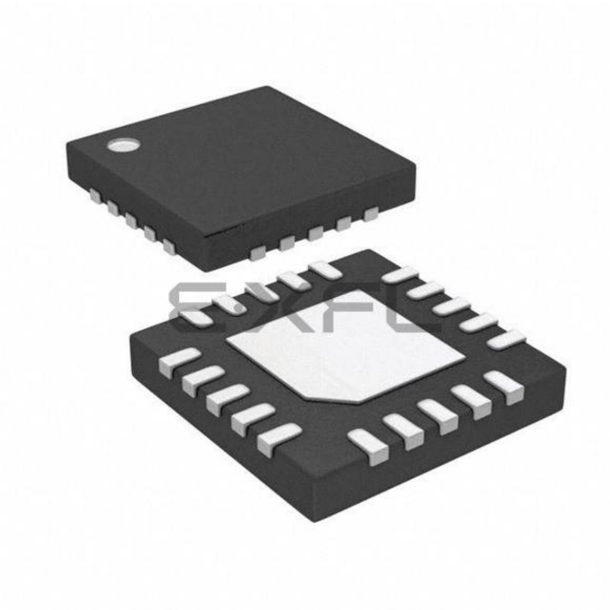


Welcome to E-XFL.COM

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	LINbus, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	12
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 12x10b; 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/pic16f1578t-i-gz

3.2.1 READING PROGRAM MEMORY AS DATA

There are two methods of accessing constants in program memory. The first method is to use tables of RETLW instructions. The second method is to set an FSR to point to the program memory.

3.2.1.1 RETLW Instruction

The RETLW instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in Example 3-1.

EXAMPLE 3-1: RETLW INSTRUCTION

```
constants
    BRW                 ;Add Index in W to
                        ;program counter to
                        ;select data
    RETLW DATA0         ;Index0 data
    RETLW DATA1         ;Index1 data
    RETLW DATA2
    RETLW DATA3


my_function
    ;… LOTS OF CODE…
    MOVLW      DATA_INDEX
    call constants
    ;… THE CONSTANT IS IN W
```

The BRW instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, then the BRW instruction is not available so the older table read method must be used.

3.2.1.2 Indirect Read with FSR

The program memory can be accessed as data by setting bit 7 of the FSRxH register and reading the matching INDFx register. The MOVIW instruction will place the lower eight bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDF registers. Instructions that access the program memory via the FSR require one extra instruction cycle to complete. Example 3-2 demonstrates accessing the program memory via an FSR.

The HIGH operator will set bit<7> if a label points to a location in program memory.

