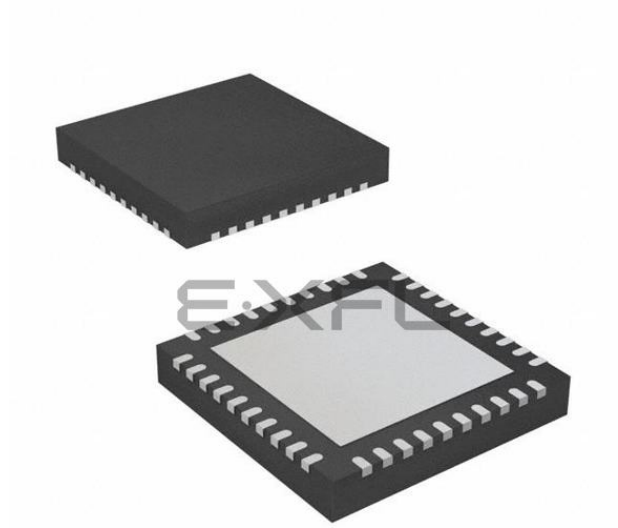


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

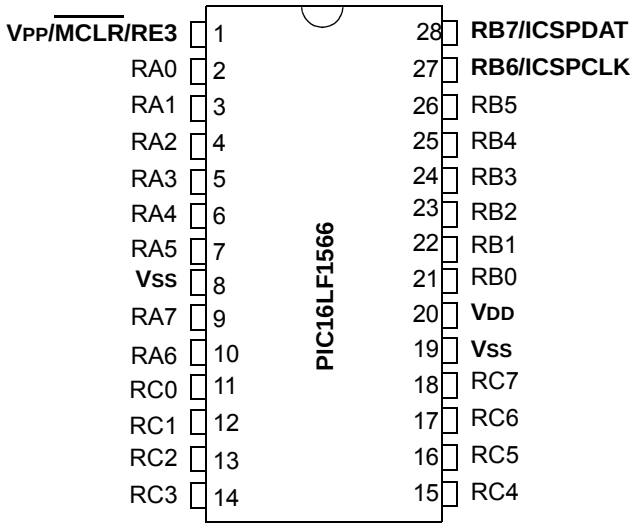
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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, POR, PWM, WDT
Number of I/O	36
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 34x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	40-UFQFN Exposed Pad
Supplier Device Package	40-UQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1567t-i-mv

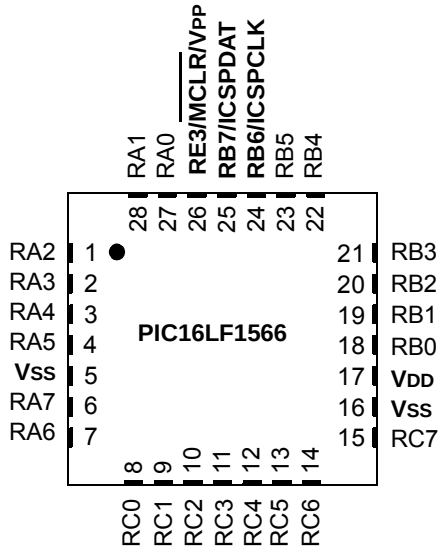
PIN DIAGRAMS

FIGURE 1: 28-PIN SPDIP, SOIC, SSOP DIAGRAM FOR PIC16LF1566



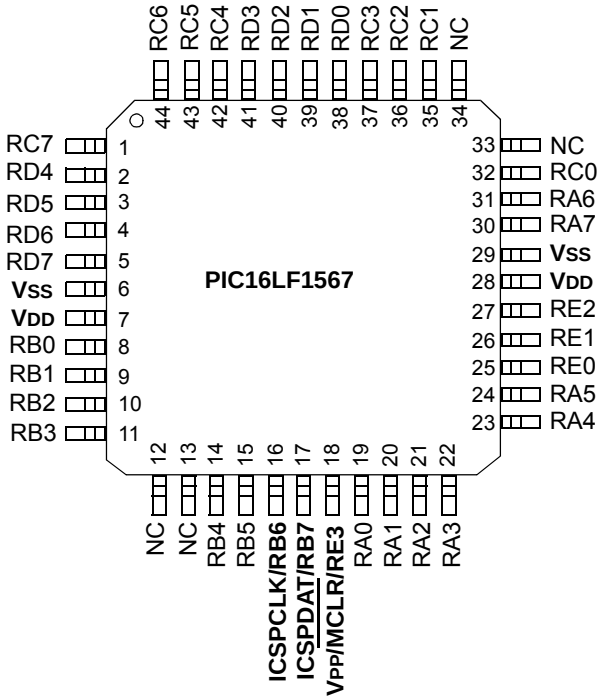
Note: See Table 2 for the pin allocation tables.

