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	64MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, HLVD, LCD, POR, PWM, WDT
Number of I/O	68
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 24x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f86j94-i-pt

TABLE 1-4: PIC18FXXJ94 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	100	80	64			
$\overline{MCLR}$	11	9	7	I	ST	Master Clear (input) or programming voltage (input). This pin is an active-low Reset to the device.
OSC1/CLKI/RP10/RA7	61	49	39			Oscillator crystal or external clock input.
OSC1				I	ST	Oscillator crystal input.
CLKI				I	CMOS	External clock source input. Always associated with pin function, OSC1. (See related OSC1/CLKI,OSC2/CLKO pins.)
RP10				I/O	ST/DIG	Remappable Peripheral Pin 10 input/output.
RA7				I/O	ST/DIG	General purpose I/O pin.
OSC2/CLKO/RP6/RA6	62	50	40			Oscillator crystal or clock output.
OSC2				O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO				O	DIG	In certain oscillator modes, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RP6				I/O	ST/DIG	Remappable Peripheral Pin 6 input/output.
RA6				I/O	ST/DIG	General purpose I/O pin.

Legend: TTL = TTL compatible input
ST = Schmitt Trigger input with CMOS levels
I = Input
P = Power
I^2C = I^2C/SMBus
CMOS = CMOS compatible input or output
Analog = Analog input
O = Output
OD = Open-Drain (no P diode to V_{DD})

4.4.5 SAVING CONTEXT DATA WITH THE DSGPRx REGISTERS

As exiting Deep Sleep mode causes a POR, most Special Function Registers (SFRs) reset to their default POR values. In addition, because the core power is not supplied in Deep Sleep mode, information in data RAM may be lost when exiting this mode. Applications which require critical data to be saved prior to Deep Sleep may use the Deep Sleep General Purpose registers, DSGPR0, DSGPR1, DSGPR2 and DSGPR3. Unlike other SFRs, the contents of these registers are preserved while the device is in Deep Sleep mode. After exiting Deep Sleep, software can restore the data by reading the registers and clearing the RELEASE bit (DSCONL<0>).

Any data stored in the DSGPRx registers must be written twice. Like other Deep Sleep control features, the write operations do not need to be sequential. However, back-to-back writes are a recommended programming practice.

Since the contents of data RAM are maintained in Retention Deep Sleep, the use of the DSGPRx registers to store critical data is not necessary in this mode.

4.4.6 I/O PINS DURING DEEP SLEEP

During Deep Sleep, general purpose I/O pins retain their previous states. Pins that are configured as inputs (TRIS bit is set), prior to entry into Deep Sleep, remain high-impedance during Deep Sleep.

Pins that are configured as outputs (TRIS bit is clear), prior to entry into Deep Sleep, will remain as output pins during Deep Sleep. While in this mode, they will drive the output level determined by their corresponding LAT bit at the time of entry into Deep Sleep.

Once the device wakes back up, all I/O pins will continue to maintain their previous states, even after the device has finished the POR sequence and is executing application code again. Pins configured as inputs during Deep Sleep will remain high-impedance and pins configured as outputs will continue to drive their previous value. After waking up, the TRIS and LAT registers will be reset. If firmware modifies the TRIS and LAT values for the I/O pins, they will not immediately go to the newly configured states. Once the firmware clears the RELEASE bit (DSCONL<0>), the I/O pins are “released”. This causes the I/O pins to take the states configured by their respective TRIS and LAT bit values.

If the Deep Sleep BOR (DSBOR) is enabled, and a DSBOR event occurs during Deep Sleep (or V_{DD} is hard-cycled to V_{SS}), the I/O pins will be immediately released, similar to clearing the RELEASE bit. All previous state information will be lost, including the general purpose DSGPR0, DSGPR1, DSGPR2 and DSGPR3 contents. DSGPRx register contents will be maintained if the V_{BAT} pin is powered.

If a $\overline{MCLR}$ Reset event occurs during Deep Sleep, the I/O pins will also be released automatically, but in this case, the DSGPR0, DSGPR1, DSGPR2 and DSGPR3 contents will remain valid.

In case of $\overline{MCLR}$ Reset and all other Deep Sleep wake-up cases, application firmware needs to clear the RELEASE bit (DSCONL<0>) in order to reconfigure the I/O pins.

4.4.7 DEEP SLEEP WATCHDOG TIMER (DSWDT)

Deep Sleep has its dedicated WDT (DSWDT). It is enabled through the DSWDTEN Configuration bit. The DSWDT is equipped with a postscaler for time-outs of 2.1 ms to 25.7 days, configurable through the Configuration bits, DSWDTPS<4:0>. Entering Deep Sleep mode automatically clears the DSWDT.

The DSWDT also has a configurable reference clock source for selecting the LPRC or SOSC. The reference clock source is configured through the DSWDTOSC Configuration bit.

Under certain circumstances, it is possible for the DSWDT clock source to be off when entering Deep Sleep mode. In this case, the clock source is turned on automatically (if DSWDT is enabled), without the need for software intervention. However, this can cause a delay in the start of the DSWDT counters. In order to avoid this delay, when using SOSC as a clock source, the application can activate SOSC prior to entering Deep Sleep mode.

4.4.8 DEEP SLEEP LOW-POWER BROWN-OUT RESET

Devices with a Deep Sleep Power-Saving mode also have a dedicated BOR for Deep Sleep modes (DSBOR). It has a trip point range of 1.7V-2.3V nominal and is enabled through the DSBOREN (CONFIG7L<3>) Configuration bit.

When the device enters a Deep Sleep mode and receives a DSBOR event, the device will not wake-up and will remain in the Deep Sleep mode. When a valid wake-up event occurs and causes the device to exit Deep Sleep mode, software can determine if a DSBOR event occurred during Deep Sleep mode by reading the BOR (DSWAKEL<6>) Status bit.

4.4.9 RTCC AND DEEP SLEEP

The RTCC can operate uninterrupted during Deep Sleep modes. It can wake-up the device from Deep Sleep by configuring an alarm. The RTCC clock source is configured with the RTCC Clock Select bits, RTCCLKSEL<1:0>. The available reference clock sources are the LPRC and SOSC. If the LPRC is used, the RTCC accuracy will directly depend on the LPRC tolerance.

If the RTCC is not required, Deep Sleep mode with the RTCC disabled, affords the lowest power consumption of any of the instruction-based power-saving modes.

REGISTER 4-1: DSCONL: DEEP SLEEP CONTROL REGISTER LOW

U-0	U-0	U-0	U-0	U-0	R-0	R/W-0, HSC	R/W-0, HS
—	—	—	—	—	r	DSBOR(1)	RELEASE(1)
bit 7							bit 0

Legend:	r = Reserved bit	HSC = Hardware Settable/Clearable bit	
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
HS = Hardware Settable bit			

bit 7-3 **Unimplemented:** Read as '0'

bit 2 **Reserved:** Maintained as '0'

bit 1 **DSBOR:** Deep Sleep BOR Event Status bit(1)
1 = DSBOR was enabled and VDD dropped below the DSBOR threshold during Deep Sleep(2)
0 = DSBOR disabled while device is in Deep Sleep mode

bit 0 **RELEASE:** I/O Pin State Release bit(1)
Upon waking from Deep Sleep, the I/O pins maintain their previous states. Clearing this bit will release the I/O pins and allow their respective TRIS and LAT bits to control their states.

