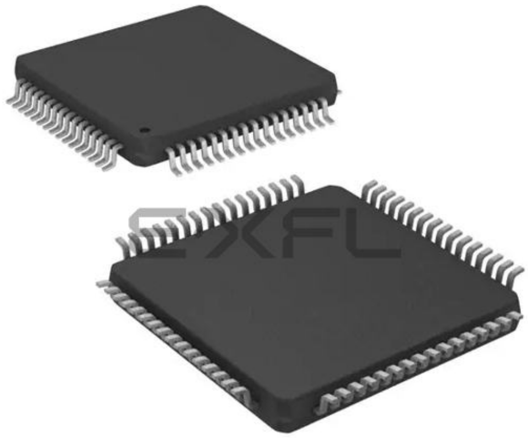
Microchip Technology - PIC18F66J94-I/PT Datasheet



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	64MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, HLVD, LCD, POR, PWM, WDT
Number of I/O	52
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 16x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f66j94-i-pt

Email: info@E-XFL.COM

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

2.2 Power Supply Pins

2.2.1 DECOUPLING CAPACITORS

The use of decoupling capacitors on every pair of power supply pins, such as VDD, VSS, AVDD and AVSS, is required.

Consider the following criteria when using decoupling capacitors:

- **Value and type of capacitor:** A 0.1 µF (100 nF), 10-20V capacitor is recommended. The capacitor should be a low-ESR device, with a resonance frequency in the range of 200 MHz and higher. Ceramic capacitors are recommended.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is no greater than 0.25 inch (6 mm).
- **Handling high-frequency noise:** If the board is experiencing high-frequency noise (upward of tens of MHz), add a second ceramic type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 µF to 0.001 µF. Place this second capacitor next to each primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible (e.g., 0.1 µF in parallel with 0.001 µF).
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB trace inductance.

2.2.2 TANK CAPACITORS

On boards with power traces running longer than six inches in length, it is suggested to use a tank capacitor for integrated circuits, including microcontrollers, to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device, and the maximum current drawn by the device in the application. In other words, select the tank capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7 µF to 47 µF.

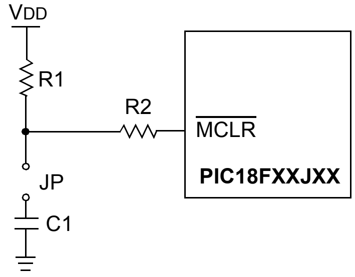
2.3 Master Clear ($\overline{\text{MCLR}}$) Pin

The $\overline{\text{MCLR}}$ pin provides two specific device functions: Device Reset, and Device Programming and Debugging. If programming and debugging are not required in the end application, a direct connection to VDD may be all that is required. The addition of other components, to help increase the application's resistance to spurious Resets from voltage sags, may be beneficial. A typical configuration is shown in Figure 2-1. Other circuit designs may be implemented, depending on the application's requirements.

During programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the $\overline{\text{MCLR}}$ pin. Consequently, specific voltage levels (VIH and VIL) and fast signal transitions must not be adversely affected. Therefore, specific values of R1 and C1 will need to be adjusted based on the application and PCB requirements. For example, it is recommended that the capacitor, C1, be isolated from the $\overline{\text{MCLR}}$ pin during programming and debugging operations by using a jumper (Figure 2-2). The jumper is replaced for normal run-time operations.

Any components associated with the $\overline{\text{MCLR}}$ pin should be placed within 0.25 inch (6 mm) of the pin.

FIGURE 2-2: EXAMPLE OF $\overline{\text{MCLR}}$ PIN CONNECTIONS



Note 1: R1 ≤ 10 kΩ is recommended. A suggested starting value is 10 kΩ. Ensure that the $\overline{\text{MCLR}}$ pin VIH and VIL specifications are met.

2: R2 ≤ 470Ω will limit any current flowing into $\overline{\text{MCLR}}$ from the external capacitor, C, in the event of $\overline{\text{MCLR}}$ pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS). Ensure that the $\overline{\text{MCLR}}$ pin VIH and VIL specifications are met.

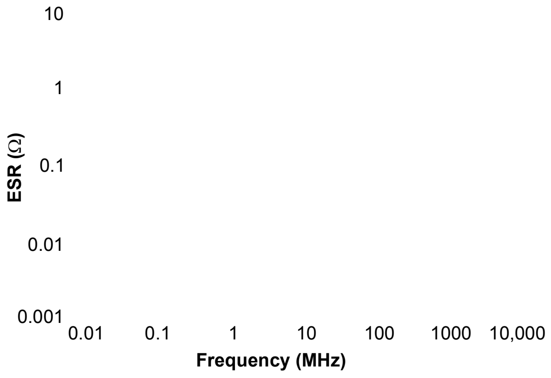
2.4 Core Voltage Regulator (VCAP/ VDDCORE)

A low-ESR (< 5Ω) capacitor is required on the VCAP pin to stabilize the output voltage of the on-chip voltage regulator. The VCAP pin must not be connected to VDD and must use a capacitor of 10 µF connected to ground. The type can be ceramic or tantalum. Suitable examples of capacitors are shown in Table 2-1. Capacitors with equivalent specification can be used.

Designers may use Figure 2-3 to evaluate ESR equivalence of candidate devices.

It is recommended that the trace length not exceed 0.25 inch (6 mm). Refer to **Section 30.0 “Electrical Specifications”** for additional information.

FIGURE 2-3: FREQUENCY vs. ESR PERFORMANCE FOR SUGGESTED VCAP



Note: Typical data measurement at 25°C, 0V DC bias.

.