FIGURE 2: 28-PIN UQFN DIAGRAM FOR PIC16LF1566



Note: See Table 2 or the pin allocation tables.

FIGURE 5: 44-PIN TQFP DIAGRAM FOR PIC16LF1567



Note: See Table 3 for the pin allocation table.

TABLE 3-5: PIC16LF1566/1567 MEMORY MAP, BANKS 8-15

	BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	CPU Core Register, see Table 3-2 for specifics															
401h																
402h																
403h																
404h																
405h																
406h																
407h																
408h																
409h																
40Ah																
40Bh																
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—	
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—	
40Eh	—	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	—	70Eh	—	78Eh	—	
40Fh	—	48Fh	—	50Fh	—	58Fh	—	60Fh	—	68Fh	—	70Fh	—	78Fh	—	
410h	—	490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—	
411h	—	491h	—	511h	—	591h	—	611h	PWM1DCL	691h	ADCTX	711h	AD1CON0	791h	AD2CON0	
412h	—	492h	—	512h	—	592h	—	612h	PWM1DCH	692h	AD1TX0	712h	ADCOMCON	792h	—	
413h	—	493h	—	513h	—	593h	—	613h	PWM1CON	693h	AD1TX1	713h	AD1CON2	793h	AD2CON2	
414h	—	494h	—	514h	—	594h	—	614h	PWM2DCL	694h	AD2TX0	714h	AD1CON3	794h	AD2CON3	
415h	TMR4	495h	—	515h	—	595h	—	615h	PWM2DCH	695h	AD2TX1	715h	ADSTAT	795h	—	
416h	PR4	496h	—	516h	—	596h	—	616h	PWM2CON	696h	—	716h	AD1PRECON	796h	AD2PRECON	
417h	T4CON	497h	—	517h	—	597h	—	617h	—	697h	—	717h	AD1ACQCON	797h	AD2ACQCON	
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	AD1GRD	798h	AD2GRD	
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	AD1CAPCON	799h	AD2CAPCON	
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	AAD1RES0L	79Ah	AAD2RES0L	
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	AAD1RES0H	79Bh	AAD2RES0H	
41Ch	—	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	AAD1RES1L	79Ch	AAD2RES1L	
41Dh	—	49Dh	—	51Dh	—	59Dh	—	61Dh	PWMTMRS	69Dh	—	71Dh	AAD1RES1H	79Dh	AAD2RES1H	
41Eh	—	49Eh	—	51Eh	—	59Eh	—	61Eh	PWM1AOE	69Eh	—	71Eh	AD1CH0	79Eh	AD2CH0	
41Fh	—	49Fh	—	51Fh	—	59Fh	—	61Fh	PWM2AOE	69Fh	—	71Fh	AD1CH1	79Fh	AD2CH1	
420h	General Purpose Register 80 Bytes	4A0h	General Purpose Register 80 Bytes	520h	General Purpose Register 80 Bytes	5A0h	General Purpose Register 80 Bytes	620h 64Fh	General Purpose Register 48 Bytes	6A0h	Unimplemented Read as '0'	720h	Unimplemented Read as '0'	7A0h	Unimplemented Read as '0'	
46Fh		4EFh		56Fh		5EFh		650h 66Fh	Unimplemented Read as '0'	6EFh		76Fh		7EFh		
470h	Accesses 70h – 7Fh	4F0h	Accesses 70h – 7Fh	570h	Accesses 70h – 7Fh	5F0h	Accesses 70h – 7Fh	670h	Accesses 70h – 7Fh	6F0h	Accesses 70h – 7Fh	770h	Accesses 70h – 7Fh	7F0h	Accesses 70h – 7Fh	
47Fh		4FFh		57Fh		5FFh		67Fh		6FFh		77Fh		7FFh		

Note 1: These ADC registers are the same as the registers in Bank 1.

TABLE 3-11: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
Bank 5											
280h	INDF0(1)	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
281h	INDF1(1)	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
282h	PCL(1)	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
283h	STATUS(1)	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu
284h	FSR0L(1)	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
285h	FSR0H(1)	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
286h	FSR1L(1)	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
287h	FSR1H(1)	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
288h	BSR(1)	—	—	—	BSR<4:0>					---0 0000	---0 0000
289h	WREG(1)	Working Register								0000 0000	uuuu uuuu
28Ah	PCLATH(1)	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
28Bh	INTCON(1)	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000
28Ch	—	Unimplemented								—	—
28Dh	—	Unimplemented								—	—
28Eh	—	Unimplemented								—	—
28Fh	—	Unimplemented								—	—
290h	—	Unimplemented								—	—
291h	—	Unimplemented								—	—
292h	—	Unimplemented								—	—
293h	—	Unimplemented								—	—
294h	—	Unimplemented								—	—
295h	—	Unimplemented								—	—
296h	—	Unimplemented								—	—
297h	—	Unimplemented								—	—
298h	—	Unimplemented								—	—
299h	—	Unimplemented								—	—
29Ah	—	Unimplemented								—	—
29Bh	—	Unimplemented								—	—
29Ch	—	Unimplemented								—	—
29Dh	—	Unimplemented								—	—
29Eh	—	Unimplemented								—	—
29Fh	—	Unimplemented								—	—

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

Note 1: These registers can be accessed from any bank.
2: PIC16LF1567.
3: These registers/bits are available at two address locations, in Bank 1 and Bank 14.
4: PIC16LF1566 only.
5: Unimplemented, read as '1'.

