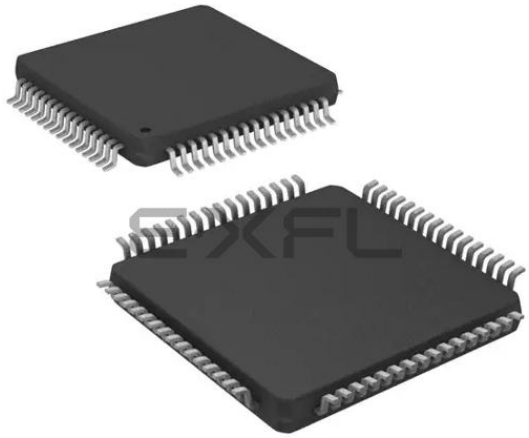


Welcome to E-XFL.COM

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	59
Program Memory Size	56KB (32K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 45x12b; D/A 1x5b
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/pic16f19197-i-pt

TABLE 4-12: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 PIC16(L)F19195/6/7 (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 2											
CPU CORE REGISTERS; see Table 4-3 for specifics											
10Ch	ADACQL	ACQ<7:0>								0000 0000	0000 0000
10Dh	ADACQH	—	—	—	ACQ<4:0>					0000 0000	0000 0000
10Eh	ADCAP	—	—	—	ADCAP<4:0>					0000 0000	0000 0000
10Fh	ADPREL	PRE<7:0>								0000 0000	0000 0000
110h	ADPREH	—	—	—	PRE<4:0>					0000 0000	0000 0000
111h	ADCON0	ON	CONT	—	CS	—	FM	—	GO	00-0 -x-0	00-0 -0-0
112h	ADCON1	PPOL	IPEN	GPOL	—	—	—	—	DSEN	000- ---0	000- ---0
113h	ADCON2	PSIS	CRS<2:0>			ACLR	MD<2:0>			0000 0000	0000 0000
114h	ADCON3	—	CALC<2:0>			SOI	TMD<2:0>			0000 0000	0000 0000
115h	ADSTAT	OV	UTHR	LTHR	MATH	—	STAT<2:0>			0000 -000	0000 -000
116h	ADREF	—	—	—	—	—	—	PREF<1:0>		0000 0000	0000 0000
117h	ADACT	—	—	—	ACT<4:0>					0000 0000	0000 0000
118h	ADCLK	—	—	CS<5:0>						0000 0000	0000 0000
119h	RC1REG	RC1REG								0000 0000	0000 0000
11Ah	TX1REG	TX1REG								0000 0000	0000 0000
11Bh	SP1BRGL	SP1BRGL								0000 0000	0000 0000
11Ch	SP1BRGH	SP1BRGH								0000 0000	0000 0000
11Dh	RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 0000	0000 0000
11Eh	TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
11Fh	BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00

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

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

TABLE 4-12: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 PIC16(L)F19195/6/7 (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 15											
CPU CORE REGISTERS; see Table 4-3 for specifics											
78Ch	—	Unimplemented								---- ----	---- ----
78Dh	—	Unimplemented								---- ----	---- ----
78Eh	—	Unimplemented								---- ----	---- ----
78Fh	—	Unimplemented								---- ----	---- ----
790h	—	Unimplemented								---- ----	---- ----
791h	—	Unimplemented								---- ----	---- ----
792h	—	Unimplemented								---- ----	---- ----
793h	—	Unimplemented								---- ----	---- ----
794h	—	Unimplemented								---- ----	---- ----
795h	—	Unimplemented								---- ----	---- ----
796h	PMD0	SYSCMD	FVRMD	ACTMD	—	—	NVMMD	—	IOCMD	000- -0-0	000- -0-0
797h	PMD1	—	—	—	TMR4MD	—	TMR2MD	TMR1MD	TMR0MD	---0 -000	0000 0000
798h	PMD2	RTCCMD	DACMD	ADCMD	—	—	CMP2MD	CMP1MD	ZCDMD	000- -000	000- -000
799h	PMD3	—	—	—	—	PWM4MD	PWM3MD	CCP2MD	CCP1MD	0000 0000	0000 0000
79Ah	PMD4	UART2MD	UART1MD	—	MSSP1MD	—	—	—	CWG1MD	00-0 ---0	00-0 ---0
79Bh	PMD5	—	SMT1MD	LCDMD	CLC4MD	CLC3MD	CLC2MD	CLC1MD	—	-000 000-	-000 000-
79Ch	—	Unimplemented								---- ----	---- ----
79Dh	—	Unimplemented								---- ----	---- ----
79Eh	—	Unimplemented								---- ----	---- ----
79Fh	—	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'.

TABLE 4-12: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 PIC16(L)F19195/6/7 (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
Banks 30-57											
CPU CORE REGISTERS; see Table 4-3 for specifics											
F0Ch — 1C9Fh	—	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'.

