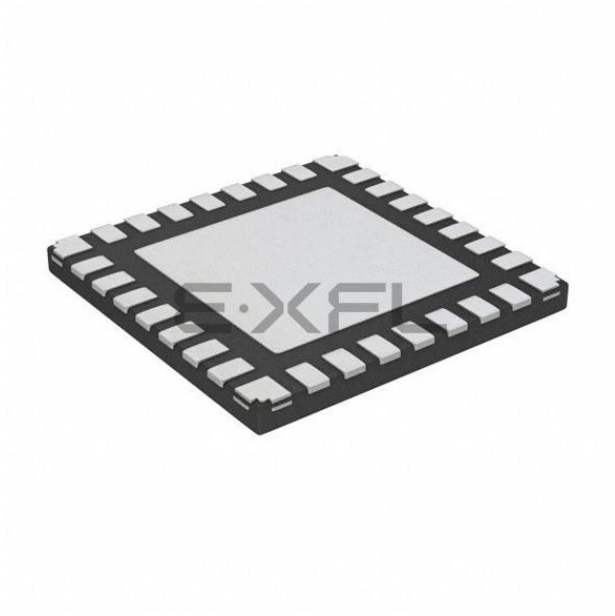


Welcome to E-XFL.COM

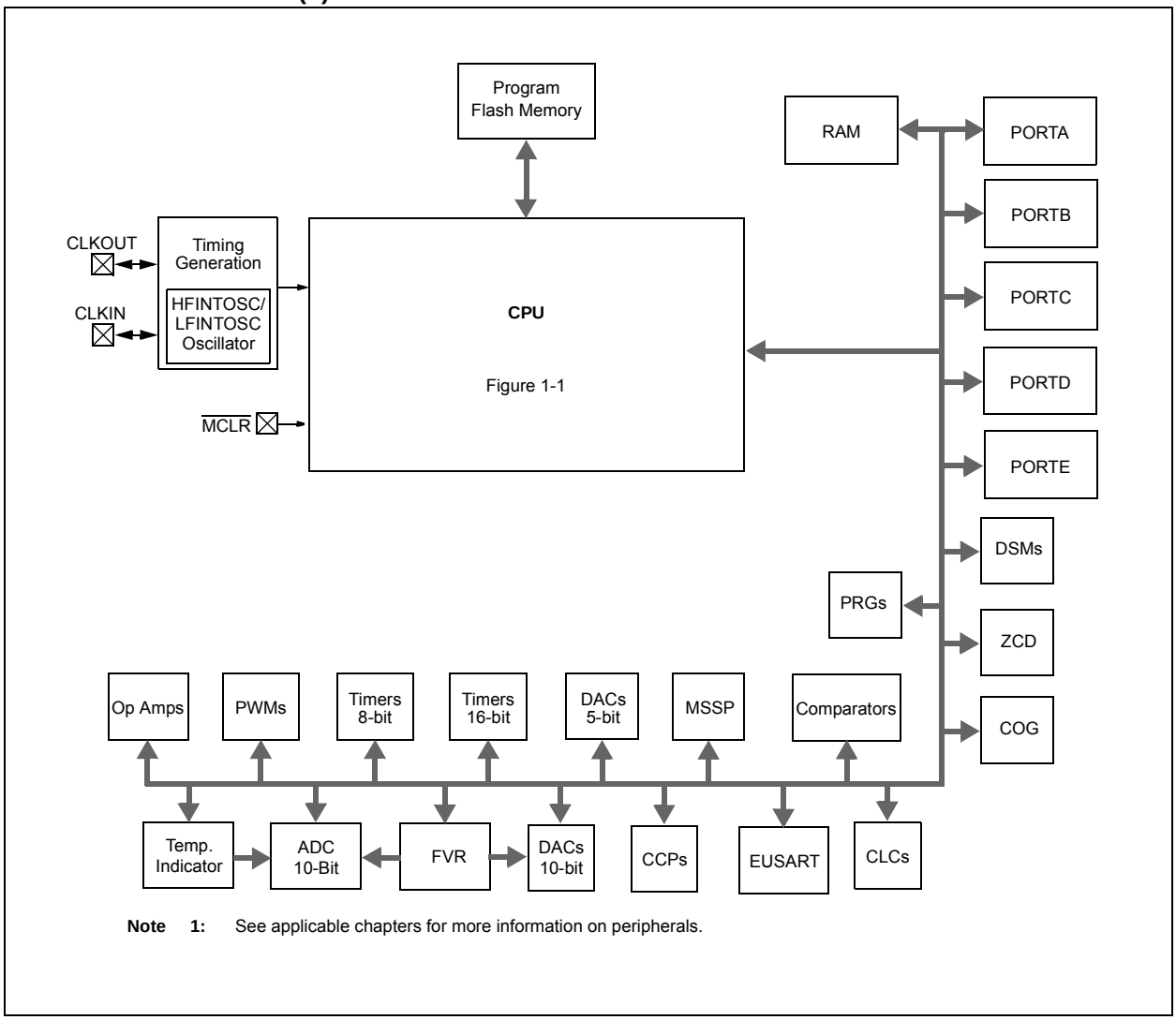
What is "Embedded - Microcontrollers"?

"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	25
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)	1.8V ~ 3.6V
Data Converters	A/D 17x10b; D/A 3x5b, 3x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UQFN Exposed Pad
Supplier Device Package	28-UQFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1778-e-mx

FIGURE 1-1: PIC16(L)F1777/8/9 BLOCK DIAGRAM



Note 1: See applicable chapters for more information on peripherals.

