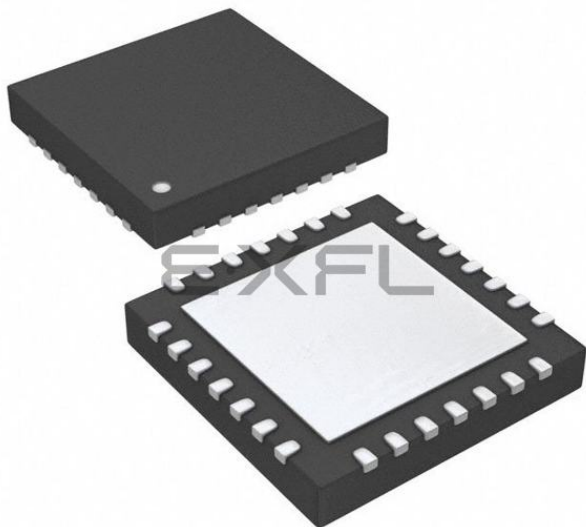




Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	44
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	224 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 43x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf15385-e-mv

TABLE 5: 48-PIN ALLOCATION TABLE (PIC16(L)F15385, PIC16(L)F15386)

I/O ⁽²⁾	48-Pin UQFN/TQFP	ADC	Reference	Comparator	NCO	DAC	Timers	CCP	PWM	CWG	MSSP	ZCD	EUSART	CLC	CLKR	Interrupt	Pull-up	Basic
RA0	21	ANA0	—	C1IN0- C2IN0-	—	—	—	—	—	—	—	—	—	CLCIN0 ⁽¹⁾	—	IOCA0	Y	—
RA1	22	ANA1	—	C1IN1- C2IN1-	—	—	—	—	—	—	—	—	—	CLCIN1 ⁽¹⁾	—	IOCA1	Y	—
RA2	23	ANA2	—	C1IN0+ C2IN0+	—	DAC1OUT1	—	—	—	—	—	—	—	—	—	IOCA2	Y	—
RA3	24	ANA3	VREF+	C1IN1+	—	DACREF+	—	—	—	—	—	—	—	—	—	IOCA3	Y	—
RA4	25	ANA4	—	C1IN1-	—	—	TOCKI ⁽¹⁾	—	—	—	—	—	—	—	—	IOCA4	Y	—
RA5	26	ANA5 ADACT	—	—	—	—	T1G ⁽¹⁾	—	—	—	SS1 ⁽¹⁾	—	—	—	—	IOCA5	Y	—
RA6	33	ANA6	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCA6	Y	CLKOUT/ OSC1
RA7	32	ANA7	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCA7	Y	CLKIN/ OSC2
RB0	8	ANB0	—	C2IN1+	—	—	—	—	—	CWG1 ⁽¹⁾	SS2 ⁽¹⁾	ZCD1	—	—	—	INT ⁽¹⁾ IOCB0	Y	—
RB1	9	ANB1	—	C1IN3- C2IN3-	—	—	—	—	—	—	SCL1 SCK1 ^(1,4)	—	—	—	—	IOCB1	Y	—
RB2	10	ANB2	—	—	—	—	—	—	—	—	SDA1 SDI1 ^(1,4)	—	—	—	—	IOCB2	Y	—
RB3	11	ANB3	—	C1IN2- C2IN2-	—	—	—	—	—	—	—	—	—	—	—	IOCB3	Y	—
RB4	16	ANB4 ADACT ⁽¹⁾	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCB4	Y	—
RB5	17	ANB5	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCB5	Y	—
RB6	18	ANB6	—	—	—	—	—	—	—	—	—	—	TX2 CK2 ⁽¹⁾	CLCIN2 ⁽¹⁾	—	IOCB6	Y	ICSPCLK
RB7	19	ANB7	—	—	—	DAC1OUT2	—	—	—	—	—	—	RX2 DT2 ⁽¹⁾	CLCIN3 ⁽¹⁾	—	IOCB7	Y	ICSPDAT
RC0	34	ANC0	—	—	—	—	SOSCO T1CKI ⁽¹⁾	—	—	—	—	—	—	—	—	IOCC0	Y	—
RC1	35	ANC1	—	—	—	—	SOSCI	CCP2 ⁽¹⁾	—	—	—	—	—	—	—	IOCC1	Y	—

- Note**
- 1: This is a PPS re-mappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins.
 - 2: All digital output signals shown in this row are PPS re-mappable. These signals may be mapped to output onto one of several PORTx pin options.
 - 3: This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.
 - 4: These pins are configured for I²C logic levels. PPS assignments to the other pins will operate, but input logic levels will be standard TTL/ST as selected by the INLVL register, instead of the I²C specific or SMBUS input buffer thresholds.

PIC16(L)F15356/75/76/85/86

TABLE 1-2: PIC16(L)F15356 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RE3/IOCE3/MCLR/VPP	RE3	TTL/ST	—	General purpose input only (when MCLR is disabled by the Configuration bit).
	IOCE3	TTL/ST	—	Interrupt-on-change input.
	MCLR	ST	—	Master clear input with internal weak pull-up resistor.
	VPP	HV	—	ICSP™ High-Voltage Programming mode entry input.
VDD	VDD	Power	—	Positive supply voltage input.
VSS	VSS	Power	—	Ground reference.

