP1 output is enabled as a falling event input
0 = CCP1 output has no effect on the falling event

bit 4 **FIS4:** COGx Falling Event Input Source 4 Enable bit
1 = Comparator 4 output is enabled as a falling event input
0 = Comparator 4 output has no effect on the falling event

bit 3 **FIS3:** COGx Falling Event Input Source 3 Enable bit
1 = Comparator 3 output is enabled as a falling event input
0 = Comparator 3 output has no effect on the falling event

bit 2 **FIS2:** COGx Falling Event Input Source 2 Enable bit
1 = Comparator 2 output is enabled as a falling event input
0 = Comparator 2 output has no effect on the falling event

bit 1 **FIS1:** COGx Falling Event Input Source 1 Enable bit
1 = Comparator 1 output is enabled as a falling event input
0 = Comparator 1 output has no effect on the falling event

bit 0 **FIS0:** COGx Falling Event Input Source 0 Enable bit
1 = Pin selected with COGxINPPS register is enabled as falling event input
0 = Pin selected with COGxINPPS register has no effect on the falling event

28.1.5 CLCx SETUP STEPS

The following steps should be followed when setting up the CLCx:

- Disable CLCx by clearing the EN bit.
- Select desired inputs using CLCxSEL0 through CLCxSEL3 registers (see Table 28-1).
- Clear any associated ANSELx bits.
- Set all TRISx bits associated with inputs.
- Clear all TRISx bits associated with outputs.
- Enable the chosen inputs through the four gates using CLCxGLS0, CLCxGLS1, CLCxGLS2, and CLCxGLS3 registers.
- Select the gate output polarities with the POLy bits of the CLCxPOL register.
- Select the desired logic function with the MODE<2:0> bits of the CLCxCON register.
- Select the desired polarity of the logic output with the POL bit of the CLCxPOL register. (This step may be combined with the previous gate output polarity step).
- If driving a device pin, set the desired Pin PPS Control register and also clear the TRISx bit corresponding to that output.
- If interrupts are desired, configure the following bits:
  - Set the INTP bit in the CLCxCON register for a rising event.
  - Set the INTN bit in the CLCxCON register for a falling event.
  - Set the CLCxIE bit of the associated PIE registers.
  - Set the GIE and PEIE bits of the INTCON register.
- Enable the CLCx by setting the EN bit of the CLCxCON register.

28.2 CLCx Interrupts

An interrupt will be generated upon a change in the output value of the CLCx when the appropriate interrupt enables are set. A rising edge detector and a falling edge detector are present in each CLC for this purpose.

The CLCxIF bit of the associated PIR registers will be set when either edge detector is triggered and its associated enable bit is set. The INTP enables rising edge interrupts and the INTN bit enables falling edge interrupts. Both are located in the CLCxCON register.

To fully enable the interrupt, set the following bits:

- EN bit of the CLCxCON register
- CLCxIE bit of the associated PIE registers
- INTP bit of the CLCxCON register (for a rising edge detection)
- INTN bit of the CLCxCON register (for a falling edge detection)
- PEIE and GIE bits of the INTCON register

The CLCxIF bit of the associated PIR registers must be cleared in software as part of the interrupt service. If another edge is detected while this flag is being cleared, the flag will still be set at the end of the sequence.

28.3 Output Mirror Copies

Mirror copies of all CLCxCON output bits are contained in the CLCxDATA register. Reading this register reads the outputs of all CLCs simultaneously. This prevents any reading skew introduced by testing or reading the CLCxOUT bits in the individual CLCxCON registers.

28.4 Effects of a Reset

The CLCxCON register is cleared to zero as the result of a Reset. All other selection and gating values remain unchanged.

28.5 Operation During Sleep

The CLCx module operates independently from the system clock and will continue to run during Sleep, provided that the input sources selected remain active.

The HFINTOSC remains active during Sleep when the CLCx module is enabled and the HFINTOSC is selected as an input source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and as a CLCx input source when the CLCx is enabled, the CPU will go Idle during Sleep, but the CLCx will continue to operate and the HFINTOSC will remain active.