TABLE 4-12: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 PIC16(L)F19195/6/7 (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 63											
CPU CORE REGISTERS; see Table 4-3 for specifics											
1F8Ch	—	Unimplemented								---- ----	---- ----
1F8Dh	—	Unimplemented								---- ----	---- ----
1F8Eh	—	Unimplemented								---- ----	---- ----
1F8Fh	—	Unimplemented								---- ----	---- ----
1F90h	—	Unimplemented								---- ----	---- ----
1F91h	—	Unimplemented								---- ----	---- ----
1F92h	—	Unimplemented								---- ----	---- ----
1F93h	—	Unimplemented								---- ----	---- ----
1F94h	—	Unimplemented								---- ----	---- ----
1F95h	—	Unimplemented								---- ----	---- ----
1F96h	—	Unimplemented								---- ----	---- ----
1F97h	—	Unimplemented								---- ----	---- ----
1F98h	—	Unimplemented								---- ----	---- ----
1F99h	—	Unimplemented								---- ----	---- ----
1F9Ah	—	Unimplemented								---- ----	---- ----
1F9Bh	—	Unimplemented								---- ----	---- ----
1F9Ch	—	Unimplemented								---- ----	---- ----
1F9Dh	—	Unimplemented								---- ----	---- ----
1F9Eh	—	Unimplemented								---- ----	---- ----
1F9Fh	—	Unimplemented								---- ----	---- ----
1FA0h	—	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'.

8.12 Start-up Sequence

Upon the release of a POR or BOR, the following must occur before the device will begin executing:

1. Power-up Timer runs to completion (if enabled).
2. Oscillator start-up timer runs to completion (if required for oscillator source).
3. $\overline{\text{MCLR}}$ must be released (if enabled).

The total time-out will vary based on oscillator configuration and Power-up Timer Configuration. See **Section 9.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for more information.

The Power-up Timer runs independently of $\overline{\text{MCLR}}$ Reset. If $\overline{\text{MCLR}}$ is kept low long enough, the Power-up Timer and oscillator start-up timer will expire. This is useful for testing purposes or to synchronize more than one device operating in parallel.

8.13 Memory Execution Violation

A memory execution violation Reset occurs if executing an instruction fetched from outside the valid execution area. There are different cases that define the Valid Execution area. If Storage Area Flash (SAF) is enabled (**Section 4.2.2 "Boot Block"**), the SAF area (Table 4-2) is not available for program execution.

Table 4-1 shows the addresses available on a device variant based on user Flash size. Execution outside this region generates a memory execution violation.

Prefetched instructions that are not executed do not cause memory execution violations. For example, a `GOTO` instruction in the last memory location will prefetch from an invalid location; this is not an error. If an instruction from an invalid location tries to execute, the memory violation is generated immediately, and any concurrent interrupt requests are not honored. When a memory execution violation is generated, the device is reset and flag $\overline{\text{MEMV}}$ is cleared in PCON1 (Register 8-3) to signal the cause. The flag needs to be set in code after a memory execution violation.