TABLE 2-1: SUITABLE CAPACITOR EQUIVALENTS

Make	Part #	Nominal Capacitance	Base Tolerance	Rated Voltage	Temp. Range
TDK	C3216X7R1C106K	10 µF	±10%	16V	-55 to 125ºC
TDK	C3216X5R1C106K	10 µF	±10%	16V	-55 to 85ºC
Panasonic	ECJ-3YX1C106K	10 µF	±10%	16V	-55 to 125ºC
Panasonic	ECJ-4YB1C106K	10 µF	±10%	16V	-55 to 85ºC
Murata	GRM32DR71C106KA01L	10 µF	±10%	16V	-55 to 125ºC
Murata	GRM31CR61C106KC31L	10 µF	±10%	16V	-55 to 85ºC

3.2 Oscillator Configuration

The oscillator source (and operating mode) that is used at a device Power-on Reset (POR) event is selected using Configuration bit settings. The Oscillator Configuration bit settings are in the Configuration registers located in the program memory (refer to **Section 28.1 “Configuration Bits”** for more information). The Primary Oscillator Configuration bits, POSCMD<1:0> (CONFIG3L<1:0>), and Oscillator Configuration bits, FOSC<2:0> (CONFIG2L<2:0>), select the oscillator source that is used at a POR. The FRC Oscillator with Postscaler (FRCDIV) is the default (unprogrammed) selection. The Secondary Oscillator, or one of the internal oscillators, may be chosen by programming these bit locations.

The Configuration bits allow users to choose between 11 different clock modes, as shown in Table 3-1.

TABLE 3-1: CONFIGURATION BIT VALUES FOR CLOCK SELECTION

Oscillator Mode	Oscillator Source	POSCMD<1:0>	FOSC<2:0>	Notes
Fast RC Oscillator with Postscaler (FRCDIV)	Internal	11	111	**1, 2**
Fast RC Oscillator divided by 16 (FRC500kHz)	Internal	11	110	**1**
Low-Power RC Oscillator (LPRC)	Internal	11	101	**1**
Secondary (Timer1) Oscillator (SOSC)	Secondary	11	100	**1**
Primary Oscillator (HS) with PLL Module (HSPLL)	Primary	10	011	
Primary Oscillator (MS) with PLL Module (MSPLL)	Primary	01	011	
Primary Oscillator (EC) with PLL Module (ECPLL)	Primary	00	011	
Primary Oscillator (HS)	Primary	10	010	
Primary Oscillator (MS)	Primary	01	010	
Primary Oscillator (EC)	Primary	00	010	
Fast RC Oscillator with PLL Module (FRCPLL)	Internal	11	001	**1**
Fast RC Oscillator (FRC)	Internal	11	000	**1**

Note 1: OSC2 pin function is determined by the CLKOEN Configuration bit.

2: Default oscillator mode for an unprogrammed (erased) device.

3.5 Primary Oscillator (POSC)

The Primary Oscillator is available on the OSC1 and OSC2 pins of the PIC18F family. In general, the Primary Oscillator can be configured for an external clock input or an external crystal. Further details of the Primary Oscillator operating modes are described in subsequent sections. The Primary Oscillator has up to 6 operating modes, summarized in Table 3-2.

TABLE 3-2: PRIMARY OSCILLATOR OPERATING MODES

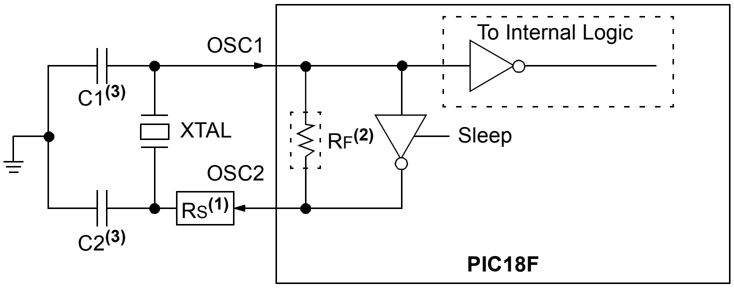
Oscillator Mode	Description	OSC2 Pin Function
EC	External clock input (0-64 MHz)	FOSC/4
ECPLL	External clock input (4-48 MHz), PLL enabled	FOSC/4, **Note 2**
HS	10 MHz-32 MHz crystal	**Note 1**
HSPLL	10 MHz-32 MHz crystal, PLL enabled	**Note 2**
MS	3.5 MHz-10 MHz crystal	**Note 1**
MSPLL	3.5 MHz-8 MHz crystal, PLL enabled	**Note 1**

Note 1: External crystal is connected to OSC1 and OSC2 in these modes.
2: Available only in devices with special PLL blocks (such as the 96 MHz PLL); the basic 4x PLL block generates clock frequencies beyond the device's operating range.

The POSCMDx and FOSCx Configuration bits (CONFIG3L<1:0> and CONFIG2L<2:0>, respectively) select the operating mode of the Primary Oscillator. The POSCMD<1:0> bits select the particular submode to be used (MS, HS or EC), while the FOSC<2:0> bits determine if the oscillator will be used by itself or with the internal PLL. The PIC18F operates from the Primary Oscillator whenever the COSCx bits (OSCCON<6:4>) are set to '010' or '011'.

Refer to the **"Electrical Characteristics"** section in the specific device data sheet for further information regarding frequency range for each crystal mode.

FIGURE 3-3: CRYSTAL OR CERAMIC RESONATOR OPERATION (MS OR HS OSCILLATOR MODE)



Note 1: A series resistor, Rs, may be required for AT strip cut crystals.
2: The internal feedback resistor, RF, is typically in the range of 2 to 10 MΩ.
3: See **Section 3.6.5 "Determining the Best Values for Oscillator Components"**.

3.15.3 CLOCK SYNCHRONIZATION

The Reference Clock Output is enabled only once (ON = 1). Note that the source of the clock and the divider values should be chosen prior to the bit being set to avoid glitches on the REFO output.

Once the ON bit is set, its value is synchronized to the reference clock domain to enable the output. This ensures that no glitches will be seen on the output. Similarly, when the ON bit is cleared, the output and the associated output enable signals will be synchronized, and disabled on the falling edge of the reference clock. Note that with large divider values, this will cause the REFO to be enabled for some period after ON is cleared.

3.15.4 OPERATION IN SLEEP MODE

If any clock source, other than the peripheral clock, is used as a base reference (i.e., ROSEL<3:0> ≠ 0001), the user has the option to configure the behavior of the oscillator in Sleep mode. The RSLP Configuration bit determines if the oscillator will continue to run in Sleep. If RSLP = 0, the oscillator will be shut down in Sleep (assuming no other consumers are requesting it). If RSLP = 1, the oscillator will continue to run in Sleep.

The Reference Clock Output is synchronized with the Sleep signal to avoid any glitches on its output.

3.15.5 MODULE ENABLE SIGNAL

The REFOx module may be enabled or disabled using the REFOxMD register bit (PMD3, bit 1 or 0). The module also needs to be turned on using the ON bit (REFO1CON<7>).

