



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	20MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	25
Program Memory Size	28KB (16K 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 17x10b
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/pic16lf1518t-i-mv

PIC16(L)F1516/7/8/9

Table of Contents

1.0	Device Overview	10
2.0	Enhanced Mid-range CPU	15
3.0	Memory Organization	17
4.0	Device Configuration	41
5.0	Oscillator Module (with Fail-Safe Clock Monitor)	47
6.0	Resets	62
7.0	Interrupts	70
8.0	Power-Down Mode (Sleep)	80
9.0	Low Dropout (LDO) Voltage Regulator	84
10.0	Watchdog Timer (WDT)	85
11.0	Flash Program Memory Control	89
12.0	I/O Ports	105
13.0	Interrupt-on-Change	124
14.0	Fixed Voltage Reference (FVR)	128
15.0	Temperature Indicator Module	130
16.0	Analog-to-Digital Converter (ADC) Module	132
17.0	Timer0 Module	145
18.0	Timer1 Module with Gate Control.....	148
19.0	Timer2 Module	159
20.0	Capture/Compare/PWM Modules	163
21.0	Master Synchronous Serial Port (MSSP) Module	171
22.0	Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART).....	222
23.0	In-Circuit Serial Programming™ (ICSP™)	251
24.0	Instruction Set Summary	253
25.0	Electrical Specifications.....	267
26.0	DC and AC Characteristics Graphs and Charts	298
27.0	Development Support.....	330
28.0	Packaging Information.....	334
	Appendix A: Data Sheet Revision History.....	355
	The Microchip Website.....	356
	Customer Change Notification Service	356
	Customer Support.....	356
	Product Identification System.....	358

PIC16(L)F1516/7/8/9

FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1516/7

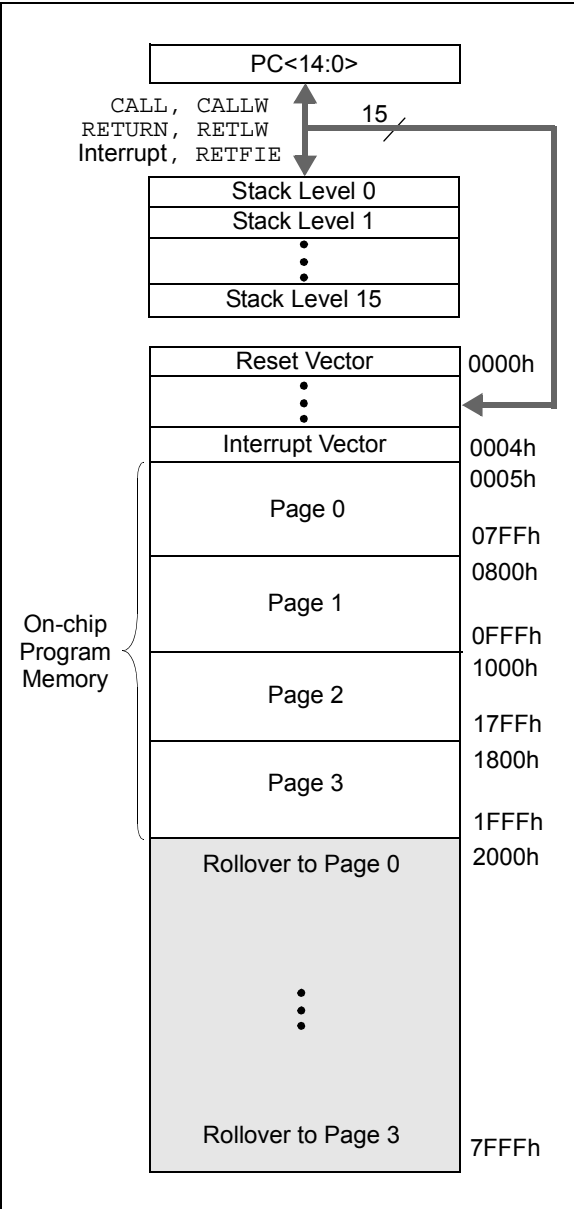
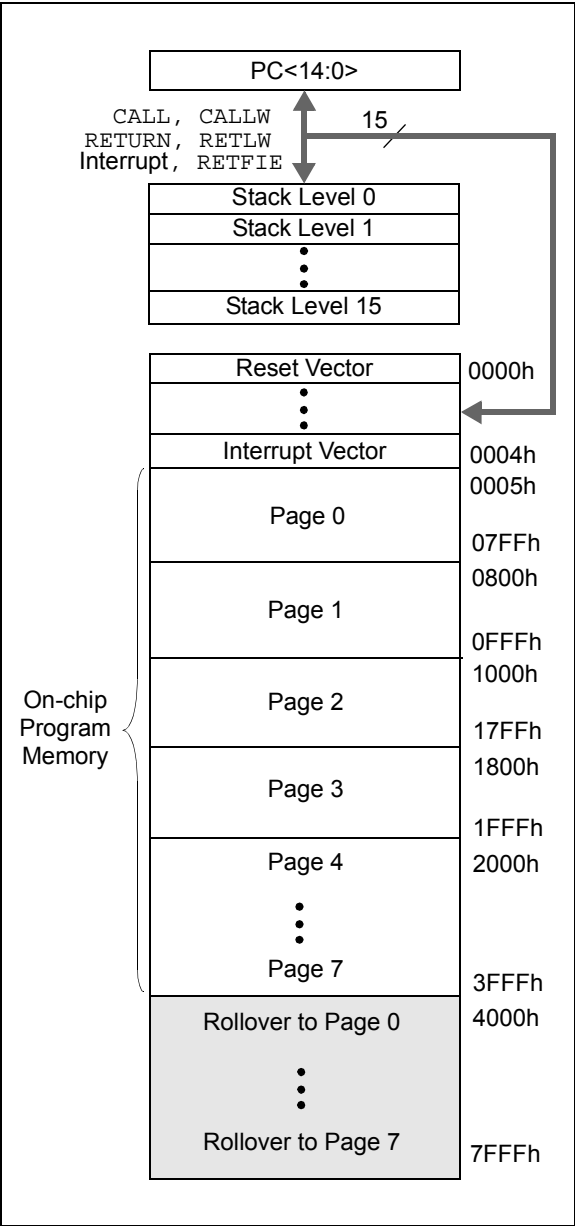