Note 1: This is the value when VDD is initially applied.
2: Unlike all other events, a Deep Sleep BOR event will not cause a wake-up from Deep Sleep; this bit is present only as a Status bit.

REGISTER 4-2: DSCONH: DEEP SLEEP CONTROL REGISTER HIGH

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HS(2)
DSEN(1)	—	—	—	—	—	—	RTCCWDIS
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
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 **DSEN:** Deep Sleep Mode Enable bit(1)
1 = Deep Sleep mode is enabled and device will enter Deep Sleep mode when the SLEEP instruction is executed
0 = Deep Sleep mode is not enabled

bit 6-1 **Unimplemented:** Read as '0'

bit 0 **RTCCWDIS:** RTCC Wake-up Disable bit(2)
1 = Wake-up from RTCC is disabled
0 = Wake-up from RTCC is enabled

Note 1: In order to enter Deep Sleep, DSEN must be written to in two separate operations. The write operations do not need to be consecutive. Before writing DSEN, the DSCON1 register should be cleared twice.
2: This is the value when VDD is initially applied.

TABLE 11-12: RPINR REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RPINR52_53	RVP7R3	RVP7R2	RVP7R1	RVP7R0	RVP6R3	RVP6R2	RVP6R1	RVP6R0
RPINR50_51	RVP5R3	RVP5R2	RVP5R1	RVP5R0	RVP4R3	RVP4R2	RVP4R1	RVP4R0
RPINR48_49	RVP3R3	RVP3R2	RVP3R1	RVP3R0	RVP2R3	RVP2R2	RVP2R1	RVP2R0
RPINR46_47	RVP1R3	RVP1R2	RVP1R1	RVP1R0	RVP0R3	RVP0R2	RVP0R1	RVP0R0
RPINR44_45	T5CKIR3	T5CKIR2	T5CKIR1	T5CKIR0	T5GR3	T5GR2	T5GR1	T5GR0
RPINR42_43	T3CKIR3	T3CKIR2	T3CKIR1	T3CKIR0	T3GR3	T3GR2	T3GR1	T3GR0
RPINR40_41	T1CKIR3	T1CKIR2	T1CKIR1	T1CKIR0	T1GR3	T1GR2	T1GR1	T1GR0
RPINR38_39	T0CKIR3	T0CKIR2	T0CKIR1	T0CKIR0	CCP10R3	CCP10R2	CCP10R1	CCP10R0
RPINR36_37	CCP9R3	CCP9R2	CCP9R1	CCP9R0	CCP8R3	CCP8R2	CCP8R1	CCP8R0
RPINR34_35	CCP7R3	CCP7R2	CCP7R1	CCP7R0	CCP6R3	CCP6R2	CCP6R1	CCP6R0
RPINR32_33	CCP5R3	CCP5R2	CCP5R1	CCP5R0	CCP4R3	CCP4R2	CCP4R1	CCP4R0
RPINR30_31	MDCIN2R3	MDCIN2R2	MDCIN2R1	MDCIN2R0	MDCIN1R3	MDCIN1R2	MDCIN1R1	MDCIN1R0
RPINR28_29	MDMINR3	MDMINR2	MDMINR1	MDMINR0	INT3R3	INT3R2	INT3R1	INT3R0
RPINR26_27	INT2R3	INT2R2	INT2R1	INT2R0	INT1R3	INT1R2	INT1R1	INT1R0
RPINR24_25	IOC7R3	IOC7R2	IOC7R1	IOC7R0	IOC6R3	IOC6R2	IOC6R1	IOC6R0
RPINR22_23	IOC5R3	IOC5R2	IOC5R1	IOC5R0	IOC4R3	IOC4R2	IOC4R1	IOC4R0
RPINR20_21	IOC3R3	IOC3R2	IOC3R1	IOC3R0	IOC2R3	IOC2R2	IOC2R1	IOC2R0
RPINR18_19	IOC1R3	IOC1R2	IOC1R1	IOC1R0	IOC0R3	IOC0R2	IOC0R1	IOC0R0
RPINR16_17	ECCP3R3	ECCP3R2	ECCP3R1	ECCP3R0	ECCP2R3	ECCP2R2	ECCP2R1	ECCP2R0
RPINR14_15	ECCP1R3	ECCP1R2	ECCP1R1	ECCP1R0	FLT0R3	FLT0R2	FLT0R1	FLT0R0
RPINR12_13	SS2R3	SS2R2	SS2R1	SS2R0	SDI2R3	SDI2R2	SDI2R1	SDI2R0
RPINR10_11	SCK2R3	SCK2R2	SCK2R1	SCK2R0	SS1R3	SS1R2	SS1R1	SS1R0
RPINR8_9	SDI1R3	SDI1R2	SDI1R1	SDI1R0	SCK1R3	SCK1R2	SCK1R1	SCK1R0
RPINR6_7	U4TXR3	U4TXR2	U4TXR1	U4TXR0	U4RXR3	U4RXR2	U4RXR1	U4RXR0
RPINR4_5	U3TXR3	U3TXR2	U3TXR1	U3TXR0	U3RXR3	U3RXR2	U3RXR1	U3RXR0
RPINR2_3	U2TXR3	U2TXR2	U2TXR1	U2TXR0	U2RXR3	U2RXR2	U2RXR1	U2RXR0
RPINR0_1	U1TXR3	U1TXR2	U1TXR1	U1TXR0	U1RXR3	U1RXR2	U1RXR1	U1RXR0

REGISTER 13-6: LCDREF: LCD REFERENCE LADDER CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LCDIRE	—	LCDCST2	LCDCST1	LCDCST0	VLCD3PE	VLCD2PE	VLCD1PE
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 **LCDIRE**: LCD Internal Reference Enable bit
1 = Internal LCD reference is enabled and connected to the internal contrast control circuit
0 = Internal LCD reference is disabled

bit 6 **Unimplemented**: Read as '0'

bit 5-3 **LCDCST<2:0>**: LCD Contrast Control bits
Selects the Resistance of the LCD Contrast Control Resistor Ladder:
111 =Resistor ladder is at maximum resistance (minimum contrast)
110 =Resistor ladder is at 6/7th of maximum resistance
101 =Resistor ladder is at 5/7th of maximum resistance
100 =Resistor ladder is at 4/7th of maximum resistance
011 =Resistor ladder is at 3/7th of maximum resistance
010 =Resistor ladder is at 2/7th of maximum resistance
001 =Resistor ladder is at 1/7th of maximum resistance
000 =Minimum resistance (maximum contrast); resistor ladder is shorted

bit 2 **VLCD3PE**: Bias3 Pin Enable bit
1 = BIAS3 level is connected to the external pin, LCDBIAS3
0 = BIAS3 level is internal (internal resistor ladder)

bit 1 **VLCD2PE**: Bias2 Pin Enable bit
1 = BIAS2 level is connected to the external pin, LCDBIAS2
0 = BIAS2 level is internal (internal resistor ladder)

bit 0 **VLCD1PE**: Bias1 Pin Enable bit
1 = BIAS1 level is connected to the external pin, LCDBIAS1
0 = BIAS1 level is internal (internal resistor ladder)

20.3.4 ENABLING SPI I/O

To enable the serial port, the peripheral must first be mapped to I/O pins using the PPS-Lite feature. To enable the SPI peripheral, the MSSPx Enable bit, SSPEN (SSPxCON1<5>) must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPxCON registers and then set the SSPEN bit. This configures the SDIx, SDOx, SCKx and $\overline{SSx}$ 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 is automatically controlled by the SPI module
- SDOx must have the TRIS bit cleared for the corresponding RPn pin.
- SCKx (Master mode) must have the TRIS bit cleared for the corresponding RPn pin
- SCKx (Slave mode) must have the TRIS bit set for the corresponding RPn pin
- $\overline{SSx}$ must have the TRIS bit set for the corresponding RPn pin.

