



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	25
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 24x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UFQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f15356t-i-mv

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

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
RC2	40	ANC2	—	—	—	—	—	CCP1 ⁽¹⁾	—	—	—	—	—	—	—	IOCC2	Y	—
RC3	41	ANC3	—	—	—	—	T2IN ⁽¹⁾	—	—	—	SCL1 SCL2 ^(1,4)	—	—	—	—	IOCC3	Y	—
RC4	46	ANC4	—	—	—	—	—	—	—	—	SDA1 SDI1 ^(1,4)	—	—	—	—	IOCC4	Y	—
RC5	47	ANC5	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCC5	Y	—
RC6	48	ANC6	—	—	—	—	—	—	—	—	—	—	TX1 CK1 ⁽¹⁾	—	—	IOCC6	Y	—
RC7	1	ANC7	—	—	—	—	—	—	—	—	—	—	RX1 DT1 ⁽¹⁾	—	—	IOCC7	Y	—
RD0	42	AND0	—	—	—	—	—	—	—	—	SCK2 SCL2 ^(1,4)	—	—	—	—	—	Y	—
RD1	43	AND1	—	—	—	—	—	—	—	—	SDA2 SDI2 ^(1,4)	—	—	—	—	—	Y	—
RD2	44	AND2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RD3	45	AND3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RD4	2	AND4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RD5	3	AND5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RD6	4	AND6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RD7	5	AND7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RE0	27	ANE0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RE1	28	ANE1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RE2	29	ANE2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RE3	20	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCE3	Y	MCLR V _{PP}
RF0	36	ANF0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RF1	37	ANF1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RF2	38	ANF2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RF3	39	ANF3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RF4	12	ANF4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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 1-4: PIC16(L)F15385/86 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RF5/ANF5	RF5	TTL/ST	CMOS/OD	General purpose I/O.
	ANF5	AN	—	ADC Channel D0 input.
RF6/ANF6	RF6	TTL/ST	CMOS/OD	General purpose I/O.
	ANF6	AN	—	ADC Channel D0 input.
RF7/ANF7	RF5	TTL/ST	CMOS/OD	General purpose I/O.
	ANF5	AN	—	ADC Channel D0 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 15-5, Table 15-6 and Table 15-7.
 - 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.

The HIGH directive will set bit 7 if a label points to a location in the program memory. This applies to the assembly code Example 4-2 shown below.

EXAMPLE 4-2: ACCESSING PROGRAM MEMORY VIA FSR

```
constants
    RETLW DATA0      ;Index0 data
    RETLW DATA1      ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
    ;... LOTS OF CODE...
    MOVLW LOW constants
    MOVWF FSR1L
    MOVLW HIGH constants
    MOVWF FSR1H
    MOVIW 0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

4.2 Memory Access Partition (MAP)

User Flash is partitioned into:

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

The user can allocate the memory usage by setting the $\overline{\text{BBEN}}$ bit, selecting the size of the partition defined by $\text{BBSIZE}[2:0]$ bits and enabling the Storage Area Flash by the $\overline{\text{SAFEN}}$ bit of the Configuration Word (see Register 5-4). Refer to Table 4-2 for the different user Flash memory partitions.

4.2.1 APPLICATION BLOCK

Default settings of the Configuration bits ($\overline{\text{BBEN}} = 1$ and $\overline{\text{SAFEN}} = 1$) assign all memory in the user Flash area to the Application Block.

4.2.2 BOOT BLOCK

If $\overline{\text{BBEN}} = 1$, the Boot Block is enabled and a specific address range is allotted as the Boot Block based on the value of the BBSIZE bits of Configuration Word (Register 5-4) and the sizes provided in Table 5-1.

4.2.3 STORAGE AREA FLASH

Storage Area Flash (SAF) is enabled by clearing the $\overline{\text{SAFEN}}$ bit of the Configuration Word in Register 5-4. If enabled, the SAF block is placed at the end of memory and spans 128 words. If the Storage Area Flash (SAF) is enabled, the SAF area is not available for program execution.

4.2.4 MEMORY WRITE PROTECTION

All the memory blocks have corresponding write protection fuses WRTAPP , WRTB and WRTC bits in the Configuration Word 4 (Register 5-4). If write-protected locations are written from NVMCON registers, memory is not changed and the WRERR bit defined in Register 12-5 is set as explained in **Section 13.3.8 “WRERR Bit”**.

4.2.5 MEMORY VIOLATION

A Memory Execution Violation Reset occurs while executing an instruction that has been fetched from outside a valid execution area, clearing the $\overline{\text{MEMV}}$ bit. Refer to **Section 8.12 “Memory Execution Violation”** for the available valid program execution areas and the PCON1 register definition (Register 8-3) for $\overline{\text{MEMV}}$ bit conditions.

TABLE 4-11: 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 17												
CPU CORE REGISTERS; see Table 4-3 for specifics												
88Ch	CPUDOZE	IDLEN	DOZEN	ROI	DOE	—	DOZE2	DOZE1	DOZE0	0000 -000	u000 -000	
88Dh	OSCCON1	—	NOSC<2:0>			NDIV<3:0>				-qqq 0000	-qqq 0000	
88Eh	OSCCON2	—	COSC<2:0>			CDIV<3:0>				-qqq qqqq	-qqq qqqq	
88Fh	OSCCON3	CSWHOLD	SOSCPWR	—	ORDY	NOSCR	—	—	—	00-0 0---	00-0 0---	
890h	OSCSTAT	EXTOR	HFOR	MFOR	LFOR	SOR	ADOR	—	PLL R	q000 qq-0	qqqq qq-q	
891h	OSCEN	EXTOEN	HFOEN	MFOEN	LFOEN	SOSCEN	ADOEN	—	—	0000 00--	0000 00--	
892h	OSCTUNE	—	—	HFTUN<5:0>						--10 0000	--10 0000	
893h	OSCFRQ	—	—	—	—	—	HFFRQ<2:0>			---- -qqq	---- -qqq	
894h	—	Unimplemented								—	—	
895h	CLKRCON	CLKREN	—	—	CLKRDC<1:0>		CLKRDIV<2:0>			0--x xxxxx	0--u uuuu	
896h	CLKRCLK	—	—	—	—	CLKRCLK<3:0>					---- 0000	---- 0000
897h — 89Fh	—	Unimplemented								—	—	

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

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

REGISTER 10-7: PIE5: PERIPHERAL INTERRUPT ENABLE REGISTER 5

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
CLC4IE	CLC3IE	CLC2IE	CLC1IE	—	—	—	TMR1GIE
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

HS = Hardware set

- bit 7 **CLC4IE:** CLC4 Interrupt Enable bit
 1 = CLC4 interrupt enabled
 0 = CLC4 interrupt disabled
- bit 6 **CLC3IE:** CLC3 Interrupt Enable bit
 1 = CLC3 interrupt enabled
 0 = CLC3 interrupt disabled
- bit 5 **CLC2IE:** CLC2 Interrupt Enable bit
 1 = CLC2 interrupt enabled
 0 = CLC2 interrupt disabled
- bit 4 **CLC1IE:** CLC1 Interrupt Enable bit
 1 = CLC1 interrupt enabled
 0 = CLC1 interrupt disabled
- bit 3-1 **Unimplemented:** Read as '0'
- bit 0 **TMR1GIE:** Timer1 Gate Interrupt Enable bit
 1 = Enables the Timer1 gate acquisition interrupt
 0 = Disables the Timer1 gate acquisition interrupt

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE7.

