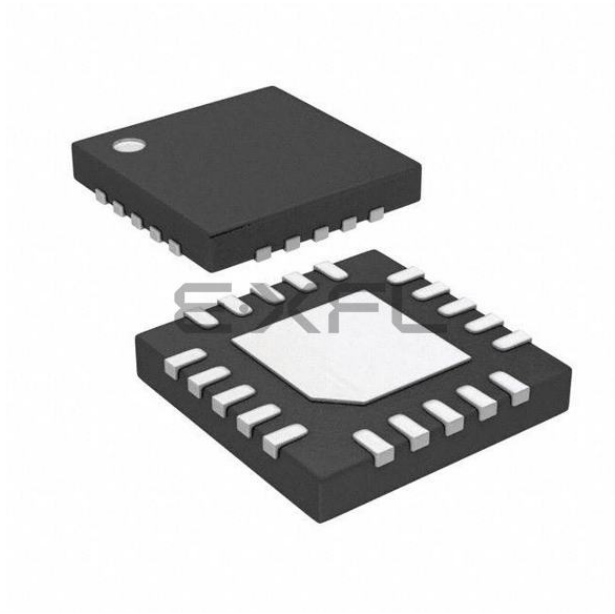


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 17x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-UFQFN Exposed Pad
Supplier Device Package	20-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f15345t-i-gz

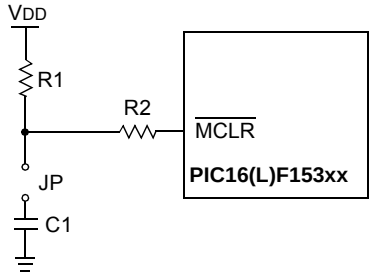
2.3 Master Clear (MCLR) Pin

The MCLR pin provides two specific device functions: Device Reset, and Device Programming and Debugging. If programming and debugging are not required in the end application, a direct connection to VDD may be all that is required. The addition of other components, to help increase the application's resistance to spurious Resets from voltage sags, may be beneficial. A typical configuration is shown in Figure 2-1. Other circuit designs may be implemented, depending on the application's requirements.

During programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the MCLR pin. Consequently, specific voltage levels (VIH and VIL) and fast signal transitions must not be adversely affected. Therefore, specific values of R1 and C1 will need to be adjusted based on the application and PCB requirements. For example, it is recommended that the capacitor, C1, be isolated from the MCLR pin during programming and debugging operations by using a jumper (Figure 2-2). The jumper is replaced for normal run-time operations.

Any components associated with the MCLR pin should be placed within 0.25 inch (6 mm) of the pin.

FIGURE 2-2: EXAMPLE OF MCLR PIN CONNECTIONS



Note 1: R1 ≤ 10 kΩ is recommended. A suggested starting value is 10 kΩ. Ensure that the MCLR pin VIH and VIL specifications are met.

2: R2 ≤ 470Ω will limit any current flowing into MCLR from the external capacitor, C1, in the event of MCLR pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS). Ensure that the MCLR pin VIH and VIL specifications are met.

2.4 ICSP™ Pins

The PGC and PGD pins are used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes. It is recommended to keep the trace length between the ICSP connector and the ICSP pins on the device as short as possible. If the ICSP connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of ohms, not to exceed 100Ω.

Pull-up resistors, series diodes and capacitors on the PGC and PGD pins are not recommended as they will interfere with the programmer/debugger communications to the device. If such discrete components are an application requirement, they should be removed from the circuit during programming and debugging. Alternatively, refer to the AC/DC characteristics and timing requirements information in the respective device Flash programming specification for information on capacitive loading limits, and pin input voltage high (VIH) and input low (VIL) requirements.

For device emulation, ensure that the "Communication Channel Select" (i.e., PGCx/PGDx pins), programmed into the device, matches the physical connections for the ICSP to the Microchip debugger/emulator tool.

For more information on available Microchip development tools connection requirements, refer to **Section 39.0 "Development Support"**.

TABLE 4-10: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR
Bank 7											
CPU CORE REGISTERS; see Table 4-3 for specifics											
38Ch	PWM6DCL	DC<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
38Dh	PWM6DCH	DC<9:0>								xxxx xxxx	uuuu uuuu
38Eh	PWM6CON	EN	—	OUT	POL	—	—	—	—	0-00 ----	0-00 ----
38Fh — 39Fh	—	Unimplemented								—	—

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

TABLE 4-10: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR
Bank 61 (Continued)											
1EC5h	SSP1CLKPPS	—	—	SSP1CLKPPS<5:0>						--01 0011	--uu uuuu
1EC6h	SSP1DATPPS	—	—	SSP1DATPPS<5:0>						--01 0100	--uu uuuu
1EC7h	SSP1SSPPS	—	—	SSP1SSPPS<5:0>						--00 0101	--uu uuuu
1EC8h — 1ECAh	—	Unimplemented								—	—
1ECBh	RX1DTPPS	—	—	RX1DTPPS<5:0>						--01 0111	--uu uuuu
1ECCh	TX1CKPPS	—	—	TX1CKPPS<5:0>						--01 0110	--uu uuuu
1ECDh	RX2DTPPS	—	—	RX2DTPPS<5:0>						--00 1111	--uu uuuu
1ECEh	TX2CKPPS	—	—	TX2CKPPS<5:0>						--00 1110	--uu uuuu
1ECFh — 1EEFh	—	Unimplemented								—	—