EXAMPLE 3-2: ACCESSING PROGRAM MEMORY VIA FSR

```
constants
  DW     DATA0         ;First constant
  DW     DATA1         ;Second constant
  DW     DATA2
  DW     DATA3
my_function
  ;… LOTS OF CODE…
  MOVLW  DATA_INDEX
  ADDLW  LOW constants
  MOVWF  FSR1L
  MOVLW  HIGH constants;MSb is set
                        automatically
  MOVWF  FSR1H
  BTFSC  STATUS,C      ;carry from ADDLW?
  INCF   FSR1H,f       ;yes
  MOVIW  0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

TABLE 3-8: PIC16(L)F1575/9 MEMORY MAP, BANKS 8-15

	BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15
400h–40Bh	Core Registers (Table 3-2)	480h–48Bh	Core Registers (Table 3-2)	500h–50Bh	Core Registers (Table 3-2)	580h–58Bh	Core Registers (Table 3-2)	600h–60Bh	Core Registers (Table 3-2)	680h–68Bh	Core Registers (Table 3-2)	700h–70Bh	Core Registers (Table 3-2)	780h–78Bh	Core Registers (Table 3-2)
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—
40Eh	—	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	—	70Eh	—	78Eh	—
40Fh	—	48Fh	—	50Fh	—	58Fh	—	60Fh	—	68Fh	—	70Fh	—	78Fh	—
410h	—	490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—
411h	—	491h	—	511h	—	591h	—	611h	—	691h	CWG1DBR	711h	—	791h	—
412h	—	492h	—	512h	—	592h	—	612h	—	692h	CWG1DBF	712h	—	792h	—
413h	—	493h	—	513h	—	593h	—	613h	—	693h	CWG1CON0	713h	—	793h	—
414h	—	494h	—	514h	—	594h	—	614h	—	694h	CWG1CON1	714h	—	794h	—
415h	—	495h	—	515h	—	595h	—	615h	—	695h	CWG1CON2	715h	—	795h	—
416h	—	496h	—	516h	—	596h	—	616h	—	696h	—	716h	—	796h	—
417h	—	497h	—	517h	—	597h	—	617h	—	697h	—	717h	—	797h	—
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	—	798h	—
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	—	799h	—
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	—	79Ah	—
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	—	79Bh	—
41Ch	—	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	—	79Ch	—
41Dh	—	49Dh	—	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	—	79Dh	—
41Eh	—	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—
41Fh	—	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	—	71Fh	—	79Fh	—
420h–46Fh	General Purpose Register 80 Bytes	4A0h–4EFh	General Purpose Register 80 Bytes	520h–56Fh	General Purpose Register 80 Bytes	5A0h–5EFh	General Purpose Register 80 Bytes	620h–63Fh 640h–66Fh	General Purpose Register 32 Bytes Unimplemented Read as ‘0’	6A0h–6EFh	Unimplemented Read as ‘0’	720h–76Fh	Unimplemented Read as ‘0’	7A0h–7EFh	Unimplemented Read as ‘0’
470h–47Fh	Accesses 70h – 7Fh	4F0h–4FFh	Accesses 70h – 7Fh	570h–57Fh	Accesses 70h – 7Fh	5F0h–5FFh	Accesses 70h – 7Fh	670h–67Fh	Accesses 70h – 7Fh	6F0h–6FFh	Accesses 70h – 7Fh	770h–77Fh	Accesses 70h – 7Fh	7F0h–7FFh	Accesses 70h – 7Fh

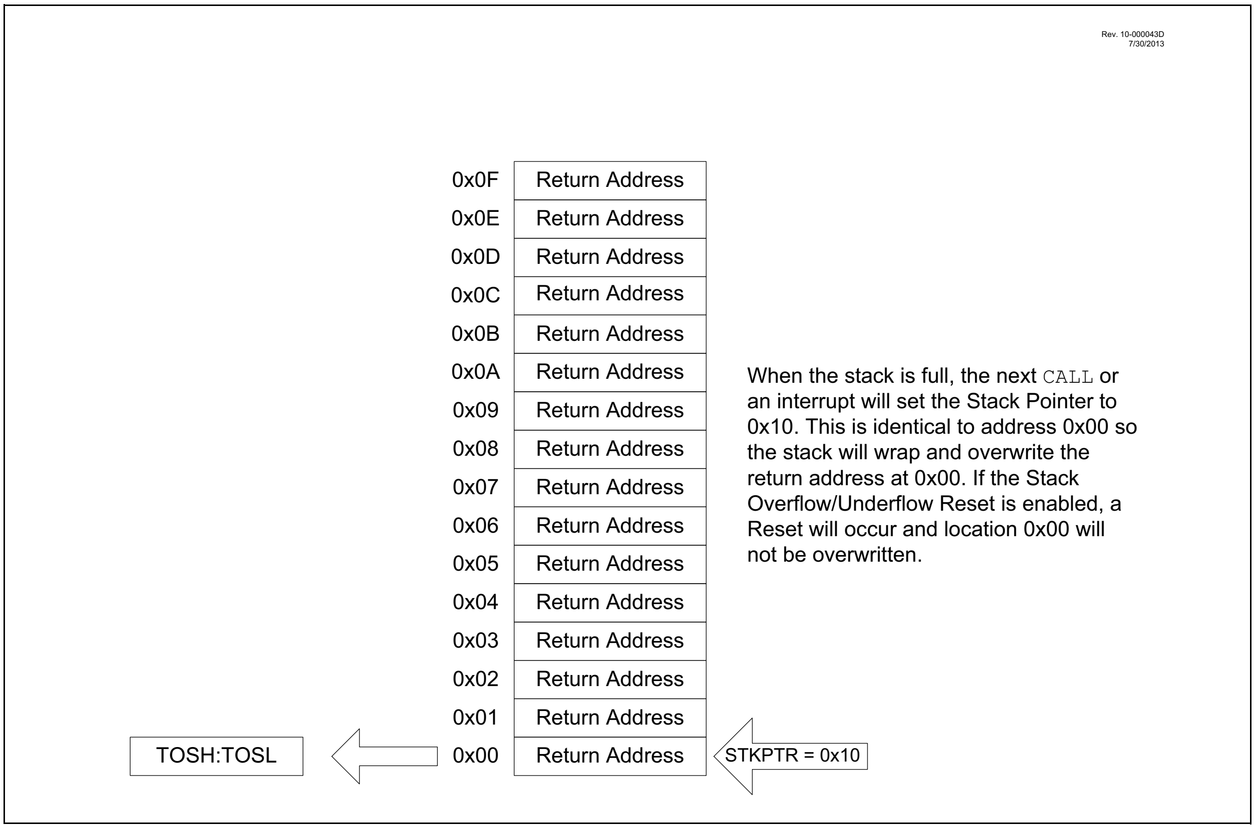
Legend: (shaded) = Unimplemented data memory locations, read as ‘0’

TABLE 3-13: PIC16(L)F1574/5/8/9 MEMORY MAP, BANK 31

Address	Bank 31
F8Ch–FE3h	Unimplemented Read as ‘0’
FE4h	STATUS_SHAD
FE5h	WREG_SHAD
FE6h	BSR_SHAD
FE7h	PCLATH_SHAD
FE8h	FSR0L_SHAD
FE9h	FSR0H_SHAD
FEAh	FSR1L_SHAD
FEBh	FSR1H_SHAD
FECh	—
FEDh	STKPTR
FEEh	TOSL
FEFh	TOSH

Legend: = Unimplemented data memory locations, read as ‘0’.

FIGURE 3-8: ACCESSING THE STACK EXAMPLE 4



3.5.2 OVERFLOW/UNDERFLOW RESET