FIGURE 3-2: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1518/9



REGISTER 4-2: CONFIG2: CONFIGURATION WORD 2

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-1
LVP	DEBUG	LPBOR	BORV	STVREN	—
bit 13					bit 8

U-1	U-1	U-1	R/P-1	U-1	U-1	R/P-1	R/P-1
—	—	—	VCAPEN ⁽¹⁾	—	—	WRT<1:0>	
bit 7							bit 0

Legend:

R = Readable bit

P = Programmable bit

U = Unimplemented bit, read as '1'

'0' = Bit is cleared

'1' = Bit is set

-n = Value when blank or after Bulk Erase

- bit 13 **LVP:** Low-Voltage Programming Enable bit
1 = Low-voltage programming enabled
0 = High-voltage on MCLR must be used for programming
- bit 12 **DEBUG:** In-Circuit Debugger Mode bit
1 = In-Circuit Debugger disabled, ICSPCLK and ICSPDAT are general purpose I/O pins
0 = In-Circuit Debugger enabled, ICSPCLK and ICSPDAT are dedicated to the debugger
- bit 11 **LPBOR:** Low-Power BOR
1 = Low-Power BOR is disabled
0 = Low-Power BOR is enabled
- bit 10 **BORV:** Brown-out Reset Voltage Selection bit⁽²⁾
1 = Brown-out Reset voltage (Vbor), low trip point selected.
0 = Brown-out Reset voltage (Vbor), high trip point selected.
- bit 9 **STVREN:** Stack Overflow/Underflow Reset Enable bit
1 = Stack Overflow or Underflow will cause a Reset
0 = Stack Overflow or Underflow will not cause a Reset
- bit 8-5 **Unimplemented:** Read as '1'
- bit 4 **VCAPEN:** Voltage Regulator Capacitor Enable bits⁽¹⁾
If PIC16LF1516/7/8/9 (regulator disabled):
These bits are ignored. All VCAP pin functions are disabled.
If PIC16F1516/7/8/9 (regulator enabled):
0 = VCAP functionality is enabled on RA5
1 = All VCAP pin functions are disabled
- bit 3-2 **Unimplemented:** Read as '1'
- bit 1-0 **WRT<1:0>:** Flash Memory Self-Write Protection bits
8 kW Flash memory (PIC16(L)F1516/7 only):
11 = Write protection off
10 = 000h to 1FFh write-protected, 200h to 1FFFh may be modified by PMCON control
01 = 000h to FFFh write-protected, 1000h to 1FFFh may be modified by PMCON control
00 = 000h to 1FFFh write-protected, no addresses may be modified by PMCON control
16 kW Flash memory (PIC16(L)F1518/9 only):
11 = Write protection off
10 = 000h to 1FFh write-protected, 200h to 3FFFh may be modified by PMCON control
01 = 000h to 1FFFh write-protected, 2000h to 3FFFh may be modified by PMCON control
00 = 000h to 3FFFh write-protected, no addresses may be modified by PMCON control

Note 1: PIC16F1516/7/8/9 only.

Note 2: See Vbor parameter for specific trip point voltages.

5.2.2.3 Internal Oscillator Frequency Selection

The system clock speed can be selected via software using the Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register.

The outputs of the 16 MHz HFINTOSC postscaler and the LFINTOSC connects to a multiplexer (see Figure 5-1). The Internal Oscillator Frequency Select bits IRCF<3:0> of the OSCCON register select the frequency output of the internal oscillators. One of the following frequencies can be selected via software:

- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz
- 500 kHz (default after Reset)
- 250 kHz
- 125 kHz
- 62.5 kHz
- 31.25 kHz
- 31 kHz (LFINTOSC)

Note: Following any Reset, the IRCF<3:0> bits of the OSCCON register are set to '0111' and the frequency selection is set to 500 kHz. The user can modify the IRCF bits to select a different frequency.

The IRCF<3:0> bits of the OSCCON register allow duplicate selections for some frequencies. These duplicate choices can offer system design trade-offs. Lower power consumption can be obtained when changing oscillator sources for a given frequency. Faster transition times can be obtained between frequency changes that use the same oscillator source.

5.2.2.4 Internal Oscillator Clock Switch Timing

When switching between the HFINTOSC and the LFINTOSC, the new oscillator may already be shut down to save power (see Figure 5-7). If this is the case, there is a delay after the IRCF<3:0> bits of the OSCCON register are modified before the frequency selection takes place. The OSCSTAT register will reflect the current active status of the HFINTOSC and LFINTOSC oscillators. The sequence of a frequency selection is as follows:

1. IRCF<3:0> bits of the OSCCON register are modified.
2. If the new clock is shut down, a clock start-up delay is started.
3. Clock switch circuitry waits for a falling edge of the current clock.
4. The current clock is held low and the clock switch circuitry waits for a rising edge in the new clock.
5. The new clock is now active.
6. The OSCSTAT register is updated as required.
7. Clock switch is complete.