FIGURE 8-3: RESET START-UP SEQUENCE

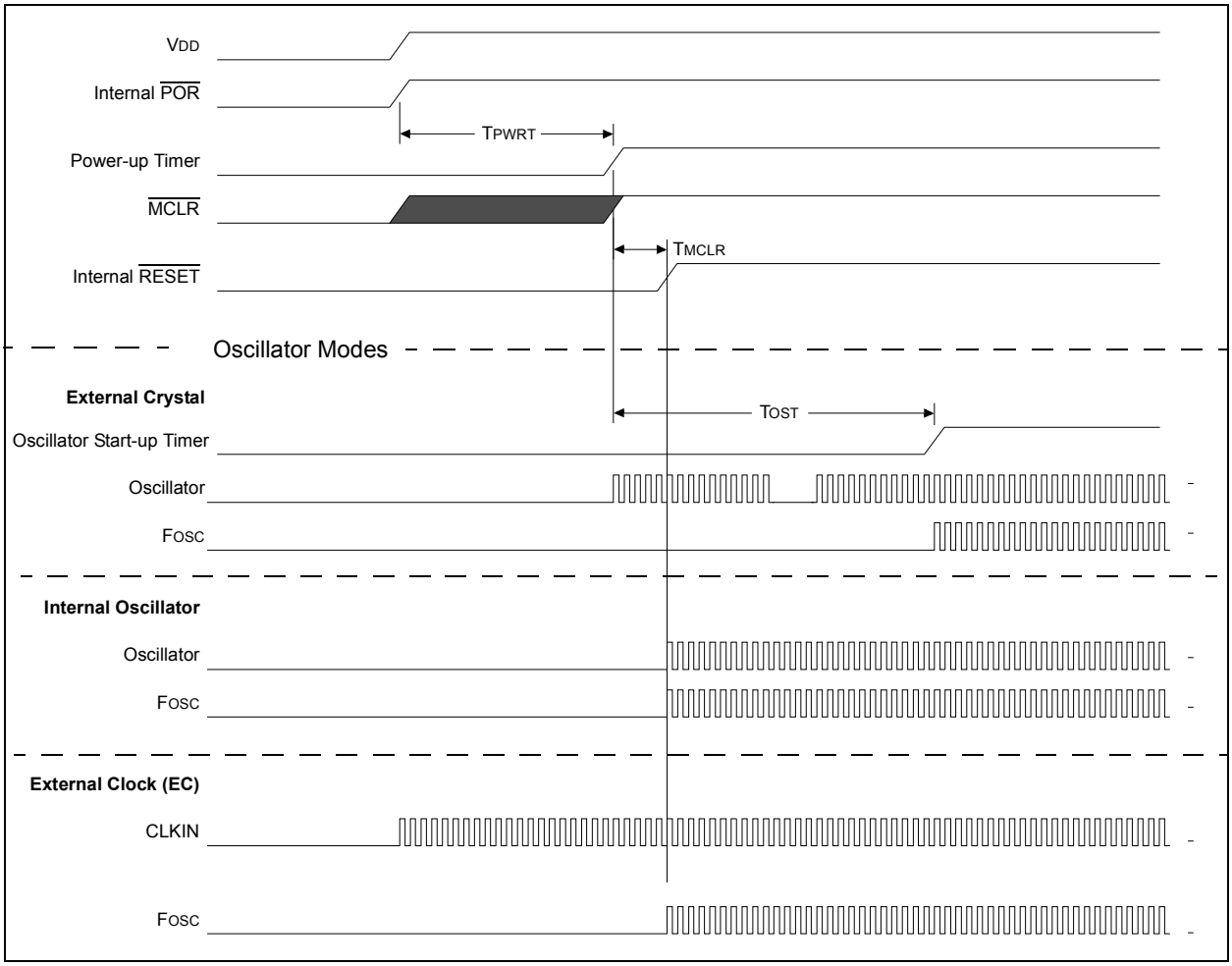


TABLE 9-1: NOSC/COSC BIT SETTINGS

NOSC<2:0>/ COSC<2:0>	Clock Source
111	EXTOSC(1)
110	HFINTOSC(2)
101	LFINTOSC
100	SOSC
011	Reserved (operates like NOSC = 110)
010	EXTOSC with 4x PLL(1)
001	HFINTOSC with 2x PLL(1)
000	Reserved (operates like NOSC = 110)

Note 1: EXTOSC configured by the FEXTOSC bits of Configuration Word 1 (Register 5-1).

2: HFINTOSC settings are configured with the HFFRQ bits of the OSCFRQ register (Register 9-6).

TABLE 9-2: NDIV/CDIV BIT SETTINGS

NDIV<3:0>/ CDIV<3:0>	Clock divider
1111-1010	Reserved
1001	512
1000	256
0111	128
0110	64
0101	32
0100	16
0011	8
0010	4
0001	2
0000	1

REGISTER 10-18: PIR7: PERIPHERAL INTERRUPT REQUEST REGISTER 7

U-0	U-0	R/W/HS-0/0	U-0	U-0	U-0	U-0	R/W/HS-0/0
—	—	NVMIF	—	—	—	—	CWG1IF
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 7-6 **Unimplemented:** Read as '0'

bit 5 **NVMIF:** Nonvolatile Memory (NVM) Interrupt Flag bit
1 = The requested NVM operation has completed
0 = NVM interrupt not asserted

bit 4-1 **Unimplemented:** Read as '0'

bit 0 **CWG1IF:** CWG1 Interrupt Flag bit
1 = CWG1 has gone into shutdown
0 = CWG1 is operating normally, or interrupt cleared

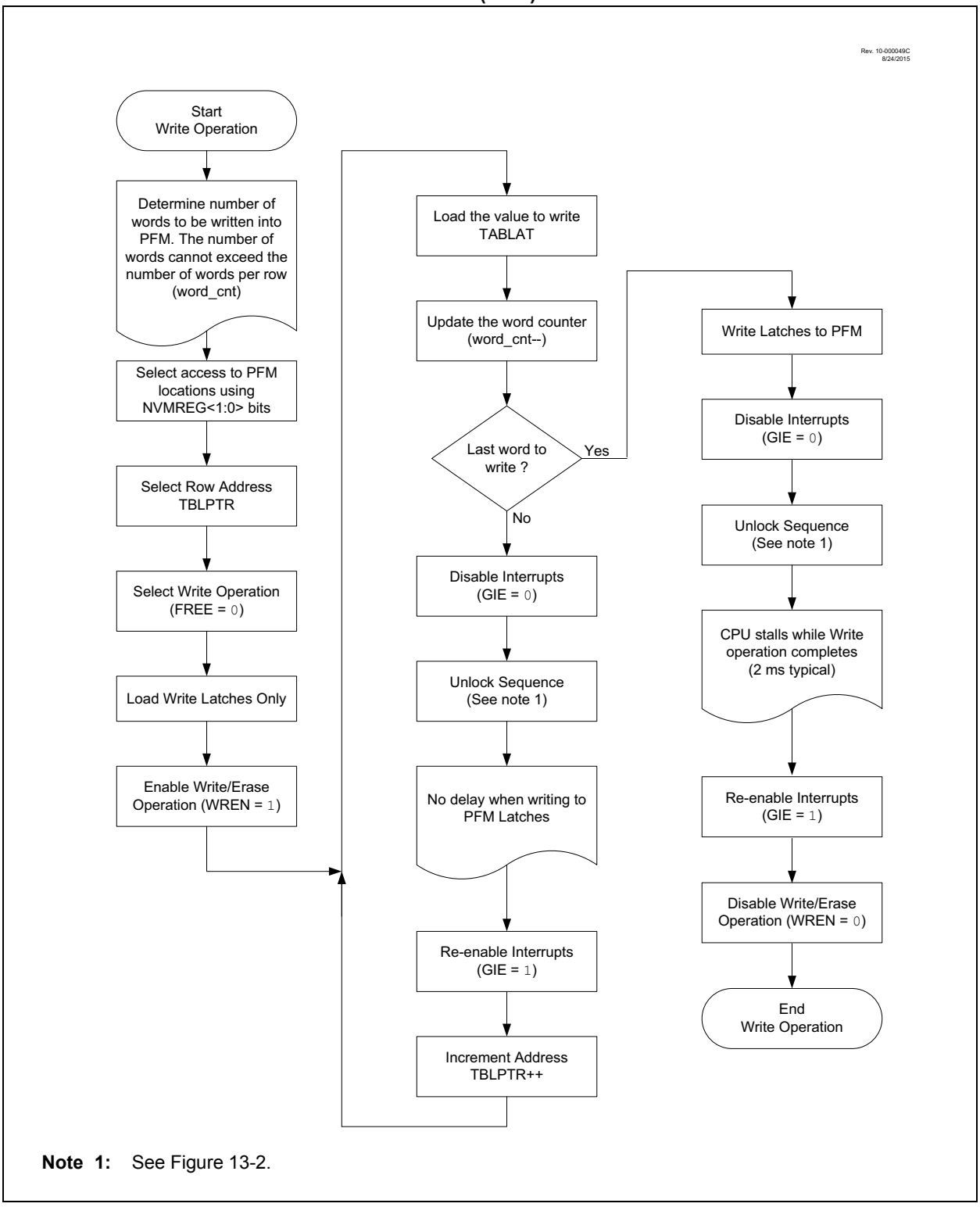
Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

