# E·XFL



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I <sup>2</sup> C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 12x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1508-e-p

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

# TABLE 3-5: PIC16(L)F1508/9 MEMORY MAP, BANK 8-23

	BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15
400h		480h		500h		580h		600h		680h		700h		780h	
	Core Registers (Table 3-2)		Core Registers (Table 3-2)		Core Registers (Table 3-2)		Core Registers (Table 3-2)		Core Registers (Table 3-2)		Core Registers (Table 3-2)		Core Registers (Table 3-2)		Core Registers (Table 3-2)
40Bh		48Bh		50Bh		58Bh		60Bh		68Bh		70Bh		78Bh	
40Ch	_	48Ch	_	50Ch	_	58Ch	—	60Ch	—	68Ch	_	70Ch	_	78Ch	—
40Dh	_	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—
40Eh	_	48Eh	_	50Eh	_	58Eh	_	60Eh	_	68Eh	—	70Eh		78Eh	—
40Fh	_	48Fh	_	50Fh	_	58Fh	_	60Fh	_	68Fh	—	70Fh		78Fh	—
410h		490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—
411h	_	491h	_	511h	_	591h	_	611h	PWM1DCL	691h	CWG1DBR	711h		791h	—
412h	_	492h	_	512h	_	592h	_	612h	PWM1DCH	692h	CWG1DBF	712h		792h	—
413h	—	493h	—	513h	—	593h	_	613h	PWM1CON	693h	CWG1CON0	713h	—	793h	—
414h	_	494h	_	514h	_	594h	_	614h	PWM2DCL	694h	CWG1CON1	714h		794h	—
415h		495h	—	515h	—	595h	—	615h	PWM2DCH	695h	CWG1CON2	715h	—	795h	
416h	_	496h	_	516h	_	596h	_	616h	PWM2CON	696h	—	716h		796h	—
417h	—	497h	—	517h	—	597h	_	617h	PWM3DCL	697h	—	717h	—	797h	—
418h	_	498h	NCO1ACCL	518h	_	598h	_	618h	PWM3DCH	698h	—	718h	_	798h	—
419h	—	499h	NCO1ACCH	519h	—	599h	_	619h	PWM3CON	699h	—	719h	—	799h	—
41Ah	_	49Ah	NCO1ACCU	51Ah	_	59Ah	_	61Ah	PWM4DCL	69Ah	—	71Ah		79Ah	—
41Bh	_	49Bh	NCO1INCL	51Bh	_	59Bh	_	61Bh	PWM4DCH	69Bh	—	71Bh		79Bh	—
41Ch	—	49Ch	NCO1INCH	51Ch	—	59Ch	—	61Ch	PWM4CON	69Ch	—	71Ch	—	79Ch	—
41Dh	_	49Dh	_	51Dh	_	59Dh	_	61Dh	_	69Dh	—	71Dh		79Dh	—
41Eh	—	49Eh	NCO1CON	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—
41Fh	_	49Fh	NCO1CLK	51Fh	_	59Fh	_	61Fh	_	69Fh	—	71Fh		79Fh	—
420h		4A0h		520h		5A0h		620h		6A0h		720h		7A0h	
	Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'		Unimplemented Read as '0'
46Fh		4FFh		56Fh		5EFh		66Fh		6FFh		76Fh		7FFh	
470h		4F0h		570h		5F0h		670h		6F0h		770h		7F0h	
-	Accesses		Accesses		Accesses		Accesses		Accesses		Accesses	-	Accesses		Accesses
	70h - 7Fh		70h - 7Fh		70h - 7Fh		70h - 7Fh		70h – 7Fh		70h – 7Fh		70h – 7Fh		70h - 7Fh
4754	/011 /111	4554		<b>C</b> 7 <b>Cb</b>				0754	7011 7111		7011 7111	7754		7664	7011 7111
4750		4660		57FI		SEEU		07FI		0FF11		//F0		7660	
	BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23
800h		880h		900h		980h		A00h		A80h		B00h		B80h	
	Core Registers (Table 3-2)		Core Registers (Table 3-2)		Core Registers (Table 3-2)		Core Registers (Table 3-2)		Core Registers (Table 3-2)		Core Registers (Table 3-2)		Core Registers (Table 3-2)		Core Registers (Table 3-2)
80Bh		88Bh		90Bh		98Bh		A0Bh		A8Bh		B0Bh		B8Bh	
80Ch		88Ch		90Ch		98Ch		A0Ch		A8Ch		B0Ch		B8Ch	
0001	Unimplemented	22011	Unimplemented	00011	Unimplemented	00011	Unimplemented		Unimplemented		Unimplemented	2001	Unimplemented	2001	Unimplemented
	Read as '0'		Read as '0'		Read as '0'		Read as '0'		Read as '0'		Read as '0'		Read as '0'		Read as '0'
86Fh		8EFh		96Fh		9EFh		A6Fh		AEFh		B6Fh		BEFh	
870h		8F0h		970h		9F0h		A70h		AF0h		B70h		BF0h	
	Accesses 70h – 7Fh		Accesses 70h – 7Fh		Accesses 70h – 7Fh		Accesses 70h – 7Fh		Accesses 70h – 7Fh		Accesses 70h – 7Fh		Accesses 70h – 7Fh		Accesses 70h – 7Fh
87Fh		8FFh		97Fh		9FFh		A7Fh		AFFh		B7Fh		BFFh	