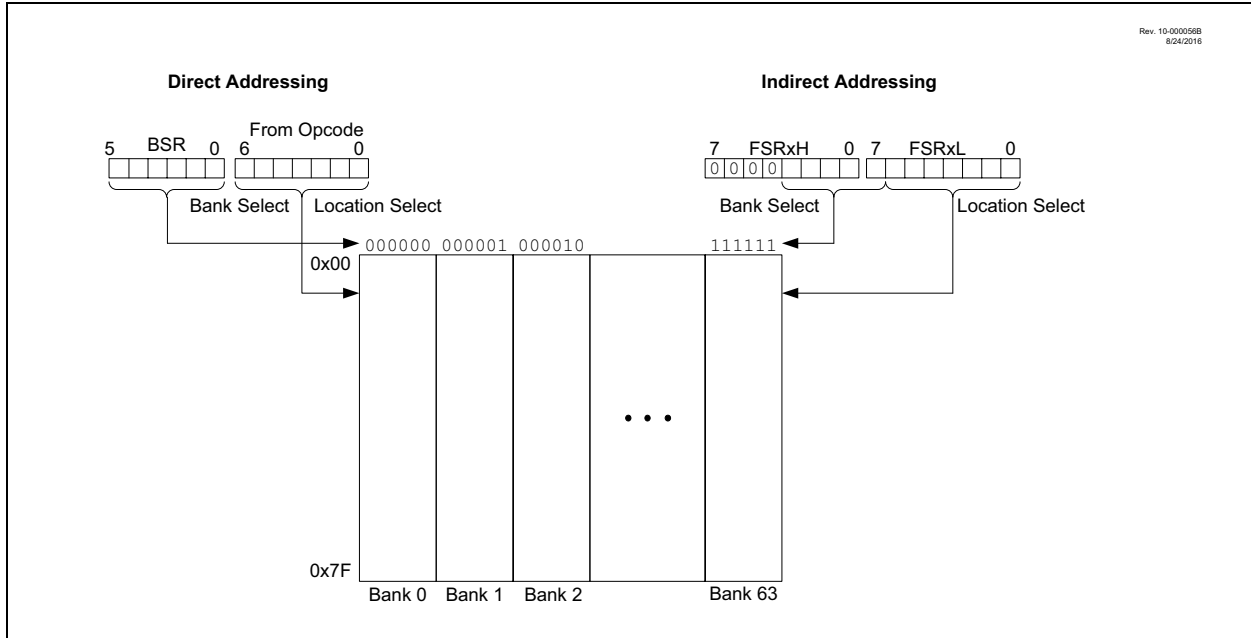
Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open-Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels I²C = Schmitt Trigger input with I²C
HV = High Voltage XTAL = Crystal levels

- Note** 1: This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins. Refer to Table 15-4 for details on which PORT pins may be used for this signal.
- 2: All output signals shown in this row are PPS remappable. These signals may be mapped to output onto one of several PORTx pin options as described in Table 15-3.
- 3: This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.
- 4: These pins are configured for I²C logic levels. The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST, as selected by the INLVL register, instead of the I²C specific or SMBus input buffer thresholds.

4.6.1 TRADITIONAL/BANKED DATA MEMORY

The traditional or banked data memory is a region from FSR address 0x000 to FSR address 0x1FFF. The addresses correspond to the absolute addresses of all SFR, GPR and common registers.

FIGURE 4-11: TRADITIONAL/BANKED DATA MEMORY MAP



13.3.4 NVMREG WRITE TO PROGRAM MEMORY

Program memory is programmed using the following steps:

1. Load the address of the row to be programmed into NVMADRH:NVMADRL.
2. Load each write latch with data.
3. Initiate a programming operation.
4. Repeat steps 1 through 3 until all data is written.

Before writing to program memory, the word(s) to be written must be erased or previously unwritten. Program memory can only be erased one row at a time. No automatic erase occurs upon the initiation of the write.

Program memory can be written one or more words at a time. The maximum number of words written at one time is equal to the number of write latches. See Figure 13-4 (row writes to program memory with 32 write latches) for more details.

The write latches are aligned to the Flash row address boundary defined by the upper ten bits of NVMADRH:NVMADRL, (NVMADRH<6:0>:NVMADRL<7:5>) with the lower five bits of NVMADRL, (NVMADRL<4:0>) determining the write latch being loaded. Write operations do not cross these boundaries. At the completion of a program memory write operation, the data in the write latches is reset to contain 0x3FFF.

The following steps should be completed to load the write latches and program a row of program memory. These steps are divided into two parts. First, each write latch is loaded with data from the NVMDATH:NVMDATL using the unlock sequence with LWLO = 1. When the last word to be loaded into the write latch is ready, the LWLO bit is cleared and the unlock sequence executed. This initiates the programming operation, writing all the latches into Flash program memory.

Note: The special unlock sequence is required to load a write latch with data or initiate a Flash programming operation. If the unlock sequence is interrupted, writing to the latches or program memory will not be initiated.

1. Set the WREN bit of the NVMCON1 register.
2. Clear the NVMREGS bit of the NVMCON1 register.
3. Set the LWLO bit of the NVMCON1 register. When the LWLO bit of the NVMCON1 register is '1', the write sequence will only load the write latches and will not initiate the write to Flash program memory.
4. Load the NVMADRH:NVMADRL register pair with the address of the location to be written.
5. Load the NVMDATH:NVMDATL register pair with the program memory data to be written.
6. Execute the unlock sequence (**Section 13.3.2 "NVM Unlock Sequence"**). The write latch is now loaded.
7. Increment the NVMADRH:NVMADRL register pair to point to the next location.
8. Repeat steps 5 through 7 until all but the last write latch has been loaded.
9. Clear the LWLO bit of the NVMCON1 register. When the LWLO bit of the NVMCON1 register is '0', the write sequence will initiate the write to Flash program memory.
10. Load the NVMDATH:NVMDATL register pair with the program memory data to be written.
11. Execute the unlock sequence (**Section 13.3.2 "NVM Unlock Sequence"**). The entire program memory latch content is now written to Flash program memory.