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

20.3.5 TYPICAL CONNECTION

Figure 20-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCKx signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

FIGURE 20-2: SPI MASTER/SLAVE CONNECTION

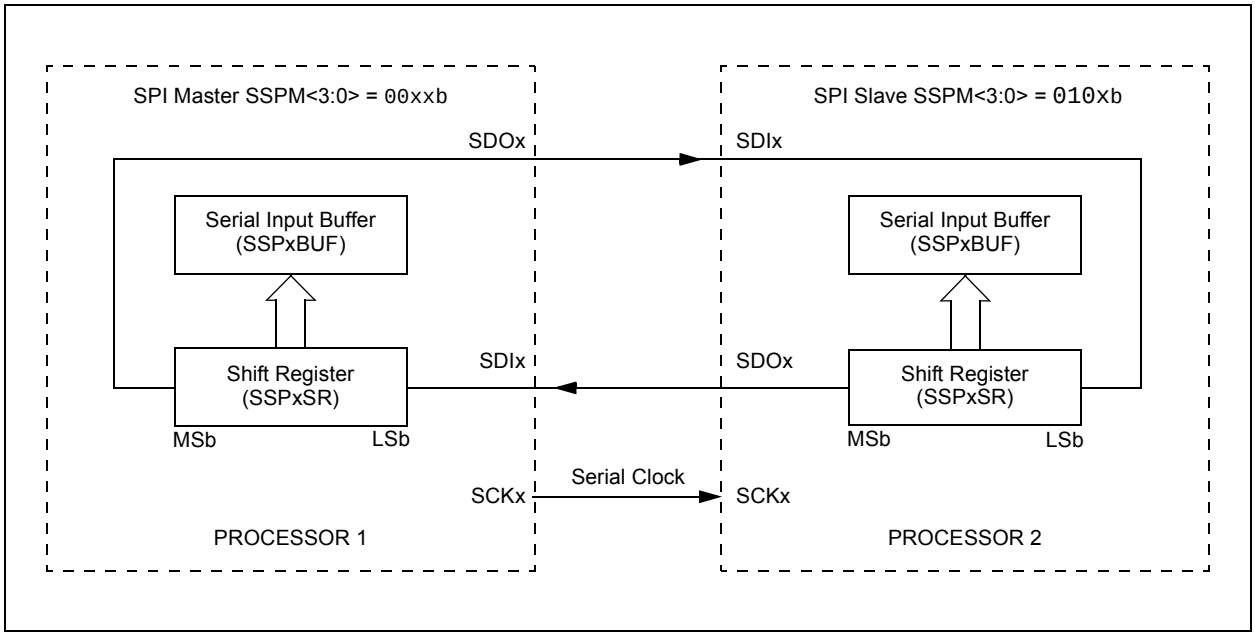


FIGURE 20-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

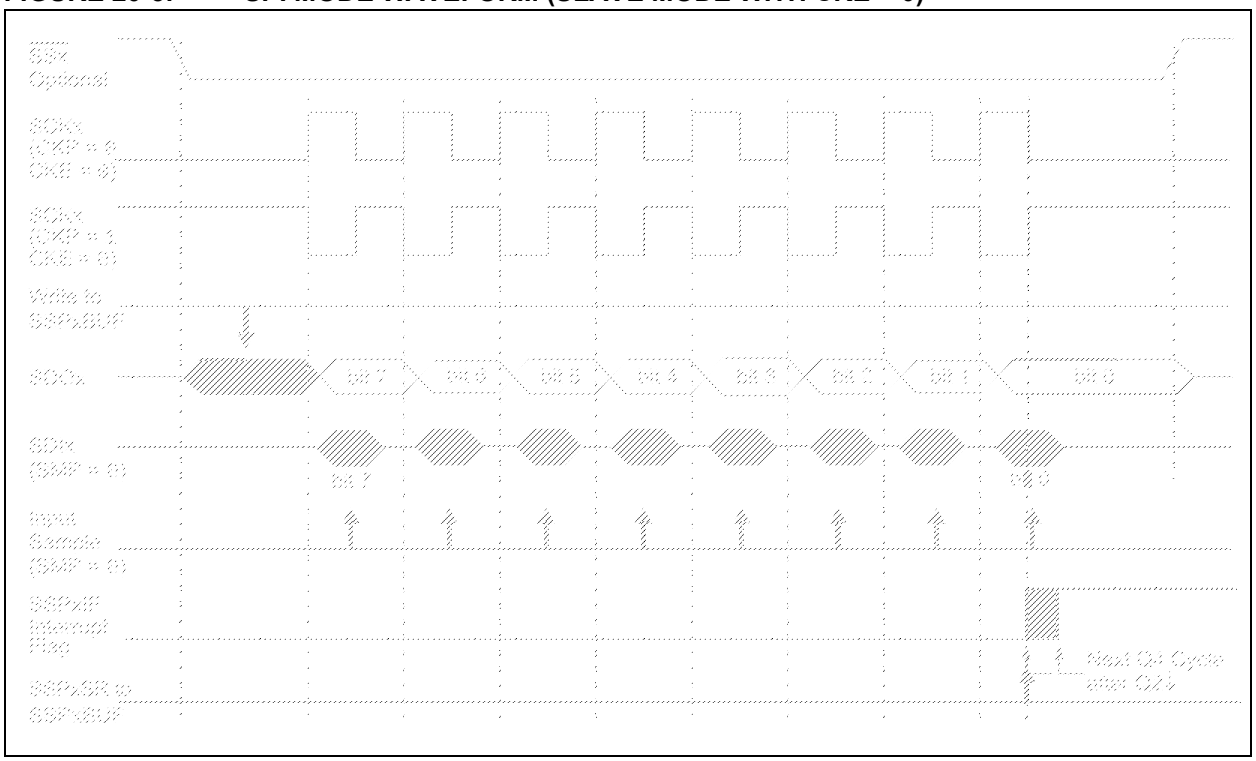
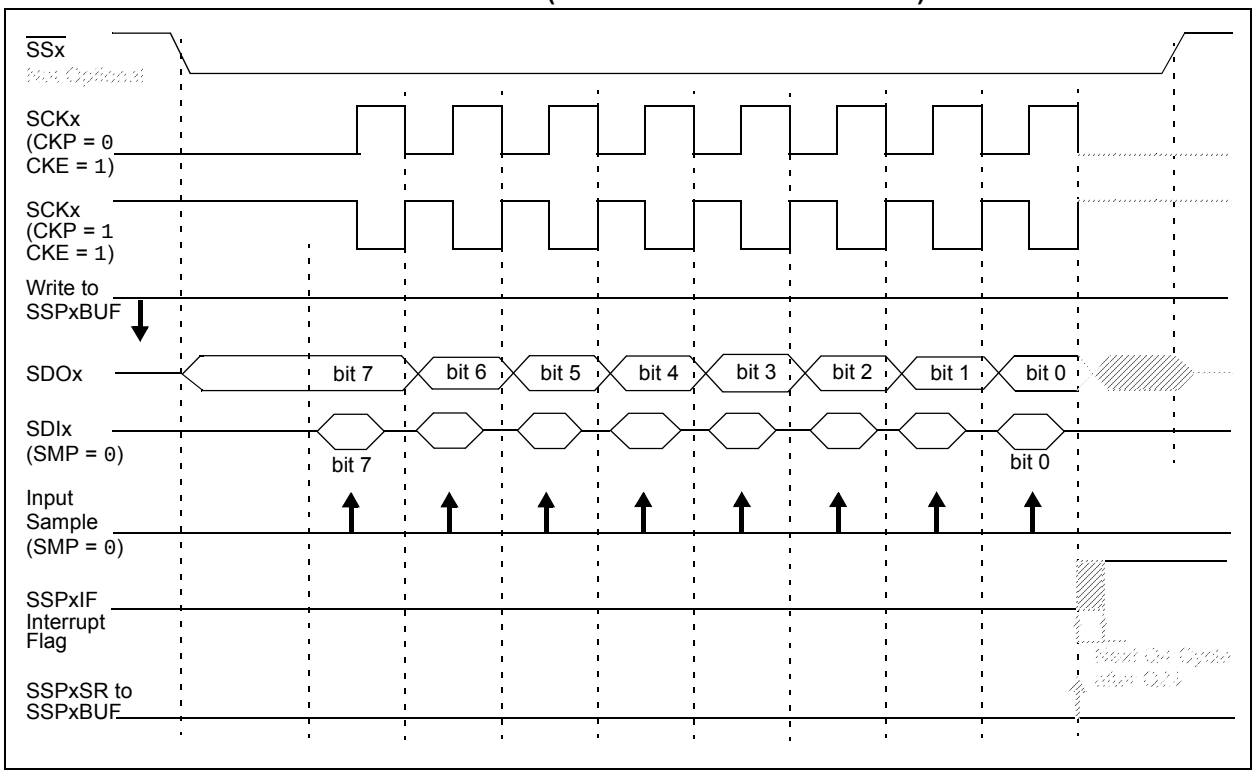


