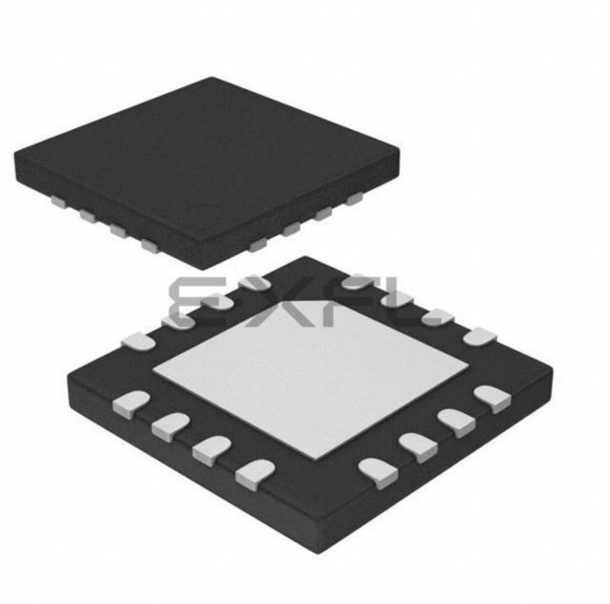


Welcome to E-XFL.COM

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	12
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 11x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	16-UQFN Exposed Pad
Supplier Device Package	16-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf18326-i-jq

- Up to 18 I/O Pins:
  - Individually programmable pull-ups
  - Slew rate control
  - Interrupt-on-change with edge-select
  - Input level selection control (ST or TTL)
  - Digital open-drain enable
- Peripheral Pin Select (PPS):
  - I/O pin remapping of digital peripherals
- Timer modules:
  - Timer0:
    - 8/16-bit timer/counter
    - Synchronous or asynchronous operation
    - Programmable prescaler/postscaler
    - Time base for capture/compare function
  - Timer1/3/5 with gate control:
    - 16-bit timer/counter
    - Programmable internal or external clock sources
    - Multiple gate sources
    - Multiple gate modes
    - Time base for capture/compare function
  - Timer2/4/6:
    - 8-bit timers
    - Programmable prescaler/postscaler
    - Time base for PWM function

Analog Peripherals

- 10-bit Analog-to-Digital Converter (ADC):
  - 17 external channels
  - Conversion available during Sleep
- Comparator:
  - Two comparators
  - Fixed Voltage Reference at non-inverting input(s)
  - Comparator outputs externally accessible
- 5-bit Digital-to-Analog Converter (DAC):
  - 5-bit resolution, rail-to-rail
  - Positive Reference Selection
  - Unbuffered I/O pin output
  - Internal connections to ADCs and comparators
- Voltage Reference:
  - Fixed Voltage Reference with 1.024V, 2.048V and 4.096V output levels Flexible Oscillator Structure
- High-Precision Internal Oscillator:
  - Software-selectable frequency range up to 32 MHz
  - ±1% at nominal 4 MHz calibration point
- 4x PLL with External Sources
- Low-Power Internal 31 kHz Oscillator (LFINTOSC)
- External Low-Power 32 kHz Crystal Oscillator (SOSC)
- External Oscillator Block with:
  - Three Crystal/Resonator modes up to 20 MHz
  - Three External Clock modes up to 20 MHz
  - Fail-Safe Clock Monitor
    - Detects clock source failure
  - Oscillator Start-up Timer (OST)
    - Ensures stability of crystal oscillator sources

TABLE 1-3: PIC16(L)F18346 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RA5/ANA5/T1CKI(1)/ T3CKI(1)/ T5CKI(1)/ SOSCIN/SOSCI/ CLKIN/OSC1	RA5	TTL/ST	CMOS	General purpose I/O.
	ANA5	AN	—	ADC Channel A5 input.
	T1CKI	TTL/ST	—	TMR1 Clock input.
	T3CKI	TTL/ST	—	TMR3 Clock input.
	T5CKI	TTL/ST	—	TMR5 Clock input.
	SOSCIN	TTL/ST	—	Secondary Oscillator input connection.
	SOSCI	XTAL	—	Secondary Oscillator connection.
	CLKIN	TTL/ST	—	External clock input.
	OSC1	XTAL	—	Crystal/Resonator (LP, XT, HS modes).
RB4/ANB4/SDI1(1)/ SDA1(1,3)/ CLCIN2(1)	RB4	TTL/ST	CMOS	General purpose I/O.
	ANB4	AN	—	ADC Channel B4 input.
	SDI1	TTL/ST	CMOS	SPI Data input 1.
	SDA1	I^{2}C	OD	I^{2}C Data 1.
	CLCIN2	TTL/ST	—	Configurable Logic Cell 2 input.
RB5/ANB5/SDI2(1)/ SDA2(1,3)/ RX(1)/DT/CLCIN3(1)	RB5	TTL/ST	CMOS	General purpose I/O.
	ANB5	AN	—	ADC Channel B5 input.
	SDI2	TTL/ST	CMOS	SPI Data input 2.
	SDA2	I^{2}C	OD	I^{2}C Data 2.
	RX	TTL/ST	CMOS	EUSART asynchronous input.
	DT	TTL/ST	CMOS	EUSART synchronous data output.
	CLCIN3	TTL/ST	—	Configurable Logic Cell 3 input.
RB6/ANB6/SCK1(1)/ SCL1(1,3)	RB6	TTL/ST	CMOS	General purpose I/O.
	ANB6	AN	—	ADC Channel B6 input.
	SCK1	TTL/ST	CMOS	SPI Clock 1.
	SCL1	I^{2}C	OD	I^{2}C Clock 1.
RB7/ANB7/SCK2(1)/ SCL2(1,3)	RB7	TTL/ST	CMOS	General purpose I/O.
	ANB7	AN	—	ADC Channel B7 input.
	SCK2	TTL/ST	CMOS	SPI Clock 2.
	SCL2	I^{2}C	OD	I^{2}C Clock 2.
RC0/ANC0/C2IN0+	RC0	TTL/ST	CMOS	General purpose I/O.
	ANC0	AN	—	ADC Channel C0 input.
	C2IN0+	AN	—	Comparator C2 positive input.
RC1/ANC1/C1IN1-/C2IN1-	RC1	TTL/ST	CMOS	General purpose I/O.
	ANC1	AN	—	ADC Channel C1 input.
	C1IN1-	AN	—	Comparator C1 negative input.
	C2IN1-	AN	—	Comparator C2 negative input.
RC2/ANC2/C1IN2-/C2IN2-/ MDCIN1(1)	RC2	TTL/ST	CMOS	General purpose I/O.
	ANC2	AN	—	ADC Channel C2 input.
	C1IN2-	AN	—	Comparator C1 negative input.
	C2IN2-	AN	—	Comparator C2 negative input.
	MDCIN1	TTL/ST	—	Modular Carrier input 1.