Note: The program memory write latches are reset to the blank state (0x3FFF) at the completion of every write or erase operation. As a result, it is not necessary to load all the program memory write latches. Unloaded latches will remain in the blank state.

An example of the complete write sequence is shown in Example 13-4. The initial address is loaded into the NVMADRH:NVMADRL register pair; the data is loaded using indirect addressing.

13.3.6 NVMREG ACCESS TO DEVICE INFORMATION AREA, DEVICE CONFIGURATION AREA, USER ID, DEVICE ID AND CONFIGURATION WORDS

NVMREGS can be used to access the following memory regions:

- Device Information Area (DIA)
- Device Configuration Information (DCI)
- User ID region
- Device ID and Revision ID
- Configuration Words

The value of NVMREGS is set to '1' in the NVMCON1 register to access these regions. The memory regions listed above would be pointed to by PC<15> = 1, but not all addresses reference valid data. Different access may exist for reads and writes. Refer to Table 13-3.

When read access is initiated on an address outside the parameters listed in Table 13-3, the NVMDATH: NVMDATL register pair is cleared, reading back '0's.

TABLE 13-3: NVMREGS ACCESS TO DEVICE INFORMATION AREA, DEVICE CONFIGURATION AREA, USER ID, DEVICE ID AND CONFIGURATION WORDS (NVMREGS = 1)

Address	Function	Read Access	Write Access
8000h-8003h	User IDs	Yes	Yes
8005h-8006h	Device ID/Revision ID	Yes	No
8007h-800Bh	Configuration Words 1-5	Yes	No
8100h-82FFh	DIA and DCI	Yes	No

14.10 PORTE Registers

14.10.1 DATA REGISTER

PORTE is a 4-bit wide port. The corresponding data direction register is TRISE (Register 14-33). Setting a TRISE bit (= 1) will make the corresponding PORTE pin an input (i.e., disable the output driver). Clearing a TRISE bit (= 0) will make the corresponding PORTE pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). Figure 14-1 shows how to initialize PORTE.

Reading the PORTE register (Register 14-33) 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 (LATE).

14.10.2 DIRECTION CONTROL

The TRISE register (Register 14-34) controls the PORTE pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISE register are maintained set when using them as analog inputs. I/O pins configured as analog inputs always read '0'.

Note: The TRISE3 bit is a read-only bit and it always reads a '1'.

14.10.3 OPEN-DRAIN CONTROL

The ODCONE register (Register 14-38) controls the open-drain feature of the port. Open-drain operation is independently selected for each pin. When an ODCONE bit is set, the corresponding port output becomes an open-drain driver capable of sinking current only. When an ODCONE bit is cleared, the corresponding port output pin is the standard push-pull drive capable of sourcing and sinking current.

Note: It is not necessary to set open-drain control when using the pin for I²C; the I²C module controls the pin and makes the pin open-drain.

14.10.4 SLEW RATE CONTROL

The SLRCONE register (Register 14-39) controls the slew rate option for each port pin. Slew rate control is independently selectable for each port pin. When an SLRCONE bit is set, the corresponding port pin drive is slew rate limited. When an SLRCONE bit is cleared, the corresponding port pin drive slews at the maximum rate possible.

14.10.5 INPUT THRESHOLD CONTROL

The INLVLE register (Register 14-40) controls the input voltage threshold for each of the available PORTE input pins. A selection between the Schmitt Trigger CMOS or the TTL Compatible thresholds is available. The input threshold is important in determining the value of a read of the PORTE register and also the level at which an interrupt-on-change occurs, if that feature is enabled. See Table 37-4 for more information on threshold levels.

Note: Changing the input threshold selection should be performed while all peripheral modules are disabled. Changing the threshold level during the time a module is active may inadvertently generate a transition associated with an input pin, regardless of the actual voltage level on that pin.

14.10.6 ANALOG CONTROL

The ANSELE register (Register 14-36) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELE 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 ANSELE bits has no effect on digital output functions. A pin with TRIS clear and ANSELE 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 ANSELE bits default to the Analog mode after Reset. To use any pins as digital general purpose or peripheral inputs, the corresponding ANSEL bits must be initialized to '0' by user software.

14.10.7 WEAK PULL-UP CONTROL

The WPUE register (Register 14-37) controls the individual weak pull-ups for each port pin.

PIC16(L)F15356/75/76/85/86

TABLE 17-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPT-ON-CHANGE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	146
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	147
IOCAP	IOCAP7	IOCAP6	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	255
IOCAN	IOCAN7	IOCAN6	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	255
IOCAF	IOCAF7	IOCAF6	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	256
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	257
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	257
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	258
IOCCP	IOCCP7	IOCCP6	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0	259
IOCCN	IOCCN7	IOCCN6	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0	259
IOCCF	IOCCF7	IOCCF6	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0	259
IOCEP	—	—	—	—	IOCEP3	IOCEP2 ⁽¹⁾	IOCEP1 ⁽¹⁾	IOCEP0 ⁽¹⁾	260
IOCEN	—	—	—	—	IOCEN3	IOCEN2 ⁽¹⁾	IOCEN1 ⁽¹⁾	IOCEN0 ⁽¹⁾	260
IOCEF	—	—	—	—	IOCEF3	IOCEF2 ⁽¹⁾	IOCEF1 ⁽¹⁾	IOCEF0 ⁽¹⁾	261

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by interrupt-on-change.