3.4.5 CORE FUNCTION REGISTERS SUMMARY

The Core Function registers listed in Table 3-17 can be addressed from any Bank.

TABLE 3-17: 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', r = reserved. Shaded locations are unimplemented, read as '0'.

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

TABLE 3-18: 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 27											
D8Ch	—	Unimplemented								—	—
D8Dh	—	Unimplemented								—	—
D8Eh	PWMEN	—	—	—	—	MPWM12EN(3)	MPWM11EN	MPWM6EN	MPWM5EN	---- 0000	---- 0000
D8Fh	PWMLD	—	—	—	—	MPWM12LD(3)	MPWM11LD	MPWM6LD	MPWM5LD	---- 0000	---- 0000
D90h	PWMOUT	—	—	—	—	MPWM12OUT(3)	MPWM11OUT	MPWM6OUT	MPWM5OUT	---- 0000	---- 0000
D91h	PWM5PHL	PH<7:0>								xxxx xxxx	uuuu uuuu
D92h	PWM5PHH	PH<15:8>								xxxx xxxx	uuuu uuuu
D93h	PWM5DCL	DC<7:0>								xxxx xxxx	uuuu uuuu
D94h	PWM5DCH	DC<15:8>								xxxx xxxx	uuuu uuuu
D95h	PWM5PRL	PR<7:0>								xxxx xxxx	uuuu uuuu
D96h	PWM5PRH	PR<15:8>								xxxx xxxx	uuuu uuuu
D97h	PWM5OFL	OF<7:0>								xxxx xxxx	uuuu uuuu
D98h	PWM5OFH	OF<15:8>								xxxx xxxx	uuuu uuuu
D99h	PWM5TMRL	TMR<7:0>								0000 0000	0000 0000
D9Ah	PWM5TMRH	TMR<15:8>								0000 0000	0000 0000
D9Bh	PWM5CON	EN	—	OUT	POL	MODE<1:0>		—	—	0-00 00--	0-00 00--
D9Ch	PWM5INTE	—	—	—	—	OFIE	PHIE	DCIE	PRIE	---- 0000	---- 0000
D9Dh	PWM5INTF	—	—	—	—	OFIF	PHIF	DCIF	PRIF	---- 0000	---- 0000
D9Eh	PWM5CLKCON	—	PS<2:0>			—	—	CS<1:0>		-000 --00	-000 --00
D9Fh	PWM5LDCON	LDA	LDT	—	—	—	—	LDS<1:0>		00-- --00	00-- --00
DA0h	PWM5OFCON	—	OFM<1:0>		OFO	—	—	OFS<1:0>		-000 --00	-000 --00
DA1h	PWM6PHL	PH<7:0>								xxxx xxxx	uuuu uuuu
DA2h	PWM6PHH	PH<15:8>								xxxx xxxx	uuuu uuuu
DA3h	PWM6DCL	DC<7:0>								xxxx xxxx	uuuu uuuu
DA4h	PWM6DCH	DC<15:8>								xxxx xxxx	uuuu uuuu
DA5h	PWM6PRL	PR<7:0>								xxxx xxxx	uuuu uuuu
DA6h	PWM6PRH	PR<15:8>								xxxx xxxx	uuuu uuuu
DA7h	PWM6OFL	OF<7:0>								xxxx xxxx	uuuu uuuu
DA8h	PWM6OFH	OF<15:8>								xxxx xxxx	uuuu uuuu
DA9h	PWM6TMRL	TMR<7:0>								0000 0000	0000 0000
DAAh	PWM6TMRH	TMR<15:8>								0000 0000	0000 0000

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: Unimplemented on PIC16LF1777/8/9.
3: Unimplemented on PIC16(L)F1778.

