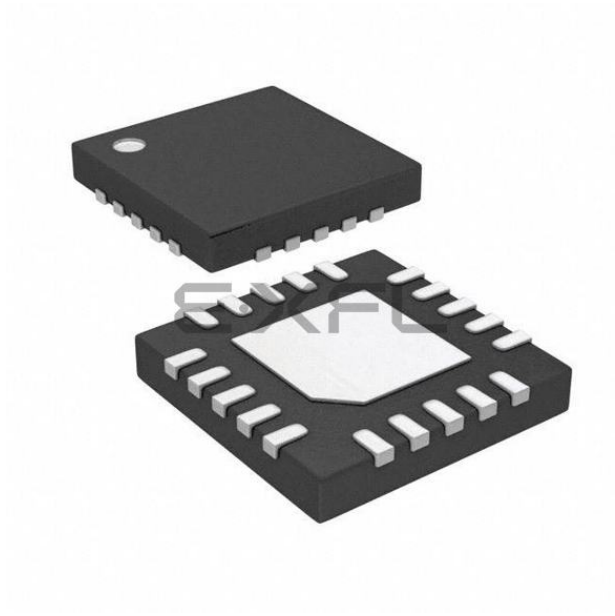


Welcome to E-XFL.COM

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	18
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 17x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-UFQFN Exposed Pad
Supplier Device Package	20-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f18345-e-gz

TABLE 1: 14/16-PIN ALLOCATION TABLE (PIC16(L)F18325) (CONTINUED)

I/O(2)	14-Pin PDIP/SOIC/TSSOP	16-Pin UQFN	ADC	Reference	Comparator	NCO	DAC	DSM	Timers	CCP	PWM	CWG	MSSP	EUSART	CLC	CLKR	Interrupt	Pull-up	Basic
OUT(2)	—	—	—	—	C1OUT	NCO1	—	DSM	TMR0	CCP1	PWM5	CWG1A CWG2A	SDA1(3) SDA2(3)	CK	CLC1OUT	CLKR	—	—	—
	—	—	—	—	C2OUT	—	—	—	—	CCP2	PWM6	CWG1B CWG2B	SCL1(3) SCL2(3)	DT	CLC2OUT	—	—	—	—
	—	—	—	—	—	—	—	—	—	CCP3	—	CWG1C CWG2C	SDO1 SDO2	TX	CLC3OUT	—	—	—	—
	—	—	—	—	—	—	—	—	—	CCP4	—	CWG1D CWG2D	SCK1 SCK2	—	CLC4OUT	—	—	—	—

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers.

3: These peripheral functions are bidirectional. The output pin selections must be the same as the input pin selections.

4: These pins are configured for I^2C logic levels; clock and data signals may be assigned to any of these pins. Assignments to the other pins (e.g., RA5) will operate, but logic levels will be standard TTL/ST as selected by the INLVL register.

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

Address	Name	PIC16(L)F18325	PIC16(L)F18345	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 1													
CPU CORE REGISTERS; see Table 4-2 for specifics													
08Ch	TRISA			—	—	TRISA5	TRISA4	—	TRISA2	TRISA1	TRISA0	--11 -111	--11 -111
08Dh	TRISB	X	—	Unimplemented								—	—
		—	X	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
08Eh	TRISC	X	—	—	—	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	--11 1111	--11 1111
		—	X	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
08Fh	—	—		Unimplemented								—	—
090h	PIE0			—	—	TMR0IE	IOCIE	—	—	—	INTE	--00 ---0	--00 ---0
091h	PIE1			TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	BCL1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
092h	PIE2			TMR6IE	C2IE	C1IE	NVMIE	SSP2IE	BCL2IE	TMR4IE	NCO1IE	0000 0000	0000 0000
093h	PIE3			OSFIE	CSWIE	TMR3GIE	TMR3IE	CLC4IE	CLC3IE	CLC2IE	CLC1IE	0000 0000	0000 0000
094h	PIE4			CWG2IE	CWG1IE	TMR5GIE	TMR5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE	0000 0000	0000 0000
095h	—	—		Unimplemented								—	—
096h	—	—		Unimplemented								—	—
097h	WDTCON			—	—	WDTPS<4:0>					SWDTEN	--01 0110	--01 0110
098h	—	—		Unimplemented								—	—
099h	—	—		Unimplemented								—	—
09Ah	—	—		Unimplemented								—	—
09Bh	ADRESL			ADRESL<7:0>								xxxx xxxx	uuuu uuuu
09Ch	ADRESH			ADRESH<7:0>								xxxx xxxx	uuuu uuuu
09Dh	ADCON0			CHS<5:0>						GO/$\overline{\text{DONE}}$	ADON	0000 0000	0000 0000
09Eh	ADCON1			ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		0000 -000	0000 -000
09Fh	ADACT			—	—	—	ADACT<4:0>					---0 0000	---0 0000

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 PIC16F18325/18345.

2: Register accessible from both User and ICD Debugger.

4.4 Stack

All devices have a 16-level x 15-bit wide hardware stack (refer to Figure 4-4 through Figure 4-7). The stack space is not part of either program or data space. The PC is PUSHed onto the stack when CALL or CALLW instructions are executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer and does not cause a Reset when either a Stack Overflow or Underflow occur if the STVREN bit is programmed to '0' (Configuration Words). This means that after the stack has been PUSHed sixteen times, the seventeenth PUSH overwrites the value that was stored from the first PUSH. The eighteenth PUSH overwrites the second PUSH (and so on). The STKOVF and STKUNF flag bits will be set on an Overflow/Underflow, regardless of whether the Reset is enabled.

If the STVREN bit in Configuration Words is programmed to '1', the device will be Reset if the stack is PUSHed beyond the sixteenth level or POPed beyond the first level, setting the appropriate bits (STKOVF or STKUNF, respectively) in the PCON register.

Note 1:	There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, CALLW, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

4.4.1 ACCESSING THE STACK

The stack is accessible through the TOSH, TOSL and STKPTR registers. STKPTR is the current value of the Stack Pointer. TOSH:TOSL register pair points to the TOP of the stack. Both registers are read/writable. TOS is split into TOSH and TOSL due to the 15-bit size of the PC. To access the stack, adjust the value of STKPTR, which will position TOSH:TOSL, then read/write to TOSH:TOSL. STKPTR is five bits to allow detection of overflow and underflow.

Note:	Care should be taken when modifying the STKPTR while interrupts are enabled.

