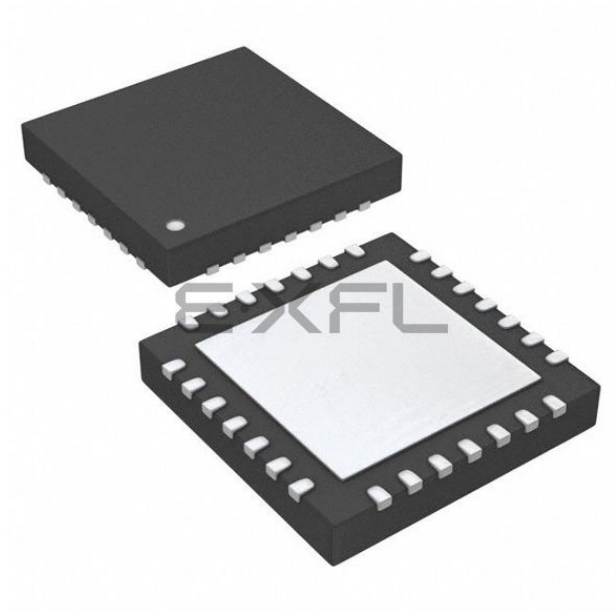


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

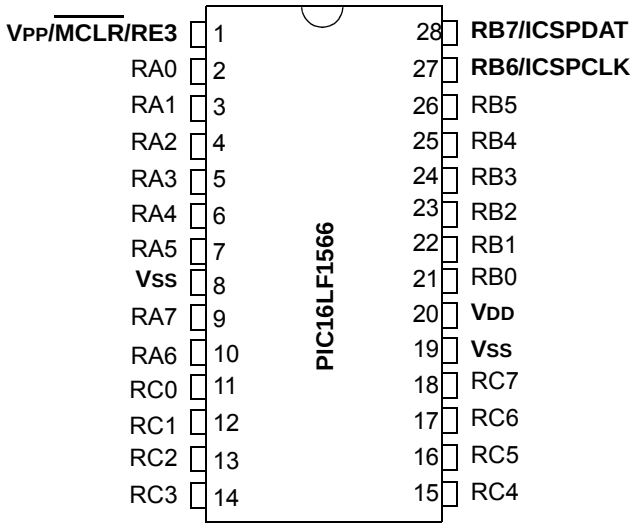
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	25
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 23x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UFQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1566t-i-mv

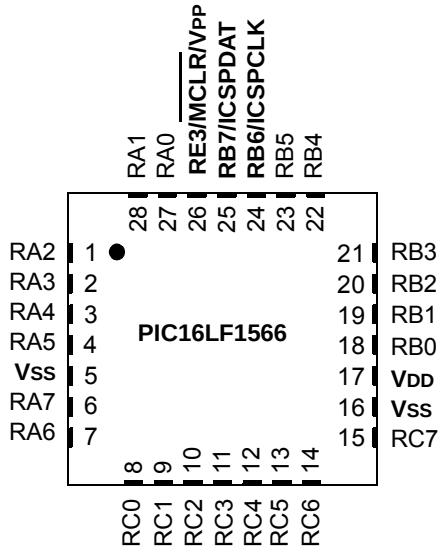
PIN DIAGRAMS

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.

PIN ALLOCATION TABLES

TABLE 2: 28-PIN ALLOCATION TABLE (PIC16LF1566)

I/O	28-Pin SPDIP/SOIC/SSOP	28-Pin UQFN	Analog Channel	ADC and CVD	Timers	PWM	EUSART	MSSP	Interrupt	Pull-up	Basic
RA0	2	27	AN20	—	—	PWM10	—	SS1(1)	—	—	—
RA1	3	28	AN10	—	—	PWM11	—	SS2	—	—	—
RA2	4	1	AN0	VREF-	—	PWM12	—	—	—	—	—
RA3	5	2	AN1	VREF+	—	PWM13	—	—	—	—	—
RA4	6	3	AN2	—	T0CKI	—	—	—	—	—	—
RA5	7	4	AN21	—	—	—	—	SS1(1)	—	—	—
RA6	10	7	AN22	ADTRIG	—	—	—	—	—	—	CLKOUT
RA7	9	6	AN11	—	—	—	—	—	—	—	CLKIN
RB0	21	18	AN16	—	—	PWM20	—	—	INT IOC	Y	—
RB1	22	19	AN27	—	—	PWM21	—	—	IOC	Y	—
RB2	23	20	AN17	—	—	PWM22	—	—	IOC	Y	—
RB3	24	21	AN28	—	—	PWM23	—	—	IOC	Y	—
RB4	25	22	AN18	AD1GRDA(1) AD2GRDA(1)	—	—	—	—	IOC	Y	—
RB5	26	23	AN29	AD1GRDA(1) AD2GRDA(1)	T1G	—	—	—	IOC	Y	—
RB6	27	24	AN19	AD1GRDB(1) AD2GRDB(1)	—	—	—	—	IOC	Y	ICSPCLK ICDCLK
RB7	28	25	AN40	AD1GRDB(1) AD2GRDB(1)	—	—	—	—	IOC	Y	ICSPDAT ICDDAT
RC0	11	8	AN12	—	T1CKI	—	—	SDO2	—	—	—
RC1	12	9	AN23	—	—	PWM2	—	SCL2 SCK2	—	—	—
RC2	13	10	AN13	—	—	PWM1	—	SDA2 SDI2	—	—	—
RC3	14	11	AN24	—	—	—	—	SCL1 SCK1	—	—	—
RC4	15	12	AN14	—	—	—	—	SDA1 SDI1	—	—	—
RC5	16	13	AN25	—	—	—	—	SDO1 I^2CLVL	—	—	—

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.