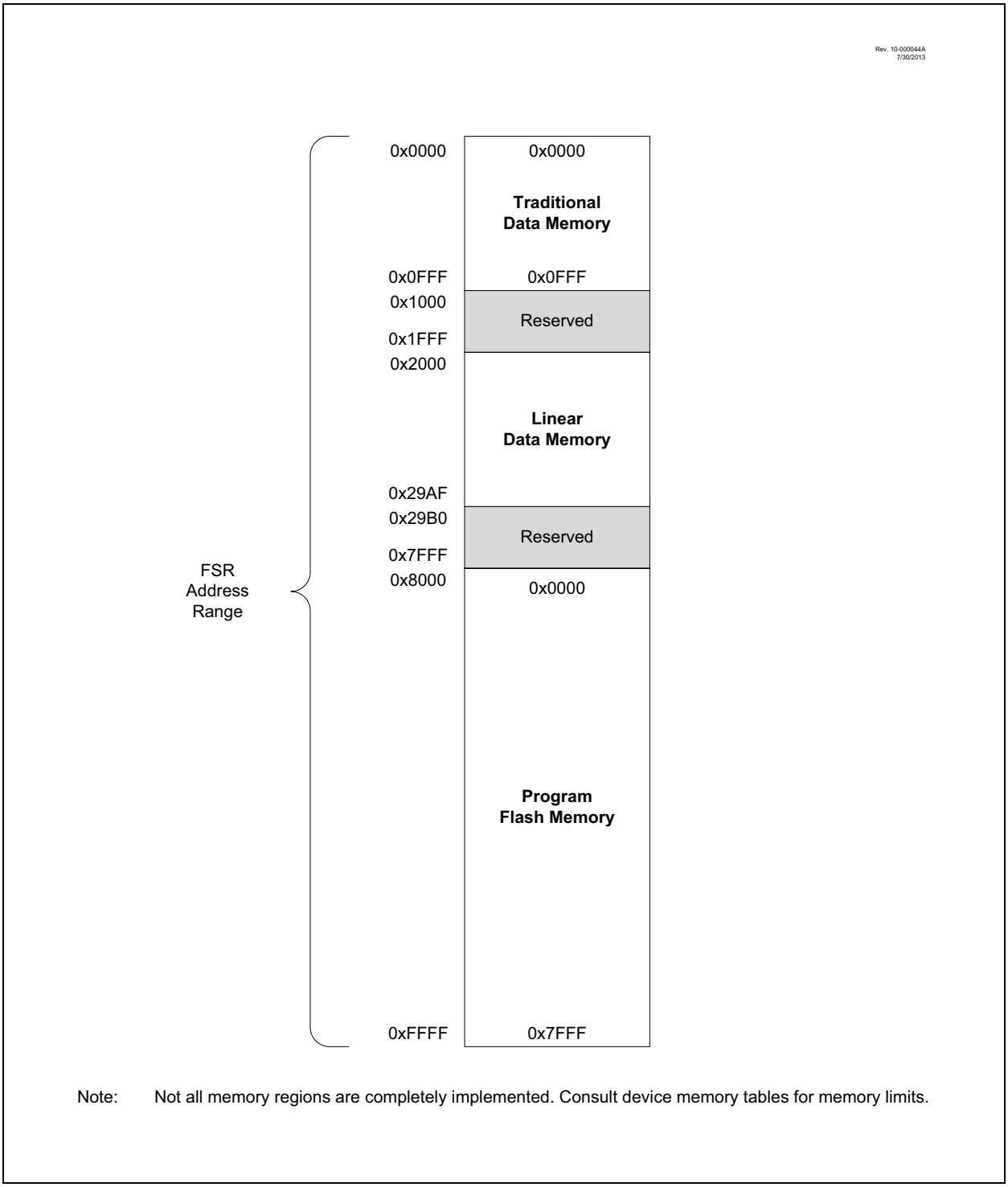
TABLE 3-18: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

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 30 (Continued)											
F2Bh	CLC3GLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	xxxx xxxx	uuuu uuuu
F2Ch	CLC3GLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	xxxx xxxx	uuuu uuuu
F2Dh	CLC3GLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	xxxx xxxx	uuuu uuuu
F2Eh	CLC4CON	EN	OE	OUT	INTP	INTN	MODE<2:0>			0000 0000	0000 0000
F2Fh	CLC4POL	POL	—	—	—	G4POL	G3POL	G2POL	G1POL	0--- xxxx	0--- uuuu
F30h	CLC4SEL0	D1S<7:0>								xxxx xxxx	uuuu uuuu
F31h	CLC4SEL1	D2S<7:0>								xxxx xxxx	uuuu uuuu
F32h	CLC4SEL2	D3S<7:0>								xxxx xxxx	uuuu uuuu
F33h	CLC4SEL3	D4S<7:0>								xxxx xxxx	uuuu uuuu
F34h	CLC4GLS0	G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N	xxxx xxxx	uuuu uuuu
F35h	CLC4GLS1	G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N	xxxx xxxx	uuuu uuuu
F36h	CLC4GLS2	G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N	xxxx xxxx	uuuu uuuu
F37h	CLC4GLS3	G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N	xxxx xxxx	uuuu uuuu
F2Eh — F6Fh	—	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: Unimplemented on PIC16LF1777/8/9.
3: Unimplemented on PIC16(L)F1778.

FIGURE 3-9: INDIRECT ADDRESSING



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

5.2.2.5 Internal Oscillator Frequency Selection

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register.

The postscaled output of the 16 MHz HFINTOSC, 500 kHz MFINTOSC, and 31 kHz LFINTOSC connect to a multiplexer (see Figure 5-1). The Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register select the frequency output of the internal oscillators. One of the following frequencies can be selected via software:

- 32 MHz (requires 4x PLL)
- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz
- 500 kHz (default after Reset)
- 250 kHz
- 125 kHz
- 62.5 kHz
- 31.25 kHz
- 31 kHz (LFINTOSC)

Note:	Following any Reset, the IRCF<3:0> bits of the OSCCON register are set to '0111' and the frequency selection is set to 500 kHz. The user can modify the IRCF bits to select a different frequency.

The IRCF<3:0> bits of the OSCCON register allow duplicate selections for some frequencies. These duplicate choices can offer system design trade-offs. Lower power consumption can be obtained when changing oscillator sources for a given frequency. Faster transition times can be obtained between frequency changes that use the same oscillator source.

5.2.2.6 32 MHz Internal Oscillator Frequency Selection

The Internal Oscillator Block can be used with the 4x PLL associated with the External Oscillator Block to produce a 32 MHz internal system clock source. The following settings are required to use the 32 MHz internal clock source:

- The FOSC bits in Configuration Words must be set to use the INTOSC source as the device system clock (FOSC<2:0> = 100).
- The SCS bits in the OSCCON register must be cleared to use the clock determined by FOSC<2:0> in Configuration Words (SCS<1:0> = 00).
- The IRCF bits in the OSCCON register must be set to the 8 MHz HFINTOSC set to use (IRCF<3:0> = 1110).
- The SPLLEN bit in the OSCCON register must be set to enable the 4x PLL, or the PLLEN bit of the Configuration Words must be programmed to a '1'.