TABLE 10-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	147
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	148
PIE1	OSFIE	CSWIE	—	—	—	—	ADTIE	ADIE	149
PIE2	—	ZCDIE	—	—	—	—	C2IE	C1IE	150
PIE3	RC2IE	TX2IE	RC1IE	TX1IE	—	—	BCL1IE	SSP1IE	151
PIE4	—	—	—	—	TMR4IE	—	TMR2IE	TMR1IE	152
PIE5	CLC4IE	CLC3IE	CLC2IE	CLC1IE	—	—	—	TMR1GIE	153
PIE6	CRIE	—	—	—	—	—	CCP2IE	CCP1IE	154
PIE7	—	—	NVMIE	—	—	—	—	CWG1IE	155
PIE8	LCDIE	RTCCIE	—	—	—	SMT1PWAIE	SMT1PRAIE	SMT1IE	156
PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	157
PIR1	OSFIF	CSWIF	—	—	—	—	ADTIF	ADIF	158
PIR2	—	ZCDIF	—	—	—	—	C2IF	C1IF	159
PIR3	RC2IF	TX2IF	RC1IF	TX1IF	—	—	BCL1IF	SSP1IF	160
PIR4	—	—	—	—	TMR4IF	—	TMR2IF	TMR1IF	161
PIR5	CLC4IF	CLC3IF	CLC2IF	CLC1IF	—	—	—	TMR1GIF	162
PIR6	CRIF	—	—	—	—	—	CCP2IF	CCP1IF	163
PIR7	—	—	NVMIF	—	—	—	—	CWG1IF	164
PIR8	LCDIF	RTCCIF	—	—	—	SMT1PWAIF	SMT1PRAIF	SMT1IF	165

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

FIGURE 13-5: PROGRAM FLASH MEMORY (PFM) WRITE FLOWCHART



Note 1: See Figure 13-2.

REGISTER 14-38: SLRCONE: PORTE SLEW RATE CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1
SLRE7	SLRE6	SLRE5	SLRE4	SLRE3	—	SLRE1	SLRE0
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 7-3 **SLRE<7:3>:** PORTE Slew Rate Enable bits
For RE<7:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

bit 2 **Unimplemented:** Read as '0'

bit 1-0 **SLRE<1:0>:** PORTE Slew Rate Enable bits
For RE<1:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

REGISTER 14-39: INLVLE: PORTE INPUT LEVEL CONTROL REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1
INLVLE7	INLVLE6	INLVLE5	INLVLE4	INLVLE3	—	INLVLE1	INLVLE0
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 7-3 **INLVLE<7:3>:** PORTE Input Level Select bits
For RE<7:3> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

bit 2 **Unimplemented:** Read as '0'

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

15.3 Bidirectional Pins

PPS selections for peripherals with bidirectional signals on a single pin must be made so that the PPS input and PPS output select the same pin. Peripherals that have bidirectional signals include:

- EUSART (synchronous operation)
- MSSP (I^2C)

Note:	The I^2C SCLx and SDAx functions can be remapped through PPS. However, only the RB1, RB2, RC3 and RC4 pins have the I^2C and SMBus specific input buffers implemented (I^2C mode disables INLVL and sets thresholds that are specific for I^2C). If the SCLx or SDAx functions are mapped to some other pin (other than RB1, RB2, RC3 or RC4), the general purpose TTL or ST input buffers (as configured based on INLVL register setting) will be used instead. In most applications, it is therefore recommended only to map the SCLx and SDAx pin functions to the RB1, RB2, RC3 or RC4 pins.