If the STVREN bit in Configuration Words is programmed to '1', the device will be reset if the stack is PUSHed beyond the sixteenth level or POPed beyond the first level, setting the appropriate bits (STKOVF or STKUNF, respectively) in the PCON register.

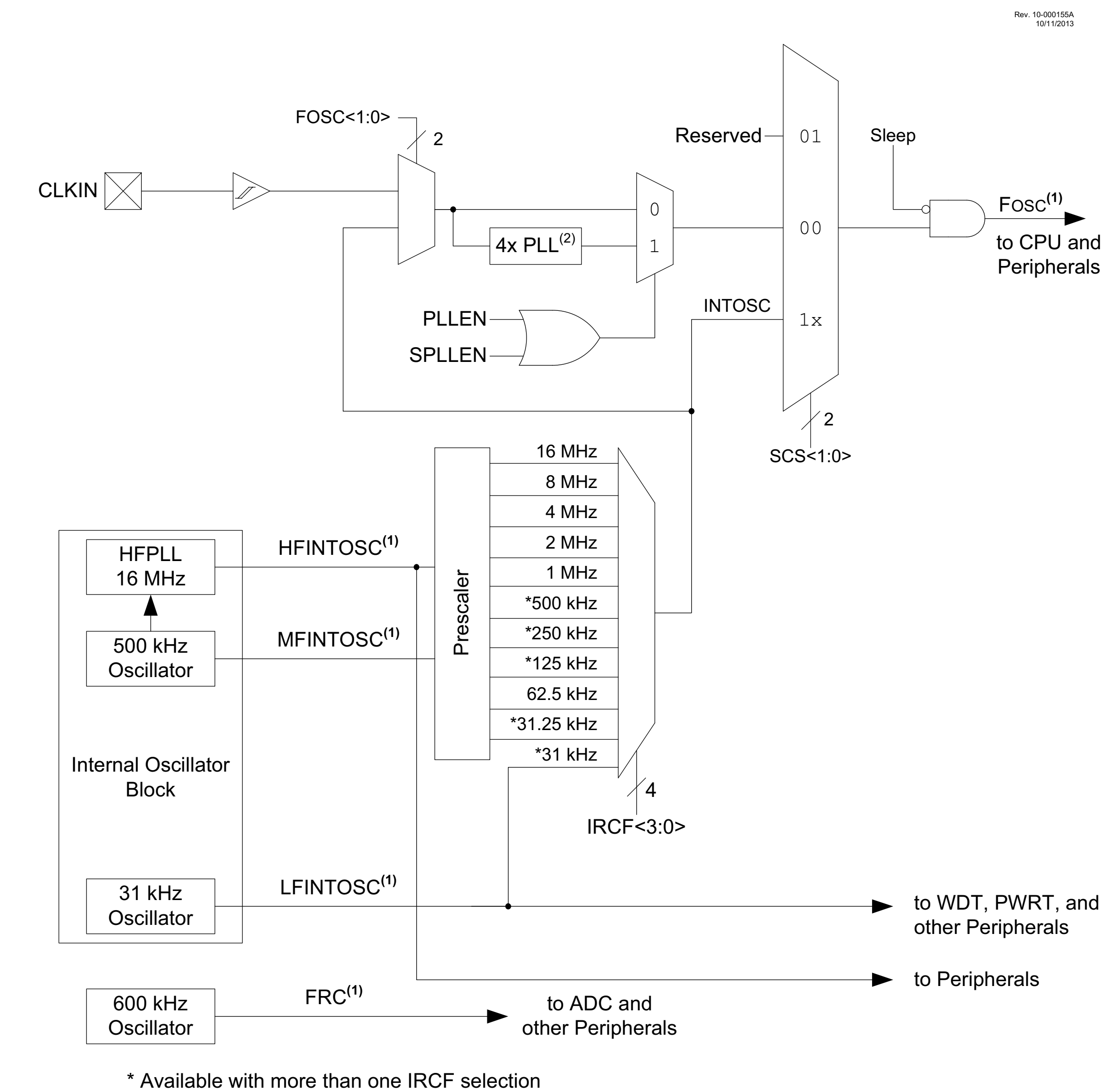
3.6 Indirect Addressing

The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the File Select Registers (FSR). If the FSRn address specifies one of the two INDFn registers, the read will return '0' and the write will not occur (though Status bits may be affected). The FSRn register value is created by the pair FSRnH and FSRnL.

The FSR registers form a 16-bit address that allows an addressing space with 65536 locations. These locations are divided into three memory regions:

- Traditional Data Memory
- Linear Data Memory
- Program Flash Memory

FIGURE 5-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM



Note 1: See **Section 5.2 “Clock Source Types”**.

2: If FOSC<1:0> = 00, 4x PLL can only be used if IRCF<3:0> = 1110.

5.2.2 INTERNAL CLOCK SOURCES

The device may be configured to use the internal oscillator block as the system clock by performing one of the following actions:

- Program the FOSC<1:0> bits in Configuration Words to select the INTOSC clock source, which will be used as the default system clock upon a device Reset.
- Write the SCS<1:0> bits in the OSCCON register to switch the system clock source to the internal oscillator during run-time. See **Section 5.3 "Clock Switching"**for more information.

In **INTOSC** mode, CLKIN is available for general purpose I/O. CLKOUT is available for general purpose I/O or CLKOUT.

The function of the OSC2/CLKOUT pin is determined by the $\overline{\text{CLKOUTEN}}$ bit in Configuration Words.

The internal oscillator block has two independent oscillators and a dedicated Phase Lock Loop, HFPLL that can produce one of three internal system clock sources.

1. The **HFINTOSC** (High-Frequency Internal Oscillator) is factory calibrated and operates at 16 MHz. The HFINTOSC source is generated from the 500 kHz MFINTOSC source and the dedicated Phase Lock Loop, HFPLL. The frequency of the HFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 5-3).
2. The **MFINTOSC** (Medium-Frequency Internal Oscillator) is factory calibrated and operates at 500 kHz. The frequency of the MFINTOSC can be user-adjusted via software using the OSCTUNE register (Register 5-3).
3. The **LFINTOSC** (Low-Frequency Internal Oscillator) is uncalibrated and operates at 31 kHz.

5.2.2.1 HFINTOSC

The High-Frequency Internal Oscillator (HFINTOSC) is a factory calibrated 16 MHz internal clock source. The frequency of the HFINTOSC can be altered via software using the OSCTUNE register (Register 5-3).

The output of the HFINTOSC connects to a postscaler and multiplexer (see Figure 5-1). One of multiple frequencies derived from the HFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See **Section 5.2.2.8 "Internal Oscillator Clock Switch Timing"** for more information.

The HFINTOSC is enabled by:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired HF frequency, and
- FOSC<1:0> = `00`, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '`1x`'.

A fast start-up oscillator allows internal circuits to power up and stabilize before switching to HFINTOSC.

The High-Frequency Internal Oscillator Ready bit (HFIOFR) of the OSCSTAT register indicates when the HFINTOSC is running.