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

FIGURE 4-5: ACCESSING THE STACK EXAMPLE 2

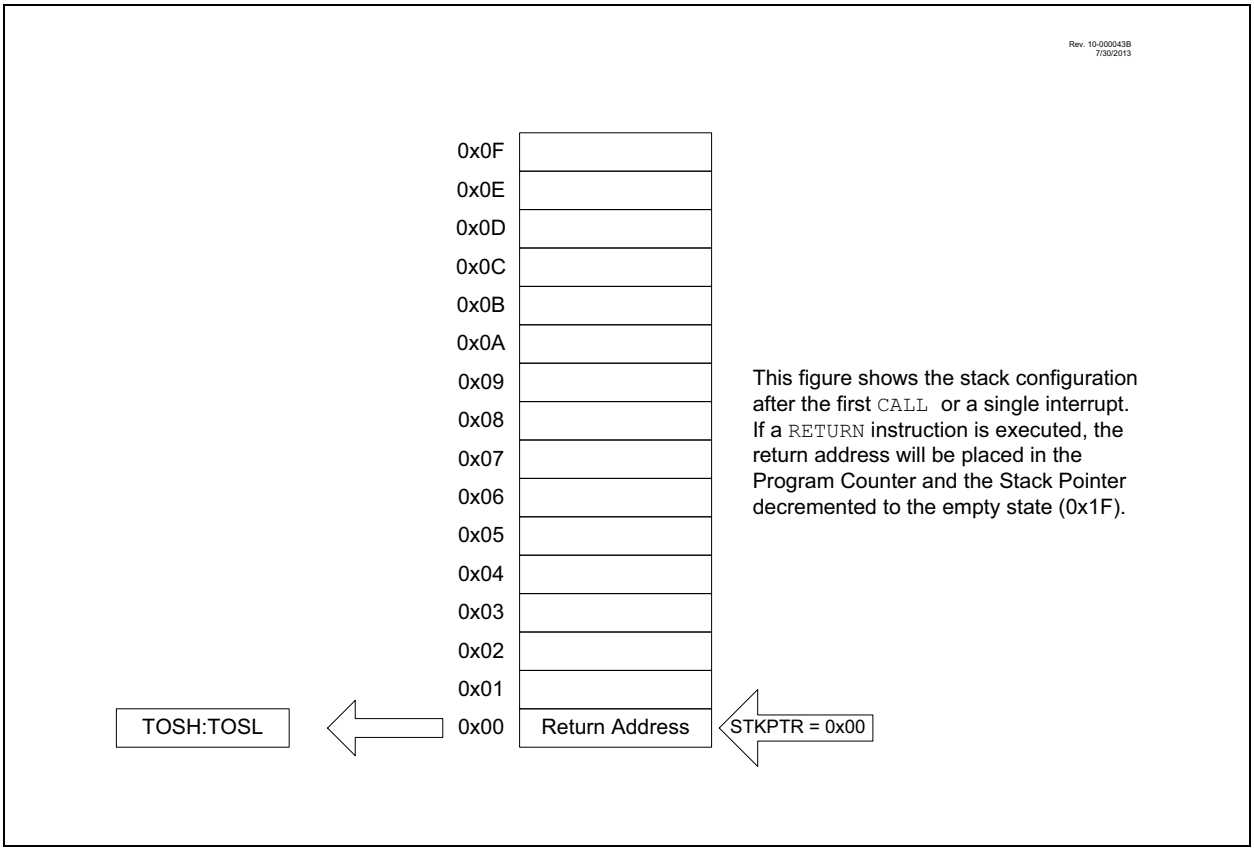
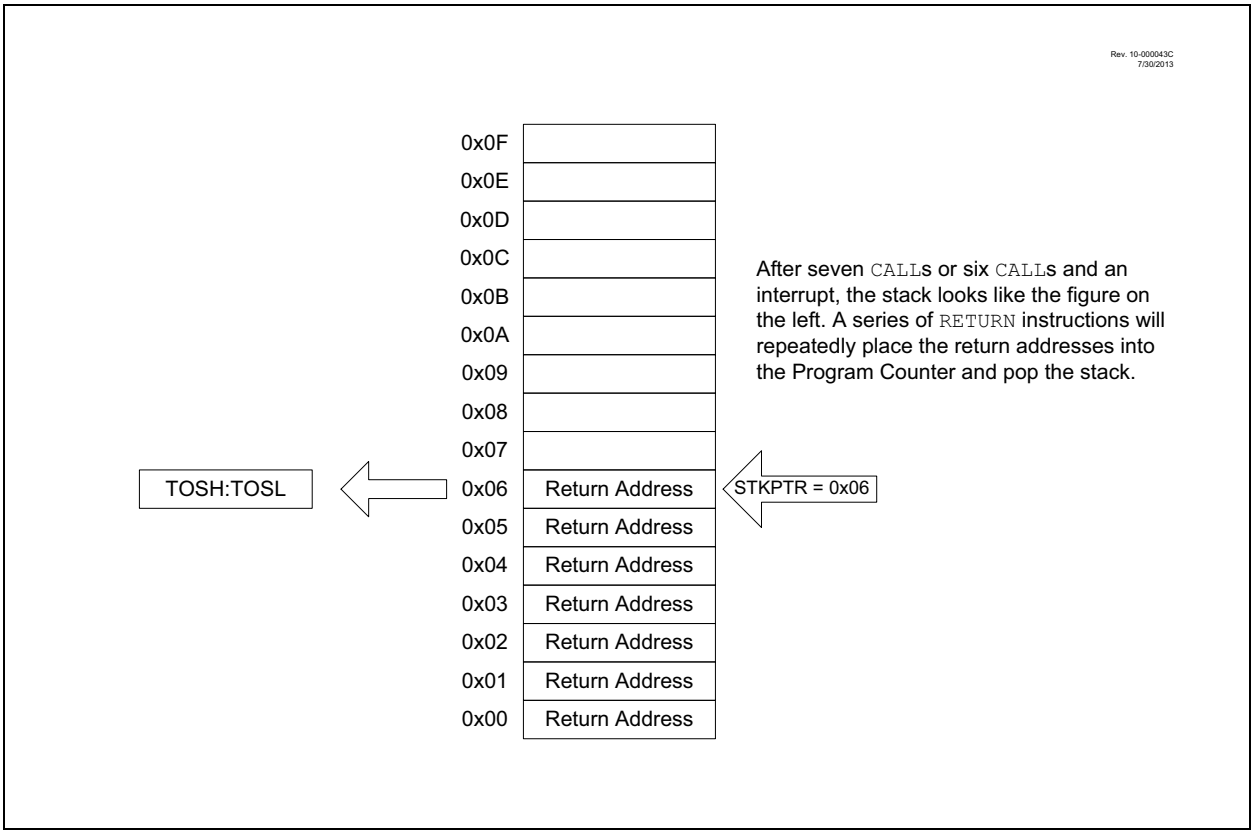