3.15.5.1 Registers and Bits

This module provides the following device registers and/or bits:

- REFOxCON – Reference Clock Output Control Register
- REFOxCON1 – Reference Clock Output Control 1 Register
- REFOxCON2 – Reference Clock Output Control 2 Register
- REFOxCON3 – Reference Clock Output Control 3 Register

REGISTER 10-2: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	IOCIP
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 3 **INTEDG3:** External Interrupt 3 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **INT3IP:** INT3 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **IOCIP:** RB Port Change Interrupt Priority bit
1 = High priority
0 = Low priority

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

10.7 IPR Registers

The IPR registers contain the individual priority bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are six Peripheral Interrupt Priority registers (IPR1 through IPR6). Using the priority bits requires that the Interrupt Priority Enable (IPEN) bit (RCON<7>) be set.

REGISTER 10-16: IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	TMR1GIP	TMR2IP	TMR1IP
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **PSPIP:** Parallel Slave Port Read/Write Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **ADIP:** A/D Converter Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **RC1IP:** EUSART1 Receive Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **TX1IP:** EUSART1 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **SSP1IP:** Master Synchronous Serial Port 1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **TMR1GIP:** Timer1 Gate Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR2IP:** TMR2 to PR2 Match Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR1IP:** TMR1 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 10-18: IPR3: PERIPHERAL INTERRUPT PRIORITY REGISTER 3

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR5GIP	LCDIP	RC2IP	TX2IP	CTMUIP	CCP2IP	CCP1IP	RTCCIP
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **TMR5GIP:** TMR5 Gate Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **LCDIP:** LCD Ready Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **RC2IP:** EUSART2 Receive Priority Flag bit
1 = High priority
0 = Low priority

bit 4 **TX2IP:** EUSART2 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **CTMUIP:** CTMU Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **CCP2IP:** CCP2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **CCP1IP:** ECCP1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **RTCCIP:** RTCC Interrupt Priority bit
1 = High priority
0 = Low priority

FIGURE 12-5: CARRIER LOW SYNCHRONIZATION (MDCHSYNC = 0, MDCLSYNC = 1)

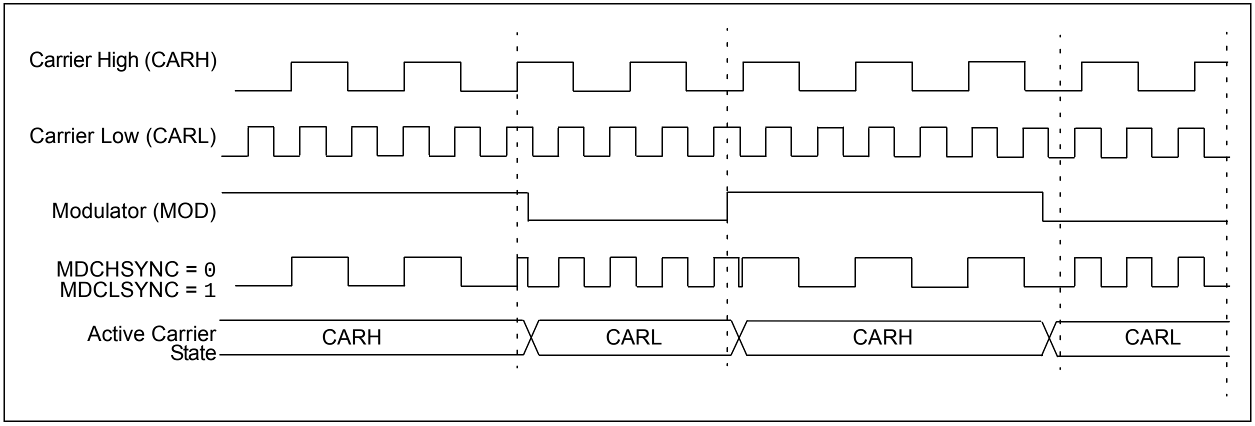


FIGURE 12-6: FULL SYNCHRONIZATION (MDCHSYNC = 1, MDCLSYNC = 1)