The High-Frequency Internal Oscillator Status Locked bit (HFIOFL) of the OSCSTAT register indicates when the HFINTOSC is running within 2% of its final value.

The High-Frequency Internal Oscillator Stable bit (HFIOFS) of the OSCSTAT register indicates when the HFINTOSC is running within 0.5% of its final value.

5.2.2.2 MFINTOSC

The Medium-Frequency Internal Oscillator (MFINTOSC) is a factory calibrated 500 kHz internal clock source. The frequency of the MFINTOSC can be altered via software using the OSCTUNE register (Register 5-3).

The output of the MFINTOSC connects to a postscaler and multiplexer (see Figure 5-1). One of nine frequencies derived from the MFINTOSC can be selected via software using the IRCF<3:0> bits of the OSCCON register. See **Section 5.2.2.8 "Internal Oscillator Clock Switch Timing"** for more information.

The MFINTOSC is enabled by:

- Configure the IRCF<3:0> bits of the OSCCON register for the desired HF frequency, and
- FOSC<1:0> = `00`, or
- Set the System Clock Source (SCS) bits of the OSCCON register to '`1x`'

The Medium-Frequency Internal Oscillator Ready bit (MFIOFR) of the OSCSTAT register indicates when the MFINTOSC is running.

REGISTER 7-2: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE
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 **TMR1GIE:** Timer1 Gate Interrupt Enable bit
1 = Enables the Timer1 gate acquisition interrupt
0 = Disables the Timer1 gate acquisition interrupt

bit 6 **ADIE:** Analog-to-Digital Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

bit 5 **RCIE:** USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt

bit 4 **TXIE:** USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt

bit 3-2 **Unimplemented:** Read as ‘0’

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the Timer2 to PR2 match interrupt
0 = Disables the Timer2 to PR2 match interrupt

bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit
1 = Enables the Timer1 overflow interrupt
0 = Disables the Timer1 overflow interrupt

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

TABLE 7-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			178
PIE1	TMR1GIE	ADIE	RCIE	TXIE	—	—	TMR2IE	TMR1IE	87
PIE2	—	C2IE	C1IE	—	—	—	—	—	88
PIE3	PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	—	—	—	89
PIR1	TMR1GIF	ADIF	RCIF	TXIF	—	—	TMR2IF	TMR1IF	90
PIR2	—	C2IF	C1IF	—	—	—	—	—	91
PIR3	PWM4IF	PWM3IF	PWM2IF	PWM1IF	—	—	—	—	92

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used by interrupts.

9.1 Independent Clock Source

The WDT derives its time base from the 31 kHz LFINTOSC internal oscillator. Time intervals in this chapter are based on a nominal interval of 1 ms. See **Section 27.0 “Electrical Specifications”** for the LFINTOSC tolerances.

9.2 WDT Operating Modes

The Watchdog Timer module has four operating modes controlled by the WDTE<1:0> bits in Configuration Words. See Table 9-1.

9.2.1 WDT IS ALWAYS ON

When the WDTE bits of Configuration Words are set to ‘`11`’, the WDT is always on.

WDT protection is active during Sleep.

9.2.2 WDT IS OFF IN SLEEP

When the WDTE bits of Configuration Words are set to ‘`10`’, the WDT is on, except in Sleep.

WDT protection is not active during Sleep.

9.2.3 WDT CONTROLLED BY SOFTWARE

When the WDTE bits of Configuration Words are set to ‘`01`’, the WDT is controlled by the SWDTEN bit of the WDTCON register.