Note:	When using the PLLEN bit of the Configuration Words, the 4x PLL cannot be disabled by software and the SPLLEN option will not be available.

The 4x PLL is not available for use with the internal oscillator when the SCS bits of the OSCCON register are set to '1x'. The SCS bits must be set to '00' to use the 4x PLL with the internal oscillator.

REGISTER 11-20: LATC: PORTC DATA LATCH 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
LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
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 **LATC<7:0>**: PORTC Output Latch Value bits

REGISTER 11-21: ANSELC: PORTC ANALOG SELECT 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	U-0	U-0
ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—
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 7-2 **ANSC<7:2>**: Analog Select between Analog or Digital Function on pins RC<7:2>**(1)**
1 = Analog input. Pin is assigned as analog input**(1)**. Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

bit 1-0 **Unimplemented:** Read as '0'

Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

16.2 ADC Operation

16.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/$\overline{\text{DONE}}$ bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion.

Note:	The GO/$\overline{\text{DONE}}$ bit should not be set in the same instruction that turns on the ADC. Refer to **Section 16.2.6 "ADC Conversion Procedure"**.

16.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/$\overline{\text{DONE}}$ bit
- Set the ADIF Interrupt Flag bit
- Update the ADRESH and ADRESL registers with new conversion result

16.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/$\overline{\text{DONE}}$ bit can be cleared in software. The ADRESH and ADRESL registers will be updated with the partially complete Analog-to-Digital conversion sample. Incomplete bits will match the last bit converted.

Note:	A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

16.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC oscillator source is selected, the ADC waits one additional instruction before starting the conversion. This allows the SLEEP instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

16.2.5 AUTO-CONVERSION TRIGGER

The Auto-conversion Trigger allows periodic ADC measurements without software intervention. When a rising edge of the selected source occurs, the GO/$\overline{\text{DONE}}$ bit is set by hardware.

The Auto-conversion Trigger source is selected with the TRIGSEL<5:0> bits of the ADCON2 register.

Using the Auto-conversion Trigger does not assure proper ADC timing. It is the user's responsibility to ensure that the ADC timing requirements are met.

See Table 16-2 for auto-conversion sources.

TABLE 16-2: AUTO-CONVERSION SOURCES

Source Peripheral	Signal Name
CCP1	CCP1_trigger
CCP2	CCP2_trigger
CCP7	CCP7_trigger
CCP8(1)	CCP8_trigger
Timer0	T0_overflow
Timer1	T1_overflow
Timer3	T3_overflow
Timer5	T5_overflow
Timer2	T2_postscaled
Timer4	T4_postscaled
Timer6	T6_postscaled
Timer8	T8_postscaled
Comparator C1	sync_C1OUT
Comparator C2	sync_C2OUT
Comparator C3	sync_C3OUT
Comparator C4	sync_C4OUT
Comparator C5	sync_C5OUT
Comparator C6	sync_C6OUT
Comparator C7(1)	sync_C7OUT
Comparator C8(1)	sync_C8OUT
CLC1	LC1_out
CLC2	LC2_out
CLC3	LC3_out
CLC4	LC4_out
PWM3	PWM3OUT
PWM4	PWM4OUT
PWM9	PWM9OUT
PWM9	PR/PH/OF/DC9_match
PWM5	PR/PH/OF/DC5_match
PWM6	PR/PH/OF/DC6_match
PWM10(1)	PR/PH/OF/DC10_match
PWM11	PR/PH/OF/DC11_match
PWM12(1)	PR/PH/OF/DC12_match
ADCACT	ADCACTPPS Pin

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

REGISTER 24-4: CCPxCAP: CCPx CAPTURE INPUT SELECTION 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
—	—	—	—	CTS<3:0>			
bit 7							bit 0