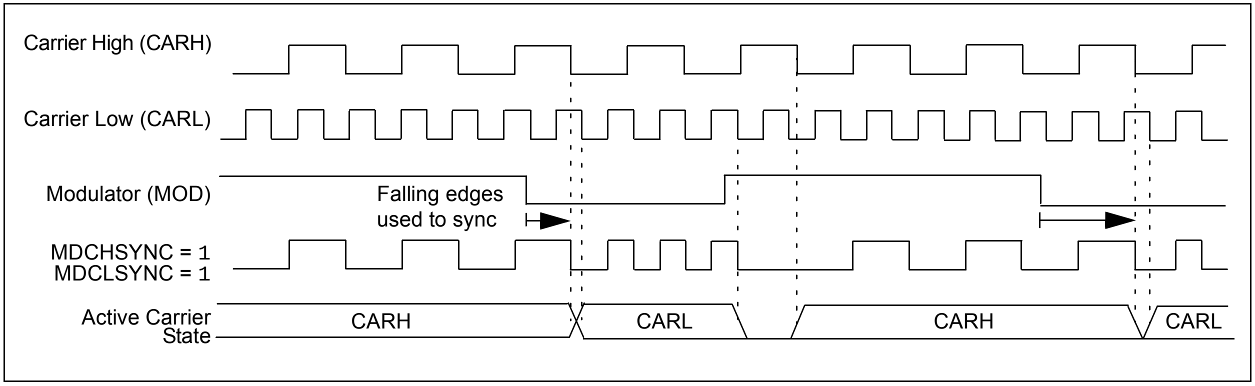
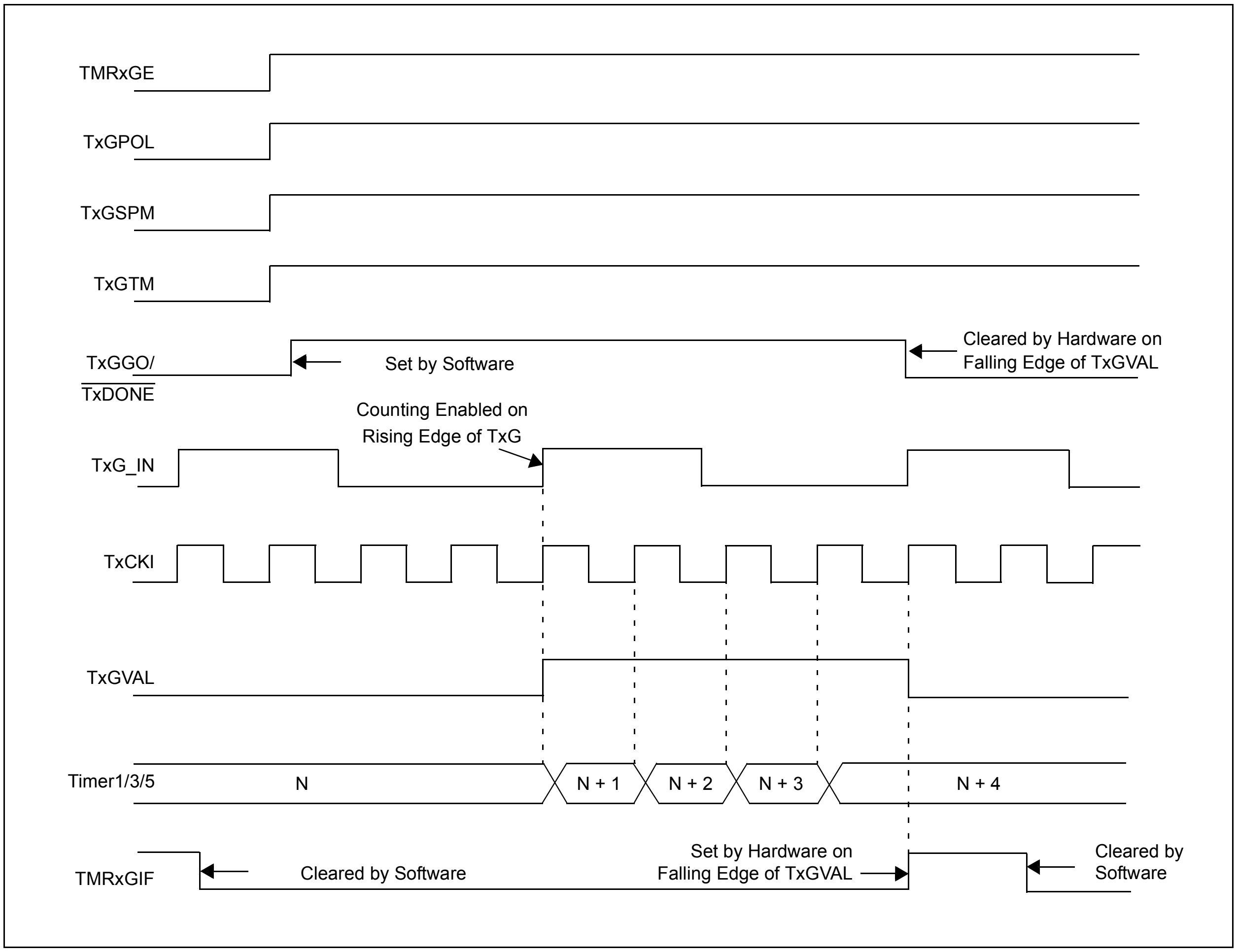


FIGURE 15-5: TIMER1/3/5 GATE SINGLE PULSE AND TOGGLE COMBINED MODE



15.5.5 TIMER1/3/5 GATE VALUE STATUS

When Timer1/3/5 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 TxGVAL bit (TxGCON<2>). The TxGVAL bit is valid even when the Timer1/3/5 gate is not enabled (TMRxGE bit is cleared).

15.5.6 TIMER1/3/5 GATE EVENT INTERRUPT

When the Timer1/3/5 gate event interrupt is enabled, it is possible to generate an interrupt upon the completion of a gate event. When the falling edge of TxGVAL occurs, the TMRxGIF flag bit in the PIRx register will be set. If the TMRxGIE bit in the PIEx register is set, then an interrupt will be recognized.

The TMRxGIF flag bit operates even when the Timer1/3/5 gate is not enabled (TMRxGE bit is cleared).

17.6 Register Maps

Table 17-5, Table 17-6 and Table 17-7 summarize the registers associated with the RTCC module.

TABLE 17-5: RTCC CONTROL REGISTERS

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RTCCON1	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0
RTCCAL	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
RTCCON2	PWCEN	PWCPOL	PWCCPRE	PWCSPRE	RTCCLKSEL1	RTCCLKSEL0	RTCSECSEL1	RTCSECSEL
ALRMCFG	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
ALRMRPT	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0
PMD3	DSMMD	CTMUMD	ADCMD	RTCCMD	LCDMD	PSPMD	REFO1MD	REFO2MD

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal for 80-pin devices.

TABLE 17-6: RTCC VALUE REGISTERS

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RTCVALH	RTCC Value High Register Window based on RTCPTR<1:0>							
RTCVALL	RTCC Value Low Register Window based on RTCPTR<1:0>							

TABLE 17-7: ALARM VALUE REGISTERS

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ALRMVALH	Alarm Value High Register Window based on ALRMPTR<1:0>							
ALRMVALL	Alarm Value Low Register Window based on ALRMPTR<1:0>							

REGISTER 19-4: CCPRxL: CCPx PERIOD LOW BYTE REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CCPRxL7	CCPRxL6	CCPRxL5	CCPRxL4	CCPRxL3	CCPRxL2	CCPRxL1	CCPRxL0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **CCPRxL<7:0>:** CCPx Period Register Low Byte bits
Capture mode: Capture Register Low Byte
Compare mode: Compare Register Low Byte
PWM mode: Duty Cycle Register

REGISTER 19-5: CCPRxH: CCPx PERIOD HIGH BYTE REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CCPRxH7	CCPRxH6	CCPRxH5	CCPRxH4	CCPRxH3	CCPRxH2	CCPRxH1	CCPRxH0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **CCPRxH<7:0>:** CCPx Period Register High Byte bits
Capture mode: Capture Register High Byte
Compare mode: Compare Register High Byte
PWM mode: Duty Cycle Buffer Register

REGISTER 20-8: SSPxCON2: MSSPx CONTROL REGISTER 2 (I^{2}C MASTER MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT[1]	ACKEN[2]	RCEN[2]	PEN[2]	RSEN[2]	SEN[2]
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **GCEN:** General Call Enable bit