FIGURE 4-6: ACCESSING THE STACK EXAMPLE 3



12.7 Register Definitions: Windowed Watchdog Timer Control

REGISTER 12-1: WDTCON0: WATCHDOG TIMER CONTROL REGISTER 0

U-0	U-0	R/W(3)-q/q(2)	R/W(3)-q/q(2)	R/W(3)-q/q(2)	R/W(3)-q/q(2)	R/W(3)-q/q(2)	R/W-0/0
—	—	WDTPS<4:0>(1)					SWDTEN
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 7-6 **Unimplemented:** Read as '0'

bit 5-1 **WDTPS<4:0>:** Watchdog Timer Prescale Select bits(1)

Bit Value = Prescale Rate

11111 = Reserved. Results in minimum interval (1:32)
•
•
•
10011 = Reserved. Results in minimum interval (1:32)

10010 = 1:8388608 (2^{23}) (Interval 256s nominal)
10001 = 1:4194304 (2^{22}) (Interval 128s nominal)
10000 = 1:2097152 (2^{21}) (Interval 64s nominal)
01111 = 1:1048576 (2^{20}) (Interval 32s nominal)
01110 = 1:524288 (2^{19}) (Interval 16s nominal)
01101 = 1:262144 (2^{18}) (Interval 8s nominal)
01100 = 1:131072 (2^{17}) (Interval 4s nominal)
01011 = 1:65536 (Interval 2s nominal) (Reset value)
01010 = 1:32768 (Interval 1s nominal)
01001 = 1:16384 (Interval 512 ms nominal)
01000 = 1:8192 (Interval 256 ms nominal)
00111 = 1:4096 (Interval 128 ms nominal)
00110 = 1:2048 (Interval 64 ms nominal)
00101 = 1:1024 (Interval 32 ms nominal)
00100 = 1:512 (Interval 16 ms nominal)
00011 = 1:256 (Interval 8 ms nominal)
00010 = 1:128 (Interval 4 ms nominal)
00001 = 1:64 (Interval 2 ms nominal)
00000 = 1:32 (Interval 1 ms nominal)

bit 0 **SWDTEN:** Software Enable/Disable for Watchdog Timer bit

If WDTE<1:0> = 1x:
This bit is ignored.
If WDTE<1:0> = 01:
1 = WDT is turned on
0 = WDT is turned off
If WDTE<1:0> = 00:
This bit is ignored.

Note 1: Times are approximate. WDT time is based on 31 kHz LFINTOSC.

2: When WDTCPS <4:0> in CONFIG3 = 11111, the Reset value of WDTPS<4:0> is 01011. Otherwise, the Reset value of WDTPS<4:0> is equal to WDTCPS<4:0> in CONFIG3.

3: When WDTCPS <4:0> in CONFIG3 ≠ 11111, these bits are read-only.

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.

REGISTER 14-11: LATB: PORTB DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0
LATB7	LATB6	LATB5	LATB4	—	—	—	—
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 **LATB<7:4>**: RB<7:4> Output Latch Value bits(1)

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

Note 1: Writes to PORTB are actually written to corresponding LATB register. Reads from PORTB register returns actual I/O pin values.

REGISTER 14-12: ANSELB: PORTB ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	U-0	U-0	U-0
ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—
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 **ANSB<7:4>**: Analog Select between Analog or Digital Function on pins RB<7:4>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

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

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.

TABLE 14-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PORTC	RC7(1)	RC6(1)	RC5	RC4	RC3	RC2	RC1	RC0	189
TRISC	TRISC7(1)	TRISC6(1)	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	189
LATC	LATC7(1)	LATC6(1)	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	189
ANSELC	ANSC7(1)	ANSC6(1)	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	190
WPUC	WPUC7(1)	WPUC6(1)	WPUC5	WPUC4	WPUC3	WPUC2	WPUC1	WPUC0	190
ODCONC	ODCC7(1)	ODCC6(1)	ODCC5	ODCC4	ODCC3	ODCC2	ODCC1	ODCC0	191
SLRCONC	SLRC7(1)	SLRC6(1)	SLRC5	SLRC4	SLRC3	SLRC2	SLRC1	SLRC0	191
INLVLC	INLVLC7(1)	INLVLC6(1)	INLVLC5	INLVLC4	INLVLC3	INLVLC2	INLVLC1	INLVLC0	191

Legend: – = unimplemented locations read as '0'. Shaded cells are not used by PORTC.
Note 1: Present on PIC16(L)F15345 only.