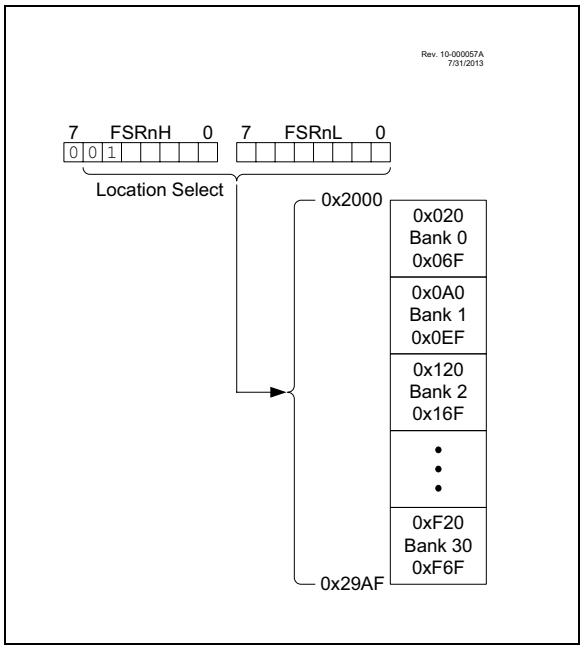
3.5.2 LINEAR DATA MEMORY

The linear data memory is the region from FSR address 0x2000 to FSR address 0x29AF. This region is a virtual region that points back to the 80-byte blocks of GPR memory in all the banks.

Unimplemented memory reads as 0x00. Use of the linear data memory region allows buffers to be larger than 80 bytes because incrementing the FSR beyond one bank will go directly to the GPR memory of the next bank.

The 16 bytes of common memory are not included in the linear data memory region.

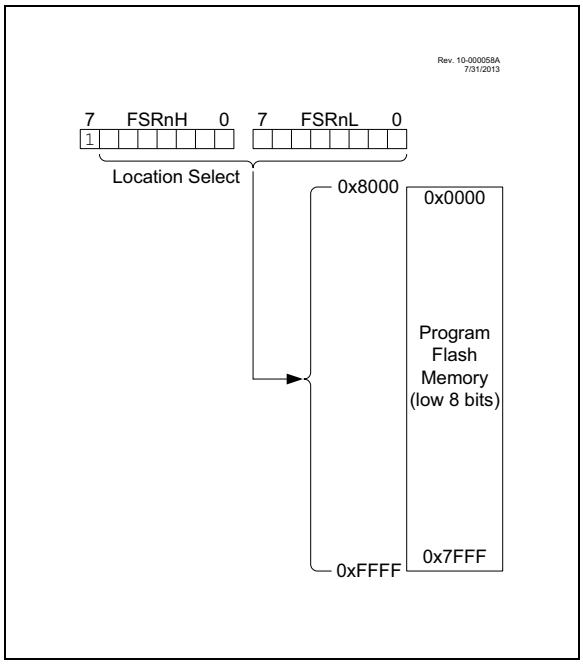
FIGURE 3-10: LINEAR DATA MEMORY MAP



3.5.3 PROGRAM FLASH MEMORY

To make constant data access easier, the entire program Flash memory is mapped to the upper half of the FSR address space. When the MSb of FSRnH is set, the lower 15 bits are the address in program memory which will be accessed through INDF. Only the lower eight bits of each memory location is accessible via INDF. Writing to the program Flash memory cannot be accomplished via the FSR/INDF interface. All instructions that access program Flash memory via the FSR/INDF interface will require one additional instruction cycle to complete.

FIGURE 3-11: PROGRAM FLASH MEMORY MAP



6.4 Low-Power Brown-out Reset (LPBOR)

The Low-Power Brown-out Reset (LPBOR) operates like the BOR to detect low voltage conditions on the VDD pin. When too low of a voltage is detected, the device is held in Reset. When this occurs, a register bit (BOR) is changed to indicate that a BOR Reset has occurred. The $\overline{BOR}$ bit in PCON is used for both BOR and the LPBOR. Refer to Register 6-2.

The LPBOR voltage threshold (VLPBOR) has a wider tolerance than the BOR (VBOR), but requires much less current (LPBOR current) to operate. The LPBOR is intended for use when the BOR is configured as disabled (BOREN = 00) or disabled in Sleep mode (BOREN = 10).

Refer to Figure 6-1 to see how the LPBOR interacts with other modules.

6.4.1 ENABLING LPBOR

The LPBOR is controlled by the $\overline{LPBOR}$ bit of Configuration Words. When the device is erased, the LPBOR module defaults to disabled.

6.5 $\overline{MCLR}$

The $\overline{MCLR}$ is an optional external input that can reset the device. The $\overline{MCLR}$ function is controlled by the MCLRE bit of Configuration Words and the LVP bit of Configuration Words (Table 6-2).

TABLE 6-2: $\overline{MCLR}$ CONFIGURATION

MCLRE	LVP	$\overline{MCLR}$
0	0	Disabled
1	0	Enabled
x	1	Enabled

6.5.1 $\overline{MCLR}$ ENABLED

When $\overline{MCLR}$ is enabled and the pin is held low, the device is held in Reset. The $\overline{MCLR}$ pin is connected to VDD through an internal weak pull-up.

