



Welcome to [E-XFL.COM](https://www.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	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1826-e-mv

3.0 MEMORY ORGANIZATION

There are three types of memory in PIC16(L)F1826/27: Data Memory, Program Memory and Data EEPROM Memory⁽¹⁾.

- Program Memory
- Data Memory
 - Core Registers
 - Special Function Registers
 - General Purpose RAM
 - Common RAM
 - Device Memory Maps
 - Special Function Registers Summary
- Data EEPROM memory⁽¹⁾

Note 1: The Data EEPROM Memory and the method to access Flash memory through the EECON registers is described in **Section 11.0 “Data EEPROM and Flash Program Memory Control”**.

The following features are associated with access and control of program memory and data memory:

- PCL and PCLATH
- Stack
- Indirect Addressing

3.1 Program Memory Organization

The enhanced mid-range core has a 15-bit program counter capable of addressing a 32K x 14 program memory space. Table 3-1 shows the memory sizes implemented for the PIC16(L)F1826/27 family. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figures 3-1 and 3-2).

TABLE 3-1: DEVICE SIZES AND ADDRESSES

Device	Program Memory Space (Words)	Last Program Memory Address
PIC16(L)F1826	2,048	07FFh
PIC16(L)F1827	4,096	0FFFh

PIC16(L)F1826/27

7.10 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON register are updated to indicate the cause of the Reset. Table 7-3 and Table 7-4 show the Reset conditions of these registers.

TABLE 7-3: RESET STATUS BITS AND THEIR SIGNIFICANCE

STKOVF	STKUNF	RMCLR	RI	POR	BOR	TO	PD	Condition
0	0	1	1	0	x	1	1	Power-on Reset
0	0	1	1	0	x	0	x	Illegal, \overline{TO} is set on \overline{POR}
0	0	1	1	0	x	x	0	Illegal, \overline{PD} is set on \overline{POR}
0	0	1	1	u	0	1	1	Brown-out Reset
u	u	u	u	u	u	0	u	WDT Reset
u	u	u	u	u	u	0	0	WDT Wake-up from Sleep
u	u	u	u	u	u	1	0	Interrupt Wake-up from Sleep
u	u	0	u	u	u	u	u	\overline{MCLR} Reset during normal operation
u	u	0	u	u	u	1	0	\overline{MCLR} Reset during Sleep
u	u	u	0	u	u	u	u	RESET Instruction Executed
1	u	u	u	u	u	u	u	Stack Overflow Reset (STVREN = 1)
u	1	u	u	u	u	u	u	Stack Underflow Reset (STVREN = 1)

TABLE 7-4: RESET CONDITION FOR SPECIAL REGISTERS⁽²⁾

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	0000h	---1 1000	00-- 110x
\overline{MCLR} Reset during normal operation	0000h	---u uuuu	uu-- 0uuu
\overline{MCLR} Reset during Sleep	0000h	---1 0uuu	uu-- 0uuu
WDT Reset	0000h	---0 uuuu	uu-- uuuu
WDT Wake-up from Sleep	PC + 1	---0 0uuu	uu-- uuuu
Brown-out Reset	0000h	---1 1uuu	00-- 11u0
Interrupt Wake-up from Sleep	PC + 1 ⁽¹⁾	---1 0uuu	uu-- uuuu
RESET Instruction Executed	0000h	---u uuuu	uu-- u0uu
Stack Overflow Reset (STVREN = 1)	0000h	---u uuuu	1u-- uuuu
Stack Underflow Reset (STVREN = 1)	0000h	---u uuuu	u1-- uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

Note 1: When the wake-up is due to an interrupt and Global Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

2: If a Status bit is not implemented, that bit will be read as '0'.

PIC16(L)F1826/27

10.1 Independent Clock Source

The WDT derives its time base from the 31 kHz LFINTOSC internal oscillator. Time intervals in this chapter are based on a nominal interval of 1 ms. See **Section 30.0 “Electrical Specifications”** for the LFINTOSC tolerances.

10.2 WDT Operating Modes

The Watchdog Timer module has four operating modes controlled by the WDTE<1:0> bits in Configuration Word 1. See Table 10-1.

10.2.1 WDT IS ALWAYS ON

When the WDTE bits of Configuration Word 1 are set to ‘11’, the WDT is always on.

WDT protection is active during Sleep.

10.2.2 WDT IS OFF IN SLEEP

When the WDTE bits of Configuration Word 1 are set to ‘10’, the WDT is on, except in Sleep.

WDT protection is not active during Sleep.

10.2.3 WDT CONTROLLED BY SOFTWARE

When the WDTE bits of Configuration Word 1 are set to ‘01’, the WDT is controlled by the SWDTEN bit of the WDTCON register.

WDT protection is unchanged by Sleep. See Table 10-1 for more details.

TABLE 10-1: WDT OPERATING MODES

WDTE<1:0>	SWDTEN	Device Mode	WDT Mode
11	X	X	Active
10	X	Awake	Active
		Sleep	Disabled
01	1	X	Active
	0		Disabled
00	X	X	Disabled

TABLE 10-2: WDT CLEARING CONDITIONS

Conditions	WDT
WDTE<1:0> = 00	Cleared
WDTE<1:0> = 01 and SWDTEN = 0	
WDTE<1:0> = 10 and enter Sleep	
CLRWDT Command	
Oscillator Fail Detected	
Exit Sleep + System Clock = T1OSC, EXTRC, INTOSC, EXTCLK	
Exit Sleep + System Clock = XT, HS, LP	Cleared until the end of OST
Change INTOSC divider (IRCF bits)	Unaffected

10.3 Time-Out Period

The WDTPS bits of the WDTCON register set the time-out period from 1 ms to 256 seconds (nominal). After a Reset, the default time-out period is 2 seconds.

10.4 Clearing the WDT

The WDT is cleared when any of the following conditions occur:

- Any Reset
- CLRWDT instruction is executed
- Device enters Sleep
- Device wakes up from Sleep
- Oscillator fail event
- WDT is disabled
- Oscillator Start-up Timer (OST) is running

See Table 10-2 for more information.

10.5 Operation During Sleep

When the device enters Sleep, the WDT is cleared. If the WDT is enabled during Sleep, the WDT resumes counting.

When the device exits Sleep, the WDT is cleared again. The WDT remains clear until the OST, if enabled, completes. See **Section 5.0 “Oscillator Module (With Fail-Safe Clock Monitor)”** for more information on the OST.

When a WDT time-out occurs while the device is in Sleep, no Reset is generated. Instead, the device wakes up and resumes operation. The TO and PD bits in the STATUS register are changed to indicate the event. See **Register 3-1** for more information.

11.3 Flash Program Memory Overview

It is important to understand the Flash program memory structure for erase and programming operations. Flash Program memory is arranged in rows. A row consists of a fixed number of 14-bit program memory words. A row is the minimum block size that can be erased by user software.

Flash program memory may only be written or erased if the destination address is in a segment of memory that is not write-protected, as defined in bits WRT<1:0> of Configuration Word 2.