TABLE 15-4: PPS OUTPUT SIGNAL ROUTING OPTIONS (PIC16(L)F15325)

Output Signal Name	RxyPPS Register Value	Remappable to Pins of PORTx	
		PIC16(L)F15325	
		PORTA	PORTC
CLKR	0x1B	•	•
NCO1OUT	0x1A	•	•
TMR0	0x19	•	•
SDO1/SDA1	0x16	•	•
SCK1/SCL1	0x15	•	•
C2OUT	0x14	•	•
C1OUT	0x13	•	•
DT2	0x12	•	•
TX2/CK2	0x11	•	•
DT1	0x10	•	•
TX1/CK1	0x0F	•	•
PWM6OUT	0x0E	•	•
PWM5OUT	0x0D	•	•
PWM4OUT	0x0C	•	•
PWM3OUT	0x0B	•	•
CCP2	0x0A	•	•
CCP1	0x09	•	•
CWG1D	0x08	•	•
CWG1C	0x07	•	•
CWG1B	0x06	•	•
CWG1A	0x05	•	•
CLC4OUT	0x04	•	•
CLC3OUT	0x03	•	•
CLC2OUT	0x02	•	•
CLC1OUT	0x01	•	•

TABLE 15-5: PPS OUTPUT SIGNAL ROUTING OPTIONS (PIC16(L)F15345)

Output Signal Name	RxyPPS Register Value	Remappable to Pins of PORTx		
		PIC16(L)F15345		
		PORTA	PORTB	PORTC
CLKR	0x1B	•	•	•
NCO1OUT	0x1A	•	•	•
TMR0	0x19	•	•	•
SDO1/SDA1	0x16	•	•	•
SCK1/SCL1	0x15	•	•	•
C2OUT	0x14	•	•	•
C1OUT	0x13	•	•	•
DT2	0x12	•	•	•
TX2/CK2	0x11	•	•	•
DT1	0x10	•	•	•
TX1/CK1	0x0F	•	•	•
PWM6OUT	0x0E	•	•	•
PWM5OUT	0x0D	•	•	•
PWM4OUT	0x0C	•	•	•
PWM3OUT	0x0B	•	•	•
CCP2	0x0A	•	•	•
CCP1	0x09	•	•	•
CWG1D	0x08	•	•	•
CWG1C	0x07	•	•	•
CWG1B	0x06	•	•	•
CWG1A	0x05	•	•	•
CLC4OUT	0x04	•	•	•
CLC3OUT	0x03	•	•	•
CLC2OUT	0x02	•	•	•
CLC1OUT	0x01	•	•	•

TABLE 15-6: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
PPSLOCK	—	—	—	—	—	—	—	PPSLOCKED	200
INTPPS	—	—	INTPPS<5:0>						199
T0CKIPPS	—	—	T0CKIPPS<5:0>						199
T1CKIPPS	—	—	T1CKIPPS<5:0>						199
T1GPPS	—	—	T1GPPS<5:0>						199
T2AINPPS			T2AINPPS<5:0>						199
CCP1PPS	—	—	CCP1PPS<5:0>						199
CCP2PPS	—	—	CCP2PPS<5:0>						199
CWG1PPS	—	—	CWG1PPS<5:0>						199
SSP1CLKPPS	—	—	SSP1CLKPPS<5:0>						199
SSP1DATPPS	—	—	SSP1DATPPS<5:0>						199
SSP1SSPPS	—	—	SSP1SSPPS<5:0>						199
RX1PPS	—	—	RXPPS<5:0>						200
TX1PPS	—	—	TXPPS<5:0>						199
CLCIN0PPS	—	—	CLCIN0PPS<5:0>						199
CLCIN1PPS	—	—	CLCIN1PPS<5:0>						199
CLCIN2PPS	—	—	CLCIN2PPS<5:0>						199
CLCIN3PPS	—	—	CLCIN3PPS<5:0>						199
RX2PPS	—	—	RX2PPS<5:0>						199
TX2PPS	—	—	TX2PPS<5:0>						199
ADACTPPS	—	—	ADACTPPS<5:0>						199
RA0PPS	—	—	—	RA0PPS<4:0>					200
RA1PPS	—	—	—	RA1PPS<4:0>					200
RA2PPS	—	—	—	RA2PPS<4:0>					200
RA3PPS	—	—	—	RA3PPS<4:0>					200
RA4PPS	—	—	—	RA4PPS<4:0>					200
RA5PPS	—	—	—	RA5PPS<4:0>					200
RB4PPS(1)	—	—	—	RB4PPS<4:0>					200
RB5PPS(1)	—	—	—	RB5PPS<4:0>					200
RB6PPS(1)	—	—	—	RB6PPS<4:0>					200
RB7PPS(1)	—	—	—	RB7PPS<4:0>					200
RC0PPS	—	—	—	RC0PPS<4:0>					200

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the PPS module.

Note 1: Present on PIC16(L)F15345 only.

16.0 PERIPHERAL MODULE DISABLE

The PIC16(L)F15325/45 provides the ability to disable selected modules, placing them into the lowest possible Power mode.

For legacy reasons, all modules are ON by default following any Reset.