WDT protection is unchanged by Sleep. See Table 9-1 for more details.

TABLE 9-1: WDT OPERATING MODES

WDTE<1:0>	SWDTEN	Device Mode	WDT Mode
11	X	X	Active
10	X	Awake	Active
		Sleep	Disabled
01	1	X	Active
	0	X	Disabled
00	X	X	Disabled

9.3 Time-Out Period

The WDTPS bits of the WDTCON register set the time-out period from 1 ms to 256 seconds (nominal). After a Reset, the default time-out period is two seconds.

9.4 Clearing the WDT

The WDT is cleared when any of the following conditions occur:

- Any Reset
- `CLRWDT` instruction is executed
- Device enters Sleep
- Device wakes up from Sleep
- Oscillator fail
- WDT is disabled
- Oscillator Start-up Timer (OST) is running

See Table 9-2 for more information.

9.5 Operation During Sleep

When the device enters Sleep, the WDT is cleared. If the WDT is enabled during Sleep, the WDT resumes counting. When the device exits Sleep, the WDT is cleared again.

The WDT remains clear until the OST, if enabled, completes. See **Section 5.0 “Oscillator Module”** for more information on the OST.

When a WDT time-out occurs while the device is in Sleep, no Reset is generated. Instead, the device wakes up and resumes operation. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register are changed to indicate the event. The $\overline{RWDT}$ bit in the PCON register can also be used. See **Section 3.0 “Memory Organization”** for more information.

TABLE 9-2: WDT CLEARING CONDITIONS

Conditions	WDT
WDTE<1:0> = `00`	Cleared
WDTE<1:0> = `01` and SWDTEN = `0`	
WDTE<1:0> = `10` and enter Sleep	
`CLRWDT` Command	
Oscillator Fail Detected	
Exit Sleep + System Clock = EXTRC, INTOSC, EXTCLK	
Change INTOSC divider (IRCF bits)	Unaffected

See Table 10-1 for Erase Row size and the number of write latches for Flash program memory.

TABLE 10-1: FLASH MEMORY ORGANIZATION BY DEVICE

<table>
<tr><th>Device</th><th>Row Erase (words)</th><th>Write Latches (words)</th></tr>
<tr><td>PIC16(L)F1574</td><td rowspan="4">32</td><td rowspan="4">32</td></tr>
<tr><td>PIC16(L)F1575</td></tr>
<tr><td>PIC16(L)F1578</td></tr>
<tr><td>PIC16(L)F1579</td></tr>
</table>

10.2.1 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must:

1. Write the desired address to the PMADRH:PMADRL register pair.
2. Clear the CFGS bit of the PMCON1 register.
3. Then, set control bit RD of the PMCON1 register.

Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "`BSF PMCON1,RD`" instruction to be ignored. The data is available in the very next cycle, in the PMDATH:PMDATL register pair; therefore, it can be read as two bytes in the following instructions.

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

> **Note:** The two instructions following a program memory read are required to be `NOP`s. This prevents the user from executing a 2-cycle instruction on the next instruction after the RD bit is set.

FIGURE 10-1: FLASH PROGRAM MEMORY READ FLOWCHART

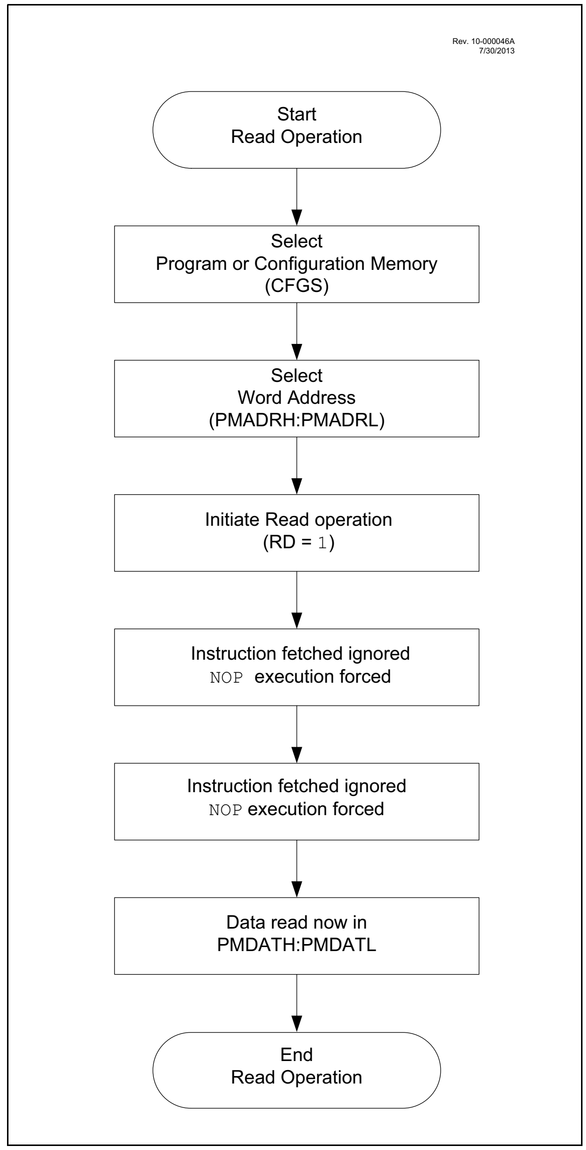
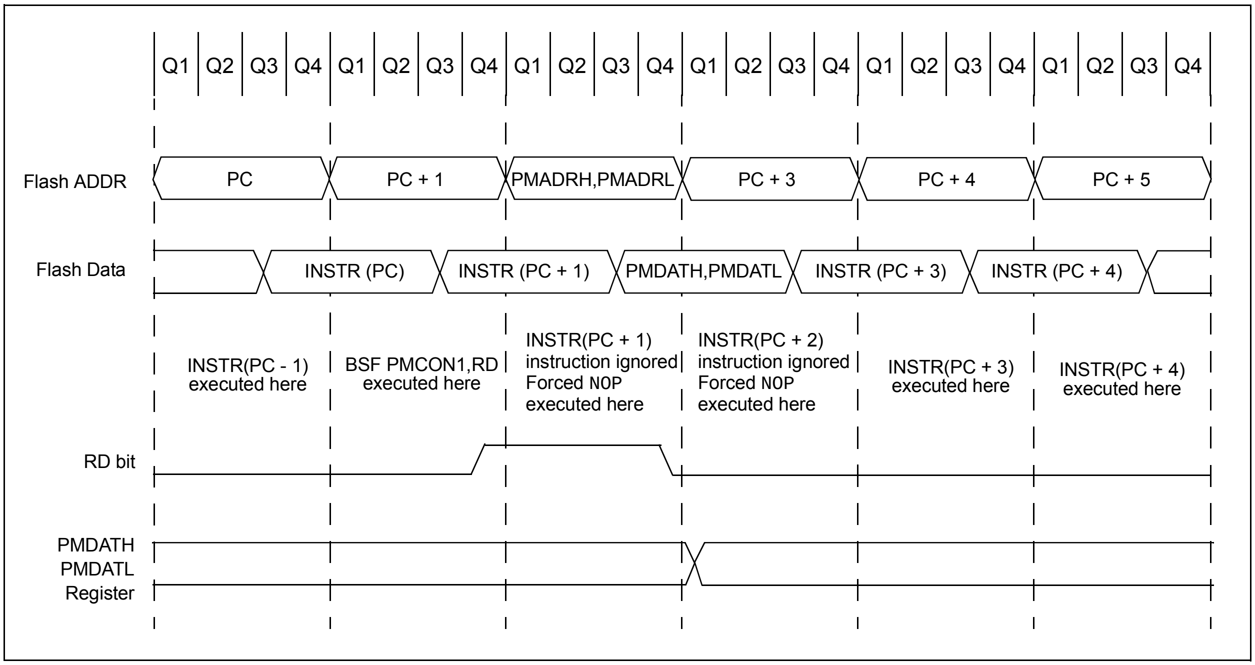


FIGURE 10-2: FLASH PROGRAM MEMORY READ CYCLE EXECUTION



EXAMPLE 10-1: FLASH PROGRAM MEMORY READ

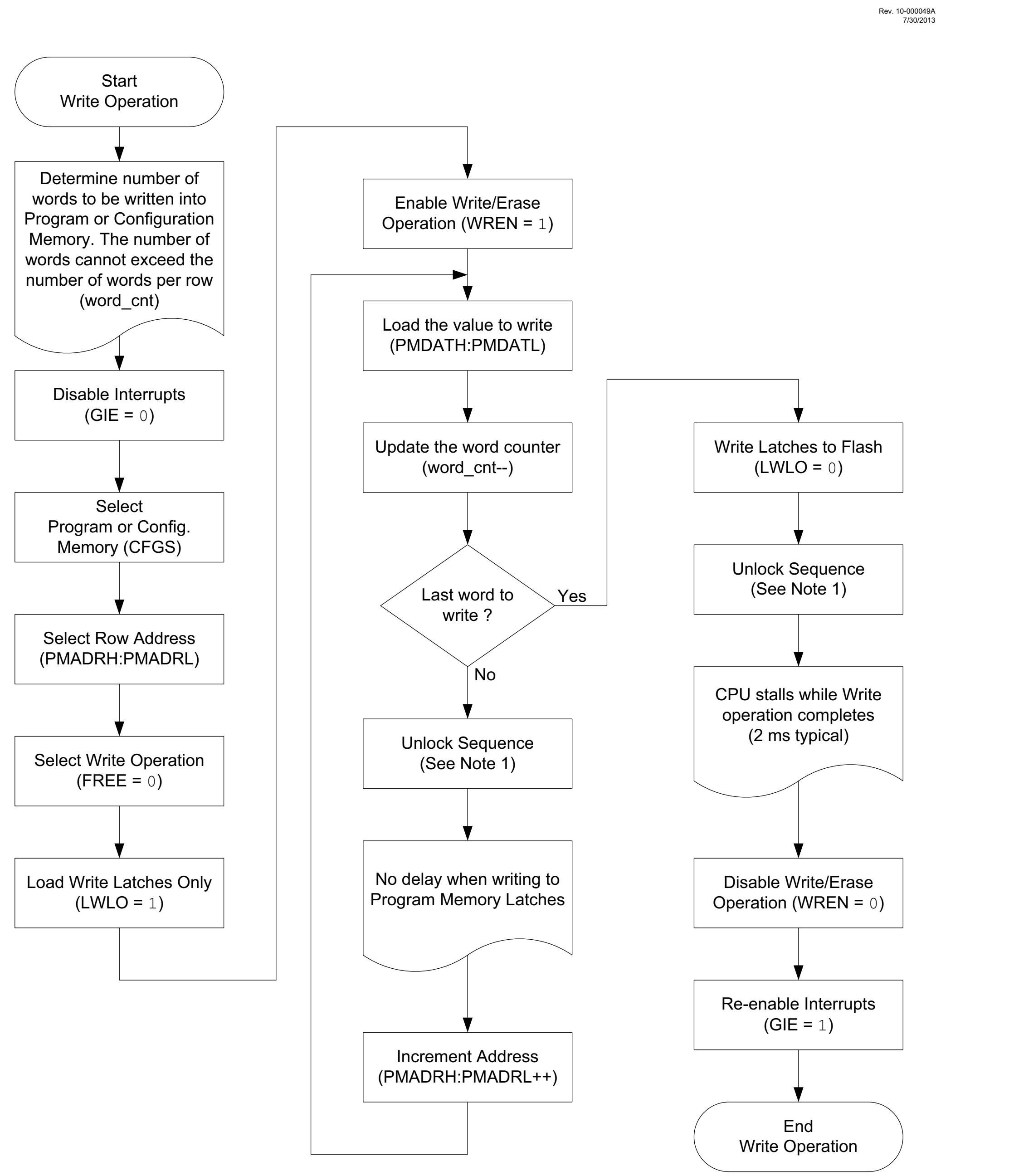
```
* This code block will read 1 word of program
* memory at the memory address:
    PROG_ADDR_HI : PROG_ADDR_LO
*   data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