This will have a direct effect on the Sleep mode current.

REGISTER 28-11: CLCxDATA: CLCx DATA OUTPUT

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	—	—	—	MLC3OUT	MLC2OUT	MLC1OUT
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 7-3 **Unimplemented:** Read as '0'

bit 2 **MLC3OUT:** Mirror copy of LC3OUT bit

bit 1 **MLC2OUT:** Mirror copy of LC2OUT bit

bit 0 **MLC1OUT:** Mirror copy of LC1OUT bit

TABLE 28-4: SUMMARY OF REGISTERS ASSOCIATED WITH CLCx

Name	Bit7	Bit6	Bit5	Bit4	BIt3	Bit2	Bit1	Bit0	Register on Page
ANSELA	—	—	—	ANSA4	—	ANSA<2:0>			137
ANSELB(1)	ANSB<7:4>				—	—	—	—	143
ANSELC	ANSC<7:6>(1)		—	—	ANSC<3:0>				148
CLCxCON	EN	—	OUT	INTP	INTN	MODE<2:0>			338
CLCDATA	—	—	—	—	—	MLC3OUT	MLC2OUT	MLC1OUT	345
CLCxGLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	341
CLCxGLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	342
CLCxGLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	343
CLCxGLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	344
CLCxPOL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	339
CLCxSEL0	—	—	D1S<5:0>						340
CLCxSEL1	—	—	D2S<5:0>						340
CLCxSEL2	—	—	D3S<5:0>						340
CLCxSEL3	—	—	D4S<5:0>						341
CLCxPPS	—	—	—	CLCxPPS<4:0>					154, 156
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	101
PIE3	PWM6IE(1)	PWM5IE	COG1IE	ZCDIE	COG2IE(1)	CLC3IE	CLC2IE	CLC1IE	104
PIR3	PWM6IF(1)	PWM5IF	COG1IF	ZCDIF	COG2IF(1)	CLC3IF	CLC2IF	CLC1IF	107
RxyPPS	—	—	—	RxyPPS<4:0>					154
TRISA	—	—	TRISA<5:4>		—(3)	TRISA<2:0>			136
TRISB(1)	TRISB<7:4>				—	—	—	—	142
TRISC	TRISC<7:6>(1)		TRISC<5:0>						147

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

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

2: Unimplemented, read as '1'.