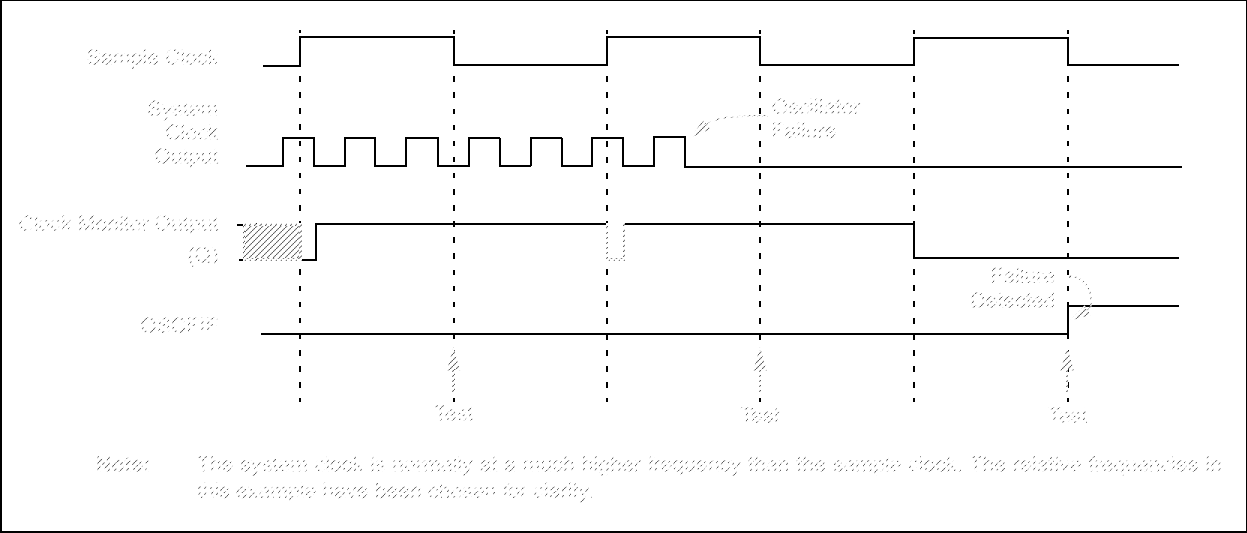
See Figure 5-7 for more details.

If the internal oscillator speed is switched between two clocks of the same source, there is no start-up delay before the new frequency is selected. Clock switching time delays are shown in Table 5-1.

Start-up delay specifications are located in the oscillator tables of **Section 25.0 “Electrical Specifications”**.

PIC16(L)F1516/7/8/9

FIGURE 5-10: FSCM TIMING DIAGRAM



9.0 LOW DROPOUT (LDO) VOLTAGE REGULATOR

The PIC16F1516/7/8/9 has an internal Low Dropout Regulator (LDO) which provides operation above 3.6V. The LDO regulates a voltage for the internal device logic while permitting the VDD and I/O pins to operate at a higher voltage. There is no user enable/disable control available for the LDO, it is always active. The PIC16LF1516/7/8/9 operates at a maximum VDD of 3.6V and does not incorporate an LDO.

A device I/O pin may be configured as the LDO voltage output, identified as the VCAP pin. Although not required, an external low-ESR capacitor may be connected to the VCAP pin for additional regulator stability.

The $\overline{\text{VCAPEN}}$ bit of Configuration Words enables or disables the VCAP pin. Refer to Table 9-1.

On power-up, the external capacitor will load the LDO voltage regulator. To prevent erroneous operation, the device is held in Reset while a constant current source charges the external capacitor. After the cap is fully charged, the device is released from Reset. For more information on the constant current rate, refer to the LDO Regulator Characteristics Table in **Section 25.0 “Electrical Specifications”**.

TABLE 9-1: $\overline{\text{VCAPEN}}$ SELECT BIT

$\overline{\text{VCAPEN}}$	Pin
0	RA5

TABLE 9-2: SUMMARY OF CONFIGURATION WORD WITH LDO

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
CONFIG2	13:8			LVP	$\overline{\text{DEBUG}}$	$\overline{\text{LPBOR}}$	BORV	STVREN	—	43
	7:0	—	—	—	$\overline{\text{VCAPEN}}$	—	—	WRT<1:0>		

Legend: — = unimplemented locations read as '0'. Shaded cells are not used by LDO.

Note 1: PIC16F1516/7/8/9 only.