Note 1: Present only in PIC16(L)F15375/76/85/86.

PIC16(L)F15356/75/76/85/86

29.1.6 OPERATION IN SLEEP MODE

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the PWMx pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

29.1.7 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 9.0 “Oscillator Module (with Fail-Safe Clock Monitor)”** for additional details.

29.1.8 EFFECTS OF RESET

Any Reset will force all ports to Input mode and the PWMx registers to their Reset states.

TABLE 29-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 20 MHz)

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

TABLE 29-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (Fosc = 8 MHz)

PWM Frequency	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

29.1.9 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the module for using the PWMx outputs:

1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
2. Configure the PWM output polarity by configuring the PWMxPOL bit of the PWMxCON register.
3. Load the PR2 register with the PWM period value, as determined by Equation 29-1.
4. Load the PWMxDCH register and bits <7:6> of the PWMxDCL register with the PWM duty cycle value, as determined by Equation 29-2.
5. Configure and start Timer2:
 - Clear the TMR2IF interrupt flag bit of the PIR4 register.
 - Select the Timer2 prescale value by configuring the CKPS<2:0> bits of the T2CON register.
 - Enable Timer2 by setting the Timer2 ON bit of the T2CON register.

6. Wait until the TMR2IF is set.

7. When the TMR2IF flag bit is set:

- Clear the associated TRIS bit(s) to enable the output driver.
- Route the signal to the desired pin by configuring the RxyPPS register.
- Enable the PWMx module by setting the PWMxEN bit of the PWMxCON register.

In order to send a complete duty cycle and period on the first PWM output, the above steps must be followed in the order given. If it is not critical to start with a complete PWM signal, then the PWM module can be enabled during Step 2 by setting the PWMxEN bit of the PWMxCON register.

30.3 Selectable Input Sources

The CWG generates the output waveforms from the input sources in Table 30-2.

TABLE 30-2: SELECTABLE INPUT SOURCES

Source Peripheral	Signal Name
CWG input PPS pin	CWG1IN_PPS
CCP1	CCP1_out
CCP2	CCP2_out
PWM3	PWM3_out
PWM4	PWM4_out
PWM5	PWM5_out
PWM6	PWM6_out
NCO	NCO1_out
Comparator C1	C1OUT_sync
Comparator C2	C2OUT_sync
CLC1	LC1_out
CLC2	LC2_out
CLC3	LC3_out
CLC4	LC4_out

The input sources are selected using the CWG1ISM register.

30.4 Output Control

30.4.1 POLARITY CONTROL

The polarity of each CWG output can be selected independently. When the output polarity bit is set, the corresponding output is active-high. Clearing the output polarity bit configures the corresponding output as active-low. However, polarity does not affect the override levels. Output polarity is selected with the POLx bits of the CWG1CON1. Auto-shutdown and steering options are unaffected by polarity.

30.10 Auto-Shutdown

Auto-shutdown is a method to immediately override the CWG output levels with specific overrides that allow for safe shutdown of the circuit. The shutdown state can be either cleared automatically or held until cleared by software. The auto-shutdown circuit is illustrated in Figure 30-12.

30.10.1 SHUTDOWN

The shutdown state can be entered by either of the following two methods:

- Software generated
- External Input

30.10.1.1 Software Generated Shutdown

Setting the SHUTDOWN bit of the CWG1AS0 register will force the CWG into the shutdown state.

When the auto-restart is disabled, the shutdown state will persist as long as the SHUTDOWN bit is set.

When auto-restart is enabled, the SHUTDOWN bit will clear automatically and resume operation on the next rising edge event.

30.10.2 EXTERNAL INPUT SOURCE

External shutdown inputs provide the fastest way to safely suspend CWG operation in the event of a Fault condition. When any of the selected shutdown inputs goes active, the CWG outputs will immediately go to the selected override levels without software delay. Several input sources can be selected to cause a shutdown condition. All input sources are active-low. The sources are:

- Comparator C1OUT_sync
- Comparator C2OUT_sync
- Timer2 – TMR2_postscaled
- CWG1IN input pin

Shutdown inputs are selected using the CWG1AS1 register (Register 30-6).

Note: Shutdown inputs are level sensitive, not edge sensitive. The shutdown state cannot be cleared, except by disabling auto-shutdown, as long as the shutdown input level persists.

30.11 Operation During Sleep

The CWG module operates independently from the system clock and will continue to run during Sleep, provided that the clock and input sources selected remain active.

The HFINTOSC remains active during Sleep when all the following conditions are met:

- CWG module is enabled
- Input source is active
- HFINTOSC is selected as the clock source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and the CWG clock source, when the CWG is enabled and the input source is active, then the CPU will go idle during Sleep, but the HFINTOSC will remain active and the CWG will continue to operate. This will have a direct effect on the Sleep mode current.

32.3.1 CLOCK STRETCHING

When a slave device has not completed processing data, it can delay the transfer of more data through the process of clock stretching. An addressed slave device may hold the SCL clock line low after receiving or sending a bit, indicating that it is not yet ready to continue. The master that is communicating with the slave will attempt to raise the SCL line in order to transfer the next bit, but will detect that the clock line has not yet been released. Because the SCL connection is open-drain, the slave has the ability to hold that line low until it is ready to continue communicating.