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

#### 3.6.2 LINEAR DATA MEMORY

The linear data memory is the region from FSR address 0x2000 to FSR address 0x29AF. This region is a virtual region that points back to the 80-byte blocks of GPR memory in all the banks.

Unimplemented memory reads as 0x00. Use of the linear data memory region allows buffers to be larger than 80 bytes because incrementing the FSR beyond one bank will go directly to the GPR memory of the next bank.

The 16 bytes of common memory are not included in the linear data memory region.

FIGURE 3-10: LINEAR DATA MEMORY MAP



#### 3.6.3 PROGRAM FLASH MEMORY

To make constant data access easier, the entire program Flash memory is mapped to the upper half of the FSR address space. When the MSb of FSRnH is set, the lower 15 bits are the address in program memory which will be accessed through INDF. Only the lower eight bits of each memory location is accessible via INDF. Writing to the program Flash memory cannot be accomplished via the FSR/INDF interface. All instructions that access program Flash memory via the FSR/INDF interface will require one additional instruction cycle to complete.

#### FIGURE 3-11: PROGRAM FLASH MEMORY MAP



# 5.2 Clock Source Types

Clock sources can be classified as external, internal or peripheral.

External clock sources rely on external circuitry for the clock source to function. Examples are: oscillator modules (ECH, ECM, ECL modes), quartz crystal resonators or ceramic resonators (LP, XT and HS modes) and Resistor-Capacitor (EXTRC) mode circuits.

Internal clock sources are contained within the oscillator module. The internal oscillator block has two internal oscillators that are used to generate the internal system clock sources: the 16 MHz High-Frequency Internal Oscillator (HFINTOSC) and the 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The peripheral clock source is a nominal 600 kHz internal RC oscillator, FRC. The FRC is traditionally used with the ADC module, but is sometimes available to other peripherals. See **Section 5.2.2.4** "**Peripheral Clock Sources**".

The system clock can be selected between external or internal clock sources via the System Clock Select (SCS) bits in the OSCCON register. See **Section 5.3 "Clock Switching**" for additional information.

#### 5.2.1 EXTERNAL CLOCK SOURCES

An external clock source can be used as the device system clock by performing one of the following actions:

- Program the FOSC<2:0> bits in the Configuration Words to select an external clock source that will be used as the default system clock upon a device Reset.
- Write the SCS<1:0> bits in the OSCCON register to switch the system clock source to:
  - Secondary oscillator during run-time, or
  - An external clock source determined by the value of the FOSC bits.

See **Section 5.3 "Clock Switching**" for more information.

#### 5.2.1.1 EC Mode

The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the OSC1 input. OSC2/CLKOUT is available for general purpose I/O or CLKOUT. Figure 5-2 shows the pin connections for EC mode.

EC mode has three power modes to select from through the Fosc bits in the Configuration Words:

- ECH High-power, 4-20 MHz
- ECM Medium-power, 0.5-4 MHz
- ECL Low-power, 0-0.5 MHz

The Oscillator Start-up Timer (OST) is disabled when EC mode is selected. Therefore, there is no delay in operation after a Power-on Reset (POR) or wake-up from Sleep. Because the PIC<sup>®</sup> MCU design is fully static, stopping the external clock input will have the effect of halting the device while leaving all data intact. Upon restarting the external clock, the device will resume operation as if no time had elapsed.