13.2 FSR and INDF Access

The FSR and INDF registers allow indirect access to the PFM.

13.2.1 FSR READ

With the intended address loaded into an FSR register a `MOVIW` instruction or read of INDF will read data from the PFM.

Reading from NVM requires one instruction cycle. The CPU operation is suspended during the read, and resumes immediately after. Read operations return a single byte of memory.

13.2.2 FSR WRITE

Writing/erasing the NVM through the FSR registers (ex. `MOVWI` instruction) is not supported in the PIC16(L)F15356/75/76/85/86 devices.

13.3 NVMREG Access

The NVMREG interface allows read/write access to all the locations accessible by FSRs, and also read/write access to the User ID locations, and read-only access to the device identification, revision, and Configuration data.

Writing or erasing of NVM via the NVMREG interface is prevented when the device is write-protected.

13.3.1 NVMREG READ OPERATION

To read a NVM location using the NVMREG interface, the user must:

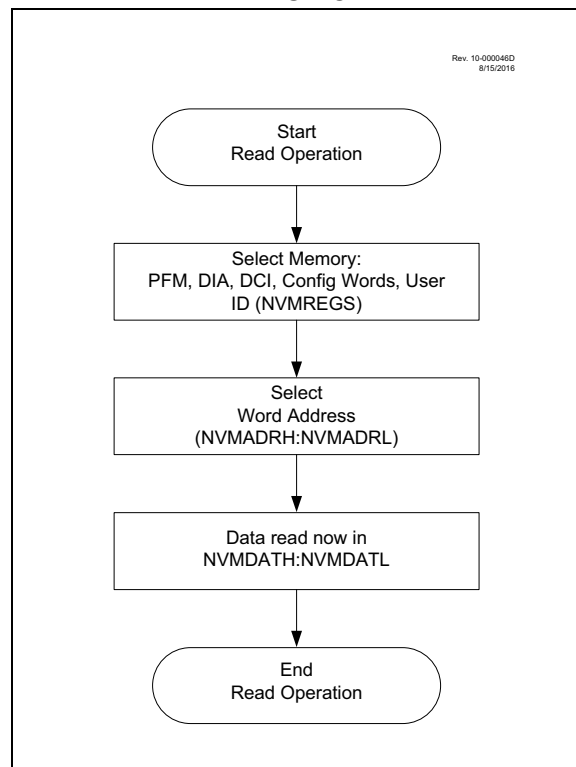
1. Clear the NVMREGS bit of the NVMCON1 register if the user intends to access PFM locations, or set NVMREGS if the user intends to access User ID, or Configuration locations.
2. Write the desired address into the NVMADRH:NVMADRL register pair (Table 13-2).
3. Set the RD bit of the NVMCON1 register to initiate the read.

Once the read control bit is set, the CPU operation is suspended during the read, and resumes immediately after. The data is available in the very next cycle, in the NVMDATH:NVMDATL register pair; therefore, it can be read as two bytes in the following instructions.

NVMDATH:NVMDATL register pair will hold this value until another read or until it is written to by the user.

Upon completion, the RD bit is cleared by hardware.

FIGURE 13-1: FLASH PROGRAM MEMORY READ FLOWCHART



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

REGISTER 14-28: ANSD: PORTD ANALOG SELECT 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
ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0
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 **ANSD<7:0>**: Analog Select between Analog or Digital Function on Pins RD<7:0>, respectively⁽¹⁾
 0 = Digital I/O. Pin is assigned to port or digital special function.
 1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital input buffer disabled.

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

REGISTER 14-29: WPUD: WEAK PULL-UP PORTD 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
WPUD7	WPUD6	WPUD5	WPUD4	WPUD3	WPUD2	WPUD1	WPUD0
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 **WPUD<7:0>**: Weak Pull-up Register bits⁽¹⁾
 1 = Pull-up enabled
 0 = Pull-up disabled

Note 1: The weak pull-up device is automatically disabled if the pin is configured as an output.

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

TABLE 20-1: ADC CLOCK PERIOD (TAD) Vs. DEVICE OPERATING FREQUENCIES

ADC Clock Period (TAD)		Device Frequency (Fosc)					
ADC Clock Source	ADCS<2:0>	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	62.5ns ⁽²⁾	100 ns ⁽²⁾	125 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	2.0 µs
Fosc/4	100	125 ns ⁽²⁾	200 ns ⁽²⁾	250 ns ⁽²⁾	500 ns ⁽²⁾	1.0 µs	4.0 µs
Fosc/8	001	0.5 µs ⁽²⁾	400 ns ⁽²⁾	0.5 µs ⁽²⁾	1.0 µs	2.0 µs	8.0 µs ⁽³⁾
Fosc/16	101	800 ns	800 ns	1.0 µs	2.0 µs	4.0 µs	16.0 µs ⁽³⁾
Fosc/32	010	1.0 µs	1.6 µs	2.0 µs	4.0 µs	8.0 µs ⁽³⁾	32.0 µs ⁽²⁾
Fosc/64	110	2.0 µs	3.2 µs	4.0 µs	8.0 µs ⁽³⁾	16.0 µs ⁽²⁾	64.0 µs ⁽²⁾
ADCRC	x11	1.0-6.0 µs ^(1,4)	1.0-6.0 µs ^(1,4)	1.0-6.0 µs ^(1,4)	1.0-6.0 µs ^(1,4)	1.0-6.0 µs ^(1,4)	1.0-6.0 µs ^(1,4)

Legend: Shaded cells are outside of recommended range.