During normal program operation, CALL, CALLW and Interrupts will increment STKPTR while RETLW, RETURN, and RETFIE will decrement STKPTR. At any time, STKPTR can be read to see how many levels remain available on the stack. The STKPTR always points at the currently used place on the stack. Therefore, a CALL or CALLW will increment the STKPTR and then write the PC, and a return will write the PC and then decrement the STKPTR.

Reference Figure 4-4 through Figure 4-7 for examples of accessing the stack.

FIGURE 4-6: ACCESSING THE STACK EXAMPLE 3

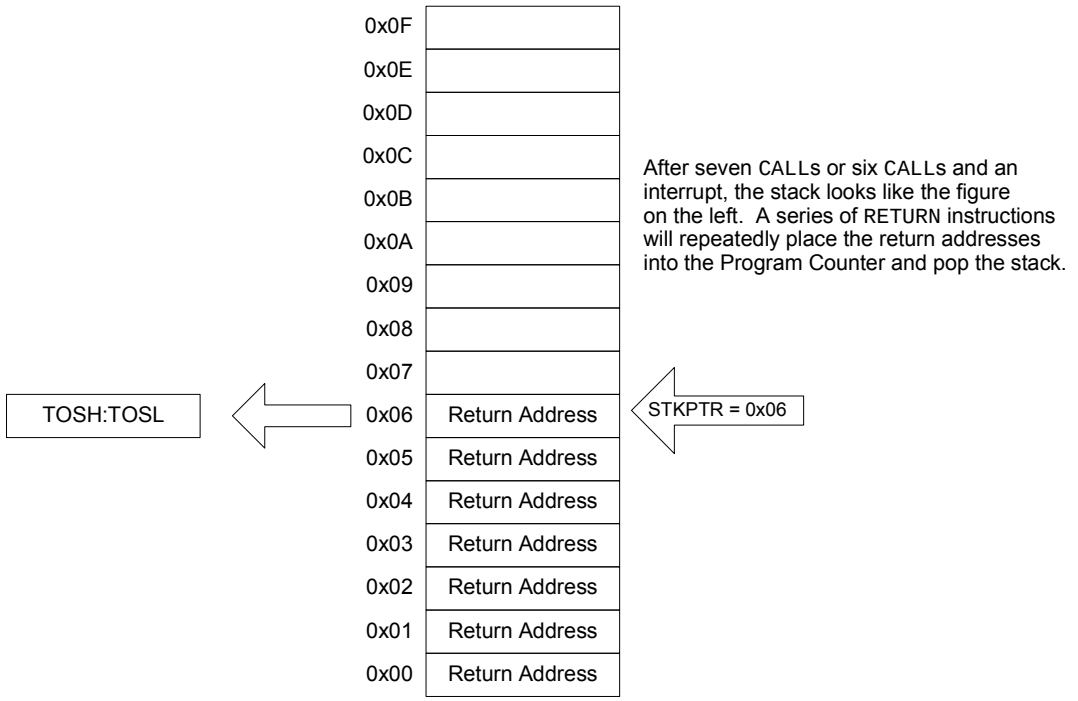
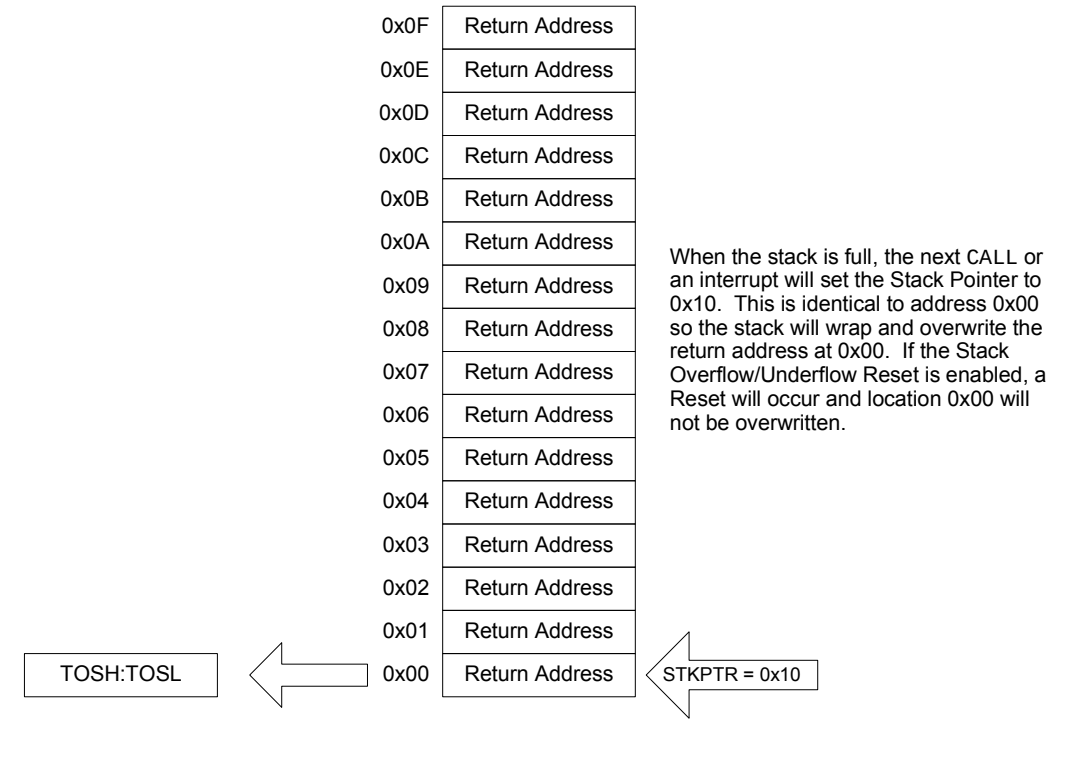


FIGURE 4-7: ACCESSING THE STACK EXAMPLE 4



5.3 Code Protection

Code protection allows the device to be protected from unauthorized access. Program memory protection and data memory are controlled independently. Internal access to the program memory is unaffected by any code protection setting.

5.3.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the $\overline{CP}$ bit in Configuration Words. When $\overline{CP}$ = 0, external reads and writes of program memory are inhibited and a read will return all '0's. The CPU can continue to read program memory, regardless of the protection bit settings. Self-write writing the program memory is dependent upon the write protection setting. See **Section 5.4 "Write Protection"** for more information.

5.3.2 DATA MEMORY PROTECTION

The entire data EEPROM is protected from external reads and writes by the $\overline{CPD}$ bit in the Configuration Words. When $\overline{CPD}$ = 0, external reads and writes of EEPROM memory are inhibited and a read will return all '0's. The CPU can continue to read and write EEPROM memory, regardless of the protection bit settings.