The device has a noise filter in the $\overline{MCLR}$ Reset path. The filter will detect and ignore small pulses.

Note:	A Reset does not drive the $\overline{MCLR}$ pin low.

6.5.2 $\overline{MCLR}$ DISABLED

When $\overline{MCLR}$ is disabled, the pin functions as a general purpose input and the internal weak pull-up is under software control. See **Section 11.3 "PORTA Registers"** for more information.

6.6 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a CLRWDT instruction within the time-out period. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register are changed to indicate the WDT Reset. See **Section 9.0 "Watchdog Timer (WDT)"** for more information.

6.7 RESET Instruction

A RESET instruction will cause a device Reset. The $\overline{RI}$ bit in the PCON register will be set to '0'. See Table 6-4 for default conditions after a RESET instruction has occurred.

6.8 Stack Overflow/Underflow Reset

The device can reset when the Stack Overflows or Underflows. The STKOVF or STKUNF bits of the PCON register indicate the Reset condition. These Resets are enabled by setting the STVREN bit in Configuration Words. See **Section 3.4.2 "Overflow/Underflow Reset"** for more information.

6.9 Programming Mode Exit

Upon exit of Programming mode, the device will behave as if a POR had just occurred.

6.10 Power-up Timer

The Power-up Timer optionally delays device execution after a BOR or POR event. This timer is typically used to allow VDD to stabilize before allowing the device to start running.

The Power-up Timer is controlled by the $\overline{PWRTE}$ bit of Configuration Words.

6.11 Start-up Sequence

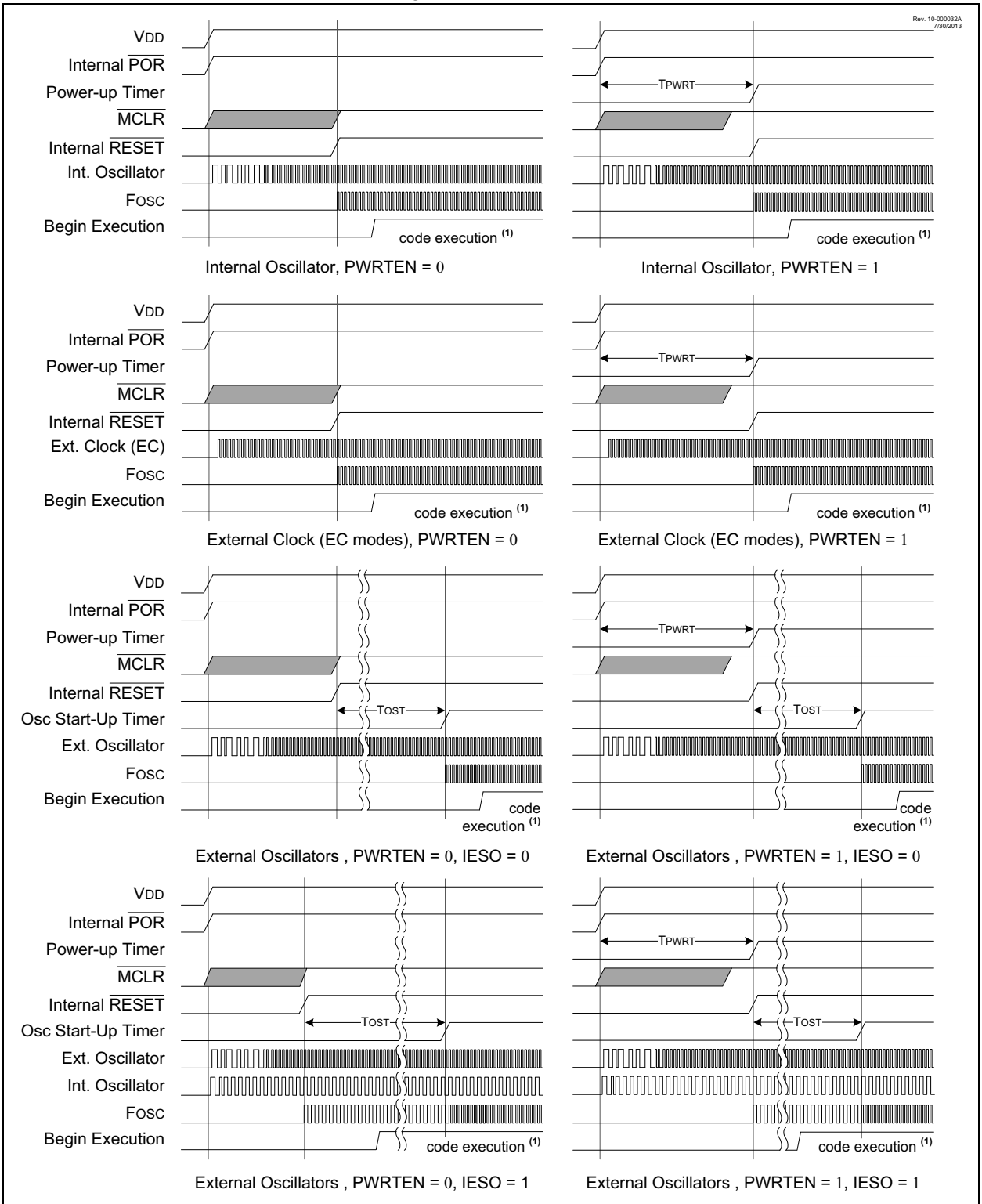
Upon the release of a POR or BOR, the following must occur before the device will begin executing:

1. Power-up Timer runs to completion (if enabled).
2. $\overline{MCLR}$ must be released (if enabled).

The total time-out will vary based on oscillator configuration and Power-up Timer configuration. See **Section 5.0 "Oscillator Module"** for more information.