Unused in Master mode.

bit 6 **ACKSTAT:** Acknowledge Status bit (Master Transmit mode only)

1 = Acknowledge was not received from slave
0 = Acknowledge was received from slave

bit 5 **ACKDT:** Acknowledge Data bit (Master Receive mode only)[1]

1 = Not Acknowledge
0 = Acknowledge

bit 4 **ACKEN:** Acknowledge Sequence Enable bit[2]

1 = Initiates Acknowledge sequence on SDAx and SCLx pins and transmits ACKDT data bit; automatically cleared by hardware
0 = Acknowledge sequence is Idle

bit 3 **RCEN:** Receive Enable bit (Master Receive mode only)[2]

1 = Enables Receive mode for I^{2}C
0 = Receive is Idle

bit 2 **PEN:** Stop Condition Enable bit[2]

1 = Initiates Stop condition on SDAx and SCLx pins; automatically cleared by hardware
0 = Stop condition is Idle

bit 1 **RSEN:** Repeated Start Condition Enable bit[2]

1 = Initiates Repeated Start condition on SDAx and SCLx pins; automatically cleared by hardware
0 = Repeated Start condition is Idle

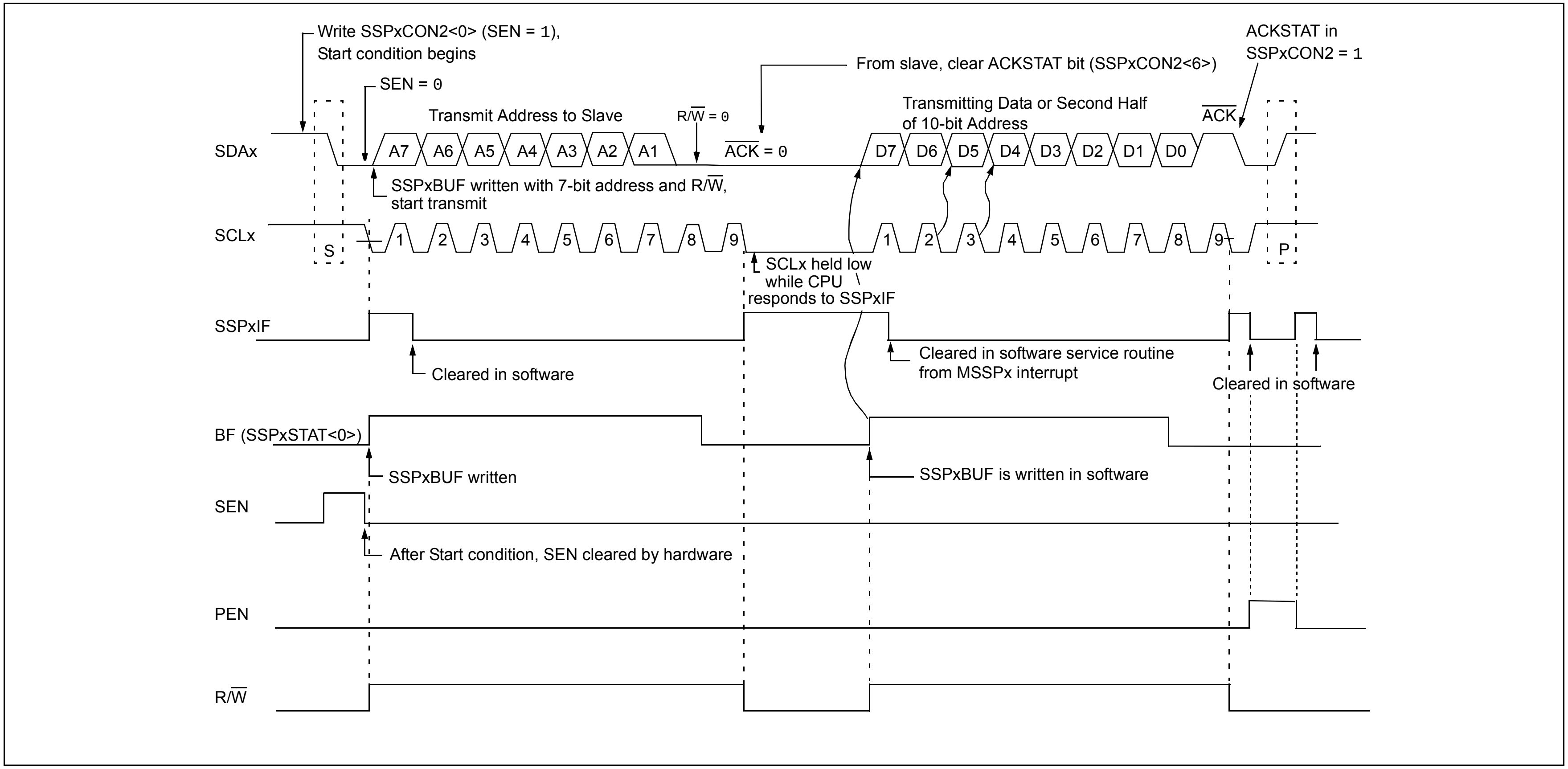
bit 0 **SEN:** Start Condition Enable bit[2]

1 = Initiates Start condition on SDAx and SCLx pins; automatically cleared by hardware
0 = Start condition is Idle

Note 1: The value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.

2: If the I^{2}C module is active, these bits may not be set (no spooling) and the SSPxBUF may not be written (or writes to the SSPxBUF are disabled).

FIGURE 20-23: I^2C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)



21.1 Baud Rate Generator (BRG)

The BRG is a dedicated, 8-bit or 16-bit generator that supports both the Asynchronous and Synchronous modes of the EUSARTx. By default, the BRG operates in 8-bit mode; setting the BRG16 bit (BAUDCONx<3>) selects 16-bit mode.

The SPBRGHx:SPBRGx register pair controls the period of a free-running timer. In Asynchronous mode, bits, BRGH (TXSTAx<2>) and BRG16 (BAUDCONx<3>), also control the baud rate. In Synchronous mode, BRGH is ignored. Table 21-1 shows the formula for computation of the baud rate for different EUSARTx modes which only apply in Master mode (internally generated clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRGHx:SPBRGx registers can be calculated using the formulas in Table 21-1. From this, the error in baud rate can be determined. An example calculation is shown in Example 21-1. Typical baud rates and error values for the various Asynchronous modes are shown in Table 21-2. It may be advantageous to use the high baud rate (BRGH = 1) or the 16-bit BRG to reduce the baud rate error, or achieve a slow baud rate for a fast oscillator frequency.

