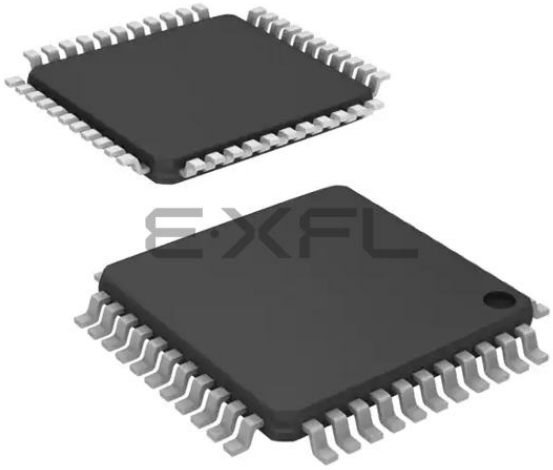


Welcome to E-XFL.COM

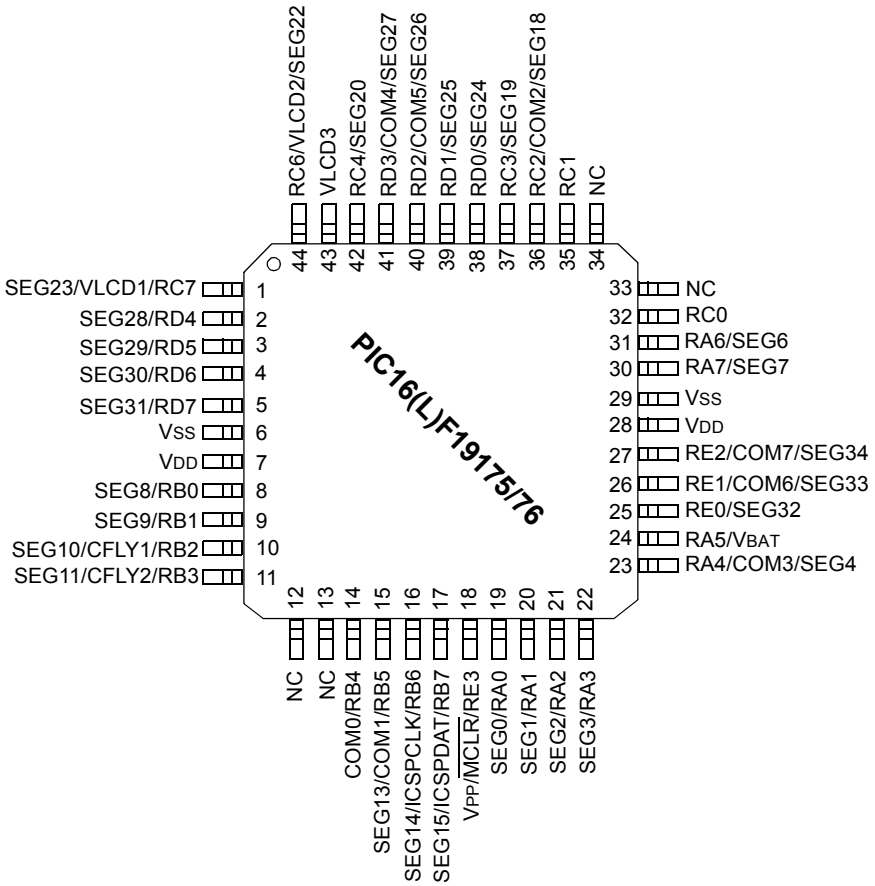
What is "Embedded - Microcontrollers"?

"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, LCD, POR, PWM, WDT
Number of I/O	35
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 31x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f19175-i-pt

FIGURE 5: 44-PIN TQFP PIN DIAGRAM FOR PIC16(L)F19175/76



Note: See Table 4 for the pin allocation table.

TABLE 4-12: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 PIC16(L)F19155/56/75/76/85/86 (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR
Bank 12											
CPU CORE REGISTERS; see Table 4-3 for specifics											
60Ch	CWG1CLKCON	—	—	—	—	—	—	—	CS	0000 0000	0000 0000
60Dh	CWG1ISM	—	—	—	—	IS<3:0>				0000 0000	0000 0000
60Eh	CWG1DBR	—	—	DBR<5:0>						0000 0000	0000 0000
60Fh	CWG1DBF	—	—	DBF<5:0>						0000 0000	0000 0000
610h	CWG1CON0	EN	LD	—	—	—	MODE<2:0>			00-- -000	00-- -000
611h	CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	--x- 0000	--u- 0000
612h	CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	0001 01--	0001 01--
613h	CWG1AS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	0000 0000	---u 0000
614h	CWG1STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	0000 0000	0000 0000
615h	—	Unimplemented								---- ----	---- ----
616h	—	Unimplemented								---- ----	---- ----
617h	—	Unimplemented								---- ----	---- ----
618h	—	Unimplemented								---- ----	---- ----
619h	—	Unimplemented								---- ----	---- ----
61Ah	—	Unimplemented								---- ----	---- ----
61Bh	—	Unimplemented								---- ----	---- ----
61Ch	—	Unimplemented								---- ----	---- ----
61Dh	—	Unimplemented								---- ----	---- ----
61Eh	—	Unimplemented								---- ----	---- ----
61Fh	—	Unimplemented								---- ----	---- ----

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

Note 1: Unimplemented data memory locations, read as '0'.

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.

Development tools, such as device programmers and debuggers, may be used to read the Device ID, Revision ID and Configuration Words. These locations can also be read from the NVMCON register.