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

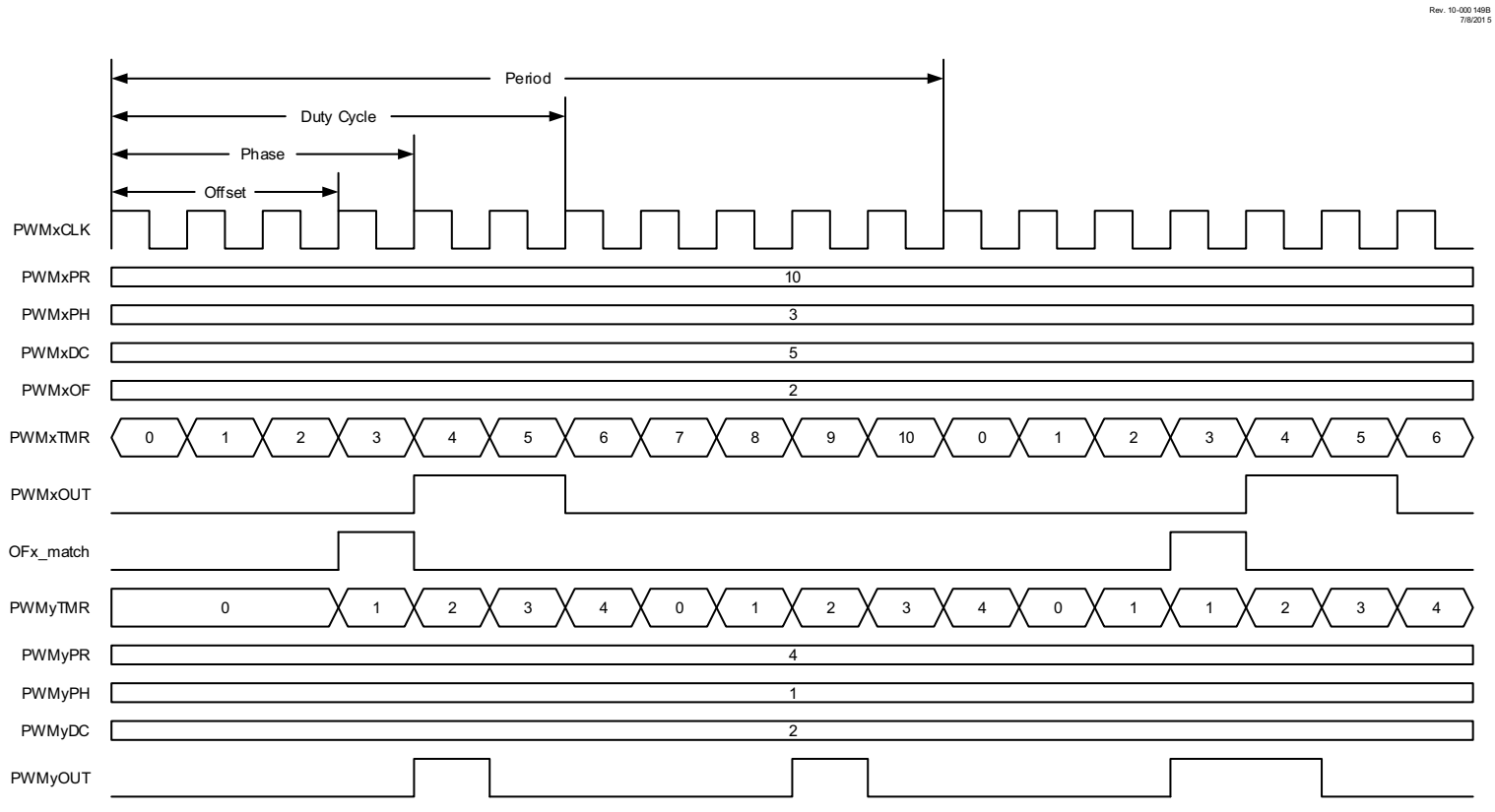
bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **CTS<3:0>:** Capture Trigger Input Selection bits

1101 = IOC_event
1100 = LC4_output
1011 = LC3_output
1010 = LC2_output
1001 = LC1_output
1000 = C8_sync_out(1)
0111 = C7_sync_out(1)
0110 = C6_sync_out
0101 = C5_sync_out
0100 = C4_sync_out
0011 = C3_sync_out
0010 = C2_sync_out
0001 = C1_sync_out
0000 = Pin selected with the CCPxPPS register

Note 1: PIC16LF1777/9 only.

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



26.7 Register Definitions: PWM Control

Long bit name prefixes for the 16-bit PWM peripherals are shown in Table 26-2. Refer to **Section 1.1 "Register and Bit naming conventions"** for more information

TABLE 26-2:

Peripheral	Bit Name Prefix
PWM5	PWM5
PWM6	PWM6
PWM11	PWM11
PWM12(1)	PWM12

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

REGISTER 26-1: PWMxCON: PWM CONTROL REGISTER

R/W-0/0	U-0	R/HS/HC-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0
EN	—	OUT	POL	MODE<1:0>		—	—
bit 7							bit 0

Legend:		
HC = Bit is cleared by hardware		HS = Bit is set by hardware
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 **EN:** PWM Module Enable bit
1 = Module is enabled
0 = Module is disabled

bit 6 **Unimplemented:** Read as '0'

bit 5 **OUT:** Output State of the PWM module

bit 4 **POL:** PWM Output Polarity Control bit
1 = PWM output active state is low
0 = PWM output active state is high

bit 3-2 **MODE<1:0>:** PWM Mode Control bits
11 = Center Aligned mode
10 = Toggle On Match mode
01 = Set On Match mode
00 = Standard PWM mode

bit 1-0 **Unimplemented:** Read as '0'

REGISTER 27-3: COGxRIS0: COG RISING EVENT INPUT SELECTION 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
RIS7	RIS6	RIS5	RIS4	RIS3	RIS2	RIS1	RIS0
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 **RIS<7:0>:** Source Rising Event Input <n> Source Enable bits(1). See Table 27-5.
1 = Source <n> output is enabled as a rising event input
0 = Source <n> output has no effect on the rising event

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

REGISTER 27-4: COGxRIS1: COG RISING EVENT INPUT SELECTION 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
RIS15	RIS14	RIS13	RIS12	RIS11	RIS10	RIS9	RIS8
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 15-8 **RIS<15:8>:** COGx Rising Event Input <n> Source Enable bits(1). See Table 27-5.
1 = Source <n> output is enabled as a rising event input
0 = Source <n> output has no effect on the rising event

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

TABLE 27-5: RISING/FALLING EVENT INPUT SOURCES

Bit <n>	COG1	COG2	COG3(1)	COG3(2)	COG4(1)
15	LC4_out	LC4_out	LC4_out	LC4_out	LC4_out
14	LC3_out	LC3_out	LC3_out	LC3_out	LC3_out
13	LC2_out	LC2_out	LC2_out	LC2_out	LC2_out
12	LC1_out	LC1_out	LC1_out	LC1_out	LC1_out
11	MD1_out	MD2_out	MD3_out	MD3_out	MD4_out
10	PWM6_output	PWM6_output	PWM12_output	Reserved	PWM12_output
9	PWM5_output	PWM5_output	PWM11_output	PWM11_output	PWM11_output
8	PWM4_output	PWM4_output	PWM10_output	Reserved	PWM10_output
7	PWM3_output	PWM3_output	PWM9_output	PWM9_output	PWM9_output
6	CCP2_out	CCP2_out	CCP8_out	CCP7_out	CCP8_out
5	CCP1_out	CCP1_out	CCP7_out	CCP1_out	CCP7_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 with COG1PPS	Pin selected with COG2PPS	Pin selected with COG3PPS	Pin selected with COG3PPS	Pin selected with COG4PPS

Note 1: PIC16(L)F1777/9 only.
2: PIC16(L)F1778 only.