EXAMPLE 11-2: ERASING ONE ROW OF PROGRAM MEMORY

```

; This row erase routine assumes the following:
; 1. A valid address within the erase row is loaded in ADDRHL:ADDRL
; 2. ADDRHL and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)

      BCF      INTCON,GIE      ; Disable ints so required sequences will execute properly
      BANKSEL PMADRL
      MOVF     ADDRL,W         ; Load lower 8 bits of erase address boundary
      MOVWF    PMADRL
      MOVF     ADDRHL,W        ; Load upper 6 bits of erase address boundary
      MOVWF    PMADRH
      BCF      PMCON1,CFGSR    ; Not configuration space
      BSF      PMCON1,FREER    ; Specify an erase operation
      BSF      PMCON1,WREN     ; Enable writes

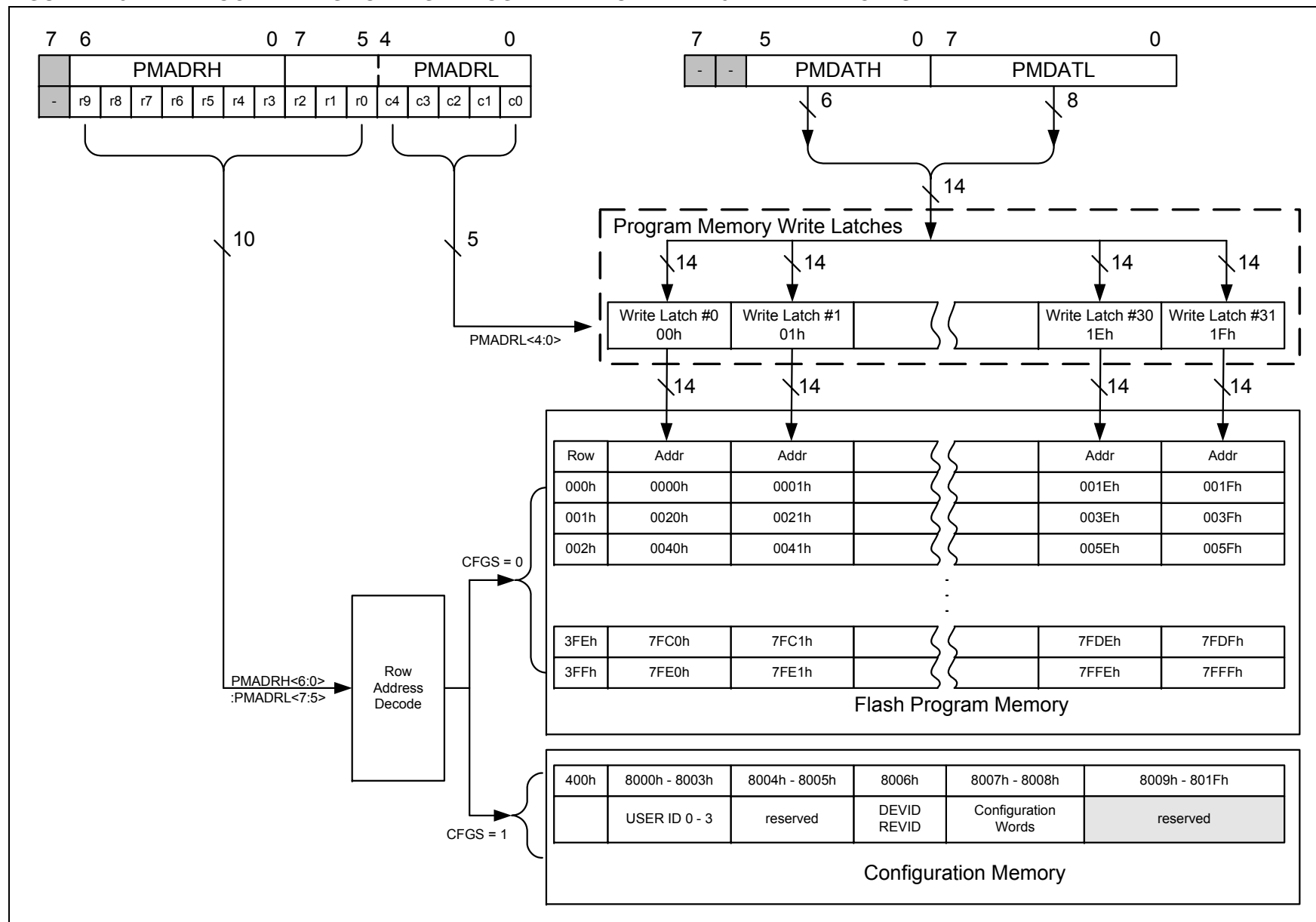
      MOVLW    55h             ; Start of required sequence to initiate erase
      MOVWF    PMCON2          ; Write 55h
      MOVLW    0AAh            ;
      MOVWF    PMCON2          ; Write AAh
      BSF      PMCON1,WR       ; Set WR bit to begin erase
      NOP      ; NOP instructions are forced as processor starts
      NOP      ; row erase of program memory.
      ;
      ; The processor stalls until the erase process is complete
      ; after erase processor continues with 3rd instruction

      BCF      PMCON1,WREN     ; Disable writes
      BSF      INTCON,GIE     ; Enable interrupts

```

Required
Sequence

FIGURE 11-5: BLOCK WRITES TO FLASH PROGRAM MEMORY WITH 32 WRITE LATCHES



PIC16(L)F1516/7/8/9

REGISTER 12-9: ANSELB: PORTB ANALOG SELECT REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
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 7-6 **Unimplemented:** Read as '0'

bit 5-0 **ANSB<5:0>:** Analog Select between Analog or Digital Function on pins RB<5:0>, respectively
0 = Digital I/O. Pin is assigned to port or digital special function.
1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital input buffer disabled.

Note 1: 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.

REGISTER 12-10: WPUB: WEAK PULL-UP PORTB 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
WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0
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 **WPUB<7:0>:** Weak Pull-up Register bits
1 = Pull-up enabled
0 = Pull-up disabled

Note 1: Global WPUEN bit of the OPTION_REG register must be cleared for individual pull-ups to be enabled.
2: The weak pull-up device is automatically disabled if the pin is in configured as an output.

TABLE 12-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	112
APFCON	—	—	—	—	—	—	SSSEL	CCP2SEL	105
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	111
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			146
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	111
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	111
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	112

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

PIC16(L)F1516/7/8/9

16.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADC Interrupt Flag is the ADIF bit in the PIR1 register. The ADC Interrupt Enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

- Note 1:** The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.

2: The ADC operates during Sleep only when the FRC oscillator is selected.