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 I^{2}C = Schmitt Trigger input with I^{2}C
HV = High Voltage XTAL = Crystal levels

Note 1: Default peripheral input. Input can be moved to any other pin with the PPS input selection registers. See Register 13-2.

2: All pin outputs default to PORT latch data. Any pin can be selected as a digital peripheral output with the PPS output selection registers. See Register 13-2.

3: These I^{2}C functions are bidirectional. The output pin selections must be the same as the input pin selections.

4.2.3 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the application to control the desired operation of peripheral functions in the device. The Special Function Registers occupy the 20 bytes after the core registers of every data memory bank (addresses x0Ch/x8Ch through x1Fh/x9Fh), with the exception of banks 27, 28, and 29 (PPS and CLC registers). The registers associated with the operation of the peripherals are described in the appropriate peripheral chapter of this data sheet.

4.2.4 GENERAL PURPOSE RAM

There are up to 80 bytes of GPR in each data memory bank. The general purpose RAM can be accessed in a non-banked method via the FSRs. This can simplify access to large memory structures. See **Section 4.5.2 "Linear Data Memory"** for more information.

4.2.5 COMMON RAM

There are 16 bytes of common RAM accessible from all banks.

4.2.6 DEVICE MEMORY MAPS

The memory maps for PIC16(L)F18326/18346 are as shown in Table 4-4.

TABLE 4-3: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (ALL BANKS)(1)

Bank Offset	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
All Banks											
000h	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
001h	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	xxxx xxxx
002h	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
003h	STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	---1 1000	---q quuu
004h	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
005h	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
006h	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
007h	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
008h	BSR	—	—	—	BSR4	BSR3	BSR2	BSR1	BSR0	---0 0000	---0 0000
009h	WREG	Working Register								0000 0000	uuuu uuuu
00Ah	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
00Bh	INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	00-- ---1	00-- ---1

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

Note 1: These registers can be accessed from any bank.

TABLE 4-4: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18326	PIC16(L)F18346	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
Bank 28													
CPU CORE REGISTERS; see Table 4-2 for specifics													
E21h	SSP1DATPPS	X	—	—	—	—	SSP1DATPPS<4:0>					---1 0001	---u uuuu
		—	X	—	—	—	SSP1DATPPS<4:0>					---0 1100	---u uuuu
E22h	SSP1SSPPS	X	—	—	—	—	SSP1SSPPS<4:0>					---1 0011	---u uuuu
		—	X	—	—	—	SSP1SSPPS<4:0>					---1 0100	---u uuuu
E23h	—	—		Unimplemented								—	—
E24h	RXPPS	X	—	—	—	—	RXPPS<4:0>					---1 0101	---u uuuu
		—	X	—	—	—	RXPPS<4:0>					---0 1101	---u uuuu
E25h	TXPPS	X	—	—	—	—	TXPPS<4:0>					---1 0100	---u uuuu
		—	X	—	—	—	TXPPS<4:0>					---0 1111	---u uuuu
E26h	—	—		Unimplemented								—	—
E27h	—	—		Unimplemented								—	—
E28h	CLCIN0PPS	X	—	—	—	—	CLCIN0PPS<4:0>					---1 0011	---u uuuu
		—	X	—	—	—	CLCIN0PPS<4:0>					---0 0010	---u uuuu
E29h	CLCIN1PPS	X	—	—	—	—	CLCIN1PPS<4:0>					---0 0100	---u uuuu
		—	X	—	—	—	CLCIN1PPS<4:0>					---1 0011	---u uuuu
E2Ah	CLCIN2PPS	X	—	—	—	—	CLCIN2PPS<4:0>					---1 0001	---u uuuu
		—	X	—	—	—	CLCIN2PPS<4:0>					---0 1100	---u uuuu
E2Bh	CLCIN3PPS	X	—	—	—	—	CLCIN3PPS<4:0>					---0 0101	---u uuuu
		—	X	—	—	—	CLCIN3PPS<4:0>					---0 1101	---u uuuu
E2Ch	T3CKIPPS	X	—	—	—	—	T3CKIPPS<4:0>					---1 0001	---u uuuu
		—	X	—	—	—	T3CKIPPS<4:0>					---0 0101	---u uuuu
E2Dh	T3GPPS	X	—	—	—	—	T3GPPS<4:0>					---1 0001	---u uuuu
		—	X	—	—	—	T3GPPS<4:0>					---1 0100	---u uuuu
E2Eh	T5CKIPPS	X	—	—	—	—	T5CKIPPS<4:0>					---1 0001	---u uuuu
		—	X	—	—	—	T5CKIPPS<4:0>					---0 0101	---u uuuu
E2Fh	T5GPPS	X	—	—	—	—	T5GPPS<4:0>					---1 0001	---u uuuu
		—	X	—	—	—	T5GPPS<4:0>					---1 0100	---u uuuu

Legend: x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.
Note 1: Only on PIC16F18326/18346.
2: Register accessible from both User and ICD Debugger.

Preliminary

7.2.1.3 Oscillator Start-up Timer (OST)

If the oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from OSC1. This occurs following a Power-on Reset (POR), Brown-out Reset (BOR), or a wake-up from Sleep. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the oscillator module.

7.2.1.4 4x PLL

The oscillator module contains a PLL that can be used with external clock sources to provide a system clock source. The input frequency for the PLL must fall within specifications. See the PLL Clock Timing Specifications in Table 35-9.

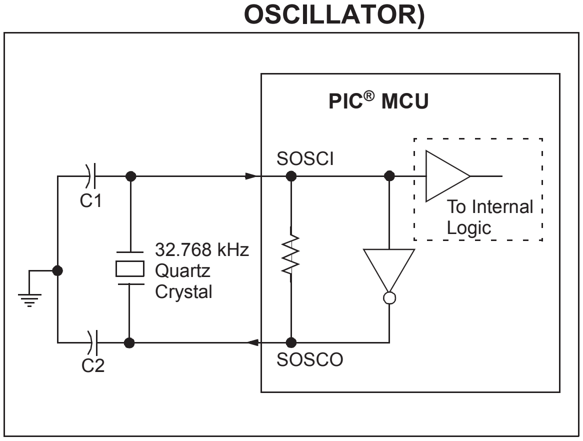
The PLL may be enabled for use by one of two methods:

1. Program the RSTOSC bits in the Configuration Word 1 to enable the EXTOSC with 4x PLL.
2. Write the NOSC<2:0> bits in the OSCCON1 register to enable the EXTOSC with 4x PLL.