TABLE 3-11: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
Bank 8											
400h	INDF0(1)	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
401h	INDF1(1)	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
402h	PCL(1)	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
403h	STATUS(1)	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu
404h	FSR0L(1)	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
405h	FSR0H(1)	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
406h	FSR1L(1)	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
407h	FSR1H(1)	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
408h	BSR(1)	—	—	—	BSR<4:0>					---0 0000	---0 0000
409h	WREG(1)	Working Register								0000 0000	uuuu uuuu
40Ah	PCLATH(1)	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
40Bh	INTCON(1)	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000
40Ch to 414h	—	Unimplemented								—	—
415h	TMR4	TMR4								0000 0000	0000 0000
416h	PR4	PR4								11111111	11111111
417h	T4CON	—	T4OUTPS				TMR4ON	T4CKPS		-000 0000	-000 0000
418h to 41Fh	—	Unimplemented								—	—

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

Note 1: These registers can be accessed from any bank.
2: PIC16LF1567.
3: These registers/bits are available at two address locations, in Bank 1 and Bank 14.
4: PIC16LF1566 only.
5: Unimplemented, read as '1'.

11.5 PORTB Registers (PIC16LF1567 Only)

11.5.1 DATA REGISTER

PORTB is a 4-bit wide, bidirectional port. The corresponding data direction register is TRISB (Register 11-7). Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input (i.e., disable the output driver). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). Example 11-1 shows how to initialize an I/O port.

Reading the PORTB register (Register 11-6) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch (LATB).

11.5.2 DIRECTION CONTROL

The TRISB register (Register 11-7) controls the PORTB pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISB register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

11.5.3 ANALOG CONTROL

The ANSELB register (Register 11-9) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELB bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELB bits has no effect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

Note:	The ANSELB bits default to the Analog mode after Reset. To use any pins as digital general purpose or peripheral inputs, the corresponding ANSELx bits must be initialized to '0' by user software.

11.5.4 PORTB FUNCTIONS AND OUTPUT PRIORITIES

Each PORTB pin is multiplexed with other functions. The pins, their combined functions and their output priorities are shown in Table 11-5.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the highest priority.

Analog input functions, such as ADC and comparator inputs, are not shown in the priority lists. These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below in Table 11-5.

TABLE 11-5: PORTB OUTPUT PRIORITY

Pin Name	Function Priority(1)
RB0	INT PWM20 RB0
RB1	PWM21 RB1
RB2	PWM22 RB2
RB3	PWM23 RB3
RB4	ADxGRDA RB4
RB5	ADxGRDA RB5
RB6	ICSPCLK ADxGRDB RB6
RB7	ICSPDAT ADxGRDB RB7

Note 1: Priority listed from highest to lowest.

REGISTER 16-3: AD1CH1: HARDWARE CVD 1 SECONDARY CHANNEL SELECT REGISTER(1,2,3,4)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CH35(5)	CH34(5)	CH33(5)	CH32(5)	CH31(5)	CH30(5)	CH19	CH18
bit 7							bit 0

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

bit 7-0 **CHx:** Channel x to A/S 1 Connection(1, 2, 3, 4)

1 = ANx is connected to A/D 1

0 = ANx is not connected to A/D 1

Note 1: This register selects secondary channels which are connected in parallel to the primary channel selected in AD1CON0. Precharge bias is applied to both the primary and secondary channels.

2: If the same channel is selected as both primary and secondary then the selection as primary takes precedence.

3: Enabling these bits automatically overrides the corresponding TRISx bit to tri-state the selected pin.

4: In the same way that the CHS bits in AD1CON0 only close the switch when the ADC is enabled, these connections and the TRISx overrides are only active if the ADC is enabled by setting ADxON.

5: PIC16LF1567 only. Unimplemented/ Read as '0' on PIC16LF1566.