5.4 Write Protection

Write protection allows the device to be protected from unintended self-writes. Applications, such as boot loader software, can be protected while allowing other regions of the program memory to be modified.

The WRT<1:0> bits in Configuration Words define the size of the program memory block that is protected.

5.5 User ID

Four memory locations (8000h-8003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are readable and writable during normal execution. See **Section 11.4.7 "NVMREG EEPROM, User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations. For more information on checksum calculation, see the *"PIC16(L)F183XX Memory Programming Specification"* (DS40001738).

5.6 Device ID and Revision ID

The 14-bit device ID word is located at 8006h and the 14-bit revision ID is located at 8005h. These locations are read-only and cannot be erased or modified. See **Section 11.4.7 "NVMREG EEPROM, User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations.

Development tools, such as device programmers and debuggers, may be used to read the Device ID and Revision ID.

19.1.3 PWM RESOLUTION

PWM resolution, expressed in number of bits, defines the maximum number of discrete steps that can be present in a single PWM period. For example, a 10-bit resolution will result in 1024 discrete steps, whereas an 8-bit resolution will result in 256 discrete steps.

The maximum PWM resolution is ten bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 19-4.

EQUATION 19-4:

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

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

19.1.4 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the PWMx pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

19.1.5 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 7.0, Oscillator Module** for additional details.

19.1.6 EFFECTS OF RESET

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

19.1.7 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the module for using the PWMx outputs:

1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
2. Configure the PWM output polarity by configuring the PWMxPOL bit of the PWMxCON register.
3. Load the PR2 register with the PWM period value, as determined by Equation 19-1.
4. Load the PWMxDCH register and bits <7:6> of the PWMxDCL register with the PWM duty cycle value, as determined by Equation 19-2.
5. Configure and start Timer2:
   - Clear the TMR2IF interrupt flag bit of the PIR1 register.
   - Select the Timer2 prescale value by configuring the T2CKPS bit of the T2CON register.
   - Enable Timer2 by setting the TMR2ON bit of the T2CON register.
6. Wait until the TMR2IF is set.
7. When the TMR2IF flag bit is set:
   - Clear the associated TRIS bit(s) to enable the output driver.
   - Route the signal to the desired pin by configuring the RxyPPS register.
   - Enable the PWMx module by setting the PWMxEN bit of the PWMxCON register.

In order to send a complete duty cycle and period on the first PWM output, the above steps must be followed in the order given. If it is not critical to start with a complete PWM signal, then the PWM module can be enabled during Step 2 by setting the PWMxEN bit of the PWMxCON register.

21.1 CLCx Setup

Programming the CLCx module is performed by configuring the four stages in the logic signal flow. The four stages are:

- Data selection
- Data gating
- Logic function selection
- Output polarity

Each stage is setup at run time by writing to the corresponding CLCx Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

21.1.1 DATA SELECTION

There are 36 signals available as inputs to the configurable logic.

Data selection is through four multiplexers as indicated on the left side of Figure 21-2. Data inputs in the figure are identified by the 'LCx_in' signal name.

Table 21-1 correlates the input number to the actual signal for each CLC module. The column labeled 'LCxDyS<5:0> Value' indicates the MUX selection code for the selected data input. LCxDyS is an abbreviation to identify the specific input multiplexer: LCxD1S<5:0> through LCxD4S<5:0>.

Data inputs are selected with CLCxSEL0 through CLCxSEL3 registers (Register 21-3 through Register 21-6).

TABLE 21-1: CLCx DATA INPUT SELECTION

LCxDyS<5:0> Value	CLCx Input Source
100011 [35]	TMR6/PR6 match
100010 [34]	TMR5 overflow
100001 [33]	TMR4/PR4 match
100000 [32]	TMR3 overflow
11111 [31]	FOSC
11110 [30]	HFINTOSC
11101 [29]	LFINTOSC
11100 [28]	ADCRC
11011 [27]	IOCIF int flag bit
11010 [26]	TMR2/PR2 match
11001 [25]	TMR1 overflow
11000 [24]	TMR0 overflow
10111 [23]	EUSART1 (DT) output
10110 [22]	EUSART1 (TX/CK) output
10101 [21]	SDA2
10100 [20]	SCL2
10011 [19]	SDA1
10010 [18]	SCL1
10001 [17]	PWM6 output
10000 [16]	PWM5 output
01111 [15]	CCP4 output
01110 [14]	CCP3 output
01101 [13]	CCP2 output
01100 [12]	CCP1 output
01011 [11]	CLKR output
01010 [10]	DSM output
01001 [9]	C2 output
01000 [8]	C1 output
00111 [7]	CLC4 output
00110 [6]	CLC3 output
00101 [5]	CLC2 output
00100 [4]	CLC1 output
00011 [3]	CLCIN3PPS
00010 [2]	CLCIN2PPS
00001 [1]	CLCIN1PPS
00000 [0]	CLCIN0PPS

22.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADC Interrupt Flag is the ADIF bit in the PIR1 register. The ADC Interrupt Enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

Note 1:	The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.
2:	The ADC operates during Sleep only when the ADCRC oscillator is selected.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the ADIE bit of the PIE1 register and the PEIE bit of the INTCON register must both be set and the GIE bit of the INTCON register must be cleared. If all three of these bits are set, the execution will switch to the Interrupt Service Routine.

22.1.6 RESULT FORMATTING

The 10-bit ADC conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON1 register controls the output format.

Figure 22-3 shows the two output formats.