FIGURE 20-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



20.5.7 CLOCK STRETCHING

Both 7-Bit and 10-Bit Slave modes implement automatic clock stretching during a transmit sequence.

The SEN bit (SSPxCON2<0>) allows clock stretching to be enabled during receives. Setting SEN will cause the SCLx pin to be held low at the end of each data receive sequence.

20.5.7.1 Clock Stretching for 7-Bit Slave Receive Mode (SEN = 1)

In 7-Bit Slave Receive mode, on the falling edge of the ninth clock at the end of the $\overline{ACK}$ sequence, if the BF bit is set, the CKP bit in the SSPxCON1 register is automatically cleared, forcing the SCLx output to be held low. The CKP bit, being cleared to '0', will assert the SCLx line low. The CKP bit must be set in the user's ISR before reception is allowed to continue. By holding the SCLx line low, the user has time to service the ISR and read the contents of the SSPxBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring (see Figure 20-15).

Note 1: If the user reads the contents of the SSPxBUF before the falling edge of the ninth clock, thus clearing the BF bit, the CKP bit will not be cleared and clock stretching will not occur.

2: The CKP bit can be set in software regardless of the state of the BF bit. The user should be careful to clear the BF bit in the ISR before the next receive sequence in order to prevent an overflow condition.

20.5.7.2 Clock Stretching for 10-Bit Slave Receive Mode (SEN = 1)

In 10-Bit Slave Receive mode, during the address sequence, clock stretching automatically takes place but CKP is not cleared. During this time, if the UA bit is set after the ninth clock, clock stretching is initiated. The UA bit is set after receiving the upper byte of the 10-bit address, and following the receive of the second byte of the 10-bit address, with the R/$\overline{W}$ bit cleared to '0'. The release of the clock line occurs upon updating SSPxADD. Clock stretching will occur on each data receive sequence as described in 7-bit mode.

Note: If the user polls the UA bit and clears it by updating the SSPxADD register before the falling edge of the ninth clock occurs, and if the user hasn't cleared the BF bit by reading the SSPxBUF register before that time, then the CKP bit will still NOT be asserted low. Clock stretching, on the basis of the state of the BF bit, only occurs during a data sequence, not an address sequence.

20.5.7.3 Clock Stretching for 7-Bit Slave Transmit Mode

The 7-Bit Slave Transmit mode implements clock stretching by clearing the CKP bit after the falling edge of the ninth clock if the BF bit is clear. This occurs regardless of the state of the SEN bit.

The user's ISR must set the CKP bit before transmission is allowed to continue. By holding the SCLx line low, the user has time to service the ISR and load the contents of the SSPxBUF before the master device can initiate another transmit sequence (see Figure 20-10).

Note 1: If the user loads the contents of SSPxBUF, setting the BF bit before the falling edge of the ninth clock, the CKP bit will not be cleared and clock stretching will not occur.

2: The CKP bit can be set in software, regardless of the state of the BF bit.

20.5.7.4 Clock Stretching for 10-Bit Slave Transmit Mode

In 10-Bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-Bit Slave Receive mode. The first two addresses are followed by a third address sequence, which contains the high-order bits of the 10-bit address and the R/$\overline{W}$ bit set to '1'. After the third address sequence is performed, the UA bit is not set, the module is now configured in Transmit mode and clock stretching is controlled by the BF flag as in 7-Bit Slave Transmit mode (see Figure 20-13).

