



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	64MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 35x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f45k42-e-ml

PIC18(L)F26/27/45/46/47/55/56/57K42

4.0 MEMORY ORGANIZATION

There are three types of memory in PIC18 microcontroller devices:

- Program Flash Memory
- Data RAM
- Data EEPROM

As Harvard architecture devices, the data and program memories use separate buses; this allows for concurrent access of the two memory spaces. The data EEPROM, for practical purposes, can be regarded as a peripheral device, since it is addressed and accessed through a set of control registers.

Additional detailed information on the operation of the Program Flash Memory and Data EEPROM Memory is provided in [Section 13.0 “Nonvolatile Memory \(NVM\) Control”](#).

4.1 Program Flash Memory Organization

PIC18 microcontrollers implement a 21-bit program counter, which is capable of addressing a 2 Mbyte program memory space. Accessing any unimplemented memory will return all '0's (a NOP instruction).

These devices contain the following:

- PIC18(L)F45/55K42: 32 Kbytes of Flash memory, up to 16,384 single-word instructions
- PIC18(L)F26/46/56K42: 64 Kbytes of Flash memory, up to 32,768 single-word instructions
- PIC18(L)F27/47/57K42: 128 Kbytes of Flash memory, up to 65,536 single-word instructions

The Reset vector for the device is at address 000000h. PIC18(L)F26/27/45/46/47/55/56/57K42 devices feature a vectored interrupt controller with a dedicated interrupt vector table in the program memory, see [Section 9.0 “Interrupt Controller”](#).

Note: For memory information on this family of devices, see [Table 4-1](#) and [Table 4-3](#).

4.2 Memory Access Partition (MAP)

Program Flash memory is partitioned into:

- Application Block
- Boot Block, and
- Storage Area Flash (SAF) Block

4.2.1 APPLICATION BLOCK

Application block is where the user's program resides by default. Default settings of the configuration bits ($\overline{\text{BBEN}} = 1$ and $\overline{\text{SAFEN}} = 1$) assign all memory in the program Flash memory area to the application block. The WRTAPP configuration bit is used to protect the application block.

4.2.2 BOOT BLOCK

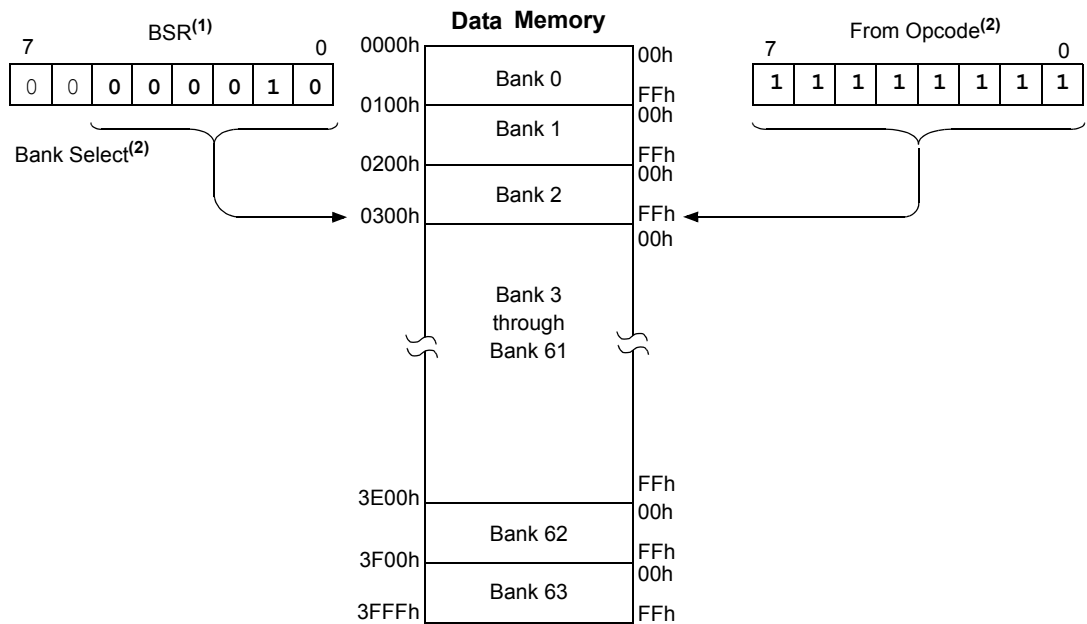
Boot block is an area in program memory that is ideal for storing bootloader code. Code placed in this area can be executed by the CPU. The boot block can be write-protected, independent of the main application block. The Boot Block is enabled by the $\overline{\text{BBEN}}$ bit and size is based on the value of the BBSIZE bits of Configuration word ([Register 5-7](#)), see [Table 5-1](#) for boot block sizes. The $\overline{\text{WRTB}}$ Configuration bit is used to write-protect the Boot Block.

4.2.3 STORAGE AREA FLASH

Storage Area Flash (SAF) is the area in program memory that can be used as data storage. SAF is enabled by the $\overline{\text{SAFEN}}$ bit of the Configuration word in [Register 5-7](#). If enabled, the code placed in this area cannot be executed by the CPU. The SAF block is placed at the end of memory and spans 128 Words. The $\overline{\text{WRTSAF}}$ Configuration bit is used to write-protect the Storage Area Flash.

Note: If write-protected locations are written to, memory is not changed and the WRERR bit defined in [Register 13-1](#) is set.

FIGURE 4-5: USE OF THE BANK SELECT REGISTER (DIRECT ADDRESSING)



Note 1: The Access RAM bit of the instruction can be used to force an override of the selected bank (BSR<5:0>) to the registers of the Access Bank.

PIC18(L)F26/27/45/46/47/55/56/57K42

5.5 Device ID and Revision ID

The 16-bit device ID word is located at 3F FFEh and the 16-bit revision ID is located at 3F FFFCh. These locations are read-only and cannot be erased or modified.