Writing a new value to the SPBRGHx:SPBRGx registers causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate. When operated in the Synchronous mode, SPBRGH:SPBRG values of 0000h and 0001h are not supported. In the Asynchronous mode, all BRG values may be used.

21.1.1 OPERATION IN POWER-MANAGED MODES

The device clock is used to generate the desired baud rate. When one of the power-managed modes is entered, the new clock source may be operating at a different frequency. This may require an adjustment to the value in the SPBRGx register pair.

21.1.2 SAMPLING

The data on the RXx pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RXx pin.

TABLE 21-1: 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 SPBRGHx:SPBRGx register pair

EXAMPLE 21-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, and 8-bit BRG:

Desired Baud Rate = FOSC/(64 ([SPBRGHx:SPBRGx] + 1))
Solving for SPBRGHx:SPBRGx:
X = ((FOSC/Desired Baud Rate)/64) – 1
= ((16000000/9600)/64) – 1
= [25.042] = 25
Calculated Baud Rate = 16000000/(64 (25 + 1))
= 9615
Error = (Calculated Baud Rate – Desired Baud Rate)/Desired Baud Rate
= (9615 – 9600)/9600 = 0.16%

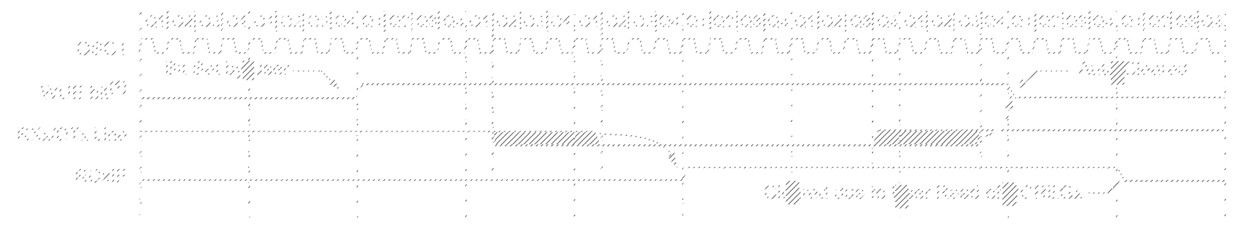
21.2.4.2 Special Considerations Using the WUE Bit

The timing of WUE and RCxIF events may cause some confusion when it comes to determining the validity of received data. As noted, setting the WUE bit places the EUSARTx in an Idle mode. The wake-up event causes a receive interrupt by setting the RCxIF bit. The WUE bit is cleared after this when a rising edge is seen on RXx/DTx. The interrupt condition is then cleared by reading the RCREGx register. Ordinarily, the data in RCREGx will be dummy data and should be discarded.

The fact that the WUE bit has been cleared (or is still set), and the RCxIF flag is set, should not be used as an indicator of the integrity of the data in RCREGx. Users should consider implementing a parallel method in firmware to verify received data integrity.

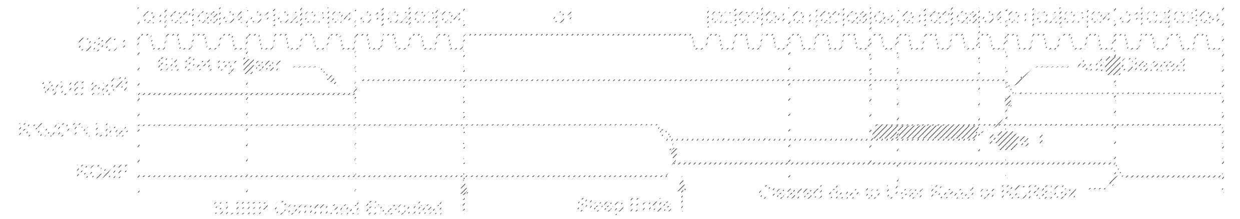
To assure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

FIGURE 21-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING NORMAL OPERATION



Note 1: The EUSARTx remains in Idle while the WUE bit is set.

FIGURE 21-9: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



Note 1: If the wake-up event requires long oscillator warm-up time, the auto-clear of the WUE bit can occur before the oscillator is ready. This sequence should not depend on the presence of Q clocks.

2: The EUSARTx remains in Idle while the WUE bit is set.

REGISTER 22-24: ADCTMUEN0H: CTMU ENABLE REGISTER 0 HIGH (HIGH WORD)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN15	CTMUEN14	CTMUEN13	CTMUEN12	CTMUEN11	CTMUEN10	CTMUEN9	CTMUEN8
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **CTMUEN<15:8>:** CTMU Enabled During Conversion bits
1 = CTMU is enabled and connected to the selected channel during conversion
0 = CTMU is not connected to this channel

Note 1: The actual number of channels available depends on which channels are implemented on a specific device; refer to the device data sheet for details. Unimplemented channels are read as ‘0’.

REGISTER 22-25: ADCTMUEN0L: CTMU ENABLE REGISTER 0 LOW (LOW WORD)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN7	CTMUEN6	CTMUEN5	CTMUEN4	CTMUEN3	CTMUEN2	CTMUEN1	CTMUEN0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-0 **CTMUEN<7:0>:** CTMU Enabled During Conversion bits
1 = CTMU is enabled and connected to the selected channel during conversion
0 = CTMU is not connected to this channel

Note 1: The actual number of channels available depends on which channels are implemented on a specific device; refer to the device data sheet for details. Unimplemented channels are read as ‘0’.

The BUFM bit (ADCON2L<1>) controls how the buffer is filled. When BUFM is '1', the buffer is split into two equal halves: a lower half (ADCBUF0 through ADCBUF[(n/2) - 1]) and an upper half (ADCBUF[n/2] through ADCBUFn), where n is the number of available analog channels (both internal and external). The buffers will alternately receive the conversion results after each interrupt event. The initial buffer used after BUFM is set is the lower group.

When BUFM is '0', the entire buffer is used for all conversion sequences.