REGISTER 16-4: AD2CH0: HARDWARE CVD 2 SECONDARY CHANNEL SELECT REGISTER(1,2,3,4)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20
bit 7							bit 0

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

bit 7-0 **CHx:** Channel x to A/D 2 Connection bit(1,2,3,4,5)
1 = ANx is connected to A/D 2
0 = ANx is not connected to A/D 2

Note 1: This register selects secondary channels which are connected in parallel to the primary channel selected in ADxCON1. Precharge bias is applied to both the primary and secondary channels.
2: If the same channel is selected as both primary (ADxCON1) and secondary then the selection as primary takes precedence.
3: Enabling these bits automatically overrides the corresponding TRISx.x bit to tri-state the selected pin.
4: In the same way that the CHSx bits in ADCON0 only close the switch when the A/D is enabled, these connections and the TRIS overrides are only active if the A/D is enabled by setting ADxON.

REGISTER 16-5: AD2CH1: ANALOG-TO-DIGITAL (A/D) 2 SECONDARY CHANNEL SELECT REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CH45(5)	CH44(5)	CH43(5)	CH42(5)	CH41(5)	CH40	CH29	CH28
bit 7							bit 0

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

bit 7-0 **CHx:** Channel x to A/D 2 Connection bit(1,2,3,4)
1 = ANx is connected to A/D 2
0 = ANx is not connected to A/D 2

Note 1: This register selects secondary channels which are connected in parallel to the primary channel selected in ADxCON1. Precharge bias is applied to both the primary and secondary channels.
2: If the same channel is selected as both primary (ADxCON1) and secondary then the selection as primary takes precedence.
3: Enabling these bits automatically overrides the corresponding TRISx.x bit to tri-state the selected pin.
4: In the same way that the CHSx bits in ADCON0 only close the switch when the A/D is enabled, these connections and the TRIS overrides are only active if the A/D is enabled by setting ADxON.
5: PIC16LF1567 only. Unimplemented / Read as '0' on PIC16LF1566

REGISTER 16-12: ADxGRD: HARDWARE CVD GUARD RING CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	U-0	R/W-0/0
GRDxBOE(2)	GRDxAOE(2)	GRDxPOL(1,2)	—	—	—	—	TXxPOL
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 **GRDxBOE:** Guard Ring B Output Enable bit(2,3,5)
- 1 = ADC guard ring output is enabled to ADxGRDB(6) pin. Its corresponding TRISx bit must be clear.
- 0 = No ADC guard ring function to this pin is enabled

bit 6 **GRDxAOE:** Guard Ring A Output Enable bit(1,3,5)
- 1 = ADC Guard Ring Output is enabled to ADxGRDA(6) pin. Its corresponding TRISx, x bit must be clear.
- 0 = No ADC Guard Ring function is enabled

bit 5 **GRDxPOL:** Guard Ring Polarity Selection bit(4)
- 1 = ADCx guard ring outputs start as digital high during precharge stage
- 0 = ADCx guard ring outputs start as digital low during precharge stage

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

bit 0 **TXxPOL:** ADC x TX Polarity Select(3,4,5). ADxTXy registers determine location of TX pins.
- 1 = TX starts as digital high during Precharge stage
- 0 = TX starts as digital low during Precharge stage

Note 1: If precharge is enabled (ADxPRE! = '000000'), then Guard A switches polarity at the start of Acquisition / Charge Share. If precharge is disabled, then Guard A switches polarity as soon as the GO/DONEx bit is set.

2: Output function "B" is constant throughout all stages of the conversion cycle. In a dual sample setup it will switch polarity at the start of precharge.

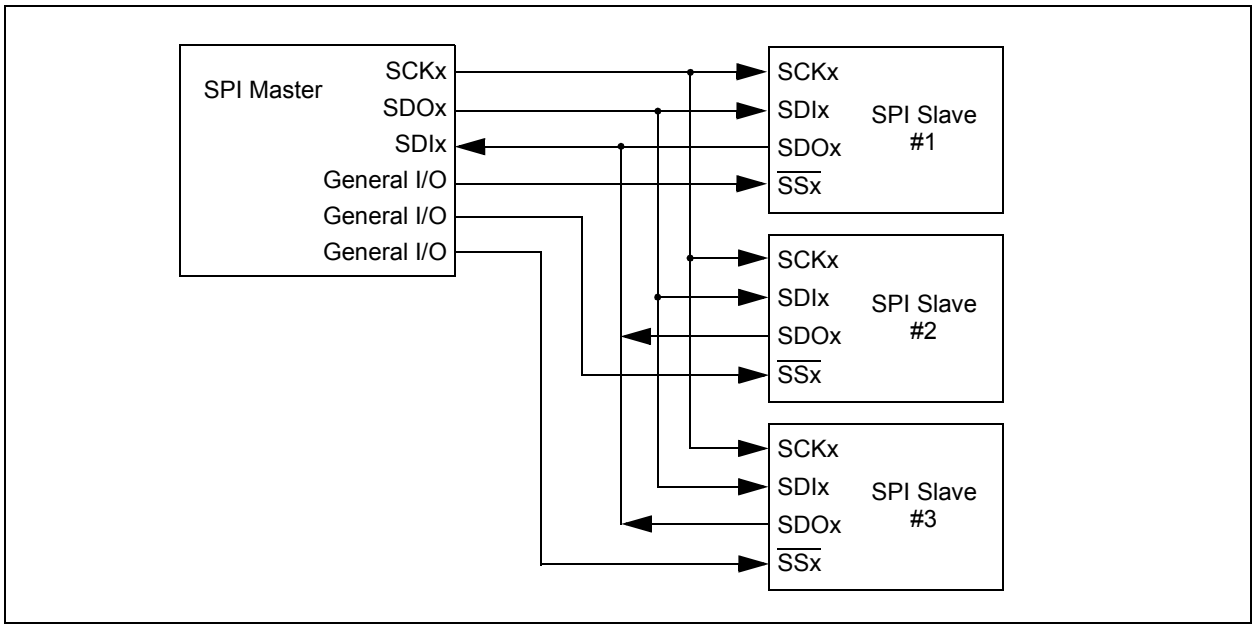
3: The corresponding TRISx,x bit must be set to '0' to enable output.