Development tools, such as device programmers and debuggers, may be used to read the Device ID, Revision ID and Configuration Words. Refer to [13.0 “Nonvolatile Memory \(NVM\) Control”](#) for more information on accessing these locations.

5.6 Register Definitions: Device ID and Revision ID

REGISTER 5-11: DEVICE ID: DEVICE ID REGISTER

R	R	R	R	R	R	R	R
DEV<15:8>							
bit 15				bit 8			

R	R	R	R	R	R	R	R
DEV<7:0>							
bit 7				bit 0			

Legend:

R = Readable bit

'1' = Bit is set

0' = Bit is cleared

x = Bit is unknown

bit 15-0

DEV<15:0>: Device ID bits

Device	Device ID
PIC18F26K42	6C60h
PIC18F27K42	6C40h
PIC18F45K42	6C20h
PIC18F46K42	6C00h
PIC18F47K42	6BE0h
PIC18F55K42	6BC0h
PIC18F56K42	6BA0h
PIC18F57K42	6B80h
PIC18LF26K42	6DA0h
PIC18LF27K42	6D80h
PIC18LF45K42	6D60h
PIC18LF46K42	6D40h
PIC18LF47K42	6D20h
PIC18LF55K42	6D00h
PIC18LF56K42	6CE0h
PIC18LF57K42	6CC0h

PIC18(L)F26/27/45/46/47/55/56/57K42

6.3 Register Definitions: BOR Control

REGISTER 6-1: BORCON: BROWN-OUT RESET CONTROL REGISTER

R/W-1/u	U-0	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN	—	—	—	—	—	—	BORRDY
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

q = Value depends on condition

bit 7 **SBOREN:** Software Brown-out Reset Enable bit

If **BOREN** \neq 01:

SBOREN is read/write, but has no effect on the BOR.

If **BOREN** = 01:

1 = BOR Enabled

0 = BOR Disabled

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

bit 0 **BORRDY:** Brown-out Reset Circuit Ready Status bit

1 = The Brown-out Reset Circuit is active and armed

0 = The Brown-out Reset Circuit is disabled or is warming up

PIC18(L)F26/27/45/46/47/55/56/57K42

REGISTER 9-20: PIE6: PERIPHERAL INTERRUPT ENABLE REGISTER 6

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
TMR3GIE	TMR3IE	U2IE	U2EIE	U2TXIE	U2RXIE	I2C2EIE	I2C2IE
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

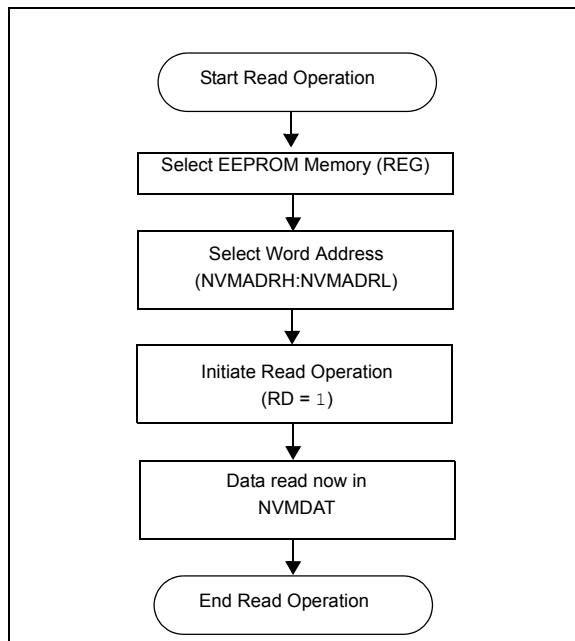
bit 7	TMR3GIE: TMR3 Gate Interrupt Enable bit 1 = Enabled 0 = Disabled
bit 6	TMR3IE: TMR3 Interrupt Enable bit 1 = Enabled 0 = Disabled
bit 5	U2IE: UART2 Interrupt Enable bit 1 = Enabled 0 = Disabled
bit 4	U2EIE: UART2 Framing Error Interrupt Enable bit 1 = Enabled 0 = Disabled
bit 3	U2TXIE: UART2 Transmit Interrupt Enable bit 1 = Enabled 0 = Disabled
bit 2	U2RXIE: UART2 Receive Interrupt Enable bit 1 = Enabled 0 = Disabled
bit 1	I2C2EIE: I ² C2 Error Interrupt Enable bit 1 = Enabled 0 = Disabled
bit 0	I2C2IE: I ² C2 Interrupt Enable bit 1 = Enabled 0 = Disabled

13.3.3 READING THE DATA EEPROM MEMORY

To read a data memory location, the user must write the address to the NVMADRL and NVMADRH register pair, clear REG<1:0> control bit in NVMCON1 register to access Data EEPROM locations and then set control bit, RD. The data is available on the very next instruction cycle; therefore, the NVMDAT register can be read by the next instruction. NVMDAT will hold this value until another read operation, or until it is written to by the user (during a write operation).

The basic process is shown in [Example 13-5](#).

FIGURE 13-11: DATA EEPROM READ FLOWCHART



13.3.4 WRITING TO THE DATA EEPROM MEMORY

To write an EEPROM data location, the address must first be written to the NVMADRL and NVMADRH register pair and the data written to the NVMDAT register. The sequence in [Example 13-6](#) must be followed to initiate the write cycle.