After a row has been erased, the user can reprogram all or a portion of this row. Data to be written into the program memory row is written to 14-bit wide data write latches. These write latches are not directly accessible to the user, but may be loaded via sequential writes to the EEDATH:EEDATL register pair.

Note: If the user wants to modify only a portion of a previously programmed row, then the contents of the entire row must be read and saved in RAM prior to the erase.

The number of data write latches may not be equivalent to the number of row locations. During programming, user software may need to fill the set of write latches and initiate a programming operation multiple times in order to fully reprogram an erased row. For example, a device with a row size of 32 words and eight write latches will need to load the write latches with data and initiate a programming operation four times.

The size of a program memory row and the number of program memory write latches may vary by device. See Table 11-1 for details.

TABLE 11-1: FLASH MEMORY ORGANIZATION BY DEVICE

Device	Erase Block (Row) Size/ Boundary	Number of Write Latches/ Boundary
PIC16(L)F1826/27	32 words, EEADRL<4:0> = 00000	32 words, EEADRL<4:0> = 00000

11.3.1 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must:

1. Write the Least and Most Significant address bits to the EEADRH:EEADRL register pair.
2. Clear the CFGS bit of the EECON1 register.
3. Set the EEPGD control bit of the EECON1 register.
4. Then, set control bit RD of the EECON1 register.

Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "BSF EECON1, RD" instruction to be ignored. The data is available in the very next cycle, in the EEDATH:EEDATL register pair; therefore, it can be read as two bytes in the following instructions.

EEDATH:EEDATL register pair will hold this value until another read or until it is written to by the user.

Note 1: The two instructions following a program memory read are required to be NOPs. This prevents the user from executing a two-cycle instruction on the next instruction after the RD bit is set.

- 2: Flash program memory can be read regardless of the setting of the \overline{CP} bit.

NOTES:

22.0 TIMER2/4/6 MODULES

There are up to three identical Timer2-type modules available. To maintain pre-existing naming conventions, the Timers are called Timer2, Timer4 and Timer6 (also Timer2/4/6).

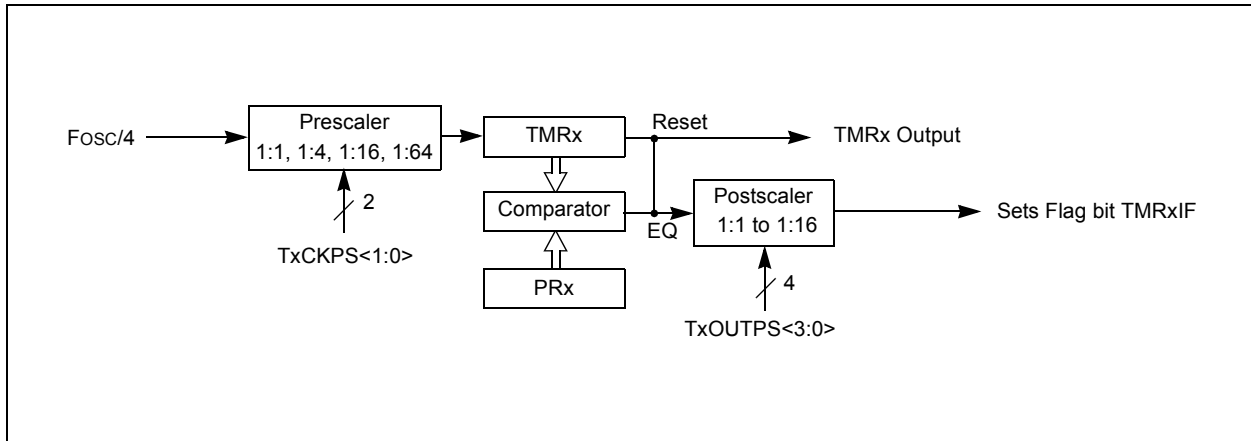
Note: The 'x' variable used in this section is used to designate Timer2, Timer4, or Timer6. For example, TxCON references T2CON, T4CON, or T6CON. PRx references PR2, PR4, or PR6.

The Timer2/4/6 modules incorporate the following features:

- 8-bit Timer and Period registers (TMRx and PRx, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16, and 1:64)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMRx match with PRx, respectively
- Optional use as the shift clock for the MSSPx modules (Timer2 only)

See Figure 22-1 for a block diagram of Timer2/4/6.

FIGURE 22-1: TIMER2/4/6 BLOCK DIAGRAM



23.0 DATA SIGNAL MODULATOR

The Data Signal Modulator (DSM) is a peripheral which allows the user to mix a data stream, also known as a modulator signal, with a carrier signal to produce a modulated output.

Both the carrier and the modulator signals are supplied to the DSM module either internally, from the output of a peripheral, or externally through an input pin.

The modulated output signal is generated by performing a logical “AND” operation of both the carrier and modulator signals and then provided to the MDOUT pin.

The carrier signal is comprised of two distinct and separate signals. A carrier high (CARH) signal and a carrier low (CARL) signal. During the time in which the modulator (MOD) signal is in a logic high state, the DSM mixes the carrier high signal with the modulator signal. When the modulator signal is in a logic low state, the DSM mixes the carrier low signal with the modulator signal.

Using this method, the DSM can generate the following types of Key Modulation schemes:

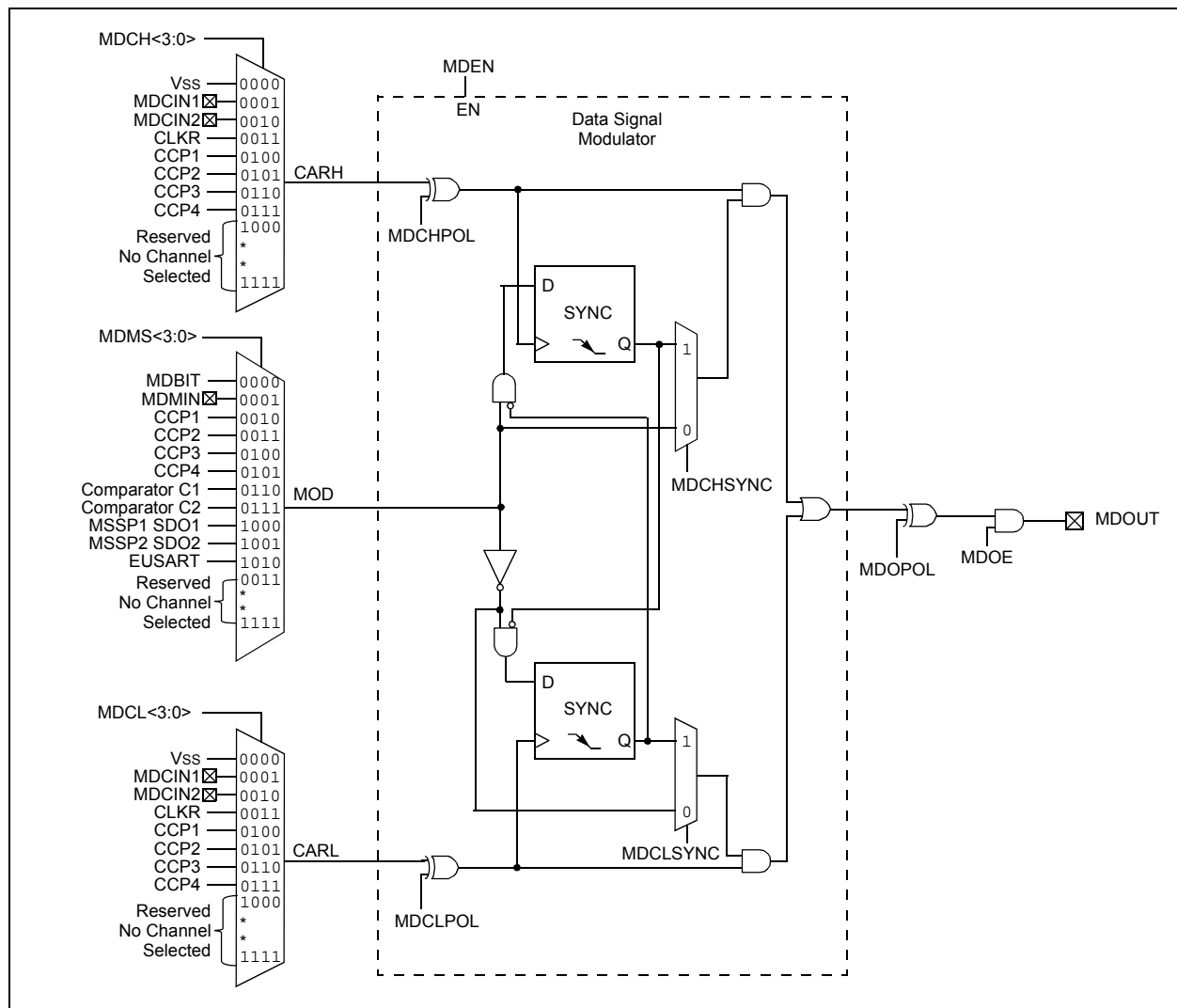
- Frequency-Shift Keying (FSK)
- Phase-Shift Keying (PSK)
- On-Off Keying (OOK)

Additionally, the following features are provided within the DSM module:

- Carrier Synchronization
- Carrier Source Polarity Select
- Carrier Source Pin Disable
- Programmable Modulator Data
- Modulator Source Pin Disable
- Modulated Output Polarity Select
- Slew Rate Control

Figure 23-1 shows a Simplified Block Diagram of the Data Signal Modulator peripheral.

FIGURE 23-1: SIMPLIFIED BLOCK DIAGRAM OF THE DATA SIGNAL MODULATOR



24.2.5 COMPARE DURING SLEEP

The Compare mode is dependent upon the system clock (Fosc) for proper operation. Since Fosc is shut down during Sleep mode, the Compare mode will not function properly during Sleep.

24.2.6 ALTERNATE PIN LOCATIONS

This module incorporates I/O pins that can be moved to other locations with the use of the alternate pin function registers, APFCON0 and APFCON1. To determine which pins can be moved and what their default locations are upon a reset, see **Section 12.1 “Alternate Pin Function”** for more information.

TABLE 24-4: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON0	RXDTSEL	SDO1SEL	SS1SEL	P2BSEL ⁽²⁾	CCP2SEL ⁽²⁾	P1DSEL	P1CSEL	CCP1SEL	119
CCPxCON	PxM1 ⁽¹⁾	PxM0 ⁽¹⁾	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	226
CCPRxL	Capture/Compare/PWM Register x Low Byte (LSB)								204*
CCPRxH	Capture/Compare/PWM Register x High Byte (MSB)								204*
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	170
CM1CON1	C1INTP	C1INTN	C1PCH1	C1PCH0	—	—	C1NCH1	C1NCH0	171
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	170
CM2CON1	C2INTP	C2INTN	C2PCH1	C2PCH0	—	—	C2NCH1	C2NCH0	171
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIE2	OSFIE	C2IE	C1IE	EEIE	BCL1IE	—	—	CCP2IE ⁽²⁾	88
PIE3 ⁽²⁾	—	—	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	89
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	91
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	—	—	CCP2IF ⁽²⁾	92
PIR3 ⁽²⁾	—	—	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	93
T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN \bar{C}	—	TMR1ON	185
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/DONE	T1GVAL	T1GSS1	T1GSS0	186
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								177*
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								177*
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	122
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	127

Legend: — = Unimplemented locations, read as '0'. Shaded cells are not used by Compare mode.

* Page provides register information.

Note 1: Applies to ECCP modules only.

2: PIC16(L)F1827 only.

PIC16(L)F1826/27

24.4.6.1 Steering Synchronization

The STRxSYNC bit of the PSTRxCON register gives the user two selections of when the steering event will happen. When the STRxSYNC bit is '0', the steering event will happen at the end of the instruction that writes to the PSTRxCON register. In this case, the output signal at the Px<D:A> pins may be an incomplete PWM waveform. This operation is useful when the user firmware needs to immediately remove a PWM signal from the pin.

When the STRxSYNC bit is '1', the effective steering update will happen at the beginning of the next PWM period. In this case, steering on/off the PWM output will always produce a complete PWM waveform.

Figures 24-19 and 24-20 illustrate the timing diagrams of the PWM steering depending on the STRxSYNC setting.

24.4.7 START-UP CONSIDERATIONS

When any PWM mode is used, the application hardware must use the proper external pull-up and/or pull-down resistors on the PWM output pins.

The CCPxM<1:0> bits of the CCPxCON register allow the user to choose whether the PWM output signals are active-high or active-low for each pair of PWM output pins (PxA/PxC and PxB/PxD). The PWM output polarities must be selected before the PWM pin output drivers are enabled. Changing the polarity

configuration while the PWM pin output drivers are enable is not recommended since it may result in damage to the application circuits.

The PxA, PxB, PxC and PxD output latches may not be in the proper states when the PWM module is initialized. Enabling the PWM pin output drivers at the same time as the Enhanced PWM modes may cause damage to the application circuit. The Enhanced PWM modes must be enabled in the proper Output mode and complete a full PWM cycle before enabling the PWM pin output drivers. The completion of a full PWM cycle is indicated by the TMRxIF bit of the PIRx register being set as the second PWM period begins.

Note: When the microcontroller is released from Reset, all of the I/O pins are in the high-impedance state. The external circuits must keep the power switch devices in the Off state until the microcontroller drives the I/O pins with the proper signal levels or activates the PWM output(s).

24.4.8 ALTERNATE PIN LOCATIONS

This module incorporates I/O pins that can be moved to other locations with the use of the alternate pin function registers, APFCON0 and APFCON1. To determine which pins can be moved and what their default locations are upon a reset, see **Section 12.1 "Alternate Pin Function"** for more information.