5.7 Register Definitions: Device and Revision

REGISTER 5-6: DEVID: DEVICE ID REGISTER

R	R	R	R	R	R
DEV<13:8>					
bit 13					bit 8

R	R	R	R	R	R	R	R
DEV<7:0>							
bit 7							bit 0

Legend:	
R = Readable bit	
'1' = Bit is set	'0' = Bit is cleared

bit 13-0 **DEV<13:0>:** Device ID bits

Device	DEVID<13:0> Values
PIC16F19155/56	11 0000 1001 1110 (309Eh)
PIC16LF19155/56	11 0000 1001 1111 (309Fh)
PIC16F19175/76	11 0000 1010 0000 (30A0h)
PIC16LF19175/76	11 0000 1010 0001 (30A1h)
PIC16F19185/86	11 0000 1010 0010 (30A2h)
PIC16LF19185/86	11 0000 1010 0011 (30A3h)

6.1 Microchip Unique identifier (MUI)

The PIC16(L)F19155/56/75/76/85/86 devices are individually encoded during final manufacturing with a Microchip Unique Identifier, or MUI. The MUI cannot be erased by a Bulk Erase command or any other user-accessible means. This feature allows for manufacturing traceability of Microchip Technology devices in applications where this is required. It may also be used by the application manufacturer for a number of functions that require unverified unique identification, such as:

- Tracking the device
- Unique serial number

The MUI consists of nine program words. When taken together, these fields form a unique identifier. The MUI is stored in nine read-only locations, located between 8100h to 8109h in the DIA space. Table 6-1 lists the addresses of the identifier words.

Note:	For applications that require verified unique identification, contact the local Microchip Technology sales office to create a Serialized Quick Turn Programming (SQTP^{sm}) option.

6.2 External Unique Identifier (EUI)

The EUI data is stored at locations 810Ah to 8111h in the program memory region. This region is an optional space for placing application specific information. The data is coded per customer requirements during manufacturing.

Note:	Data is stored in this address range on receiving a request from the customer. The customer may contact the local sales representative, or Field Applications Engineer, and provide them the unique identifier information that is supposed to be stored in this region.

6.3 Analog-to-Digital Conversion Data of the Temperature Indicator

The purpose of the temperature indicator module is to provide a temperature-dependent voltage that can be measured by an analog module, see **Section 20.0 "Temperature Indicator Module (TIM)"**.

The DIA table contains the internal ADC measurement values of the temperature indicator for low and high range at fixed points of reference. The values are measured during test and are unique to each device. The measurement data is stored in the DIA memory region as hexadecimal numbers corresponding to the ADC conversion result. The calibration data can be used to plot the approximate sensor output voltage, V_{TSENSE} vs. Temperature curve without having to make calibration measurements in the application. For more information on the operation of the temperature indicator, refer to **Section 20.0 "Temperature Indicator Module (TIM)"**.

- **TSLR1-TSLR3**: Address 8112h to 8114h store the measurements for the low-range setting of the temperature indicator at V_{DD} = 3V.
- **TSHR1-TSHR3**: Address 8115h to 8117h store the measurements for the high-range setting of the temperature indicator at V_{DD} = 3V.
- The stored measurements are made by the device ADC using the internal V_{REF} = 2.048V.

6.4 Fixed Voltage Reference Data

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of V_{DD}, with 1.024V, 2.048V or 4.096V selectable output levels. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- ADC positive reference
- Comparator positive input
- Digital-to-Analog Converter

For more information on the FVR, refer to **Section 18.0 "Fixed Voltage Reference (FVR)"**.

The DIA stores measured FVR voltages for this device in mV for the different buffer settings of 1x, 2x or 4x at program memory locations 8118h to 811Dh.

- FVRA1X stores the value of ADC FVR1 Output voltage for 1x setting (in mV)
- FVRA2X stores the value of ADC FVR1 Output Voltage for 2x setting (in mV)
- FVRA4X stores the value of ADC FVR1 Output Voltage for 4x setting (in mV)
- FVRC1X stores the value of Comparator FVR2 output voltage for 1x setting (in mV)
- FVRC2X stores the value of Comparator FVR2 output voltage for 2x setting (in mV)
- FVRC4X stores the value of Comparator FVR2 output voltage for 4x setting (in mV)

REGISTER 10-16: PIR5: PERIPHERAL INTERRUPT REQUEST REGISTER 5

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	U-0	U-0	U-0	R/W/HS-0/0
CLC4IF	CLC3IF	CLC2IF	CLC1IF	—	—	—	TMR1GIF
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	HS = Hardware set

bit 7 **CLC4IF:** CLC4 Interrupt Flag bit
1 = A CLC4OUT interrupt condition has occurred (must be cleared in software)
0 = No CLC4 interrupt event has occurred

bit 6 **CLC3IF:** CLC3 Interrupt Flag bit
1 = A CLC3OUT interrupt condition has occurred (must be cleared in software)
0 = No CLC3 interrupt event has occurred

bit 5 **CLC2IF:** CLC2 Interrupt Flag bit
1 = A CLC2OUT interrupt condition has occurred (must be cleared in software)
0 = No CLC2 interrupt event has occurred

bit 4 **CLC1IF:** CLC1 Interrupt Flag bit
1 = A CLC1OUT interrupt condition has occurred (must be cleared in software)
0 = No CLC1 interrupt event has occurred

bit 3-1 **Unimplemented:** Read as '0'

bit 0 **TMR1GIF:** Timer1 Gate Interrupt Flag bit
1 = The Timer1 Gate has gone inactive (the acquisition is complete)
0 = The Timer1 Gate has not gone inactive

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 13-6: NVMCON2: NONVOLATILE MEMORY CONTROL 2 REGISTER

W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0
NVMCON2<7:0>							
bit 7							bit 0

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

bit 7-0 **NVMCON2<7:0>:** Flash Memory Unlock Pattern bits

To unlock writes, a 55h must be written first followed by an AAh before setting the WR bit of the NVMCON1 register. The value written to this register is used to unlock the writes.

TABLE 13-5: SUMMARY OF REGISTERS ASSOCIATED WITH NONVOLATILE MEMORY (NVM)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	164
PIE7	—	—	NVMIE	—	—	—	—	CWG1IE	172
PIR7	—	—	NVMIF	—	—	—	—	CWG1IF	181
NVMCON1	—	NVMREGS	LWLO	FREE	WRERR	WREN	WR	RD	217
NVMCON2	NVMCON2<7:0>								218
NVMADRL	NVMADR<7:0>								216
NVMADRH	—(1)	NVMADR<14:8>							216
NVMDATL	NVMDAT<7:0>								216
NVMDATH	—	—	NVMDAT<13:8>						216

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

Note 1: Unimplemented, read as '1'.