    BANKSEL  PMADRL            ; Select Bank for PMCON registers
    MOVLW    PROG_ADDR_LO      ;
    MOVWF    PMADRL            ; Store LSB of address
    MOVLW    PROG_ADDR_HI      ;
    MOVWF    PMADRH            ; Store MSB of address

    BCF      PMCON1,CFGS       ; Do not select Configuration Space
    BSF      PMCON1,RD         ; Initiate read
    NOP                        ; Ignored (Figure 10-2)
    NOP                        ; Ignored (Figure 10-2)

    MOVF     PMDATL,W          ; Get LSB of word
    MOVWF    PROG_DATA_LO      ; Store in user location
    MOVF     PMDATH,W          ; Get MSB of word
    MOVWF    PROG_DATA_HI      ; Store in user location
```

FIGURE 10-6: FLASH PROGRAM MEMORY WRITE FLOWCHART



Note 1: See Figure 10-3.

12.8 Register Definitions: PPS Input Selection

REGISTER 12-1: xxxPPS: PERIPHERAL xxx INPUT SELECTION

U-0	U-0	U-0	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u	R/W-q/u
—	—	—	xxxPPS<4:0>				
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 peripheral

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-3 **xxxPPS<4:3>:** Peripheral xxx Input PORT Selection bits
11 = Reserved. Do not use.
10 = Peripheral input is PORTC
01 = Peripheral input is PORTB(2)
00 = Peripheral input is PORTA

bit 2-0 **xxxPPS<2:0>:** Peripheral xxx Input Bit Selection bits (1)
111 = Peripheral input is from PORTx Bit 7 (Rx7)
110 = Peripheral input is from PORTx Bit 6 (Rx6)
101 = Peripheral input is from PORTx Bit 5 (Rx5)
100 = Peripheral input is from PORTx Bit 4 (Rx4)
011 = Peripheral input is from PORTx Bit 3 (Rx3)
010 = Peripheral input is from PORTx Bit 2 (Rx2)
001 = Peripheral input is from PORTx Bit 1 (Rx1)
000 = Peripheral input is from PORTx Bit 0 (Rx0)

Note 1: See Table 12-1 for xxxPPS register list and Reset values.
2: PIC16(L)F1578/9 only.

REGISTER 12-2: RxyPPS: PIN Rxy OUTPUT SOURCE SELECTION REGISTER

U-0	U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	RxyPPS<4:0>				
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 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RxyPPS<4:0>:** Pin Rxy Output Source Selection bits
Selection code determines the output signal on the port pin.
See Table 12-2 for the selection codes

TABLE 14-1: PERIPHERALS REQUIRING THE FIXED VOLTAGE REFERENCE (FVR)

Peripheral	Conditions	Description
HFINTOSC	FOSC<2:0> = 010 and IRCF<3:0> = 000x	INTOSC is active and device is not in Sleep.
BOR	BOREN<1:0> = 11	BOR always enabled.
	BOREN<1:0> = 10 and BORFS = 1	BOR disabled in Sleep mode, BOR Fast Start enabled.
	BOREN<1:0> = 01 and BORFS = 1	BOR under software control, BOR Fast Start enabled.
LDO	All PIC16F1574/5/8/9 devices, when VREGPM = 1 and not in Sleep	The device runs off of the Low-Power Regulator when in Sleep mode.

16.1.5 INTERRUPTS

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