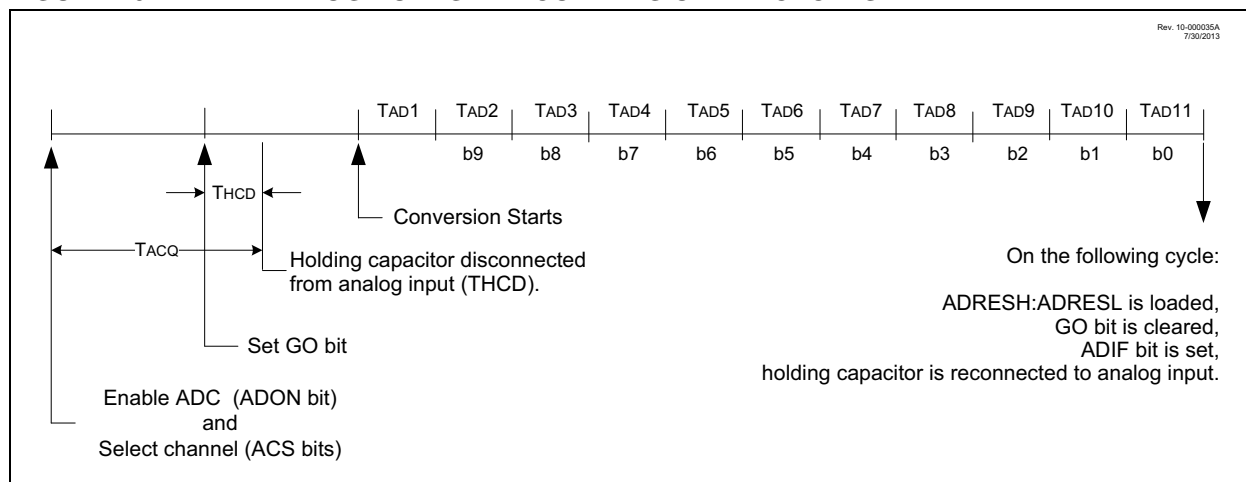
Note 1: See TAD parameter for ADCRC source typical TAD value.

2: These values violate the required TAD time.

3: Outside the recommended TAD time.

4: The ADC clock period (TAD) and total ADC conversion time can be minimized when the ADC clock is derived from the system clock Fosc. However, the ADCRC oscillator source must be used when conversions are to be performed with the device in Sleep mode.

FIGURE 20-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES



25.0 TIMER0 MODULE

The Timer0 module is an 8/16-bit timer/counter with the following features:

- 16-bit timer/counter
- 8-bit timer/counter with programmable period
- Synchronous or asynchronous operation
- Selectable clock sources
- Programmable prescaler (independent of Watchdog Timer)
- Programmable postscaler
- Operation during Sleep mode
- Interrupt on match or overflow
- Output on I/O pin (via PPS) or to other peripherals

25.1 Timer0 Operation

Timer0 can operate as either an 8-bit timer/counter or a 16-bit timer/counter. The mode is selected with the T016BIT bit of the T0CON register.

25.1.1 16-BIT MODE

In normal operation, TMR0 increments on the rising edge of the clock source. A 15-bit prescaler on the clock input gives several prescale options (see prescaler control bits, T0CKPS<3:0> in the T0CON1 register).

25.1.1.1 Timer0 Reads and Writes in 16-Bit Mode

TMR0H is not the actual high byte of Timer0 in 16-bit mode. It is actually a buffered version of the real high byte of Timer0, which is neither directly readable nor writable (see Figure 25-1). TMR0H is updated with the contents of the high byte of Timer0 during a read of TMR0L. This provides the ability to read all 16 bits of Timer0 without having to verify that the read of the high and low byte was valid, due to a rollover between successive reads of the high and low byte.

Similarly, a write to the high byte of Timer0 must also take place through the TMR0H Buffer register. The high byte is updated with the contents of TMR0H when a write occurs to TMR0L. This allows all 16 bits of Timer0 to be updated at once.

25.1.2 8-BIT MODE

In normal operation, TMR0 increments on the rising edge of the clock source. A 15-bit prescaler on the clock input gives several prescale options (see prescaler control bits, T0CKPS<3:0> in the T0CON1 register).

The value of TMR0L is compared to that of the Period buffer, a copy of TMR0H, on each clock cycle. When the two values match, the following events happen:

- TMR0_out goes high for one prescaled clock period
- TMR0L is reset
- The contents of TMR0H are copied to the period buffer

In 8-bit mode, the TMR0L and TMR0H registers are both directly readable and writable. The TMR0L register is cleared on any device Reset, while the TMR0H register initializes at FFh.

Both the prescaler and postscaler counters are cleared on the following events:

- A write to the TMR0L register
- A write to either the T0CON0 or T0CON1 registers
- Any device Reset – Power-on Reset (POR), MCLR Reset, Watchdog Timer Reset (WDTR) or Brown-out Reset (BOR)

25.1.3 COUNTER MODE

In Counter mode, the prescaler is normally disabled by setting the T0CKPS bits of the T0CON1 register to '0000'. Each rising edge of the clock input (or the output of the prescaler if the prescaler is used) increments the counter by '1'.

25.1.4 TIMER MODE

In Timer mode, the Timer0 module will increment every instruction cycle as long as there is a valid clock signal and the T0CKPS bits of the T0CON1 register (Register 25-2) are set to '0000'. When a prescaler is added, the timer will increment at the rate based on the prescaler value.

25.1.5 ASYNCHRONOUS MODE

When the T0ASYNC bit of the T0CON1 register is set (T0ASYNC = '1'), the counter increments with each rising edge of the input source (or output of the prescaler, if used). Asynchronous mode allows the counter to continue operation during Sleep mode provided that the clock also continues to operate during Sleep.

