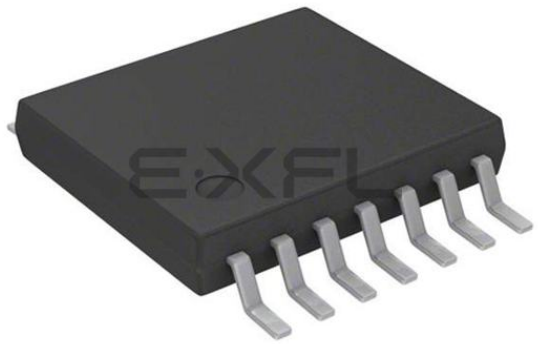


Welcome to E-XFL.COM

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	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 11x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	14-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	14-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f18324-i-st

TABLE 4-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18324	PIC16(L)F18344	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 3													
CPU CORE REGISTERS; see Table 4-2 for specifics													
18Ch	ANSELA			—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	--xx -xxx	--uu -uuu
18Dh	ANSELB	X	—	Unimplemented								—	—
		—		ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	xxxx ----	uuuu ----
18Eh	ANSELC		—	—	—	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	--xx xxxx	--uu uuuu
		—	X	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	xxxx xxxx	uuuu uuuu
18Fh	—	—		Unimplemented								—	—
190h	—	—		Unimplemented								—	—
191h	—	—		Unimplemented								—	—
192h	—	—		Unimplemented								—	—
193h	—	—		Unimplemented								—	—
194h	—	—		Unimplemented								—	—
195h	—	—		Unimplemented								—	—
196h	—	—		Unimplemented								—	—
197h	VREGCON(1)			—	—	—	—	—	—	VREGPM	Reserved	---- --01	---- --01
198h	—	—		Unimplemented								—	—
199h	RC1REG			RC1REG<7:0>								0000 0000	0000 0000
19Ah	TX1REG			TX1REG<7:0>								0000 0000	0000 0000
19Bh	SP1BRGL			SP1BRG<7:0>								0000 0000	0000 0000
19Ch	SP1BRGH			SP1BRG<15:8>								0000 0000	0000 0000
19Dh	RC1STA			SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19Eh	TX1STA			CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TMRT	TX9D	0000 0010	0000 0010
19Fh	BAUD1CON			ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00

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

Note 1: Only on PIC16F18324/18344.

2: Register accessible from both User and ICD Debugger.

TABLE 4-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18324	PIC16(L)F18344	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													
CPU CORE REGISTERS; see Table 4-2 for specifics													
40Ch to 410h	—	—		Unimplemented								—	—
411h	TMR3L			TMR3L<7:0>								xxxx xxxx	uuuu uuuu
412h	TMR3H			TMR3H<7:0>								xxxx xxxx	uuuu uuuu
413h	T3CON			TMR3CS<1:0>		T3CKPS<1:0>		T3SOSC	$\overline{\text{T3SYNC}}$	—	TMR3ON	0000 00-0	uuuu uu-u
414h	T3GCON			TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/$\overline{\text{DONE}}$	T3GVAL	T3GSS<1:0>		0000 0x00	uuuu uxuu
415h	TMR4			TMR4<7:0>								0000 0000	0000 0000
416h	PR4			PR4<7:0>								1111 1111	1111 1111
417h	T4CON			—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>		-000 0000	-000 0000
418h	TMR5L			TMR5L<7:0>								xxxx xxxx	uuuu uuuu
419h	TMR5H			TMR5H<7:0>								xxxx xxxx	uuuu uuuu
41Ah	T5CON			TMR5CS<1:0>		T5CKPS<1:0>		T5SOSC	$\overline{\text{T5SYNC}}$	—	TMR5ON	0000 00-0	uuuu uu-u
41Bh	T5GCON			TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/$\overline{\text{DONE}}$	T5GVAL	T5GSS<1:0>		0000 0x00	uuuu uxuu
41Ch	TMR6			TMR6<7:0>								0000 0000	0000 0000
41Dh	PR6			PR6<7:0>								1111 1111	1111 1111
41Eh	T6CON			—	T6OUTPS<3:0>				TMR6ON	T6CKPS<1:0>		-000 0000	-000 0000
41Fh	—	—		Unimplemented								—	—

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

Note 1: Only on PIC16F18324/18344.

2: Register accessible from both User and ICD Debugger.