The write will not begin if NVM Unlock sequence, described in [Section 13.1.4 “NVM Unlock Sequence”](#), is not exactly followed for each byte. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in NVMCON1 must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code execution (i.e., runaway programs). The WREN bit should be kept clear at all times, except when updating the EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, NVMCON1, NVMADRL, NVMADRH and NVMDAT cannot be modified. The WR bit will be inhibited from being set unless the WREN bit is set. Both WR and WREN cannot be set with the same instruction.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. A single Data EEPROM word is written and the operation includes an implicit erase cycle for that word (it is not necessary to set FREE). CPU execution continues in parallel and at the completion of the write cycle, the WR bit is cleared in hardware and the NVM Interrupt Flag bit (NVMIF) is set. The user can either enable this interrupt or poll this bit. NVMIF must be cleared by software.

PIC18(L)F26/27/45/46/47/55/56/57K42

REGISTER 14-6: CRCACCL: CRC ACCUMULATOR LOW BYTE REGISTER

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
ACC<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 7-0 **ACC<7:0>**: CRC Accumulator Register bits

REGISTER 14-7: CRCSHIFTH: CRC SHIFT HIGH BYTE REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SHIFT<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 7-0 **SHIFT<15:8>**: CRC Shifter Register bits
Reading from this register reads the CRC Shifter.

REGISTER 14-8: CRCSHIFTL: CRC SHIFT LOW BYTE REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SHIFT<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 7-0 **SHIFT<7:0>**: CRC Shifter Register bits
Reading from this register reads the CRC Shifter.

PIC18(L)F26/27/45/46/47/55/56/57K42

REGISTER 14-18: SCANTRIG: SCAN TRIGGER SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	TSEL<3:0>			
bit 7				bit 0			

Legend:

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

bit 7-4

Unimplemented: Read as '0'

bit 3-0

TSEL<3:0>: Scanner Data Trigger Input Selection bits

1111 = Reserved

•
•
•

1010 = Reserved

1001 = SMT1_output

1000 = TMR6_postscaled

0111 = TMR5_output

0110 = TMR4_postscaled

0101 = TMR3_output

0100 = TMR2_postscaled

0011 = TMR1_output

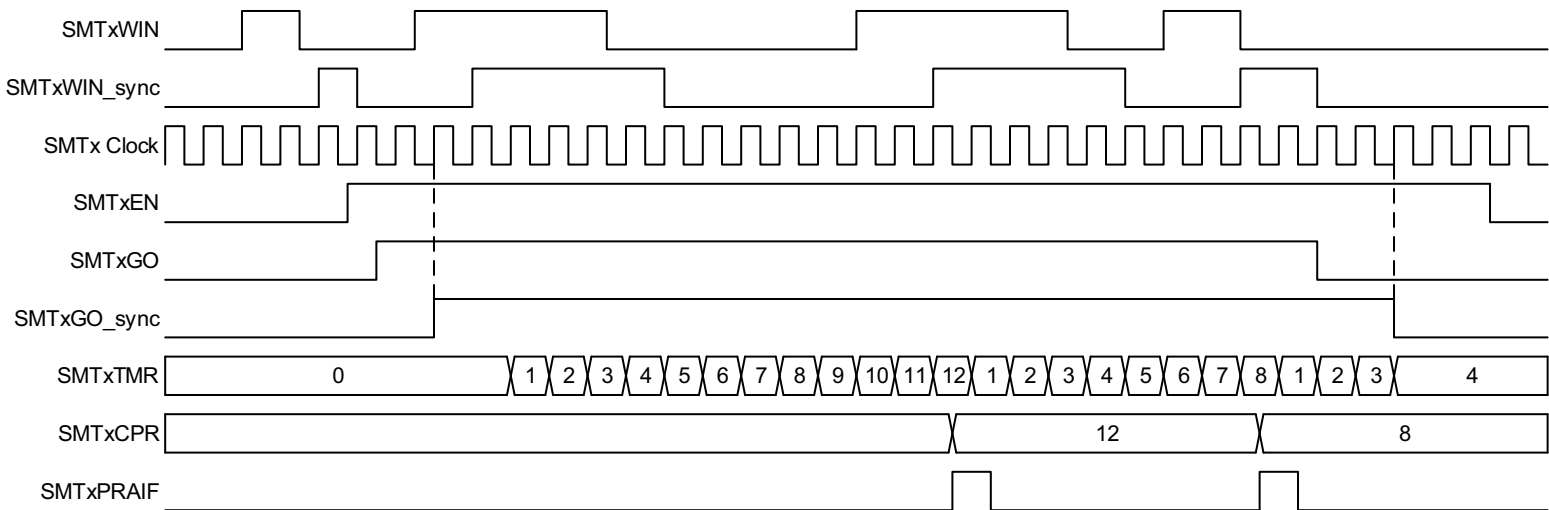
0010 = TMR0_output

0001 = CLKREF_output

0000 = LFINTOSC

Rev. 10-000 182A
12/19/2013

FIGURE 25-10: WINDOWED MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM



25.6.8 CAPTURE MODE

This mode captures the Timer value based on a rising or falling edge on the SMTWINx input and triggers an interrupt. This mimics the capture feature of a CCP module. The timer begins incrementing upon the GO bit being set, and updates the value of the SMT1CPR register on each rising edge of SMTWINx, and updates the value of the CPW register on each falling edge of the SMTWINx. The timer is not reset by any hardware conditions in this mode and must be reset by software, if desired. See [Figure 25-16](#) and [Figure 25-17](#).

