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	14-SOIC (0.154", 3.90mm Width)
Supplier Device Package	14-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1765-e-sl

1.0 DEVICE OVERVIEW

The PIC16(L)F1764/5/8/9 are described within this data sheet. See Table 2 for available package configurations.

Figure 1-1 shows a block diagram of the PIC16(L)F1764/5 devices. Figure 1-2 shows a block diagram of the PIC16(L)F1768/9 devices. Table 1-2 and Table 1-3 show the pinout descriptions.

Refer to Table 1-1 for peripherals available per device.

TABLE 1-1: DEVICE PERIPHERAL SUMMARY

Peripheral		PIC16(L)F1764	PIC16(L)F1765	PIC16(L)F1768	PIC16(L)F1769
Analog-to-Digital Converter (ADC)		●	●	●	●
Fixed Voltage Reference (FVR)		●	●	●	●
Zero-Cross Detection (ZCD)		●	●	●	●
Temperature Indicator		●	●	●	●
Complementary Output Generator (COG)					
	COG1	●	●	●	●
	COG2			●	●
Programmable Ramp Generator (PRG)					
	PRG1	●	●	●	●
	PRG2			●	●
10-Bit Digital-to-Analog Converter (DAC)					
	DAC1	●	●	●	●
	DAC2			●	●
5-Bit Digital-to-Analog Converter (DAC)					
	DAC3	●	●	●	●
	DAC4			●	●
Capture/Compare/PWM (CCP/ECCP) Modules					
	CCP1	●	●	●	●
	CCP2			●	●
Comparators					
	C1	●	●	●	●
	C2	●	●	●	●
	C3			●	●
	C4			●	●
Configurable Logic Cell (CLC)					
	CLC1	●	●	●	●
	CLC2	●	●	●	●
	CLC3	●	●	●	●
Data Signal Modulator (DSM)					
	DSM1	●	●	●	●
	DSM2			●	●

TABLE 1-1: DEVICE PERIPHERAL SUMMARY (CONTINUED)

Peripheral		PIC16(L)F1764	PIC16(L)F1765	PIC16(L)F1768	PIC16(L)F1769
Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter (EUSART)					
	EUSART	●	●	●	●
Master Synchronous Serial Ports					
	MSSP	●	●	●	●
Op Amp					
	Op Amp 1	●	●	●	●
	Op Amp 2			●	●
10-Bit Pulse-Width Modulator (PWM)					
	PWM3	●	●	●	●
	PWM4			●	●
16-Bit Pulse-Width Modulator (PWM)					
	PWM5	●	●	●	●
	PWM6			●	●
8-Bit Timers					
	Timer0	●	●	●	●
	Timer2	●	●	●	●
	Timer4	●	●	●	●
	Timer6	●	●	●	●
16-Bit Timers					
	Timer1	●	●	●	●
	Timer3	●	●	●	●
	Timer5	●	●	●	●

TABLE 3-6: PIC16(L)F1769 MEMORY MAP (BANKS 0-7)

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	Core Registers (Table 3-2)	080h	Core Registers (Table 3-2)	100h	Core Registers (Table 3-2)	180h	Core Registers (Table 3-2)	200h	Core Registers (Table 3-2)	280h	Core Registers (Table 3-2)	300h	Core Registers (Table 3-2)	380h	Core Registers (Table 3-2)
00Bh		08Bh		10Bh		18Bh		20Bh		28Bh		30Bh		38Bh	
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	WPUA	28Ch	ODCONA	30Ch	SLRCONA	38Ch	INLVLA
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	ODCONB	30Dh	SLRCONB	38Dh	INLVLB
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	WPUC	28Eh	ODCONC	30Eh	SLRCONC	38Eh	INLVLC
00Fh	—	08Fh	—	10Fh	CMOUT	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	—	090h	—	110h	CM1CON0	190h	—	210h	—	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON1	191h	PMADRL	211h	SSP1BUF	291h	CCPR1L	311h	—	391h	IOCAP
012h	PIR2	092h	PIE2	112h	CM1NSEL	192h	PMADRH	212h	SSP1ADD	292h	CCPR1H	312h	—	392h	IOCAN
013h	PIR3	093h	PIE3	113h	CM1PSEL	193h	PMDATL	213h	SSP1MSK	293h	CCP1CON	313h	—	393h	IOCAF
014h	PIR4	094h	PIE4	114h	CM2CON0	194h	PMDATH	214h	SSP1STAT	294h	CCP1CAP	314h	—	394h	IOCBP
015h	TMR0	095h	OPTION_REG	115h	CM2CON1	195h	PMCON1	215h	SSP1CON1	295h	—	315h	—	395h	IOCBN
016h	TMR1L	096h	PCON	116h	CM2NSEL	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—	396h	IOCBF
017h	TMR1H	097h	WDTCON	117h	CM2PSEL	197h	VREGCON(1)	217h	SSP1CON3	297h	—	317h	—	397h	IOCCP
018h	T1CON	098h	OSCTUNE	118h	CM3CON0	198h	—	218h	—	298h	CCPR2L	318h	—	398h	IOCCN
019h	T1GCON	099h	OSCCON	119h	CM3CON1	199h	RC1REG	219h	—	299h	CCPR2H	319h	—	399h	IOCCF
01Ah	T2TMR	09Ah	OSCSTAT	11Ah	CM3NSEL	19Ah	TX1REG	21Ah	—	29Ah	CCP2CON	31Ah	—	39Ah	—
01Bh	T2PR	09Bh	ADRESL	11Bh	CM3PSEL	19Bh	SP1BRGL	21Bh	—	29Bh	CCP2CAP	31Bh	MD2CON0	39Bh	MD1CON0
01Ch	T2CON	09Ch	ADRESH	11Ch	CM4CON0	19Ch	SP1BRGH	21Ch	—	29Ch	—	31Ch	MD2CON1	39Ch	MD1CON1
01Dh	T2HLT	09Dh	ADCON0	11Dh	CM4CON1	19Dh	RC1STA	21Dh	BORCON	29Dh	—	31Dh	MD2SRC	39Dh	MD1SRC
01Eh	T2CLKCON	09Eh	ADCON1	11Eh	CM4NSEL	19Eh	TX1STA	21Eh	FVRCON	29Eh	CCPTMRS	31Eh	MD2CARL	39Eh	MD1CARL
01Fh	T2RST	09Fh	ADCON2	11Fh	CM4PSEL	19Fh	BAUD1CON	21Fh	ZCD1CON	29Fh	—	31Fh	MD2CARH	39Fh	MD1CARH
020h	General Purpose Register 80 Bytes	0A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	General Purpose Register 80 Bytes	220h	General Purpose Register 80 Bytes	2A0h	General Purpose Register 80 Bytes	320h	General Purpose Register 80 Bytes	3A0h	General Purpose Register 80 Bytes
06Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
070h	Common RAM 70h – 7Fh	0F0h	Accesses 70h-7Fh	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		17Fh		1FFh		27Fh		2FFh		37Fh		3FFh	

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

Note 1: Unimplemented on PIC16LF1769.