Clock stretching allows receivers that cannot keep up with a transmitter to control the flow of incoming data.

32.3.2 ARBITRATION

Each master device must monitor the bus for Start and Stop bits. If the device detects that the bus is busy, it cannot begin a new message until the bus returns to an Idle state.

However, two master devices may try to initiate a transmission on or about the same time. When this occurs, the process of arbitration begins. Each transmitter checks the level of the SDA data line and compares it to the level that it expects to find. The first transmitter to observe that the two levels do not match, loses arbitration, and must stop transmitting on the SDA line.

For example, if one transmitter holds the SDA line to a logical one (lets it float) and a second transmitter holds it to a logical zero (pulls it low), the result is that the SDA line will be low. The first transmitter then observes that the level of the line is different than expected and concludes that another transmitter is communicating.

The first transmitter to notice this difference is the one that loses arbitration and must stop driving the SDA line. If this transmitter is also a master device, it also must stop driving the SCL line. It then can monitor the lines for a Stop condition before trying to reissue its transmission. In the meantime, the other device that has not noticed any difference between the expected and actual levels on the SDA line continues with its original transmission.

Slave Transmit mode can also be arbitrated, when a master addresses multiple slaves, but this is less common.

32.4 I²C MODE OPERATION

All MSSP I²C communication is byte oriented and shifted out MSb first. Six SFR registers and two interrupt flags interface the module with the PIC[®] microcontroller and user software. Two pins, SDA and SCL, are exercised by the module to communicate with other external I²C devices.

32.4.1 BYTE FORMAT

All communication in I²C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent back. After the eighth falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

32.4.2 DEFINITION OF I²C TERMINOLOGY

There is language and terminology in the description of I²C communication that have definitions specific to I²C. That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Philips I²C specification.

32.4.3 SDA AND SCL PINS

Selection of any I²C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain. These pins should be set by the user to inputs by setting the appropriate TRIS bits.

Note 1: Any device pin can be selected for SDA and SCL functions with the PPS peripheral. These functions are bidirectional. The SDA input is selected with the SSPDATPPS registers. The SCL input is selected with the SSPCLKPPS registers. Outputs are selected with the RxyPPS registers. It is the user's responsibility to make the selections so that both the input and the output for each function is on the same pin.

FIGURE 32-12: I²C START AND STOP CONDITIONS

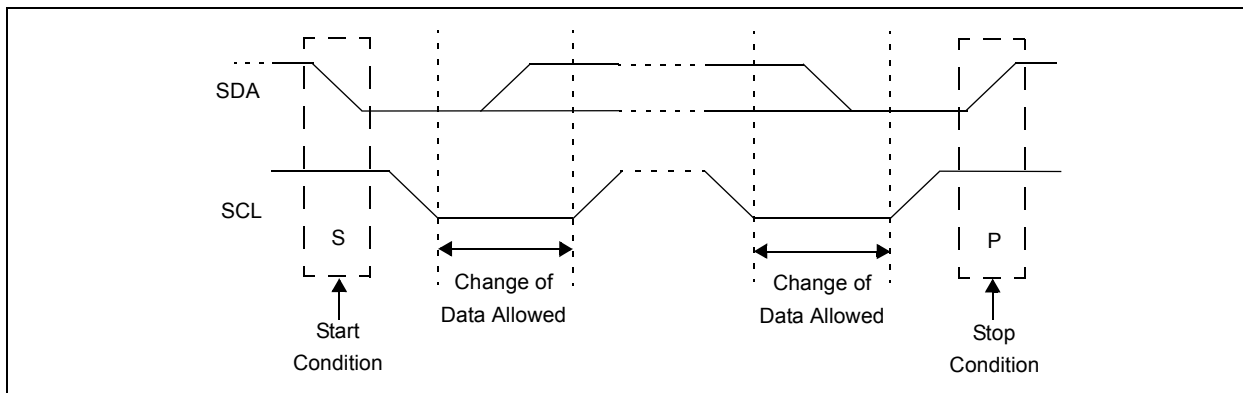
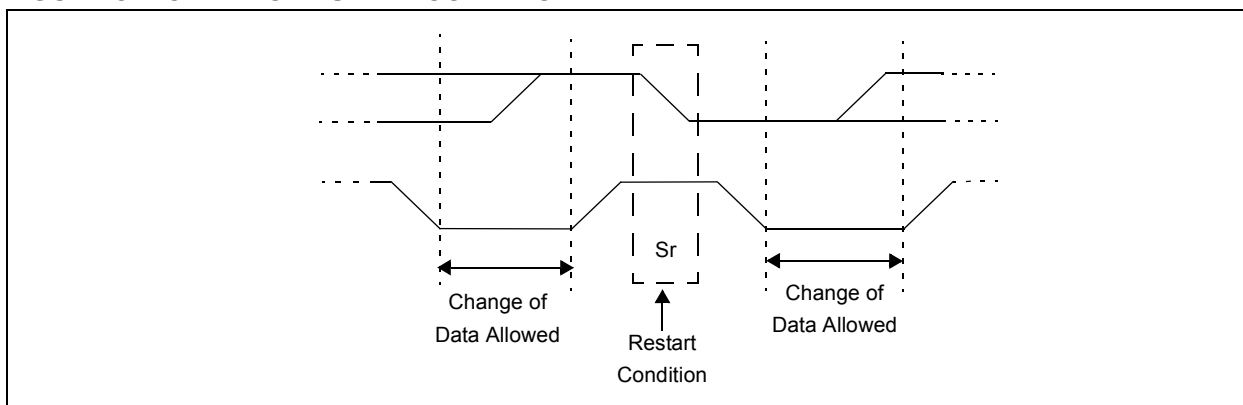