FIGURE 24-19: EXAMPLE OF STEERING EVENT AT END OF INSTRUCTION (STRxSYNC = 0)

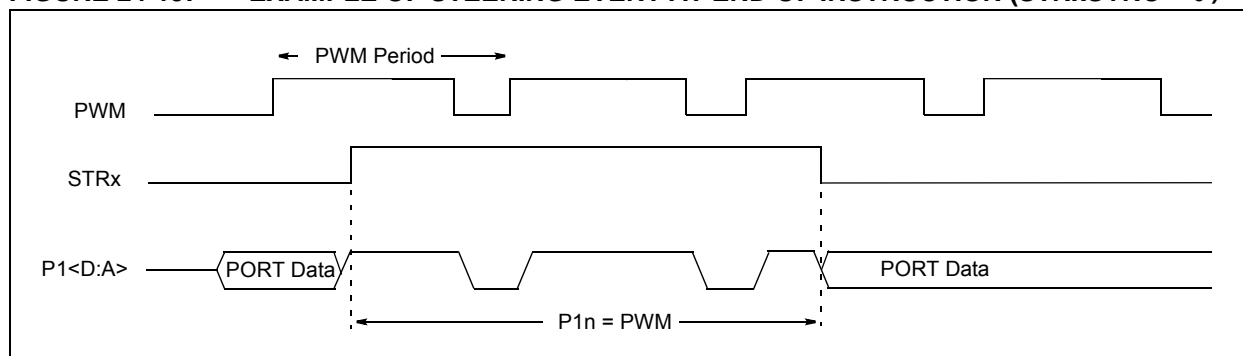
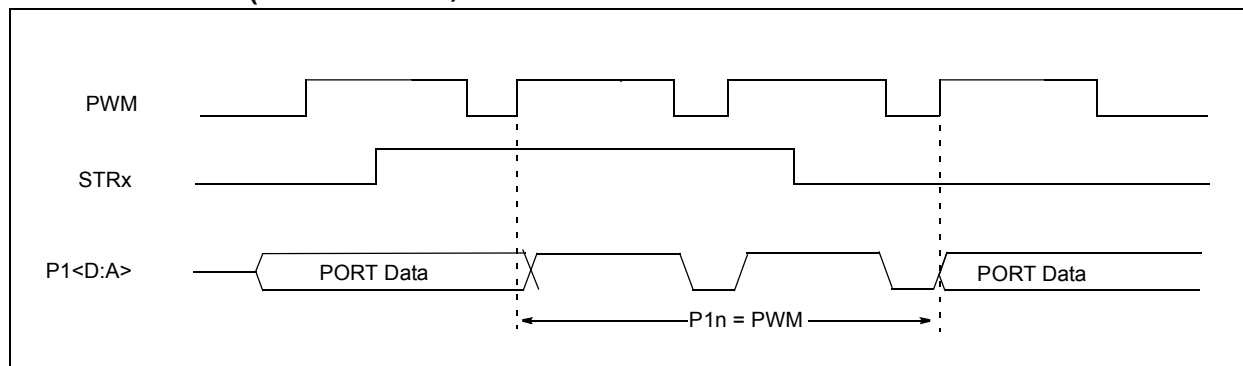


FIGURE 24-20: EXAMPLE OF STEERING EVENT AT BEGINNING OF INSTRUCTION (STRxSYNC = 1)



PIC16(L)F1826/27

25.2.2 SPI MODE OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPxCON1<5:0> and SSPxSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCKx is the clock output)
- Slave mode (SCKx is the clock input)
- Clock Polarity (Idle state of SCKx)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCKx)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

To enable the serial port, SSPx Enable bit, SSPxEN of the SSPxCON1 register, must be set. To reset or reconfigure SPI mode, clear the SSPxEN bit, re-initialize the SSPxCONx registers and then set the SSPxEN bit. This configures the SDIx, SDOx, SCKx and $\overline{\text{SSx}}$ pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

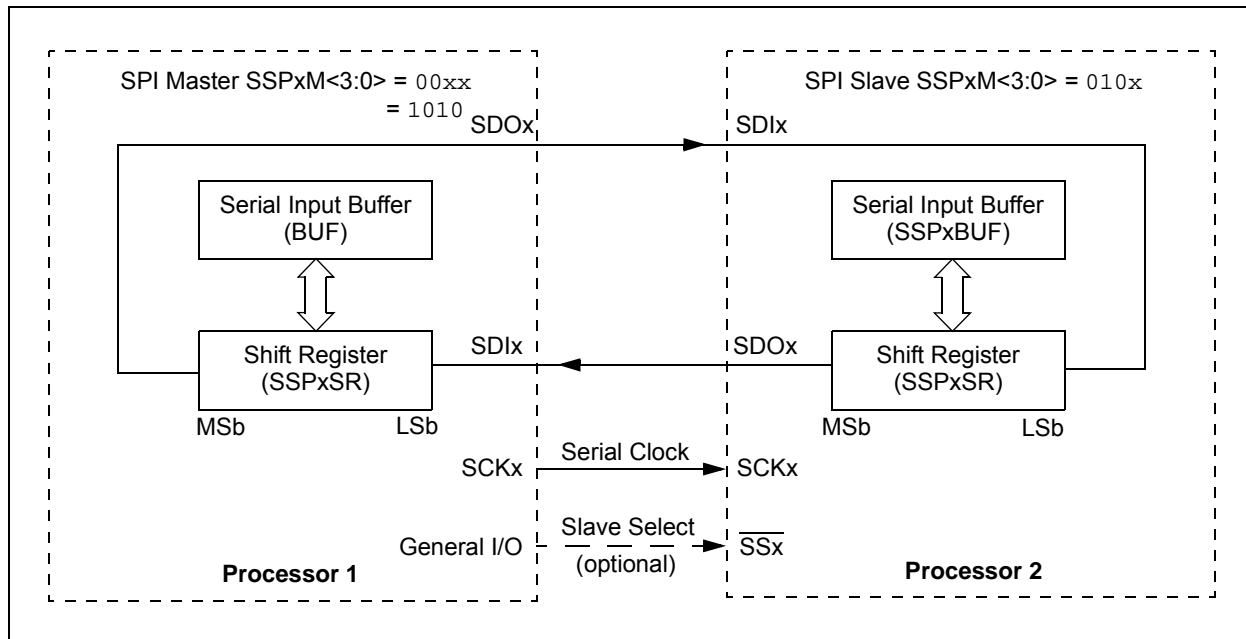
- SDIx must have corresponding TRIS bit set
- SDOx must have corresponding TRIS bit cleared
- SCKx (Master mode) must have corresponding TRIS bit cleared
- SCKx (Slave mode) must have corresponding TRIS bit set
- $\overline{\text{SSx}}$ must have corresponding TRIS bit set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

The MSSPx consists of a transmit/receive shift register (SSPxSR) and a buffer register (SSPxBUF). The SSPxSR shifts the data in and out of the device, MSb first. The SSPxBUF holds the data that was written to the SSPxSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPxBUF register. Then, the Buffer Full Detect bit, BF of the SSPxSTAT register, and the interrupt flag bit, SSPxIF, are set. This double-buffering of the received data (SSPxBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPxBUF register during transmission/reception of data will be ignored and the write collision detect bit WCOL of the SSPxCON1 register, will be set. User software must clear the WCOL bit to allow the following write(s) to the SSPxBUF register to complete successfully.