3.3.6 CORE FUNCTION REGISTERS SUMMARY

The core function registers listed in Table 3-15 can be addressed from any bank.

TABLE 3-15: CORE FUNCTION REGISTERS SUMMARY(1)

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on All Other Resets
Bank 0-31											
x00h or x80h	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
x01h or x81h	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
x02h or x82h	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
x03h or x83h	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu
x04h or x84h	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
x05h or x85h	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
x06h or x86h	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
x07h or x87h	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
x08h or x88h	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000
x09h or x89h	WREG	Working Register								0000 0000	uuuu uuuu
x0Ah or x8Ah	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
x0Bh or x8Bh	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000

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

Note 1: These registers can be addressed from any bank.

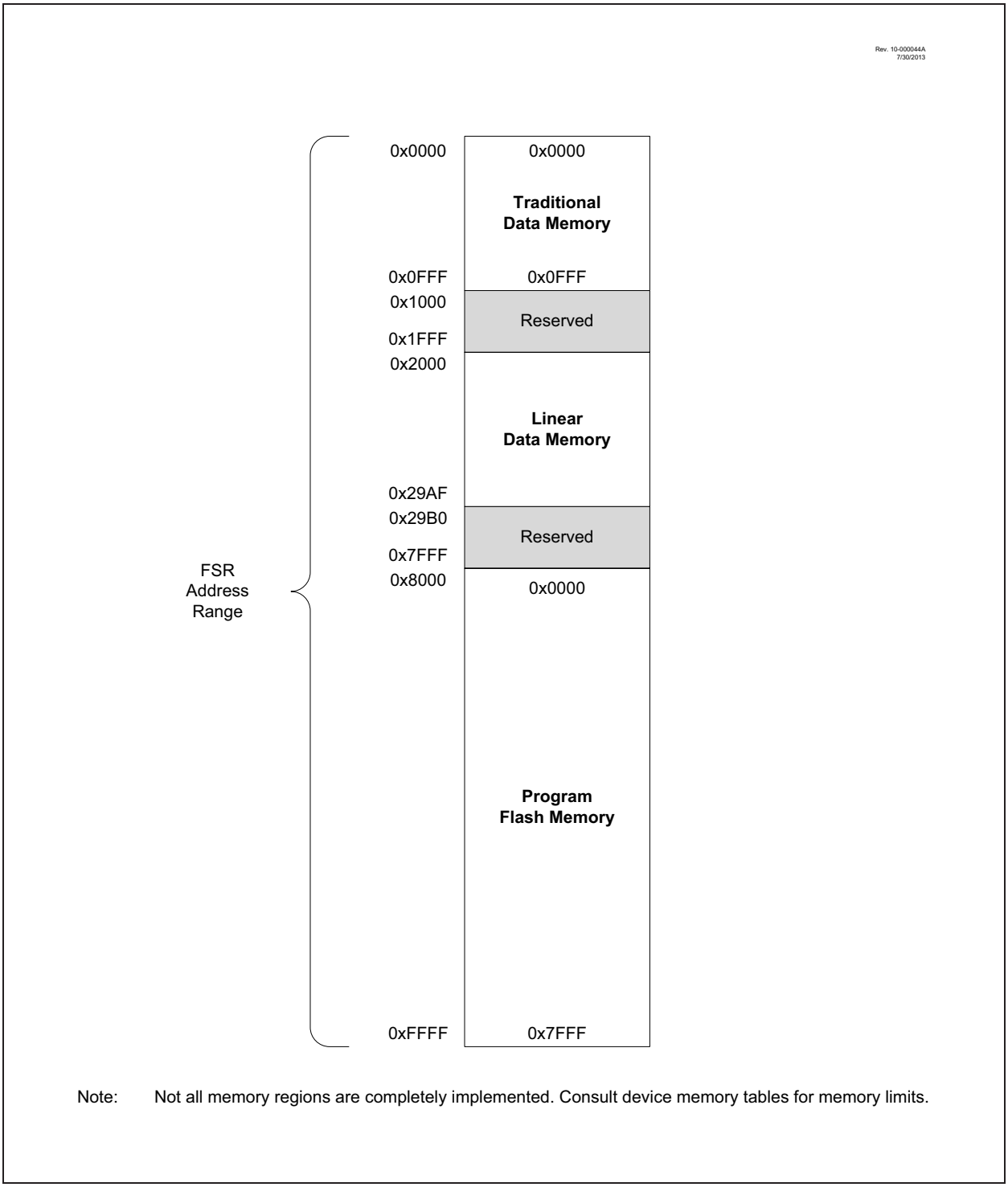
TABLE 3-16: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on All Other Resets
Bank 8											
40Ch — 40Dh	—	Unimplemented								—	—
40Eh	HIDRVC	—	—	HIDC<5:4>		—	—	—	—	--00 ----	--00 ----
40Fh — 412h	—	Unimplemented								—	—
413h	T4TMR	Holding Register for the 8-Bit TMR4 Register								0000 0000	0000 0000
413h	T4PR	TMR4 Period Register								1111 1111	1111 1111
415h	T4CON	ON	CKPS<2:0>			OUTPS<3:0>				0000 0000	0000 0000
416h	T4HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					0000 0000	0000 0000
417h	T4CLKCON	—	—	—	—	CS<3:0>				---- 0000	---- 0000
418h	T4RST	—	—	—	—	RSEL<3:0>				---- 0000	---- 0000
419h	—	Unimplemented								—	—
41Ah	T6TMR	Holding Register for the 8-Bit TMR4 Register								0000 0000	0000 0000
41Bh	T6PR	TMR4 Period Register								1111 1111	1111 1111
41Ch	T6CON	ON	CKPS<2:0>			OUTPS<3:0>				0000 0000	0000 0000
41Dh	T6HLT	PSYNC	CKPOL	CKSYNC	MODE<4:0>					0000 0000	0000 0000
41Eh	T6CLKCON	—	—	—	—	CS<3:0>				---- 0000	---- 0000
41Fh	T6RST	—	—	—	—	RSEL<3:0>				---- 0000	---- 0000
Bank 9											
48Ch to 492h	—	Unimplemented								—	—
493h	TMR3L	Holding Register for the Least Significant Byte of the 16-Bit TMR1 Register								xxxx xxxx	uuuu uuuu
494h	TMR3H	Holding Register for the Most Significant Byte of the 16-Bit TMR1 Register								xxxx xxxx	uuuu uuuu
495h	T3CON	CS<1:0>		CKPS<1:0>		OSCEN	$\overline{\text{SYNC}}$	—	ON	0000 00-0	uuuu uu-u
496h	T3GCON	GE	GPOL	GTM	GSPM	GGO/$\overline{\text{DONE}}$	GVAL	GSS<1:0>		0000 0x00	uuuu uxuu
497h to 499h	—	Unimplemented								—	—
49Ah	TMR5L	Holding Register for the Least Significant Byte of the 16-Bit TMR1 Register								xxxx xxxx	uuuu uuuu
49Bh	TMR5H	Holding Register for the Most Significant Byte of the 16-Bit TMR1 Register								xxxx xxxx	uuuu uuuu
49Ch	T5CON	CS<1:0>		CKPS<1:0>		OSCEN	$\overline{\text{SYNC}}$	—	ON	0000 00-0	uuuu uu-u
49Dh	T5GCON	GE	GPOL	GTM	GSPM	GGO/$\overline{\text{DONE}}$	GVAL	GSS<1:0>		0000 0x00	uuuu uxuu
49Eh to 49Fh	—	Unimplemented								—	—

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

Note 1: Unimplemented, read as '1'.
2: PIC16(L)F1768/9 only.
3: PIC16(L)F1764/5 only.
4: Unimplemented on PIC16LF1764/5/8/9.