REGISTER 7-4: OSCSTAT1: OSCILLATOR STATUS REGISTER 1

R-q/q	R-q/q	U-0	R-q/q	R-q/q	R-q/q	U-0	R-q/q
EXTOR	HFOR	—	LFOR	SOR	ADOR	—	PLLR
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 = Reset value is determined by hardware

bit 7 **EXTOR:** EXTOSC (external) Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 6 **HFOR:** HFINTOSC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **LFOR:** LFINTOSC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 3 **SOR:** Secondary Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 2 **ADOR:** ADCRC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **PLLR:** PLL is Ready bit
1 = The PLL is ready to be used
0 = The PLL is not enabled, the required input source is not ready, or the PLL is not ready.

REGISTER 7-6: OSCFRQ: HFINTOSC FREQUENCY SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
—	—	—	—	HFFRQ<3:0>			
bit 7							bit 0

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

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

bit 3-0 **HFFRQ<3:0>:** HFINTOSC Frequency Selection bits

HFFRQ<2:0>	Nominal Freq. (MHz) (NOSC = 110)	2x PLL Freq. (MHz) (NOSC = 000)
0000	1	Reserved
0001	2	
0010	Reserved	
0011	4	
0100	8	16
0101	12	24
0110	16	32
0111	32	Reserved
1xxx	32	Reserved

9.2 IDLE Mode

When the IDLE Enable (IDLEN) bit is clear (IDLEN = 0), the SLEEP instruction will put the device into full Sleep mode (see **Section 9.3 "Sleep Mode"**). When IDLEN is set (IDLEN = 1), the SLEEP instruction will put the device into IDLE mode. In IDLE mode, the CPU and memory operations are halted, but the peripheral clocks continue to run. This mode is similar to DOZE mode, except that in IDLE both the CPU and program memory are shut off.

Note:	Peripherals using FOSC will continue running while in IDLE (but not in SLEEP). Peripherals using HFINTOSC, LFINTOSC, or SOSC will continue operation in both IDLE and SLEEP.

Note:	If CLKOUT is enabled (CLKOUT = 0, Configuration Word 1), the output will continue operating while in IDLE.

9.2.0.1 IDLE and Interrupts

IDLE mode ends when an interrupt occurs (even if GIE = 0), but IDLEN is not changed. The device can re-enter IDLE by executing the SLEEP instruction.

If Recover-on-Interrupt is enabled (ROI = 1), the interrupt that brings the device out of IDLE also restores full-speed CPU execution when DOZE is also enabled.

9.2.0.2 IDLE and WDT

When in IDLE, the WDT Reset is blocked and will instead wake the device. The WDT wake-up is not an interrupt, therefore ROI does not apply.

Note:	The WDT can bring the device out of IDLE, in the same way it brings the device out of Sleep. The DOZEN bit is not affected.

9.3 Sleep Mode

Sleep mode is entered by executing the SLEEP instruction, while the Idle Enable (IDLEN) bit of the CPUDOZE register is clear (IDLEN = 0). If the SLEEP instruction is executed while the IDLEN bit is set (IDLEN = 1), the CPU will enter the Idle mode (**Section 9.3.3 "Low Power Sleep Mode"**).

Upon entering Sleep mode, the following conditions exist:

1. Resets other than WDT are not affected by Sleep mode; WDT will be cleared but keeps running if enabled for operation during Sleep
2. The $\overline{PD}$ bit of the STATUS register is cleared
3. The $\overline{TO}$ bit of the STATUS register is set
4. The CPU and System clocks are disabled
5. 31 kHz LFINTOSC, HFINTOSC and SOSC will remain enabled if any peripheral has requested them as a clock source or if the HFOEN, LFOEN, or SOSCEN bits of the OSCEN register are set.
6. ADC is unaffected if the dedicated ADCRC oscillator is selected. When the ADC clock source is something other than ADCRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.
7. I/O ports maintain the status they had before SLEEP was executed (driving high, low, or high-impedance) only if no peripheral connected to the I/O port is active.

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 any 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 24.0 "5-Bit Digital-to-Analog Converter (DAC1) Module"** and **Section 16.0 "Fixed Voltage Reference (FVR)"** for more information on these modules.