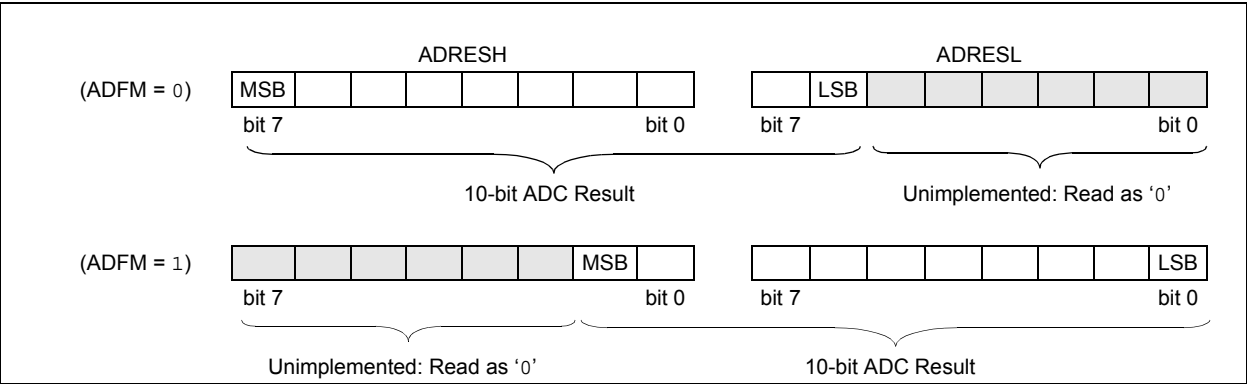
This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the `SLEEP` instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the GIE and PEIE bits of the INTCON register must be disabled. If the GIE and PEIE bits of the INTCON register are enabled, execution will switch to the Interrupt Service Routine.

16.1.6 RESULT FORMATTING

The 10-bit ADC conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON1 register controls the output format.

Figure shows the two output formats.

FIGURE 16-3: 10-BIT ADC CONVERSION RESULT FORMAT



PIC16(L)F1516/7/8/9

20.0 CAPTURE/COMPARE/PWM MODULES

The Capture/Compare/PWM module is a peripheral which allows the user to time and control different events, and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate Pulse-Width Modulated signals of varying frequency and duty cycle.

This family of devices contains two standard Capture/Compare/PWM modules (CCP1 and CCP2).

Note 1: In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.

2: Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to CCPx module. Register names, module signals, I/O pins, and bit names may use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module, when required.

TABLE 20-1: PWM RESOURCES

Device Name	CCP1	CCP2
PIC16(L)F1516/7/8/9	Standard PWM	Standard PWM

20.1 Capture Mode

The Capture mode function described in this section is available and identical for CCP modules CCP1 and CCP2.

Capture mode makes use of the 16-bit Timer1 resource. When an event occurs on the CCPx pin, the 16-bit CCPRxH:CCPRxL register pair captures and stores the 16-bit value of the TMR1H:TMR1L register pair, respectively. An event is defined as one of the following and is configured by the CCPxM<3:0> bits of the CCPxCON register:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

When a capture is made, the Interrupt Request Flag bit CCPxIF of the PIRx register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRxH, CCPRxL register pair is read, the old captured value is overwritten by the new captured value.

Figure 20-1 shows a simplified diagram of the Capture operation.

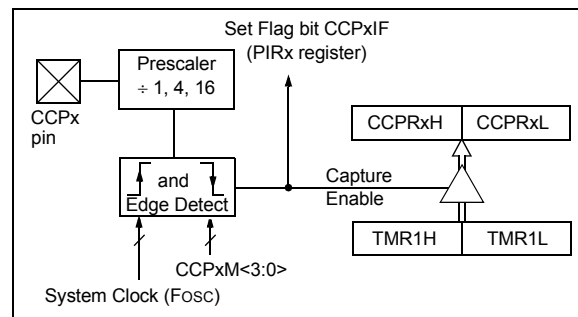
20.1.1 CCP PIN CONFIGURATION

In Capture mode, the CCPx pin should be configured as an input by setting the associated TRIS control bit.

Also, the CCP2 pin function can be moved to alternative pins using the APFCON register. Refer to Register 12-1 for more details.

Note: If the CCPx pin is configured as an output, a write to the port can cause a capture condition.

FIGURE 20-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



20.1.2 TIMER1 MODE RESOURCE

Timer1 must be running in Timer mode or Synchronized Counter mode for the CCP module to use the capture feature. In Asynchronous Counter mode, the capture operation may not work.

See **Section 18.0 “Timer1 Module with Gate Control”** for more information on configuring Timer1.

20.1.3 SOFTWARE INTERRUPT MODE

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit of the PIRx register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF interrupt flag bit of the PIRx register following any change in Operating mode.

PIC16(L)F1516/7/8/9

21.2.3 SPI MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK line. The master determines when the slave (Processor 2, Figure 21-5) is to broadcast data by the software protocol.