FIGURE 3-8: INDIRECT ADDRESSING



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

REGISTER 7-2: PIE1: PERIPHERAL INTERRUPT ENABLE 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
TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE
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 **TMR1GIE:** Timer1 Gate Interrupt Enable bit
1 = Enables the Timer1 gate acquisition interrupt
0 = Disables the Timer1 gate acquisition interrupt

bit 6 **ADIE:** Analog-to-Digital Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

bit 5 **RCIE:** EUSART Receive Interrupt Enable bit
1 = Enables the EUSART receive interrupt
0 = Disables the EUSART receive interrupt

bit 4 **TXIE:** EUSART Transmit Interrupt Enable bit
1 = Enables the EUSART transmit interrupt
0 = Disables the EUSART transmit interrupt

bit 3 **SSP1IE:** Master Synchronous Serial Port (MSSP) Interrupt Enable bit
1 = Enables the MSSP interrupt
0 = Disables the MSSP interrupt

bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1 **TMR2IE:** TMR2 to T2PR Match Interrupt Enable bit
1 = Enables the Timer2 to T2PR match interrupt
0 = Disables the Timer2 to T2PR match interrupt

bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit
1 = Enables the Timer1 overflow interrupt
0 = Disables the Timer1 overflow interrupt

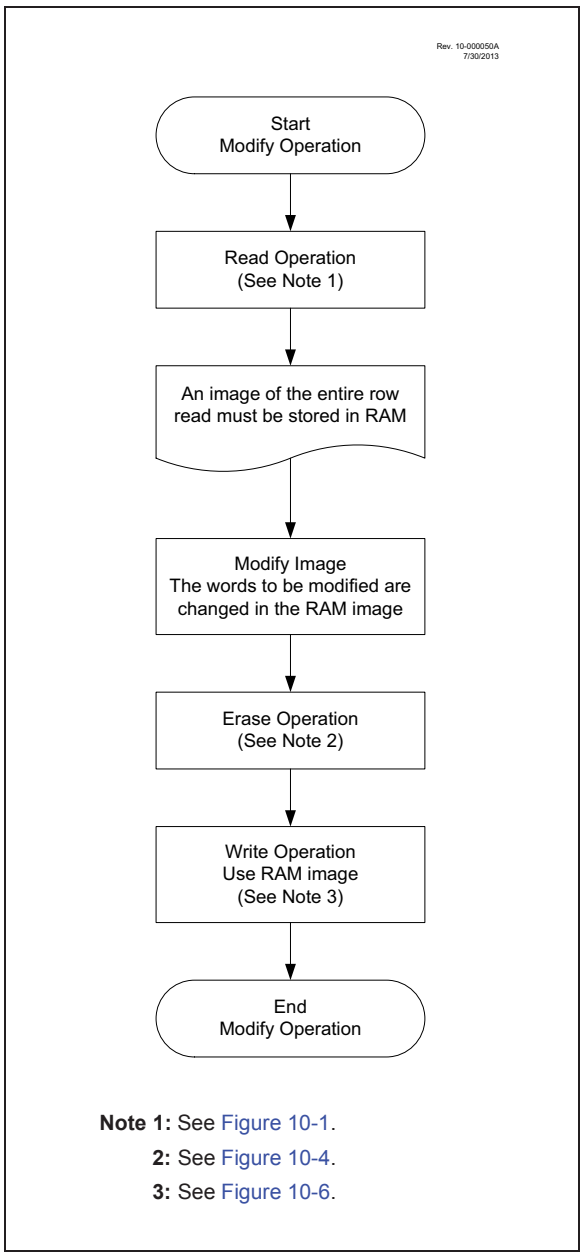
Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

10.3 Modifying Flash Program Memory

When modifying existing data in a program memory row, and data within that row must be preserved, it must first be read and saved in a RAM image. Program memory is modified using the following steps:

1. Load the starting address of the row to be modified.
2. Read the existing data from the row into a RAM image.
3. Modify the RAM image to contain the new data to be written into program memory.
4. Load the starting address of the row to be rewritten.
5. Erase the program memory row.
6. Load the write latches with data from the RAM image.
7. Initiate a programming operation.

FIGURE 10-7: FLASH PROGRAM MEMORY MODIFY FLOWCHART



Note 1: See Figure 10-1.
2: See Figure 10-4.
3: See Figure 10-6.

11.6 Register Definitions: PORTC

REGISTER 11-17: PORTC: PORTC 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
RC<7:0>(1,2)							
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 **RC<7:0>**: PORTC General Purpose I/O Pin bits(1,2)
1 = Port pin is ≥ VIH
0 = Port pin is ≤ VIL

Note 1: Writes to PORTC are actually written to corresponding LATC register. Reads from the PORTC register are the return of actual I/O pin values.
2: RC<7:6> are available on PIC16(L)F1768/9 only.

REGISTER 11-18: TRISC: PORTC TRI-STATE REGISTER

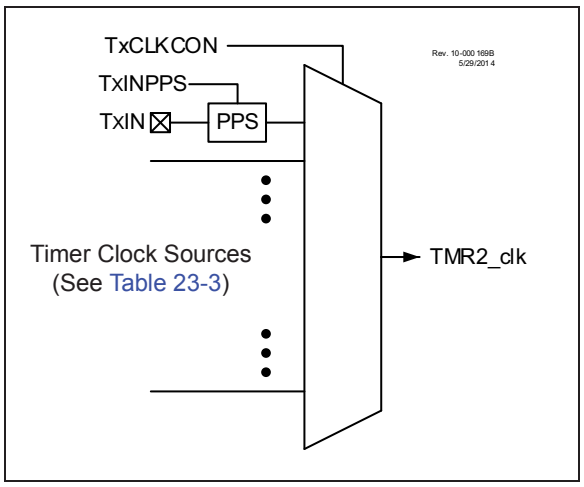
R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISC<7:0>(1)							
bit 7							bit 0

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 7-0 **TRISC<7:0>:** PORTC Tri-State Control bits(1)
1 = PORTC pin is configured as an input (tri-stated)
0 = PORTC pin is configured as an output

Note 1: TRISC<7:6> are available on PIC16(L)F1768/9 only.