25.1.6 SYNCHRONOUS MODE

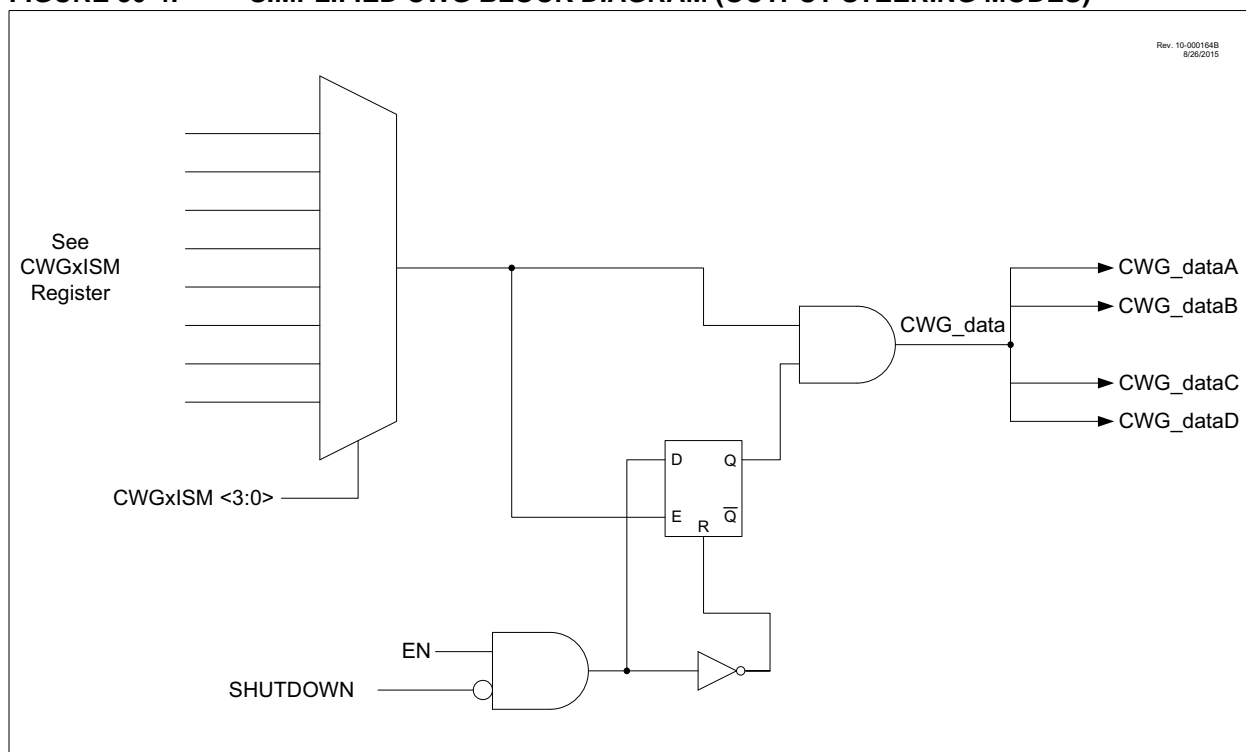
When the T0ASYNC bit of the T0CON1 register is clear (T0ASYNC = 0), the counter clock is synchronized to the system oscillator (Fosc/4). When operating in Synchronous mode, the counter clock frequency cannot exceed Fosc/4.

30.1.4 STEERING MODES

In Steering modes, the data input can be steered to any or all of the four CWG output pins. In Synchronous Steering mode, changes to steering selection registers take effect on the next rising input.

In Non-Synchronous mode, steering takes effect on the next instruction cycle. Additional details are provided in **Section 30.9 “CWG Steering Mode”**.

FIGURE 30-4: SIMPLIFIED CWG BLOCK DIAGRAM (OUTPUT STEERING MODES)



30.2 Clock Source

The CWG module allows the following clock sources to be selected:

- Fosc (system clock)
- HFINTOSC (16 MHz only)

The clock sources are selected using the CS bit of the CWG1CLKCON register.

30.9 CWG Steering Mode

In Steering mode (MODE = 00x), the CWG allows any combination of the CWG1x pins to be the modulated signal. The same signal can be simultaneously available on multiple pins, or a fixed-value output can be presented.

When the respective STRx bit of CWG1OCON0 is '0', the corresponding pin is held at the level defined. When the respective STRx bit of CWG1OCON0 is '1', the pin is driven by the input data signal. The user can assign the input data signal to one, two, three, or all four output pins.

The POLx bits of the CWG1CON1 register control the signal polarity only when STRx = 1.

The CWG auto-shutdown operation also applies in Steering modes as described in **Section 30.10 "Auto-Shutdown"**. An auto-shutdown event will only affect pins that have STRx = 1.

30.9.1 STEERING SYNCHRONIZATION

Changing the MODE bits allows for two modes of steering, synchronous and asynchronous.

When MODE = 000, the steering event is asynchronous and will happen at the end of the instruction that writes to STRx (that is, immediately). In this case, the output signal at the output pin may be an incomplete waveform. This can be useful for immediately removing a signal from the pin.

When MODE = 001, the steering update is synchronous and occurs at the beginning of the next rising edge of the input data signal. In this case, steering the output on/off will always produce a complete waveform.

Figure 30-10 and Figure 30-11 illustrate the timing of asynchronous and synchronous steering, respectively.