7.2.1.5 Secondary Oscillator

The secondary oscillator is a separate oscillator block that can be used as an alternate system clock source. The secondary oscillator is optimized for 32.768 kHz, and can be used with an external crystal oscillator connected to the SOSCI and SOSCO device pins, or an external clock source connected to the SOSCIN pin. The secondary oscillator can be selected during run-time using clock switching. Refer to **Section 7.3 "Clock Switching"** for more information.

FIGURE 7-5: QUARTZ CRYSTAL OPERATION (SECONDARY OSCILLATOR)



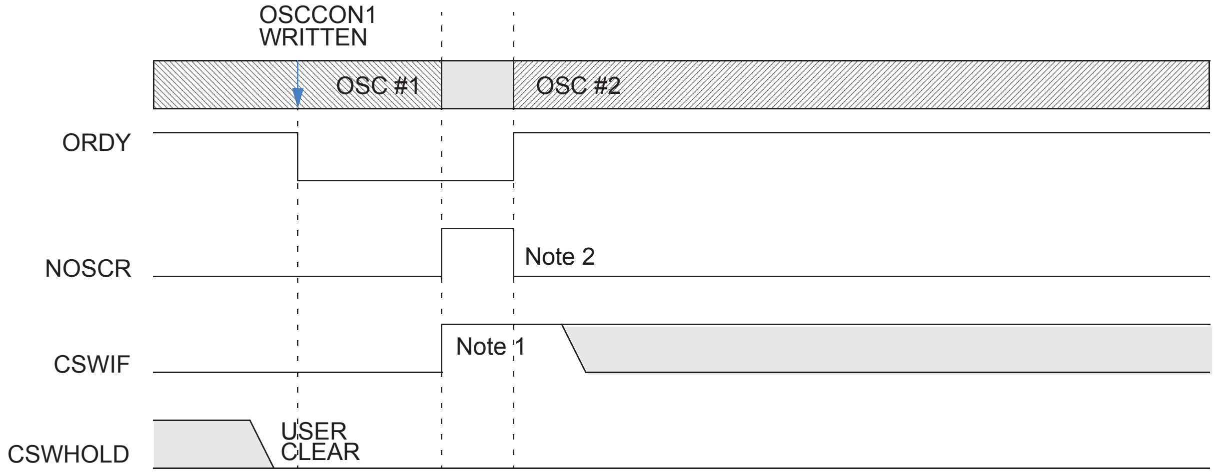
Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

2: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

3: For oscillator design assistance, reference the following Microchip Application Notes:

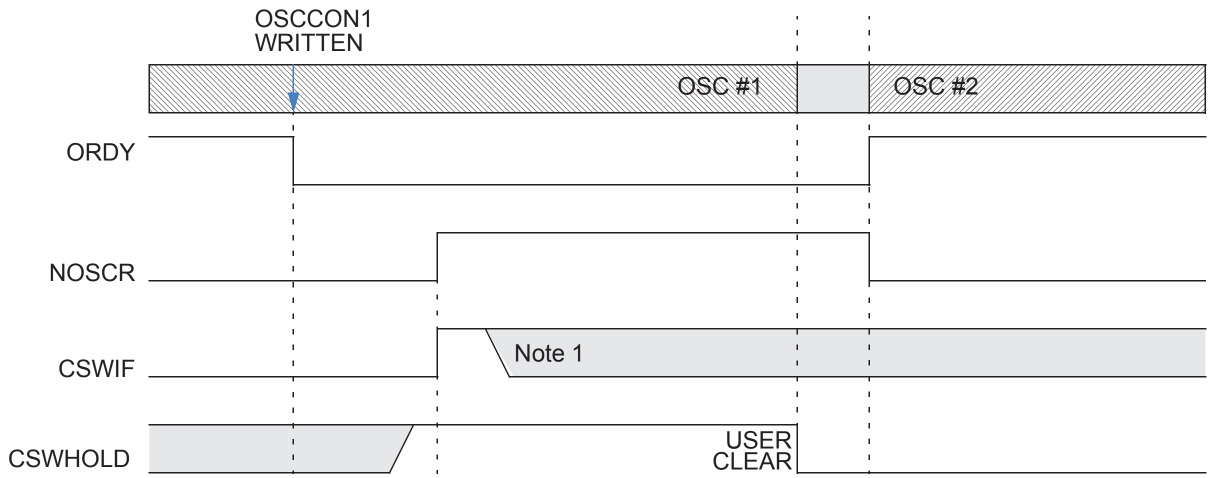
- *AN826, Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices* (DS00826)
- *AN849, Basic PICmicro® Oscillator Design* (DS00849)
- *AN943, Practical PICmicro® Oscillator Analysis and Design* (DS00943)
- *AN949, Making Your Oscillator Work* (DS00949)
- *TB097, Interfacing a Micro Crystal MS1V-T1K 32.768 kHz Tuning Fork Crystal to a PIC16F690/SS* (DS91097)
- *AN1288, Design Practices for Low-Power External Oscillators* (DS01288)

FIGURE 7-6: CLOCK SWITCH (CSWHOLD = 0)



Note 1: CSWIF is asserted coincident with NOSCR; interrupt is serviced at OSC#2 speed.
2: The assertion of NOSCR is hidden from the user because it appears only for the duration of the switch.

FIGURE 7-7: CLOCK SWITCH (CSWHOLD = 1)



Note 1: CSWIF is asserted coincident with NOSCR, and may be cleared before or after clearing CSWHOLD = 0.

EXAMPLE 11-3: ERASING ONE ROW OF PROGRAM FLASH MEMORY

```
; This sample row erase routine assumes the following:
; 1.A valid address within the erase row is loaded in variables ADDRH:ADDRL
; 2.ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)

BANKSEL     NVMADRL
MOVF        ADDRL,W
MOVWF       NVMADRL            ; Load lower 8 bits of erase address boundary
MOVF        ADDRH,W
MOVWF       NVMADRH            ; Load upper 6 bits of erase address boundary
BCF         NVMCON1,NVMREGS    ; Choose Program Flash Memory area
BSF         NVMCON1,FREE       ; Specify an erase operation
BSF         NVMCON1,WREN       ; Enable writes
BCF         INTCON,GIE         ; Disable interrupts during unlock sequence

; -------------------------------REQUIRED UNLOCK SEQUENCE:------------------------------

MOVLW       55h                ; Load 55h to get ready for unlock sequence
MOVWF       NVMCON2            ; First step is to load 55h into NVMCON2
MOVLW       AAh                ; Second step is to load AAh into W
MOVWF       NVMCON2            ; Third step is to load AAh into NVMCON2
BSF         NVMCON1,WR         ; Final step is to set WR bit

; --------------------------------------------------------------------------------------

BSF         INTCON,GIE         ; Re-enable interrupts, erase is complete
BCF         NVMCON1,WREN       ; Disable writes
```