FIGURE 23-2: TIMER2 CLOCK SOURCE BLOCK DIAGRAM



23.1 Timer2 Operation

Timer2 operates in three major modes:

- Free-Running Period
- One-Shot
- Monostable

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

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

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

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

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

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

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

Note: TMR2 is not cleared when T2CON is written.

23.1.1 FREE-RUNNING PERIOD MODE

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

23.1.2 ONE-SHOT MODE

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

23.1.3 MONOSTABLE MODE

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

23.2 Timer2 Output

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

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

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

24.3.1 STANDARD PWM OPERATION

The standard PWM function described in this section is available and identical for all CCP modules.

The standard PWM mode generates a Pulse-Width Modulation (PWM) signal on the CCPx pin with up to 10 bits of resolution. The period, duty cycle and resolution are controlled by the following registers:

- T2PR/T4PR/T6PR registers
- T2CON/T4CON/T6CON registers
- CCPRxH:CCPRxL register pair

Figure 24-3 shows a simplified block diagram of PWM operation.

Note 1:	The corresponding TRISx bit must be cleared to enable the PWM output on the CCPx pin.
2:	Clearing the CCPxCON register will relinquish control of the CCPx pin.

24.3.2 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for standard PWM operation:

1. Disable the CCPx pin output driver by setting the associated TRISx bit.
2. Select the timer associated with the PWM by setting the CCPTMRS register.
3. Load the associated T2PR/T4PR/T6PR register with the PWM period value.
4. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
5. Load the CCPRxH:CCPRxL register pair with the PWM duty cycle value.
6. Configure and start the timer selected in Step 2:
   - Clear the timer interrupt flag bit of the PIRx register. See Note below.
   - Configure the CKPSx bits of the TxCON register with the timer prescale value.
   - Enable the timer by setting the ON bit of the TxCON register.
7. Enable PWM output pin:
   - Wait until the timer overflows and the timer interrupt bit of the PIRx register is set. See Note below.
   - Enable the CCPx pin output driver by clearing the associated TRISx bit.

Note:	In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then Step 6 may be ignored.

26.3 Offset Modes

The Offset modes provide the means to adjust the waveform of a slave PWM module relative to the waveform of a master PWM module in the same device.

26.3.1 INDEPENDENT RUN MODE

In Independent Run mode (OFM<1:0> = 00), the PWM module is unaffected by the other PWM modules in the device. The PWMxTMR associated with the PWM module in this mode starts counting as soon as the EN bit associated with this PWM module is set, and continues counting until the EN bit is cleared. Period events reset the PWMxTMR to zero, after which, the timer continues to count.

A detailed timing diagram of this mode used with Standard PWM mode is shown in Figure 26-8.

26.3.2 SLAVE RUN MODE WITH SYNC START

In Slave Run mode with Sync Start (OFM<1:0> = 01), the slave PWMxTMR waits for the master's OF_match event. When this event occurs, if the EN bit is set, the PWMxTMR begins counting and continues to count until software clears the EN bit. Slave period events reset the PWMxTMR to zero, after which, the timer continues to count.

A detailed timing diagram of this mode used with Standard PWM mode is shown in Figure 26-9.

26.3.3 ONE-SHOT SLAVE MODE WITH SYNC START

In One-Shot Slave mode with Synchronous Start (OFM<1:0> = 10), the slave PWMxTMR waits until the master's OF_match event. The timer then begins counting, starting from the value that is already in the timer, and continues to count until the period match event. When the period event occurs, the timer resets to zero and stops counting. The timer then waits until the next master OF_match event, after which, it begins counting again to repeat the cycle. An OF_match event that occurs before the slave PWM has completed the triggered period will be ignored. A slave period that is greater than the master period, but less that twice the master period, will result in a slave output every other master period.

Note:	During the time the slave timers are resetting to zero, if another offset match event is received, it is possible that the slave PWM would not recognize this match event and the slave timers would fail to begin counting again. This would result in missing duty cycles in the output of the slave PWM. To prevent this from happening, avoid using the same period for both the master and slave PWMs.

A detailed timing diagram of this mode used with Standard PWM mode is shown in Figure 26-10.

26.3.4 CONTINUOUS RUN SLAVE MODE WITH SYNC START AND TIMER RESET

In Continuous Run Slave mode with Synchronous Start and Timer Reset (OFM<1:0> = 11), the slave PWMxTMR is inhibited from counting after the slave PWM enable is set. The first master OF_match event starts the slave PWMxTMR. Subsequent master OF_match events reset the slave PWMxTMR timer value back to 1, after which, the slave PWMxTMR continues to count. The next master OF_match event resets the slave PWMxTMR back to 1 to repeat the cycle. Slave period events that occur before the master's OF_match event will reset the slave PWMxTMR to zero, after which, the timer will continue to count. Slaves operating in this mode must have a PWMxPH register pair value equal to, or greater than, 1; otherwise, the phase match event will not occur precluding the start of the PWM output duty cycle.

The offset timing will persist if both the master and slave PWMxPR values are the same and the Slave Offset mode is changed to Independent Run mode while the PWM module is operating.

A detailed timing diagram of this mode used in Standard PWM mode is shown in Figure 26-11.

Note:	Unexpected results will occur if the slave PWM_clock is a higher frequency than the master PWM_clock.