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

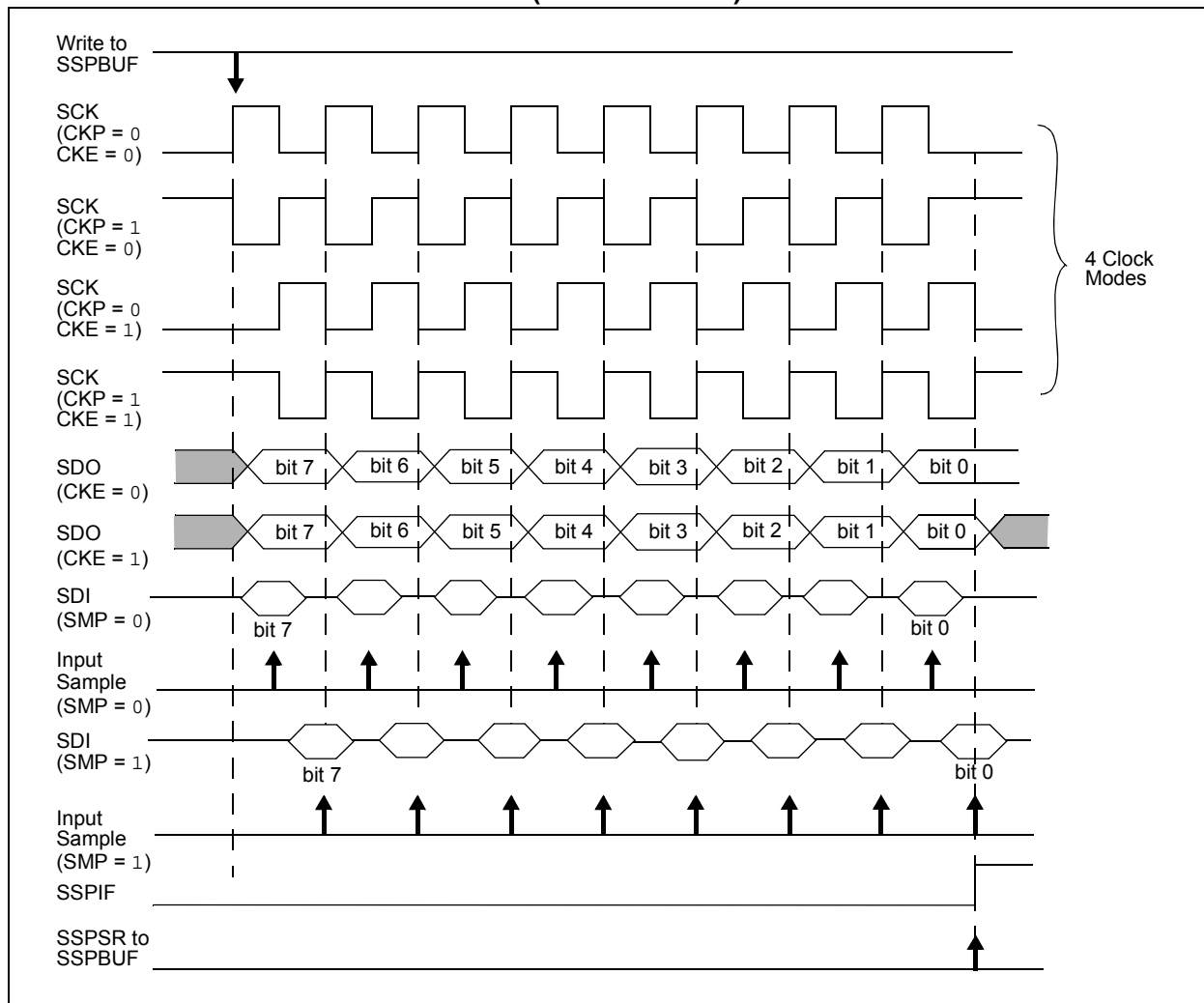
The clock polarity is selected by appropriately programming the CKP bit of the SSPCON1 register and the CKE bit of the SSPSTAT register. This then, would give waveforms for SPI communication as shown in Figure 21-6, Figure 21-8, Figure 21-9 and Figure 21-10, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

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

Figure 21-6 shows the waveforms for Master mode.

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

FIGURE 21-6: SPI MODE WAVEFORM (MASTER MODE)



PIC16(L)F1516/7/8/9

21.5.2.2 7-bit Reception with AHEN and DHEN

Slave device reception with AHEN and DHEN set operate the same as without these options with extra interrupts and clock stretching added after the 8th falling edge of SCL. These additional interrupts allow the slave software to decide whether it wants to $\overline{\text{ACK}}$ the receive address or data byte, rather than the hardware. This functionality adds support for PMBus™ that was not present on previous versions of this module.

This list describes the steps that need to be taken by slave software to use these options for I²C communication. Figure 21-16 displays a module using both address and data holding. Figure 21-17 includes the operation with the SEN bit of the SSPCON2 register set.

1. S bit of SSPSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
2. Matching address with $\overline{\text{R}/\overline{\text{W}}}$ bit clear is clocked in. SSPIF is set and CKP cleared after the 8th falling edge of SCL.
3. Slave clears the SSPIF.
4. Slave can look at the ACKTIM bit of the SSPCON3 register to determine if the SSPIF was after or before the $\overline{\text{ACK}}$.
5. Slave reads the address value from SSPBUF, clearing the BF flag.
6. Slave sets $\overline{\text{ACK}}$ value clocked out to the master by setting ACKDT.
7. Slave releases the clock by setting CKP.
8. SSPIF is set after an $\overline{\text{ACK}}$, not after a NACK.
9. If SEN = 1 the slave hardware will stretch the clock after the $\overline{\text{ACK}}$.
10. Slave clears SSPIF.

<p>Note: SSPIF is still set after the 9th falling edge of SCL even if there is no clock stretching and BF has been cleared. Only if NACK is sent to Master is SSPIF not set</p>
--

11. SSPIF set and CKP cleared after 8th falling edge of SCL for a received data byte.
12. Slave looks at ACKTIM bit of SSPCON3 to determine the source of the interrupt.
13. Slave reads the received data from SSPBUF clearing BF.
14. Steps 7-14 are the same for each received data byte.
15. Communication is ended by either the slave sending an $\overline{\text{ACK}} = 1$, or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop Detect is disabled, the slave will only know by polling the P bit of the SSTSTAT register.

21.5.3.3 7-bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSPCON3 register enables additional clock stretching and interrupt generation after the 8th falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSPIF interrupt is set.