TABLE 13-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE (CONTINUED)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>					159
CLCIN3PPS	—	—	—	CLCIN3PPS<4:0>					159
RA0PPS	—	—	—	RA0PPS<4:0>					160
RA1PPS	—	—	—	RA1PPS<4:0>					160
RA2PPS	—	—	—	RA2PPS<4:0>					160
RA4PPS	—	—	—	RA4PPS<4:0>					160
RA5PPS	—	—	—	RA5PPS<4:0>					160
RB4PPS(1)	—	—	—	RB4PPS<4:0>					160
RB5PPS(1)	—	—	—	RB5PPS<4:0>					160
RB6PPS(1)	—	—	—	RB6PPS<4:0>					160
RB7PPS(1)	—	—	—	RB7PPS<4:0>					160
RC0PPS	—	—	—	RC0PPS<4:0>					160
RC1PPS	—	—	—	RC1PPS<4:0>					160
RC2PPS	—	—	—	RC2PPS<4:0>					160
RC3PPS	—	—	—	RC3PPS<4:0>					160
RC4PPS	—	—	—	RC4PPS<4:0>					160
RC5PPS	—	—	—	RC5PPS<4:0>					160
RC6PPS(1)	—	—	—	RC6PPS<4:0>					160
RC7PPS(1)	—	—	—	RC7PPS<4:0>					160

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

Note 1: PIC16(L)F18344 only.

18.12 Register Definitions: Comparator Control

REGISTER 18-1: CMxCON0: COMPARATOR Cx CONTROL REGISTER 0

R/W-0/0	R-0/0	U-0	R/W-0/0	U-0	R/W-1/1	R/W-0/0	R/W-0/0
CxON	CxOUT	—	CxPOL	—	CxSP	CxHYS	CxSYNC
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 **CxON:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled and consumes no active power

bit 6 **CxOUT:** Comparator Output bit
If CxPOL = 1 (inverted polarity):
1 = CxVP < CxVN
0 = CxVP > CxVN
If CxPOL = 0 (non-inverted polarity):
1 = CxVP > CxVN
0 = CxVP < CxVN

bit 5 **Unimplemented:** Read as '0'

bit 4 **CxPOL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

bit 3 **Unimplemented:** Read as '0'.

bit 2 **CxSP:** Comparator Speed/Power Select bit
1 = Comparator operates in Normal-Power, High-Speed mode
0 = Reserved. (do not use)

bit 1 **CxHYS:** Comparator Hysteresis Enable bit
1 = Comparator hysteresis enabled
0 = Comparator hysteresis disabled

bit 0 **CxSYNC:** Comparator Output Synchronous Mode bit
1 = Comparator output to Timer1 and I/O pin is synchronous to changes on Timer1 clock source. Output updated on the falling edge of Timer1 clock source.
0 = Comparator output to Timer1 and I/O pin is asynchronous

REGISTER 18-3: CMOUT: COMPARATOR OUTPUT REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	MC2OUT	MC1OUT
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 **Unimplemented:** Read as ‘0’

bit 1 **MC2OUT:** Mirror Copy of C2OUT bit

bit 0 **MC1OUT:** Mirror Copy of C1OUT bit

TABLE 18-3: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	141
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	147
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	154
TRISA	—	—	TRISA5	TRISA4	—	TRISA2	TRISA1	TRISA0	140
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	146
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	153
CMxCON0	CxON	CxOUT	—	CxPOL	—	CxSP	CxHYS	CxSYNC	187
CMxCON1	CxINTP	CxINTN	CxPCH<2:0>			CxNCH<2:0>			188
CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	189
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		177
DACCON0	DAC1EN	—	DAC1OE	—	DAC1PSS<1:0>		—	DAC1NSS	260
DACCON1	—	—	—	DAC1R<4:0>					261
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	98
PIE2	TMR6IE	C2IE	C1IE	NVMIE	—	—	TMR4IE	NCO1IE	101
PIR2	TMR6IF	C2IF	C1IF	NVMIF	—	—	TMR4IF	NCO1IF	106
CMPxPPS	—	—	—	CMPxPPS<4:0>					159
CLCINxPPS	—	—	—	CLCINxPPS<4:0>					159
MDMINPPS	—	—	—	MDMINPPS<4:0>					159
T1GPPS	—	—	—	T1GPPS<4:0>					159
CWGxAS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	215

Legend: — = unimplemented location, read as ‘0’. Shaded cells are unused by the comparator module.
Note 1: PIC16(L)F18344 only.