32.5.4 SLAVE MODE 10-BIT ADDRESS RECEPTION

This section describes a standard sequence of events for the MSSP module configured as an I^2C slave in 10-bit Addressing mode.

Figure 32-20 is used as a visual reference for this description.

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

1. Bus starts Idle.
2. Master sends Start condition; S bit of SSPxSTAT is set; SSPxIF is set if Interrupt on Start detect is enabled.
3. Master sends matching high address with R/$\overline{W}$ bit clear; UA bit of the SSPxSTAT register is set.
4. Slave sends $\overline{ACK}$ and SSPxIF is set.
5. Software clears the SSPxIF bit.
6. Software reads received address from SSPxBUF clearing the BF flag.
7. Slave loads low address into SSPxADD, releasing SCL.
8. Master sends matching low address byte to the slave; UA bit is set.

Note:	Updates to the SSPxADD register are not allowed until after the $\overline{ACK}$ sequence.

9. Slave sends $\overline{ACK}$ and SSPxIF is set.

Note:	If the low address does not match, SSPxIF and UA are still set so that the slave software can set SSPxADD back to the high address. BF is not set because there is no match. CKP is unaffected.

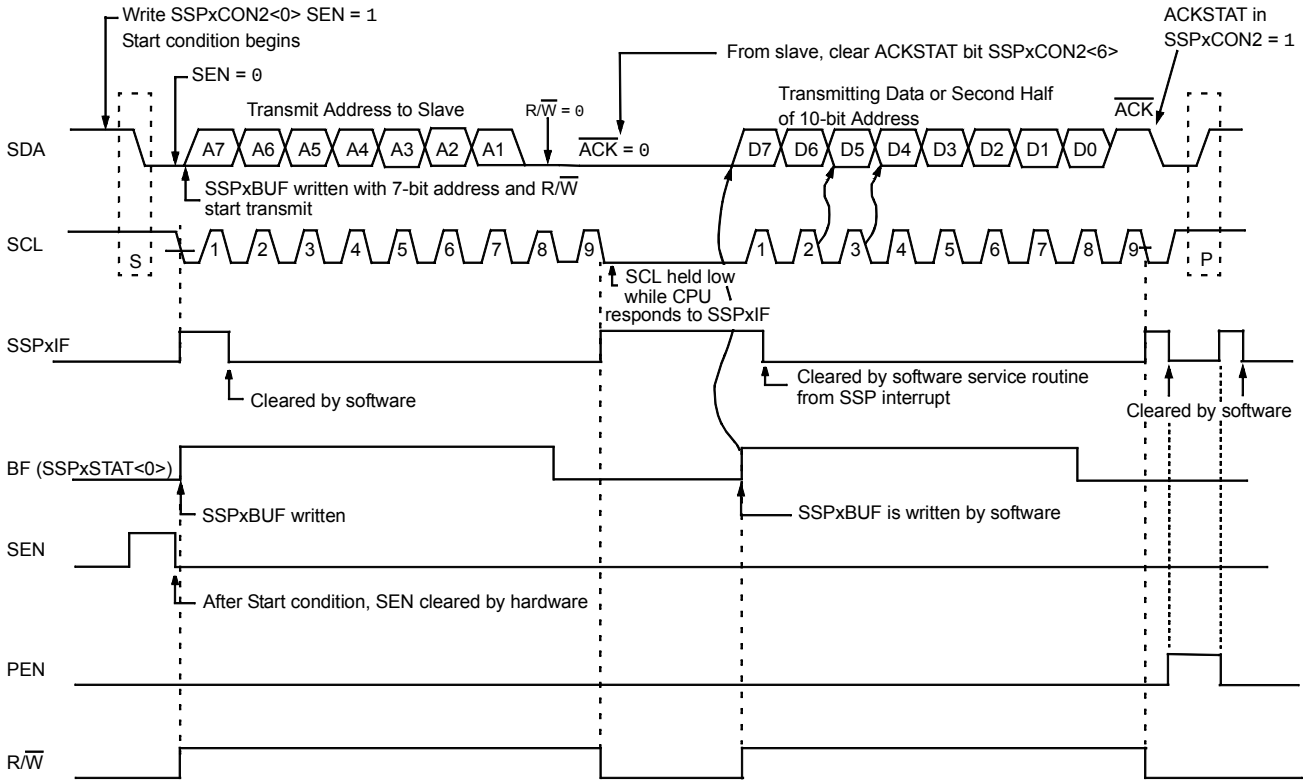
10. Slave clears SSPxIF.
11. Slave reads the received matching address from SSPxBUF clearing BF.
12. Slave loads high address into SSPxADD.
13. Master clocks a data byte to the slave and clocks out the slaves $\overline{ACK}$ on the ninth SCL pulse; SSPxIF is set.
14. If SEN bit of SSPxCON2 is set, CKP is cleared by hardware and the clock is stretched.
15. Slave clears SSPxIF.
16. Slave reads the received byte from SSPxBUF clearing BF.
17. If SEN is set the slave sets CKP to release the SCL.
18. Steps 13-17 repeat for each received byte.
19. Master sends Stop to end the transmission.