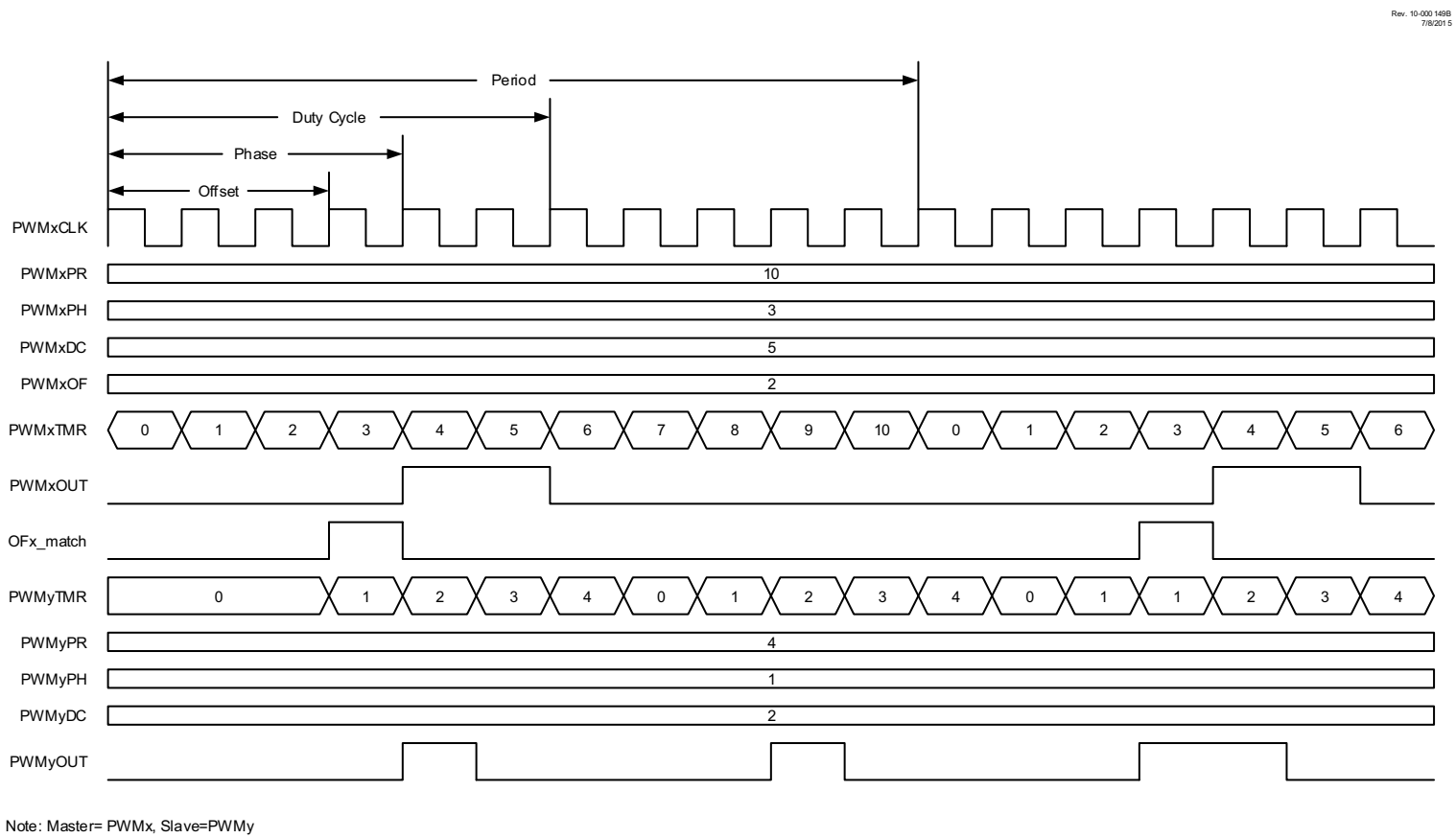
26.3.5 OFFSET MATCH IN CENTER-ALIGNED MODE

When a master is operating in Center-Aligned mode, the offset match event depends on which direction the PWMxTMR is counting. Clearing the OFO bit of the PWMxOFCON register will cause the OF_match event to occur when the timer is counting up. Setting the OFO bit of the PWMxOFCON register will cause the OF_match event to occur when the timer is counting down. The OFO bit is ignored in Non-Center-Aligned modes.

The OFO bit is double-buffered and requires setting the LDA bit to take effect when the PWM module is operating.

Detailed timing diagrams of Center-Aligned mode using offset match control in Independent Slave with Sync Start mode can be seen in Figure 26-12 and Figure 26-13.

FIGURE 26-11: CONTINUOUS SLAVE RUN MODE WITH IMMEDIATE RESET AND SYNC START TIMING DIAGRAM



Note: Master= PWMx, Slave=PWMy

FIGURE 27-6: SIMPLIFIED COG BLOCK DIAGRAM (PUSH-PULL MODE, MD<2:0> = 5)