REGISTER 20-4: CWGxDAT: CWGx DATA 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
—	—	—	—	DAT<3:0>			
bit 7							bit 0

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

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **DAT<3:0>:** CWG Data Input Selection bits

DAT	Data Source
0000	CWGxPPS
0001	C1OUT
0010	C2OUT
0011	CCP1
0100	CCP2
0101	CCP3
0110	CCP4
0111	PWM5
1000	PWM6
1001	NCO1
1010	CLC1
1011	CLC2
1100	CLC3
1101	CLC4
1110	Reserved
1111	Reserved

TABLE 20-2: SUMMARY OF REGISTERS ASSOCIATED WITH CWGx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
TRISA	—	—	TRISA5	TRISA4	—(2)	TRISA2	TRISA1	TRISA0	140
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	141
TRISB(1)	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	146
ANSELB(1)	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	147
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	153
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	154
PIR4	CWG2IF	CWG1IF	TMR5GIF	TMR5IF	CCP4IF	CCP3IF	CCP2IF	CCP1IF	108
PIE4	CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	103
CWG1CON0	EN	LD	—	—	—	MODE<2:0>			210
CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	211
CWG1CLKCON	—	—	—	—	—	—	—	CS	211
CWG1DAT	—	—	—	—	DAT<3:0>				212
CWG1STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	213
CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	214
CWG1AS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	215
CWG1DBR	—	—	DBR<5:0>						215
CWG1DBF	—	—	DBF<5:0>						216
CWG1PPS	—	—	—	CWG1PPS<4:0>					159
CWG2CON0	EN	LD	—	—	—	MODE<2:0>			210
CWG2CON1	—	—	IN	—	POLD	POLC	POLB	POLA	211
CWG2CLKCON	—	—	—	—	—	—	—	CS	211
CWG2DAT	—	—	—	—	DAT<3:0>				212
CWG2STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	213
CWG2AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	214
CWG2AS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	215
CWG2DBR	—	—	DBR<5:0>						215
CWG2DBF	—	—	DBF<5:0>						216
CWG2PPS	—	—	—	CWG2PPS<4:0>					159

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by interrupts.

Note 1: PIC16(L)F18344 only.
2: Unimplemented, read as '0'.

21.0 CONFIGURABLE LOGIC CELL (CLC)

The Configurable Logic Cell (CLCx) provides programmable logic that operates outside the speed limitations of software execution. The logic cell takes up to 36 input signals and, through the use of configurable gates, reduces the 36 inputs to four logic lines that drive one of eight selectable single-output logic functions.

Input sources are a combination of the following:

- I/O pins
- Internal clocks
- Peripherals
- Register bits

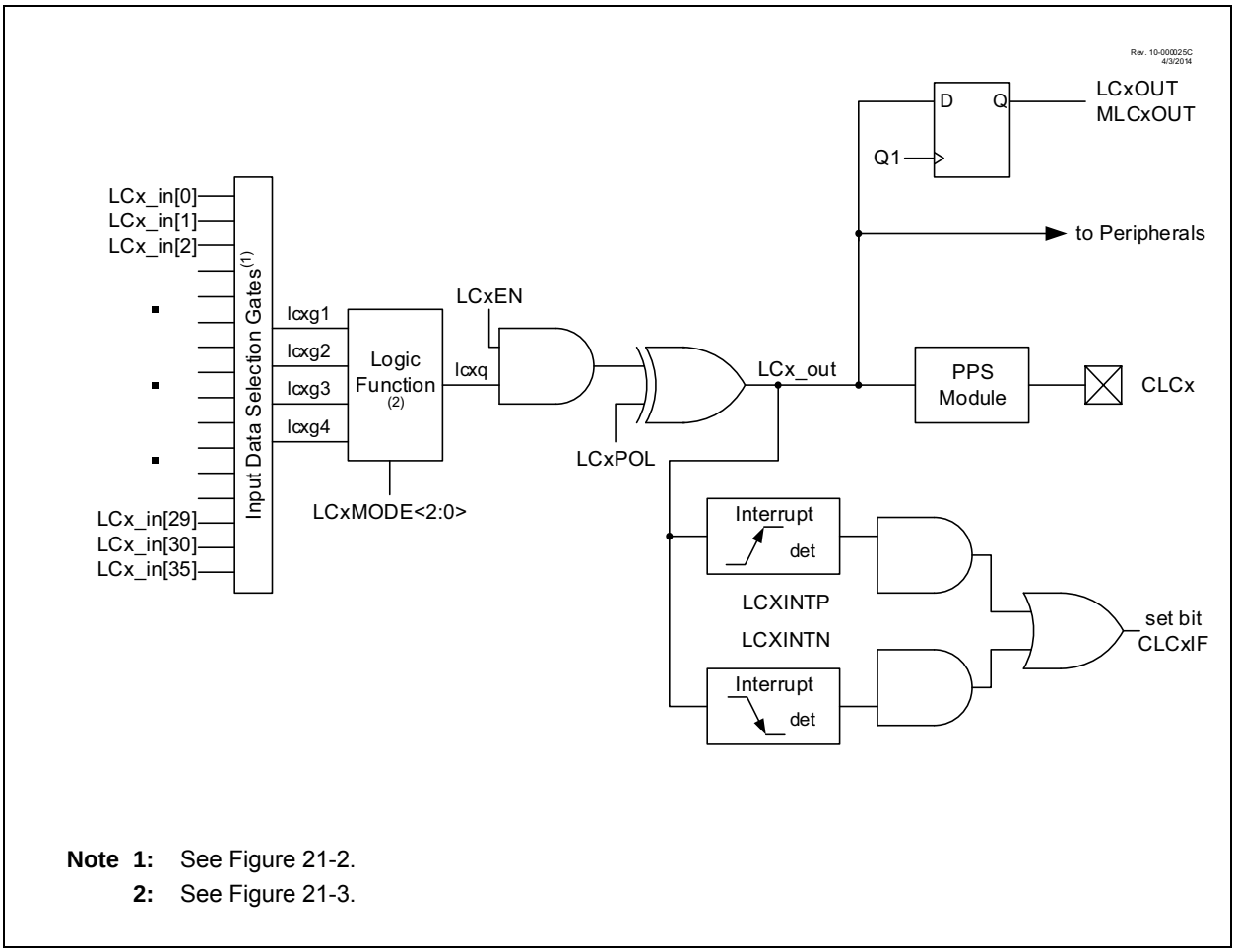
The output can be directed internally to peripherals and to an output pin.