FIGURE 32-9: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

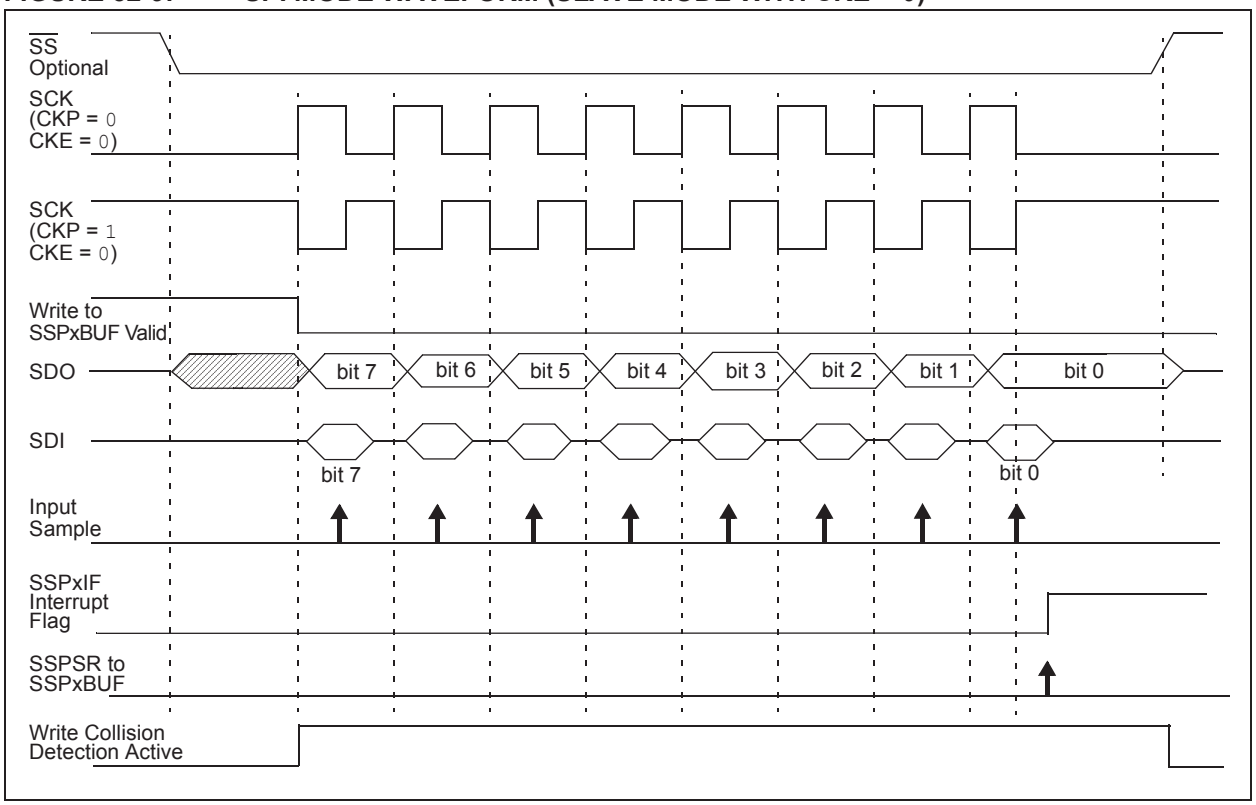


FIGURE 32-10: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)