### 5.2.1.2 LP, XT, HS Modes

The LP, XT and HS modes support the use of quartz crystal resonators or ceramic resonators connected to OSC1 and OSC2 (Figure 5-3). The three modes select a low, medium or high gain setting of the internal inverter-amplifier to support various resonator types and speed.

LP Oscillator mode selects the lowest gain setting of the internal inverter-amplifier. LP mode current consumption is the least of the three modes. This mode is designed to drive only 32.768 kHz tuning-fork type crystals (watch crystals).

**XT** Oscillator mode selects the intermediate gain setting of the internal inverter-amplifier. XT mode current consumption is the medium of the three modes. This mode is best suited to drive resonators with a medium drive level specification.

**HS** Oscillator mode selects the highest gain setting of the internal inverter-amplifier. HS mode current consumption is the highest of the three modes. This mode is best suited for resonators that require a high drive setting.

Figure 5-3 and Figure 5-4 show typical circuits for quartz crystal and ceramic resonators, respectively.

#### 5.2.2 INTERNAL CLOCK SOURCES

The device may be configured to use the internal oscillator block as the system clock by performing one of the following actions:

- Program the FOSC<2:0> bits in Configuration Words to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the SCS<1:0> bits in the OSCCON register to switch the system clock source to the internal oscillator during run-time. See Section
   5.3 "Clock Switching" for more information.

In **INTOSC** mode, OSC1/CLKIN is available for general purpose I/O. OSC2/CLKOUT is available for general purpose I/O or CLKOUT.

The function of the OSC2/CLKOUT pin is determined by the CLKOUTEN bit in Configuration Words.

The internal oscillator block has two independent oscillators that provides the internal system clock source.

- 1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 16 MHz.
- 2. The **LFINTOSC** (Low-Frequency Internal Oscillator) operates at 31 kHz.

#### 5.2.2.1 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 16 MHz internal clock source.

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 5-1). The frequency derived from the HFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See **Section 5.2.2.6 "Internal Oscillator Clock Switch Timing"** for more information.

The HFINTOSC is enabled by:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired HF frequency, and
- FOSC<2:0> = 100, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '1x'.

A fast start-up oscillator allows internal circuits to power-up and stabilize before switching to HFINTOSC.

The High-Frequency Internal Oscillator Ready bit (HFIOFR) of the OSCSTAT register indicates when the HFINTOSC is running.

The High-Frequency Internal Oscillator Stable bit (HFIOFS) of the OSCSTAT register indicates when the HFINTOSC is running within 0.5% of its final value.

#### 5.2.2.2 LFINTOSC

The Low-Frequency Internal Oscillator (LFINTOSC) is a 31 kHz internal clock source.

The output of the LFINTOSC connects to a multiplexer (see Figure 5-1). Select 31 kHz, via software, using the IRCF<3:0> bits of the OSCCON register. See **Section 5.2.2.6** "Internal Oscillator Clock Switch Timing" for more information. The LFINTOSC is also the frequency for the Power-up Timer (PWRT), Watchdog Timer (WDT) and Fail-Safe Clock Monitor (FSCM).

The LFINTOSC is enabled by selecting 31 kHz (IRCF<3:0> bits of the OSCCON register = 000) as the system clock source (SCS bits of the OSCCON register = 1x), or when any of the following are enabled:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired LF frequency, and
- FOSC<2:0> = 100, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '1x'.

Peripherals that use the LFINTOSC are:

- Power-up Timer (PWRT)
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)

The Low-Frequency Internal Oscillator Ready bit (LFIOFR) of the OSCSTAT register indicates when the LFINTOSC is running.

#### 5.2.2.3 FRC

The FRC clock is an uncalibrated, nominal 600 kHz peripheral clock source.

The FRC is automatically turned on by the peripherals requesting the FRC clock.

The FRC clock continues to run during Sleep.

