



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	16MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	3
Program Memory Size	896B (512 x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	64 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 3x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	SOT-23-6
Supplier Device Package	SOT-23-6
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic10lf322t-e-ot

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

PIC10(L)F320/322

1.0 DEVICE OVERVIEW

The PIC10(L)F320/322 are described within this data sheet. They are available in 6/8-pin packages. Figure 1-1 shows a block diagram of the PIC10(L)F320/322 devices. Table 1-2 shows the pinout descriptions.

Reference Table 1-1 for peripherals available per device.

TABLE 1-1:DEVICE PERIPHERALSUMMARY

Peripheral		PIC10(L)F320	PIC10(L)F322
Analog-to-Digital Converter	(ADC)	•	•
Configurable Logic Cell (CL	•	•	
Complementary Wave Gene	•	•	
Fixed Voltage Reference (F	٠	•	
Numerically Controlled Osci	illator (NCO)	٠	•
Temperature Indicator		٠	•
PWM Modules			
	PWM1	٠	•
	•	•	
Timers			
	Timer0	•	•
	Timer2	٠	•

2.0 MEMORY ORGANIZATION

These devices contain the following types of memory:

- Program Memory
 - Configuration Word
 - Device ID
 - User ID
 - Flash Program Memory
- Data Memory
 - Core Registers
 - Special Function Registers
 - General Purpose RAM
 - Common RAM

The following features are associated with access and control of program memory and data memory:

- PCL and PCLATH
- Stack
- Indirect Addressing

TABLE 2-1: DEVICE SIZES AND ADDRESSES

2.1	Program	Memory	Organization
A	i i ogi um	moniory	organization

The mid-range core has a 13-bit program counter capable of addressing 8K x 14 program memory space. This device family only implements up to 512 words of the 8K program memory space. Table 2-1 shows the memory sizes implemented for the PIC10(L)F320/322 family. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figures 2-1, and 2-2).

Device	Device Program Memory Space (Words)		High-Endurance Flash Memory Address Range ⁽¹⁾		
PIC10(L)F320	256	00FFh	0080h-00FFh		
PIC10(L)F322	512	01FFh	0180h-01FFh		

Note 1: High-endurance Flash applies to low byte of each address in the range.

2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-3 shows the two situations for the loading of the PC. The upper example in Figure 2-3 shows how the PC is loaded on a write to PCL (PCLATH<4:0> \rightarrow PCH). The lower example in Figure 2-3 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> \rightarrow PCH).

FIGURE 2-3: LOADING OF PC IN DIFFERENT SITUATIONS



2.3.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<12:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by writing the desired upper five bits to the PCLATH register. When the lower eight bits are written to the PCL register, all 13 bits of the program counter will change to the values contained in the PCLATH register and those being written to the PCL register.

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). Care should be exercised when jumping into a look-up table or program branch table (computed GOTO) by modifying the PCL register. Assuming that PCLATH is set to the table start address, if the table length is greater than 255 instructions or if the lower eight bits of the memory address rolls over from 0xFF to 0x00 in the middle of the table, then PCLATH must be incremented for each address rollover that occurs between the table beginning and the target location within the table.

For more information refer to Application Note AN556, *"Implementing a Table Read"* (DS00556).

2.3.2 STACK

All devices have an 8-level x 13-bit wide hardware stack (see Figure 2-1). The stack space is not part of either program or data space and the Stack Pointer is not readable or writable. The PC is PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1:	There are no Status bits to indicate Stack Overflow or Stack Underflow conditions.
2:	There are no instructions/mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL, RETURN, RETLW and RETFIE instructions or the vectoring to an interrupt address.

2.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR and the IRP bit of the STATUS register, as shown in Figure 2-4.

A simple program to clear RAM location 40h-7Fh using indirect addressing is shown in Example 2-1.

EXAMPLE 2-1:	INDIRECT ADDRESSING
--------------	---------------------

	MOVLW	0x40	;initialize pointer
	MOVWF	FSR	;to RAM
NEXT	CLRF	INDF	;clear INDF register
	INCF	FSR	;inc pointer
	BTFSS	FSR,7	;all done?
	GOTO	NEXT	;no clear next
CONTIN	IUE		;yes continue

4.0 OSCILLATOR MODULE

4.1 Overview

The oscillator module has a variety of clock sources and selection features that allow it to be used in a range of applications while maximizing performance and minimizing power consumption. Figure 4-1 illustrates a block diagram of the oscillator module.