The Power-up Timer runs independently of $\overline{MCLR}$ Reset. If $\overline{MCLR}$ is kept low long enough, the Power-up Timer will expire. Upon bringing $\overline{MCLR}$ high, the device will begin execution after 10 FOSC cycles (see Figure 6-3). This is useful for testing purposes or to synchronize more than one device operating in parallel.

FIGURE 6-3: RESET START-UP SEQUENCE



Note 1: Code execution begins 10 FOSC cycles after the FOSC clock is released.

8.0 POWER-DOWN MODE (SLEEP)

The Power-Down mode is entered by executing a SLEEP instruction.

Upon entering Sleep mode, the following conditions exist:

1. WDT will be cleared but keeps running, if enabled for operation during Sleep.
2. $\overline{PD}$ bit of the STATUS register is cleared.
3. $\overline{TO}$ bit of the STATUS register is set.
4. CPU clock is disabled.
5. 31 kHz LFINTOSC is unaffected and peripherals that operate from it may continue operation in Sleep.
6. Timer1 and peripherals that operate from Timer1 continue operation in Sleep when the Timer1 clock source selected is:
   - LFINTOSC
   - T1CKI
   - Timer1 oscillator
7. ADC is unaffected, if the dedicated FRC oscillator is selected.
8. I/O ports maintain the status they had before SLEEP was executed (driving high, low or high-impedance).
9. Resets other than WDT are not affected by Sleep mode.

Refer to individual chapters for more details on peripheral operation during Sleep.

To minimize current consumption, the following conditions should be considered:

- I/O pins should not be floating
- External circuitry sinking current from I/O pins
- Internal circuitry sourcing current from I/O pins
- Current draw from pins with internal weak pull-ups
- Modules using 31 kHz LFINTOSC

I/O pins that are high-impedance inputs should be pulled to V_{DD} or V_{SS} externally to avoid switching currents caused by floating inputs.

Examples of internal circuitry that might be sourcing current include the FVR module. See **Section 13.0 "Fixed Voltage Reference (FVR)"** for more information on this module.

8.1 Wake-up from Sleep

The device can wake-up from Sleep through one of the following events:

1. External Reset input on $\overline{MCLR}$ pin, if enabled
2. BOR Reset, if enabled
3. POR Reset
4. Watchdog Timer, if enabled
5. Any external interrupt
6. Interrupts by peripherals capable of running during Sleep (see individual peripheral for more information)

The first three events will cause a device Reset. The last three events are considered a continuation of program execution. To determine whether a device Reset or wake-up event occurred, refer to **Section 6.12 "Determining the Cause of a Reset"**.

When the SLEEP instruction is being executed, the next instruction (PC + 1) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be enabled. Wake-up will occur regardless of the state of the GIE bit. If the GIE bit is disabled, the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is enabled, the device executes the instruction after the SLEEP instruction, the device will then call the Interrupt Service Routine. In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

The WDT is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

TABLE 9-3: SUMMARY OF REGISTERS ASSOCIATED WITH WATCHDOG TIMER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		69
PCON	STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	77
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	25
WDTCON	—	—	WDTPS<4:0>					SWDTEN	93

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by Watchdog Timer.

TABLE 9-4: SUMMARY OF CONFIGURATION WORD WITH WATCHDOG TIMER

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG1	13:8	—	—	—	—	$\overline{\text{CLKOUTEN}}$	BOREN<1:0>		—	59
	7:0	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE<1:0>		—	FOSC<1:0>		

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

REGISTER 11-21: LATE(1): PORTE DATA LATCH REGISTER

U-1	U-1	U-1	U-1	U-1	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	—	LATE2	LATE1	LATE0
bit 7							bit 0

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

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

bit 2-0 **LATE<2:0>**: PORTE Output Latch Value bits(2)

Note 1: Functions available on PIC16LF1567 only.

REGISTER 11-22: ANSELE(1): PORTE ANALOG SELECT REGISTER

U-1	U-1	U-1	U-1	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	—	ANSE2	ANSE1	ANSE0
bit 7							bit 0

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

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

bit 2-0 **ANSE<2:0>**: Analog Select between Analog or Digital Function on pins RE<7:0>, respectively
1 = Analog input. Pin is assigned as analog input(2). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

Note 1: Functions available on PIC16LF1567 only.
2: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

TABLE 11-13: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELE(1)	—	—	—	—	—	ANSE2	ANSE1	ANSE0	130
LATE(1)	—	—	—	—	—	LATE2	LATE1	LATE0	130
PORTE	—	—	—	—	RE3	RE2	RE1	RE0	129
TRISE(1)	—	—	—	—	—(2)	TRISE2	TRISE1	TRISE0	129

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by PORTE.

Note 1: Functions available on PIC16LF1567 only.
2: Unimplemented, read as '1'.

Each ADC can enable the TX output on any or all of its associated analog channels using the ADxTX0 and ADxTX1 registers. The shared analog channels have TX enable bits in the ADCTX register. Once enabled and while ADxON = 1, the TX outputs of the ADC are active at all times except if the ADC is currently selecting the channel for conversion with the CHS bits of ADxCON0.