Refer to Figure 21-1 for a simplified diagram showing signal flow through the CLCx.

Possible configurations include:

- Combinatorial Logic
  - AND
  - NAND
  - AND-OR
  - AND-OR-INVERT
  - OR-XOR
  - OR-XNOR
- Latches
  - S-R
  - Clocked D with Set and Reset
  - Transparent D with Set and Reset
  - Clocked J-K with Reset

FIGURE 21-1: CLCx SIMPLIFIED BLOCK DIAGRAM



Note 1: See Figure 21-2.
2: See Figure 21-3.

22.3 ADC Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 22-4. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), refer to Figure 22-4. **The maximum recommended impedance for analog sources is 10 kΩ.**

As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an ADC acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 22-1 may be used. This equation assumes that 1/2 LSb error is used (1,024 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

EQUATION 22-1: ACQUISITION TIME EXAMPLE

Assumptions: $Temperature = 50°C$ *and external impedance of* $10k\Omega$ *5.0V* V_{DD}

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$
$$= 2\mu s + T_C + [(Temperature - 25°C)(0.05\mu s/°C)]$$

The value for T_C *can be approximated with the following equations:*

$$V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) = V_{CHOLD}$$;[1] V_{CHOLD} *charged to within 1/2 lsb*

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{CHOLD}$$;[2] V_{CHOLD} *charge response to* $V_{APPLIED}$

$$V_{APPLIED}\left(1 - e^{\frac{-T_C}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right)$$;*combining [1] and [2]*

Note: Where n = number of bits of the ADC.

Solving for T_C*:*

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ ln(1/2047)$$
$$= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ ln(0.0004885)$$
$$= 1.37\mu s$$

Therefore:

$$T_{ACQ} = 2\mu s + 892ns + [(50°C - 25°C)(0.05\mu s/°C)]$$
$$= 4.62\mu s$$

Note 1: The reference voltage (VREF) has no effect on the equation, since it cancels itself out.

2: The charge holding capacitor (CHOLD) is not discharged after each conversion.

3: The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

26.6 Timer0 Interrupts

The Timer0 Interrupt Flag bit (TMR0IF) is set when either of the following conditions occur:

- 8-bit TMR0L matches the TMR0H value
- 16-bit TMR0 rolls over from FFFFh

When the postscaler bits (T0OUTPS<3:0>) are set to 1:1 operation (no division), the T0IF Flag bit will be set with every TMR0 match or rollover. In general, the TMR0IF Flag bit will be set every T0OUTPS +1 matches or rollovers.

If Timer0 interrupts are enabled (TMR0IE bit of the PIE0 register = 1), the CPU will be interrupted and the device may wake from Sleep (see **Section 26.5 "Operation During Sleep"** for more details).