When the application software is expecting to receive valid data, the SSPxBUF should be read before the next byte of data to transfer is written to the SSPxBUF. The Buffer Full bit, BF of the SSPxSTAT register, indicates when SSPxBUF has been loaded with the received data (transmission is complete). When the SSPxBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSPx interrupt is used to determine when the transmission/reception has completed. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur.

FIGURE 25-5: SPI MASTER/SLAVE CONNECTION



25.2.3 SPI MASTER MODE

The master can initiate the data transfer at any time because it controls the SCKx line. The master determines when the slave (Processor 2, Figure 25-5) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPxBUF register is written to. If the SPI is only going to receive, the SDOx output could be disabled (programmed as an input). The SSPxSR register will continue to shift in the signal present on the SDIx pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPxBUF register as if a normal received byte (interrupts and Status bits appropriately set).

The clock polarity is selected by appropriately programming the CKP bit of the SSPxCON1 register and the CKE bit of the SSPxSTAT register. This then, would give waveforms for SPI communication as shown in Figure 25-6, Figure 25-8 and Figure 25-9, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- $F_{osc}/4$ (or T_{CY})
- $F_{osc}/16$ (or $4 * T_{CY}$)
- $F_{osc}/64$ (or $16 * T_{CY}$)
- Timer2 output/2
- $F_{osc}/(4 * (SSPxADD + 1))$

Figure 25-6 shows the waveforms for Master mode.

When the CKE bit is set, the SDOx data is valid before there is a clock edge on SCKx. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPxBUF is loaded with the received data is shown.

FIGURE 25-6: SPI MODE WAVEFORM (MASTER MODE)

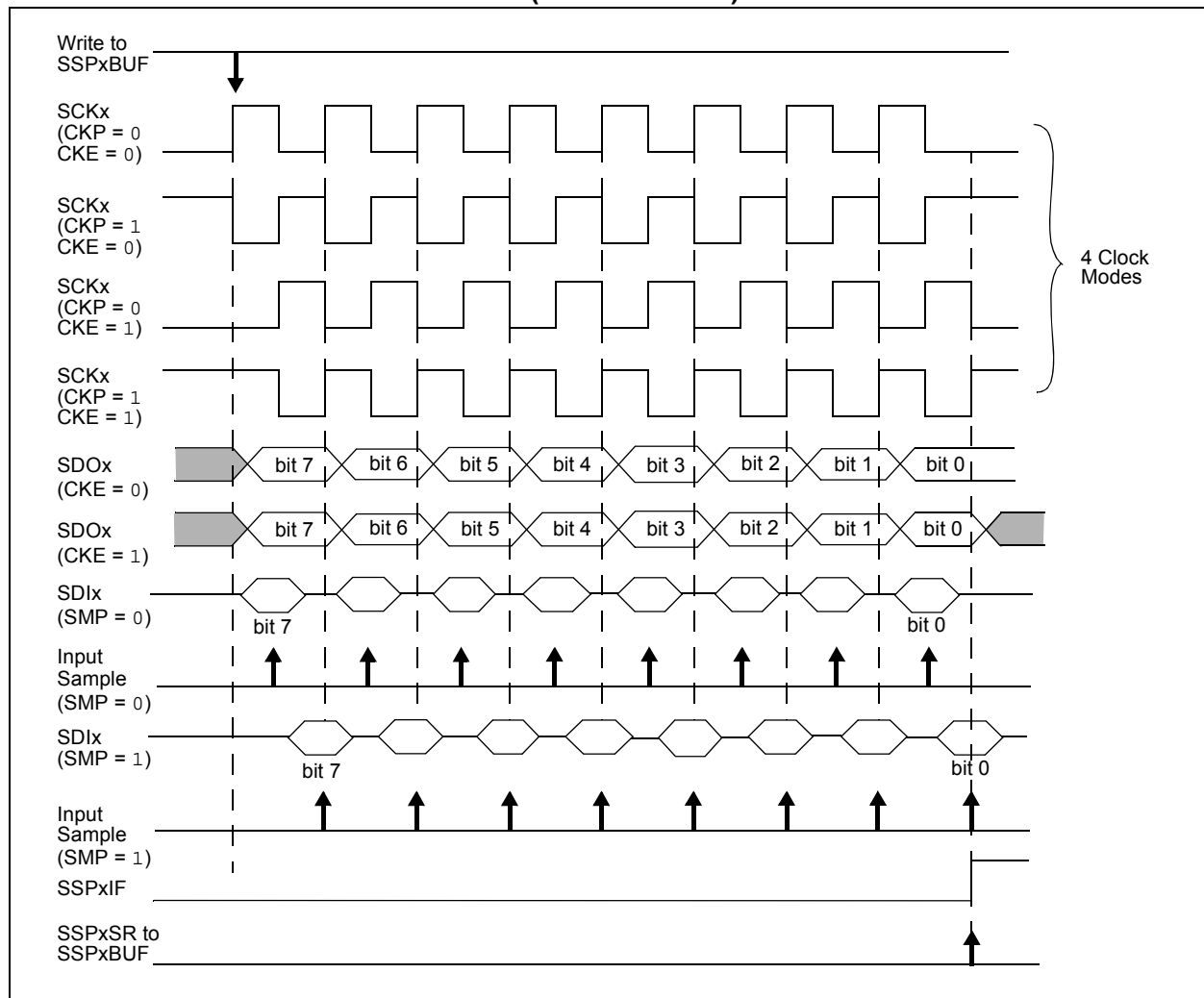


FIGURE 25-16: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 1, DHEN = 1)