REGISTER 21-3: BAUDCONx: BAUD RATE CONTROL REGISTER x

R/W-0	R-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	IREN	WUE	ABDEN
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 **ABDOVF:** Auto-Baud Acquisition Rollover Status bit
1 = A BRG rollover has occurred during Auto-Baud Rate Detect mode (must be cleared in software)
0 = No BRG rollover has occurred

bit 6 **RCIDL:** Receive Operation Idle Status bit
1 = Receive operation is Idle
0 = Receive operation is active

bit 5 **RXDTP:** Data/Receive Polarity Select bit
Asynchronous mode:
1 = Receive data (RXx) is inverted (active-low)
0 = Receive data (RXx) is not inverted (active-high)
Asynchronous IrDA mode:
No effect on operation
Synchronous mode:
1 = Data (DTx) is inverted (active-low)
0 = Data (DTx) is not inverted (active-high)

bit 4 **TXCKP:** Synchronous Clock Polarity Select bit
Asynchronous mode:
1 = Idle state for transmit (TXx) is a low level
0 = Idle state for transmit (TXx) is a high level
Asynchronous IrDA mode:
1 = Idle state for IrDA transmit (TX) is a high level ('1')
0 = Idle state for IrDA transmit (TX) is a low level ('0')
Synchronous mode:
1 = Idle state for clock (CKx) is a high level
0 = Idle state for clock (CKx) is a low level

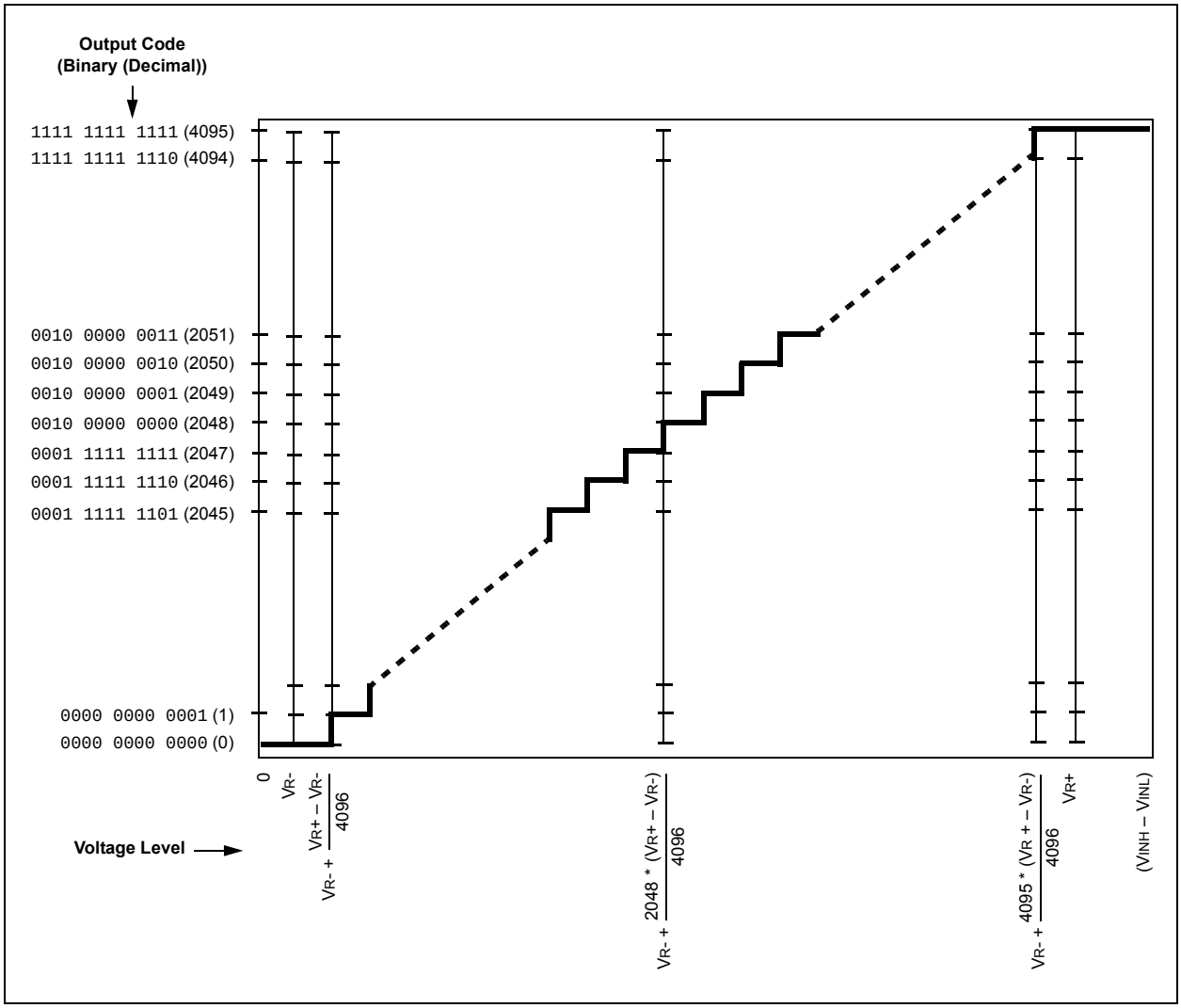
bit 3 **BRG16:** 16-Bit Baud Rate Register Enable bit
1 = 16-bit Baud Rate Generator – SPBRGHx and SPBRGx
0 = 8-bit Baud Rate Generator – SPBRGx only (Compatible mode), SPBRGHx value is ignored

bit 2 **IREN:** IrDA® Encoder and Decoder Enable bit
Asynchronous mode:
1 = IrDA encoder and decoder are enabled (Asynchronous IrDA mode is active)
0 = IrDA encoder and decoder are disabled
Synchronous mode:(1)
No effect on operation.

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = EUSARTx will continue to sample the RXx pin – interrupt is generated on the falling edge; bit is cleared in hardware on following rising edge
0 = RXx pin is not monitored or rising edge detected
Synchronous mode:
Unused in this mode.

Note 1: This feature is only available in Asynchronous mode with the 16x clock preset. The 16x clock is present for both the x16 and x64 BRG configurations.

FIGURE 22-21: 12-BIT A/D TRANSFER FUNCTION



For the 10-bit transfer function (when 10-bit resolution is available):

- The first code transition occurs when the input voltage is ((VR+) - (VR-))/1024 or 1.0 LSb.
- The '00 0000 0001' code is centered at VR- + (1.5 * (((VR+) - (VR-)) / 1024).
- The '10 0000 0000' code is centered at VREFL + (512.5 * (((VR+) - (VR-)) /1024).
- An input voltage less than VR- + (((VR-) - (VR-)) / 1024) converts as '00 0000 0000'.
- An input voltage greater than (VR-) + ((1023 (VR+)) - (VR-))/1024) converts as '11 1111 1111'.

To minimize the effects of digital noise on the A/D module operation, the user should select a conversion trigger source that ensures the A/D conversion will take place in Sleep mode. The automatic conversion trigger option can be used for sampling and conversion in Sleep (SSRC<3:0> = 0111). To use the automatic conversion option, the ADON bit should be set in the instruction prior to the SLEEP instruction.

Note:	For the A/D module to operate in Sleep, the A/D clock source must be set to RC (ADRC = 1).

22.11.3 A/D OPERATION DURING CPU IDLE MODE

The module will continue normal operation when the device enters Idle mode. If the A/D interrupt is enabled (ADIE = 1), the device will wake-up from Idle mode when the A/D interrupt occurs. If the respective global interrupt enable bit(s) are also set, program execution will resume at the A/D Interrupt Service Routine (ISR). After the ISR completes, execution will continue from the instruction after the SLEEP instruction that placed the device in Idle mode.

22.11.4 PERIPHERAL MODULE DISABLE (PMD) REGISTER

The Peripheral Module Disable (PMD) registers provide a method to disable the A/D module by stopping all clock sources supplied to that module. When a peripheral is disabled via the appropriate PMDx control bit, the peripheral is in a minimum power consumption state. The control and STATUS registers associated with the peripheral will also be disabled, so writes to those registers will have no effect and read values will be invalid. The A/D module is enabled only when the ADCMD bit in the PMD3 register is cleared.