26.7 Timer0 Output

The Timer0 output can be routed to any I/O pin via the RxyPPS output selection register (see **Section 13.0 "Peripheral Pin Select (PPS) Module"** for additional information). The Timer0 output can also be used by other peripherals, such as the auto-conversion trigger of the Analog-to-Digital Converter. Finally, the Timer0 output can be monitored through software via the Timer0 Output bit (T0OUT) of the T0CON0 register (Register 26-3).

TMR0_out will be one postscaled clock period when a match occurs between TMR0L and TMR0H in 8-bit mode, or when TMR0 rolls over in 16-bit mode. When a match condition occurs, the Timer0 output will toggle every T0OUTPS + 1 match. The total Timer0 period takes two match events to occur, and creates a 50% duty cycle output.

30.2 SPI Mode Overview

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

The SPI bus specifies four signal connections:

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

Figure 30-1 shows the block diagram of the MSSP module when operating in SPI mode.

The SPI bus operates with a single master device and one or more slave devices. When multiple slave devices are used, an independent Slave Select connection can be used to address each slave individually.

Figure 30-4 shows a typical connection between a master device and multiple slave devices.

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

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

Figure 30-5 shows a typical connection between two processors configured as master and slave devices.

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

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

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

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

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

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

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

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

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

Transmissions must be performed in multiples of eight clock pulses. When there is no more data to be transmitted, the master stops sending the clock signal and it deselects the slave.

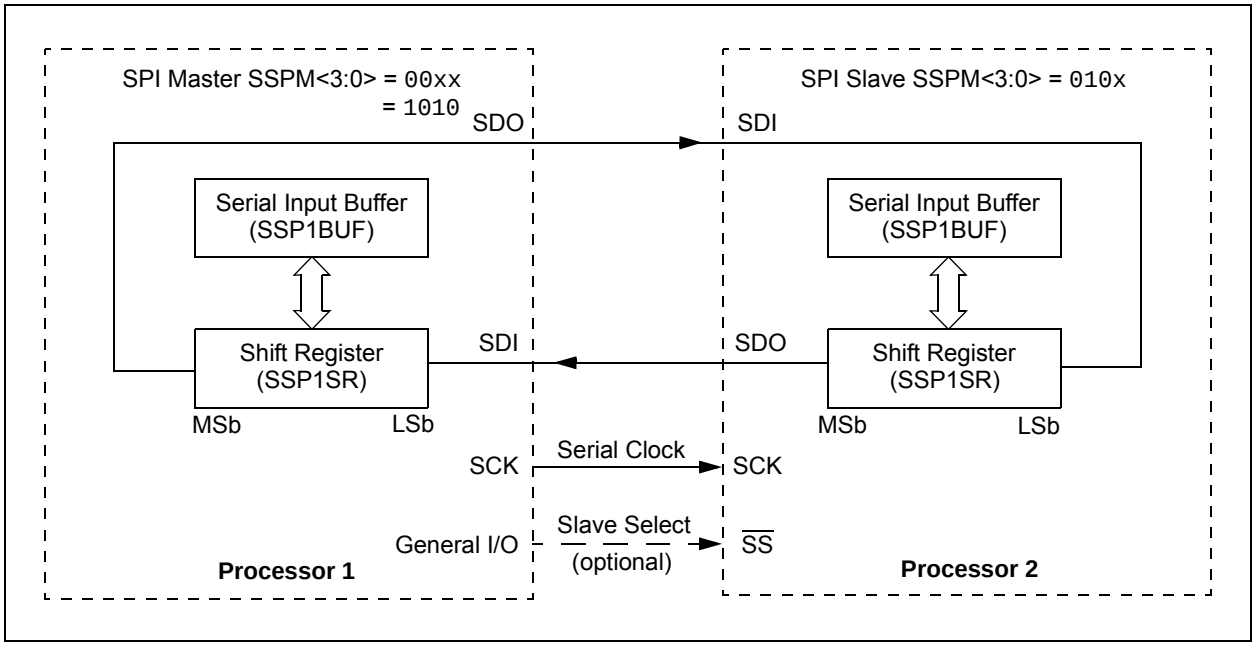
Every slave device connected to the bus that has not been selected through its slave select line must disregard the clock and transmission signals and must not transmit out any data of its own.

The MSSP consists of a transmit/receive shift register (SSP1SR) and a buffer register (SSP1BUF). The SSP1SR shifts the data in and out of the device, MSb first. The SSP1BUF holds the data that was written to the SSP1SR until the received data is ready. Once the eight bits of data have been received, that byte is moved to the SSP1BUF register. Then, the Buffer Full Detect bit, BF of the SSP1STAT register, and the interrupt flag bit, SSP1IF, are set. Any write to the SSP1BUF register during transmission/reception of data will be ignored and the write collision detect bit WCOL of the SSP1CON1 register, will be set. User software must clear the WCOL bit to allow the following write(s) to the SSP1BUF register to complete successfully.