TABLE 11-2: NVM ORGANIZATION AND ACCESS INFORMATION

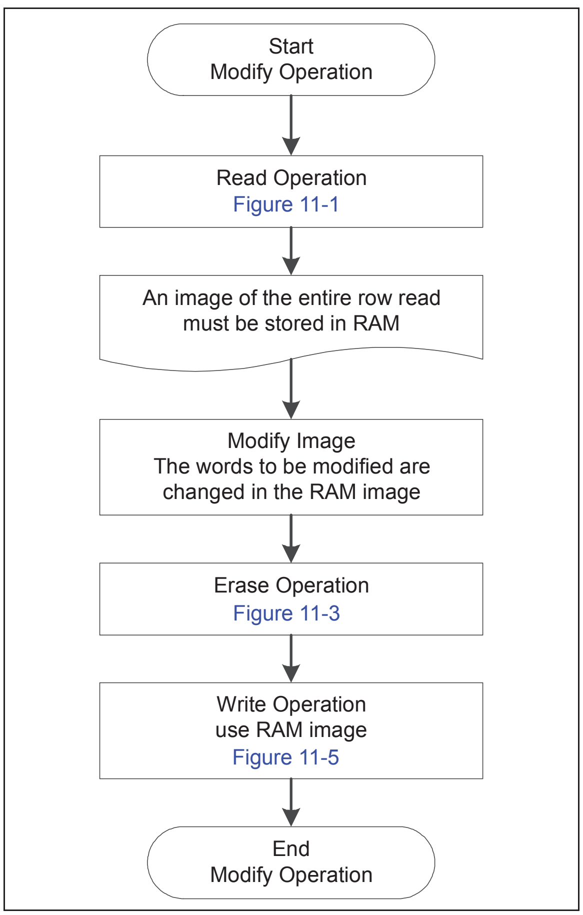
Master Values			NVMREG Access			FSR Access	
Memory Function	**Program Counter (PC), ICSP™ Address**	**Memory Type**	**NVMREGS bit (NVMCON1)**	**NVMADR <14:0>**	**Allowed Operations**	**FSR Address**	**FSR Programming Address**
Reset Vector	0000h	Program Flash Memory	0	0000h	READ WRITE	8000h	READ-ONLY
User Memory	0001h		0	0001h		8001h	
	0003h			0003h		8003h	
INT Vector	0004h		0	0004h		8004h	
User Memory	0005h		0	0005h		8005h	
	3FFFh			3FFFh		BFFFh	
User ID	No PC Address	Program Flash Memory	1	0000h	READ	No Access	
				0003h			
Reserved		—	—	0004h	—		
Rev ID		Program Flash Memory	1	0005h	READ		
Device ID			1	0006h			
CONFIG1			1	0007h			
CONFIG2			1	0008h			
CONFIG3			1	0009h			
CONFIG4				000Ah			
User Memory		EEPROM	1	7000h	READ WRITE	7000h	READ-ONLY
				70FFh		70FFh	

11.4.6 MODIFYING PROGRAM FLASH MEMORY

When modifying existing data in a program memory row, and data within that row must be preserved, it must first be read and saved in a RAM image. Program memory is modified using the following steps:

1. Load the starting address of the row to be modified.
2. Read the existing data from the row into a RAM image.
3. Modify the RAM image to contain the new data to be written into program memory.
4. Load the starting address of the row to be rewritten.
5. Erase the program memory row.
6. Load the write latches with data from the RAM image.
7. Initiate a programming operation.

FIGURE 11-6: PROGRAM FLASH MEMORY MODIFY FLOWCHART



REGISTER 12-15: SLRCONB: PORTB SLEW RATE CONTROL 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
SLRB7	SLRB6	SLRB5	SLRB4	—	—	—	—
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 **SLRB<7:4>:** PORTB Slew Rate Control on pins RB<7:4>, respectively
1 = Slew rate enabled
0 = Slew rate disabled

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

REGISTER 12-16: INLVLB: PORTB INPUT LEVEL CONTROL 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
INLVLB7	INLVLB6	INLVLB5	INLVLB4	—	—	—	—
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 **INLVLB<7:4>:** PORTB Input Level Select on pins RB<7:4>, respectively
1 = ST input used for PORT reads
0 = TTL input used for PORT reads

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

16.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of VDD, with 1.024V, 2.048V or 4.096V selectable output levels. The output of the FVR subsystem can be configured to supply a reference voltage to the following:

- ADC input channel
- ADC positive reference
- Comparator positive input
- Digital-to-Analog Converter (DAC)

The FVR can be enabled by setting the FVREN bit of the FVRCON register.

Note:	Fixed Voltage Reference output cannot exceed VDD.

16.1 Independent Gain Amplifiers

The output of the FVR, which is supplied to the ADC, Comparators and DAC, is routed through two independent programmable gain amplifiers. Each amplifier can be programmed for a gain of 1x, 2x or 4x, to produce the three possible voltage levels.

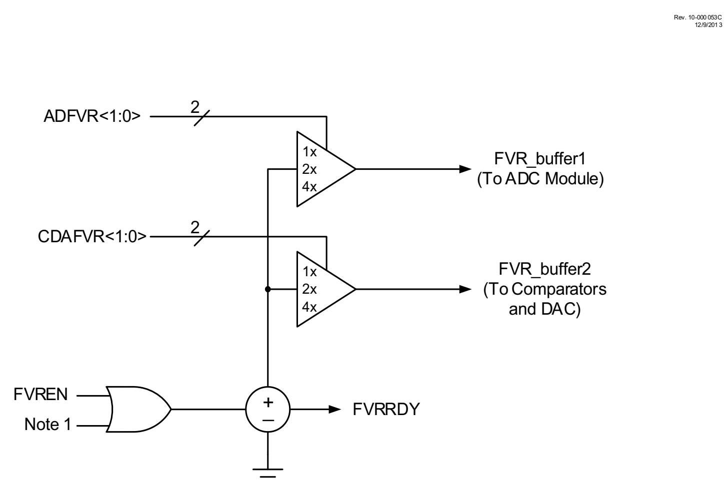
The ADFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module. Reference **Section 22.0 "Analog-to-Digital Converter (ADC) Module"** for additional information.

The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the DAC and comparator module. Reference **Section 24.0 "5-bit Digital-to-Analog Converter (DAC1) Module"** and **Section 18.0 "Comparator Module"** for additional information.