15.4 PPS Lock

The PPS includes a mode in which all input and output selections can be locked to prevent inadvertent changes. PPS selections are locked by setting the PPSLOCKED bit of the PPSLOCK register. Setting and clearing this bit requires a special sequence as an extra precaution against inadvertent changes. Examples of setting and clearing the PPSLOCKED bit are shown in Example 15-1.

EXAMPLE 15-1: PPS LOCK/UNLOCK SEQUENCE

```
; suspend interrupts
    BCF     INTCON,GIE
;   BANKSEL PPSLOCK     ; set bank
; required sequence, next 5 instructions
    MOVLW   0x55
    MOVWF   PPSLOCK
    MOVLW   0xAA
    MOVWF   PPSLOCK
; Set PPSLOCKED bit to disable writes or
; Clear PPSLOCKED bit to enable writes
    BSF     PPSLOCK,PPSLOCKED
; restore interrupts
    BSF     INTCON,GIE
```

15.5 PPS Permanent Lock

The PPS can be permanently locked by setting the PPS1WAY Configuration bit. When this bit is set, the PPSLOCKED bit can only be cleared and set one time after a device Reset. This allows for clearing the PPSLOCKED bit so that the input and output selections can be made during initialization. When the PPSLOCKED bit is set after all selections have been made, it will remain set and cannot be cleared until after the next device Reset event.

15.6 Operation During Sleep

PPS input and output selections are unaffected by Sleep.

15.7 Effects of a Reset

A device Power-on-Reset (POR) clears all PPS input and output selections to their default values (Permanent Lock Removed). All other Resets leave the selections unchanged. Default input selections are shown in Table 15-1 and Table 15-2.

19.5.5 BURST AVERAGE MODE

The Burst Average mode (ADMD = 011) acts the same as the Average mode in most respects. The one way it differs is that it continuously retriggers ADC sampling until the CNT value is greater than or equal to RPT, even if Continuous Sampling mode (see **Section 19.5.8 "Continuous Sampling mode"**) is not enabled. This allows for a threshold comparison on the average of a short burst of ADC samples.

19.5.6 LOW-PASS FILTER MODE

The Low-Pass Filter mode (ADMD = 100) acts similarly to the Average mode in how it handles samples (accumulates samples until CNT value is greater than or equal to RPT, then triggers threshold comparison. CNT does not reset once it is greater or equal to RPT. Thus CNT will be greater than RPT for all subsequent samples until CNT is reset by the user), but instead of a simple average, it performs a low-pass filter operation on all of the samples, reducing the effect of high-frequency noise on the average, then performs a threshold comparison on the results. (see Table 19-2 for a more detailed description of the mathematical operation). In this mode, the ADCRS bits determine the cut-off frequency of the low-pass filter (as demonstrated by Table 19-3).

19.5.7 THRESHOLD COMPARISON

At the end of each computation:

- The conversion results are latched and held stable at the end-of-conversion.
- The error is calculated based on a difference calculation which is selected by the ADCALC<2:0> bits in the ADCON3 register. The value can be one of the following calculations (see Register 19-4 for more details):
  - The first derivative of single measurements
  - The CVD result in CVD mode
  - The current result vs. a setpoint
  - The current result vs. the filtered/average result
  - The first derivative of the filtered/average value
  - Filtered/average value vs. a setpoint
- The result of the calculation (ERR) is compared to the upper and lower thresholds, UTH<ADUTHH:ADUTHL> and LTH<ADLTHH:ADLTHL> registers, to set the ADUTHR and ADLTHR flag bits. The threshold logic is selected by ADTMD<2:0> bits in the ADCON3 register. The threshold trigger option can be one of the following:
  - Never interrupt
  - Error is less than lower threshold
  - Error is greater than or equal to lower threshold
  - Error is between thresholds (inclusive)
  - Error is outside of thresholds
  - Error is less than or equal to upper threshold
  - Error is greater than upper threshold
  - Always interrupt regardless of threshold test results
  - If the threshold condition is met, the threshold interrupt flag ADTIF is set.

Note 1:	The threshold tests are signed operations.
2:	If OV is set, a threshold interrupt is signaled.

27.5 Operation Examples

Unless otherwise specified, the following notes apply to the following timing diagrams:

- Both the prescaler and postscaler are set to 1:1 (both the CKPS and OUTPS bits in the TxCON register are cleared).
- The diagrams illustrate any clock except Fosc/4 and show clock-sync delays of at least two full cycles for both ON and Timer2/4_ers. When using Fosc/4, the clock-sync delay is at least one instruction period for Timer2/4_ers; ON applies in the next instruction period.
- The PWM Duty Cycle and PWM output are illustrated assuming that the timer is used for the PWM function of the CCP module as described in **Section 29.0 "Capture/Compare/PWM Modules"**. The signals are not a part of the Timer2/4 module.

27.5.1 SOFTWARE GATE MODE

This mode corresponds to legacy Timer2/4 operation. The timer increments with each clock input when ON = 1 and does not increment when ON = 0. When the TMRx count equals the PRx period count the timer resets on the next clock and continues counting from 0. Operation with the ON bit software controlled is illustrated in Figure 27-4. With PRx = 5, the counter advances until TMRx = 5, and goes to zero with the next clock.