When the application software is expecting to receive valid data, the SSP1BUF should be read before the next byte of data to transfer is written to the SSP1BUF. The Buffer Full bit, BF of the SSP1STAT register, indicates when SSP1BUF has been loaded with the received data (transmission is complete). When the SSP1BUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur.

The SSP1SR is not directly readable or writable and can only be accessed by addressing the SSP1BUF register.

FIGURE 30-5: SPI MASTER/SLAVE CONNECTION



30.2.6 SPI OPERATION IN SLEEP MODE

In SPI Master mode, when the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all eight bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

30.3 I²C Mode Overview

The Inter-Integrated Circuit (I^2C) bus is a multi-master serial data communication bus. Devices communicate in a master/slave environment where the master devices initiate the communication. A slave device is controlled through addressing.

- The I^2C bus specifies two signal connections:
- Serial Clock (SCL)
- Serial Data (SDA)

Figure 30-2 and Figure 30-3 show the block diagram of the MSSP module when operating in I^2C mode.

Both the SCL and SDA connections are bidirectional open-drain lines, each requiring pull-up resistors for the supply voltage. Pulling the line to ground is considered a logical zero and letting the line float is considered a logical one.

Figure 30-11 shows a typical connection between two processors configured as master and slave devices.

The I^2C bus can operate with one or more master devices and one or more slave devices.

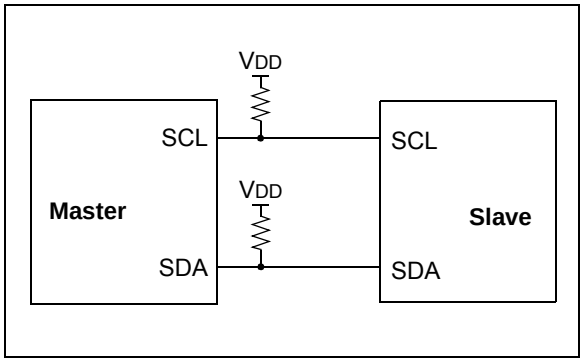
There are four potential modes of operation for a given device:

- Master Transmit mode
  (master is transmitting data to a slave)
- Master Receive mode
  (master is receiving data from a slave)
- Slave Transmit mode
  (slave is transmitting data to a master)
- Slave Receive mode
  (slave is receiving data from the master)

To begin communication, the master device sends out a Start condition followed by the address byte of the slave it intends to communicate with. This is followed by a single Read/Write bit, which determines whether the master intends to transmit to or receive data from the slave device.

If the requested slave exists on the bus, it will respond with an Acknowledge bit, otherwise known as an $\overline{ACK}$. The master then continues to either transmit or receive data from the slave device.