FIGURE 22-3: 10-BIT ADC CONVERSION RESULT FORMAT

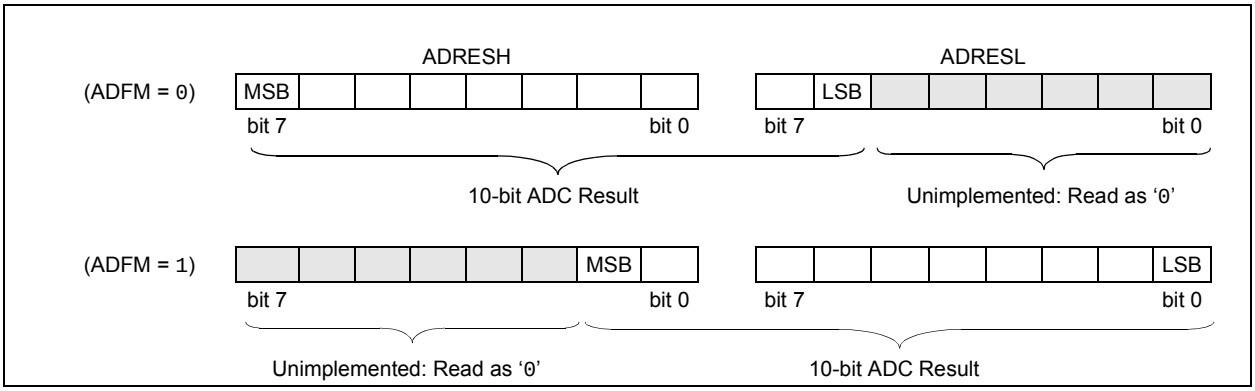
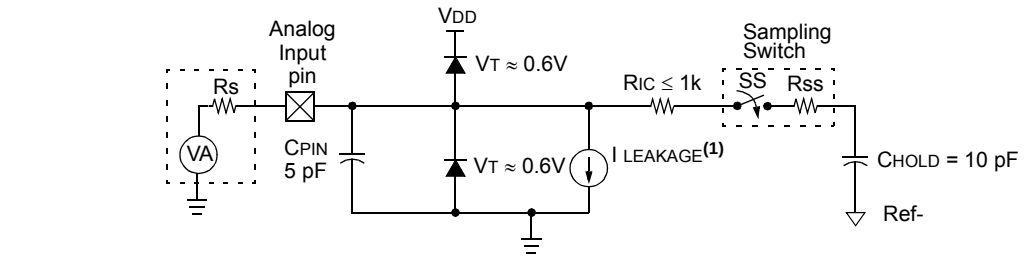
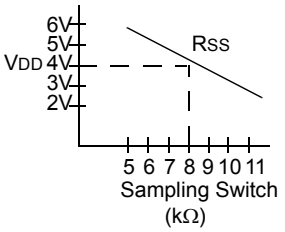


FIGURE 22-4: ANALOG INPUT MODEL



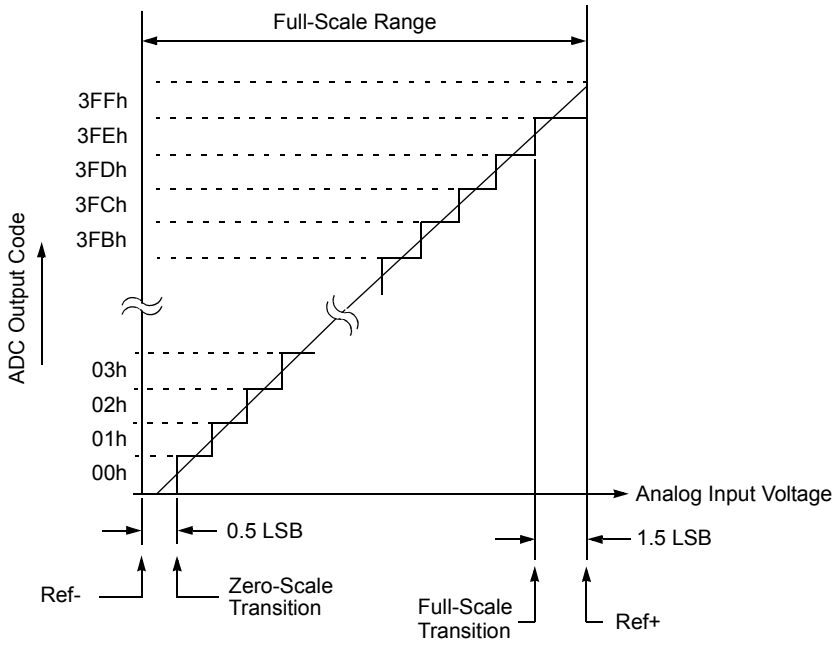


Legend:

CHOLD	= Sample/Hold Capacitance
CPIN	= Input Capacitance
I LEAKAGE	= Leakage current at the pin due to various junctions
RIC	= Interconnect Resistance
RSS	= Resistance of Sampling Switch
SS	= Sampling Switch
VT	= Threshold Voltage
RS	= External series resistance

Note 1: Refer to Table 35-4 (parameter D060).

FIGURE 22-5: ADC TRANSFER FUNCTION



23.2 Fixed Duty Cycle (FDC) Mode

In Fixed Duty Cycle (FDC) mode, every time the accumulator overflows (NCO_overflow), the output is toggled. This provides a 50% duty cycle with a constant frequency, provided that the increment value remains constant. The FDC frequency can be calculated using Equation 23-2. The FDC frequency is half of the overflow frequency since it takes two overflow events to generate one FDC clock period. For more information, see Figure 23-2.

EQUATION 23-2: FDC FREQUENCY

$$F_{fdc} = F_{overflow} / 2$$

The FDC mode is selected by clearing the N1PFM bit in the NCO1CON register.

23.3 Pulse Frequency (PF) Mode

In Pulse Frequency (PF) mode, every time the accumulator overflows (NCO_overflow), the output becomes active for one or more clock periods. Once the clock period expires, the output returns to an inactive state. This provides a pulsed output. The output becomes active on the rising clock edge immediately following the overflow event. For more information, see Figure 23-2.

The value of the active and inactive states depends on the polarity bit, N1POL, in the NCO1CON register.

The PF mode is selected by setting the N1PFM bit in the NCO1CON register.

23.3.1 OUTPUT PULSE WIDTH CONTROL

When operating in PF mode, the active state of the output can vary in width by multiple clock periods. Various pulse widths are selected with the N1PWS<2:0> bits in the NCO1CLK register.

When the selected pulse width is greater than the accumulator overflow time frame, the output of the NCO1 does not toggle.

23.4 Output Polarity Control

The last stage in the NCO1 module is the output polarity. The N1POL bit in the NCO1CON register selects the output polarity. Changing the polarity while the interrupts are enabled will cause an interrupt for the resulting output transition.

The NCO1 output can be used internally by source code or other peripherals. Accomplish this by reading the N1OUT (read-only) bit of the NCO1CON register.