16.2 FVR Stabilization Period

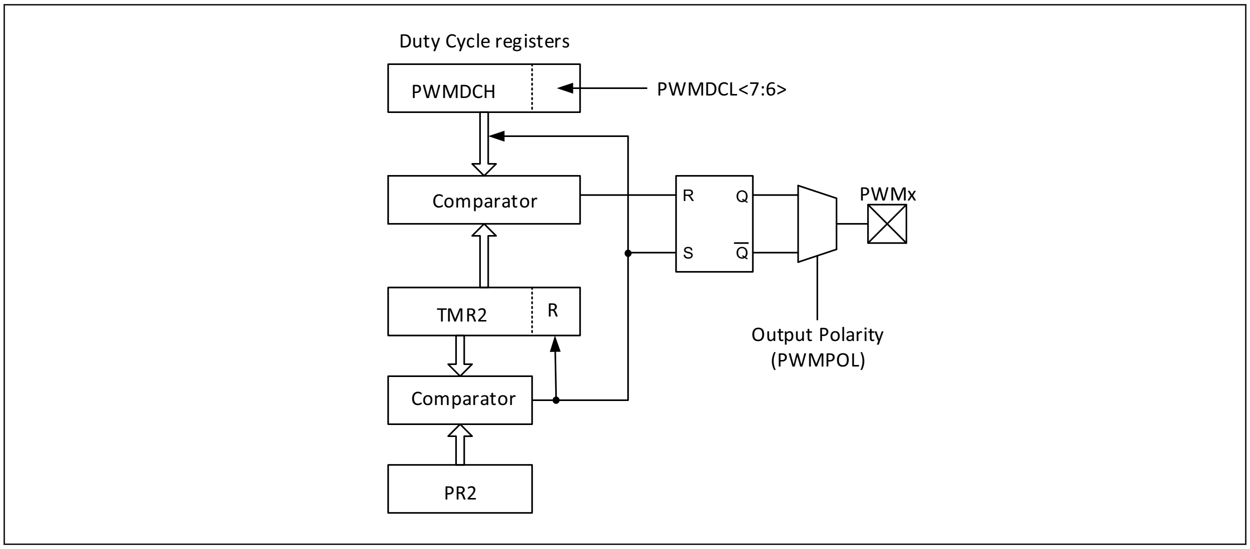
When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. See Table 35-16 for FVR start-up times.

FIGURE 16-1: VOLTAGE REFERENCE BLOCK DIAGRAM



Note: Any peripheral requiring the fixed reference (see Table 16-1).

FIGURE 19-2: SIMPLIFIED PWM BLOCK DIAGRAM



19.1.1 PWM PERIOD

Referring to Figure 19-1, the PWM output has a period and a pulse width. The frequency of the PWM is the inverse of the period (1/period).

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

EQUATION 19-1: PWM PERIOD

$$PWM\ Period = [(PR2) + 1] \bullet 4 \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

Note: TOSC = 1/FOSC

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The PWMx pin is set (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM pulse width is latched from PWMxDC.

Note:	If the pulse-width value is greater than the period, the assigned PWM pin(s) will remain unchanged.

19.1.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the PWMxDC register. The PWMxDCH contains the eight MSbs and bits <7:6> of the PWMxDCL register contain the two LSbs.

The PWMDC register is double-buffered and can be updated at any time. This double buffering is essential for glitch-free PWM operation. New values take effect when TMR2 = PR2. Note that PWMDC is left-justified.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

Equation 19-2 is used to calculate the PWM pulse width.

Equation 19-3 is used to calculate the PWM duty cycle ratio.

EQUATION 19-2: PULSE WIDTH

$$Pulse\ Width = (PWMxDC) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

EQUATION 19-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(PWMxDC)}{4(PR2 + 1)}$$

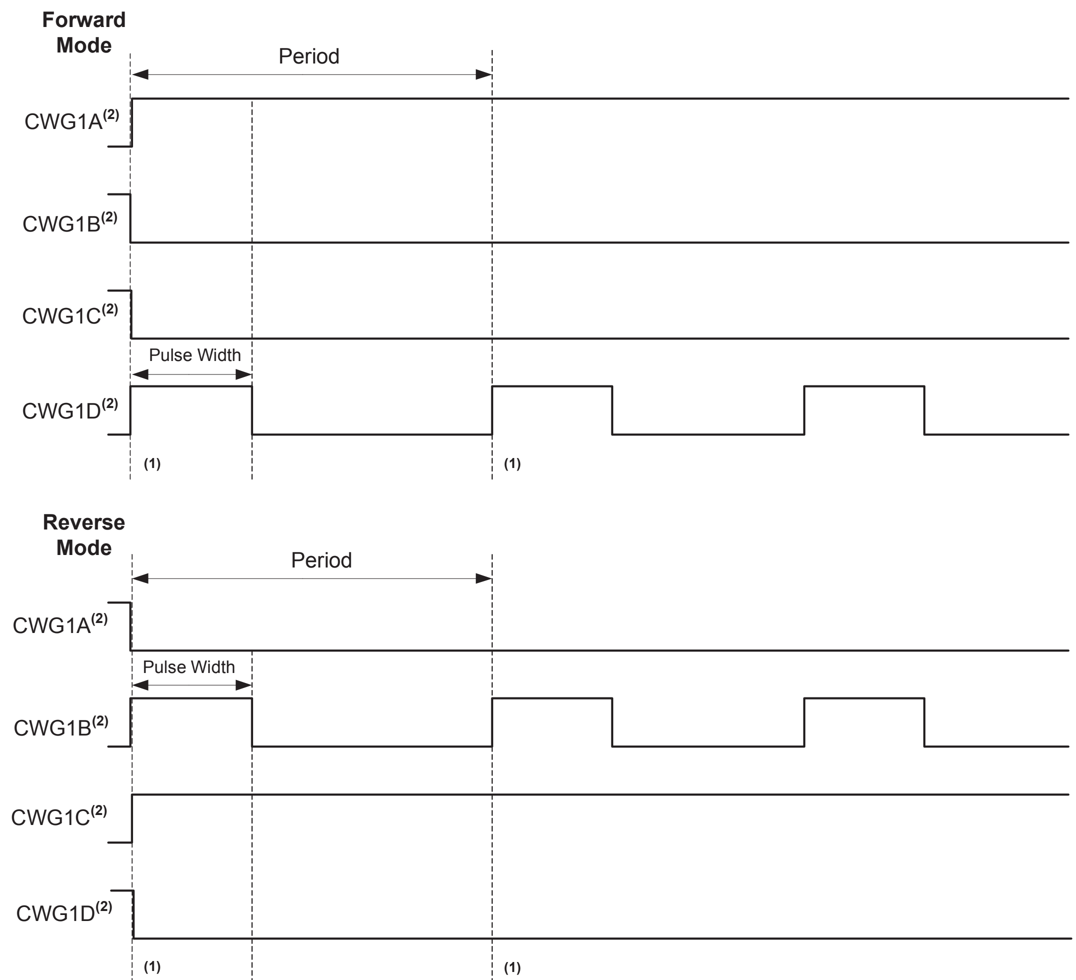
20.2.4.1 Full-Bridge Forward Mode

In Full-Bridge Forward mode (MODE<2:0> = `010`), CWGxA is driven to its active state and CWGxD is modulated while CWGxB and CWGxC are driven to their inactive state, as illustrated at the top of Figure 20-6.