22.12 Design Tips

Question 1: How can I optimize the system performance of the A/D Converter?

Answer: There are three main things to consider in optimizing A/D performance:

1. Make sure you are meeting all of the timing specifications. If you are turning the module off and on, there is a minimum delay you must wait before taking a sample. If you are changing input channels, there is a minimum delay you must wait for this as well, and finally, there is TAD, which is the time selected for each bit conversion. This is selected in AD1CON3 and should be within a certain range, as specified in **Section 30.0 "Electrical Specifications"**. If TAD is too short, the result may not be fully con- verted before the con- version is terminated, and if TAD is made too long, the voltage on the sampling capacitor can decay before the conversion is complete. These timing specifications are provided in the "Electrical Characteristics" section of the device data sheets.
2. Often, the source impedance of the analog signal is high (greater than 2.5 kΩ), so the current drawn from the source by leakage, and to charge the sample capacitor, can affect accuracy. If the input signal does not change too quickly, try putting a 0.1 uF capacitor on the analog input. This capacitor will charge to the analog voltage being sampled and supply the instantaneous current needed to charge the internal holding capacitor.
3. Put the device into Sleep mode before the start of the A/D conversion. The RC clock source selection is required for conversions in Sleep mode. This technique increases accuracy, because digital noise from the CPU and other peripherals is minimized.

Question 2: Do you know of a good reference on A/D Converters?

Answer: A good reference for understanding A/D conversions is the "Analog-Digital Conversion

Handbook third edition, published by Prentice Hall (ISBN 0-13-03-2848-0).

Question 3: My combination of channels/samples and samples/interrupt is greater than the size of the buffer. What will happen to the buffer?

Answer: This configuration is not recommended. The buffer will contain unknown results.

22.13 Related Application Notes

This section lists application notes that are related to this section of the data sheet. These application notes may not be written specifically for the PIC18F device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the 12-Bit A/D Converter with Threshold Detect module are:

AN546, Using the Analog-to-Digital (A/D) Converter (DS00546)

AN557, Four-Channel Digital Voltmeter with Display and Keyboard (DS00557)

AN693, Understanding A/D Converter Performance Specifications (DS00693)

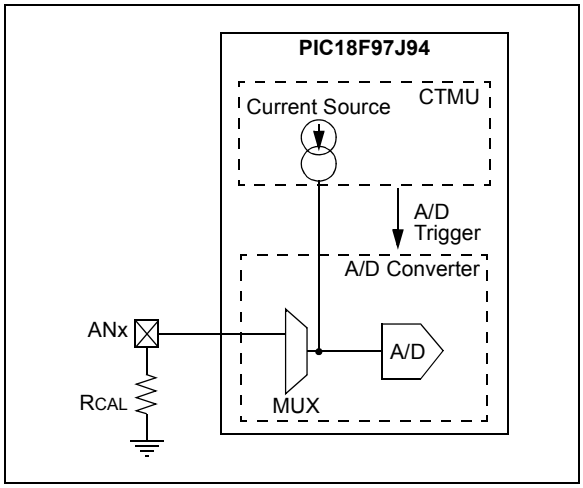
Note:	Visit the Microchip web site (www.microchip.com) for additional application notes and code examples for the PIC18F family of devices.

The CTMU current source may be trimmed with the ITRIMx bits in CTMUCON1, using an iterative process to get the exact current desired. Alternatively, the nominal value without adjustment may be used. That value may be stored by software for use in all subsequent capacitive or time measurements.

To calculate the optimal value for RCAL, the nominal current must be chosen.

For example, if the A/D Converter reference voltage is 3.3V, use 70% of full scale (or 2.31V) as the desired approximate voltage to be read by the A/D Converter. If the range of the CTMU current source is selected to be 0.55 μA, the resistor value needed is calculated as RCAL = 2.31V/0.55 μA, for a value of 4.2 MΩ. Similarly, if the current source is chosen to be 5.5 μA, RCAL would be 420,000Ω, and 42,000Ω if the current source is set to 55 μA.

FIGURE 26-2: CTMU CURRENT SOURCE CALIBRATION CIRCUIT



A value of 70% of full-scale voltage is chosen to make sure that the A/D Converter is in a range that is well above the noise floor. If an exact current is chosen to incorporate the trimming bits from CTMUCON1, the resistor value of RCAL may need to be adjusted accordingly. RCAL also may be adjusted to allow for available resistor values. RCAL should be of the highest precision available in light of the precision needed for the circuit that the CTMU will be measuring. A recommended minimum would be 0.1% tolerance.

The following examples show a typical method for performing a CTMU current calibration.

- Example 26-1 demonstrates how to initialize the A/D Converter and the CTMU.

  This routine is typical for applications using both modules.

- Example 26-2 demonstrates one method for the actual calibration routine.

  This method manually triggers the A/D Converter to demonstrate the entire step-wise process. It is also possible to automatically trigger the conversion by setting the CTMU's CTTRIG bit (CTMUCON<0>).

BZ **Branch if Zero**

Syntax: BZ n

Operands: -128 ≤ n ≤ 127

Operation: if Zero bit is '1',
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0000	nnnn	nnnn

Description: If the Zero bit is '1', then the program will branch.

The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a 2-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: HERE BZ Jump

Before Instruction
PC = address (HERE)
After Instruction
If Zero = 1;
PC = address (Jump)
If Zero = 0;
PC = address (HERE + 2)

CALL **Subroutine Call**

Syntax: CALL k {,s}

Operands: 0 ≤ k ≤ 1048575
s ∈ [0,1]

Operation: (PC) + 4 → TOS,
k → PC<20:1>;
if s = 1
(W) → WS,
(STATUS) → STATUSS,
(BSR) → BSRS

Status Affected: None

Encoding:

1st word (k<7:0>)	1110	110s	$k_7$kkk	kkkk$_0$
2nd word(k<19:8>)	1111	k_{19}kkk	kkkk	kkkk$_8$

Description: Subroutine call of entire 2-Mbyte memory range. First, return address (PC+ 4) is pushed onto the return stack. If 's' = 1, the W, STATUS and BSR registers are also pushed into their respective shadow registers, WS, STATUSS and BSRS. If 's' = 0, no update occurs. Then, the 20-bit value 'k' is loaded into PC<20:1>. CALL is a 2-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>,	Push PC to stack	Read literal 'k'<19:8>, Write to PC
No operation	No operation	No operation	No operation

Example: HERE CALL THERE,1

Before Instruction
PC = address (HERE)
After Instruction
PC = address (THERE)
TOS = address (HERE + 4)
WS = W
BSRS = BSR
STATUSS = STATUS

IORLW **Inclusive OR Literal with W**

Syntax: IORLW k

Operands: $0 \le k \le 255$

Operation: (W) .OR. k → W

Status Affected: N, Z

Encoding:

0000	1001	kkkk	kkkk

Description: The contents of W are ORed with the 8-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example: IORLW 35h

Before Instruction
W = 9Ah
After Instruction
W = BFh

IORWF **Inclusive OR W with f**

Syntax: IORWF f {,d {,a}}

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (W) .OR. (f) → dest

Status Affected: N, Z

Encoding:

0001	00da	ffff	ffff

Description: Inclusive OR W with register 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f'.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: IORWF RESULT, 0, 1

Before Instruction
RESULT = 13h
W = 91h
After Instruction
RESULT = 13h
W = 93h

MOVLW **Move Literal to W**

Syntax: MOVLW k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow W$

Status Affected: None

Encoding:

0000	1110	kkkk	kkkk

Description: The 8-bit literal ‘k’ is loaded into W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write to W

Example: MOVLW 5Ah

After Instruction
W = 5Ah

MOVWF **Move W to f**

Syntax: MOVWF f {,a}

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: $(W) \rightarrow f$

Status Affected: None

Encoding:

0110	111a	ffff	ffff

Description: Move data from W to register ‘f’. Location ‘f’ can be anywhere in the 256-byte bank.