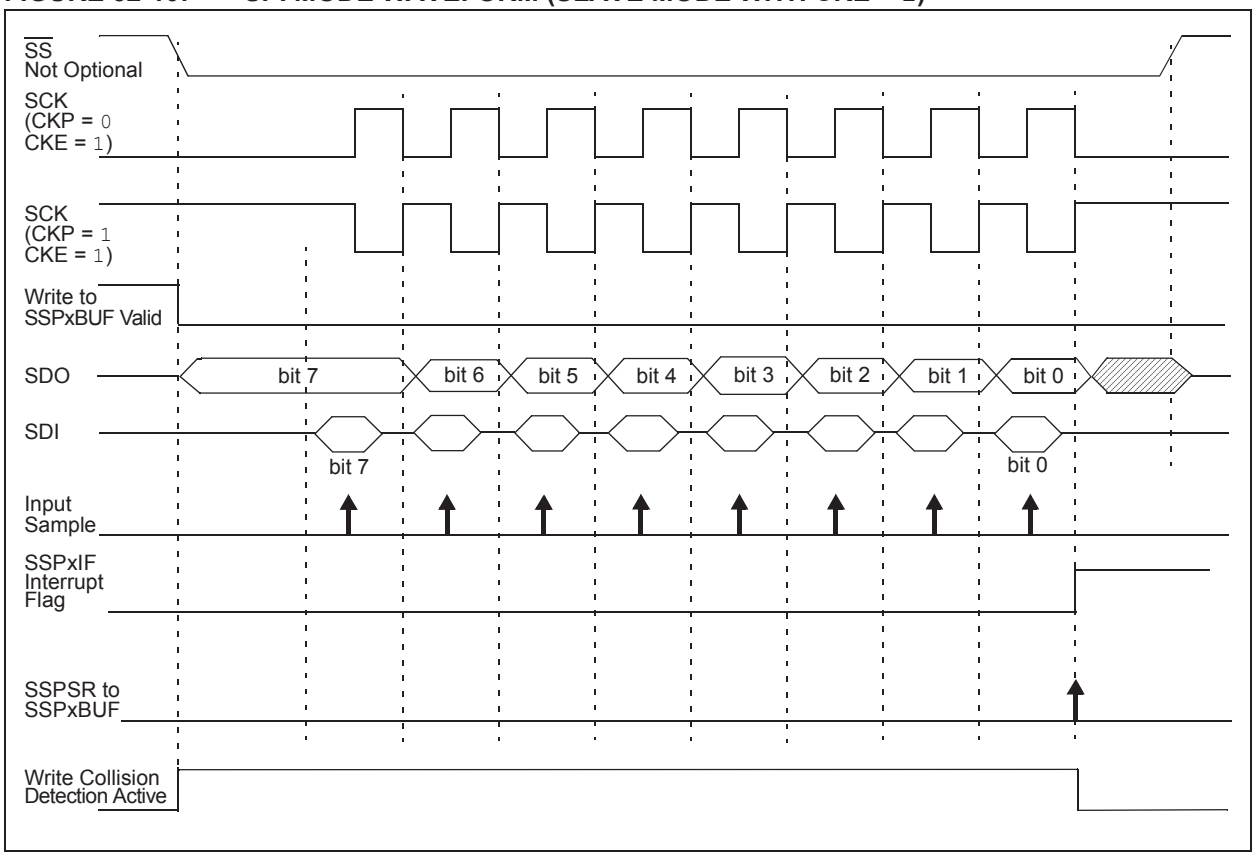


TABLE 32-3: SUMMARY OF REGISTERS ASSOCIATED WITH I^2C OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
ANSELA	—	—	—	ANSA4	—	ANSA<2:0>			137
ANSELB(1)	ANSB<7:4>				—	—	—	—	143
ANSELC	ANSC<7:6>(1)		—	—	ANSC<3:0>				148
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
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	105
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF(1)	C3IF(1)	CCP2IF(1)	106
RxyPPS	—	—	—	RxyPPS<4:0>					154
SSPCLKPPS	—	—	—	SSPCLKPPS<4:0>					154, 156
SSPDATPPS	—	—	—	SSPDATPPS<4:0>					154, 156
SSPSSPPS	—	—	—	SSPSSPPS<4:0>					154, 156
SSP1ADD	ADD<7:0>								430
SSP1BUF	Synchronous Serial Port Receive Buffer/Transmit Register								380*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				426
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	428
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	429
SSP1MSK	MSK<7:0>								430
SSP1STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	424
TRISA	—	—	TRISA<5:4>		—(2)	TRISA<2:0>			136
TRISB(1)	TRISB<7:4>				—	—	—	—	142
TRISC	TRISC<7:6>(1)		TRISC<5:0>						147

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the MSSP module in I^2C mode.

* Page provides register information.

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

2: Unimplemented, read as '1'.

33.6 EUSART Operation During Sleep

The EUSART will remain active during Sleep only in the Synchronous Slave mode. All other modes require the system clock and therefore, cannot generate the necessary signals to run the Transmit or Receive Shift registers during Sleep.

Synchronous Slave mode uses an externally generated clock to run the Transmit and Receive Shift registers.

33.6.1 SYNCHRONOUS RECEIVE DURING SLEEP

To receive during Sleep, all the following conditions must be met before entering Sleep mode:

- RCxSTA and TXxSTA Control registers must be configured for synchronous slave reception (see **Section 33.5.2.4 "Synchronous Slave Reception Setup"**).
- If interrupts are desired, set the RCIE bit of the PIE1 register, and the GIE and PEIE bits of the INTCON register.
- The RCIF interrupt flag must be cleared by reading RCxREG to unload any pending characters in the receive buffer.

Upon entering Sleep mode, the device will be ready to accept data and clocks on the RX/DT and TX/CK pins, respectively. When the data word has been completely clocked in by the external device, the RCIF interrupt flag bit of the PIR1 register will be set; thereby, waking the processor from Sleep.

Upon waking from Sleep, the instruction following the `SLEEP` instruction will be executed. If the Global Interrupt Enable (GIE) bit of the INTCON register is also set, then the Interrupt Service Routine at address, 004h, will be called.