PIC18(L)F26/27/45/46/47/55/56/57K42

REGISTER 27-10: CLCxGLS3: GATE 3 LOGIC SELECT REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N
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 **G4D4T:** Gate 3 Data 4 True (non-inverted) bit
1 = CLCIN3 (true) is gated into CLCx Gate 3
0 = CLCIN3 (true) is not gated into CLCx Gate 3
- bit 6 **G4D4N:** Gate 3 Data 4 Negated (inverted) bit
1 = CLCIN3 (inverted) is gated into CLCx Gate 3
0 = CLCIN3 (inverted) is not gated into CLCx Gate 3
- bit 5 **G4D3T:** Gate 3 Data 3 True (non-inverted) bit
1 = CLCIN2 (true) is gated into CLCx Gate 3
0 = CLCIN2 (true) is not gated into CLCx Gate 3
- bit 4 **G4D3N:** Gate 3 Data 3 Negated (inverted) bit
1 = CLCIN2 (inverted) is gated into CLCx Gate 3
0 = CLCIN2 (inverted) is not gated into CLCx Gate 3
- bit 3 **G4D2T:** Gate 3 Data 2 True (non-inverted) bit
1 = CLCIN1 (true) is gated into CLCx Gate 3
0 = CLCIN1 (true) is not gated into CLCx Gate 3
- bit 2 **G4D2N:** Gate 3 Data 2 Negated (inverted) bit
1 = CLCIN1 (inverted) is gated into CLCx Gate 3
0 = CLCIN1 (inverted) is not gated into CLCx Gate 3
- bit 1 **G4D1T:** Gate 4 Data 1 True (non-inverted) bit
1 = CLCIN0 (true) is gated into CLCx Gate 3
0 = CLCIN0 (true) is not gated into CLCx Gate 3
- bit 0 **G4D1N:** Gate 3 Data 1 Negated (inverted) bit
1 = CLCIN0 (inverted) is gated into CLCx Gate 3
0 = CLCIN0 (inverted) is not gated into CLCx Gate 3

28.1 NCO Operation

The NCO operates by repeatedly adding a fixed value to an accumulator. Additions occur at the input clock rate. The accumulator will overflow with a carry periodically, which is the raw NCO output (NCO_overflow). This effectively reduces the input clock by the ratio of the addition value to the maximum accumulator value. See [Equation 28-1](#).

The NCO output can be further modified by stretching the pulse or toggling a flip-flop. The modified NCO output is then distributed internally to other peripherals and can be optionally output to a pin. The accumulator overflow also generates an interrupt (NCO_overflow).

The NCO period changes in discrete steps to create an average frequency. This output depends on the ability of the receiving circuit (i.e., CWG or external resonant converter circuitry) to average the NCO output to reduce uncertainty.

EQUATION 28-1: NCO OVERFLOW FREQUENCY

$$F_{OVERFLOW} = \frac{\text{NCO Clock Frequency} \times \text{Increment Value}}{2^{20}}$$

28.1.1 NCO CLOCK SOURCES

Clock sources available to the NCO include:

- FOSC
- HFINTOSC
- LFINTOSC
- MFINTOSC/4 (32 kHz)
- MFINTOSC (500 kHz)
- CLC1/2/3/4_out
- CLKREF
- SOSC

The NCO clock source is selected by configuring the N1CK<2:0> bits in the NCO1CLK register.

28.1.2 ACCUMULATOR

The accumulator is a 20-bit register. Read and write access to the accumulator is available through three registers:

- NCO1ACCL
- NCO1ACCH
- NCO1ACCU

28.1.3 ADDER

The NCO Adder is a full adder, which operates independently from the source clock. The addition of the previous result and the increment value replaces the accumulator value on the rising edge of each input clock.

28.1.4 INCREMENT REGISTERS

The increment value is stored in three registers making up a 20-bit incrementer. In order of LSB to MSB they are:

- NCO1INCL
- NCO1INCH
- NCO1INCUI

When the NCO module is enabled, the NCO1INCUI and NCO1INCH registers should be written first, then the NCO1INCL register. Writing to the NCO1INCL register initiates the increment buffer registers to be loaded simultaneously on the second rising edge of the NCO_clk signal.

The registers are readable and writable. The increment registers are double-buffered to allow value changes to be made without first disabling the NCO module.

When the NCO module is disabled, the increment buffers are loaded immediately after a write to the increment registers.

Note: The increment buffer registers are not user-accessible.

FIGURE 31-13: AUTO-WAKE-UP BIT (WUE) TIMING DURING NORMAL OPERATION

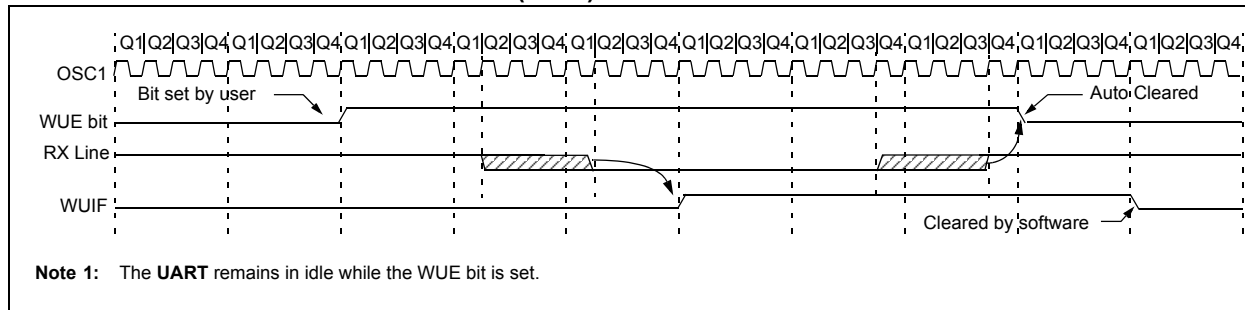
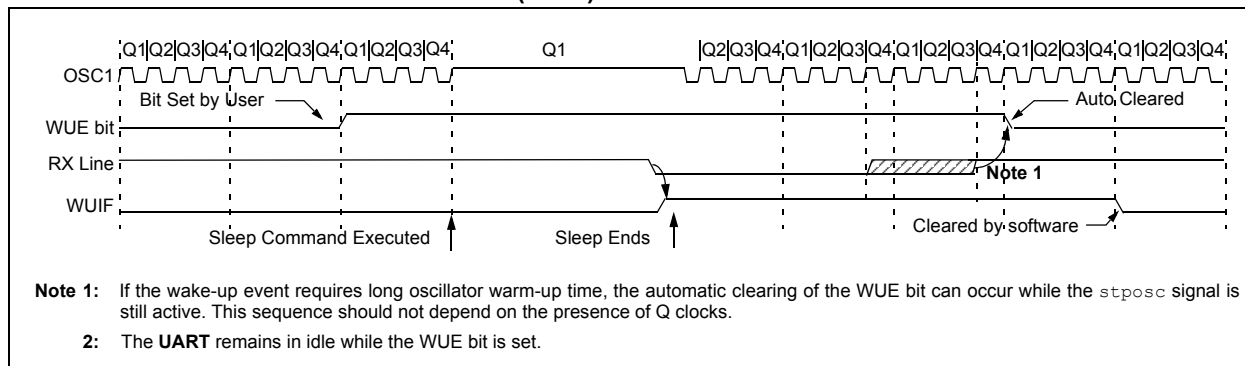