> **Note 1:** The ADIF bit is set at the completion of every conversion, regardless of whether or not the ADC interrupt is enabled.
>
> **2:** The ADC operates during Sleep only when the FRC oscillator is selected.

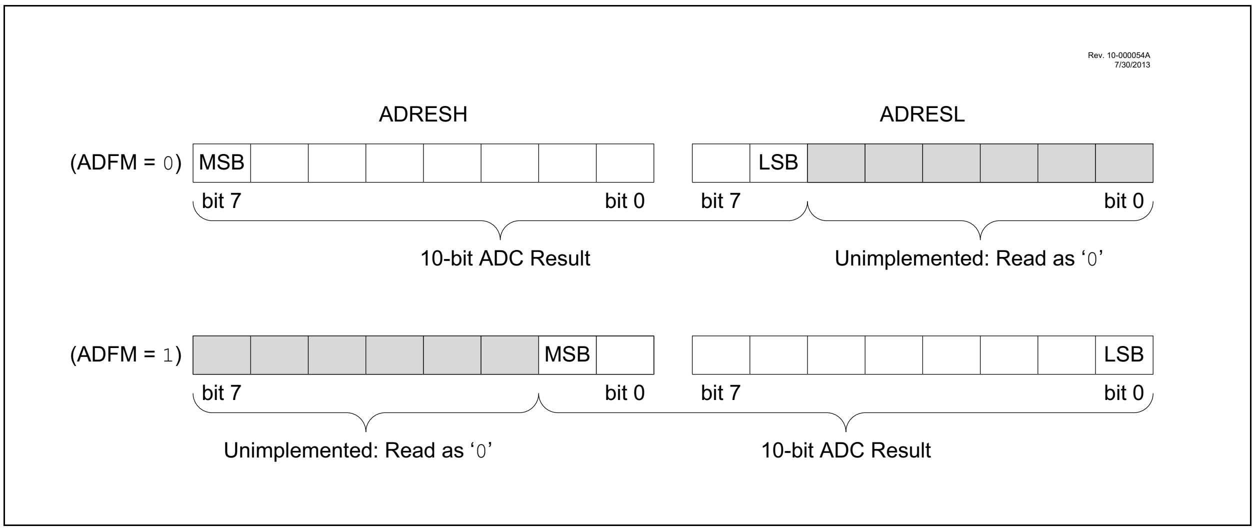
This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the `SLEEP` instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the ADIE bit of the PIE1 register and the PEIE bit of the INTCON register must both be set and the GIE bit of the INTCON register must be cleared. If all three of these bits are set, the execution will switch to the Interrupt Service Routine.

16.1.6 RESULT FORMATTING

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

Figure 16-3 shows the two output formats.

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



22.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is a serial I/O communications peripheral. It contains all the clock generators, shift registers and data buffers necessary to perform an input or output serial data transfer independent of device program execution. The EUSART, also known as a Serial Communications Interface (SCI), can be configured as a full-duplex asynchronous system or half-duplex synchronous system. Full-Duplex mode is useful for communications with peripheral systems, such as CRT terminals and personal computers. Half-Duplex Synchronous mode is intended for communications with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs or other microcontrollers. These devices typically do not have internal clocks for baud rate generation and require the external clock signal provided by a master synchronous device.

The EUSART module includes the following capabilities:

- Full-duplex asynchronous transmit and receive
- Two-character input buffer
- One-character output buffer
- Programmable 8-bit or 9-bit character length
- Address detection in 9-bit mode
- Input buffer overrun error detection
- Received character framing error detection
- Half-duplex synchronous master
- Half-duplex synchronous slave
- Programmable clock polarity in synchronous modes
- Sleep operation

The EUSART module implements the following additional features, making it ideally suited for use in Local Interconnect Network (LIN) bus systems:

- Automatic detection and calibration of the baud rate
- Wake-up on Break reception
- 13-bit Break character transmit

Block diagrams of the EUSART transmitter and receiver are shown in Figure 22-1 and Figure 22-2.

FIGURE 22-1: EUSART TRANSMIT BLOCK DIAGRAM

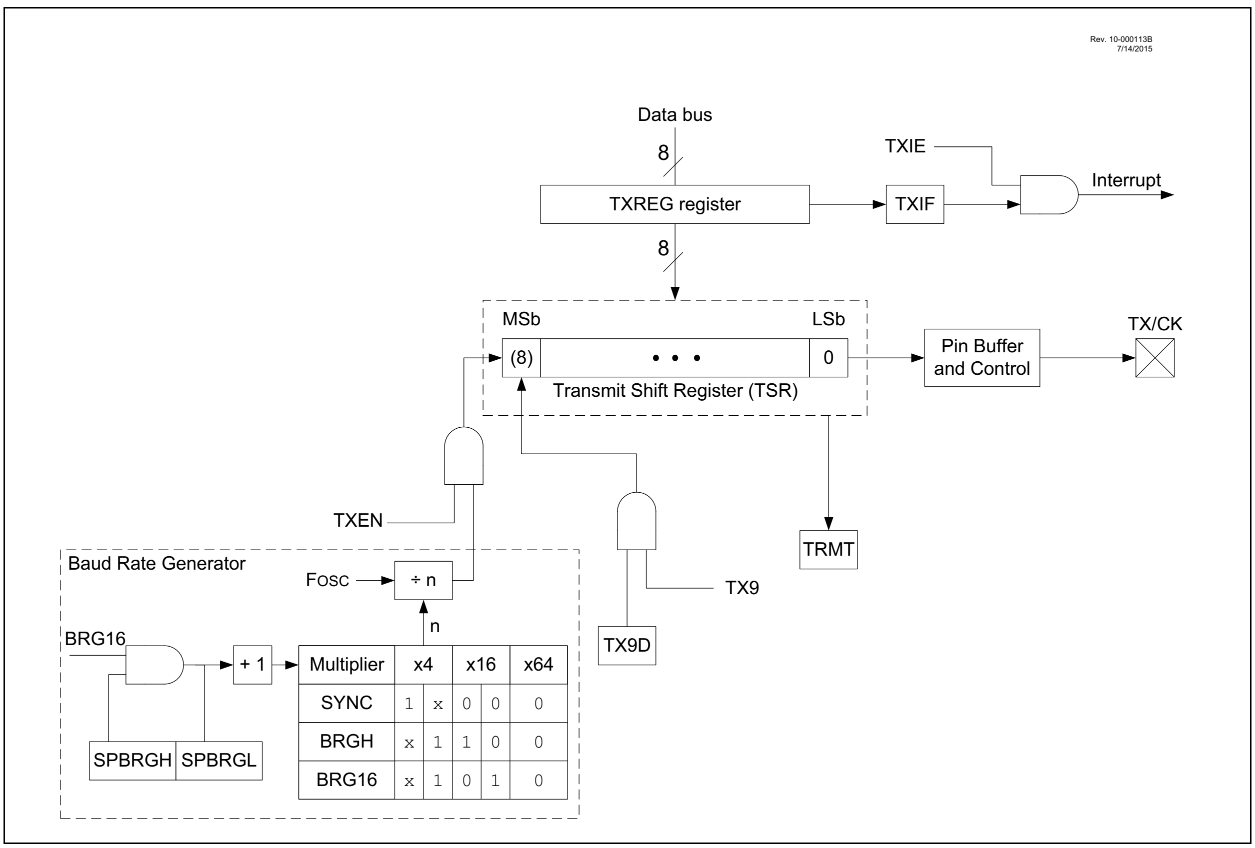


TABLE 24-2: SUMMARY OF REGISTERS ASSOCIATED WITH CWG

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	ANSA4	—	ANSA2	ANSA1	ANSA0	121
CWG1CON0	G1EN	—	—	G1POLB	G1POLA	—	—	G1CS0	253
CWG1CON1	G1ASDLB<1:0>		G1ASDLA<1:0>		—	G1IS<2:0>			254
CWG1CON2	G1ASE	G1ARSEN	—	—	G1ASDSC2	G1ASDSC1	G1ASDSPPS	—	255
CWG1DBF	—	—	CWG1DBF<5:0>						256
CWG1DBR	—	—	CWG1DBR<5:0>						256