> **Note:** When the BUFM bit is set, the user should not program the SMPIx bits to a value that specifies more than (n/2) conversions per interrupt.

The decision to use the split buffer feature will depend upon how much time is available to move the buffer contents after the interrupt, as determined by the application. If the application can quickly unload a full buffer within the time it takes to sample and convert one channel, the BUFM bit can be '0', and up to 16 conversions may be done per interrupt. The application will have one sample/convert time before the first buffer location is overwritten.

If the processor cannot unload the buffer within the sample and conversion time, the BUFM bit should be '1'. For example, if SMPI<4:0> = 00111, then eight conversions will be loaded into the lower half of the buffer, following which, an interrupt may occur. The next eight conversions will be loaded into the upper half of the buffer. The processor will, therefore, have the entire time between interrupts to move the eight conversions out of the buffer.

22.6.2.2 Buffer Fill Status

When the conversion result buffer is split (BUFM = 1), the BUFS Status bit (ADCON2L<7>) indicates which half of the buffer that the A/D Converter is currently writing. If BUFS = 0, the A/D Converter is filling the lower group and the user application should read conversion values from the upper group. If BUFS = 1, the situation is reversed, and the user application should read conversion values from the lower group.

22.6.2.3 Channel Indexed Mode

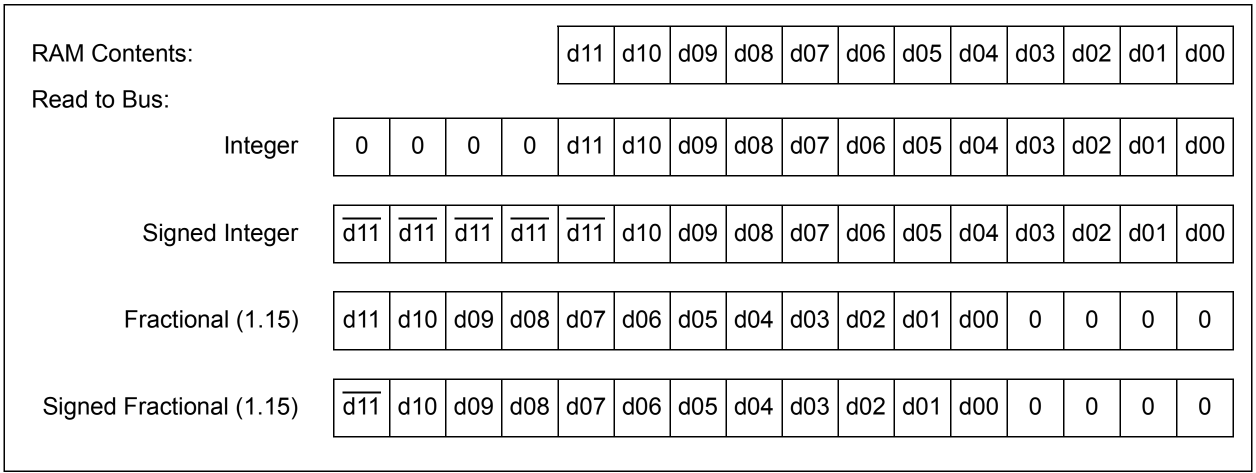
When BUFREGEN = 1, FIFO operation is disabled. In this Fill mode, the conversion result for each channel is written only to the buffer location that corresponds to that channel. For example, any conversions performed on AN0 are stored only in ADCBUF0. The same holds true for AN1 and ADCBUF1, and so on. Subsequent conversions on a particular channel that occur, prior to an interrupt, will result in any previous data in that location being overwritten.

Channel Indexed mode is particularly useful when used with the Threshold Detect feature, as this allows the user to easily test for a particular condition on a specific analog channel without creating an excess of CPU overhead. This is covered in more detail in **Section 22.7 “Threshold Detect Operation”**.

22.6.3 BUFFER DATA FORMATS

The results of each A/D conversion are 12 bits wide (optionally, 10 bits wide in some devices). To maintain data format compatibility, the result of each conversion is automatically converted to one of four selectable, 16-bit formats. The FORM<1:0> bits (ADCON1H<1:0>) select the format. Figure 22-10 and Figure 22-11 show the data output formats that can be selected. Table 22-2 through Table 22-5 show the numerical equivalents for the various conversion result codes.

FIGURE 22-10: A/D OUTPUT DATA FORMATS (12-BIT)



RAM Contents:					d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Read to Bus:																
Integer	0	0	0	0	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Signed Integer	$\overline{d11}$	$\overline{d11}$	$\overline{d11}$	$\overline{d11}$	$\overline{d11}$	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Fractional (1.15)	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0
Signed Fractional (1.15)	$\overline{d11}$	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0

27.5 USB Interrupts

The USB module can generate multiple interrupt conditions. To accommodate all of these interrupt sources, the module is provided with its own interrupt logic structure, similar to that of the microcontroller. USB interrupts are enabled with one set of control registers and trapped with a separate set of flag registers. All sources are funneled into a single USB Oscillator Fail Interrupt Flag bit, USBIF (PIR2<4>), in the microcontroller's interrupt logic.

Figure 27-7 provides the interrupt logic for the USB module. There are two layers of interrupt registers in the USB module. The top level consists of overall USB status interrupts; these are enabled and flagged in the UIE and UIR registers, respectively. The second level consists of USB error conditions, which are enabled and flagged in the UEIR and UEIE registers. An interrupt condition in any of these triggers a USB Error Interrupt Flag (UERRIF) in the top level.

Interrupts may be used to trap routine events in a USB transaction. Figure 27-8 provides some common events within a USB frame and its corresponding interrupts.