FIGURE 30-10: EXAMPLE OF ASYNCHRONOUS STEERING EVENT (MODE<2:0> = 000)

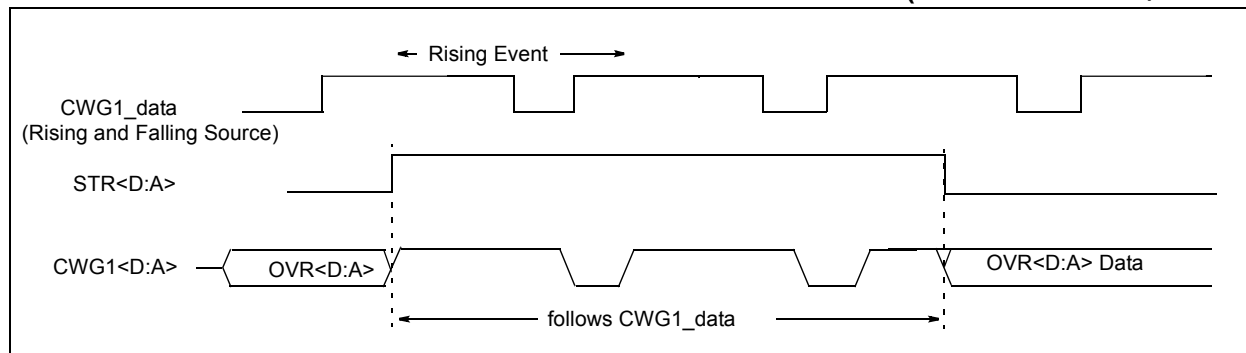
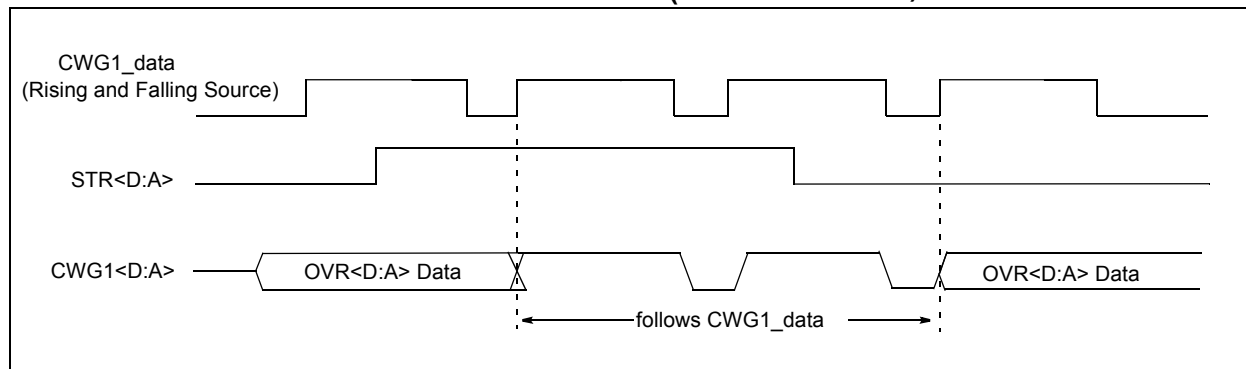


FIGURE 30-11: EXAMPLE OF STEERING EVENT (MODE<2:0> = 001)



30.13 Register Definitions: CWG Control

Long bit name prefixes for the CWG peripherals are shown in **Section 1.1 “Register and Bit Naming Conventions”**.

REGISTER 30-1: CWG1CON0: CWG1 CONTROL REGISTER 0

R/W-0/0	R/W/HC-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
EN	LD ⁽¹⁾	—	—	—	MODE<2:0>		
bit 7							bit 0

Legend:

HC = Bit is cleared by hardware

HS = Bit is set by hardware

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 **EN:** CWG1 Enable bit

1 = Module is enabled

0 = Module is disabled

bit 6 **LD:** CWG1 Load Buffer bits⁽¹⁾

1 = Buffers to be loaded on the next rising/falling event

0 = Buffers not loaded

bit 5-3 **Unimplemented:** Read as ‘0’

bit 2-0 **MODE<2:0>:** CWG1 Mode bits

111 = Reserved

110 = Reserved

101 = CWG outputs operate in Push-Pull mode

100 = CWG outputs operate in Half-Bridge mode

011 = CWG outputs operate in Reverse Full-Bridge mode

010 = CWG outputs operate in Forward Full-Bridge mode

001 = CWG outputs operate in Synchronous Steering mode

000 = CWG outputs operate in Steering mode

Note 1: This bit can only be set after EN = 1 and cannot be set in the same instruction that EN is set.

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

REGISTER 30-2: CWG1CON1: CWG1 CONTROL REGISTER 1

U-0	U-0	R-x	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IN	—	POLD	POLC	POLB	POLA
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

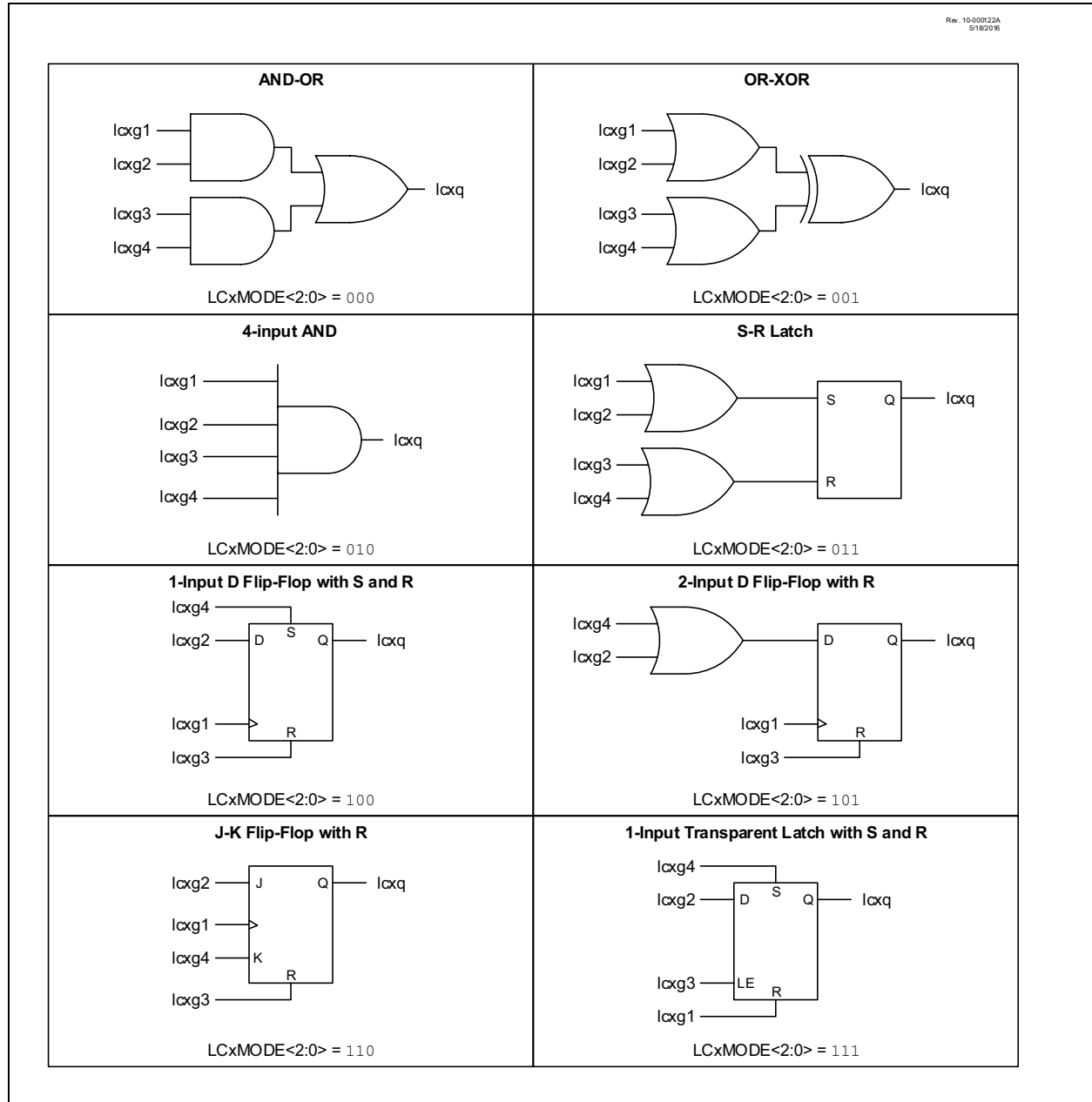
'1' = Bit is set

'0' = Bit is cleared

q = Value depends on condition

bit 7-6	Unimplemented: Read as '0'
bit 5	IN: CWG Input Value bit
bit 4	Unimplemented: Read as '0'
bit 3	POLD: CWG1D Output Polarity bit 1 = Signal output is inverted polarity 0 = Signal output is normal polarity
bit 2	POLC: CWG1C Output Polarity bit 1 = Signal output is inverted polarity 0 = Signal output is normal polarity
bit 1	POLB: CWG1B Output Polarity bit 1 = Signal output is inverted polarity 0 = Signal output is normal polarity
bit 0	POLA: CWG1A Output Polarity bit 1 = Signal output is inverted polarity 0 = Signal output is normal polarity

FIGURE 31-3: PROGRAMMABLE LOGIC FUNCTIONS



32.6.13.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

- A low level is sampled on SDA when SCL goes from low level to high level (Case 1).
- SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1' (Case 2).

When the user releases SDA and the pin is allowed to float high, the BRG is loaded with SSPxADD and counts down to zero. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 32-36). If SDA is sampled high, the BRG is reloaded and begins

counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, see Figure 32-37.

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

FIGURE 32-36: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

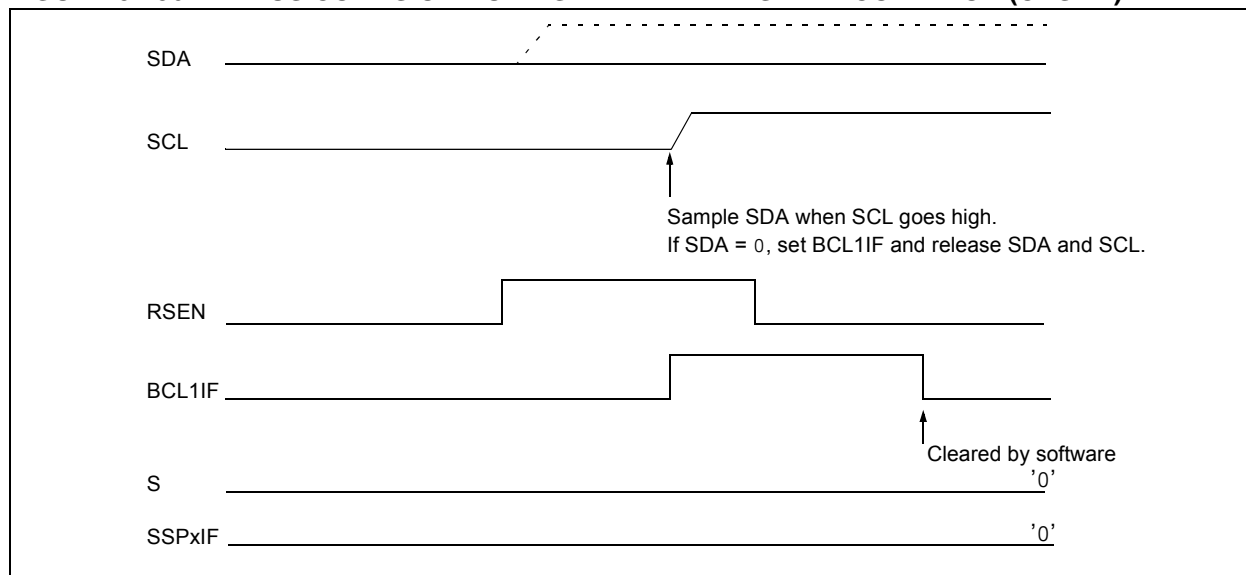
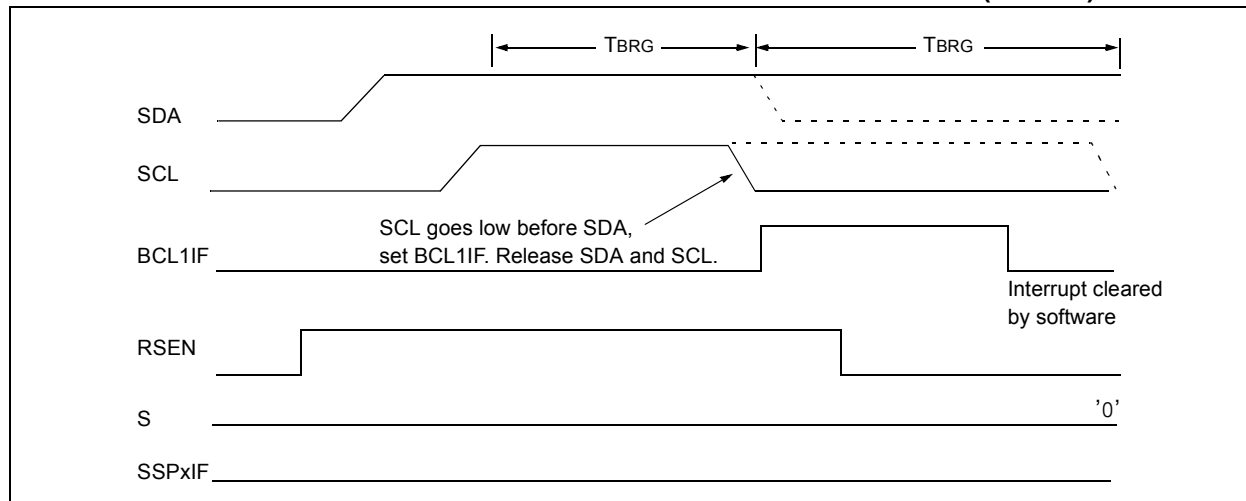


FIGURE 32-37: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



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

REGISTER 32-2: SSPxCON1: SSPx CONTROL REGISTER 1

R/C/HS-0/0	R/C/HS-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
WCOL	SSPOV ⁽¹⁾	SSPEN	CKP	SSPM<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

HS = Bit is set by hardware

C = User cleared

- bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)
1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision
- bit 6 **SSPOV:** Receive Overflow Indicator bit⁽¹⁾
In SPI mode:
1 = A new byte is received while the SSPxBUF register is still holding the previous data. In case of overflow, the data in SSPxSR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSPxBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register (must be cleared in software).