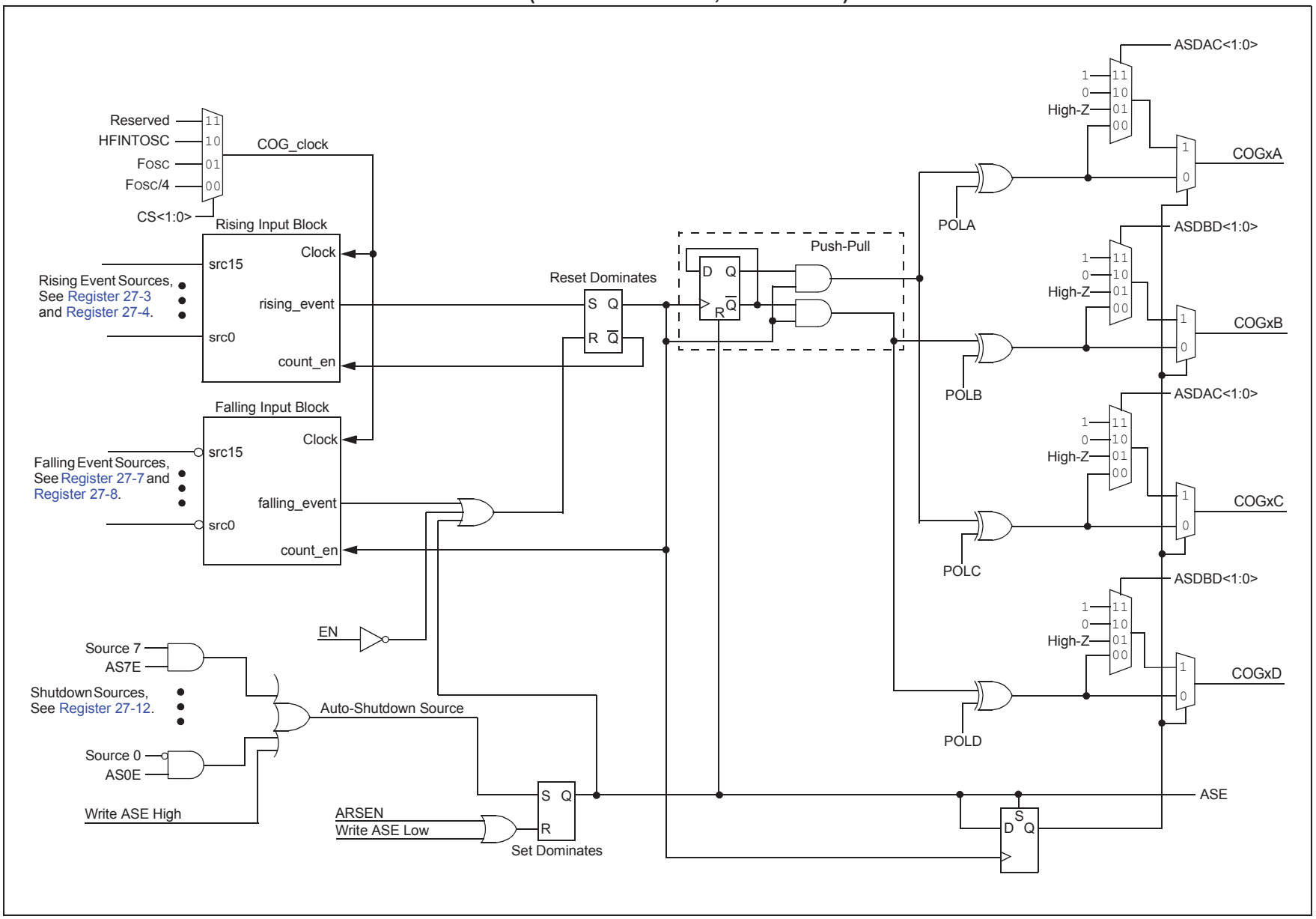
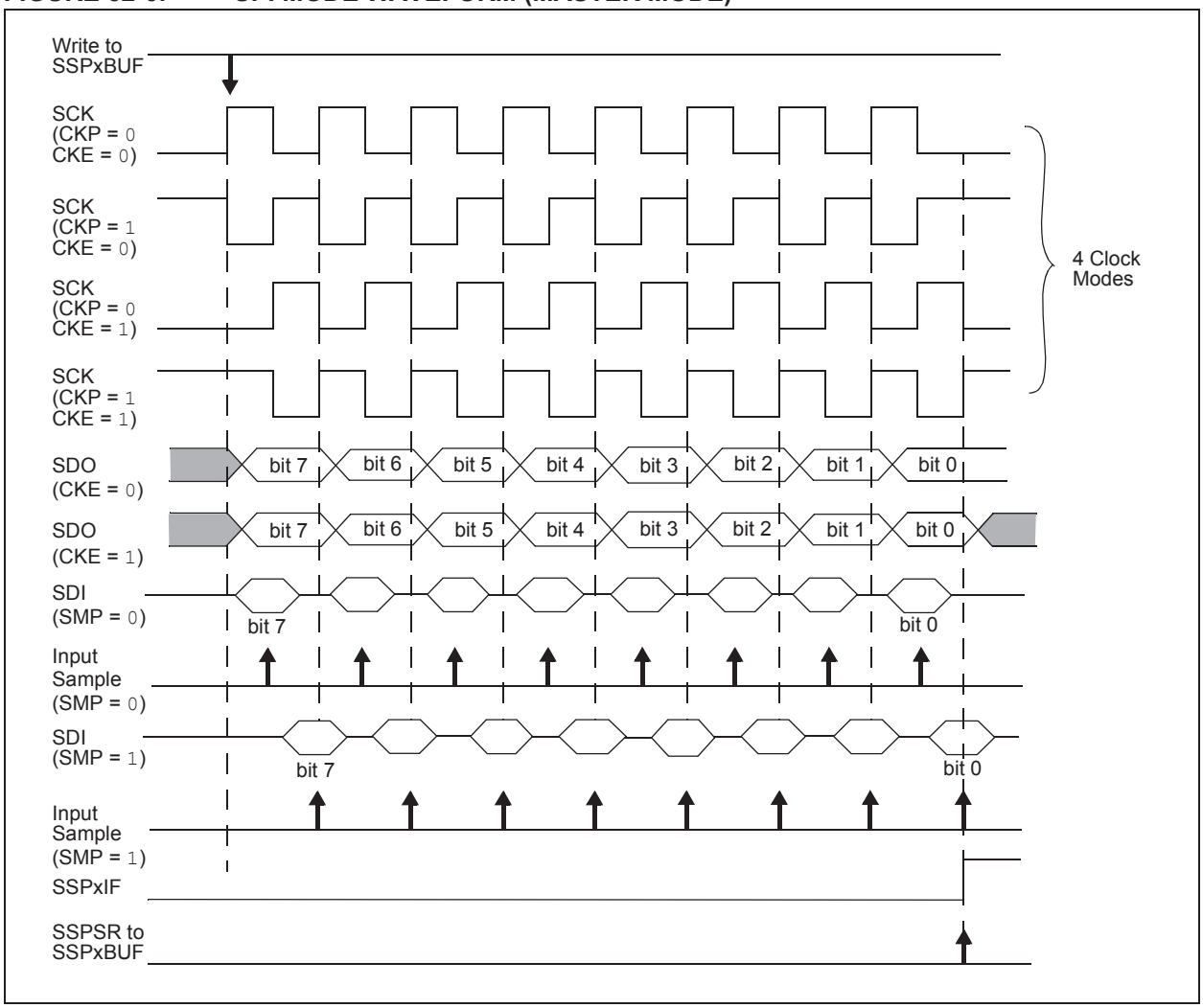


FIGURE 32-6: SPI MODE WAVEFORM (MASTER MODE)



32.2.4 SPI SLAVE MODE

In Slave mode, the data is transmitted and received as external clock pulses appear on SCK. When the last bit is latched, the SSPxIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit of the SSPxCON1 register.

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. The shift register is clocked from the SCK pin input and when a byte is received, the device will generate an interrupt. If enabled, the device will wake-up from Sleep.

32.2.4.1 Daisy-Chain Configuration

The SPI bus can sometimes be connected in a daisy-chain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole chain acts as one large communication shift register. The daisy-chain feature only requires a single Slave Select line from the master device.

Figure 32-7 shows the block diagram of a typical daisy-chain connection when operating in SPI mode.

In a daisy-chain configuration, only the most recent byte on the bus is required by the slave. Setting the BOEN bit of the SSPxCON3 register will enable writes to the SSPxBUF register, even if the previous byte has not been read. This allows the software to ignore data that may not apply to it.

32.4.5 START CONDITION

The I^2C specification defines a Start condition as a transition of SDA from a high to a low state while the SCL line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an active state. Figure 32-12 shows wave forms for Start and Stop conditions.

A bus collision can occur on a Start condition if the module samples the SDA line low before asserting it low. This does not conform to the I^2C specification that states no bus collision can occur on a Start.

32.4.6 STOP CONDITION

A Stop condition is a transition of the SDA line from low-to-high state while the SCL line is high.

Note: At least one SCL low time must appear before a Stop is valid, therefore, if the SDA line goes low, then high again while the SCL line stays high, only the Start condition is detected.

32.4.7 RESTART CONDITION

A Restart is valid any time that a Stop would be valid. A master can issue a Restart if it wishes to hold the bus after terminating the current transfer. A Restart has the same effect on the slave that a Start would, resetting all slave logic and preparing it to clock in an address. The master may want to address the same or another slave. Figure 32-13 shows the wave form for a Restart condition.

In 10-Bit Addressing Slave mode, a Restart is required for the master to clock data out of the addressed slave. Once a slave has been fully addressed, matching both high and low address bytes, the master can issue a Restart and the high address byte with the R/$\overline{W}$ bit set. The slave logic will then hold the clock and prepare to clock out data.

After a full match with R/$\overline{W}$ clear in 10-bit mode, a prior match flag is set and maintained until a Stop condition, a high address with R/$\overline{W}$ clear or high address match fails.

32.4.8 START/STOP CONDITION INTERRUPT MASKING

The SCIE and PCIE bits of the SSPxCON3 register can enable the generation of an interrupt in Slave modes that do not typically support this function. In Slave modes where interrupt on Start and Stop detect is already enabled, these bits will have no effect.