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	U-0
OSFIF	C2IF	C1IF	<u> </u>	BCL1IF	NCO1IF		—
bit 7							bit 0
Legend:							
R = Readal	ble bit	W = Writable b	bit	U = Unimpler	mented bit, read	as '0'	
u = Bit is ur	nchanged	x = Bit is unkn	own	-n/n = Value a	at POR and BO	R/Value at all o	other Resets
'1' = Bit is s	set	'0' = Bit is clea	red				
bit 7	OSFIF: Osci	llator Fail Interru	pt Flag bit				
	1 = Interrupt 0 = Interrupt	is pending is not pending					
bit 6	C2IF: Compa	arator C2 Interru	pt Flag bit				
	1 = Interrupt 0 = Interrupt	is pending is not pending					
bit 5	C1IF: Compa	arator C1 Interru	pt Flag bit				
	1 = Interrupt	is pending					
hit 1		ns not pending	,				
bit 2		SD Rue Collision		log hit			
DILS	1 = Interrunt	is nending	r interrupt F	lag bit			
	0 = Interrupt	is not pending					
bit 2	NCO1IF: Nu	merically Contro	lled Oscillat	or Flag bit			
	1 = Interrupt 0 = Interrupt	is pending is not pending					
bit 1-0	Unimpleme	nted: Read as '0	,				
Note:	Interrupt flag bits a condition occurs, its corresponding Interrupt Enable	are set when an i regardless of the enable bit or the bit, GIE of the I	nterrupt state of Global NTCON				

### REGISTER 7-6: PIR2: PERIPHERAL INTERRUPT REQUEST REGISTER 2

register. User software should ensure the appropriate interrupt flag bits are clear prior

to enabling an interrupt.

# 11.5 PORTB Registers

#### 11.5.1 DATA REGISTER

PORTB is a 4-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 11-8). Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., disable the output driver). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTB register (Register 11-7) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATB).

#### 11.5.2 DIRECTION CONTROL

The TRISB register (Register 11-8) controls the PORTB pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISB register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

#### 11.5.3 ANALOG CONTROL

The ANSELB register (Register 11-10) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELB bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELB bits has no effect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

Note:	The ANSELB bits default to the Analog
	mode after Reset. To use any pins as
	digital general purpose or peripheral
	inputs, the corresponding ANSEL bits
	must be initialized to '0' by user software.

# 11.5.4 PORTB FUNCTIONS AND OUTPUT PRIORITIES

Each PORTB pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 11-5.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input functions, such as ADC and comparator inputs, are not shown in the priority lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below in Table 11-5.

Pin Name	Function Priority <sup>(1)</sup>
RB4	SDA
	RB4
RB5	RB5
RB6	SCL
	SCK
	RB6
RB7	CLC3
	TX
	RB7

# TABLE 11-5: PORTB OUTPUT PRIORITY

Note 1: Priority listed from highest to lowest.

2: Default pin (see APFCON register).

3: Alternate pin (see APFCON register).

# 16.0 5-BIT DIGITAL-TO-ANALOG CONVERTER (DAC) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 32 selectable output levels.

The positive input source (VSOURCE+) of the DAC can be connected to:

- External VREF+ pin
- VDD supply voltage

The negative input source (VSOURCE-) of the DAC can be connected to:

Vss

The output of the DAC (DACx\_output) can be selected as a reference voltage to the following:

- Comparator positive input
- · ADC input channel
- DACxOUT1 pin
- DACxOUT2 pin

The Digital-to-Analog Converter (DAC) can be enabled by setting the DACEN bit of the DACxCON0 register.



# FIGURE 16-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	110
APFCON	—	—	—	SSSEL	T1GSEL	—	CLC1SEL	NCO1SEL	107
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
OSCSTAT	SOSCR	—	OSTS	HFIOFR			LFIOFR	HFIOFS	60
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	SSP1IE —		TMR1IE	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	_	TMR2IF	TMR1IF	79
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Count								159*
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Count							159*	
TRISA	—	—	TRISA5	TRISA4	_(1)	_(1) TRISA2		TRISA0	109
T1CON	TMR1C	TMR1CS<1:0> T1CKPS<1:0>		S<1:0>	T10SCEN	T1SYNC	—	TMR10N	163
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>		164

#### TABLE 19-5: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER1

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the Timer1 module.

\* Page provides register information.

Note 1: Unimplemented, read as '1'.