Figure 21-19 displays a standard waveform of a 7-bit Address Slave Transmission with AHEN enabled.

1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSPSTAT is set; SSPIF is set if interrupt on Start detect is enabled.
3. Master sends matching address with $\overline{R/W}$ bit set. After the 8th falling edge of the SCL line the CKP bit is cleared and SSPIF interrupt is generated.
4. Slave software clears SSPIF.
5. Slave software reads ACKTIM bit of SSPCON3 register, and $\overline{R/W}$ and D/A of the SSPSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPBUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to ACK or not ACK and sets the ACKDT bit of the SSPCON2 register accordingly.
8. Slave sets the CKP bit releasing SCL.
9. Master clocks in the \overline{ACK} value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSPIF after the ACK if the $\overline{R/W}$ bit is set.
11. Slave software clears SSPIF.
12. Slave loads value to transmit to the master into SSPBUF setting the BF bit.

Note: SSPBUF cannot be loaded until after the \overline{ACK} .

13. Slave sets the CKP bit, releasing the clock.
14. Master clocks out the data from the slave and sends an ACK value on the 9th SCL pulse.
15. Slave hardware copies the \overline{ACK} value into the ACKSTAT bit of the SSPCON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not \overline{ACK} the slave releases the bus allowing the master to send a Stop and end the communication.

Note: Master must send a not \overline{ACK} on the last byte to ensure that the slave releases the SCL line to receive a Stop.

21.6 I²C MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in the SSPCON1 register and by setting the SSPEN bit. In Master mode, the SDA and SCK pins must be configured as inputs. The MSSP peripheral hardware will override the output driver TRIS controls when necessary to drive the pins low.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I²C bus may be taken when the P bit is set, or the bus is Idle.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit condition detection. Start and Stop condition detection is the only active circuitry in this mode. All other communication is done by the user software directly manipulating the SDA and SCL lines.

The following events will cause the SSP Interrupt Flag bit, SSPIF, to be set (SSP interrupt, if enabled):

- Start condition detected
- Stop condition detected
- Data transfer byte transmitted/received
- Acknowledge transmitted/received
- Repeated Start generated

Note 1: The MSSP module, when configured in I²C Master mode, does not allow queuing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur

2: When in Master mode, Start/Stop detection is masked and an interrupt is generated when the SEN/PEN bit is cleared and the generation is complete.

21.6.1 I²C MASTER MODE OPERATION

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted eight bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received eight bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

A Baud Rate Generator is used to set the clock frequency output on SCL. See **Section 21.7 “Baud Rate Generator”** for more detail.

TABLE 24-3: INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb		LSb			
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C, DC, Z	2
ADDWFC	f, d	Add with Carry W and f	1	11	1101	dfff	ffff	C, DC, Z	2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	2
ASRF	f, d	Arithmetic Right Shift	1	11	0111	dfff	ffff	C, Z	2
LSLF	f, d	Logical Left Shift	1	11	0101	dfff	ffff	C, Z	2
LSRF	f, d	Logical Right Shift	1	11	0110	dfff	ffff	C, Z	2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRW	—	Clear W	1	00	0001	0000	00xx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	2
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	2
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		2
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C, DC, Z	2
SUBWFB	f, d	Subtract with Borrow W from f	1	11	1011	dfff	ffff	C, DC, Z	2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	2
BYTE-ORIENTED SKIP OPERATIONS									
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1, 2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		2
BIT-ORIENTED SKIP OPERATIONS									
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		1, 2
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		1, 2
LITERAL OPERATIONS									
ADDLW	k	Add literal and W	1	11	1110	kkkk	kkkk	C, DC, Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	Move literal to BSR	1	00	0000	001k	kkkk		
MOVLW	k	Move literal to PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	Move literal to W	1	11	0000	kkkk	kkkk		
SUBLW	k	Subtract W from literal	1	11	1100	kkkk	kkkk	C, DC, Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: If the Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

2: If this instruction addresses an INDF register and the MSb of the corresponding FSR is set, this instruction will require one additional instruction cycle.

24.2 Instruction Descriptions

ADDFSR Add Literal to FSRn

Syntax:	[<i>label</i>] ADDFSR FSRn, k
Operands:	$-32 \leq k \leq 31$ $n \in [0, 1]$
Operation:	$FSR(n) + k \rightarrow FSR(n)$
Status Affected:	None
Description:	The signed 6-bit literal 'k' is added to the contents of the FSRnH:FSRnL register pair. FSRn is limited to the range 0000h - FFFFh. Moving beyond these bounds will cause the FSR to wrap-around.

ANDLW AND literal with W

Syntax:	[<i>label</i>] ANDLW k
Operands:	$0 \leq k \leq 255$
Operation:	$(W) .AND. (k) \rightarrow (W)$
Status Affected:	Z
Description:	The contents of W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register.

ADDLW Add literal and W

Syntax:	[<i>label</i>] ADDLW k
Operands:	$0 \leq k \leq 255$
Operation:	$(W) + k \rightarrow (W)$
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the 8-bit literal 'k' and the result is placed in the W register.

ANDWF AND W with f

Syntax:	[<i>label</i>] ANDWF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operation:	$(W) .AND. (f) \rightarrow (\text{destination})$
Status Affected:	Z
Description:	AND 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'.