FIGURE 32-12: I^2C START AND STOP CONDITIONS

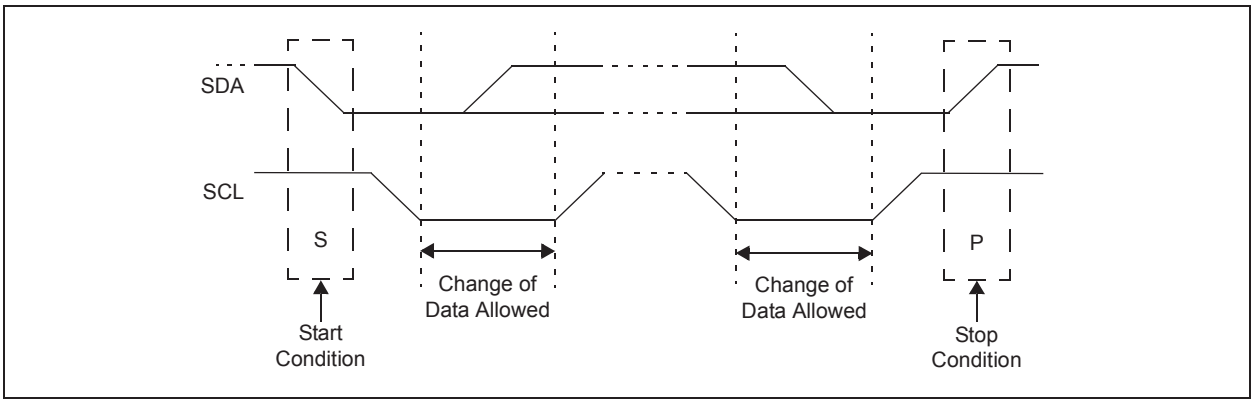


FIGURE 32-13: I^2C RESTART CONDITION

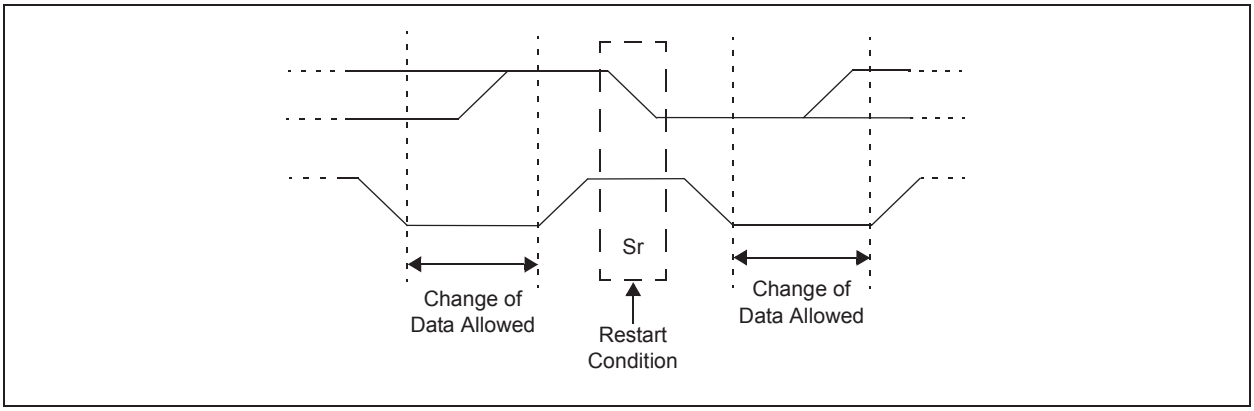
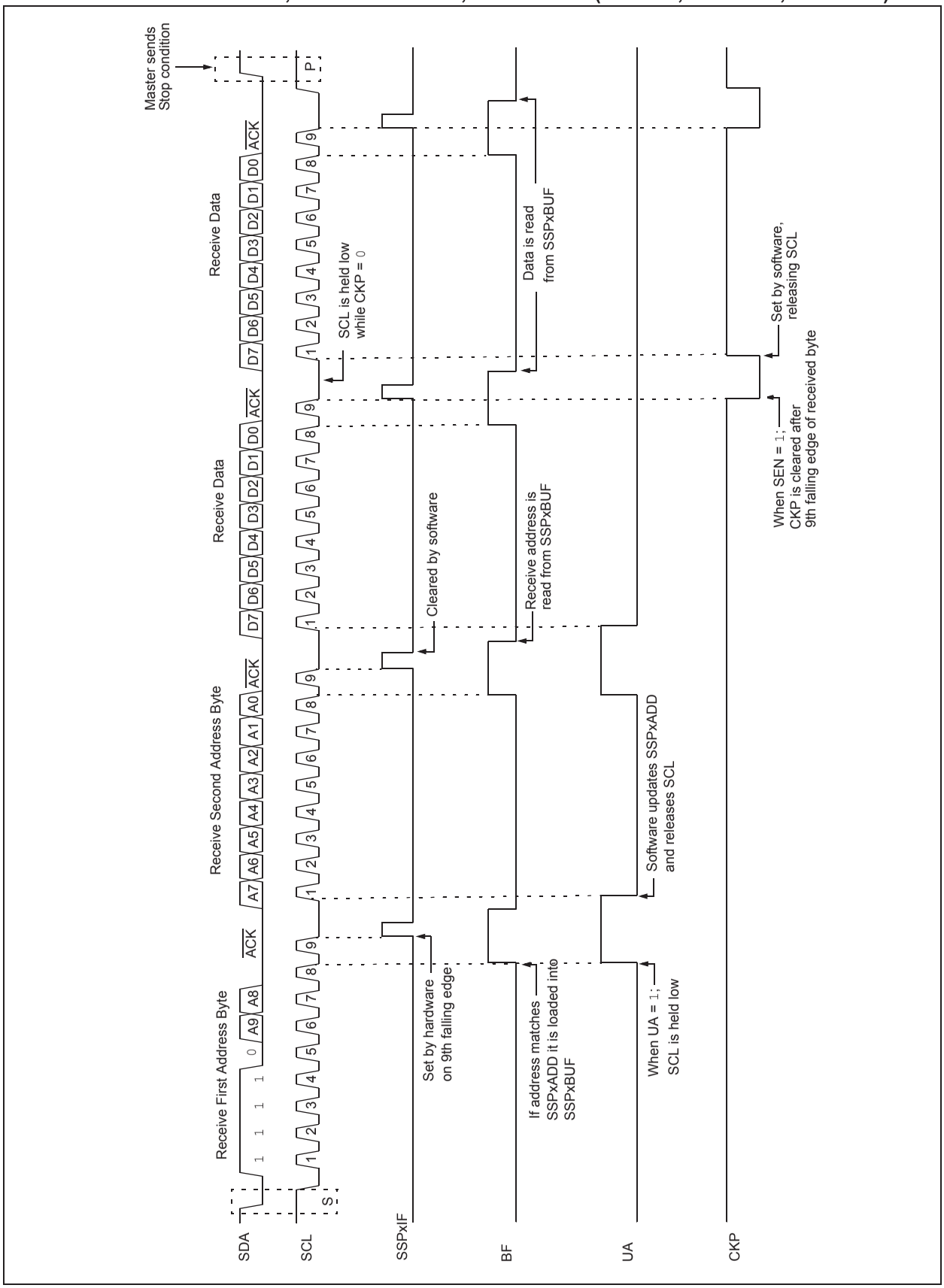


FIGURE 32-20: I²C SLAVE, 10-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 0, DHEN = 0)



35.0 INSTRUCTION SET SUMMARY

Each instruction is a 14-bit word containing the operation code (opcode) and all required operands. The opcodes are broken into three broad categories.