The system can be configured to use an internal calibrated high-frequency oscillator as clock source, with a choice of selectable speeds via software.

Clock source modes are configured by the FOSC bit in Configuration Word (CONFIG).

- 1. EC oscillator from CLKIN.
- 2. INTOSC oscillator, CLKIN not enabled.





5.4 Low-Power Brown-out Reset (LPBOR)

The Low-Power Brown-Out Reset (LPBOR) is an essential part of the Reset subsystem. Refer to Figure 5-1 to see how the BOR interacts with other modules.

The LPBOR is used to monitor the external VDD pin. When too low of a voltage is detected, the device is held in Reset. When this occurs, a register bit ($\overline{\text{BOR}}$) is changed to indicate that a BOR Reset has occurred. The same bit is set for both the BOR and the LPBOR. Refer to Register 5-2.

5.4.1 ENABLING LPBOR

The LPBOR is controlled by the LPBOR bit of Configuration Word. When the device is erased, the LPBOR module defaults to enabled.

5.4.1.1 LPBOR Module Output

The output of the LPBOR module is a signal indicating whether or not a Reset is to be asserted. This signal is OR'd together with the Reset signal of the BOR module to provide the generic BOR signal which goes to the PCON register and to the power control block.

5.5 MCLR

The $\overline{\text{MCLR}}$ is an optional external input that can reset the device. The $\overline{\text{MCLR}}$ function is controlled by the MCLRE and the LVP bit of Configuration Word (Table 5-2).

TABLE 5-2: MCLR CONFIGURATION

MCLRE	LVP	MCLR
0	0	Disabled
1	0	Enabled
x	1	Enabled

5.5.1 MCLR ENABLED

When MCLR is enabled and the pin is held low, the device is held in Reset. The MCLR pin is connected to VDD through an internal weak pull-up.

The device has a noise filter in the $\overline{\text{MCLR}}$ Reset path. The filter will detect and ignore small pulses.

```
Note: A Reset does not drive the \overline{MCLR} pin low.
```

5.5.2 MCLR DISABLED

When MCLR is disabled, the pin functions as a general purpose input and the internal weak pull-up is under software control.

5.6 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a CLRWDT instruction within the time-out period. The TO and PD bits in the STATUS register are changed to indicate the WDT Reset. See **Section 8.0** "**Watchdog Timer**" for more information.

5.7 Programming Mode ICSP Exit

Upon exit of Programming mode, the device will behave as if a POR had just occurred.

5.8 Power-Up Timer

The Power-up Timer optionally delays device execution after a BOR or POR event. This timer is typically used to allow VDD to stabilize before allowing the device to start running.

The Power-up Timer is controlled by the $\overrightarrow{\text{PWRTE}}$ bit of Configuration Word.

5.9 Start-up Sequence

Upon the release of a POR or BOR, the following must occur before the device will begin executing:

- 1. Power-up Timer runs to completion (if enabled).
- 2. MCLR must be released (if enabled).

The total time-out will vary based on oscillator configuration and Power-up Timer configuration. See **Section 4.0 "Oscillator Module"** for more information.

The Power-up Timer runs independently of MCLR Reset. If MCLR is kept low long enough, the Power-up Timer will expire. Upon bringing MCLR high, the device will begin execution after 10 FOSC cycles (see Figure 5-3). This is useful for testing purposes or to synchronize more than one device operating in parallel.

PIC10(L)F320/322



- 2: Asynchronous interrupt latency = 3-5 TCY. Synchronous latency = 3-4 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.
- 3: For minimum width of INT pulse, refer to AC specifications in Section 24.0 "Electrical Specifications".
- 4: INTF is enabled to be set any time during the Q4-Q1 cycles.

6.3 Interrupts During Sleep

Some interrupts can be used to wake from Sleep. To wake from Sleep, the peripheral must be able to operate without the system clock. The interrupt source must have the appropriate Interrupt Enable bit(s) set prior to entering Sleep.

On waking from Sleep, if the GIE bit is also set, the processor will branch to the interrupt vector. Otherwise, the processor will continue executing instructions after the SLEEP instruction. The instruction directly after the SLEEP instruction will always be executed before branching to the ISR. Refer to the Section 7.0 "Power-Down Mode (Sleep)" for more details.