FIGURE 31-14: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP



31.18 Transmitting a Break

The UART module has the capability of sending either a fixed length Break period or a software timed Break period. The fixed length Break consists of a Start bit, followed by 12 '0' bits and a Stop bit. The software timed Break is generated by setting and clearing the BRKOVF bit in the UxCON1 register.

To send the fixed length Break, set the SENDB and TXEN bits in the UxCON0 register. The Break sequence is then initiated by a write to UxTXB. The timed Break will occur first, followed by the character written to UxTXB that initiated the Break. The initiating character is typically the Sync character of the LIN specification.

SENB is disabled in the LIN and DMX modes because those modes generate the Break sequence automatically.

The SENDB bit is automatically reset by hardware after the Break Stop bit is complete.

The TXMTIF bit in the UxERRIR register indicates when the transmit operation is active or idle, just as it does during normal transmission. See Figure 31-15 for the timing of the Break sequence.

31.19 Receiving a Break

The UART has counters to detect when the RX input remains in the space state for an extended period of time. When this happens, the RXBKIF bit in the UxERRIR register is set.

A Break is detected when the RX input remains in the space state for 11 bit periods for asynchronous and LIN modes, and 23 bit periods for DMX mode.

The user can select to receive the Break interrupt as soon as the Break is detected or at the end of the Break, when the RX input returns to the Idle state. When the RXBIMD bit in the UxCON1 is '1' then RXBKIF is set immediately upon Break detection. When RXBIMD is '0' then RXBKIF is set when the RX input returns to the Idle state.

31.20 UART Operation During Sleep

The UART ceases to operate during Sleep. The safe way to wake the device from Sleep by a serial operation is to use the Wake-on-Break feature of the UART. See Section 31.17.3, Auto-Wake-up on Break

32.6 Slave Mode

32.6.1 SLAVE MODE TRANSMIT OPTIONS

The SDO output of the SPI module in Slave mode is controlled by the TXR bit of SPIxCON2, the TRIS bit associated with the SDO pin, the Slave Select input, and the current state of the TXFIFO. This control is summarized in Table 32-2. In this table, TRISxn refers to the bit in the TRIS register corresponding to the pin that SDO has been assigned with PPS, TXR is the Transmit Data Required Control bit of SPIxCON2, SS is the state of the Slave Select input, and TXBE is the TXFIFO Buffer Empty bit of SPIxSTATUS.

32.6.1.1 SDO Drive/Tri-state

The TRIS bit associated with the SDO pin controls whether the SDO pin will tri-state. When this TRIS bit is cleared, the pin will always be driving to a level, even when the SPI module is inactive. When the SPI module is inactive (either due to the master not clocking the SCK line or the SS being false), the SDO pin will be driven to the value of the LAT bit associated with the

SDO pin. When the SPI module is active, its output is determined by both TXR and whether there is data in the TXFIFO.

When the TRIS bit associated with the SDO pin is set, the pin will only have an output level driven to it when TXR = 1 and the slave select input is true. In all other cases, the pin will be tri-stated.

32.6.1.2 SDO Output Data

The TXR bit controls the nature of the data that is transmitted in Slave mode. When TXR is set, transmitted data is taken from the TXFIFO. If the FIFO is empty, the most recently received data will be transmitted and the TXUIF flag will be set to indicate that a transmit FIFO underflow has occurred.

When TXR is cleared, the data will be taken from the TXFIFO, and the TXFIFO occupancy will not decrease. If the TXFIFO is empty, the most recently received data will be transmitted, and the TXUIF bit will not be set. However, if the TRIS bit associated with the SDO pin is set, clearing the TXR bit will cause the SPI module to not output any data to the SDO pin.

TABLE 32-2: SLAVE MODE TRANSMIT

TRISxn ⁽¹⁾	TXR	SS	TXBE	SDO State
0	0	FALSE	0	Drives state determined by LATxn(2)
0	0	FALSE	1	Drives state determined by LATxn(2)
0	0	TRUE	0	Outputs the oldest byte in the TXFIFO Does not remove data from the TXFIFO
0	0	TRUE	1	Outputs the most recently received byte
0	1	FALSE	0	Drives state determined by LATxn(2)
0	1	FALSE	1	Drives state determined by LATxn(2)
0	1	TRUE	0	Outputs the oldest byte in the TXFIFO Removes transmitted byte from the TXFIFO Decrements occupancy of TXFIFO
0	1	TRUE	1	Outputs the most recently received byte Sets the TXUIF bit of SPIxINTF
1	0	FALSE	0	Tri-stated
1	0	FALSE	1	Tri-stated
1	0	TRUE	0	Tri-stated
1	0	TRUE	1	Tri-stated
1	1	FALSE	0	Tri-stated
1	1	FALSE	1	Tri-stated
1	1	TRUE	0	Outputs the oldest byte in the TXFIFO Removes transmitted byte from the TXFIFO Decrements occupancy of TXFIFO
1	1	TRUE	1	Outputs the most recently received byte Sets the TXUIF bit of SPIxINTF

Note 1: TRISxn is the bit in the TRISx register corresponding to the pin that SDO has been assigned with PPS.
Note 2: LATxn is the bit in the LATx register corresponding to the pin that SDO has been assigned with PPS.