The NCO1 output signal is available to the following peripherals:

- CWG

REGISTER 25-4: MDCARL: MODULATION LOW CARRIER CONTROL REGISTER

U-0	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	MDCLPOL	MDCLSYNC	—	MDCL<3:0>(1)			
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 **Unimplemented:** Read as '0'

bit 6 **MDCLPOL:** Modulator Low Carrier Polarity Select bit
- 1 = Selected low carrier signal is inverted
- 0 = Selected low carrier signal is not inverted

bit 5 **MDCLSYNC:** Modulator Low Carrier Synchronization Enable bit
- 1 = Modulator waits for a falling edge on the low time carrier signal before allowing a switch to the high time carrier
- 0 = Modulator output is not synchronized to the low time carrier signal(1)

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

bit 3-0 **MDCL<3:0>** Modulator Data High Carrier Selection bits (1)
- 1111 = CLC4 output
- 1110 = CLC3 output
- 1101 = CLC2 output
- 1100 = CLC1 output
- 1011 = HFINTOSC
- 1010 = FOSC
- 1001 = Reserved. No channel connected.
- 1000 = NCO1 output
- 0111 = PWM6 output
- 0110 = PWM5 output
- 0101 = CCP2 output (PWM Output mode only)
- 0100 = CCP1 output (PWM Output mode only)
- 0011 = Reference clock module signal (CLKR)
- 0010 = MDCIN2PPS
- 0001 = MDCIN1PPS
- 0000 = VSS

Note 1: Narrowed carrier pulse widths or spurs may occur in the signal stream if the carrier is not synchronized.

27.5.2 TIMER1 GATE SOURCE SELECTION

Timer1 gate source selections are shown in Table 27-4. Source selection is controlled by the T1GSS bits of the T1GCON register. The polarity of the selected input is configurable. Polarity selection is controlled by the T1GPOL bit of the T1GCON register.

TABLE 27-4: TIMER1 GATE SOURCES

T1GSS	Timer1 Gate Source
00	Timer1 Gate Pin
01	Overflow of Timer0 (TMR0 increments from FFh to 00h)
10	Comparator 1 Output (optionally Timer1 synchronized output)
11	Comparator 2 Output (optionally Timer1 synchronized output)

27.5.2.1 T1G Pin Gate Operation

The T1G pin is one source for Timer1 gate control. It can be used to supply an external source to the Timer1 gate circuitry.

27.5.2.2 Timer0 Overflow Gate Operation

When Timer0 increments from FFh to 00h, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 gate circuitry.

27.5.2.3 Comparator C1 Gate Operation

The output resulting from a Comparator 1 operation can be selected as a source for Timer1 gate control. The Comparator 1 output can be synchronized to the Timer1 clock or left asynchronous. For more information see **Section 18.4.1 "Comparator Output Synchronization"**.

27.5.2.4 Comparator C2 Gate Operation

The output resulting from a Comparator 2 operation can be selected as a source for Timer1 gate control. The Comparator 2 output can be synchronized to the Timer1 clock or left asynchronous. For more information see **Section 18.4.1 "Comparator Output Synchronization"**.

27.5.3 TIMER1 GATE TOGGLE MODE

When Timer1 Gate Toggle mode is enabled, it is possible to measure the full period of a Timer1 gate signal, as opposed to the duration of a single pulse.

The Timer1 gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See Figure 27-4 for timing details.

Timer1 Gate Toggle mode is enabled by setting the T1GTM bit of the T1GCON register. When the T1GTM bit is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.

Note:	Enabling Toggle mode at the same time as changing the gate polarity may result in indeterminate operation.

27.5.4 TIMER1 GATE SINGLE-PULSE MODE

When Timer1 Gate Single-Pulse mode is enabled, it is possible to capture a single-pulse gate event. Timer1 Gate Single-Pulse mode is first enabled by setting the T1GSPM bit in the T1GCON register. Next, the T1GGO/$\overline{\text{DONE}}$ bit in the T1GCON register must be set. The Timer1 will be fully enabled on the next incrementing edge of the Timer1 gate signal. On the next trailing edge of the Timer1 gate signal, the T1GGO/$\overline{\text{DONE}}$ bit will automatically be cleared. No other gate events will be allowed to increment Timer1 until the T1GGO/$\overline{\text{DONE}}$ bit is once again set in software. See Figure 27-5 for timing details.

If the Single-Pulse Gate mode is disabled by clearing the T1GSPM bit in the T1GCON register, the T1GGO/$\overline{\text{DONE}}$ bit should also be cleared.

Enabling the Toggle mode and the Single-Pulse mode simultaneously will permit both sections to work together. This allows the period of the Timer1 gate source to be measured. See Figure 27-6 for timing details.

27.5.5 TIMER1 GATE VALUE STATUS

When Timer1 Gate Value Status is utilized, it is possible to read the most current level of the gate control value. The value is stored in the T1GVAL bit in the T1GCON register. The T1GVAL bit is valid even when the Timer1 gate is not enabled (TMR1GE bit is cleared).