0 = No overflow
In I²C mode:
1 = A byte is received while the SSPxBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode (must be cleared in software).
0 = No overflow
- bit 5 **SSPEN:** Synchronous Serial Port Enable bit
In both modes, when enabled, the following pins must be properly configured as input or output
In SPI mode:
1 = Enables serial port and configures SCK, SDO, SDI and \overline{SS} as the source of the serial port pins⁽²⁾
0 = Disables serial port and configures these pins as I/O port pins
In I²C mode:
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins⁽³⁾
0 = Disables serial port and configures these pins as I/O port pins
- bit 4 **CKP:** Clock Polarity Select bit
In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level
In I²C Slave mode:
SCL release control
1 = Enable clock
0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
In I²C Master mode:
Unused in this mode
- bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits
1111 = I²C Slave mode, 10-bit address with Start and Stop bit interrupts enabled
1110 = I²C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1101 = Reserved
1100 = Reserved
1011 = I²C firmware controlled Master mode (slave idle)
1010 = SPI Master mode, clock = $F_{osc}/(4 * (SSPxADD+1))$ ⁽⁵⁾
1001 = Reserved
1000 = I²C Master mode, clock = $F_{osc} / (4 * (SSPxADD+1))$ ⁽⁴⁾
0111 = I²C Slave mode, 10-bit address
0110 = I²C Slave mode, 7-bit address
0101 = SPI Slave mode, clock = SCK pin, \overline{SS} pin control disabled, \overline{SS} can be used as I/O pin
0100 = SPI Slave mode, clock = SCK pin, SS pin control enabled
0011 = SPI Master mode, clock = $T2_match/2$
0010 = SPI Master mode, clock = $F_{osc}/64$
0001 = SPI Master mode, clock = $F_{osc}/16$
0000 = SPI Master mode, clock = $F_{osc}/4$

- Note**
- 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.
 - 2: When enabled, these pins must be properly configured as input or output. Use SSPxSSPPS, SSPxCLKPPS, SSPxDATPPS, and RxyPPS to select the pins.
 - 3: When enabled, the SDA and SCL pins must be configured as inputs. Use SSPxCLKPPS, SSPxDATPPS, and RxyPPS to select the pins.
 - 4: SSPxADD values of 0, 1 or 2 are not supported for I²C mode.
 - 5: SSPxADD value of '0' is not supported. Use SSPM = 0000 instead.

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

REGISTER 32-4: SSPxCON3: SSPx CONTROL REGISTER 3

R-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
ACKTIM ⁽³⁾	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
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 **ACKTIM:** Acknowledge Time Status bit (I²C mode only)⁽³⁾
1 = Indicates the I²C bus is in an Acknowledge sequence, set on 8th falling edge of SCL clock
0 = Not an Acknowledge sequence, cleared on 9th rising edge of SCL clock
- bit 6 **PCIE:** Stop Condition Interrupt Enable bit (I²C mode only)
1 = Enable interrupt on detection of Stop condition
0 = Stop detection interrupts are disabled⁽²⁾
- bit 5 **SCIE:** Start Condition Interrupt Enable bit (I²C mode only)
1 = Enable interrupt on detection of Start or Restart conditions
0 = Start detection interrupts are disabled⁽²⁾
- bit 4 **BOEN:** Buffer Overwrite Enable bit
In SPI Slave mode:⁽¹⁾
1 = SSPxBUF updates every time that a new data byte is shifted in ignoring the BF bit
0 = If new byte is received with BF bit of the SSPxSTAT register already set, SSPOV bit of the SSPxCON1 register is set, and the buffer is not updated
In I²C Master mode and SPI Master mode:
This bit is ignored.
In I²C Slave mode:
1 = SSPxBUF is updated and ACK is generated for a received address/data byte, ignoring the state of the SSPOV bit only if the BF bit = 0.
0 = SSPxBUF is only updated when SSPOV is clear
- bit 3 **SDAHT:** SDA Hold Time Selection bit (I²C mode only)
1 = Minimum of 300 ns hold time on SDA after the falling edge of SCL
0 = Minimum of 100 ns hold time on SDA after the falling edge of SCL
- bit 2 **SBCDE:** Slave Mode Bus Collision Detect Enable bit (I²C Slave mode only)
If, on the rising edge of SCL, SDA is sampled low when the module is outputting a high state, the BCL1IF bit of the PIR3 register is set, and bus goes idle
1 = Enable slave bus collision interrupts
0 = Slave bus collision interrupts are disabled
- bit 1 **AHEN:** Address Hold Enable bit (I²C Slave mode only)
1 = Following the eighth falling edge of SCL for a matching received address byte; CKP bit of the SSPxCON1 register will be cleared and the SCL will be held low.
0 = Address holding is disabled
- bit 0 **DHEN:** Data Hold Enable bit (I²C Slave mode only)
1 = Following the eighth falling edge of SCL for a received data byte; slave hardware clears the CKP bit of the SSPxCON1 register and SCL is held low.
0 = Data holding is disabled

- Note 1:** For daisy-chained SPI operation; allows the user to ignore all but the last received byte. SSPOV is still set when a new byte is received and BF = 1, but hardware continues to write the most recent byte to SSPxBUF.
- 2:** This bit has no effect in Slave modes that Start and Stop condition detection is explicitly listed as enabled.
- 3:** The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is set.