PIC18(L)F26/27/45/46/47/55/56/57K42

Clock stretching can be enabled or disabled by the clearing or setting of CSTRDIS (clock stretching disable) bit in the I2CxCON2 register. This bit is valid only in the Multi-Master and Slave modes of operation.

33.3.12.1 Clock Stretching for Buffer Operations

If enabled, clock stretching is forced during buffer read/write operations. For example, in Slave mode if RXBF = 1 (receive buffer full), the clock will be stretched after the seventh falling edge of SCL. The SCL line is released only after the user reads data from the receive buffer. This ensures that there is never a receive data overflow. In this situation, if clock stretching is disabled, the RXO bit in I2CxCON1 is set indicating a receive overflow. When set, the module will always respond with a NACK.

Similarly, when TXBE = 1 (transmit buffer empty) and I2CCNT! = 0, the clock is stretched after the 8th falling edge of SCL. The SCL line is released only after the user loads new data into the transmit buffer. This ensures that there is never a transmit underflow. In this situation, if clock stretching is disabled, the TXUF bit in I2CxCON1 is set indicating a transmit underflow. When set, the module will always respond with a NACK.

33.3.12.2 Clock Stretching for Other Slave Operations

There are three Interrupt and Hold bits that provide clock stretching in Slave mode. These bits can also be used in conjunction with the I2CxIE bit in PIRx register to generate system level interrupts.

- Incoming address match interrupt
 - Clock stretching after an incoming matching address byte is enabled by the Address Interrupt and Hold (ADRIE) bit of the I2CxPIE register. When ADRIE = 1, the CSTR bit is set and the SCL line is stretched following the 8th falling edge of SCL of a received matching address. This allows the user to read the received address from the I2CADB0/1 registers and selectively ACK/NACK based on the received address. Clock stretching from ADRIE is released by software clearing the CSTR bit.
- Data Write Interrupt
 - The data write interrupt and hold enable (WRIE) bit is used to enable clock stretching after a received data byte. When WRIE = 1, the CSTR bit is set, and the SCL line is stretched, following the 8th falling SCL edge for incoming slave data. This bit allows user software to selectively ACK/NACK each received data byte. Clock stretching from WRIE is released by software clearing the CSTR bit.

- Acknowledge status
 - The acknowledge status time interrupt and hold enable (ACKTIE) bit is used to enable clock stretching after the ACK phase of a transmission. This bit enables clock stretching for all address/data transactions; address, write, or read. Following the ACK, the slave hardware will set CSTR. Clock stretching from ACKTIE is released by software clearing the CSTR bit.

33.3.13 DATA BYTE COUNT

The I2xCNT register is used to specify the number of bytes in a complete I²C packet. The value in this register will decrement every time a data byte is received or transmitted from the I²C module. The I2xCNT register will not decrement past zero.

If a byte transfer causes the I2xCNT register to decrement to zero, the Count Interrupt Flag bit (CNTIF) in I2CxPIR is set. This flag bit is set on the 9th falling edge of SCL for transmit operations and can be more informational if the edge is specified.

The I2xCNT register can be auto-loaded if the ACNT bit in the I2CxCON2 register is set. When ACNT bit is set, the data byte following the address byte is loaded into the I2xCNT register.

Note 1: I2xCNT decrements on the eighth (receive) or ninth (transmit) falling edge of SCL; writes during this bit time can corrupt the value.

- 2:** If the block size of the message is greater than 255, the I2xCNT register can be updated mid-message to prevent decrement to zero.

33.4 I²C Slave Mode

The I²C Slave mode operates in one of four modes selected in the Mode bits of I2CxCON0. The modes can be divided into 7- and 10-bit Addressing modes. 10-bit Addressing modes operate the same as 7-bit with some additional overhead for handling the larger addresses.

33.4.1 SLAVE ADDRESSING MODES

The I2CxADR/1/2/3 registers contain the Slave mode addresses. The first byte received after a Start or Restart condition is compared against the values stored in these registers. If the byte matches a value, it is loaded into the I2CxADB0/1 registers. If the value does not match, there is no response from the module. The I²C module can be configured in the following Slave configurations.

PIC18(L)F26/27/45/46/47/55/56/57K42

39.10 Register Definitions: HLVD Control

Long bit name prefixes for the HLVD peripheral is shown in [Table 39-1](#). Refer to [Section 1.3.2.2 “Long Bit Names”](#) for more information.