REGISTER 14-7: SLRCONA: PORTA SLEW RATE CONTROL REGISTER

R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
SLRA7	SLRA6	—	SLRA4	SLRA3	SLRA2	SLRA1	SLRA0
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 7-6 **SLRA<7:6>:** PORTA Slew Rate Enable bits
For RA<7:6> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

bit 5 **Unimplemented:** Read as ‘0’.

bit 4-0 **SLRA<4:0>:** PORTA Slew Rate Enable bits
For RA<4:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

REGISTER 14-8: INLVLA: PORTA INPUT LEVEL CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
INLVLA7	INLVLA6	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0
bit 7							bit 0

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

bit 7-0 **INLVLA<7:0>:** PORTA Input Level Select bits
For RA<7:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

REGISTER 16-5: PMD4: PMD CONTROL REGISTER 4

R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
UART2MD	UART1MD	—	MSSP1MD	—	—	—	CWG1MD
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 **UART2MD:** Disable EUSART2 bit
1 = EUSART2 module disabled
0 = EUSART2 module enabled

bit 6 **UART1MD:** Disable EUSART1 bit
1 = EUSART1 module disabled
0 = EUSART1 module enabled

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

bit 4 **MSSP1MD:** Disable MSSP1 bit
1 = MSSP1 module disabled
0 = MSSP1 module enabled

bit 3-1 **Unimplemented:** Read as '0'

bit 0 **CWG1MD:** Disable CWG1 bit
1 = CWG1 module disabled
0 = CWG1 module enabled

19.4.2 PRECHARGE CONTROL

The Precharge stage is an optional period of time that brings the external channel and internal sample and hold capacitor to known voltage levels. Precharge is enabled by writing a non-zero value to the PRE register. This stage is initiated when an ADC conversion begins, either from setting the GO bit, a special event trigger, or a conversion restart from the computation functionality. If the PRE register is cleared when an ADC conversion begins, this stage is skipped.

During the precharge time, CHOLD is disconnected from the outer portion of the sample path that leads to the external capacitive sensor and is connected to either VDD or VSS, depending on the value of the ADPPOL bit of ADCON1. At the same time, the port pin logic of the selected analog channel is overridden to drive a digital high or low out, in order to precharge the outer portion of the ADC's sample path, which includes the external sensor. The output polarity of this override is also determined by the ADPPOL bit of ADCON1. The amount of time that this charging receives is controlled by the PRE register.

Note 1:	The external charging overrides the TRIS setting of the respective I/O pin.
2:	If there is a device attached to this pin, Precharge should not be used.

19.4.3 ACQUISITION CONTROL

The Acquisition stage is an optional time for the voltage on the internal sample and hold capacitor to charge or discharge from the selected analog channel. This acquisition time is controlled by the ADACQ register. If PRE = 0, acquisition starts at the beginning of conversion. When PRE = 1, the acquisition stage begins when precharge ends.

At the start of the acquisition stage, the port pin logic of the selected analog channel is overridden to turn off the digital high/low output drivers so they do not affect the final result of the charge averaging. Also, the selected ADC channel is connected to CHOLD. This allows charge averaging to proceed between the precharged channel and the CHOLD capacitor.

Note:	When PRE! = 0, acquisition time cannot be '0'. In this case, setting ADACQ to '0' will set a maximum acquisition time (8191 ADC clock cycles). When precharge is disabled, setting ADACQ to '0' will disable hardware acquisition time control.

19.4.4 GUARD RING OUTPUTS

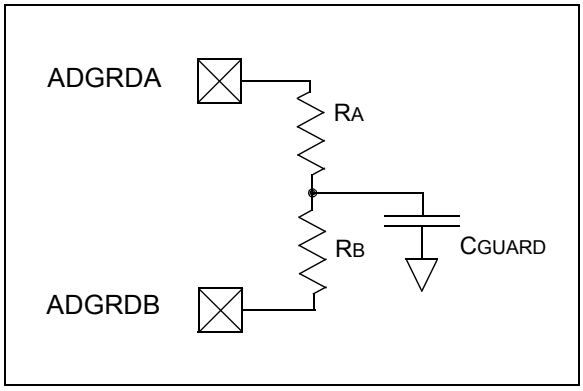
Figure 19-8 shows a typical guard ring circuit. CGUARD represents the capacitance of the guard ring trace placed on the PCB board. The user selects values for RA and RB that will create a voltage profile on CGUARD, which will match the selected acquisition channel.

The purpose of the guard ring is to generate a signal in phase with the CVD sensing signal to minimize the effects of the parasitic capacitance on sensing electrodes. It also can be used as a mutual drive for mutual capacitive sensing. For more information about active guard and mutual drive, see Application Note AN1478, "*mTouch™ Sensing Solution Acquisition Methods Capacitive Voltage Divider*" (DS01478).

The ADC has two guard ring drive outputs, ADGRDA and ADGRDB. These outputs can be routed through PPS controls to I/O pins (see **Section 15.0 "Peripheral Pin Select (PPS) Module"** for details) and the polarity of these outputs are controlled by the ADGPOL and ADIPEN bits of ADCON1.

At the start of the first precharge stage, both outputs are set to match the ADGPOL bit of ADCON1. Once the acquisition stage begins, ADGRDA changes polarity, while ADGRDB remains unchanged. When performing a double sample conversion, setting the ADIPEN bit of ADCON1 causes both guard ring outputs to transition to the opposite polarity of ADGPOL at the start of the second precharge stage, and ADGRDA toggles again for the second acquisition. For more information on the timing of the guard ring output, refer to Figure 19-8 and Figure 19-9.