16.1 Disabling a Module

Disabling a module has the following effects:

- All clock and control inputs to the module are suspended; there are no logic transitions, and the module will not function.
- The module is held in Reset:
  - Writing to SFRs is disabled
  - Reads return 00h

16.2 Enabling a module

When the register bit is cleared, the module is re-enabled and will be in its Reset state; SFR data will reflect the POR Reset values.

Depending on the module, it may take up to one full instruction cycle for the module to become active. There should be no interaction with the module (e.g., writing to registers) for at least one instruction after it has been re-enabled.

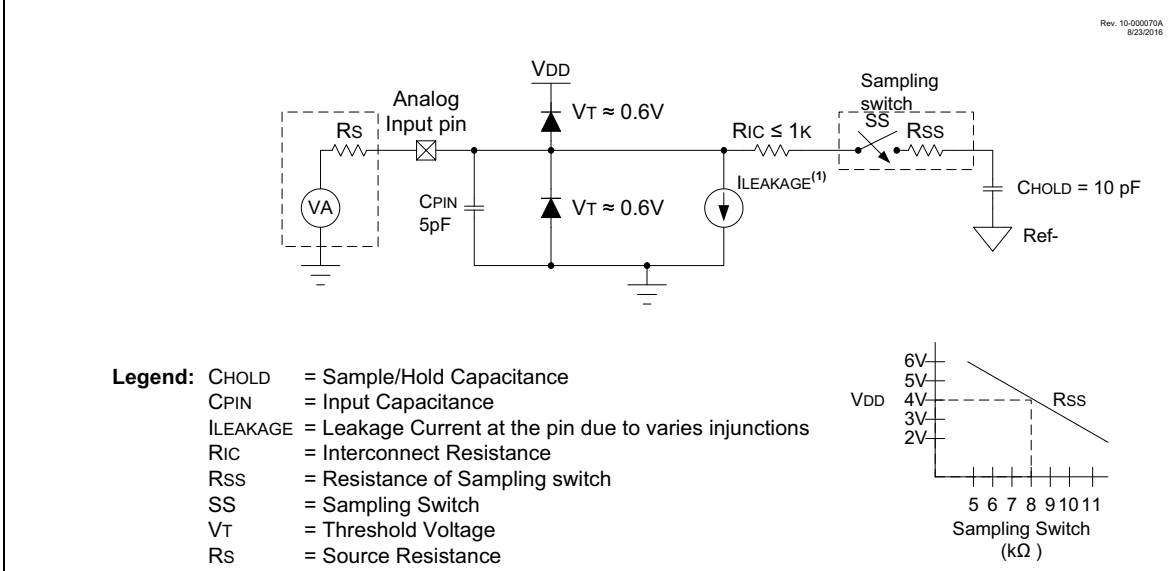
16.3 Disabling a Module

When a module is disabled, all the associated PPS selection registers (Registers xxxPPS Register 15-1, 15-2, and 15-3), are also disabled.

16.4 System Clock Disable

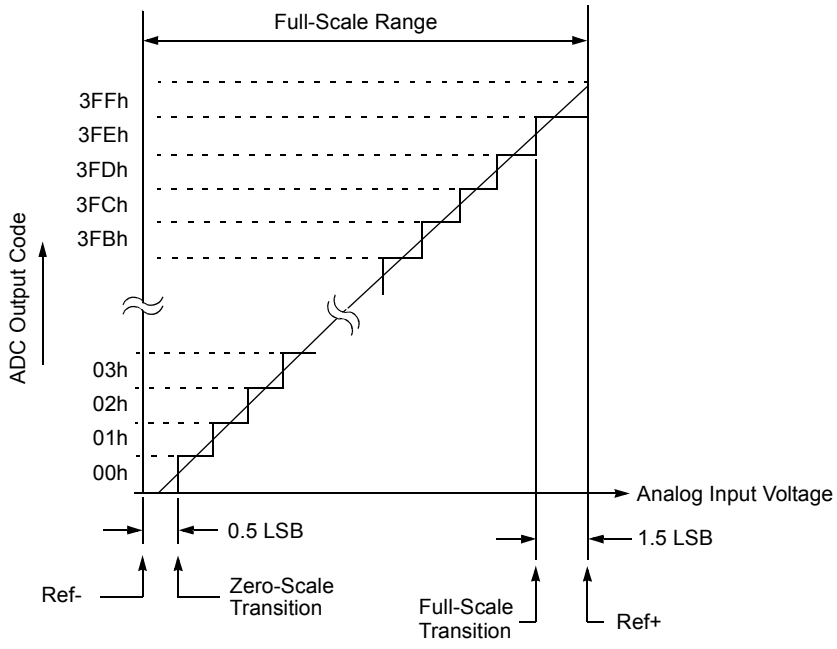
Setting SYSCMD (PMD0, Register 16-1) disables the system clock (FOSC) distribution network to the peripherals. Not all peripherals make use of SYSCLK, so not all peripherals are affected. Refer to the specific peripheral description to see if it will be affected by this bit.

FIGURE 20-4: ANALOG INPUT MODEL



Note 1: See Refer to **Section 37.0 "Electrical Specifications"**.

FIGURE 20-5: ADC TRANSFER FUNCTION



22.1 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 22-1.

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.

EQUATION 22-1: NCO OVERFLOW FREQUENCY

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

22.1.1 NCO CLOCK SOURCES

Clock sources available to the NCO include:

- HFINTOSC
- FOSC
- LC1_out
- LC2_out
- LC3_out
- LC4_out
- MFINTOSC (500 kHz)
- MFINTOSC (32 kHz)
- SOSC
- CLKR

The NCO clock source is selected by configuring the N1CKS<2:0> bits in the NCO1CLK register.