- Byte Oriented
- Bit Oriented
- Literal and Control

The literal and control category contains the most varied instruction word format.

Table 35-3 lists the instructions recognized by the MPASM™ assembler.

All instructions are executed within a single instruction cycle, with the following exceptions, which may take two or three cycles:

- Subroutine takes two cycles (CALL, CALLW)
- Returns from interrupts or subroutines take two cycles (RETURN, RETLW, RETFIE)
- Program branching takes two cycles (GOTO, BRA, BRW, BTFSS, BTFSC, DECFSZ, INCSFZ)
- One additional instruction cycle will be used when any instruction references an indirect file register and the file select register is pointing to program memory.

One instruction cycle consists of four oscillator cycles; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution rate of 1 MHz.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

35.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified and the result is stored according to either the instruction or the destination designator, 'd'. A read operation is performed on a register even if the instruction writes to that register.

TABLE 35-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F).
W	Working register (accumulator).
b	Bit address within an 8-bit file register.
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
n	FSR or INDF number (0-1).
mm	Pre/Post-Increment/Decrement mode selection.

TABLE 35-2: ABBREVIATION DESCRIPTIONS

Field	Description
PC	Program Counter
$\overline{TO}$	Time-out bit
C	Carry bit
DC	Digit Carry bit
Z	Zero bit
$\overline{PD}$	Power-Down bit

BCF	Bit Clear f
Syntax:	[*label*] BCF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	0 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is cleared.

BRA	Relative Branch
Syntax:	[*label*] BRA label [*label*] BRA $+k
Operands:	-256 ≤ label - PC + 1 ≤ 255 -256 ≤ k ≤ 255
Operation:	(PC) + 1 + k → PC
Status Affected:	None
Description:	Add the signed 9-bit literal 'k' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + k. This instruction is a 2-cycle instruction. This branch has a limited range.

BRW	Relative Branch with W
Syntax:	[*label*] BRW
Operands:	None
Operation:	(PC) + (W) → PC
Status Affected:	None
Description:	Add the contents of W (unsigned) to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + (W). This instruction is a 2-cycle instruction.

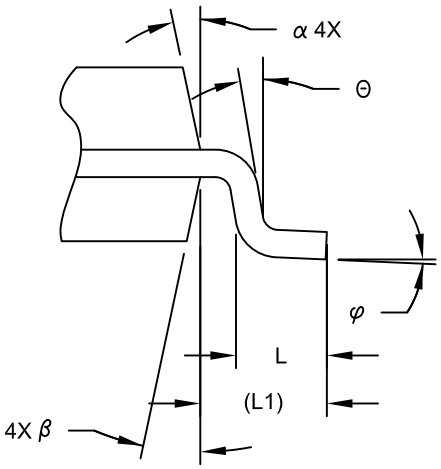
BSF	Bit Set f
Syntax:	[*label*] BSF f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	1 → (f<b>)
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

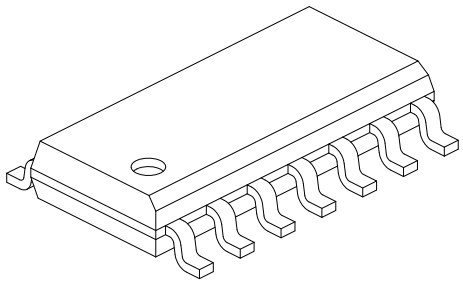
BTFSC	Bit Test f, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b ≤ 7
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction.

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', the next instruction is executed. If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction.

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

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





VIEW C

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

Notes:

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

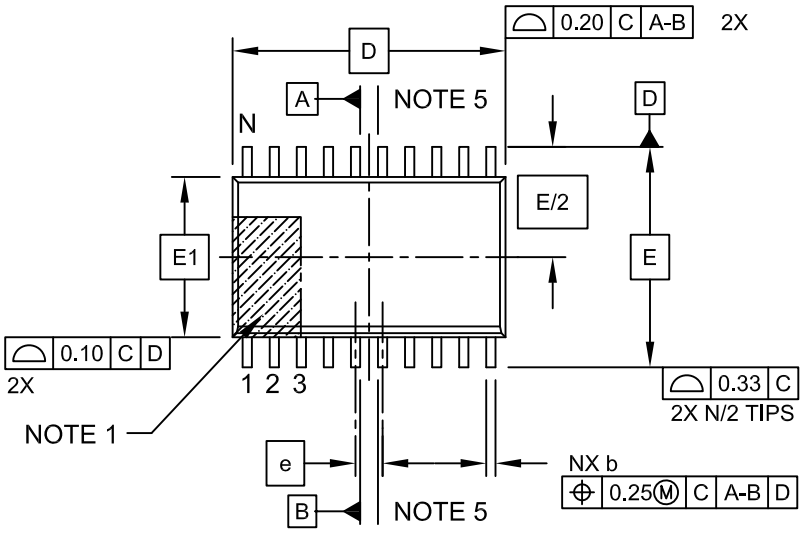
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

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

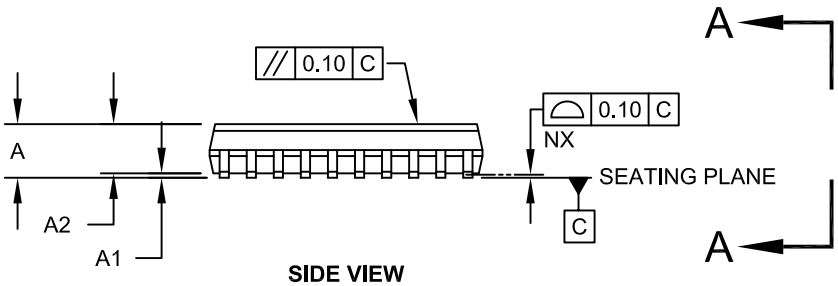
Microchip Technology Drawing No. C04-065C Sheet 2 of 2

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

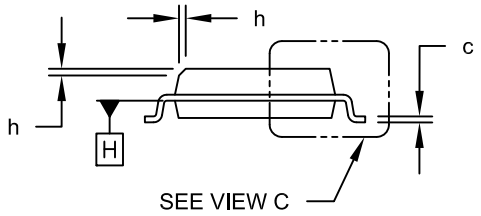
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW



VIEW A-A

Microchip Technology Drawing C04-094C Sheet 1 of 2