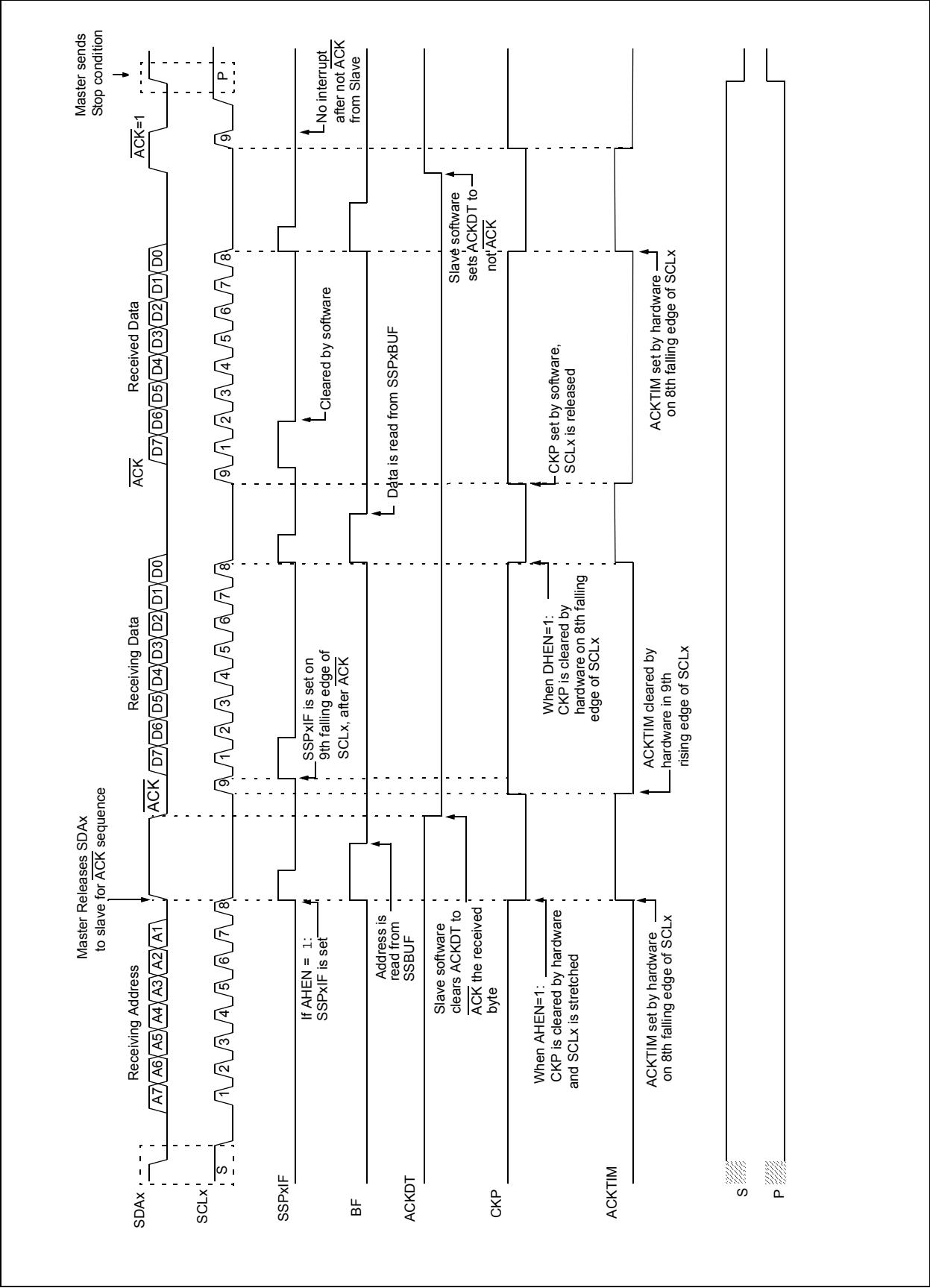
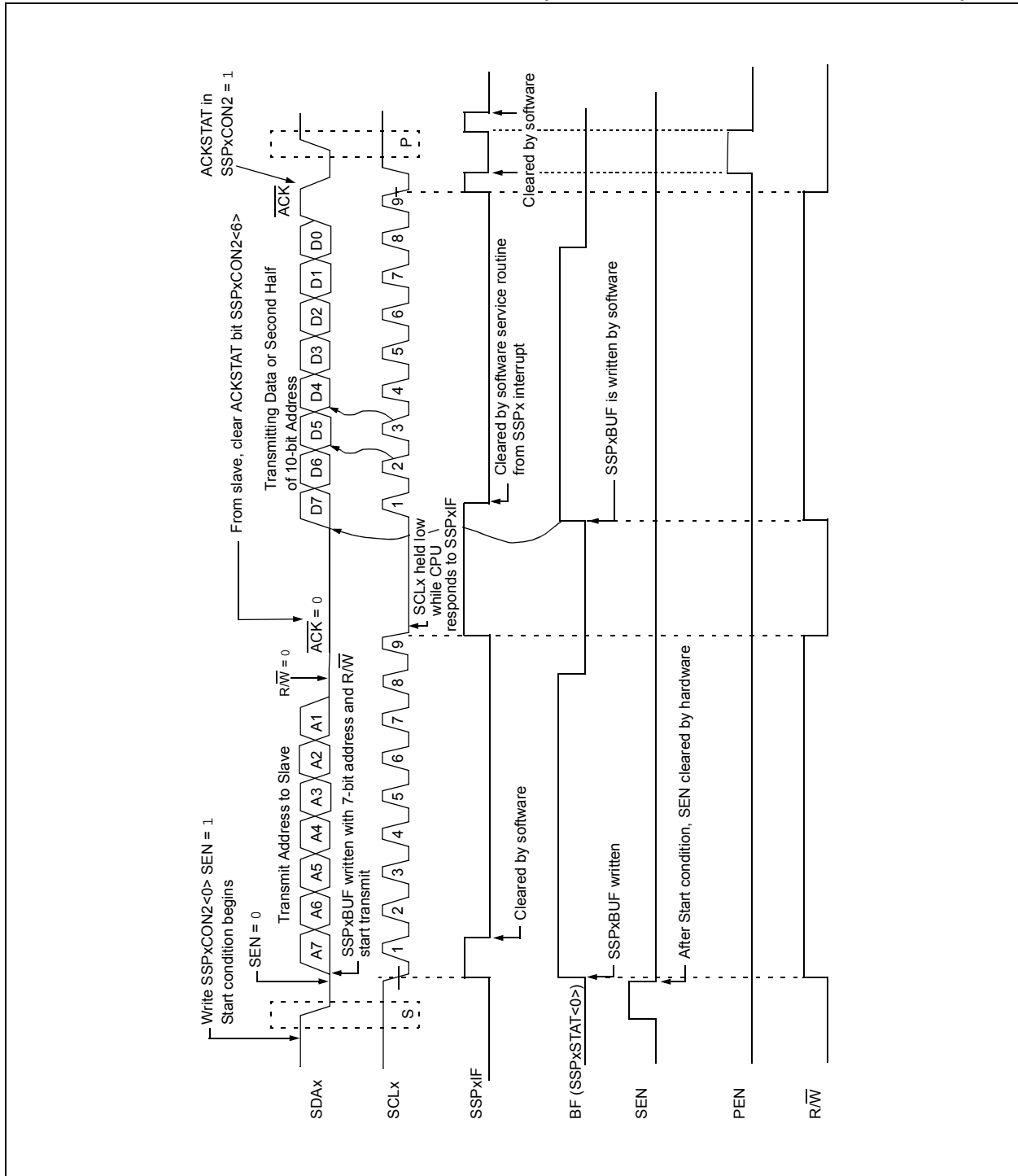


FIGURE 25-28: I²C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)