32.5.5 10-BIT ADDRESSING WITH ADDRESS OR DATA HOLD

Reception using 10-bit addressing with AHEN or DHEN set is the same as with 7-bit modes. The only difference is the need to update the SSPxADD register using the UA bit. All functionality, specifically when the CKP bit is cleared and SCL line is held low are the same. Figure 32-21 can be used as a reference of a slave in 10-bit addressing with AHEN set.

Figure 32-22 shows a standard waveform for a slave transmitter in 10-bit Addressing mode.

FIGURE 32-28: I^2C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)



32.6.13.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 32-33).

b) SCL is sampled low before SDA is asserted low (Figure 32-34).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

- the Start condition is aborted,
- the BCLIF flag is set and
- the MSSP module is reset to its Idle state (Figure 32-33).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 32-35). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to zero; if the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

Note: The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

FIGURE 32-33: BUS COLLISION DURING START CONDITION (SDA ONLY)

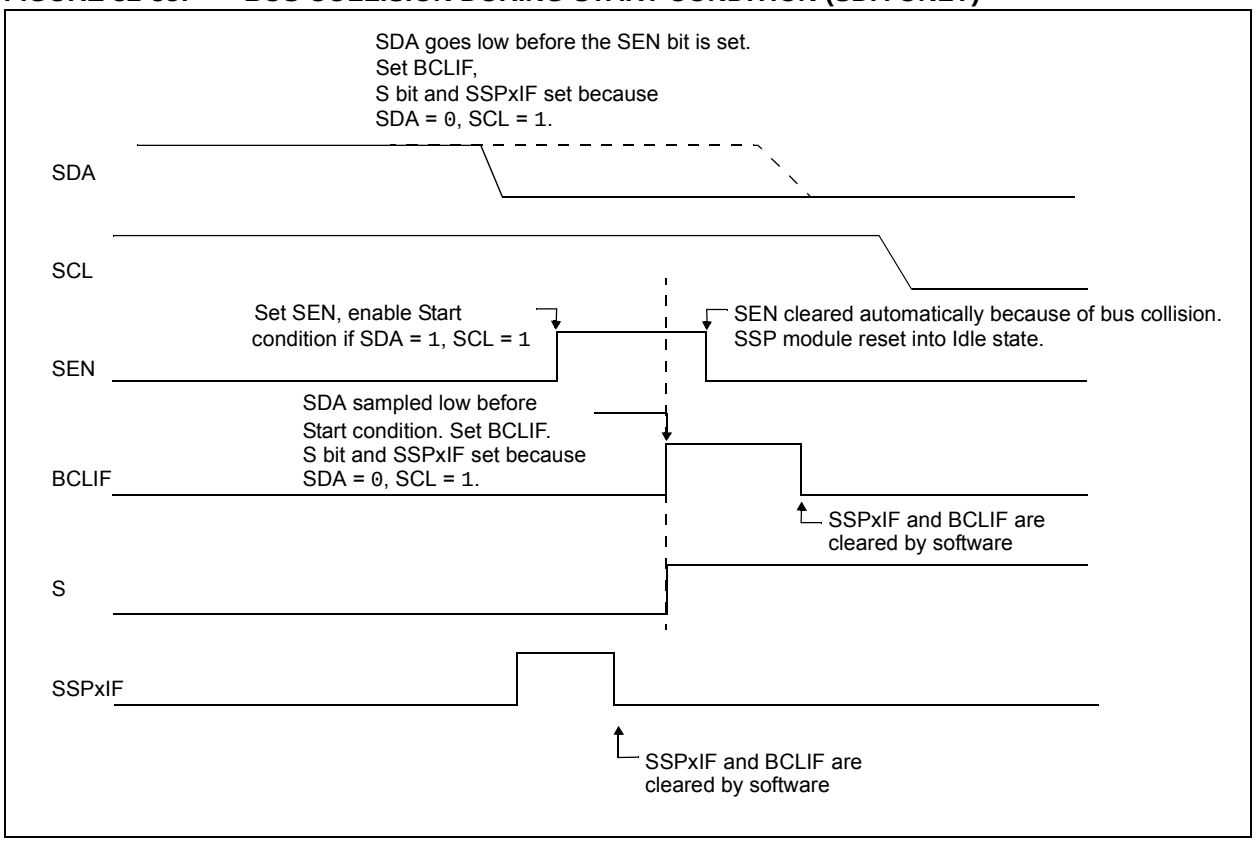
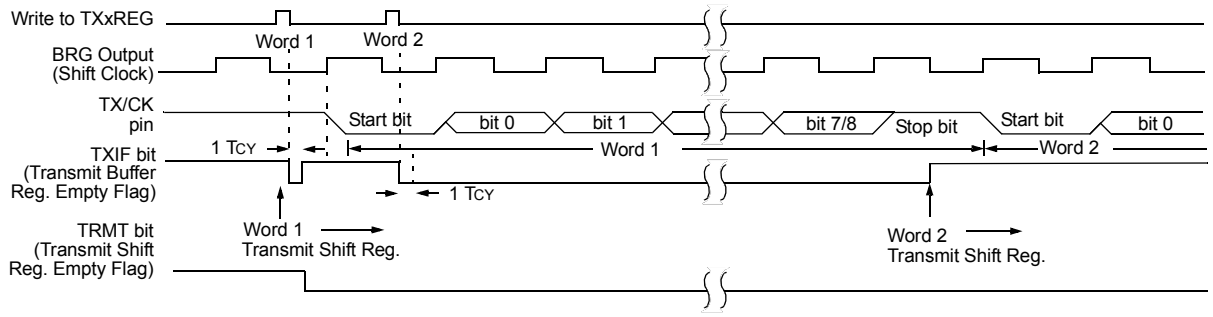


FIGURE 33-4: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 33-1: SUMMARY OF REGISTERS ASSOCIATED WITH ASYNCHRONOUS 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, 207
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 asynchronous transmission.