TRISA	—	—	TRISA5	TRISA4	—(1)	TRISA2	TRISA1	TRISA0	120

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by CWG.
Note 1: Unimplemented, read as '1'.

FIGURE 26-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

13 8	7	6 0
OPCODE	d	f (FILE #)

d = `0` for destination W
d = `1` for destination f
f = 7-bit file register address

Bit-oriented file register operations

13 10	9 7	6 0
OPCODE	b (BIT #)	f (FILE #)

b = 3-bit bit address
f = 7-bit file register address

Literal and control operations

General

13 8	7 0
OPCODE	k (literal)

k = 8-bit immediate value

`CALL` and `GOTO` instructions only

13 11	10 0
OPCODE	k (literal)

k = 11-bit immediate value

`MOVLP` instruction only

13 7	6 0
OPCODE	k (literal)

k = 7-bit immediate value

`MOVLB` instruction only

13 5	4 0
OPCODE	k (literal)

k = 5-bit immediate value

`BRA` instruction only

13 9	8 0
OPCODE	k (literal)

k = 9-bit immediate value

`FSR` Offset instructions

13 7	6	5 0
OPCODE	n	k (literal)

n = appropriate `FSR`
k = 6-bit immediate value

`FSR` Increment instructions

13 3	2	1 0
OPCODE	n	m (mode)

n = appropriate `FSR`
m = 2-bit mode value

OPCODE only

13 0
OPCODE

TABLE 27-8: OSCILLATOR PARAMETERS

Standard Operating Conditions (unless otherwise stated)								
Param. No.	**Sym.**	**Characteristic**	**Freq. Tolerance**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
OS08	HFOSC	Internal Calibrated HFINTOSC Frequency(1)	±2%	—	16.0	—	MHz	VDD = 3.0V, TA = 25°C, **(Note 2)**
OS09	LFOSC	Internal LFINTOSC Frequency	—	—	31	—	kHz	
OS10*	TWARM	HFINTOSC Wake-up from Sleep Start-up Time	—	—	5	15	μs	
		LFINTOSC Wake-up from Sleep Start-up Time	—	—	0.5	—	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 27-6: "HFINTOSC Frequency Accuracy over Device VDD and Temperature.

FIGURE 27-6: HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE

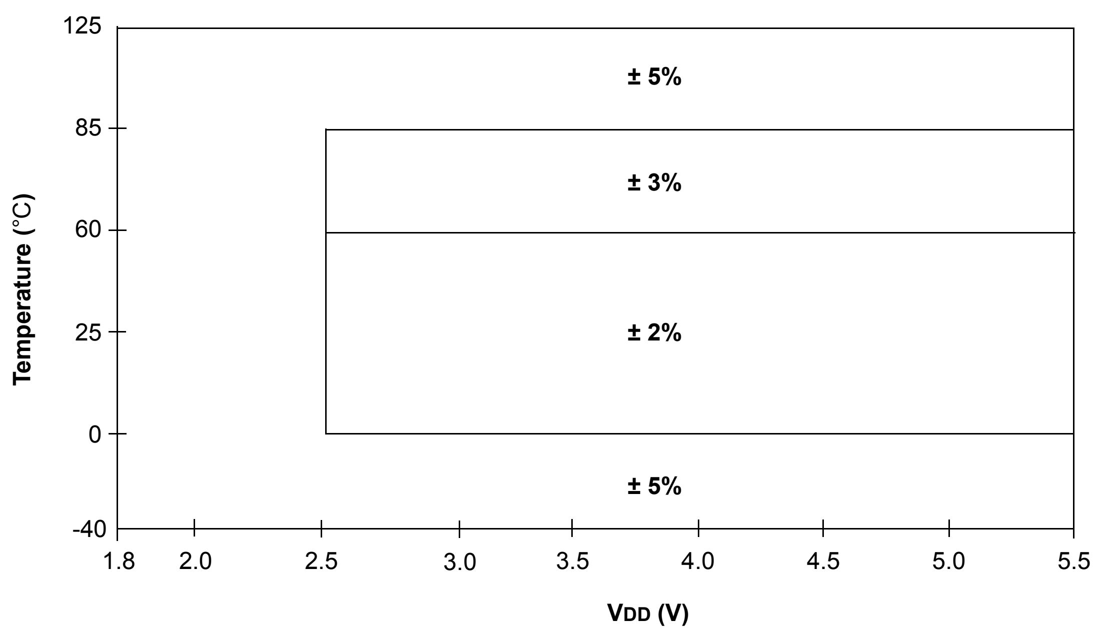


TABLE 27-9: PLL CLOCK TIMING SPECIFICATIONS (VDD = 2.7V TO 5.5V)

Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
F10	FOSC	Oscillator Frequency Range	4	—	8	MHz	
F11	FSYS	On-Chip VCO System Frequency	16	—	32	MHz	
F12	TRC	PLL Start-up Time (Lock Time)	—	—	2	ms	
F13*	ΔCLK	CLKOUT Stability (Jitter)	-0.25%	—	+0.25%	%	

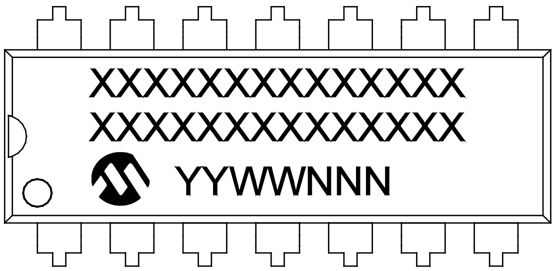
* These parameters are characterized but not tested.

† Data in "Typ" column is at 3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

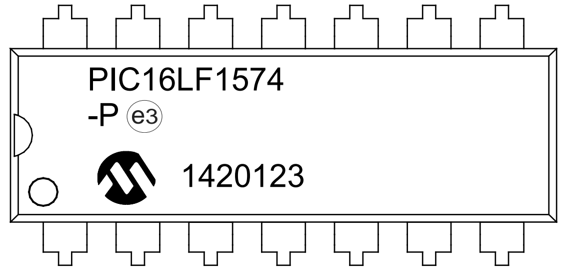
30.0 PACKAGING INFORMATION

30.1 Package Marking Information

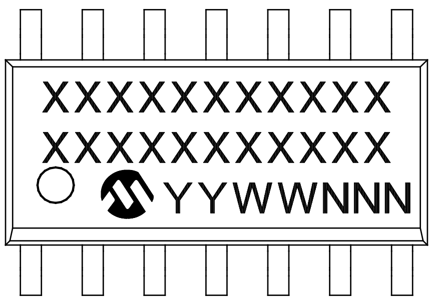
14-Lead PDIP (300 mil)



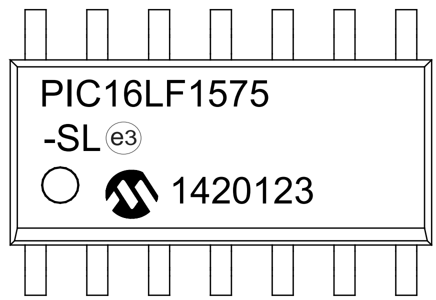
Example



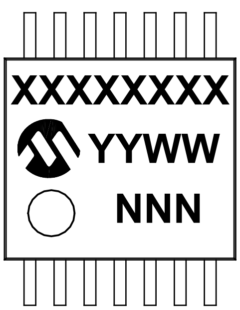
14-Lead SOIC (3.90 mm)



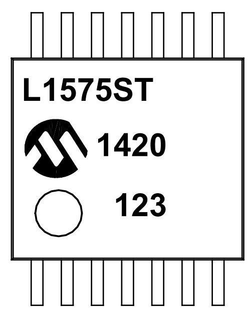
Example



14-Lead TSSOP (4.4 mm)



Example



Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC® designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.