20.2.4.2 Full-Bridge Reverse Mode

In Full-Bridge Reverse mode (MODE<2:0> = `011`), CWGxC is driven to its active state and CWGxB is modulated while CWGxA and CWGxD are driven to their inactive state, as illustrated at the bottom of Figure 20-6.

FIGURE 20-6: EXAMPLE OF FULL-BRIDGE OUTPUT



Note 1: A rising CWG1 data input creates a rising event on the modulated output.
2: Output signals shown as active-high; all WGPOLy bits are clear.

20.2.4.3 Direction Change in Full-Bridge Mode

In Full-Bridge mode, changing MODE<2:0> controls the forward/reverse direction. Changes to MODE<2:0> change to the new direction on the next rising edge of the modulated input.

A direction change is initiated in software by changing the MODE<2:0> bits of the WGxCON0 register. The sequence is illustrated in Figure 20-7.

- The associated active output CWGxA and the inactive output CWGxC are switched to drive in the opposite direction.
- The previously modulated output CWGxD is switched to the inactive state, and the previously inactive output CWGxB begins to modulate.
- CWG modulation resumes after the direction-switch dead band has elapsed.

21.7 Register Definitions: CLC Control

REGISTER 21-1: CLCxCON: CONFIGURABLE LOGIC CELL CONTROL REGISTER

R/W-0/0	U-0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
LCxEN	—	LCxOUT	LCxINTP	LCxINTN	LCxMODE<2:0>		
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 **LCxEN:** Configurable Logic Cell Enable bit
1 = Configurable logic cell is enabled and mixing input signals
0 = Configurable logic cell is disabled and has logic zero output

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **LCxOUT:** Configurable Logic Cell Data Output bit
Read-only: logic cell output data, after LCPOL; sampled from CLCxOUT.

bit 4 **LCxINTP:** Configurable Logic Cell Positive Edge Going Interrupt Enable bit
1 = CLCxIF will be set when a rising edge occurs on CLCxOUT
0 = CLCxIF will not be set

bit 3 **LCxINTN:** Configurable Logic Cell Negative Edge Going Interrupt Enable bit
1 = CLCxIF will be set when a falling edge occurs on CLCxOUT
0 = CLCxIF will not be set

bit 2-0 **LCxMODE<2:0>:** Configurable Logic Cell Functional Mode bits
111 = Cell is 1-input transparent latch with S and R
110 = Cell is J-K flip-flop with R
101 = Cell is 2-input D flip-flop with R
100 = Cell is 1-input D flip-flop with S and R
011 = Cell is S-R latch
010 = Cell is 4-input AND
001 = Cell is OR-XOR
000 = Cell is AND-OR

22.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Result formatting

22.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin will be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 12.0 "I/O Ports"** for more information.

Note:	Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

22.1.2 CHANNEL SELECTION

There are several channel selections available:

- Five PORTA pins (RA0-RA2, RA4-RA5)
- Four PORTB pins (RB4-RB7, PIC16(L)F18346 only)
- Six PORTC pins (RC0-RC5, PIC16(L)F18326)
- Eight PORTC pins (RC0-RC7, PIC16(L)F18346 only)
- Temperature Indicator
- DAC output
- Fixed Voltage Reference (FVR)
- VSS (ground)

The CHS<5:0> bits of the ADCON0 register (Register 22-1) determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 22.2 "ADC Operation"** for more information.

Note:	It is recommended that when switching from an ADC channel of a higher voltage to a channel of a lower voltage, that the user selects the Vss channel before connecting to the channel with the lower voltage. If the ADC does not have a dedicated VSS input channel, the VSS selection (DAC1R<4:0> = `b'00000'`) through the DAC output channel can be used. If the DAC is in use, a free input channel can be connected to Vss, and can be used in place of the DAC.

22.1.3 ADC VOLTAGE REFERENCE

The ADPREF<1:0> bits of the ADCON1 register provides control of the positive voltage reference. The positive voltage reference can be:

- VREF+ pin
- VDD
- FVR 2.048V
- FVR 4.096V (Not available on LF devices)

The ADNREF bit of the ADCON1 register provides control of the negative voltage reference. The negative voltage reference can be:

- VREF- pin
- VSS

See **Section 22.0 "Analog-to-Digital Converter (ADC) Module"** for more details on the Fixed Voltage Reference.

22.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS<2:0> bits of the ADCON1 register. There are seven possible clock options:

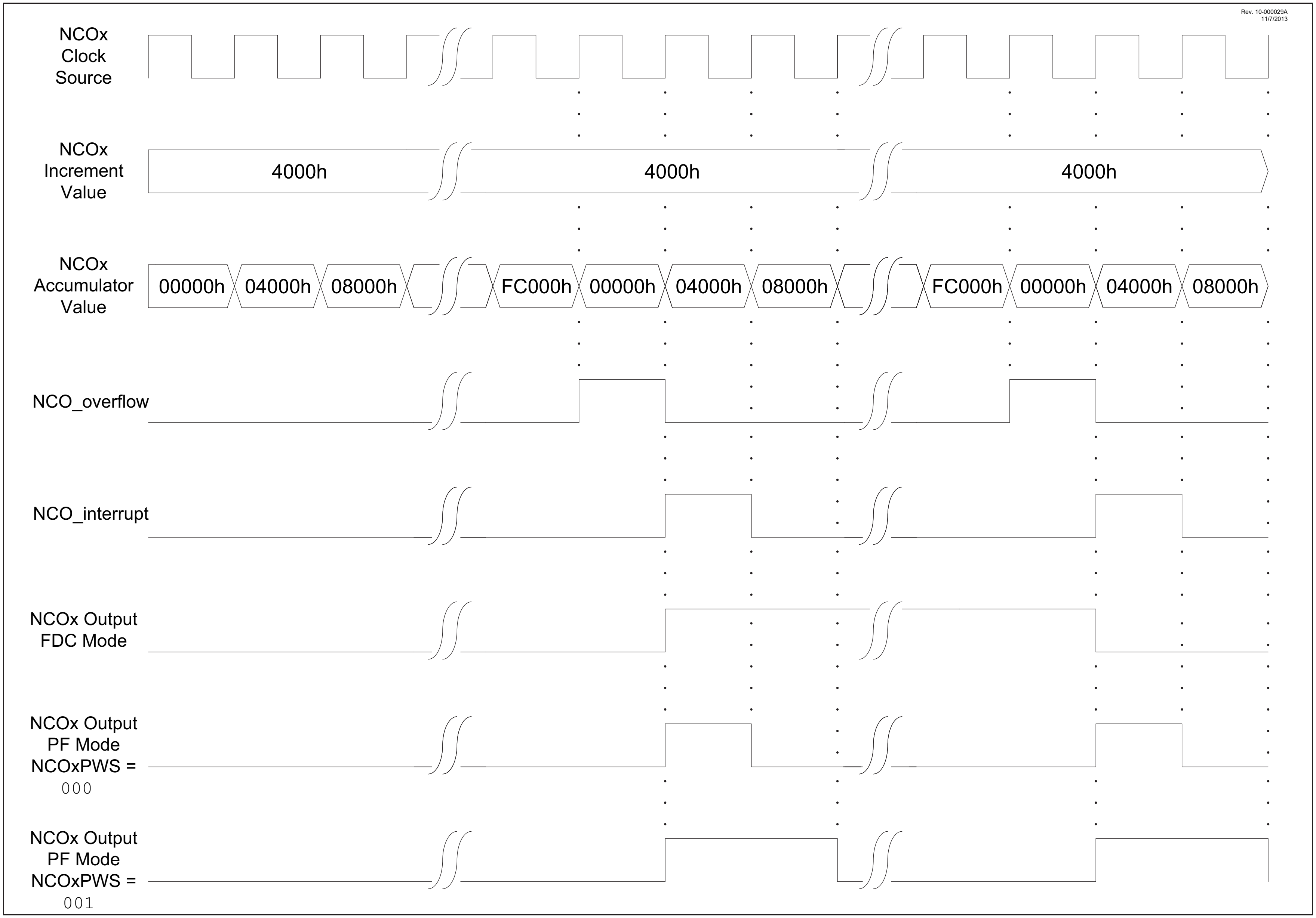
- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- ADCRC (dedicated RC oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 12 TAD periods as shown in Figure 22-2.

For correct conversion, the appropriate TAD specification must be met. Refer to Table 35-13 for more information. Table 22-1 gives examples of appropriate ADC clock selections.

Note:	Unless using the ADCRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

FIGURE 23-2: FDC OUTPUT MODE OPERATION DIAGRAM



30.6.10 SLEEP OPERATION

While in Sleep mode, the I^{2}C slave module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

30.6.11 EFFECTS OF A RESET

A Reset disables the MSSPx module and terminates the current transfer.

30.6.12 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSPx module is disabled. Control of the I^{2}C bus may be taken when the P bit of the SSPxSTAT register is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed by hardware with the result placed in the BCL1IF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

30.6.13 MULTI -MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA, by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin is '0', then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF and reset the I^{2}C port to its Idle state (Figure 30-32).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPxBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I^{2}C bus is free, the user can resume communication by asserting a Start condition.

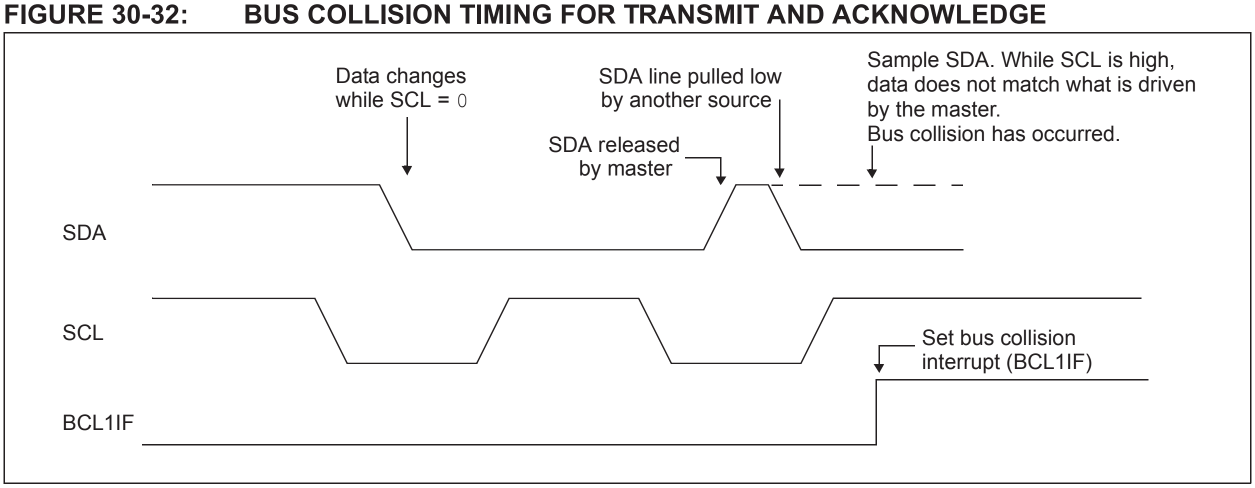
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPxCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I^{2}C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPxIF bit will be set.

A write to the SSPxBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I^{2}C bus can be taken when the P bit is set in the SSPxSTAT register, or the bus is Idle and the S and P bits are cleared.

FIGURE 30-32: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



REGISTER 30-3: SSPxCON2: SSPx CONTROL REGISTER 2 (I^2C MODE ONLY)(1)

R/W-0/0	R/HS/HC-0	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0	R/S/HC-0/0	R/S/HC-0/0	R/S/HC-0/0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
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	HC = Cleared by hardware S = User set

bit 7 **GCEN:** General Call Enable bit (in I^2C Slave mode only)
`1` = Enable interrupt when a general call address (0x00 or 00h) is received in the SSPSR
`0` = General call address disabled

bit 6 **ACKSTAT:** Acknowledge Status bit (in I^2C mode only)
`1` = Acknowledge was not received
`0` = Acknowledge was received

bit 5 **ACKDT:** Acknowledge Data bit (in I^2C mode only)
In Receive mode:
Value transmitted when the user initiates an Acknowledge sequence at the end of a receive
`1` = Not Acknowledge
`0` = Acknowledge

bit 4 **ACKEN:** Acknowledge Sequence Enable bit (in I^2C Master mode only)
In Master Receive mode:
`1` = Initiate Acknowledge sequence on SDA and SCL pins, and transmit ACKDT data bit. Automatically cleared by hardware.
`0` = Acknowledge sequence idle

bit 3 **RCEN:** Receive Enable bit (in I^2C Master mode only)
`1` = Enables Receive mode for I^2C
`0` = Receive idle

bit 2 **PEN:** Stop Condition Enable bit (in I^2C Master mode only)
`1` = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.
`0` = Stop condition Idle

bit 1 **RSEN:** Repeated Start Condition Enable bit (in I^2C Master mode only)
`1` = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.
`0` = Repeated Start condition Idle

bit 0 **SEN:** Start Condition Enable/Stretch Enable bit
In Master mode:
`1` = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.
`0` = Start condition Idle
In Slave mode:
`1` = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)
`0` = Clock stretching is disabled

Note 1: For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I^2C module is not in the idle state, these bits may not be set (no spooling) and the SPPxBUF may not be written.