4: When the ADxDSEN = 1 and ADxIPEN = 1; the polarity of this output is inverted for the second conversion time. The stored bit value does not change.

5: Outputs are maintained while ADxON = 1.

6: ADxGRD pin locations are selectable in APFCON, Register 3-9.

FIGURE 20-4: SPI MASTER AND MULTIPLE SLAVE CONNECTION



20.2.1 SPI MODE REGISTERS

The MSSPx module has five registers for SPI mode operation. These are:

- MSSPx STATUS register (SSPxSTAT)
- MSSPx Control register 1 (SSPxCON1)
- MSSPx Control register 3 (SSPxCON3)
- MSSPx Data Buffer register (SSPxBUF)
- MSSPx Address register (SSPxADD)
- MSSPx Shift register (SSPxSR) (Not directly accessible)

SSPxCON1 and SSPxSTAT are the control and STATUS registers in SPI mode operation. The SSPxCON1 register is readable and writable. The lower six bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

In one SPI master mode, SSPxADD can be loaded with a value used in the Baud Rate Generator. More information on the Baud Rate Generator is available in **Section 20.7 "Baud Rate Generator"**.

SSPxSR is the shift register used for shifting data in and out. SSPxBUF provides indirect access to the SSPxSR register. SSPxBUF is the buffer register to which data bytes are written, and from which data bytes are read.

In receive operations, SSPxSR and SSPxBUF together create a buffered receiver. When SSPxSR receives a complete byte, it is transferred to SSPxBUF and the SSPxIF interrupt is set.

During transmission, the SSPxBUF is not buffered. A write to SSPxBUF will write to both SSPxBUF and SSPxSR.

20.2.2 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, SSPEN of the SSPxCON1 register, must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPxCONx 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 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
- $\overline{SSx}$ must have corresponding TRIS bit set

20.2.3 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 20-5) 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).