FIGURE 32-13: I²C RESTART CONDITION



32.4.9 ACKNOWLEDGE SEQUENCE

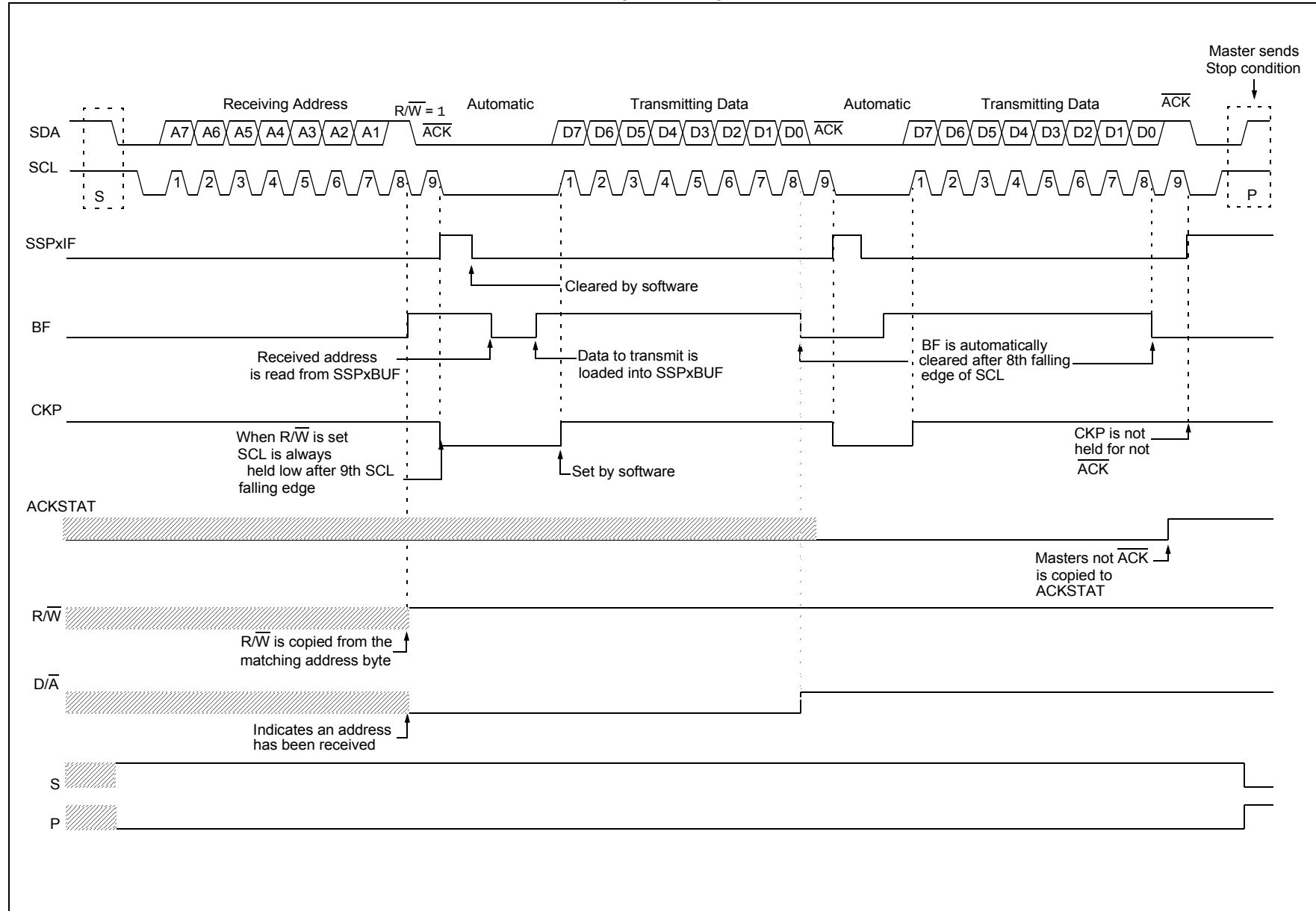
The 9th SCL pulse for any transferred byte in I²C is dedicated as an Acknowledge. It allows receiving devices to respond back to the transmitter by pulling the SDA line low. The transmitter must release control of the line during this time to shift in the response. The Acknowledge ($\overline{\text{ACK}}$) is an active-low signal, pulling the SDA line low indicates to the transmitter that the device has received the transmitted data and is ready to receive more.

The result of an $\overline{\text{ACK}}$ is placed in the ACKSTAT bit of the SSPxCON2 register.

Slave software, when the AHEN and DHEN bits are set, allow the user to set the $\overline{\text{ACK}}$ value sent back to the transmitter. The ACKDT bit of the SSPxCON2 register is set/cleared to determine the response.

There are certain conditions where an $\overline{\text{ACK}}$ will not be sent by the slave. If the BF bit of the SSPxSTAT register or the SSPOV bit of the SSPxCON1 register are set when a byte is received.

When the module is addressed, after the eighth falling edge of SCL on the bus, the ACKTIM bit of the SSPxCON3 register is set. The ACKTIM bit indicates the acknowledge time of the active bus. The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is enabled.