Polarity of the output is controlled by the TXxPOL bit of the AADxGRD register. The outputs are initialized at the start of the precharge stage to match the polarity of the TXxPOL bit. The TX output signal changes polarity immediately after the start of the acquisition phase. The value stored by TXxPOL does not change. When in double sampling mode (ADxDSEN = 1), the TX output changes polarity during the second precharge and acquisition phases if inversion is enabled (ADxIPEN = 1). For more information about the timing of the TX output, refer to Figure 16-4.

The typical mutual TX trace does not have a series resistor. If radiated emissions are a concern, a series resistor can be used to increase the rise-time at the cost of reduced noise dissipation.

To perform a combined self- and mutual-capacitance measurement, set ADxEPPOL and ADxIPPOL to opposite polarities, and set TXxPOL = ADxEPPOL.

To perform a mutual-only capacitance measurement, set ADxEEPOL and ADxIPPOL to the same polarity, and set TXxPOL = ADxEPPOL.

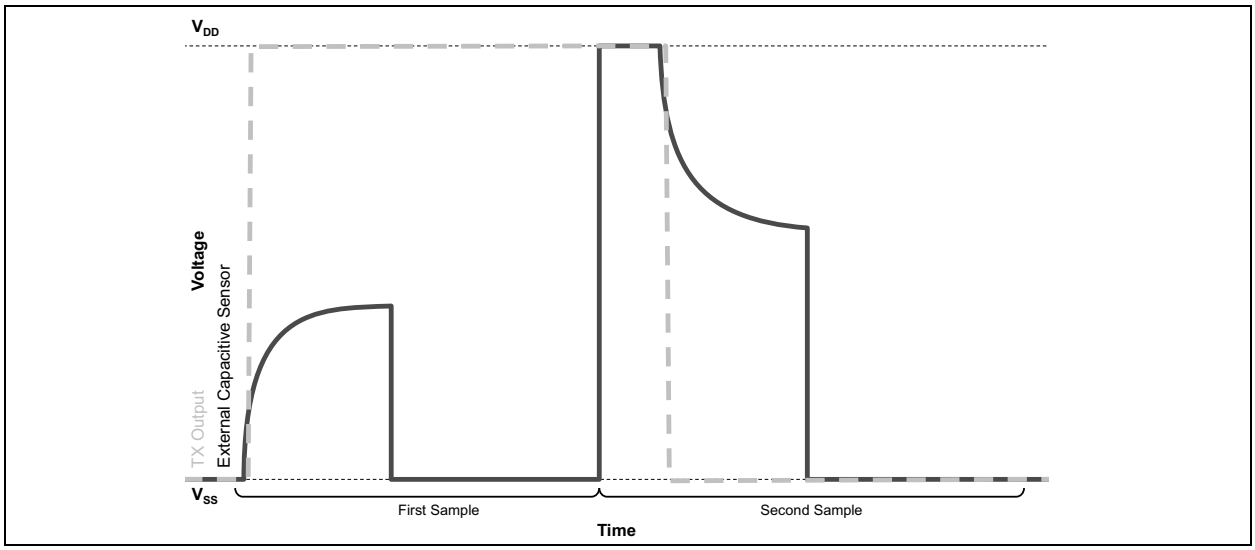
16.1.9 COMPARISON OF GUARDING AND MUTUAL CAPACITANCE

The mutual TX drivers are driven the same way as the ADxGRDA output.

Both guard and mutual drivers provide a low impedance path for noise to redirect away from the sensor to improve robustness. Mutual drivers are lower impedance due to the absence of the external voltage divider resistance.

The goal of the guard is to minimize coupling between the sensor (RX) and the environment to improve sensitivity, while the goal of the mutual TX driver is to maximize the change in coupling when the event occurs.

FIGURE 16-4: DIFFERENTIAL CVD WITH GUARD RING OUTPUT WAVEFORM



16.1.10 ADDITIONAL SAMPLE AND HOLD CAPACITOR

Additional capacitance can be added in parallel with the sample and hold capacitor (CHOLD) by setting the ADDxCAP<3:0> bits of the AADxCAP register. This bit connects a digitally programmable capacitance to the ADC conversion bus, increasing the effective internal capacitance of the sample and hold capacitor in the ADC module. Each ADC has its own additional capacitance array. This is used to improve the match between internal and external capacitance for a better sensing performance. The additional capacitance does not affect analog performance of the ADC because it is not connected during conversion. See Figure 16-1.

16.1.11 SECONDARY CHANNEL

Each ADC has one primary channel selected by CHx<4:0> bits of the AADxCON0 register. Multiple secondary channels can be connected to the ADC conversion bus by setting the bits in the AADxCH register. This allows a combined CVD scan on multiple ADC channels, which is beneficial for low-power and proximity capacitive sensing.