* Page provides register information.

CALL **Call Subroutine**

Syntax:	[*label*] CALL k
Operands:	0 ≤ k ≤ 2047
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<6:3>) → PC<14:11>
Status Affected:	None
Description:	Call Subroutine. First, return address (PC + 1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a 2-cycle instruction.

CALLW **Subroutine Call With W**

Syntax:	[*label*] CALLW
Operands:	None
Operation:	(PC) +1 → TOS, (W) → PC<7:0>, (PCLATH<6:0>) → PC<14:8>
Status Affected:	None
Description:	Subroutine call with W. First, the return address (PC + 1) is pushed onto the return stack. Then, the contents of W is loaded into PC<7:0>, and the contents of PCLATH into PC<14:8>. CALLW is a 2-cycle instruction.

CLRF **Clear f**

Syntax:	[*label*] CLRF f
Operands:	0 ≤ f ≤ 127
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Description:	The contents of register ‘f’ are cleared and the Z bit is set.

CLRW **Clear W**

Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Description:	W register is cleared. Zero bit (Z) is set.

CLRWDT **Clear Watchdog Timer**

Syntax:	[*label*] CLRWDT
Operands:	None
Operation:	00h → WDT 0 → WDT prescaler, 1 → $\overline{TO}$ 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
Description:	CLRWDT instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

COMF **Complement f**

Syntax:	[*label*] COMF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	($\bar{f}$) → (destination)
Status Affected:	Z
Description:	The contents of register ‘f’ are complemented. If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

DECF **Decrement f**

Syntax:	[*label*] DECF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) - 1 → (destination)
Status Affected:	Z
Description:	Decrement register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

RETFIE **Return from Interrupt**

Syntax:	[*label*] RETFIE k
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None
Description:	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction.
Words:	1
Cycles:	2
Example:	`RETFIE` After Interrupt PC = TOS GIE = 1

RETLW **Return with literal in W**

Syntax:	[*label*] RETLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → (W); TOS → PC
Status Affected:	None
Description:	The W register is loaded with the 8-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.
Words:	1
Cycles:	2

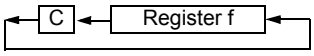
Example:	`CALL TABLE;W contains table`
	`;offset value`
	`;W now has table value`
	`TABLE`
	`ADDWF PC ;W = offset`
	`RETLW k1 ;Begin table`
	`RETLW k2 ;`
	`RETLW kn ; End of table`
	Before Instruction
	W = 0x07
	After Instruction
	W = value of k8

RETURN **Return from Subroutine**

Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a 2-cycle instruction.

RLF **Rotate Left f through Carry**

Syntax:	[*label*] RLF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	See description below
Status Affected:	C
Description:	The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example:	`RLF REG1,0` Before Instruction REG1 = 1110 0110 C = 0 After Instruction REG1 = 1110 0110 W = 1100 1100 C = 1



37.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

The graphs and tables provided in this section are for **design guidance** and are **not tested**.

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified V_{DD} range). This is for **information only** and devices are ensured to operate properly only within the specified range.

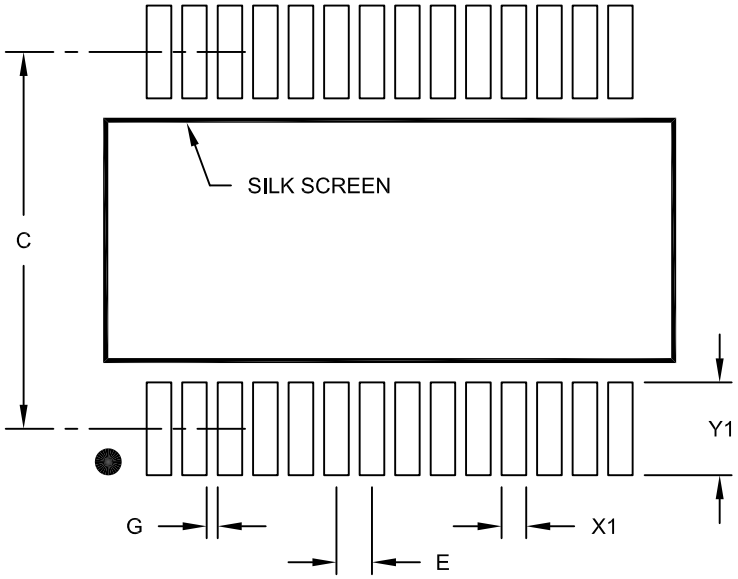
Unless otherwise noted, all graphs apply to both the L and LF devices.

Note:	The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

"Typical" represents the mean of the distribution at 25°C. "Maximum", "Max.", "Minimum" or "Min." represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over each temperature range.

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

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2073A