The  $I^2C$  interface supports the following modes and features:

- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- · Limited Multi-master support
- · 7-bit and 10-bit addressing
- · Start and Stop interrupts
- Interrupt masking
- Clock stretching
- Bus collision detection
- General call address matching
- Address masking
- · Address Hold and Data Hold modes
- · Selectable SDAx hold times

Figure 21-2 is a block diagram of the  $I^2C$  interface module in Master mode. Figure 21-3 is a diagram of the  $I^2C$  interface module in Slave mode.

Note 1: In devices with more than one MSSP module, it is very important to pay close attention to SSPxCONx register names. SSPxCON1 and SSPxCON2 registers control different operational aspects of the same module, while SSPxCON1 and SSP2CON1 control the same features for two different modules.

2: Throughout this section, generic references to an MSSPx module in any of its operating modes may be interpreted as being equally applicable to MSSPx or MSSP2. Register names, module I/O signals, and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module when required.

### FIGURE 21-2: MSSPX BLOCK DIAGRAM (I<sup>2</sup>C<sup>™</sup> MASTER MODE)



# 21.2 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 (SCKx)
- Serial Data Out (SDOx)
- Serial Data In (SDIx)
- Slave Select (SSx)

Figure 21-1 shows the block diagram of the MSSP module when operating in SPI mode.

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

Figure 21-4 shows a typical connection between a master device and multiple slave devices.

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

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

Figure 21-5 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 SDOx output pin which is connected to, and received by, the slave's SDIx input pin. The slave device transmits information out on its SDOx output pin, which is connected to, and received by, the master's SDIx 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 SDOx 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 SDOx pin) and the master device is reading this bit and saving it as the LSb of its shift register.

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

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

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

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

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

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





#### 21.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN bit of the SSPxCON2 register. When this bit is set, the SCLx pin is pulled low and the contents of the Acknowledge data bit are presented on the SDAx pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCLx pin is deasserted (pulled high). When the SCLx pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCLx pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 21-30).

#### 21.6.8.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

# 21.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDAx pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN bit of the SSPxCON2 register. At the end of a receive/transmit, the SCLx line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDAx line low. When the SDAx line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCLx pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDAx pin will be deasserted. When the SDAx pin is sampled high while SCLx is high, the P bit of the SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 21-31).

# 21.6.9.1 WCOL Status Flag

If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

### FIGURE 21-30: ACKNOWLEDGE SEQUENCE WAVEFORM



- 22.1.2.8 Asynchronous Reception Set-up:
- Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section22.4 "EUSART Baud Rate Generator (BRG)").
- 2. Clear the ANSEL bit for the RX pin (if applicable).
- 3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
- 4. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 5. If 9-bit reception is desired, set the RX9 bit.
- 6. Enable reception by setting the CREN bit.
- 7. The RCIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
- 8. Read the RCSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
- 9. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register.
- 10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

22.1.2.9 9-bit Address Detection Mode Set-up

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

- Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see Section22.4 "EUSART Baud Rate Generator (BRG)").
- 2. Clear the ANSEL bit for the RX pin (if applicable).
- 3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
- If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 5. Enable 9-bit reception by setting the RX9 bit.
- 6. Enable address detection by setting the ADDEN bit.
- 7. Enable reception by setting the CREN bit.
- The RCIF interrupt flag bit will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
- 9. Read the RCSTA register to get the error flags. The ninth data bit will always be set.
- 10. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register. Software determines if this is the device's address.
- 11. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
- 12. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.



# FIGURE 22-5: ASYNCHRONOUS RECEPTION

# 22.5.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (Section22.5.1.5 "Synchronous Master Reception"), with the following exceptions:

- Sleep
- CREN bit is always set, therefore the receiver is never idle
- · SREN bit, which is a "don't care" in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RCREG register. If the RCIE enable bit is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will branch to the interrupt vector.

- 22.5.2.4 Synchronous Slave Reception Set-up:
- 1. Set the SYNC and SPEN bits and clear the CSRC bit.
- 2. Clear the ANSEL bit for both the CK and DT pins (if applicable).
- 3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- 4. If 9-bit reception is desired, set the RX9 bit.
- 5. Set the CREN bit to enable reception.
- The RCIF bit will be set when reception is complete. An interrupt will be generated if the RCIE bit was set.
- 7. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCSTA register.
- 8. Retrieve the eight Least Significant bits from the receive FIFO by reading the RCREG register.
- 9. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

# TABLE 22-10: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	235
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	75
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	76
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	79
RCREG	EUSART Receive Data Register								
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	234
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	113
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	233

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous slave reception.