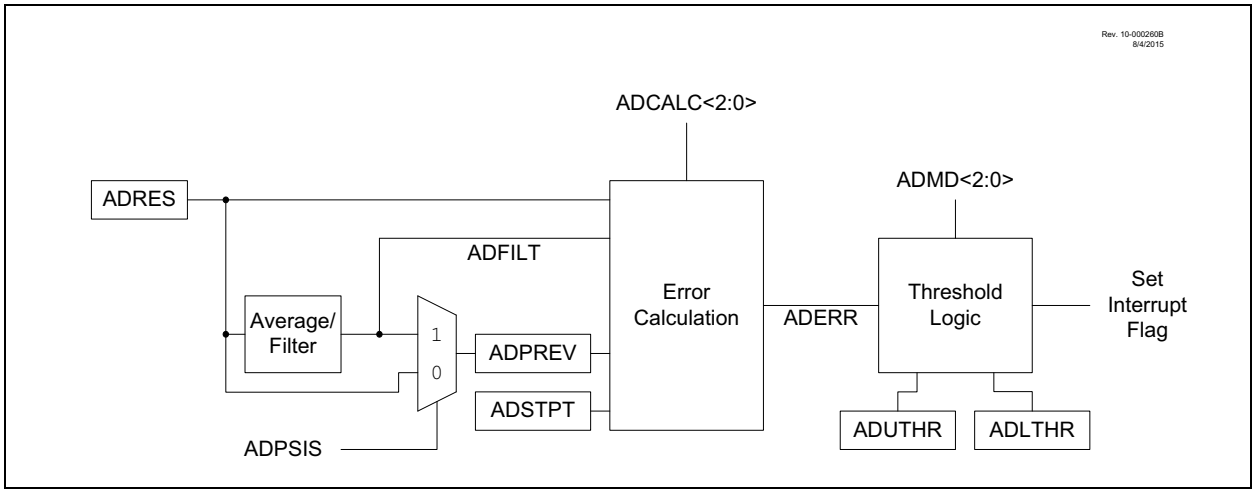
FIGURE 19-8: GUARD RING CIRCUIT



19.5 Computation Operation

The ADC module hardware is equipped with post conversion computation features. These features provide data post-processing functions that can be operated on the ADC conversion result, including digital filtering/averaging and threshold comparison functions.

FIGURE 19-10: COMPUTATIONAL FEATURES SIMPLIFIED BLOCK DIAGRAM



The operation of the ADC computational features is controlled by MD <2:0> bits in the ADCON2 register.

The module can be operated in one of five modes:

• **Basic**: In this mode, ADC conversion occurs on single (ADDSEN = 0) or double (ADDSEN = 1) samples. ADIF is set after all the conversion are complete.

• **Accumulate**: With each trigger, the ADC conversion result is added to accumulator and CNT increments. ADIF is set after each conversion. ADTIF is set according to the Calculation mode.

• **Average**: With each trigger, the ADC conversion result is added to the accumulator. When the RPT number of samples have been accumulated, a threshold test is performed. Upon the next trigger, the accumulator is cleared. For the subsequent tests, additional RPT samples are required to be accumulated.

• **Burst Average**: At the trigger, the accumulator is cleared. The ADC conversion results are then collected repetitively until RPT samples are accumulated and finally the threshold is tested.

• **Low-Pass Filter (LPF)**: With each trigger, the ADC conversion result is sent through a filter. When RPT samples have occurred, a threshold test is performed. Every trigger after that the ADC conversion result is sent through the filter and another threshold test is performed.

The five modes are summarized in Table 19-2 below.

26.0 TIMER1 MODULE WITH GATE CONTROL

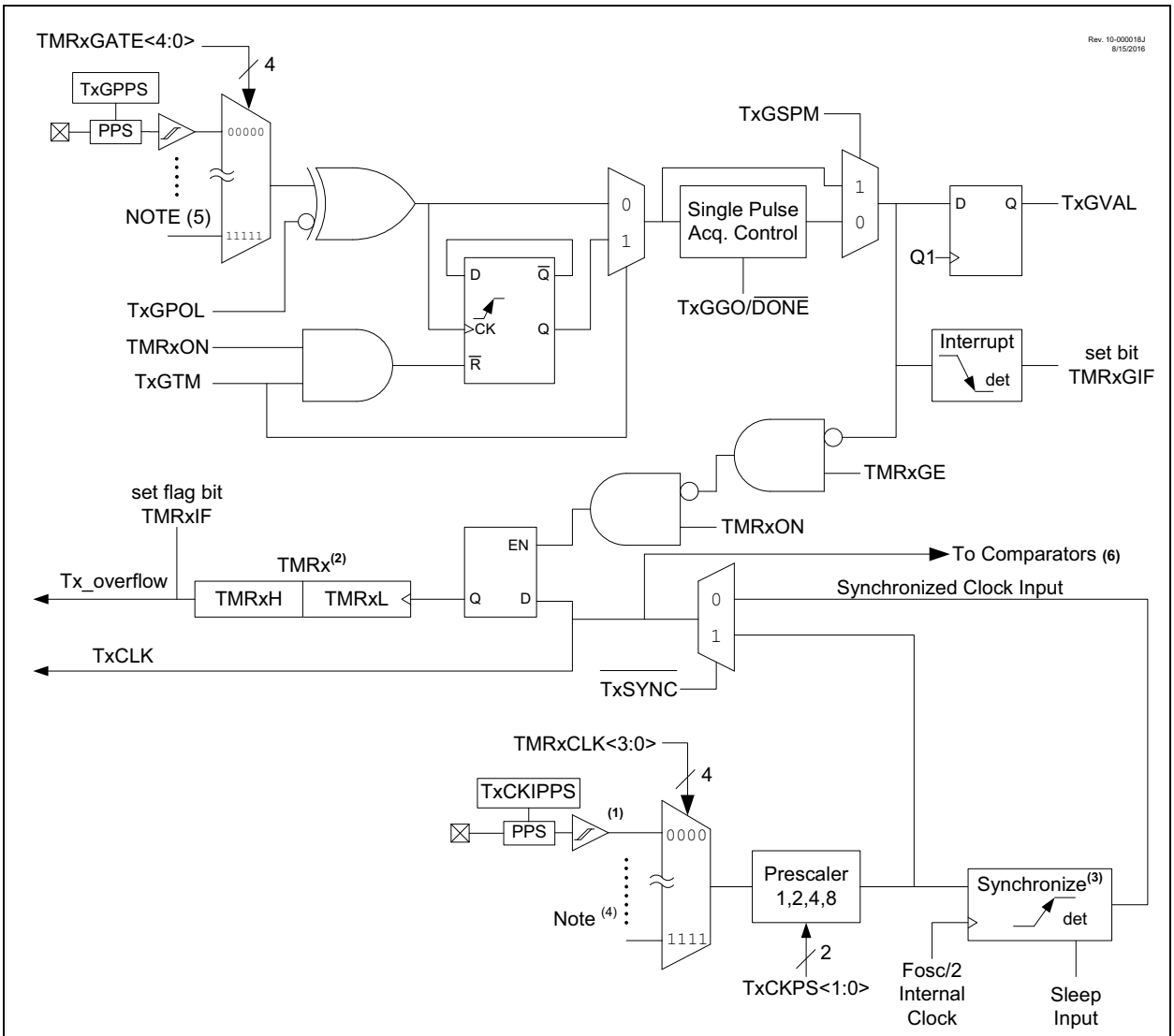
The Timer1 module is 16-bit timer/counters with the following features:

- 16-bit timer/counter register pair (TMR1H:TMR1L)
- Programmable internal or external clock source
- 2-bit prescaler
- Clock source for optional comparator synchronization
- Multiple Timer1 gate (count enable) sources
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- Time base for the Capture/Compare function
- Auto-conversion Trigger (with CCP)
- Selectable Gate Source Polarity
- Gate Toggle mode
- Gate Single-Pulse mode
- Gate Value Status
- Gate Event Interrupt

Figure 26-1 is a block diagram of the Timer1 module.

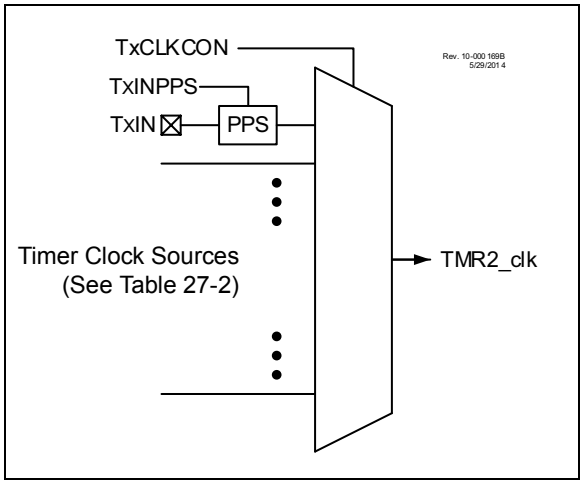
This device has one instance of Timer1 type modules.

FIGURE 26-1: TIMER1 BLOCK DIAGRAM



Note 1: ST Buffer is high speed type when using TxCKIPPS.
2: TMRx register increments on rising edge.
3: Synchronize does not operate while in Sleep.
4: See Register 26-3 for Clock source selections.
5: See Register 26-4 for GATE source selections.
6: Synchronized comparator output should not be used in conjunction with synchronized input clock.

FIGURE 27-2: TIMER2/4 CLOCK SOURCE BLOCK DIAGRAM



27.1 Timer2/4 Operation

Timer2 operates in three major modes:

- Free Running Period
- One-shot
- Monostable

Within each mode there are several options for starting, stopping, and reset. Table 27-1 lists the options.

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

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

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

- a write to the TMR2 register
- a write to the T2CON register
- any device Reset
- External Reset Source event that resets the timer.

Note:	TMR2 is not cleared when T2CON is written.

27.1.1 FREE RUNNING PERIOD MODE

The value of TMR2 is compared to that of the Period register, T2PR, on each TMR2_clk cycle. When the two values match, the comparator resets the value of TMR2 to 00h on the next rising TMR2_clk edge and increments the output postscaler counter. When the postscaler count equals the value in the OUTPS<4:0> bits of the TMRxCON1 register, a one TMR2_clk period wide pulse occurs on the TMR2_postscaled output, and the postscaler count is cleared.

27.1.2 ONE-SHOT MODE

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

27.1.3 MONOSTABLE MODE

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

27.2 Timer2/4 Output

The Timer2 module's primary output is TMR2_postscaled, which pulses for a single TMR2_clk period when the postscaler counter matches the value in the OUTPS bits of the TMR2CON register. The T2PR postscaler is incremented each time the TMR2 value matches the T2PR value. This signal can be selected as an input to several other input modules:

- The ADC module, as an Auto-conversion Trigger
- COG, as an auto-shutdown source

In addition, the Timer2 is also used by the CCP module for pulse generation in PWM mode. Both the actual TMR2 value as well as other internal signals are sent to the CCP module to properly clock both the period and pulse width of the PWM signal. See **Section 29.0 "Capture/Compare/PWM Modules"** for more details on setting up Timer2/4 for use with the CCP, as well as the timing diagrams in **Section 27.5 "Operation Examples"** for examples of how the varying Timer2 modes affect CCP PWM output.

27.3 External Reset Sources

In addition to the clock source, the Timer2 also takes in an external Reset source. This external Reset source is selected for Timer2 with the T2RST register. This source can control starting and stopping of the timer, as well as resetting the timer, depending on which mode the timer is in. The mode of the timer is controlled by the MODE<4:0> bits of the TMRxHLT register. Edge-Triggered modes require six Timer clock periods between external triggers. Level-Triggered modes require the triggering level to be at least three Timer clock periods long. External triggers are ignored while in Debug Freeze mode.