FIGURE 30-9: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

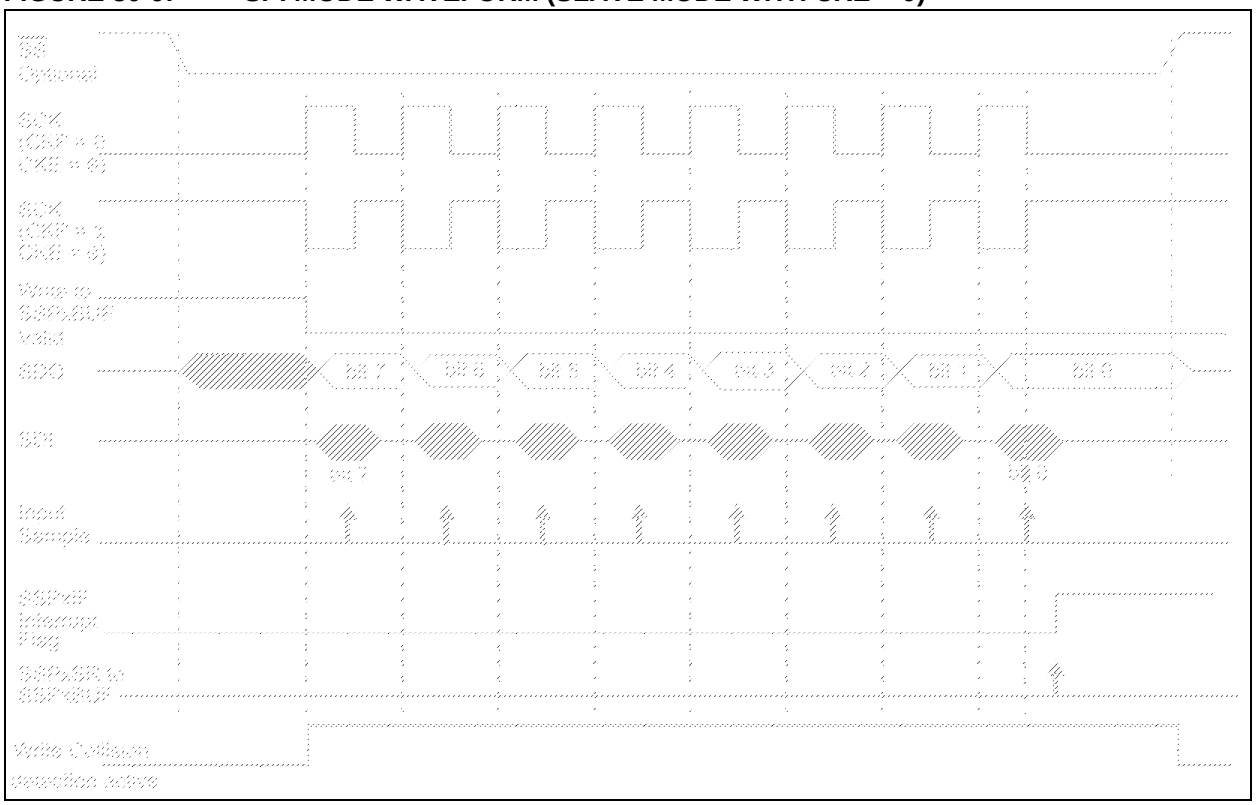
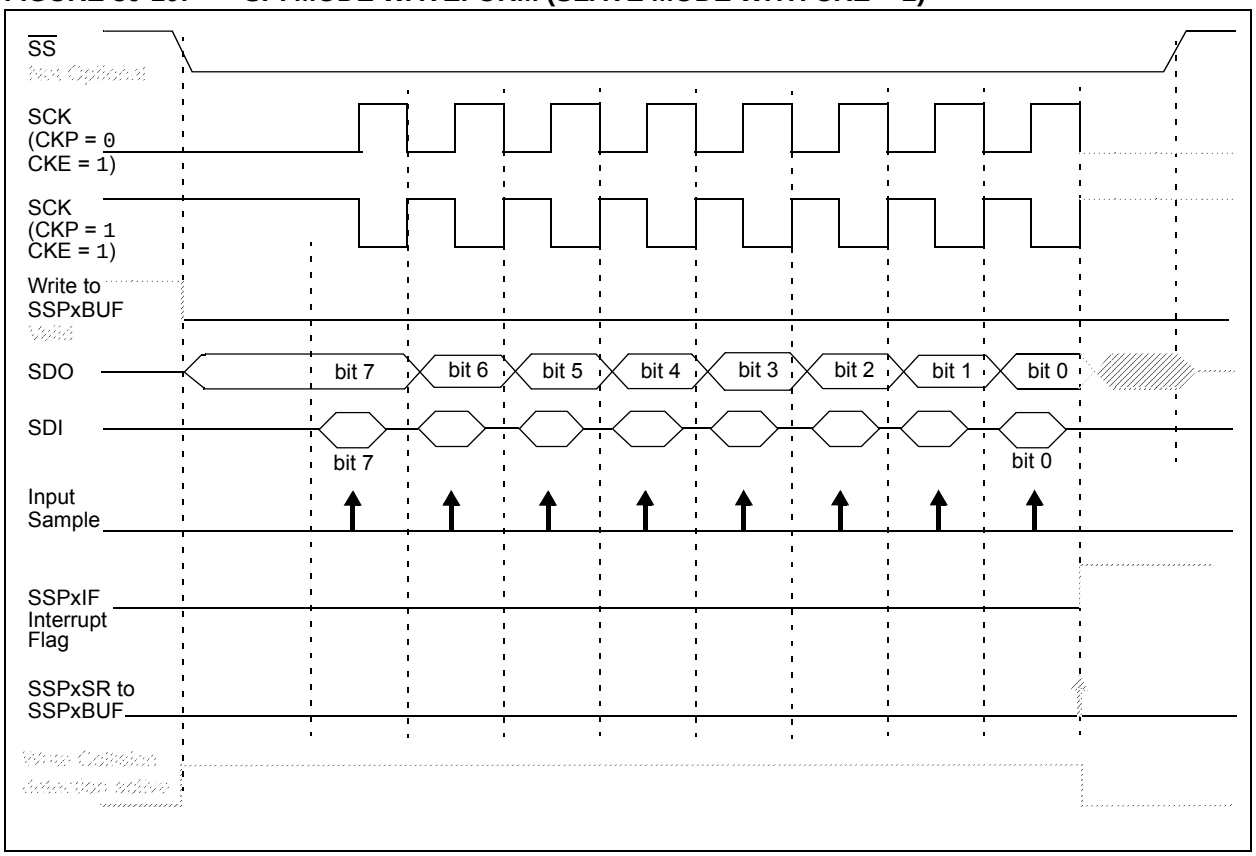


FIGURE 30-10: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



31.5 EUSART1 Operation During Sleep

The EUSART1 will remain active during Sleep only in the Synchronous Slave mode. All other modes require the system clock and therefore cannot generate the necessary signals to run the Transmit or Receive Shift registers during Sleep.

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

31.5.1 SYNCHRONOUS RECEIVE DURING SLEEP

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

- RC1STA and TX1STA Control registers must be configured for Synchronous Slave Reception (see **Section 31.4.2.4 "Synchronous Slave Reception Set-up"**).
- If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
- The RCIF interrupt flag must be cleared by reading RC1REG to unload any pending characters in the receive buffer.

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

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

31.5.2 SYNCHRONOUS TRANSMIT DURING SLEEP

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

- The RC1STA and TX1STA Control registers must be configured for synchronous slave transmission (see **Section 31.4.2.2 "Synchronous Slave Transmission Set-up"**).
- The TXIF interrupt flag must be cleared by writing the output data to the TX1REG, thereby filling the TSR and transmit buffer.
- If interrupts are desired, set the TXIE bit of the PIE1 register and the PEIE bit of the INTCON register.
- Interrupt enable bits TXIE of the PIE1 register and PEIE of the INTCON register must set.

Upon entering Sleep mode, the device will be ready to accept clocks on TX/CK pin and transmit data on the RX/DT pin. When the data word in the TSR has been completely clocked out by the external device, the pending byte in the TX1REG will transfer to the TSR and the TXIF flag will be set. Thereby, waking the processor from Sleep. At this point, the TX1REG is available to accept another character for transmission, which will clear the TXIF flag.

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

32.0 REFERENCE CLOCK OUTPUT MODULE

The Reference Clock Output module provides the ability to send a clock signal to the clock reference output pin (CLKR). The Reference Clock Output can also be used as a signal for other peripherals, such as the Data Signal Modulator (DSM).