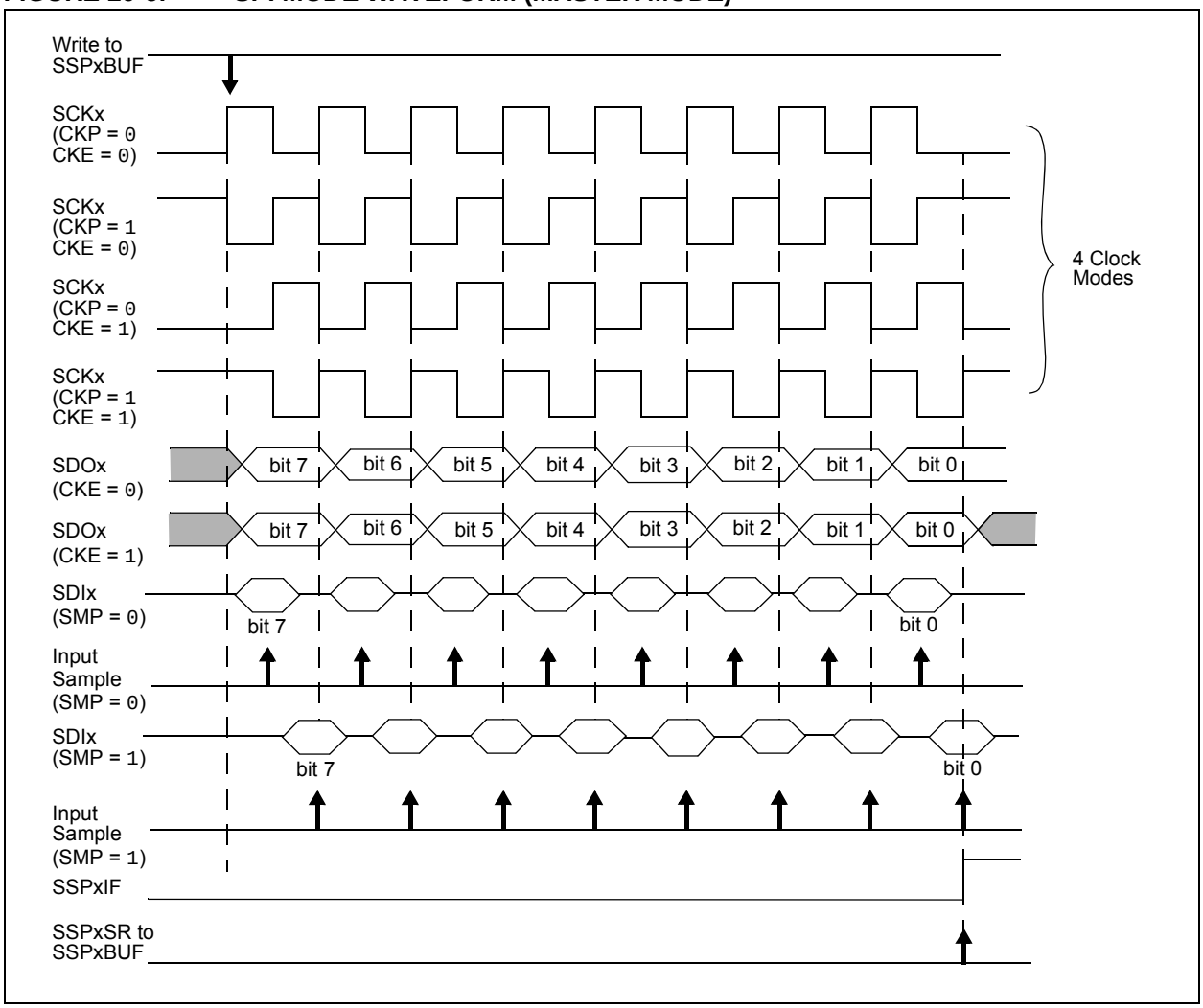
The clock polarity is selected by appropriately programming the CKP bit of the SSPxCON1 register and the CKE bit of the SSPxSTAT register. This then, would give waveforms for SPI communication as shown in Figure 20-6, Figure 20-8, Figure 20-9 and Figure 20-10, where the MSb is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 * TCY)
- FOSC/64 (or 16 * TCY)
- Timer2 output/2
- Fosc/(4 * (SSPxADD + 1))

Figure 20-6 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 20-6: SPI MODE WAVEFORM (MASTER MODE)



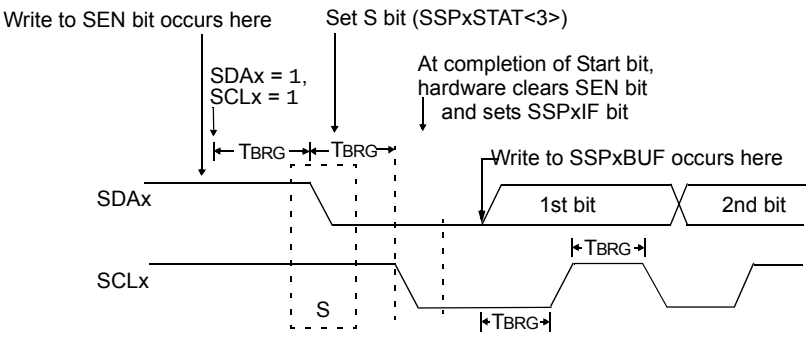
20.6.4 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition (Figure 20-26), the user sets the Start Enable bit, SEN bit of the SSPxCON2 register. If the SDAx and SCLx pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and starts its count. If SCLx and SDAx are both sampled high when the Baud Rate Generator times out (TBRG), the SDAx pin is driven low. The action of the SDAx being driven low while SCLx is high is the Start condition and causes the S bit of the SSPxSTAT1 register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit of the SSPxCON2 register will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDAx line held low and the Start condition is complete.

Note 1: If at the beginning of the Start condition, the SDAx and SCLx pins are already sampled low, or if during the Start condition, the SCLx line is sampled low before the SDAx line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLxIF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.

2: The Philips I²C Specification states that a bus collision cannot occur on a Start.

FIGURE 20-26: FIRST START BIT TIMING



REGISTER 20-2: SSPxCON1: SSPx CONTROL REGISTER 1

R/C/HS-0/0	R/C/HS-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
WCOL	SSPxOV	SSPEN	CKP	SSPM<3:0>			
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 Resets
'1' = Bit is set	'0' = Bit is cleared	HS = Bit is set by hardware C = User cleared

bit 7 **WCOL:** Write Collision Detect bit

Master mode:

1 = A write to the SSPxBUF register was attempted while the I^2C conditions were not valid for a transmission to be started

0 = No collision

Slave mode:

1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)

0 = No collision

bit 6 **SSPxOV:** Receive Overflow Indicator bit(1)

In SPI mode:

1 = A new byte is received while the SSPxBUF register is still holding the previous data. In case of overflow, the data in SSPxSR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSPxBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register (must be cleared in software).