FIGURE 32-18: I²C SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 0)

PIC16(L)F15356/75/76/85/86

REGISTER 32-5: 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
SSPxMSK<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 **SSPxMSK<7:1>**: Mask bits
 1 = The received address bit n is compared to SSPxADD<n> to detect I²C address match
 0 = The received address bit n is not used to detect I²C address match
- bit 0 **SSPxMSK<0>**: Mask bit for I²C Slave mode, 10-bit Address
 I²C Slave mode, 10-bit address (SSPM<3:0> = 0111 or 1111):
 1 = The received address bit 0 is compared to SSPxADD<0> to detect I²C address match
 0 = The received address bit 0 is not used to detect I²C address match
 I²C Slave mode, 7-bit address:
 MSK0 bit is ignored.

REGISTER 32-6: SSPxADD: MSSPx ADDRESS AND BAUD RATE REGISTER (I²C MODE)

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
SSPxADD<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	

Master mode:

- bit 7-0 **SSPxADD<7:0>**: Baud Rate Clock Divider bits

$$\text{SCL pin clock period} = ((\text{ADD}<7:0> + 1) * 4) / \text{Fosc}$$

10-Bit Slave mode – Most Significant Address Byte:

- bit 7-3 **Not used**: Unused for Most Significant Address Byte. Bit state of this register is a “don't care”. Bit pattern sent by master is fixed by I²C specification and must be equal to '11110'. However, those bits are compared by hardware and are not affected by the value in this register.
- bit 2-1 **SSPxADD<2:1>**: Two Most Significant bits of 10-bit address
- bit 0 **Not used**: Unused in this mode. Bit state is a “don't care”.

10-Bit Slave mode – Least Significant Address Byte:

- bit 7-0 **SSPxADD<7:0>**: Eight Least Significant bits of 10-bit address

7-Bit Slave mode:

- bit 7-1 **SSPxADD<7:1>**: 7-bit address
- bit 0 **Not used**: Unused in this mode. Bit state is a “don't care”.

PIC16(L)F15356/75/76/85/86

REGISTER 33-2: RCxSTA: RECEIVE STATUS AND CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0
SPEN ⁽¹⁾	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
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 **SPEN:** Serial Port Enable bit⁽¹⁾

1 = Serial port enabled

0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-Bit Receive Enable bit

1 = Selects 9-bit reception

0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit

Asynchronous mode:

Unused in this mode – value ignored

Synchronous mode – Master:

1 = Enables single receive

0 = Disables single receive

This bit is cleared after reception is complete.

Synchronous mode – Slave

Unused in this mode – value ignored

bit 4 **CREN:** Continuous Receive Enable bit

Asynchronous mode:

1 = Enables continuous receive until enable bit CREN is cleared

0 = Disables continuous receive

Synchronous mode:

1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)

0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit

Asynchronous mode 9-bit (RX9 = 1):

1 = Enables address detection – enable interrupt and load of the receive buffer when the ninth bit in the receive buffer is set

0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit

Asynchronous mode 8-bit (RX9 = 0):

Unused in this mode – value ignored

bit 2 **FERR:** Framing Error bit

1 = Framing error (can be updated by reading RCxREG register and receive next valid byte)

0 = No framing error

bit 1 **OERR:** Overrun Error bit

1 = Overrun error (can be cleared by clearing bit CREN)

0 = No overrun error

bit 0 **RX9D:** Ninth bit of Received Data

This can be address/data bit or a parity bit and must be calculated by user firmware.

Note 1: The EUSART module automatically changes the pin from tri-state to drive as needed. Configure the associated TRIS bits for TX/CK and RX/DT to 1.

PIC16(L)F15356/75/76/85/86

REGISTER 33-7: SPxBRGH^(1, 2): BAUD RATE GENERATOR HIGH REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPxBRG<15:8>							
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 **SPxBRG<15:8>**: Upper eight bits of the Baud Rate Generator

Note 1: SPxBRGH value is ignored for all modes unless BAUDxCON<BRG16> is active.

2: Writing to SPxBRGH resets the BRG counter.

PIC16(L)F15356/75/76/85/86

TABLE 36-3: INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb		LSb			
CONTROL OPERATIONS									
BRA	k	Relative Branch	2	11	001k	kkkk	kkkk		
BRW	—	Relative Branch with W	2	00	0000	0000	1011		
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CALLW	—	Call Subroutine with W	2	00	0000	0000	1010		
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
RETFIE	k	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	0100	kkkk	kkkk		
RETURN	—	Return from Subroutine	2	00	0000	0000	1000		
INHERENT OPERATIONS									
CLRWDT	—	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
NOP	—	No Operation	1	00	0000	0000	0000		
RESET	—	Software device Reset	1	00	0000	0000	0001		
SLEEP	—	Go into Standby or IDLE mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$	
TRIS	f	Load TRIS register with W	1	00	0000	0110	0fff		
C-COMPILER OPTIMIZED									
ADDFSR	n, k	Add Literal k to FSRn	1	11	0001	0nkk	kkkk		
MOVIW	n mm	Move Indirect FSRn to W with pre/post inc/dec modifier, mm	1	00	0000	0001	0nmm	Z	2, 3
MOVWI	k[n]	Move INDFn to W, Indexed Indirect.	1	11	1111	0nkk	kkkk	Z	2
	n mm	Move W to Indirect FSRn with pre/post inc/dec modifier, mm	1	00	0000	0001	1nmm		2, 3
	k[n]	Move W to INDFn, Indexed Indirect.	1	11	1111	1nkk	kkkk		2

- 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.
- 3:** See Table in the MOVIW and MOVWI instruction descriptions.