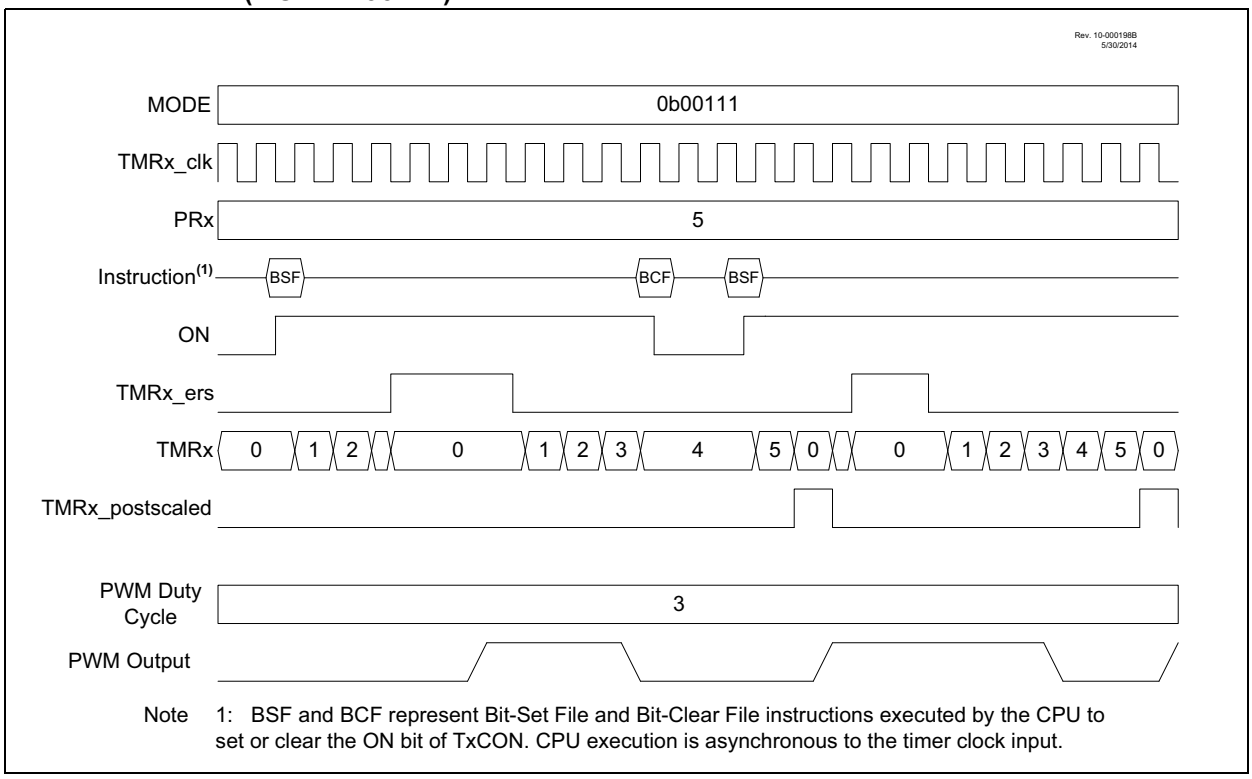
27.5.4 LEVEL-TRIGGERED HARDWARE LIMIT MODE

In the Level-Triggered Hardware Limit Timer modes the counter is reset by high or low levels of the external signal TMRx_ers, as shown in Figure 27-7. Selecting MODE<4:0> = 00110 will cause the timer to reset on a low level external signal. Selecting MODE<4:0> = 00111 will cause the timer to reset on a high level external signal. In the example, the counter is reset while TMRx_ers = 1. ON is controlled by BSF and BCF instructions. When ON = 0 the external signal is ignored.

When the CCP uses the timer as the PWM time base then the PWM output will be set high when the timer starts counting and then set low only when the timer count matches the CCPRx value. The timer is reset when either the timer count matches the PRx value or two clock periods after the external Reset signal goes true and stays true.

The timer starts counting, and the PWM output is set high, on either the clock following the PRx match or two clocks after the external Reset signal relinquishes the Reset. The PWM output will remain high until the timer counts up to match the CCPRx pulse width value. If the external Reset signal goes true while the PWM output is high then the PWM output will remain high until the Reset signal is released allowing the timer to count up to match the CCPRx value.

FIGURE 27-7: LEVEL-TRIGGERED HARDWARE LIMIT MODE TIMING DIAGRAM (MODE = 00111)



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

31.3 Selectable Input Sources

The CWG generates the output waveforms from the input sources (See Register 31-9).

The input sources are selected using the CWG1ISM register.

31.4 Output Control

31.4.1 POLARITY CONTROL

The polarity of each CWG output can be selected independently. When the output polarity bit is set, the corresponding output is active-high. Clearing the output polarity bit configures the corresponding output as active-low. However, polarity does not affect the override levels. Output polarity is selected with the POLx bits of the CWG1CON1. Auto-shutdown and steering options are unaffected by polarity.

31.8 Dead-Band Uncertainty

When the rising and falling edges of the input source are asynchronous to the CWG clock, it creates uncertainty in the dead-band time delay. The maximum uncertainty is equal to one CWG clock period. Refer to Equation 31-1 for more details.

EQUATION 31-1: DEAD-BAND UNCERTAINTY

$$T_{DEADBAND\_UNCERTAINTY} = \frac{1}{Fcwg\_clock}$$

Example:

$FCWG\_CLOCK = 16\ MHz$

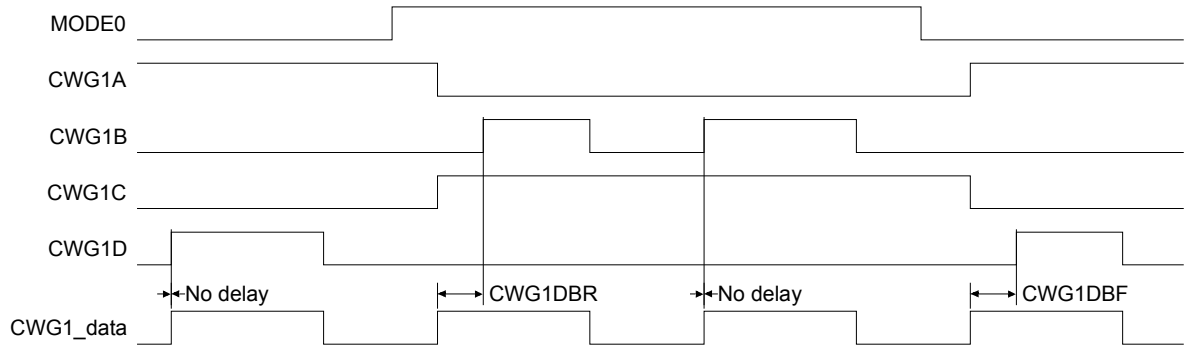
Therefore:

$$T_{DEADBAND\_UNCERTAINTY} = \frac{1}{Fcwg\_clock}$$

$$= \frac{1}{16MHz}$$

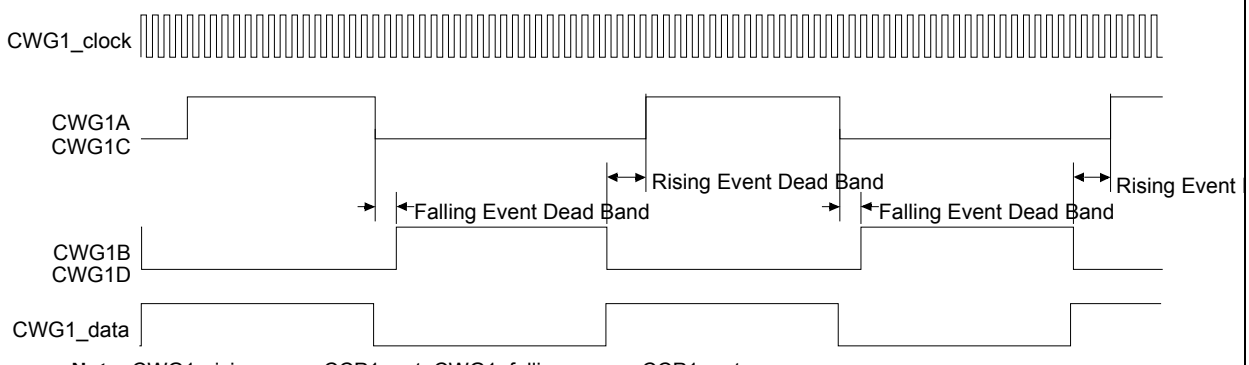
$$= 62.5ns$$

FIGURE 31-8: EXAMPLE OF PWM DIRECTION CHANGE



Note 1: WGPOL{ABCD} = 0
2: The direction bit MODE<0> (Register 31-1) can be written any time during the PWM cycle, and takes effect at the next rising CWG1_data.
3: When changing directions, CWG1A and CWG1C switch at rising CWG1_data; modulated CWG1B and CWG1D are held inactive for the dead band duration

FIGURE 31-9: CWG HALF-BRIDGE MODE OPERATION



Note: CWG1_rising_src = CCP1_out, CWG1_falling_src = ~CCP1_out

33.5.4 SLAVE MODE 10-BIT ADDRESS RECEPTION

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

Figure 33-20 is used as a visual reference for this description.

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

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

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

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

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

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

33.5.5 10-BIT ADDRESSING WITH ADDRESS OR DATA HOLD

Reception using 10-bit addressing with AHEN or DHEN set is the same as with 7-bit modes. The only difference is the need to update the SSPxADD register using the UA bit. All functionality, specifically when the CKP bit is cleared and SCL line is held low are the same. Figure 33-21 can be used as a reference of a slave in 10-bit addressing with AHEN set.

Figure 33-22 shows a standard waveform for a slave transmitter in 10-bit Addressing mode.

33.6.6 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPxBUF register. This action will set the Buffer Full flag bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high. When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an $\overline{ACK}$ bit during the ninth bit time if an address match occurred, or if data was received properly. The status of $\overline{ACK}$ is written into the ACKSTAT bit on the rising edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPxIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPxBUF, leaving SCL low and SDA unchanged (Figure 33-28).

After the write to the SSPxBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/$\overline{W}$ bit are completed. On the falling edge of the eighth clock, the master will release the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the $\overline{ACK}$ bit is loaded into the ACKSTAT Status bit of the SSPxCON2 register. Following the falling edge of the ninth clock transmission of the address, the SSPxIF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPxBUF takes place, holding SCL low and allowing SDA to float.

33.6.6.1 BF Status Flag

In Transmit mode, the BF bit of the SSPxSTAT register is set when the CPU writes to SSPxBUF and is cleared when all eight bits are shifted out.

33.6.6.2 WCOL Status Flag

If the user writes the SSPxBUF when a transmit is already in progress (i.e., SSPxSR is still shifting out a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

WCOL must be cleared by software before the next transmission.

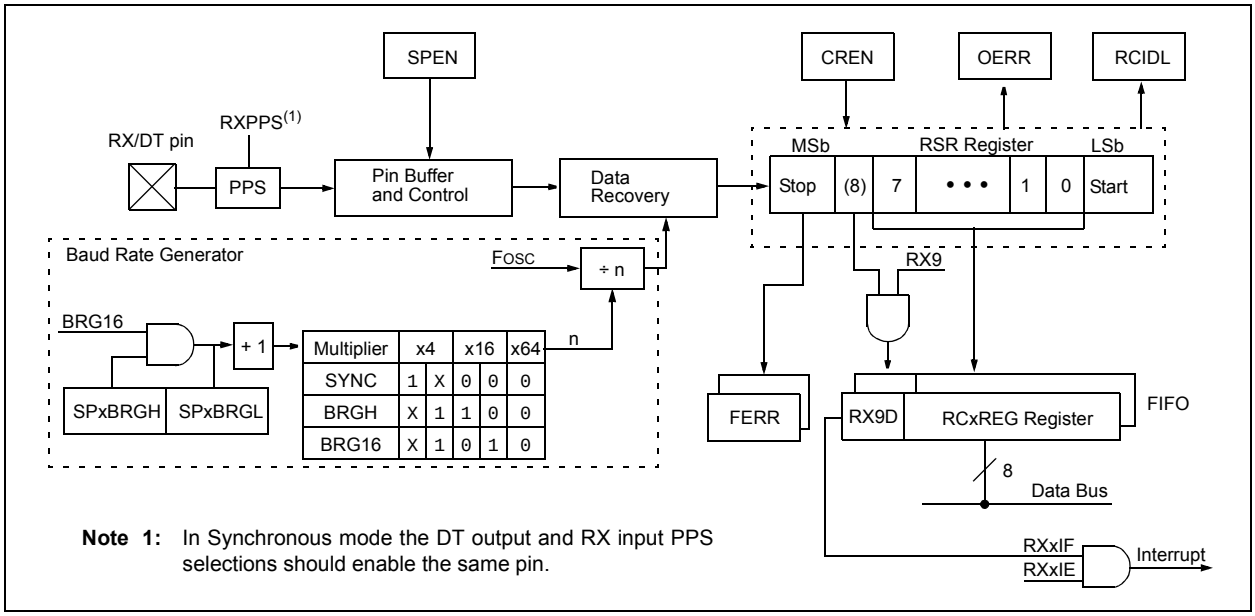
33.6.6.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit of the SSPxCON2 register is cleared when the slave has sent an Acknowledge ($\overline{ACK}$ = 0) and is set when the slave does not Acknowledge ($\overline{ACK}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