REGISTER 16-2: AD2CON0: ANALOG-TO-DIGITAL (ADC) 2 CONTROL REGISTER 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	R/W-0/0
CHS25	CHS24	CHS23	CHS22	CHS21	CHS20	GO/$\overline{\text{DONE2}}$(2)	AD2ON
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	

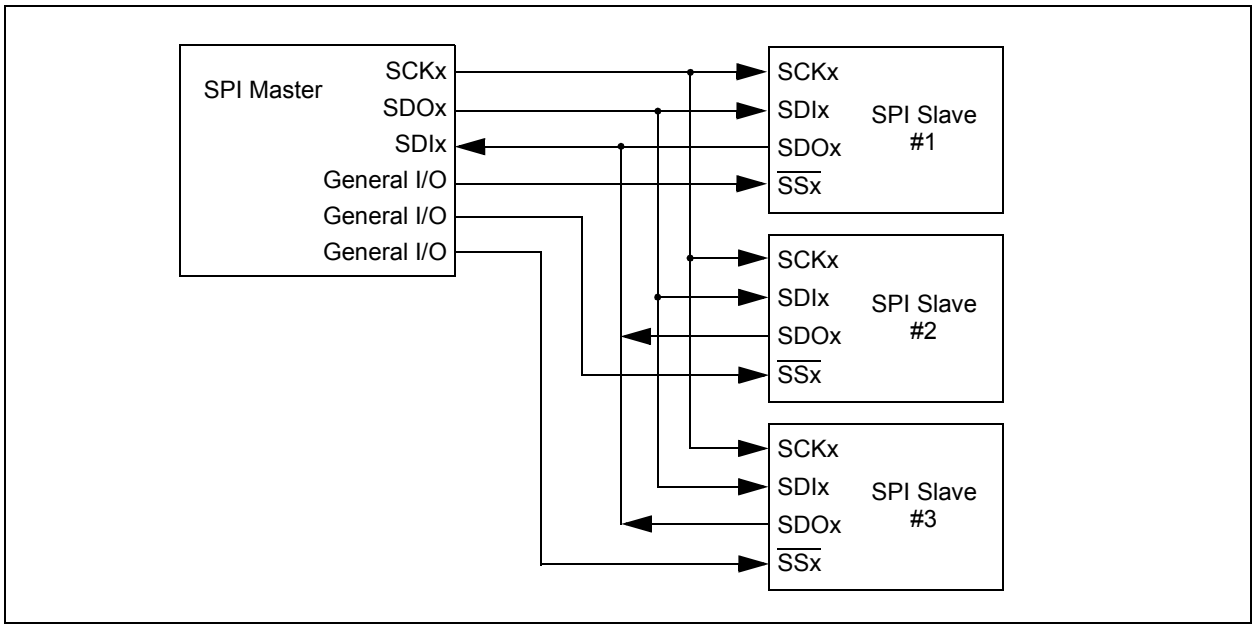
bit 7-2 **CHS25<5:0>:** Analog Channel Select bits for ADC2
When AD2ON = 0, all multiplexer inputs are disconnected.
111111 = Fixed Voltage Reference (FVREF)
111101 = Temperature Indicator
111011 = VREFH (ADC Positive Reference)
101110 - 111010 = Reserved
101001 - 101101 = Channel 41 through 45, (AN41 through AN45)(1)
101000 = Channel 40, (AN40)
011110 - 100111 = Reserved
010100 - 011101 = Channel 20 through 29, (AN20 through AN29)
000011 - 010011 = Reserved
000010 = Channel 2, (AN2)
000001 = Channel 1, (AN1)
000000 = Channel 0, (AN0)

bit 1 **GO/$\overline{\text{DONE2}}$:** ADC2 Conversion Status bit(2)
If AD2ON = 1
1 = ADC conversion in progress. Setting this bit starts the ADC conversion. When the RC clock source is selected, the ADC Module waits one instruction before starting the conversion.
0 = ADC conversion not in progress (This bit is automatically cleared by hardware when the ADC conversion is complete.)
If this bit is cleared while a conversion is in progress, the conversion will stop and the results of the conversion up to this point will be transferred to the result registers, but the AD2IF interrupt flag bit will not be set.
If AD2ON = 0
0 = ADC conversion not in progress

bit 0 **AD2ON:** ADC Module 2 Enable bit
1 = ADC converter module 2 is operating
0 = ADC converter module 2 is shut off and consumes no operating current. All Analog channels are disconnected.

Note 1: PIC16LF1567 only. Not implemented on PIC16LF1566.
2: When the AD2DSEN bit is set; the GO/$\overline{\text{DONE}}$ bit will clear after a second conversion has completed.

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

20.2.4 SPI SLAVE MODE

In Slave mode, the data is transmitted and received as external clock pulses appear on SCKx. When the last bit is latched, the SSPxIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCKx pin. The Idle state is determined by the CKP bit of the SSPxCON1 register.

While in Slave mode, the external clock is supplied by the external clock source on the SCKx pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. The shift register is clocked from the SCKx pin input and when a byte is received, the device will generate an interrupt. If enabled, the device will wake-up from Sleep.

20.2.4.1 Daisy-Chain Configuration

The SPI bus can sometimes be connected in a daisy-chain configuration. The first slave output is connected to the second slave input, the second slave output is connected to the third slave input, and so on. The final slave output is connected to the master input. Each slave sends out, during a second group of clock pulses, an exact copy of what was received during the first group of clock pulses. The whole chain acts as one large communication shift register. The daisy-chain feature only requires a single Slave Select line from the master device.

Figure 20-7 shows the block diagram of a typical daisy-chain connection when operating in SPI mode.

In a daisy-chain configuration, only the most recent byte on the bus is required by the slave. Setting the BOEN bit of the SSPxCON3 register will enable writes to the SSPxBUF register, even if the previous byte has not been read. This allows the software to ignore data that may not apply to it.