The Reference Clock Output module has the following features:

- System clock is the module source clock
- Programmable clock divider
- Selectable duty cycle

32.1 Clock Source

The Reference Clock Output module uses the system clock (FOSC) as the clock source. Any device clock switching will be reflected in the clock output.

32.1.1 CLOCK SYNCHRONIZATION

Once the reference clock enable (CLKREN) is set, the module is ensured to be glitch-free at start-up.

When the Reference Clock Output is disabled, the output signal will be disabled immediately.

Clock dividers and clock duty cycles can be changed while the module is enabled, but glitches may occur on the output. To avoid possible glitches, clock dividers and clock duty cycles should be changed only when the CLKREN is clear.

32.2 Programmable Clock Divider

The module takes the system clock input and divides it based on the value of the CLKRDIV<2:0> bits of the CLKRCON register (Register 32-1).

The following configurations can be made based on the CLKRDIV<2:0> bits:

- Base FOSC value
- FOSC divided by 2
- FOSC divided by 4
- FOSC divided by 8
- FOSC divided by 16
- FOSC divided by 32
- FOSC divided by 64
- FOSC divided by 128

The clock divider values can be changed while the module is enabled; however, in order to prevent glitches on the output, the CLKRDIV<2:0> bits should only be changed when the module is disabled (CLKREN = 0).

32.3 Selectable Duty Cycle

The CLKRDC<1:0> bits of the CLKRCON register can be used to modify the duty cycle of the output clock. A duty cycle of 25%, 50%, or 75% can be selected for all clock rates, with the exception of the undivided base FOSC value.

The duty cycle can be changed while the module is enabled; however, in order to prevent glitches on the output, the CLKRDC<1:0> bits should only be changed when the module is disabled (CLKREN = 0).

Note:	The CLKRDC1 bit is reset to '1'. This makes the default duty cycle 50%.

32.4 Operation in Sleep Mode

The Reference Clock Output module clock is based on the system clock. When the device goes to Sleep, the module outputs will remain in their current state. This will have a direct effect on peripherals using the Reference Clock Output as an input signal.

34.2 Instruction Descriptions

ADDFSR — Add Literal to FSRn

Syntax:	[*label*] ADDFSR FSRn, k
Operands:	-32 ≤ k ≤ 31 n ∈ [0, 1]
Operation:	FSR(n) + k → FSR(n)
Status Affected:	None
Description:	The signed 6-bit literal 'k' is added to the contents of the FSRnH:FSRnL register pair. FSRn is limited to the range 0000h-FFFFh. Moving beyond these bounds will cause the FSR to wrap-around.

ADDLW — Add literal and W

Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the 8-bit literal 'k' and the result is placed in the W register.

ADDWF — Add W and f

Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with 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'.

ADDWFC — ADD W and CARRY bit to f

Syntax:	[*label*] ADDWFC f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) + (C) → dest
Status Affected:	C, DC, Z
Description:	Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.

ANDLW — AND literal with W

Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register.

ANDWF — AND W with f

Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with 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'.

ASRF — Arithmetic Right Shift

Syntax:	[*label*] ASRF f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<7>)→ dest<7> (f<7:1>) → dest<6:0>, (f<0>) → C,
Status Affected:	C, Z
Description:	The contents of register 'f' are shifted one bit to the right through the Carry flag. The MSb remains unchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

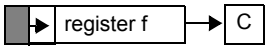
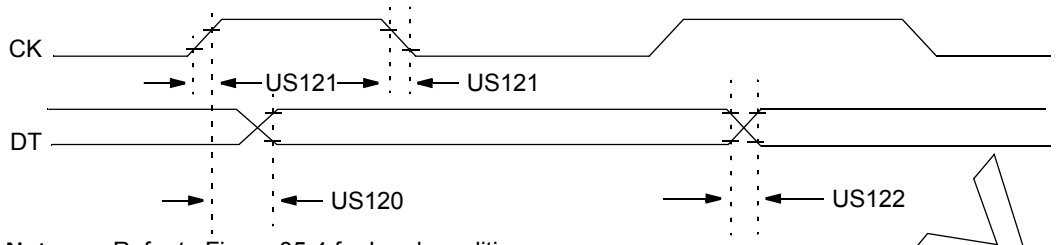


FIGURE 35-15: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING

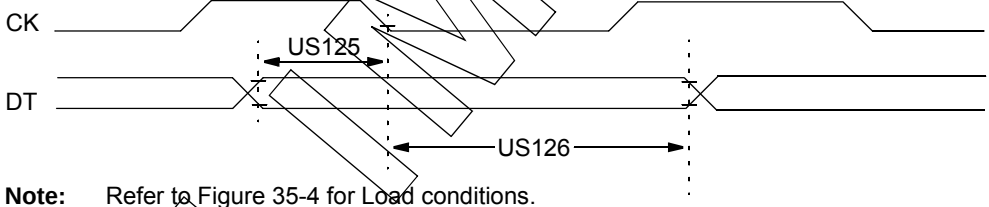


Note: Refer to Figure 35-4 for Load conditions.

TABLE 35-20: EUSART SYNCHRONOUS TRANSMISSION CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)						
Param. No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
US120	TCKH2DTV	SYNC XMIT (Master and Slave) Clock high to data-out valid	—	80	ns	3.0V ≤ VDD ≤ 5.5V
			—	100	ns	1.8V ≤ VDD ≤ 5.5V
US121	TCKRF	Clock out rise time and fall time (Master mode)	—	45	ns	3.0V ≤ VDD ≤ 5.5V
			—	50	ns	1.8V ≤ VDD ≤ 5.5V
US122	TDTRF	Data-out rise time and fall time	—	45	ns	3.0V ≤ VDD ≤ 5.5V
			—	50	ns	1.8V ≤ VDD ≤ 5.5V