0 = No overflow

In I^2C mode:

1 = A byte is received while the SSPxBUF register is still holding the previous byte. SSPxOV is a "don't care" in Transmit mode (must be cleared in software).

0 = No overflow

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

In both modes, when enabled, these pins must be properly configured as input or output

In SPI mode:

1 = Enables serial port and configures SCKx, SDOx, SDIx and $\overline{SSx}$ as the source of the serial port pins(2)

0 = Disables serial port and configures these pins as I/O port pins

In I^2C mode:

1 = Enables the serial port and configures the SDAx and SCLx pins as the source of the serial port pins(3)

0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit

In SPI mode:

1 = Idle state for clock is a high level

0 = Idle state for clock is a low level

In I^2C Slave mode:

SCLx release control

1 = Enable clock

0 = Holds clock low (clock stretch). (Used to ensure data setup time.)

In I^2C Master mode:

Unused in this mode

REGISTER 20-2: SSPxCON1: SSPx CONTROL REGISTER 1 (CONTINUED)

bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits
1111 = I^2C Slave mode, 10-bit address with Start and Stop bit interrupts enabled
1110 = I^2C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1101 = Reserved
1100 = Reserved
1011 = I^2C firmware controlled Master mode (Slave idle)
1010 = SPI Master mode, clock = FOSC/(4 * (SSPxADD+1))**(5)**
1001 = Reserved
1000 = I^2C Master mode, clock = FOSC/(4 * (SSPxADD+1))**(4)**
0111 = I^2C Slave mode, 10-bit address
0110 = I^2C Slave mode, 7-bit address
0101 = SPI Slave mode, clock = SCKx pin, $\overline{SSx}$ pin control disabled, $\overline{SSx}$ can be used as I/O pin
0100 = SPI Slave mode, clock = SCKx pin, $\overline{SSx}$ pin control enabled
0011 = SPI Master mode, clock = TMR2 output/2
0010 = SPI Master mode, clock = FOSC/64
0001 = SPI Master mode, clock = FOSC/16
0000 = SPI Master mode, clock = FOSC/4

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.
2: When enabled, these pins must be properly configured as input or output.
3: When enabled, the SDAx and SCLx pins must be configured as inputs.
4: SSPxADD values of 0, 1 or 2 are not supported for I^2C mode.
5: SSPxADD value of '0' is not supported. Use SSPM = 0000 instead.

DECFSZ **Decrement f, Skip if 0**

Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2-cycle instruction.

GOTO **Unconditional Branch**

Syntax:	[*label*] GOTO k
Operands:	$0 \le k \le 2047$
Operation:	k → PC<10:0> PCLATH<6:3> → PC<14:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a 2-cycle instruction.

INCF **Increment f**

Syntax:	[*label*] INCF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

INCFSZ **Increment f, Skip if 0**

Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2-cycle instruction.

IORLW **Inclusive OR literal with W**

Syntax:	[*label*] IORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the 8-bit literal 'k'. The result is placed in the W register.

IORWF **Inclusive OR W with f**

Syntax:	[*label*] IORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .OR. (f) → (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

LSLF **Logical Left Shift**

Syntax: [*label*] LSLF f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f<7>) → C
(f<6:0>) → dest<7:1>
0 → dest<0>

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the left through the Carry flag. A '0' is shifted into the LSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

LSRF **Logical Right Shift**

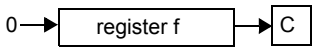
Syntax: [*label*] LSRF f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: 0 → dest<7>
(f<7:1>) → dest<6:0>,
(f<0>) → C,

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the right through the Carry flag. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.



MOVF **Move f**

Syntax: [*label*] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Description: The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example:

```
MOVF    FSR, 0
```

After Instruction
W = value in FSR register
Z = 1

MOVIW **Move INDFn to W**

Syntax: [*label*] MOVIW ++FSRn
[*label*] MOVIW --FSRn
[*label*] MOVIW FSRn++
[*label*] MOVIW FSRn--
[*label*] MOVIW k[FSRn]

Operands: n ∈ [0,1]
mm ∈ [00,01, 10, 11]
-32 ≤ k ≤ 31

Operation: INDFn → W
Effective address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)
- Unchanged

Status Affected: Z

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

MOVLB **Move literal to BSR**

Syntax: [*label*] MOVLB k

Operands: 0 ≤ k ≤ 31

Operation: k → BSR

Status Affected: None

Description: The 5-bit literal 'k' is loaded into the Bank Select Register (BSR).