TABLE 35-11: RESET, WATCHDOG TIMER, OSCILLATOR, START-UP TIMER, POWER-UP TIMER, BROWN-OUT RESET AND LOW POWER BROWN-OUT RESET SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ.†**	**Max.**	**Units**	**Conditions**
RST01	TMCLR	$\overline{\text{MCLR}}$ Pulse Width Low to ensure Reset	2	—	—	μs	
RST02	TIOZ	I/O high-impedance from Reset detection	—	—	2	μs	
RST03	TWDT	Watchdog Timer Time-out Period	10	16	27	ms	16 ms Nominal Reset Time
RST04*	TPWRT	Power-up Timer Period	40	65	140	ms	
RST05	TOST	Oscillator Start-up Timer Period(1,2)	—	1024	—	TOSC	**(Note3)**
RST06	VBOR	Brown-out Reset Voltage(4)	2.55 2.30 1.80	2.70 2.45 1.90	2.85 2.60 2.10	V V V	BORV = 0 BORV = 1 (PIC16F18326/18346) BORV = 1 (PIC16LF18326/18346)
RST07	VBORHYS	Brown-out Reset Hysteresis	0	25	75	mV	
RST08	TBORDC	Brown-out Reset Response Time	1	3	35	μs	
RST09	VLPBOR	Low-Power Brown-out Reset Voltage	1.8	2.1	2.5	V	PIC16LF18326/18346

* 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: Instruction cycle period (TCY) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min” values with an external clock applied to the OSC1 pin. When an external clock input is used, the “max” cycle time limit is “DC” (no clock) for all devices.

2: By design.

3: Period of the slower clock.

4: To ensure these voltage 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.

37.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
  - MPLAB Xpress IDE Software
  - Microchip Code Configurator (MCC)
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

37.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

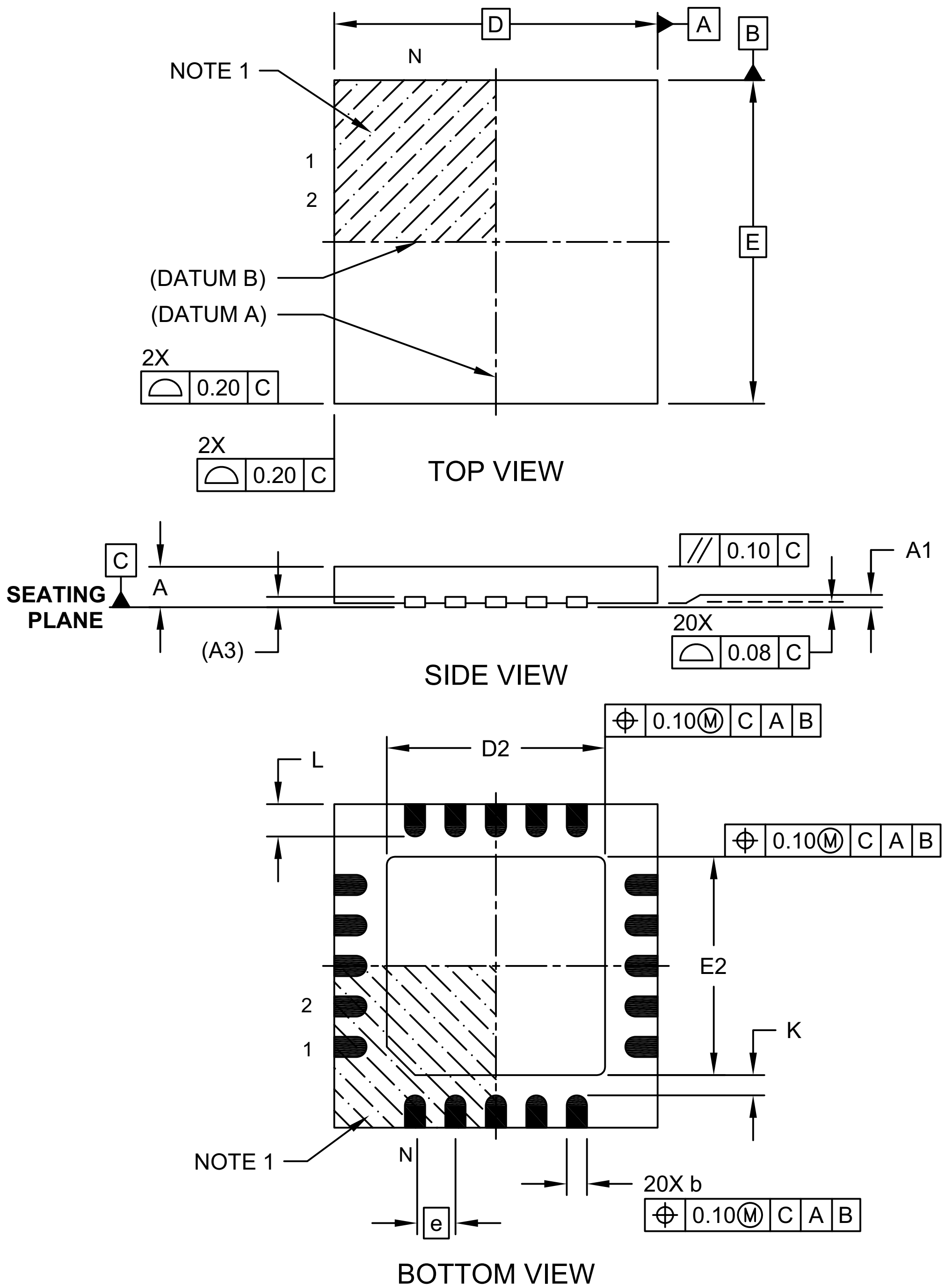
- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

20-Lead Ultra Thin Plastic Quad Flat, No Lead Package (GZ) - 4x4x0.5 mm Body [UQFN]

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



Microchip Technology Drawing C04-255A Sheet 1 of 2