\* Page provides register information.

# 24.1.5 CLCx SETUP STEPS

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

- Disable CLCx by clearing the LCxEN bit.
- Select desired inputs using CLCxSEL0 and CLCxSEL1 registers (See Table 24-1).
- Clear any associated ANSEL bits.
- Set all TRIS bits associated with inputs.
- · Clear all TRIS bits associated with outputs.
- Enable the chosen inputs through the four gates using CLCxGLS0, CLCxGLS1, CLCxGLS2, and CLCxGLS3 registers.
- Select the gate output polarities with the LCxPOLy bits of the CLCxPOL register.
- Select the desired logic function with the LCxMODE<2:0> bits of the CLCxCON register.
- Select the desired polarity of the logic output with the LCxPOL bit of the CLCxPOL register. (This step may be combined with the previous gate output polarity step).
- If driving a device, set the LCxOE bit in the CLCxCON register and also clear the TRIS bit corresponding to that output.
- If interrupts are desired, configure the following bits:
  - Set the LCxINTP bit in the CLCxCON register for rising event.
  - Set the LCxINTN bit in the CLCxCON register or falling event.
  - Set the CLCxIE bit of the associated PIE registers.
  - Set the GIE and PEIE bits of the INTCON register.
- Enable the CLCx by setting the LCxEN bit of the CLCxCON register.

# 24.2 CLCx Interrupts

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

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

To fully enable the interrupt, set the following bits:

- · LCxON bit of the CLCxCON register
- · CLCxIE bit of the associated PIE registers
- LCxINTP bit of the CLCxCON register (for a rising edge detection)
- LCxINTN bit of the CLCxCON register (for a falling edge detection)
- · PEIE and GIE bits of the INTCON register

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

# 24.3 Output Mirror Copies

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

# 24.4 Effects of a Reset

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

# 24.5 Operation During Sleep

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

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

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

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

#### FIGURE 26-1: SIMPLIFIED CWG BLOCK DIAGRAM



PIC16(L)F1508/9

DS40001609E-page 281

PIC16LF	1508/9	Standard Operating Conditions (unless otherwise stated)								
PIC16F1508/9										
Param.	Device	Min	Typt	Max	Unite		Conditions			
No.	Characteristics		וקעי	WIAN.	Units	Vdd	Note			
D016			215	360	μA	1.8	Fosc = 500 kHz,			
		_	275	480	μA	3.0	HFINTOSC			
D016			270	450	μA	2.3	Fosc = 500 kHz,			
			300	500	μA	3.0	HFINTOSC			
			350	620	μA	5.0				
D017*			410	660	μA	1.8	Fosc = 8 MHz,			
			630	970	μA	3.0	HFINTOSC			
D017*		_	530	750	μA	2.3	Fosc = 8 MHz,			
			660	1100	μA	3.0	HFINTOSC			
			730	1200	μA	5.0				
D018		_	600	940	μA	1.8	Fosc = 16 MHz,			
			970	1400	μA	3.0	HFINTOSC			
D018		_	780	1200	μA	2.3	Fosc = 16 MHz,			
			1000	1550	μA	3.0	HFINTOSC			
			1090	1700	μA	5.0				
D019A			1030	1500	μA	3.0	Fosc = 20 MHz,			
							External Clock (ECH), High-Power mode			
D019A			1060	1600	μA	3.0	Fosc = 20 MHz,			
		—	1220	1800	μA	5.0	External Clock (ECH), High-Power mode			

TABLE 29-2:	SUPPLY CURRENT	(IDD) <sup>(1,2)</sup> (	(CONTINUED)
-------------	----------------	--------------------------	-------------

These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to Vss; MCLR = VDD; WDT disabled.

- 2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.
- 3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula IR = VDD/2REXT (mA) with REXT in  $k\Omega$ .





FIGURE 30-18: IDD MAXIMUM, EXTERNAL CLOCK (ECH), HIGH-POWER MODE, PIC16F1508/9 ONLY

