MOVLP **Move literal to PCLATH**

Syntax: [*label*] MOVLP k

Operands: 0 ≤ k ≤ 127

Operation: k → PCLATH

Status Affected: None

Description: The 7-bit literal 'k' is loaded into the PCLATH register.

MOVLW **Move literal to W**

Syntax: [*label*] MOVLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W)

Status Affected: None

Description: The 8-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.

Words: 1

Cycles: 1

Example:
```
MOVLW   0x5A
```
After Instruction
W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: 0 ≤ f ≤ 127

Operation: (W) → (f)

Status Affected: None

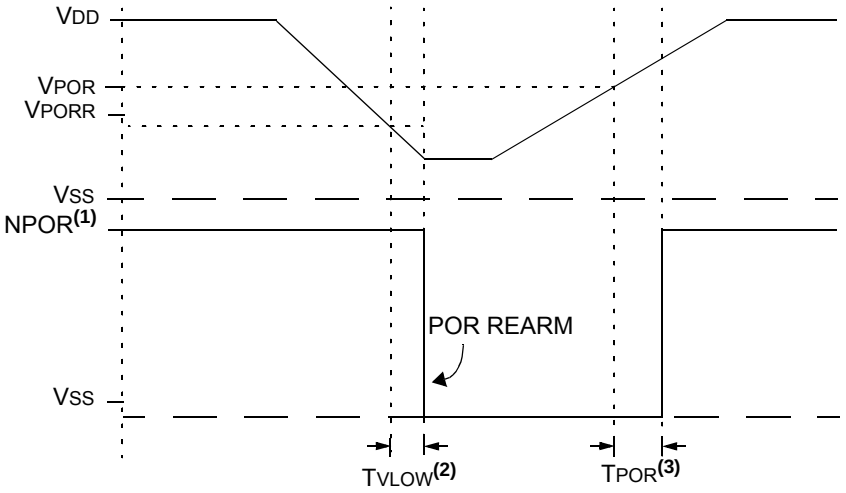
Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example:
```
MOVWF   OPTION_REG
```
Before Instruction
OPTION_REG = 0xFF
W = 0x4F
After Instruction
OPTION_REG = 0x4F
W = 0x4F

FIGURE 25-2: POR AND POR REARM WITH SLOW RISING VDD



Note 1: When NPOR is low, the device is held in Reset.
2: TPOR 1 μs typical.
3: TVLOW 2.7 μs typical.

TABLE 25-2: SUPPLY CURRENT (IDD)

PIC16LF1566/1567			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param. No.	Device Characteristics	Min.	Typ.†	Max.	Units	Conditions	
						VDD	Note
	Supply Current (IDD)(1, 2)						
D010		—	2.5	18	μA	1.8	FOSC = 31 kHz LFINTOSC mode
		—	4	20	μA	3.0	
D011		—	0.35	0.70	mA	1.8	FOSC = 8 MHz HFINTOSC mode
		—	0.55	1.10	mA	3.0	
D012		—	0.5	1.2	mA	1.8	FOSC = 16 MHz HFINTOSC mode
		—	0.8	1.75	mA	3.0	
D013		—	1.5	3.5	mA	3.0	FOSC = 32 MHz HFINTOSC mode with PLL
D014		—	3	17	μA	1.8	FOSC = 32 kHz ECL mode
		—	5	20	μA	3.0	
D015		—	12	40	μA	1.8	FOSC = 500 kHz ECL mode
		—	18	60	μA	3.0	
D016		—	25	65	μA	1.8	FOSC = 1 MHz ECM mode
		—	40	100	μA	3.0	

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

Note 1: The test conditions for all IDD measurements in active operation mode are: CLKIN = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; $\overline{MCLR}$ = VDD; WDT disabled.
2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption.

27.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

27.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

27.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

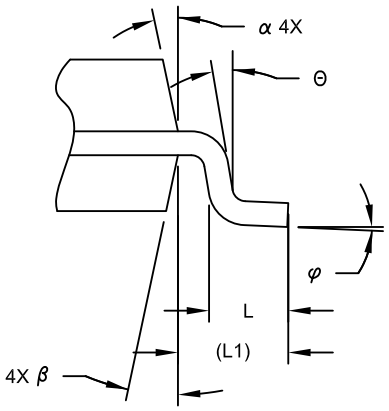
27.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

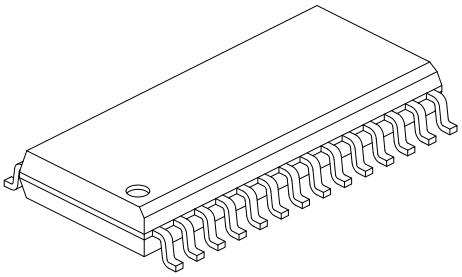
- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

28-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



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
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		17.90 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	0.18	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. 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.
4. 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.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