If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank.

If ‘a’ is ‘0’ and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write register ‘f’

Example: MOVWF REG, 0

Before Instruction
W = 4Fh
REG = FFh
After Instruction
W = 4Fh
REG = 4Fh

SUBFSR **Subtract Literal from FSR**

Syntax: SUBFSR f, k

Operands: $0 \le k \le 63$

$f \in [0, 1, 2]$

Operation: FSRf – k → FSRf

Status Affected: None

Encoding:

1110	1001	ffkk	kkkk

Description: The 6-bit literal 'k' is subtracted from the contents of the FSR specified by 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example: SUBFSR 2, 23h

Before Instruction
FSR2 = 03FFh

After Instruction
FSR2 = 03DCh

SUBULNK **Subtract Literal from FSR2 and Return**

Syntax: SUBULNK k

Operands: $0 \le k \le 63$

Operation: FSR2 – k → FSR2,
(TOS) → PC

Status Affected: None

Encoding:

1110	1001	11kk	kkkk

Description: The 6-bit literal 'k' is subtracted from the contents of the FSR2. A RETURN is then executed by loading the PC with the TOS.

The instruction takes two cycles to execute; a NOP is performed during the second cycle.

This may be thought of as a special case of the SUBFSR instruction, where f = 3 (binary '11'); it operates only on FSR2.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination
No Operation	No Operation	No Operation	No Operation

Example: SUBULNK 23h

Before Instruction
FSR2 = 03FFh
PC = 0100h

After Instruction
FSR2 = 03DCh
PC = (TOS)

TABLE 30-2: DC CHARACTERISTICS: POWER-DOWN AND SUPPLY CURRENT PIC18FXXJ94 (INDUSTRIAL)

PIC18FXXJ94 Family (Industrial)				**Standard Operating Conditions: 2V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial		
Param No.	**Typ.(1)**	**Max.**	**Units**	**Conditions**		
DC60	3.7	7.0	µA	-40°C	2.0V	Sleep(2)
	3.7	7.0	µA	+25°C		
	5.0	9.0	µA	+60°C		
	9.0	18	µA	+85°C		
	3.7	8.0	µA	-40°C	3.3V	
	3.7	8.0	µA	+25°C		
	5.0	11.0	µA	+60°C		
	10	20	µA	+85°C		
DC61	0.07	0.55	µA	-40°C	2.0V	Retention Sleep or Retention Deep Sleep(3)
	0.09	0.55	µA	+25°C		
	2.0	3.2	µA	+60°C		
	7.0	8.5	µA	+85°C		
	0.10	0.65	µA	-40°C	3.3V	
	0.15	0.65	µA	+25°C		
	2.0	3.5	µA	+60°C		
	7.2	9.0	µA	+85°C		
DC70	0.06	0.5	µA	-40°C	2.0V	Deep Sleep
	0.08	0.5	µA	+25°C		
	0.21	0.8	µA	+60°C		
	0.41	1.5	µA	+85°C		
	0.09	0.6	µA	-40°C	3.3V	
	0.11	0.6	µA	+25°C		
	0.42	1.2	µA	+60°C		
	0.8	4.8	µA	+85°C		
	0.4	3.0	µA	-40°C TO +85°C	0	RTCC with VBAT mode (LPRC or SOSC)(4)

Note 1: Data in the Typical column is at 3.3V, 25°C; typical parameters are for design guidance only and are not tested.

2: Retention regulator is disabled; SRETEN (RCON4<4>= 0), $\overline{\text{RETEN}}$ (CONFIG7L<0>= 1).

3: Retention regulator is enabled; SRETEN (RCON4<4> = 1), $\overline{\text{RETEN}}$ (CONFIG7L<0> = 0).

4: VBAT pin is connected to the battery and RTCC is running with VDD = 0.

TABLE 30-3: DC CHARACTERISTICS: POWER-DOWN AND SUPPLY CURRENT PIC18F97J94 FAMILY (INDUSTRIAL)

Param No.	**Device**	**Typ.**	**Max.**	**Units**	**Conditions**		
	Supply Current (IDD)						
	All Devices	22	55	µA	-40°C to +85°C	VDD = 2.0V	FOSC = 31 kHz, **RC_RUN**
		23	56	µA	-40°C to +85°C	VDD = 3.3V	
		21	54	µA	-40°C to +85°C	VDD = 2.0V	FOSC = 31 kHz, **RC_IDLE**
		22	55	µA	-40°C to +85°C	VDD = 3.3V	

TABLE 30-4: DC CHARACTERISTICS: POWER-DOWN AND SUPPLY CURRENT PIC18F97J94 FAMILY (INDUSTRIAL)

Param No.	Device	Typ.	Max.	Units	Conditions		
	Supply Current (I_{DD})						
	All Devices	22	55	µA	-40°C to +85°C	V_{DD} = 2.0V	F_{OSC} = 32 kHz, **SEC_RUN**
		23	56	µA	-40°C to +85°C	V_{DD} = 3.3V	
		21	54	µA	-40°C to +85°C	V_{DD} =2.0V	F_{OSC} = 32 kHz, **SEC_IDLE**
		22	55	µA	-40°C to +85°C	V_{DD} = 3.3V	

TABLE 30-5: DC CHARACTERISTICS: POWER-DOWN AND SUPPLY CURRENT PIC18F97J94 FAMILY (INDUSTRIAL)

Param No.	Device	Typ.	Max.	Units	Conditions		
	Supply Current (I_{DD})						
	All Devices	325	430	µA	-40°C to +85°C	V_{DD} = 2.0V	F_{OSC} = 1 MHz, **RC_RUN**
		325	430	µA	-40°C to +85°C	V_{DD} = 3.3V	
		540	700	µA	-40°C to +85°C	V_{DD} = 2.0V	F_{OSC} = 4 MHz, **RC_RUN**
		540	700	µA	-40°C to +85°C	V_{DD} = 3.3V	
		820	1000	µA	-40°C to +85°C	V_{DD} = 2.0V	F_{OSC} = 8 MHz, **RC_RUN**
		825	1000	µA	-40°C to +85°C	V_{DD} = 3.3V	
		275	370	µA	-40°C to +85°C	V_{DD} = 2.0V	F_{OSC} = 1 MHz, **RC_IDLE**
		275	370	µA	-40°C to +85°C	V_{DD} = 3.3V	
		345	440	µA	-40°C to +85°C	V_{DD} = 2.0V	F_{OSC} = 4 MHz, **RC_IDLE**
		345	440	µA	-40°C to +85°C	V_{DD} = 3.3V	
		435	620	µA	-40°C to +85°C	V_{DD} = 2.0V	F_{OSC} = 8 MHz, **RC_IDLE**
		435	620	µA	-40°C to +85°C	V_{DD} = 3.3V	