FIGURE 27-7: USB INTERRUPT LOGIC FUNNEL

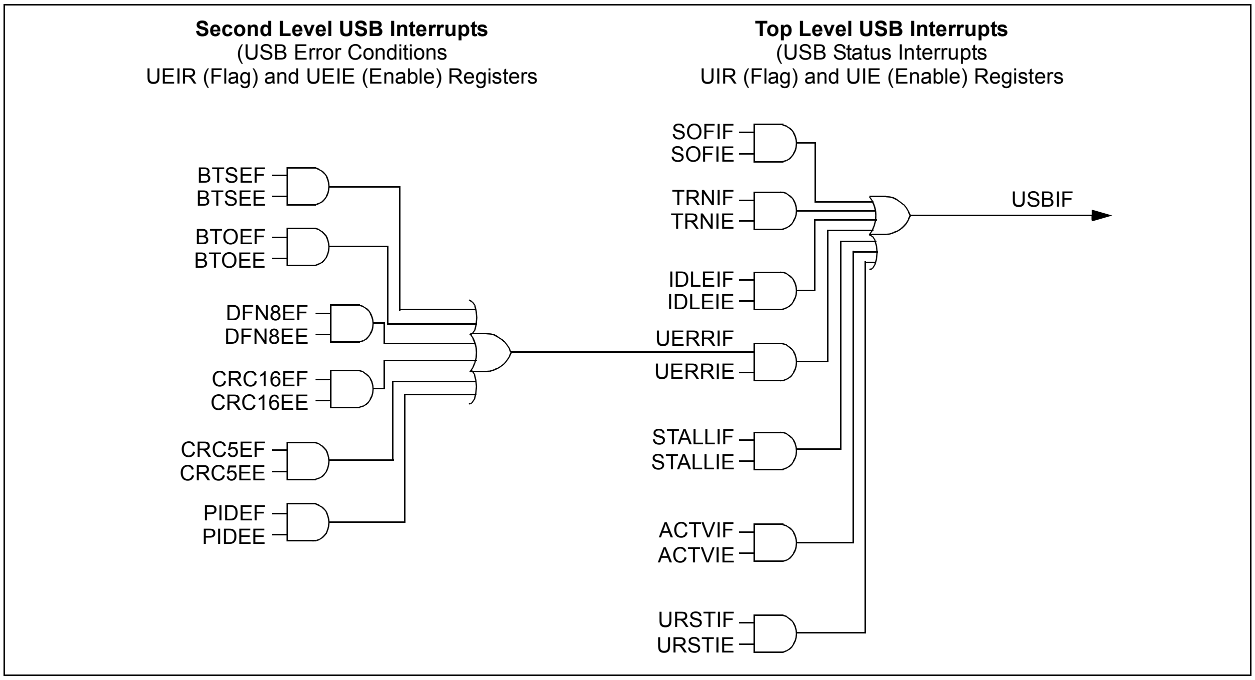
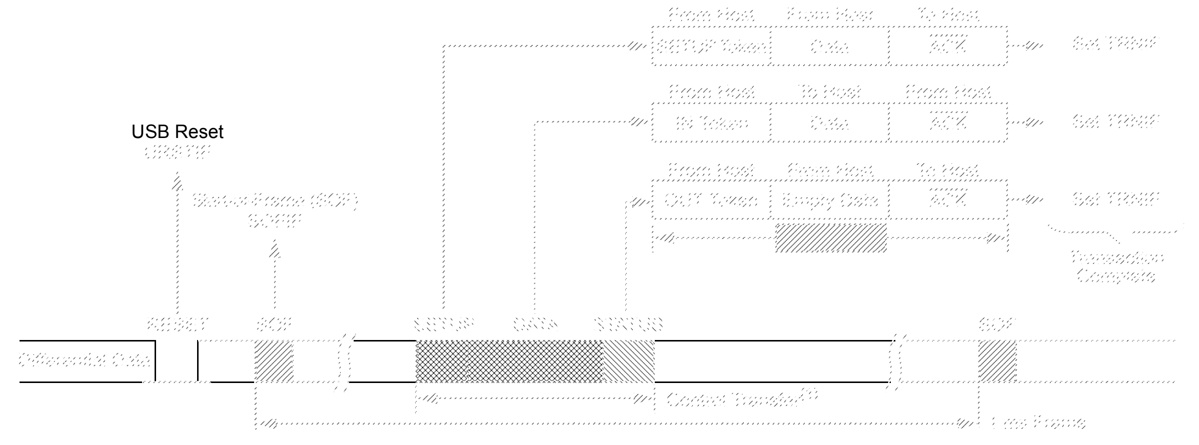


FIGURE 27-8: EXAMPLE OF A USB TRANSACTION AND INTERRUPT EVENTS



Note 1: The control transfer shown here is only an example showing events that can occur for every transaction. Typical control transfers will spread across multiple frames.

TABLE 30-10: DC CHARACTERISTICS: CTMU CURRENT SOURCE SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2V to 3.6V Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ.[1]**	**Max.**	**Units**	**Conditions**
	IOUT1	CTMU Current Source, Base Range	—	550	—	nA	CTMUCON1<1:0> = 01
	IOUT2	CTMU Current Source, 10x Range	—	5.5	—	μA	CTMUCON1<1:0> = 10
	IOUT3	CTMU Current Source, 100x Range	—	55	—	μA	CTMUCON1<1:0> = 11

Note 1: Nominal value at center point of current trim range (CTMUCON1<7:2> = 000000).

TABLE 30-11: MEMORY PROGRAMMING REQUIREMENTS

DC CHARACTERISTICS			Standard Operating Conditions Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
		Internal Program Memory Programming Specifications[1]					
D110	VPP	Voltage on $\overline{\text{MCLR}}$/VPP Pin	VDD + 1.5	—	10	V	**(Note 2, Note 3)**
D113	IDDP	Supply Current During Programming	—	—	10	mA	
		Program Flash Memory					
D130	EP	Cell Endurance	1K	20K	—	E/W	-40°C to +85°C
D131	VPR	VDD for Read					
			2	—	3.6	V	
D132B	VPEW	Voltage for Self-Timed Erase or Write Operations VDD	2	—	3.6	V	PIC18FXXKXX devices
D133A	TIW	Self-Timed Write Cycle Time	—	2	—	ms	
D133B	TIE	Self-Timed Block Erased Cycle Time	—	33	—	ms	
D134	TRETD	Characteristic Retention	10	—	—	Year	Provided no other specifications are violated
D135	IDDP	Supply Current during Programming	—	—	10	mA	
D140	TWE	Writes per Erase Cycle	—	—	1		For each physical address

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: These specifications are for programming the on-chip program memory through the use of table write instructions.

2: Required only if single-supply programming is disabled.

3: The MPLAB® ICD 2 does not support variable VPP output. Circuitry to limit the MPLAB ICD 2 VPP voltage must be placed between the MPLAB ICD 2 and the target system when programming or debugging with the MPLAB ICD 2.