37.2 Standard Operating Conditions

The standard operating conditions for any device are defined as:

Operating Voltage: $V_{DDMIN} \leq V_{DD} \leq V_{DDMAX}$

Operating Temperature: $T_{A_MIN} \leq T_A \leq T_{A_MAX}$

V_{DD} — Operating Supply Voltage⁽¹⁾

PIC16LF15356/75/76/85/86

V_{DDMIN} (Fosc ≤ 16 MHz) +1.8V

V_{DDMIN} (Fosc ≤ 32 MHz) +2.5V

V_{DDMAX} +3.6V

PIC16F15356/75/76/85/86

V_{DDMIN} (Fosc ≤ 16 MHz) +2.3V

V_{DDMIN} (Fosc ≤ 32 MHz) +2.5V

V_{DDMAX} +5.5V

T_A — Operating Ambient Temperature Range

Industrial Temperature

T_{A_MIN} -40°C

T_{A_MAX} +85°C

Extended Temperature

T_{A_MIN} -40°C

T_{A_MAX} +125°C

Note 1: See Parameter Supply Voltage, DS Characteristics: Supply Voltage.

PIC16(L)F15356/75/76/85/86

TABLE 37-8: INTERNAL OSCILLATOR PARAMETERS⁽¹⁾

Standard Operating Conditions (unless otherwise stated)							
Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
OS50	FHFOSC	Precision Calibrated HFINTOSC Frequency	—	4 8 12 16 32	—	MHz	(Note 2)
OS51	FHFOSCLP	Low-Power Optimized HFINTOSC Frequency	— —	1 2	— —	MHz MHz	
OS52	FMFOSC	Internal Calibrated MFINTOSC Frequency	—	500	—	kHz	
OS53*	FLFOSC	Internal LFINTOSC Frequency	—	31	—	kHz	
OS54*	THFOSCST	HFINTOSC Wake-up from Sleep Start-up Time	— —	11 50	20 —	μs μs	VREGPM = 0 VREGPM = 1
OS56	TLFOSCST	LFINTOSC Wake-up from Sleep Start-up Time	—	0.2	—	ms	

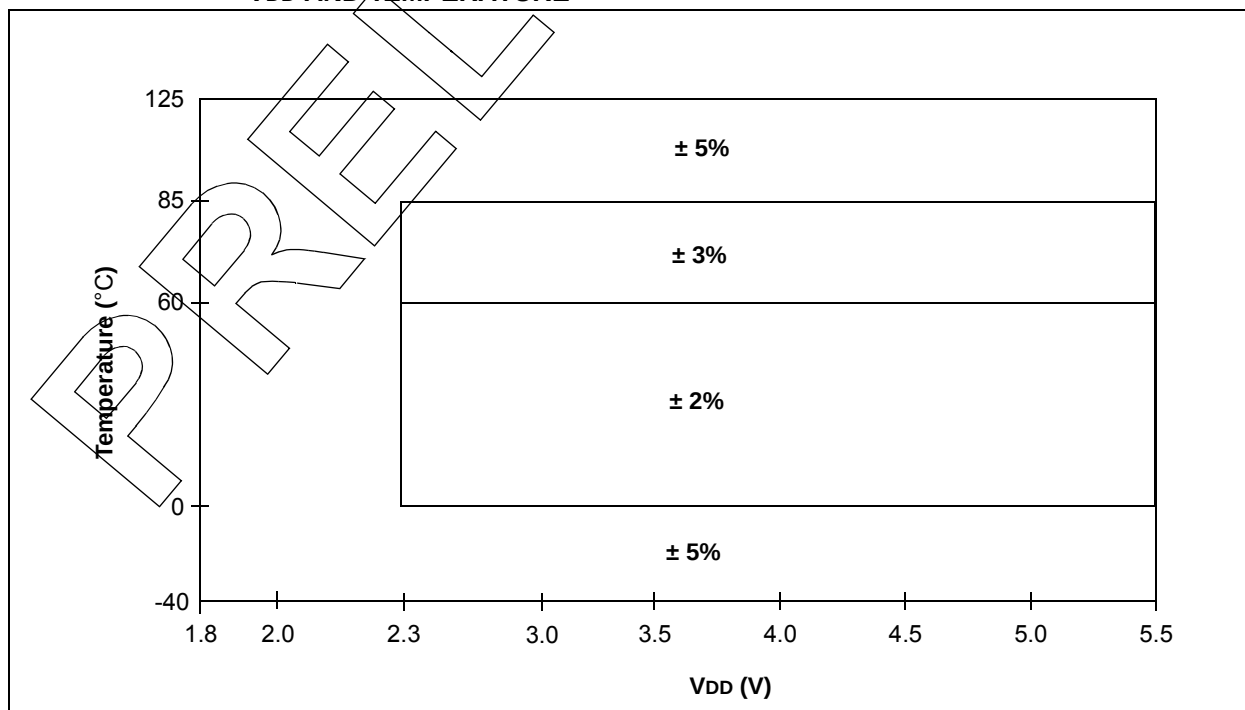
* These parameters are characterized but not tested.

† 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 oscillator frequency 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.

2: See Figure 37-6: Precision Calibrated HFINTOSC Frequency Accuracy Over Device VDD and Temperature.

FIGURE 37-6: PRECISION CALIBRATED HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE



39.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

39.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradeable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

39.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

39.9 PICkit 3 In-Circuit Debugger/Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

39.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.