33.6.2 SYNCHRONOUS TRANSMIT DURING SLEEP

To transmit during Sleep, all the following conditions must be met before entering Sleep mode:

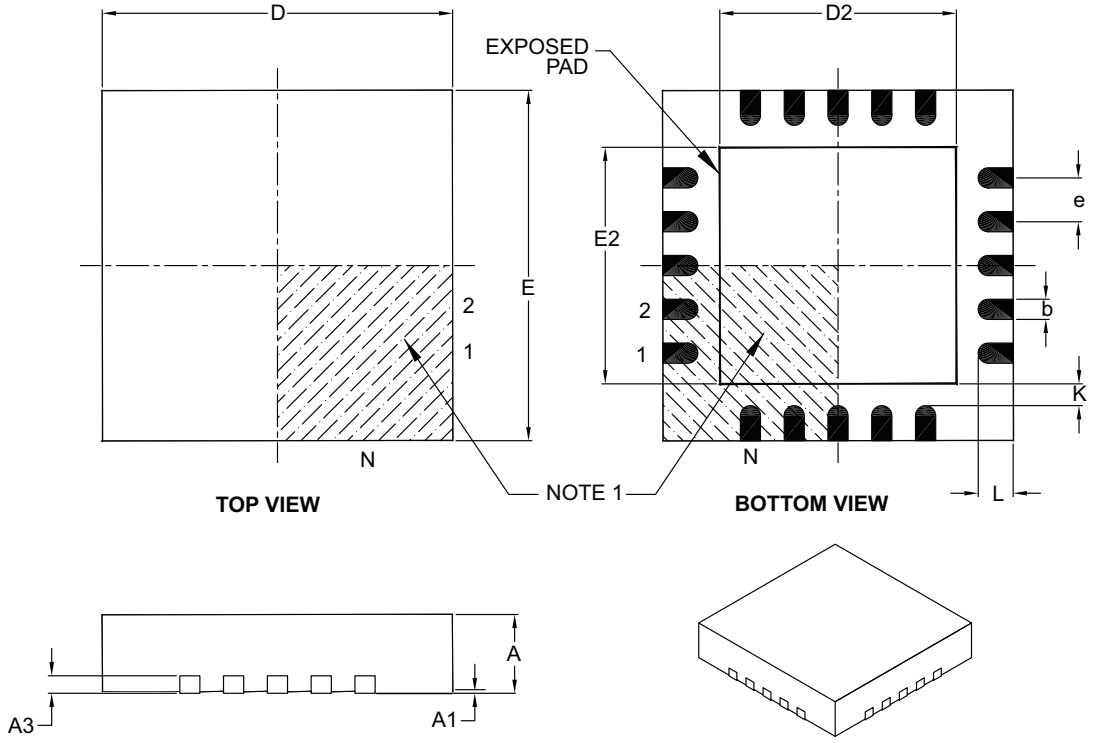
- The RCxSTA and TXxSTA Control registers must be configured for synchronous slave transmission (see **Section 33.5.2.2 "Synchronous Slave Transmission Setup"**).
- The TXIF interrupt flag must be cleared by writing the output data to the TXxREG; thereby, filling the TSR and transmit buffer.
- If interrupts are desired, set the TXIE bit of the PIE1 register and the PEIE bit of the INTCON register.
- Interrupt enable bits, TXIE of the PIE1 register and PEIE of the INTCON register, must set.

Upon entering Sleep mode, the device will be ready to accept clocks on the TX/CK pin and transmit data on the RX/DT pin. When the data word in the TSR has been completely clocked out by the external device, the pending byte in the TXxREG will transfer to the TSR and the TXIF flag will be set; thereby, waking the processor from Sleep. At this point, the TXxREG is available to accept another character for transmission, which will clear the TXIF flag.

Upon waking from Sleep, the instruction following the `SLEEP` instruction will be executed. If the Global Interrupt Enable (GIE) bit is also set, then the Interrupt Service Routine at address, 0004h, will be called.

20-Lead Plastic Quad Flat, No Lead Package (ML) – 4x4x0.9 mm Body [QFN]

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		0.50 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Width	E		4.00 BSC	
Exposed Pad Width	E2	2.60	2.70	2.80
Overall Length	D		4.00 BSC	
Exposed Pad Length	D2	2.60	2.70	2.80
Contact Width	b	0.18	0.25	0.30
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	–	–

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
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 C04-126B

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.

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.

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

Trademarks

ISBN: 978-1-5224-1442-1