33.6.6.4 Typical transmit sequence:

1. The user generates a Start condition by setting the SEN bit of the SSPxCON2 register.
2. SSPxIF is set by hardware on completion of the Start.
3. SSPxIF is cleared by software.
4. The MSSP module will wait the required start time before any other operation takes place.
5. The user loads the SSPxBUF with the slave address to transmit.
6. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPxBUF is written to.
7. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPxCON2 register.
8. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPxIF bit.
9. The user loads the SSPxBUF with eight bits of data.
10. Data is shifted out the SDA pin until all eight bits are transmitted.
11. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPxCON2 register.
12. Steps 8-11 are repeated for all transmitted data bytes.
13. The user generates a Stop or Restart condition by setting the PEN or RSEN bits of the SSPxCON2 register. Interrupt is generated once the Stop/Restart condition is complete.

FIGURE 34-2: EUSART RECEIVE BLOCK DIAGRAM



Note 1: In Synchronous mode the DT output and RX input PPS selections should enable the same pin.

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

- Transmit Status and Control (TXxSTA)
- Receive Status and Control (RCxSTA)
- Baud Rate Control (BAUDxCON)

These registers are detailed in Register 34-1, Register 34-2 and Register 34-3, respectively.

The RX input pin is selected with the RXPPS. The CK input is selected with the TXPPS register. 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 EUSART control logic will control the data direction drivers automatically.

TABLE 34-3: BAUD RATE FORMULAS

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	FOSC/[64 (n+1)]
0	0	1	8-bit/Asynchronous	FOSC/[16 (n+1)]
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	FOSC/[4 (n+1)]
1	0	x	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

Legend: x = Don't care, n = value of SPxBRGH, SPxBRGL register pair.

TABLE 34-4: BAUD RATE FOR ASYNCHRONOUS MODES

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	FOSC = 32.000 MHz			FOSC = 20.000 MHz			FOSC = 18.432 MHz			FOSC = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	1221	1.73	255	1200	0.00	239	1200	0.00	143
2400	2404	0.16	207	2404	0.16	129	2400	0.00	119	2400	0.00	71
9600	9615	0.16	51	9470	-1.36	32	9600	0.00	29	9600	0.00	17
10417	10417	0.00	47	10417	0.00	29	10286	-1.26	27	10165	-2.42	16
19.2k	19.23k	0.16	25	19.53k	1.73	15	19.20k	0.00	14	19.20k	0.00	8
57.6k	55.55k	-3.55	3	—	—	—	57.60k	0.00	7	57.60k	0.00	2
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 0											
	FOSC = 8.000 MHz			FOSC = 4.000 MHz			FOSC = 3.6864 MHz			FOSC = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	300	0.16	207	300	0.00	191	300	0.16	51
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	11	10417	0.00	5	—	—	—	—	—	—
19.2k	—	—	—	—	—	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	—	—	—	57.60k	0.00	0	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

TABLE 38-1: REGISTER FILE SUMMARY FOR PIC16(L)F19155/56/75/76/85/86 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
1E95h	—	Unimplemented								
1E96h	—	Unimplemented								
1E97h	—	Unimplemented								
1E98h	—	Unimplemented								
1E99h	—	Unimplemented								
1E9Ah	—	Unimplemented								
1E9Bh	—	Unimplemented								
1E9Ch	T2AINPPS	—	—	—	T2INPPS<4:0>					264
1E9Dh	T4AINPPS	—	—	—	T4INPPS<4:0>					264
1E9Eh	—	Unimplemented								
1E9Fh	—	Unimplemented								
1EA0h	—	Unimplemented								
1EA1h	CCP1PPS	—	—	—	CCP1PPS<4:0>					264
1EA2h	CCP2PPS	—	—	—	CCP2PPS<4:0>					264
1EA3h	—	Unimplemented								
1EA4h	—	Unimplemented								
1EA5h	—	Unimplemented								
1EA6h	—	Unimplemented								
1EA7h	—	Unimplemented								
1EA8h	—	Unimplemented								
1EA9h	SMT1WINPPS	—	—	—	SMT1WINPPS<4:0>					264
1EAAh	SMT1SIGPPS	—	—	—	SMT1SIGPPS<4:0>					264
1EABh	—	Unimplemented								
1EACh	—	Unimplemented								
1EADh	—	Unimplemented								
1EAEh	—	Unimplemented								
1EAFh	—	Unimplemented								
1EB0h	—	Unimplemented								
1EB1h	CWG1PPS	—	—	—	CWG1PPS<4:0>					264
1EB2h	—	Unimplemented								
1EB3h	—	Unimplemented								
1EB4h	—	Unimplemented								
1EB5h	—	Unimplemented								
1EB6h	—	Unimplemented								
1EB7h	—	Unimplemented								
1EB8h	—	Unimplemented								
1EB9h	—	Unimplemented								
1EBAh	—	Unimplemented								
1EBBh	CLCIN0PPS	—	—	—	CLCIN0PPS<4:0>					264
1EBCh	CLCIN1PPS	—	—	—	CLCIN1PPS<4:0>					264
1EBDh	CLCIN2PPS	—	—	—	CLCIN2PPS<4:0>					264

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

Note 1: Unimplemented data memory locations, read as '0'.