26.1.2.8 Asynchronous Reception Set-up:

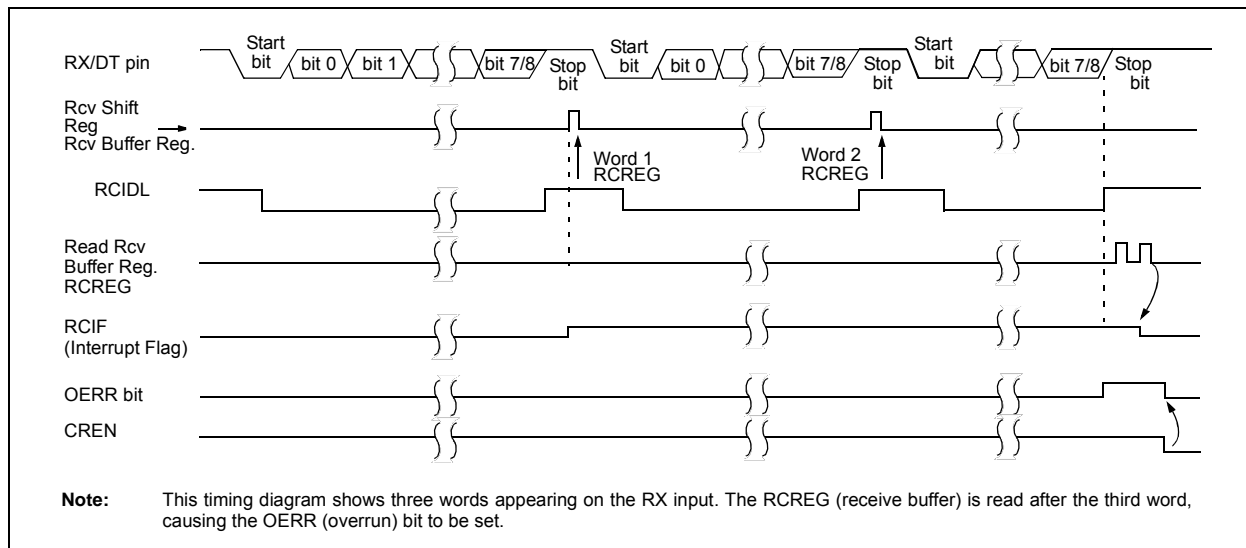
1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 26.3 “EUSART Baud Rate Generator (BRG)”**).
2. Clear the ANSEL bit for the RX pin (if applicable).
3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
4. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
5. If 9-bit reception is desired, set the RX9 bit.
6. Enable reception by setting the CREN bit.
7. The RCIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
8. Read the RCSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
9. Get the received 8 Least Significant data bits from the receive buffer by reading the RCREG register.
10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

26.1.2.9 9-bit Address Detection Mode Set-up

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

1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 26.3 “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 RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
5. Enable 9-bit reception by setting the RX9 bit.
6. Enable address detection by setting the ADDEN bit.
7. Enable reception by setting the CREN bit.
8. The RCIF interrupt flag bit will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
9. Read the RCSTA register to get the error flags. The ninth data bit will always be set.
10. Get the received 8 Least Significant data bits from the receive buffer by reading the RCREG register. Software determines if this is the device's address.
11. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
12. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.

FIGURE 26-5: ASYNCHRONOUS RECEPTION



PIC16(L)F1826/27

TABLE 26-7: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON0	RXDTSEL	SDO1SEL	SS1SEL	P2BSEL ⁽¹⁾	CCP2SEL ⁽¹⁾	P1DSEL	P1CSEL	CCP1SEL	119
APFCON1	—	—	—	—	—	—	—	TXCKSEL	119
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	296
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	91
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	295
SPBRGL	BRG<7:0>								297*
SPBRGH	BRG<15:8>								297*
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	127
TXREG	EUSART Transmit Data Register								287*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDER	BRGH	TRMT	TX9D	294

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for Synchronous Master Transmission.

* Page provides register information.

Note 1: PIC16(L)F1827 only.