ADDWF Add W and f

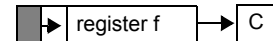
Syntax:	[<i>label</i>] ADDWF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operation:	$(W) + (f) \rightarrow (\text{destination})$
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

ASRF Arithmetic Right Shift

Syntax:	[<i>label</i>] ASRF f {,d}
Operands:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operation:	$(f<7>) \rightarrow \text{dest}<7>$ $(f<7:1>) \rightarrow \text{dest}<6:0>$, $(f<0>) \rightarrow C$,
Status Affected:	C, Z
Description:	The contents of register 'f' are shifted one bit to the right through the Carry flag. The MSb remains unchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

ADDWFC ADD W and CARRY bit to f

Syntax:	[<i>label</i>] ADDWFC f {,d}
Operands:	$0 \leq f \leq 127$ $d \in [0, 1]$
Operation:	$(W) + (f) + (C) \rightarrow \text{dest}$
Status Affected:	C, DC, Z
Description:	Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.



PIC16(L)F1516/7/8/9

RETFIE Return from Interrupt

Syntax: [*label*] RETFIE

Operands: None

Operation: TOS → PC,
1 → GIE

Status Affected: None

Description: Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a 2-cycle instruction.

Words: 1

Cycles: 2

Example:

```
RETFIE
After Interrupt
    PC = TOS
    GIE = 1
```

RETLW Return with literal in W

Syntax: [*label*] RETLW *k*

Operands: $0 \leq k \leq 255$

Operation: *k* → (W);
TOS → PC

Status Affected: None

Description: The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.

Words: 1

Cycles: 2

Example:

```
CALL TABLE;W contains table
               ;offset value
    • ;W now has table value
    •
    •
ADDWF PC ;W = offset
RETLW k1 ;Begin table
RETLW k2 ;
    •
    •
    •
RETLW kn ; End of table
```

TABLE

Before Instruction
W = 0x07

After Instruction
W = value of k8

RETURN Return from Subroutine

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

Description: Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a 2-cycle instruction.

RLF Rotate Left f through Carry

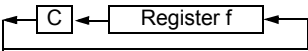
Syntax: [*label*] RLF *f*,*d*

Operands: $0 \leq f \leq 127$
d ∈ [0,1]

Operation: See description below

Status Affected: C

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



Words: 1

Cycles: 1

Example:

```
RLF    REG1,0
```

Before Instruction

```
REG1    = 1110 0110
C        = 0
```

After Instruction

```
REG1    = 1110 0110
W        = 1100 1100
C        = 1
```

25.4 DC Characteristics: Supply Current (IDD)

PIC16LF1516/7/8/9			Standard Operating Conditions (unless otherwise stated)				
			Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended				
PIC16F1516/7/8/9			Standard Operating Conditions (unless otherwise stated)				
			Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ for extended				
Param No.	Device Characteristics	Min.	Typ†	Max.	Units	Conditions	
						VDD	Note
	Supply Current (IDD) ^(1, 2, 3)						
D010		—	8.0	14	μA	1.8	Fosc = 32 kHz
		—	12.0	31	μA	3.0	LP Oscillator $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$
D010		—	11	28	μA	2.3	Fosc = 32 kHz
		—	13	38	μA	3.0	LP Oscillator
		—	14	45	μA	5.0	$-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$
D011		—	60	95	μA	1.8	Fosc = 1 MHz
		—	110	180	μA	3.0	XT Oscillator
D011		—	92	170	μA	2.3	Fosc = 1 MHz
		—	140	230	μA	3.0	XT Oscillator
		—	170	350	μA	5.0	
D012		—	150	240	μA	1.8	Fosc = 4 MHz
		—	260	430	μA	3.0	XT Oscillator
D012		—	190	450	μA	2.3	Fosc = 4 MHz
		—	310	500	μA	3.0	XT Oscillator
		—	370	650	μA	5.0	
D013		—	25	31	μA	1.8	Fosc = 500 kHz
		—	35	50	μA	3.0	EC Oscillator Low-Power mode
D013		—	25	40	μA	2.3	Fosc = 500 kHz
		—	35	55	μA	3.0	EC Oscillator
		—	40	60	μA	5.0	Low-Power mode
D014		—	120	210	μA	1.8	Fosc = 4 MHz
		—	210	380	μA	3.0	EC Oscillator Medium-Power mode
D014		—	160	250	μA	2.3	Fosc = 4 MHz
		—	260	380	μA	3.0	EC Oscillator
		—	330	480	μA	5.0	Medium-Power mode

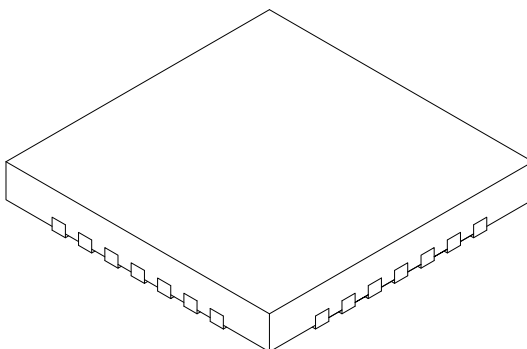
* These parameters are characterized but not tested.

† 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:** The test conditions for all IDD measurements in active operation mode are: OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD; 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.
- 3:** 0.1 μF capacitor on VCAP pin, PIC16F1516/7/8/9 only.
- 4:** For RC oscillator configurations, current through REXT is not included. The current through the resistor can be extended by the formula $I_R = V_{DD}/2R_{EXT}$ (mA) with REXT in kΩ.

28-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 4x4x0.5 mm Body [UQFN]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.40 BSC		
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.127 REF		
Overall Width	E	4.00 BSC		
Exposed Pad Width	E2	2.55	2.65	2.75
Overall Length	D	4.00 BSC		
Exposed Pad Length	D2	2.55	2.65	2.75
Contact Width	b	0.15	0.20	0.25
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- 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.

Microchip Technology Drawing C04-152A Sheet 2 of 2