22.1.2 ACCUMULATOR

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

- NCO1ACCL
- NCO1ACCH
- NCO1ACCU

22.1.3 ADDER

The NCO Adder is a full adder, which operates synchronously 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.

22.1.4 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
- NCO1INCU

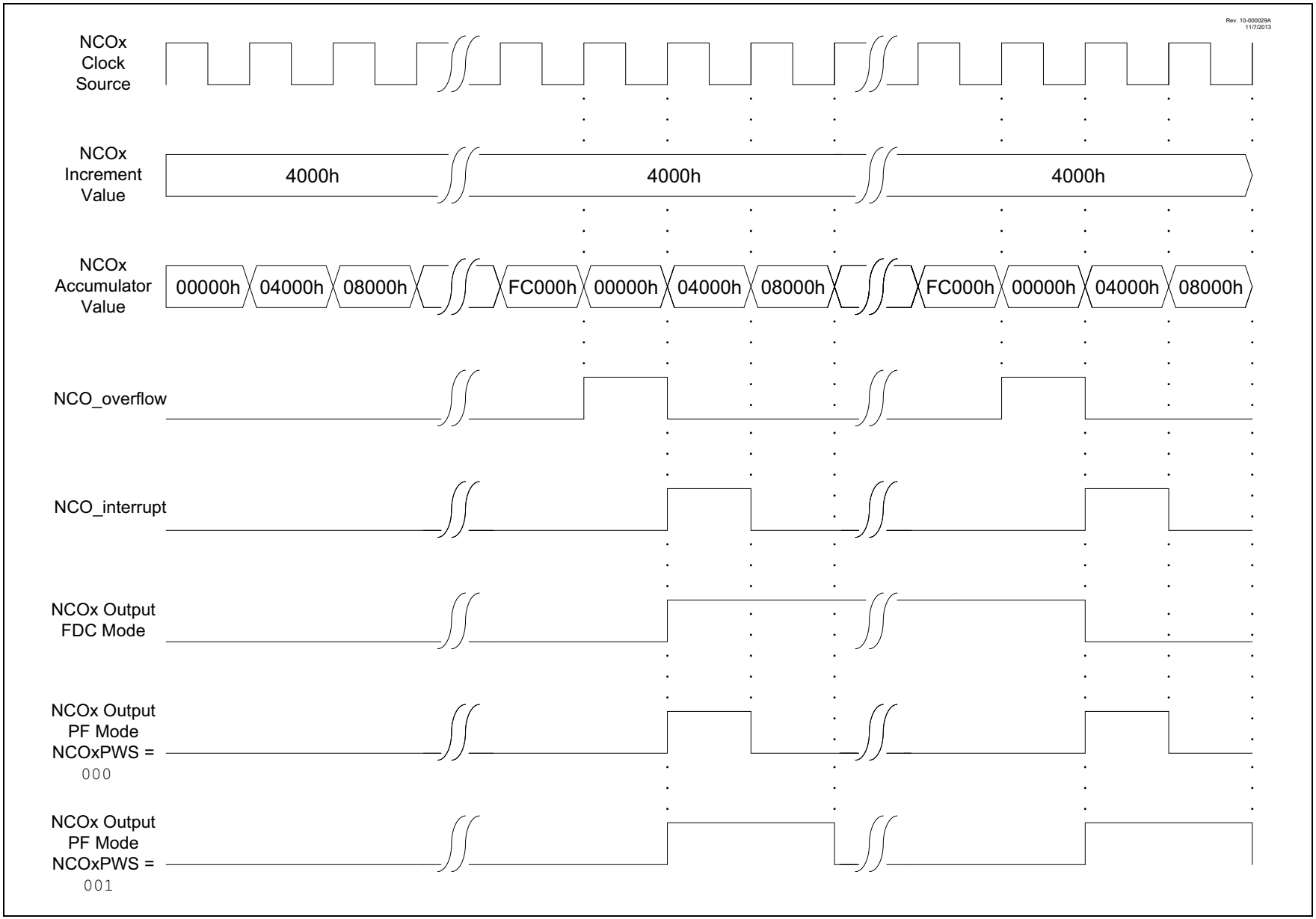
When the NCO module is enabled, the NCO1INCU and NCO1INCH registers should be written first, then the NCO1INCL register. Writing to the NCO1INCL register initiates the increment buffer registers to be loaded simultaneously on the second rising edge of the NCO_clk signal.

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 22-2: FDC OUTPUT MODE OPERATION DIAGRAM



28.2.1 CCPX PIN CONFIGURATION

The software must configure the CCPx pin as an output by clearing the associated TRIS bit and defining the appropriate output pin through the RxyPPS registers. See **Section 15.0 "Peripheral Pin Select (PPS) Module"** for more details.

The CCP output can also be used as an input for other peripherals.

Note:	Clearing the CCPxCON register will force the CCPx compare output latch to the default low level. This is not the PORT I/O data latch.

28.2.2 TIMER1 MODE RESOURCE

In Compare mode, Timer1 must be running in either Timer mode or Synchronized Counter mode. The compare operation may not work in Asynchronous Counter mode.

See **Section 26.0 "Timer1 Module with Gate Control"** for more information on configuring Timer1.

Note:	Clocking Timer1 from the system clock (FOSC) should not be used in Compare mode. In order for Compare mode to recognize the trigger event on the CCPx pin, TImer1 must be clocked from the instruction clock (FOSC/4) or from an external clock source.