TABLE 39-1:

Peripheral	Bit Name Prefix
HLVD	HLVD

REGISTER 39-1: HLVDCON0: HIGH/LOW-VOLTAGE DETECT CONTROL REGISTER 0

R/W-0/0	U-0	R-x	R-x	U-0	U-0	R/W-0/0	R/W-0/0
EN	—	OUT	RDY	—	—	INTH	INTL
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7 **EN:** High/Low-voltage Detect Power Enable bit
1 = Enables HLVD, powers up HLVD circuit and supporting reference circuitry
0 = Disables HLVD, powers down HLVD and supporting circuitry
- bit 6 **Unimplemented:** Read as '0'
- bit 5 **OUT:** HLVD Comparator Output bit
1 = Voltage \leq selected detection limit (HLVDL<3:0>)
0 = Voltage \geq selected detection limit (HLVDL<3:0>)
- bit 4 **RDY:** Band Gap Reference Voltages Stable Status Flag bit
1 = Indicates HLVD Module is ready and output is stable
0 = Indicates HLVD Module is not ready
- bit 3-2 **Unimplemented:** Read as '0'
- bit 1 **INTH:** HLVD Positive going (High Voltage) Interrupt Enable
1 = HLVDIF will be set when voltage \geq selected detection limit (SEL<3:0>)
0 = HLVDIF will not be set
- bit 0 **INTL:** HLVD Negative going (Low Voltage) Interrupt Enable
1 = HLVDIF will be set when voltage \leq selected detection limit (SEL<3:0>)
0 = HLVDIF will not be set

PIC18(L)F26/27/45/46/47/55/56/57K42

TABLE 41-2: INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected	Notes	
			MSb		LSb				
BYTE-ORIENTED FILE REGISTER INSTRUCTIONS									
ADDWF	f, d ,a	Add WREG and f	1	0010	01da	ffff	ffff	C, DC, Z, OV, N	2, 3
ADDWFC	f, d, a	Add WREG and Carry bit to f	1	0010	00da	ffff	ffff	C, DC, Z, OV, N	
ANDWF	f, d, a	AND WREG with f	1	0001	01da	ffff	ffff	Z, N	
CLRF	f, a	Clear f	1	0110	101a	ffff	ffff	Z	
COMF	f, d, a	Complement f	1	0001	11da	ffff	ffff	Z, N	
DECf	f, d, a	Decrement f	1	0000	01da	ffff	ffff	C, DC, Z, OV, N	
INCF	f, d, a	Increment f	1	0010	10da	ffff	ffff	C, DC, Z, OV, N	
IORWF	f, d, a	Inclusive OR WREG with f	1	0001	00da	ffff	ffff	Z, N	
MOVF	f, d, a	Move f to WREG or f	1	0101	00da	ffff	ffff	Z, N	
MOVFF	f _s , f _d	Move f _s (source) to 1st word f _d (destination) 2nd word	2	1100	ffff	ffff	ffff	None	
MOVFFL	f _s , f _d	Move f _s (source) to g (full destination) f _d (full destination)3rd word	3	0000	0000	0110	ffff	None	
				1111	ffff	ffff	ffgg		
				1111	gggg	gggg	gggg		
MOVWF	f, a	Move WREG to f	1	0110	111a	ffff	ffff	None	
MULWF	f, a	Multiply WREG with f	1	0000	001a	ffff	ffff	None	
NEGF	f, a	Negate f	1	0110	110a	ffff	ffff	C, DC, Z, OV, N	
RLCF	f, d, a	Rotate Left f through Carry	1	0011	01da	ffff	ffff	C, Z, N	
RLNCF	f, d, a	Rotate Left f (No Carry)	1	0100	01da	ffff	ffff	Z, N	
RRCF	f, d, a	Rotate Right f through Carry	1	0011	00da	ffff	ffff	C, Z, N	
RRNCF	f, d, a	Rotate Right f (No Carry)	1	0100	00da	ffff	ffff	Z, N	
SETF	f, a	Set f	1	0110	100a	ffff	ffff	None	
SUBFWB	f, d, a	Subtract f from WREG with borrow	1	0101	01da	ffff	ffff	C, DC, Z, OV, N	
SUBWF	f, d, a	Subtract WREG from f	1	0101	11da	ffff	ffff	C, DC, Z, OV, N	
SUBWFB	f, d, a	Subtract WREG from f with borrow	1	0101	10da	ffff	ffff	C, DC, Z, OV, N	
SWAPF	f, d, a	Swap nibbles in f	1	0011	10da	ffff	ffff	None	
XORWF	f, d, a	Exclusive OR WREG with f	1	0001	10da	ffff	ffff	Z, N	
BYTE-ORIENTED SKIP INSTRUCTIONS									
CPFSEQ	f, a	Compare f with WREG, skip =	1 (2 or 3)	0110	001a	ffff	ffff	None	1
CPFSGT	f, a	Compare f with WREG, skip >	1 (2 or 3)	0110	010a	ffff	ffff	None	1
CPFSLT	f, a	Compare f with WREG, skip <	1 (2 or 3)	0110	000a	ffff	ffff	None	1
DECFSZ	f, d, a	Decrement f, Skip if 0	1 (2 or 3)	0010	11da	ffff	ffff	None	1
DCFSNZ	f, d, a	Decrement f, Skip if Not 0	1 (2 or 3)	0100	11da	ffff	ffff	None	1
INCFSZ	f, d, a	Increment f, Skip if 0	1 (2 or 3)	0011	11da	ffff	ffff	None	1
INFSNZ	f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	0100	10da	ffff	ffff	None	1
TSTFSZ	f, a	Test f, skip if 0	1 (2 or 3)	0110	011a	ffff	ffff	None	1
BIT-ORIENTED FILE REGISTER INSTRUCTIONS									
BCF	f, b, a	Bit Clear f	1	1001	bbba	ffff	ffff	None	
BSF	f, b, a	Bit Set f	1	1000	bbba	ffff	ffff	None	
BTG	f, d, a	Bit Toggle f	1	0111	bbba	ffff	ffff	None	
BIT-ORIENTED SKIP INSTRUCTIONS									
BTFSC	f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	1011	bbba	ffff	ffff	None	1
BTFSS	f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	1010	bbba	ffff	ffff	None	1

- Note 1:** If Program Counter (PC) is modified or a conditional test is true, the instruction requires an additional cycle. The extra cycle is executed as a NOP.
- 2:** Some instructions are multi word instructions. The second/third words of these instructions will be decoded as a NOP, unless the first word of the instruction retrieves the information embedded in these 16-bits. This ensures that all program memory locations have a valid instruction.
- 3:** f_s and f_d do not cover the full memory range. 2 MSBs of bank selection are forced to 'b00 to limit the range of these instructions to lower 4k addressing space.