FIGURE 30-11: I²C MASTER/ SLAVE CONNECTION

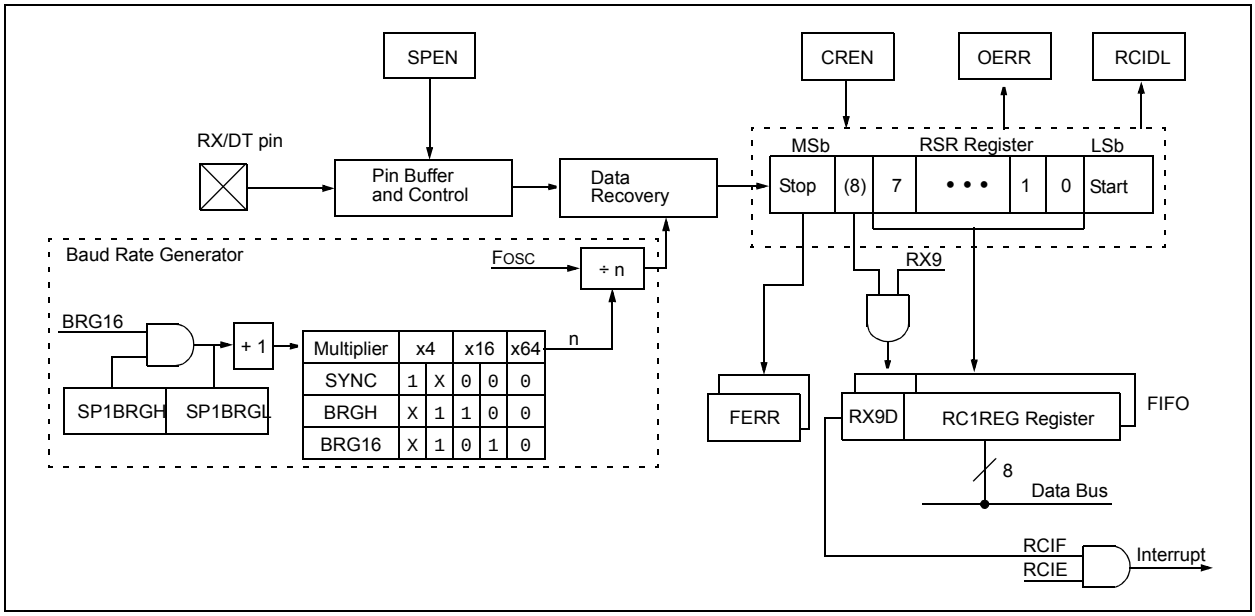


REGISTER 30-2: SSP1CON1: SSP CONTROL REGISTER 1 (CONTINUED)

bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits
1111 = I^2C Slave mode, 10-bit address with Start and Stop bit interrupts enabled
1110 = I^2C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1101 = Reserved
1100 = Reserved
1011 = I^2C firmware controlled Master mode (slave idle)
1010 = SPI Master mode, clock = FOSC/(4 * (SSPADD+1))**(5)**
1001 = Reserved
1000 = I^2C Master mode, clock = FOSC / (4 * (SSPADD+1))**(4)**
0111 = I^2C Slave mode, 10-bit address
0110 = I^2C Slave mode, 7-bit address
0101 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control disabled, $\overline{SS}$ can be used as I/O pin
0100 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control enabled
0011 = SPI Master mode, clock = T2_match/2
0010 = SPI Master mode, clock = FOSC/64
0001 = SPI Master mode, clock = FOSC/16
0000 = SPI Master mode, clock = FOSC/4

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
2: When enabled, these pins must be properly configured as input or output. Use SSP1SSPPS, SSP1CLKPPS, SSP1DATPPS, and RxyPPS to select the pins.
3: When enabled, the SDA and SCL pins must be configured as inputs. Use SSP1CLKPPS, SSP1DATPPS, and RxyPPS to select the pins.
4: SSPADD values of 0, 1 or 2 are not supported for I^2C mode.
5: SSPADD value of '0' is not supported. Use SSPM = 0000 instead.

FIGURE 31-2: EUSART1 RECEIVE BLOCK DIAGRAM



The operation of the EUSART1 module is controlled through three registers:

- Transmit Status and Control (TX1STA)
- Receive Status and Control (RC1STA)
- Baud Rate Control (BAUD1CON)

These registers are detailed in Register 31-1, Register 31-2 and Register 31-3, respectively.

The RX and CK input pins are selected with the RXPPS and CKPPS registers, respectively. TX, CK, and DT output pins are selected with each pin's RxyPPS register. Since the RX input is coupled with the DT output in Synchronous mode, it is the user's responsibility to select the same pin for both of these functions when operating in Synchronous mode. The EUSART1 control logic will control the data direction drivers automatically.

REGISTER 32-1: CLKRCON: REFERENCE CLOCK CONTROL REGISTER

R/W-0/0	U-0	U-0	R/W-1/1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CLKREN	—	—	CLKRDC<1:0>		CLKRDIV<2:0>		
bit 7							bit 0

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

bit 7 **CLKREN:** Reference Clock Module Enable bit
- 1 = Reference Clock module enabled
- 0 = Reference Clock module is disabled

bit 6-5 **Unimplemented:** Read as '0'

bit 4-3 **CLKRDC<1:0>:** Reference Clock Duty Cycle bits (1)
- 11 = Clock outputs duty cycle of 75%
- 10 = Clock outputs duty cycle of 50%
- 01 = Clock outputs duty cycle of 25%
- 00 = Clock outputs duty cycle of 0%

bit 2-0 **CLKRDIV<2:0>:** Reference Clock Divider bits
- 111 = FOSC divided by 128
- 110 = FOSC divided by 64
- 101 = FOSC divided by 32
- 100 = FOSC divided by 16
- 011 = FOSC divided by 8
- 010 = FOSC divided by 4
- 001 = FOSC divided by 2
- 000 = FOSC

Note 1: Bits are valid for Reference Clock divider values of two or larger, the base clock cannot be further divided.

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.