FIGURE 30-15: EXAMPLE SPI SLAVE MODE TIMING (CKE = 1)

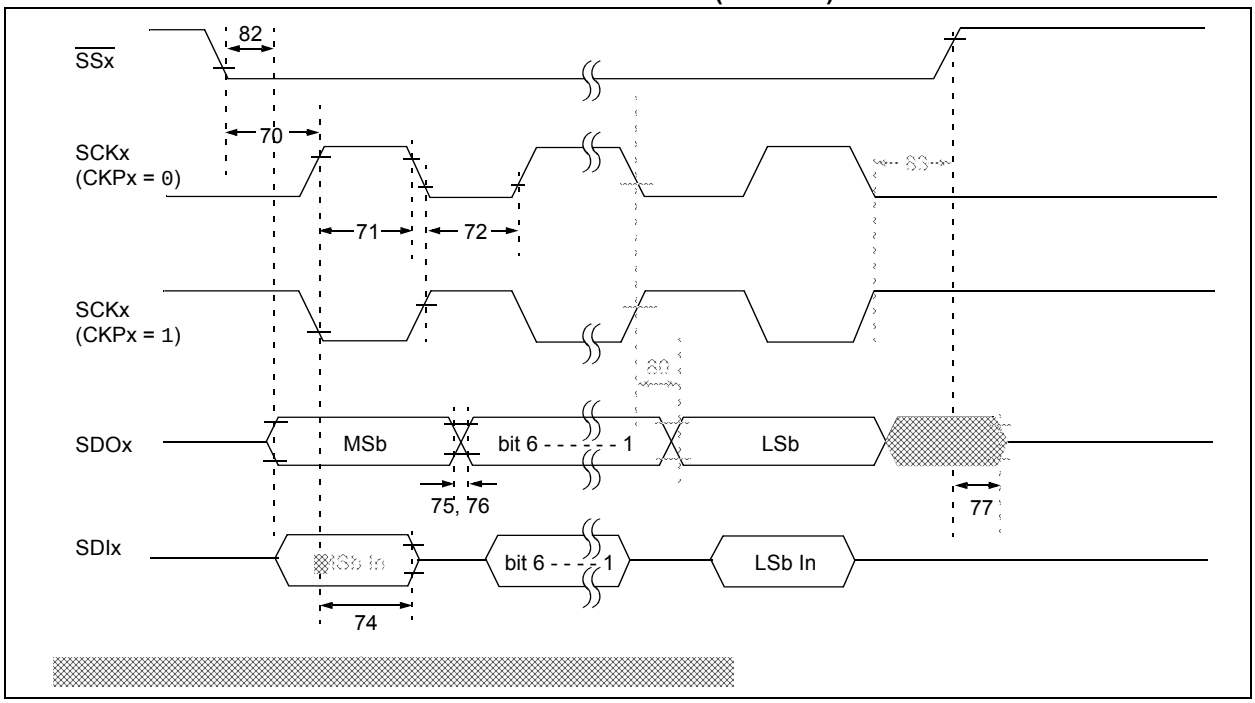


TABLE 30-33: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)

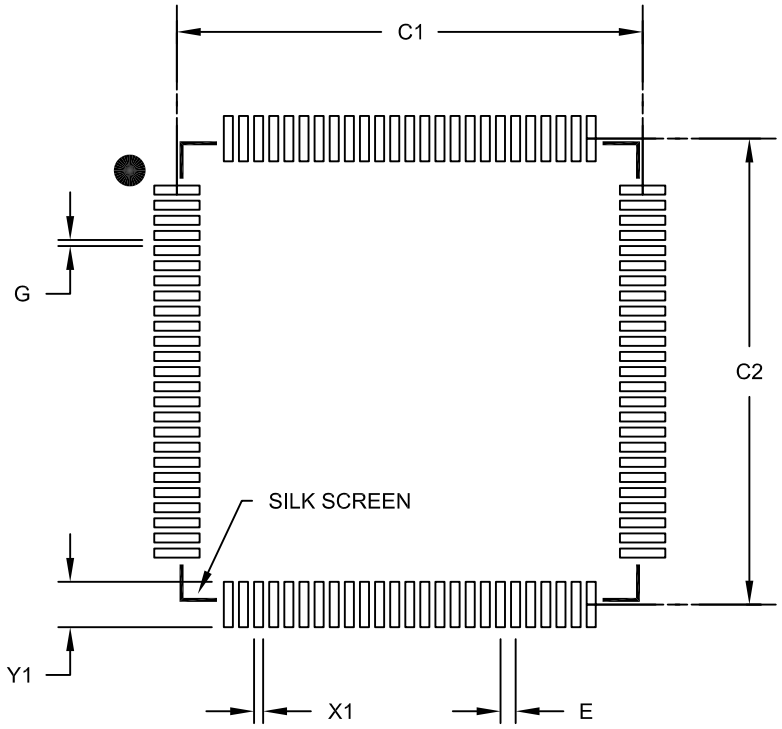
Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SSx}$ ↓ to SCKx ↓ or SCKx ↑ Input		3 Tcy	—	ns	
70A	TssL2WB	$\overline{SSx}$ to Write to SSPBUF		3 Tcy	—	ns	
71	TscH	SCKx Input High Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns	
71A			Single Byte	40	—	ns	**(Note 1)**
72	TscL	SCKx Input Low Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns	
72A			Single Byte	40	—	ns	**(Note 1)**
73A	Tb2b	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2		1.5 Tcy + 40	—	ns	**(Note 2)**
74	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge		40	—	ns	
75	TdoR	SDOx Data Output Rise Time		—	25	ns	
76	TdoF	SDOx Data Output Fall Time		—	25	ns	
77	TssH2doZ	$\overline{SSx}$ ↑ to SDOx Output High-Impedance		10	50	ns	
78	TscR	SCKx Output Rise Time (Master mode)		—	25	ns	
79	TscF	SCKx Output Fall Time (Master mode)		—	25	ns	
80	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge		—	50	ns	
82	TssL2doV	SDOx Data Output Valid after $\overline{SSx}$ ↓ Edge		—	50	ns	
83	TscH2ssH, TscL2ssH	$\overline{SSx}$ ↑ after SCKx Edge		1.5 Tcy + 40	—	ns	

Note 1: Requires the use of Parameter #73A.

2: Only if Parameter #71A and #72A are used.

100-Lead Plastic Thin Quad Flatpack (PF) - 14x14x1 mm Body 2.00 mm Footprint [TQFP]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Contact Pad Spacing	C1		15.40	
Contact Pad Spacing	C2		15.40	
Contact Pad Width (X100)	X1			0.30
Contact Pad Length (X100)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2110B