FIGURE 30-4: POR AND POR REARM WITH SLOW RISING V_{DD}

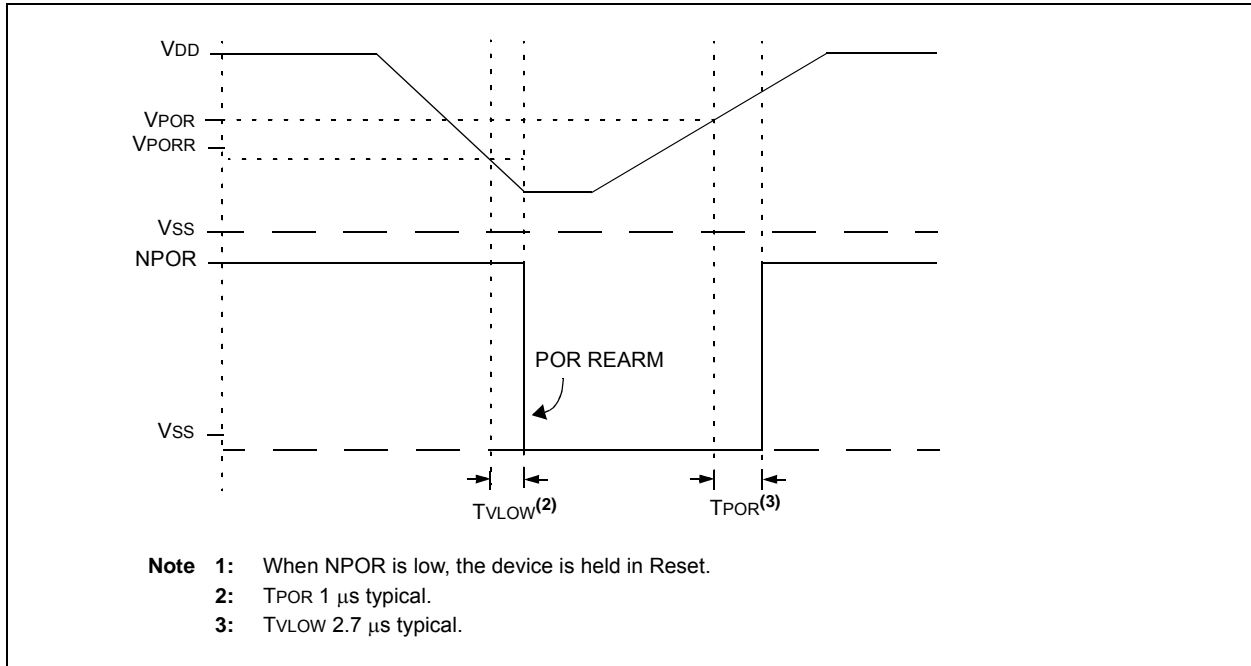


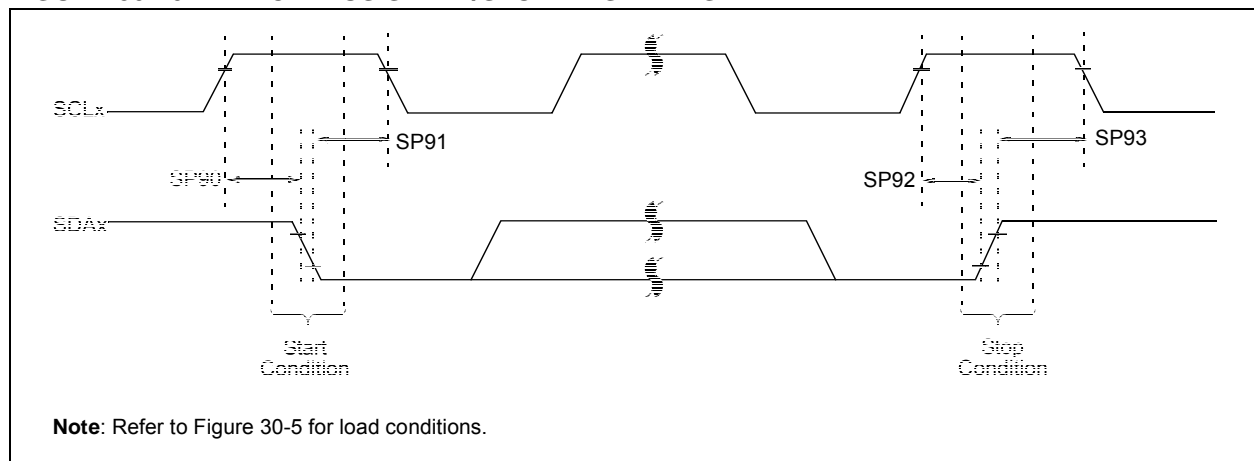
TABLE 30-14: SPI MODE REQUIREMENTS

Param No.	Symbol	Characteristic	Min.	Typ†	Max.	Units	Conditions
SP70*	TssL2sch, TssL2scL	$\overline{SSx}\downarrow$ to SCKx \downarrow or SCKx \uparrow input	Tcy	—	—	ns	
SP71*	Tsch	SCKx input high time (Slave mode)	Tcy + 20	—	—	ns	
SP72*	TscL	SCKx input low time (Slave mode)	Tcy + 20	—	—	ns	
SP73*	TdIV2sch, TdIV2scL	Setup time of SDIx data input to SCKx edge	100	—	—	ns	
SP74*	Tsch2dIL, TscL2dIL	Hold time of SDIx data input to SCKx edge	100	—	—	ns	
SP75*	TdoR	SDO data output rise time	3.0-5.5V	—	10	25	ns
			1.8-5.5V	—	25	50	ns
SP76*	TdoF	SDOx data output fall time	—	10	25	ns	
SP77*	TssH2doZ	$\overline{SSx}\uparrow$ to SDOx output high-impedance	10	—	50	ns	
SP78*	Tscr	SCKx output rise time (Master mode)	3.0-5.5V	—	10	25	ns
			1.8-5.5V	—	25	50	ns
SP79*	TscF	SCKx output fall time (Master mode)	—	10	25	ns	
SP80*	Tsch2doV, TscL2doV	SDOx data output valid after SCKx edge	3.0-5.5V	—	—	50	ns
			1.8-5.5V	—	—	145	ns
SP81*	TdoV2sch, TdoV2scL	SDOx data output setup to SCKx edge	Tcy	—	—	ns	
SP82*	TssL2doV	SDOx data output valid after $\overline{SS}\downarrow$ edge	—	—	50	ns	
SP83*	Tsch2ssH, TscL2ssH	$\overline{SSx}\uparrow$ after SCKx edge	1.5Tcy + 40	—	—	ns	

* These parameters are characterized but not tested.

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

FIGURE 30-20: I²C™ BUS START/STOP BITS TIMING



32.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
 - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
 - MPLAB C Compiler for Various Device Families
 - HI-TECH C for Various Device Families
 - MPASM™ Assembler
 - MPLINK™ Object Linker/
MPLIB™ Object Librarian
 - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
 - MPLAB SIM Software Simulator
- Emulators
 - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
 - MPLAB ICD 3
 - PICKit™ 3 Debug Express
- Device Programmers
 - PICKit™ 2 Programmer
 - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

32.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
 - Simulator
 - Programmer (sold separately)
 - In-Circuit Emulator (sold separately)
 - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

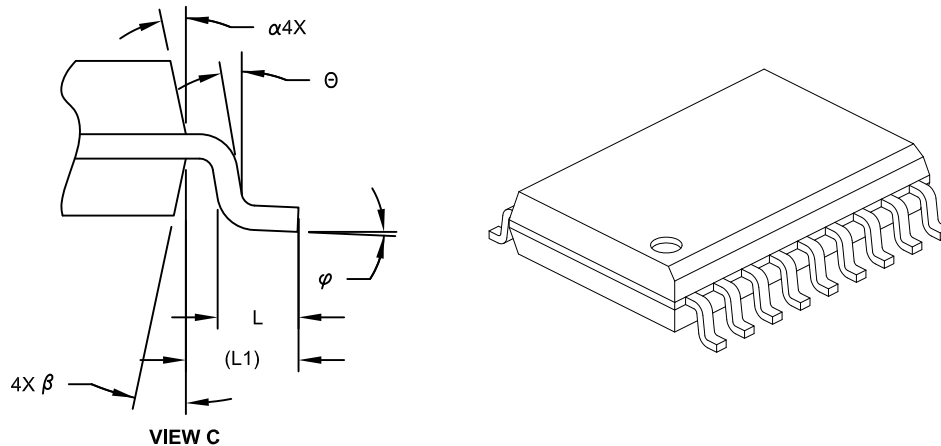
The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
 - Source files (C or assembly)
 - Mixed C and assembly
 - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

18-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	18		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	11.55 BSC		
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1	1.40 REF		
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.20	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
 - REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-051C Sheet 2 of 2

APPENDIX A: DATA SHEET REVISION HISTORY

Revision A

Original release (06/2009)

Revision B (08/09)

Revised Tables 5-3, 6-2, 12-2, 12-3; Updated Electrical Specifications; Added UQFN Package; Added SOIC and QFN Land Patterns; Updated Product ID section.

Revision C (06/10)

Updated Electrical Specification and included Enhanced Core Golden Chapters.

Revision D (04/11)

Added Char Data to release Final data sheet.

APPENDIX B: MIGRATING FROM OTHER PIC[®] DEVICES

This section provides comparisons when migrating from other similar PIC[®] devices to the PIC16(L)F1826/27 family of devices.

B.1 PIC16F648A to PIC16(L)F1827

TABLE B-1: FEATURE COMPARISON

Feature	PIC16F648A	PIC16(L)F1827
Max. Operating Speed	20 MHz	32 MHz
Max. Program Memory (Words)	4K	4K
Max. SRAM (Bytes)	256	384
Max. EEPROM (Bytes)	256	256
A/D Resolution	10-bit	10-bit
Timers (8/16-bit)	2/1	4/1
Brown-out Reset	Y	Y
Internal Pull-ups	RB<7:0>	RB<7:0>, RA5
Interrupt-on-Change	RB<7:4>	RB<7:0>, Edge Selectable
Comparator	2	2
AUSART/EUSART	1/0	0/2
Extended WDT	N	Y
Software Control Option of WDT/BOR	N	Y
INTOSC Frequencies	48 kHz or 4 MHz	31 kHz - 32 MHz
Clock Switching	Y	Y
Capacitive Sensing	N	Y
CCP/ECCP	2/0	2/2
Enhanced PIC16 CPU	N	Y
MSSPx/SSPx	0	2/0
Reference Clock	N	Y
Data Signal Modulator	N	Y
SR Latch	N	Y
Voltage Reference	N	Y
DAC	Y	Y