28.7.10 GATED COUNTER MODE

Gated Counter mode counts pulses on the SMTx_signal input, gated by the SMTxWIN input. It begins incrementing the timer upon seeing a rising edge of the SMTxWIN input and updates the SMTxCPW register upon a falling edge on the SMTxWIN input. See Figure 28-19 and Figure 28-20.

FIGURE 28-19: GATED COUNTER MODE REPEAT ACQUISITION TIMING DIAGRAM

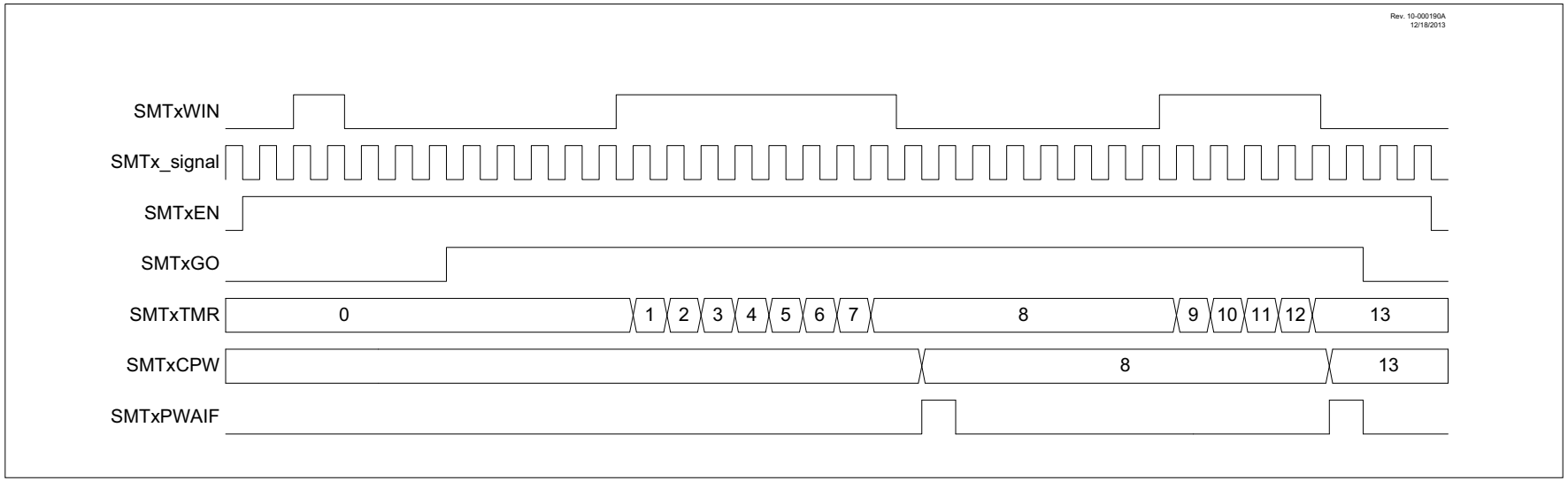
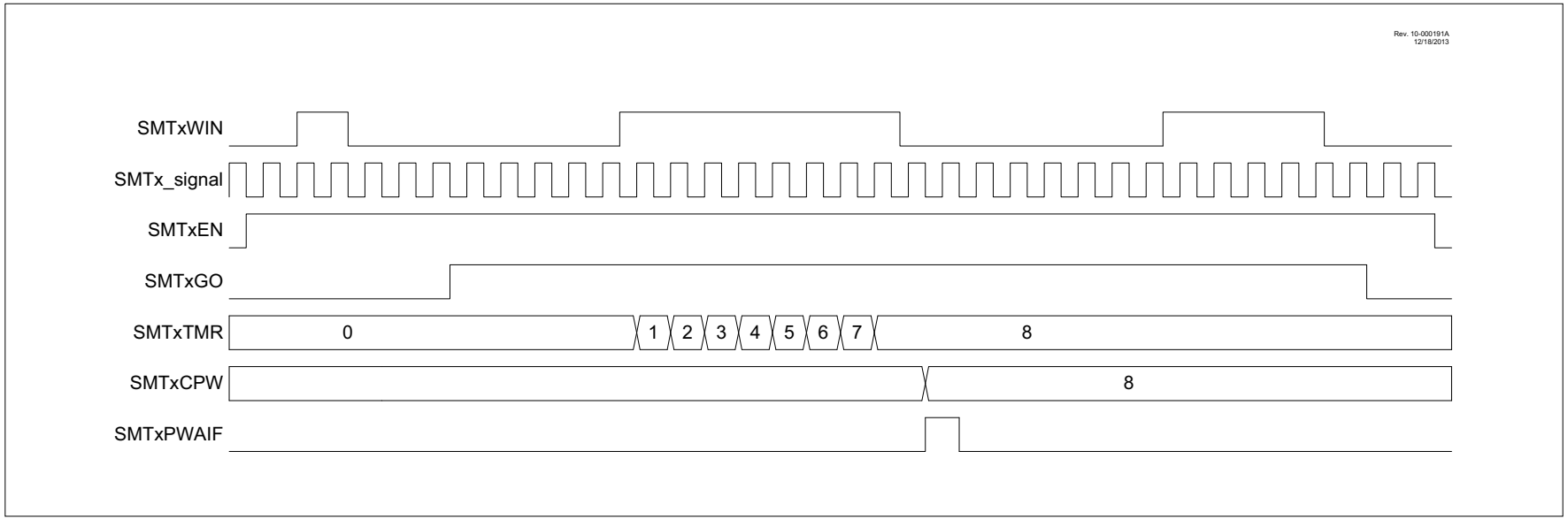