28.2.3 AUTO-CONVERSION TRIGGER

All CCPx modes set the CCP interrupt flag (CCPxIF). When this flag is set and a match occurs, an Auto-conversion Trigger can take place if the CCP module is selected as the conversion trigger source.

Refer to **Section 20.2.5 "Auto-Conversion Trigger"** for more information.

Note:	Removing the match condition by changing the contents of the CCPRxH and CCPRxL register pair, between the clock edge that generates the Auto-conversion Trigger and the clock edge that generates the Timer1 Reset, will preclude the Reset from occurring

28.2.4 COMPARE DURING SLEEP

Since FOSC is shut down during Sleep mode, the Compare mode will not function properly during Sleep, unless the timer is running. The device will wake on interrupt (if enabled).

28.3 PWM Overview

Pulse-Width Modulation (PWM) is a scheme that provides power to a load by switching quickly between fully on and fully off states. The PWM signal resembles a square wave where the high portion of the signal is considered the on state and the low portion of the signal is considered the off state. The high portion, also known as the pulse width, can vary in time and is defined in steps. A larger number of steps applied, which lengthens the pulse width, also supplies more power to the load. Lowering the number of steps applied, which shortens the pulse width, supplies less power. The PWM period is defined as the duration of one complete cycle or the total amount of on and off time combined.

PWM resolution defines the maximum number of steps that can be present in a single PWM period. A higher resolution allows for more precise control of the pulse width time and in turn the power that is applied to the load.

The term duty cycle describes the proportion of the on time to the off time and is expressed in percentages, where 0% is fully off and 100% is fully on. A lower duty cycle corresponds to less power applied and a higher duty cycle corresponds to more power applied.

Figure 28-3 shows a typical waveform of the PWM signal.

28.3.1 STANDARD PWM OPERATION

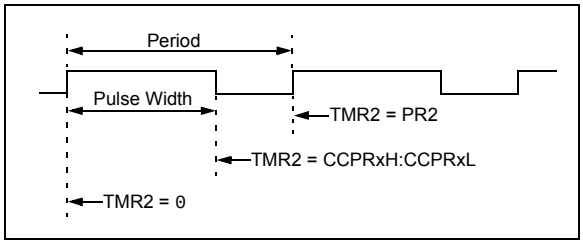
The standard PWM mode generates a Pulse-Width Modulation (PWM) signal on the CCPx pin with up to ten bits of resolution. The period, duty cycle, and resolution are controlled by the following registers:

- PR2 registers
- T2CON registers
- CCPRxL registers
- CCPxCON registers

Figure 28-4 shows a simplified block diagram of PWM operation.

Note:	The corresponding TRIS bit must be cleared to enable the PWM output on the CCPx pin.

FIGURE 28-3: CCP PWM OUTPUT SIGNAL



29.0 PULSE-WIDTH MODULATION (PWM)

The PWMx modules generate Pulse-Width Modulated (PWM) signals of varying frequency and duty cycle.

In addition to the CCP modules, the PIC16(L)F15325/45 devices contain four 10-bit PWM modules (PWM3, PWM4, PWM5 and PWM6). The PWM modules reproduce the PWM capability of the CCP modules.

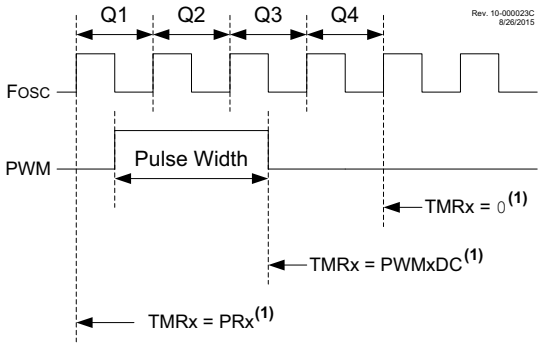
Note:	The PWM3/4/5/6 modules are four instances of the same PWM module design. Throughout this section, the lower case 'x' in register and bit names is a generic reference to the PWM module number (which should be substituted with 3, or 4, or, 5 or 6 during code development). For example, the control register is generically described in this chapter as PWMxCON, but the actual device registers are PWM3CON, PWM4CON, PWM5CON and PWM6CON. Similarly, the PWMxEN bit represents the PWM3EN, PWM4EN, PWM5EN and PWM6EN bits.

Pulse-Width Modulation (PWM) is a scheme that provides power to a load by switching quickly between fully on and fully off states. The PWM signal resembles a square wave where the high portion of the signal is considered the 'on' state (pulse width), and the low portion of the signal is considered the 'off' state. The term duty cycle describes the proportion of the 'on' time to the 'off' time and is expressed in percentages, where 0% is fully off and 100% is fully on. A lower duty cycle corresponds to less power applied and a higher duty cycle corresponds to more power applied. The PWM period is defined as the duration of one complete cycle or the total amount of on and off time combined.

PWM resolution defines the maximum number of steps that can be present in a single PWM period. A higher resolution allows for more precise control of the pulse width time and, in turn, the power that is applied to the load.

Figure 29-1 shows a typical waveform of the PWM signal.

FIGURE 29-1: PWM OUTPUT



Note 1: Timer dependent on PWMTMRS register settings.