20.2.5 SLAVE SELECT SYNCHRONIZATION

The Slave Select can also be used to synchronize communication. The Slave Select line is held high until the master device is ready to communicate. When the Slave Select line is pulled low, the slave knows that a new transmission is starting.

If the slave fails to receive the communication properly, it will be reset at the end of the transmission, when the Slave Select line returns to a high state. The slave is then ready to receive a new transmission when the Slave Select line is pulled low again. If the Slave Select line is not used, there is a risk that the slave will eventually become out of sync with the master. If the slave misses a bit, it will always be one bit off in future transmissions. Use of the Slave Select line allows the slave and master to align themselves at the beginning of each transmission.

The $\overline{SSx}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SSx}$ pin control enabled (SSPxCON1<3:0> = 0100).

When the $\overline{SSx}$ pin is low, transmission and reception are enabled and the SDOx pin is driven.

When the $\overline{SSx}$ pin goes high, the SDOx pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

Note 1:	When the SPI is in Slave mode with $\overline{SSx}$ pin control enabled (SSPxCON1<3:0> = 0100), the SPI module will reset if the $\overline{SSx}$ pin is set to VDD.
2:	When the SPI is used in Slave mode with CKE set; the user must enable $\overline{SSx}$ pin control.
3:	While operated in SPI Slave mode the SMP bit of the SSPxSTAT register must remain clear.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SSx}$ pin to a high level or clearing the SSPEN bit.

20.7 BAUD RATE GENERATOR

The MSSPx module has a Baud Rate Generator available for clock generation in both I^2C and SPI Master modes. The Baud Rate Generator (BRG) reload value is placed in the SSPxADD register (Register 20-6). When a write occurs to SSPxBUF, the Baud Rate Generator will automatically begin counting down.

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

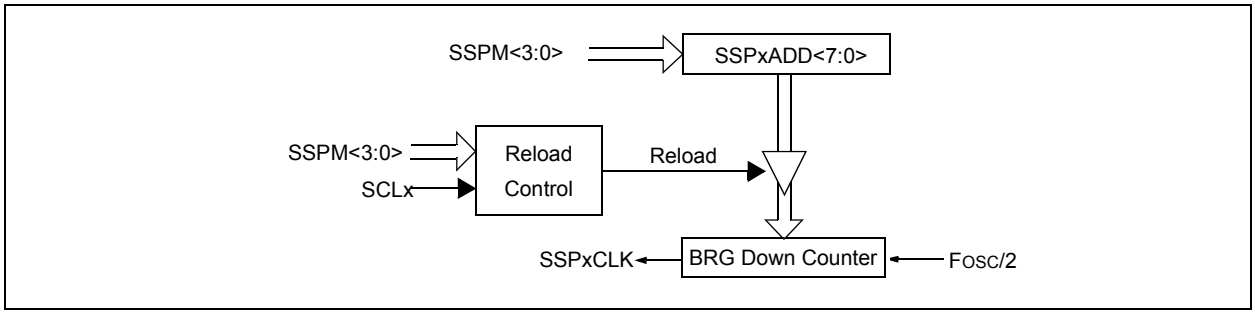
An internal signal “Reload” in Figure 20-40 triggers the value from SSPxADD to be loaded into the BRG counter. This occurs twice for each oscillation of the module clock line. The logic dictating when the reload signal is asserted depends on the mode the MSSPx is being operated in.

Table 20-4 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPxADD.

EQUATION 20-1:

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

FIGURE 20-40: BAUD RATE GENERATOR BLOCK DIAGRAM



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

TABLE 20-4: MSSPX CLOCK RATE W/BRG

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

Note 1: Refer to I/O port electrical and timing specifications in Table 25-9 and Figure 25-5 to ensure the system is designed to support the I/O timing requirements.

REGISTER 20-5: SSPxMSK: SSPx MASK REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
MSK<7:0>							
bit 7							bit 0

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

bit 7-1 **MSK<7:1>:** Mask bits
1 = The received address bit n is compared to SSPxADD<n> to detect I^2C address match
0 = The received address bit n is not used to detect I^2C address match

bit 0 **MSK<0>:** Mask bit for I^2C Slave mode, 10-bit Address
I^2C Slave mode, 10-bit address (SSPM<3:0> = 0111 or 1111):
1 = The received address bit 0 is compared to SSPxADD<0> to detect I^2C address match
0 = The received address bit 0 is not used to detect I^2C address match
I^2C Slave mode, 7-bit address, the bit is ignored

21.5.2.3 EUSART Synchronous Slave Reception

The operation of the Synchronous Master and Slave modes is identical (**Section 21.5.1.5 "Synchronous Master Reception"**), with the following exceptions:

- Sleep
- CREN bit is always set, therefore the receiver is never idle
- SREN bit, which is a "don't care" in Slave mode

A character may be received while in Sleep mode by setting the CREN bit prior to entering Sleep. Once the word is received, the RSR register will transfer the data to the RCREG register. If the RCIE enable bit is set, the interrupt generated will wake the device from Sleep and execute the next instruction. If the GIE bit is also set, the program will branch to the interrupt vector.

21.5.2.4 Synchronous Slave Reception Set-up:

1. Set the SYNC and SPEN bits and clear the CSRC bit.
2. Clear the ANSELx bit for both the CK and DT pins (if applicable).
3. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
4. If 9-bit reception is desired, set the RX9 bit.
5. Set the CREN bit to enable reception.
6. The RCIF bit will be set when reception is complete. An interrupt will be generated if the RCIE bit was set.
7. If 9-bit mode is enabled, retrieve the Most Significant bit from the RX9D bit of the RCSTA register.
8. Retrieve the eight Least Significant bits from the receive FIFO by reading the RCREG register.
9. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTA register or by clearing the SPEN bit which resets the EUSART.