FIGURE 28-20: GATED COUNTER MODE SINGLE ACQUISITION TIMING DIAGRAM



REGISTER 29-2: CCPxCAP: CAPTURE INPUT SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/x	R/W-0/x	R/W-0/x
—	—	—	—	—	CTS<2:0>		
bit 7							bit 0

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

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **CTS<2:0>:** Capture Trigger Input Selection bits

CTS	CCP1.capture	CCP2.capture
1000	RTCC_seconds	
0111	LC4_out	
0110	LC3_out	
0101	LC2_out	
0100	LC1_out	
0011	IOC_interrupt	
0010	C2OUT	
0001	C1OUT	
0000	CCP1PPS	CCP2PPS

REGISTER 29-3: CCPRxL REGISTER: CCPx REGISTER LOW BYTE

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
CCPRx<7:0>							
bit 7							bit 0

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

bit 7-0

CCPxMODE = Capture mode
CCPRxL<7:0>: Capture value of TMR1L
CCPxMODE = Compare mode
CCPRxL<7:0>: LS Byte compared to TMR1L
CCPxMODE = PWM modes when CCPxFMT = 0:
CCPRxL<7:0>: Pulse-width Least Significant eight bits
CCPxMODE = PWM modes when CCPxFMT = 1:
CCPRxL<7:6>: Pulse-width Least Significant two bits
CCPRxL<5:0>: Not used.

REGISTER 31-6: CWG1AS1: CWG1 AUTO-SHUTDOWN CONTROL REGISTER 1

U-1	U-1	U-1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E
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 7-5 **Unimplemented:** Read as '0'

bit 4 **AS4E:** CLC2 Output bit
1 = LC2_out shut-down is enabled
0 = LC2_out shut-down is disabled

bit 3 **AS3E:** Comparator C2 Output bit
1 = C2 output shut-down is enabled
0 = C2 output shut-down is disabled

bit 2 **AS2E:** Comparator C1 Output bit
1 = C1 output shut-down is enabled
0 = C1 output shut-down is disabled

bit 2 **AS1E:** TMR2 Postscale Output bit
1 = TMR2 Postscale shut-down is enabled
0 = TMR2 Postscale shut-down is disabled

bit 0 **AS0E:** CWG1 Input Pin bit
1 = Input pin selected by CWG1PPS shut-down is enabled
0 = Input pin selected by CWG1PPS shut-down is disabled

33.7 BAUD RATE GENERATOR

The MSSP module has a Baud Rate Generator available for clock generation in both I^2C and SPI Master modes. The Baud Rate Generator (BRG) reload value is placed in the SSPxADD register (Register 33-6). When a write occurs to SSPxBUF, the Baud Rate Generator will automatically begin counting down.

Once the given operation is complete, the internal clock will automatically stop counting and the clock pin will remain in its last state.

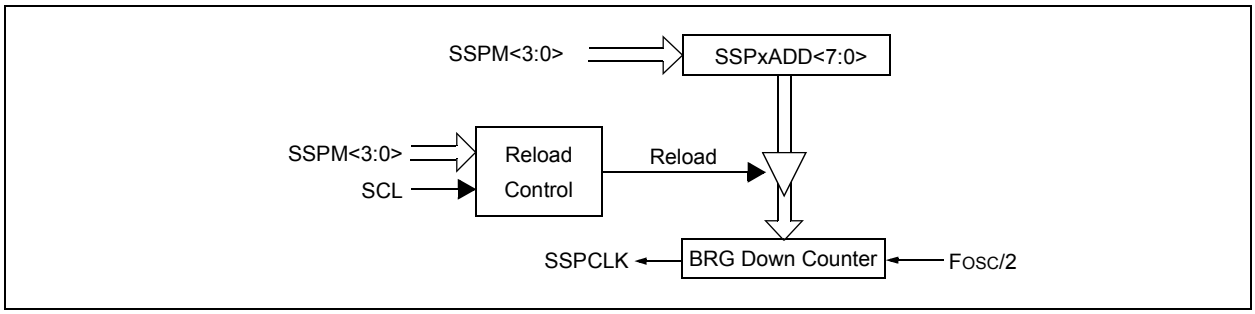
An internal signal “Reload” in Figure 33-40 triggers the value from SSPxADD to be loaded into the BRG counter. This occurs twice for each oscillation of the module clock line. The logic dictating when the reload signal is asserted depends on the mode the MSSP is being operated in.

Table 33-4 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPxADD.