FIGURE 35-16: EUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING

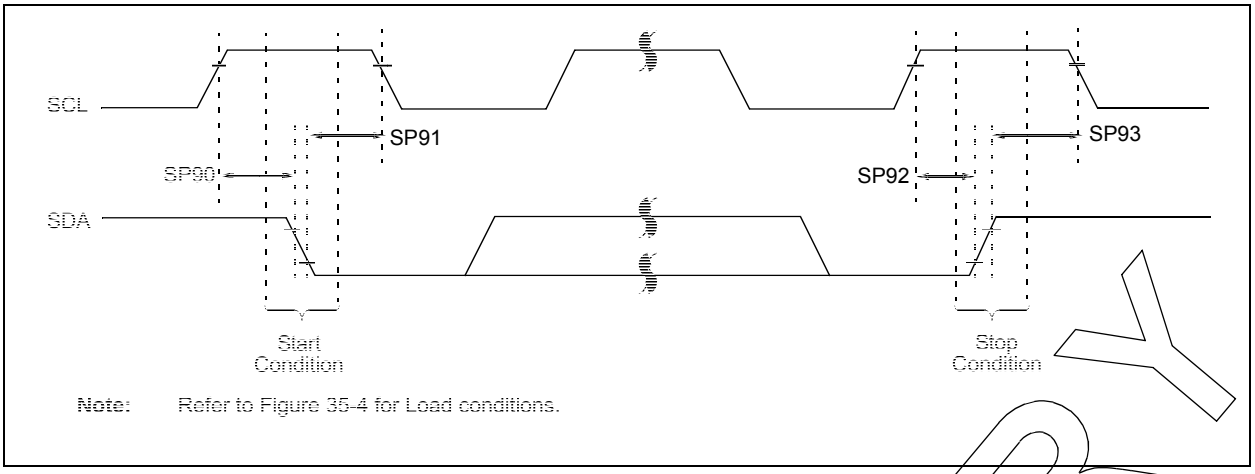


Note: Refer to Figure 35-4 for Load conditions.

TABLE 35-21: EUSART SYNCHRONOUS RECEIVE CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)						
Param. No.	**Symbol**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
US125	TDTV2CKL	SYNC RCV (Master and Slave) Data-setup before CK ↓ (DT hold time)	10	—	ns	
US126	TCKL2DTL	Data-hold after CK ↓ (DT hold time)	15	—	ns	

FIGURE 35-21: I²C BUS START/STOP BITS TIMING



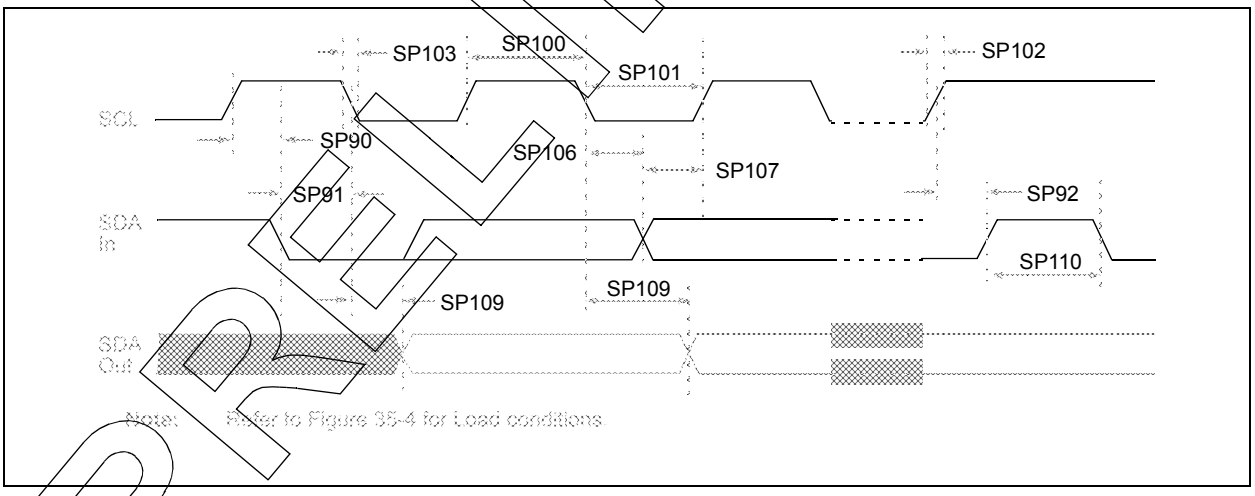
Note: Refer to Figure 35-4 for Load conditions.

TABLE 35-23: I²C BUS START/STOP BITS CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)								
Param. No.	**Symbol**	**Characteristic**		**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
SP90*	TSU:STA	Start condition Setup time	100 kHz mode	4700	—	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	600	—	—		
SP91*	THD:STA	Start condition Hold time	100 kHz mode	4000	—	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—	—		
SP92*	TSU:STO	Stop condition Setup time	100 kHz mode	4700	—	—	ns	
			400 kHz mode	600	—	—		
SP93	THD:STO	Stop condition Hold time	100 kHz mode	4000	—	—	ns	
			400 kHz mode	600	—	—		

* These parameters are characterized but not tested.

FIGURE 35-22: I²C BUS DATA TIMING



Note: Refer to Figure 35-4 for Load conditions.

37.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

37.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradeable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

37.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

37.9 PICkit 3 In-Circuit Debugger/ Programmer

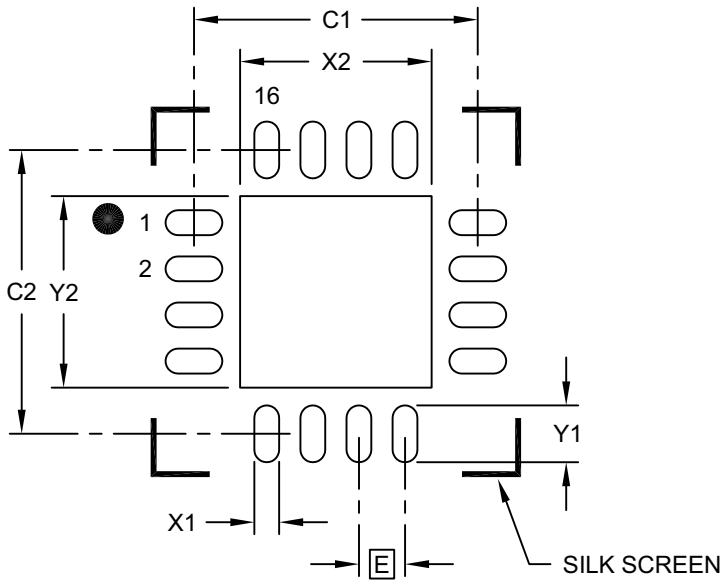
The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

37.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

16-Lead Ultra Thin Plastic Quad Flat, No Lead Package (JQ) - 4x4x0.5 mm Body [UQFN]

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	
Optional Center Pad Width	X2			2.70
Optional Center Pad Length	Y2			2.70
Contact Pad Spacing	C1		4.00	
Contact Pad Spacing	C2		4.00	
Contact Pad Width (X16)	X1			0.35
Contact Pad Length (X16)	Y1			0.80

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing C04-2257A