TABLE 21-10: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	266
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	84
PIE1	TMR1GIE	AD1IE	RCIE	TXIE	SSP1IE	SSP2IE	TMR2IE	TMR1IE	85
PIR1	TMR1GIF	AD1IF	RCIF	TXIF	SSP1IF	SSP2IF	TMR2IF	TMR1IF	87
RCREG	EUSART Receive Data Register								260*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	265
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	119
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	264

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous slave reception.

* Page provides register information.

24.0 INSTRUCTION SET SUMMARY

Each instruction is a 14-bit word containing the operation code (opcode) and all required operands. The opcodes are broken into three broad categories.

- Byte Oriented
- Bit Oriented
- Literal and Control

The literal and control category contains the most varied instruction word format.

Table lists the instructions recognized by the MPASM™ assembler.

All instructions are executed within a single instruction cycle, with the following exceptions, which may take two or three cycles:

- Subroutine takes two cycles (CALL, CALLW)
- Returns from interrupts or subroutines take two cycles (RETURN, RETLW, RETFIE)
- Program branching takes two cycles (GOTO, BRA, BRW, BTFSS, BTFSC, DECFSZ, INCSFZ)
- One additional instruction cycle will be used when any instruction references an indirect file register and the file select register is pointing to program memory.

One instruction cycle consists of 4 oscillator cycles; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution rate of 1 MHz.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

24.1 Read-Modify-Write Operations

Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

TABLE 24-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
n	FSR or INDF number (0-1)
mm	Pre-post increment-decrement mode selection

TABLE 24-2: ABBREVIATION DESCRIPTIONS

Field	Description
PC	Program Counter
$\overline{TO}$	Time-Out bit
C	Carry bit
DC	Digit Carry bit
Z	Zero bit
$\overline{PD}$	Power-Down bit

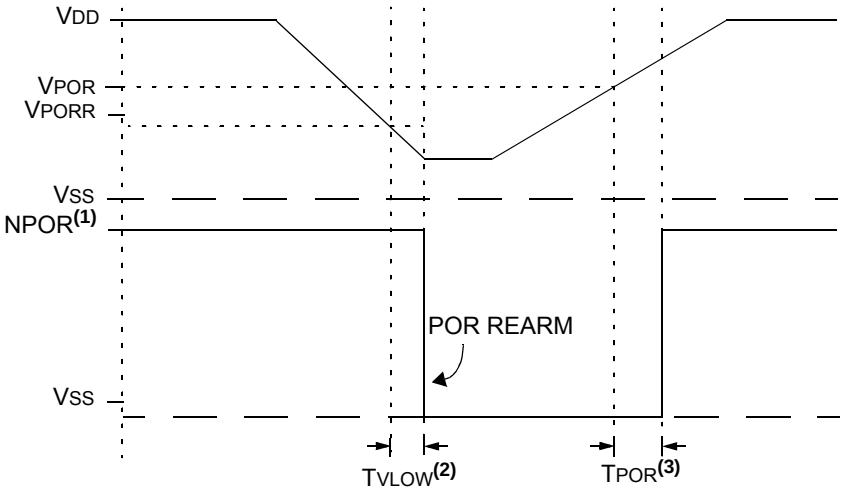
SWAPF	Swap Nibbles in f
Syntax:	[*label*] SWAPF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f<3:0>) → (destination<7:4>), (f<7:4>) → (destination<3:0>)
Status Affected:	None
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'.

TRIS	Load TRIS Register with W
Syntax:	[*label*] TRIS f
Operands:	$5 \le f \le 7$
Operation:	(W) → TRIS register 'f'
Status Affected:	None
Description:	Move data from W register to TRIS register. When 'f' = 5, TRISA is loaded. When 'f' = 6, TRISB is loaded. When 'f' = 7, TRISC is loaded.

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

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

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.

TABLE 25-10: RESET, WATCHDOG TIMER, POWER-UP TIMER AND BROWN-OUT RESET PARAMETERS

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ TA ≤ +125°C							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
30	TMCL	$\overline{MCLR}$ Pulse Width (low)	2 5	— —	— —	µs µs	-40°C to +85°C +85°C to +125°C
31	TWDTLP	Low-Power Watchdog Timer Time-out Period	10	16	27	ms	VDD = 3.3V-3.6V, 1:512 Prescaler used
33*	TPWRT	Power-up Timer Period, $\overline{PWRTE}$ = 0	40	65	140	ms	
34*	TIOZ	I/O high-impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset	—	—	2.0	µs	
35	VBOR	Brown-out Reset Voltage(1)	2.55 1.80	2.70 1.90	2.85 2.05	V V	BORV = 0 BORV = 1
35A	VLPBOR	Low-Power Brown-out	1.8	2.1	2.5	V	LPBOR = 1
36*	VHYST	Brown-out Reset Hysteresis	0	25	50	mV	-40°C to +85°C
37*	TBORDC	Brown-out Reset DC Response Time	1	3	5	µs	VDD ≤ VBOR

* 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.

Note 1: To ensure these voltage tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 µF and 0.01 µF values in parallel are recommended.

FIGURE 25-8: TIMER0 EXTERNAL CLOCK TIMINGS