EQUATION 33-1:

$$F_{CLOCK} = \frac{F_{OSC}}{(SSP1ADD + 1)(4)}$$

FIGURE 33-40: BAUD RATE GENERATOR BLOCK DIAGRAM



Note: Values of `0x00`, `0x01` and `0x02` are not valid for SSPxADD when used as a Baud Rate Generator for I^2C. This is an implementation limitation.

TABLE 33-2: MSSP CLOCK RATE W/BRG

FOSC	FCY	BRG Value	FCLOCK (2 Rollovers of BRG)
32 MHz	8 MHz	13h	400 kHz
32 MHz	8 MHz	19h	308 kHz
32 MHz	8 MHz	4Fh	100 kHz
16 MHz	4 MHz	09h	400 kHz
16 MHz	4 MHz	0Ch	308 kHz
16 MHz	4 MHz	27h	100 kHz
4 MHz	1 MHz	09h	100 kHz

Note: Refer to the I/O port electrical specifications in Table 39-4 to ensure the system is designed to support I/O requirements.

34.1.2.8 Asynchronous Reception Setup:

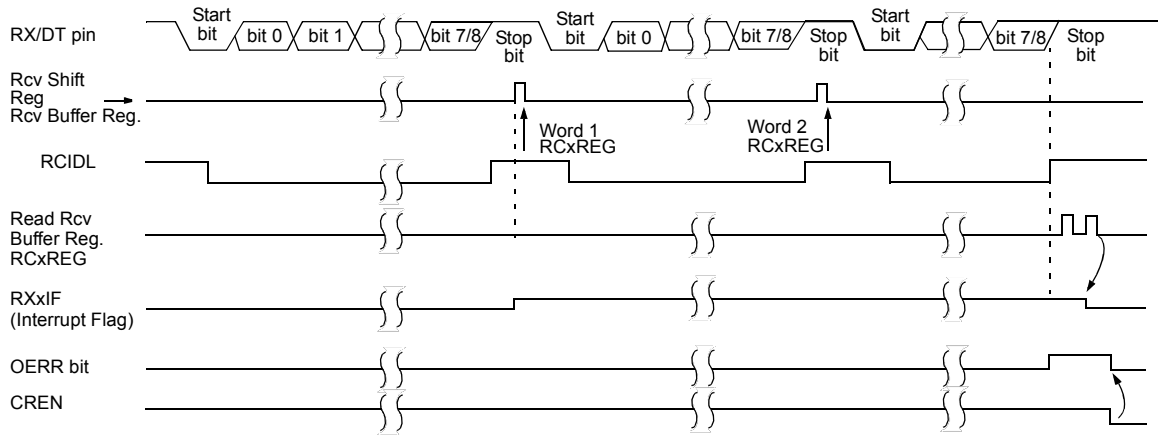
1. Initialize the SPxBRGH, SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 34.3 "EUSART Baud Rate Generator (BRG)"**).
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
4. If interrupts are desired, set the RXxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
5. If 9-bit reception is desired, set the RX9 bit.
6. Enable reception by setting the CREN bit.
7. The RXxIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RXxIE interrupt enable bit was also set.
8. Read the RCxSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
9. Get the received eight Least Significant data bits from the receive buffer by reading the RCxREG register.
10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

34.1.2.9 9-bit Address Detection Mode Setup

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

1. Initialize the SPxBRGH, SPxBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 34.3 "EUSART Baud Rate Generator (BRG)"**).
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
4. If interrupts are desired, set the RXxIE bit of the PIE3 register and the GIE and PEIE bits of the INTCON register.
5. Enable 9-bit reception by setting the RX9 bit.
6. Enable address detection by setting the ADDEN bit.
7. Enable reception by setting the CREN bit.
8. The RXxIF interrupt flag bit will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RXxIE interrupt enable bit was also set.
9. Read the RCxSTA register to get the error flags. The ninth data bit will always be set.
10. Get the received eight Least Significant data bits from the receive buffer by reading the RCxREG register. Software determines if this is the device's address.
11. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
12. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.

FIGURE 34-5: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RCxREG (receive buffer) is read after the third word, causing the OERR (overrun) bit to be set.

REGISTER 35-8: LCDREF: LCD REFERENCE VOLTAGE/CONTRAST CONTROL REGISTER

R-0	R-0	R-0	R-0	R-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	LCDCST2	LCDCST1	LCDCST0
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 7-3 **Unimplemented:** Read as '0'.

bit 2-0 **LCDCST<2:0>**: LCD Contrast Control bits(1)

Selects the resistance of the LCD contrast control resistor ladder

Bit Value	Resistor ladder
000 =	Contrast Control Bypassed (Maximum contrast).
001 =	Contrast Control Resistor ladder is at 1/7th of maximum resistance
010 =	Contrast Control Resistor ladder is at 2/7th of maximum resistance
011 =	Contrast Control Resistor ladder is at 3/7th of maximum resistance
100 =	Contrast Control Resistor ladder is at 4/7th of maximum resistance
101 =	Contrast Control Resistor ladder is at 5/7th of maximum resistance
110 =	Contrast Control Resistor ladder is at 6/7th of maximum resistance
111 =	Contrast Control Resistor ladder is at maximum resistance (Minimum contrast)

Note 1: This setting is only valid in Internal Resistance Ladder only modes.