PIC18(L)F26/27/45/46/47/55/56/57K42

TABLE 44-8: THERMAL CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)

Param No.	Sym.	Characteristic	Typ.	Units	Conditions
TH01	θ_{JA}	Thermal Resistance Junction to Ambient	60	°C/W	28-pin SPDIP package
			80	°C/W	28-pin SOIC package
			90	°C/W	28-pin SSOP package
			27.5	°C/W	28-pin UQFN 4x4 mm package
			27.5	°C/W	28-pin QFN 6x6mm package
			47.2	°C/W	40-pin PDIP package
			46	°C/W	44-pin TQFP package
			24.4	°C/W	44-pin QFN 8x8mm package
TH02	θ_{JC}	Thermal Resistance Junction to Case	31.4	°C/W	28-pin SPDIP package
			24	°C/W	28-pin SOIC package
			24	°C/W	28-pin SSOP package
			24	°C/W	28-pin UQFN 4x4mm package
			24	°C/W	28-pin QFN 6x6mm package
			24.7	°C/W	40-pin PDIP package
			14.5	°C/W	44-pin TQFP package
			20	°C/W	44-pin QFN 8x8mm package
TH03	T _{JMAX}	Maximum Junction Temperature	150	°C	
TH04	PD	Power Dissipation	—	W	$PD = P_{INTERNAL} + P_{I/O}$ ⁽³⁾
TH05	P _{INTERNAL}	Internal Power Dissipation	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}$ ⁽¹⁾
TH06	P _{I/O}	I/O Power Dissipation	—	W	$P_{I/O} = \sum (I_{OL} \times V_{OL}) + \sum (I_{OH} \times (V_{DD} - V_{OH}))$
TH07	P _{DER}	Derated Power	—	W	$P_{DER} = PD_{MAX} (T_J - T_A) / \theta_{JA}$ ⁽²⁾

Note 1: I_{DD} is current to run the chip alone without driving any load on the output pins.

2: T_A = Ambient Temperature, T_J = Junction Temperature

3: See absolute maximum ratings for total power dissipation.

PIC18(L)F26/27/45/46/47/55/56/57K42

TABLE 44-10: 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 48 64	—	MHz	(Note 2)
OS51	FHFOSCLP	Low-Power Optimized HFINTOSC Frequency	0.93 1.86	1 2	1.07 2.14	MHz MHz	
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	

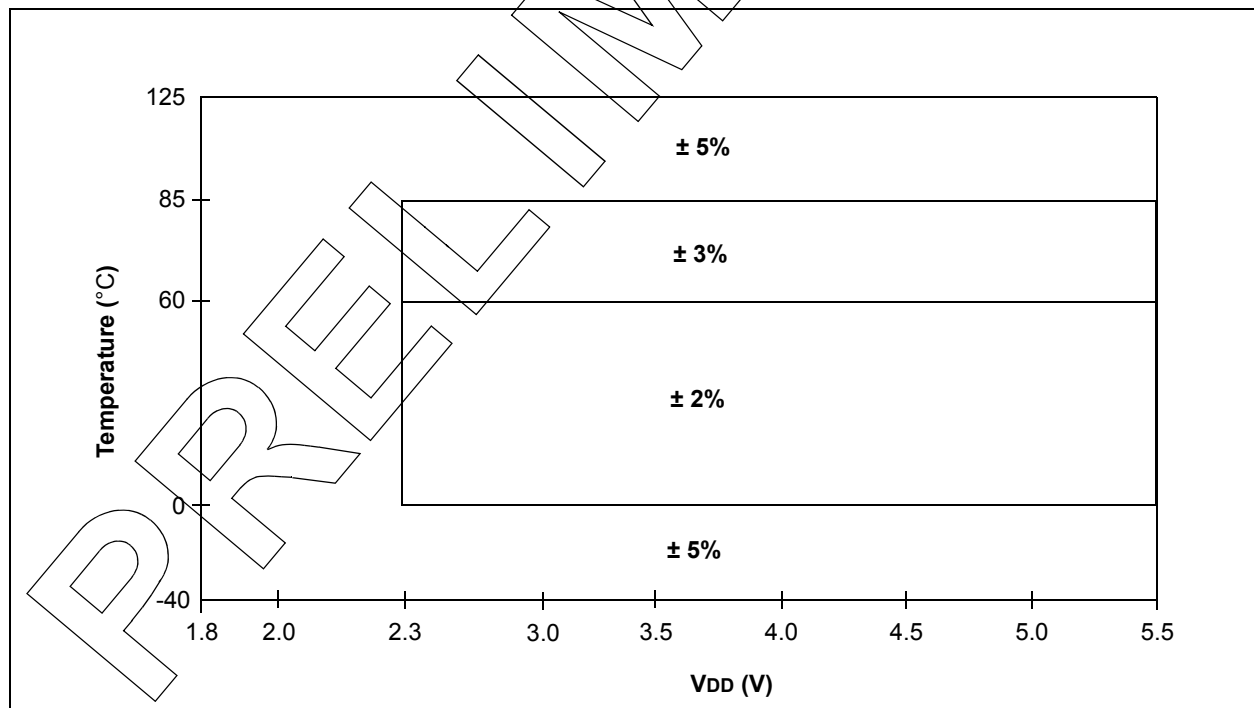
* These parameters are characterized but not tested.

† 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 44-6: Precision Calibrated HFINTOSC Frequency Accuracy Over Device VDD and Temperature](#).

FIGURE 44-6: PRECISION CALIBRATED HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE



THE MICROCHIP WEBSITE

Microchip provides online support via our website at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at: <http://www.microchip.com/support>