REGISTER 30-3: CWG1DBR: CWG1 RISING DEAD-BAND COUNTER REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	DBR<5:0>					
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	q = Value depends on condition

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **DBR<5:0>:** Rising Event Dead-Band Value for Counter bits

REGISTER 30-4: CWG1DBF: CWG1 FALLING DEAD-BAND COUNTER REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	DBF<5:0>					
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	q = Value depends on condition

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **DBF<5:0>:** Falling Event Dead-Band Value for Counter bits

32.4.4 SDA HOLD TIME

The hold time of the SDA pin is selected by the SDAHT bit of the SSP1CON3 register. Hold time is the time SDA is held valid after the falling edge of SCL. Setting the SDAHT bit selects a longer 300 ns minimum hold time and may help on buses with large capacitance.

TABLE 32-1: I²C BUS TERMS

TERM	Description
Transmitter	The device which shifts data out onto the bus.
Receiver	The device which shifts data in from the bus.
Master	The device that initiates a transfer, generates clock signals and terminates a transfer.
Slave	The device addressed by the master.
Multi-master	A bus with more than one device that can initiate data transfers.
Arbitration	Procedure to ensure that only one master at a time controls the bus. Winning arbitration ensures that the message is not corrupted.
Synchronization	Procedure to synchronize the clocks of two or more devices on the bus.
Idle	No master is controlling the bus, and both SDA and SCL lines are high.
Active	Any time one or more master devices are controlling the bus.
Addressed Slave	Slave device that has received a matching address and is actively being clocked by a master.
Matching Address	Address byte that is clocked into a slave that matches the value stored in SSP1ADD.
Write Request	Slave receives a matching address with R/$\overline{W}$ bit clear, and is ready to clock in data.
Read Request	Master sends an address byte with the R/$\overline{W}$ bit set, indicating that it wishes to clock data out of the Slave. This data is the next and all following bytes until a Restart or Stop.
Clock Stretching	When a device on the bus hold SCL low to stall communication.
Bus Collision	Any time the SDA line is sampled low by the module while it is outputting and expected high state.

32.4.5 START CONDITION

The I²C specification defines a Start condition as a transition of SDA from a high to a low state while SCL line is high. A Start condition is always generated by the master and signifies the transition of the bus from an Idle to an active state. Figure 32-12 shows wave forms for Start and Stop conditions.

32.4.6 STOP CONDITION

A Stop condition is a transition of the SDA line from low-to-high state while the SCL line is high.

Note: At least one SCL low time must appear before a Stop is valid, therefore, if the SDA line goes low then high again while the SCL line stays high, only the Start condition is detected.

32.4.7 RESTART CONDITION

A Restart is valid any time that a Stop would be valid. A master can issue a Restart if it wishes to hold the bus after terminating the current transfer. A Restart has the same effect on the slave that a Start would, resetting all slave logic and preparing it to clock in an address. The master may want to address the same or another slave. Figure 32-13 shows the wave form for a Restart condition.

In 10-bit Addressing Slave mode a Restart is required for the master to clock data out of the addressed slave. Once a slave has been fully addressed, matching both high and low address bytes, the master can issue a Restart and the high address byte with the R/$\overline{W}$ bit set. The slave logic will then hold the clock and prepare to clock out data.

32.4.8 START/STOP CONDITION INTERRUPT MASKING

The SCIE and PCIE bits of the SSP1CON3 register can enable the generation of an interrupt in Slave modes that do not typically support this function. Slave modes where interrupt on Start and Stop detect are already enabled, these bits will have no effect.

34.0 REFERENCE CLOCK OUTPUT MODULE

The reference clock output module provides the ability to send a clock signal to the clock reference output pin (CLKR).

The reference clock output module has the following features:

- Selectable input clock
- Programmable clock divider
- Selectable duty cycle

34.1 CLOCK SOURCE

The reference clock output module has a selectable clock source. The CLKRCLK register (Register 34-2) controls which input is used.

34.1.1 CLOCK SYNCHRONIZATION

Once the reference clock enable (CLKREN) is set, the module is ensured to be glitch-free at start-up.

When the reference clock output is disabled, the output signal will be disabled immediately.

Clock dividers and clock duty cycles can be changed while the module is enabled, but glitches may occur on the output. To avoid possible glitches, clock dividers and clock duty cycles should be changed only when the CLKREN is clear.

34.2 PROGRAMMABLE CLOCK DIVIDER

The module takes the system clock input and divides it based on the value of the CLKRDIV<2:0> bits of the CLKRCON register (Register 34-1).

The following configurations can be made based on the CLKRDIV<2:0> bits:

- Base clock value
- Base clock value divided by 2
- Base clock value divided by 4
- Base clock value divided by 8
- Base clock value divided by 16
- Base clock value divided by 32
- Base clock value divided by 64
- Base clock value divided by 128

The clock divider values can be changed while the module is enabled; however, in order to prevent glitches on the output, the CLKRDIV<2:0> bits should only be changed when the module is disabled (CLKREN = 0).

34.3 SELECTABLE DUTY CYCLE

The CLKRDC<1:0> bits of the CLKRCON register can be used to modify the duty cycle of the output clock. A duty cycle of 25%, 50%, or 75% can be selected for all clock rates, with the exception of the undivided base FOSC value.

The duty cycle can be changed while the module is enabled; however, in order to prevent glitches on the output, the CLKRDC<1:0> bits should only be changed when the module is disabled (CLKREN = 0).

Note:	The CLKRDC1 bit is reset to '1'. This makes the default duty cycle 50% and not 0%.

34.4 OPERATION IN SLEEP MODE

The reference clock output module clock is based on the system clock. When the device goes to Sleep, the module outputs will remain in their current state. This will have a direct effect on peripherals using the reference clock output as an input signal.

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