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	

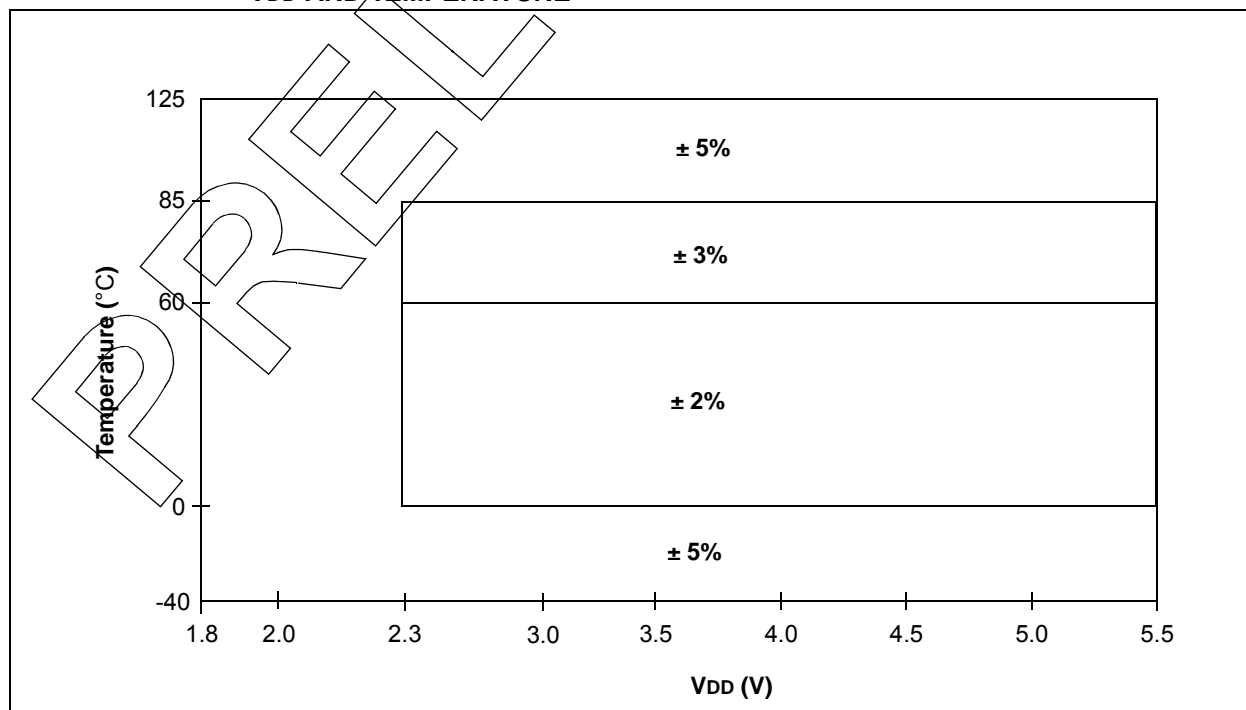
* These parameters are characterized but not tested.

† 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.

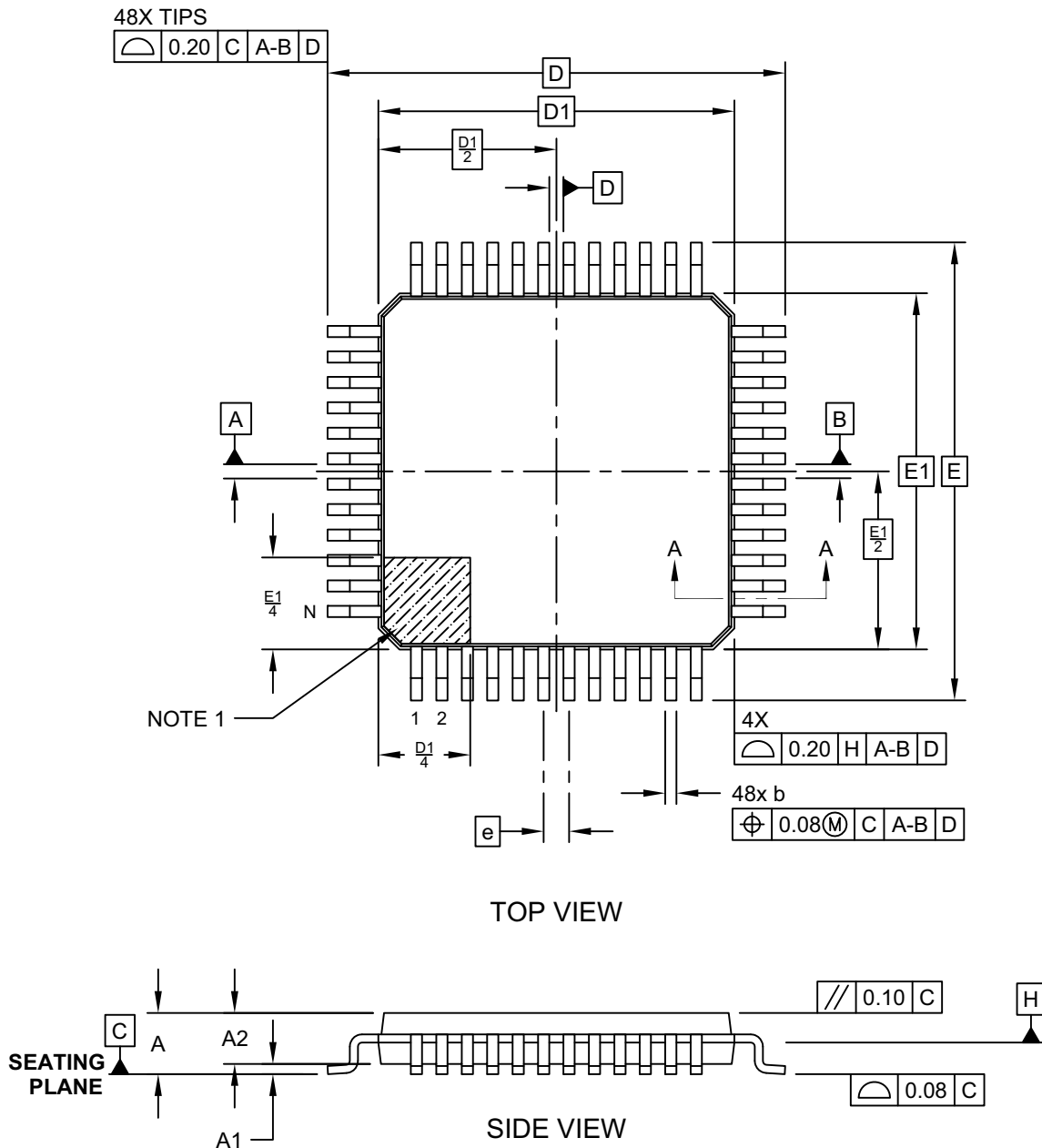
FIGURE 37-6: PRECISION CALIBRATED HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE



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

48-Lead Thin Quad Flatpack (PT) - 7x7x1.0 mm Body [TQFP]

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



Microchip Technology Drawing C04-300-PT Rev A Sheet 1 of 2