6.4 INT Pin

The INT pin can be used to generate an asynchronous edge-triggered interrupt. This interrupt is enabled by setting the INTE bit of the INTCON register. The INTEDG bit of the OPTION_REG register determines on which edge the interrupt will occur. When the INTEDG bit is set, the rising edge will cause the interrupt. When the INTEDG bit is clear, the falling edge will cause the interrupt. The INTF bit of the INTCON register will be set when a valid edge appears on the INT pin. If the GIE and INTE bits are also set, the processor will redirect program execution to the interrupt vector.

6.5 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (e.g., W and STATUS registers). This must be implemented in software.

Temporary holding registers W_TEMP and STATUS_TEMP should be placed in the last 16 bytes of GPR (see Table 1-2). This makes context save and restore operations simpler. The code shown in Example 6-1 can be used to:

- Store the W register
- · Store the STATUS register
- · Execute the ISR code
- Restore the Status (and Bank Select Bit register)
- · Restore the W register
- Note: These devices do not require saving the PCLATH. However, if computed GOTOS are used in both the ISR and the main code, the PCLATH must be saved and restored in the ISR.

EXAMPLE 6-1: SAVING STATUS AND W REGISTERS IN RAM

MOVWF	W_TEMP	;Copy W to TEMP register
SWAPF	STATUS,W	;Swap status to be saved into W
		;Swaps are used because they do not affect the status bits
MOVWF	STATUS_TEMP	;Save status to bank zero STATUS_TEMP register
:		
:(ISR)		;Insert user code here
:		
SWAPF	STATUS_TEMP,W	;Swap STATUS_TEMP register into W
		;(sets bank to original state)
MOVWF	STATUS	;Move W into STATUS register
SWAPF	W_TEMP,F	;Swap W_TEMP
SWAPF	W_TEMP,W	;Swap W_TEMP into W

EXAMPLE 9-3: WRITING TO FLASH PROGRAM MEMORY

```
; This write routine assumes the following:
         A valid starting address (the least significant bits = '00')
   ;
         is loaded in ADDRH:ADDRL
   ;
         ADDRH, ADDRL and DATADDR are all located in data memory
   ;
  BANKSEL
              PMADRH
  MOVE ADDRH,W
                 ;Load initial address
  MOVWF PMADRH
                  ;
  MOVF
        ADDRL,W
                  ;
  MOVWF PMADRL
                  ;
        DATAADDR,W ;Load initial data address
  MOVE
  MOVWF FSR
                  ;
LOOP MOVF INDF,W
                 ;Load first data byte into lower
  MOVWF PMDATL
                 ;
  INCF
       FSR,F
                 ;Next byte
                 ;Load second data byte into upper
  MOVF
        INDF,W
       PMDATH
  MOVWF
                  ;
  INCF
        FSR,F
                  ;
  BANKSEL PMCON1
  BSF PMCON1,WREN ;Enable writes
       INTCON,GIE ;Disable interrupts (if using)
  BCF
  BTFSC INTCON, GIE ; See AN576
  GOTO
       $-2
  ;
        Required Sequence
               ;Start of required write sequence:
  MOVLW 55h
  MOVWF
        PMCON2
                  ;Write 55h
  MOVLW
        0AAh
                 ;Write OAAh
  MOVWF
        PMCON2
        PMCON1,WR ;Set WR bit to begin write
  BSF
  NOP
                  ;Required to transfer data to the buffer
  NOP
                  ;registers
  BCF
       PMCON1,WREN ;Disable writes
        INTCON,GIE ; Enable interrupts (comment out if not using interrupts)
  BSF
  BANKSEL PMADRL
  MOVF
        PMADRL, W
        PMADRL, F
  INCF
                  ;Increment address
  ANDLW 0x03
                  ;Indicates when sixteen words have been programmed
  SUBLW 0x03
                  ;Change value for different size write blocks
                  ;0x0F = 16 words
                  ;0x0B = 12 words
                  ;0x07 = 8 words
                  ;0x03 = 4 words
        STATUS, Z
                  ;Exit on a match,
  BTFSS
  GOTO
        LOOP
                  ;Continue if more data needs to be written
```

W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0		
	Program Memory Control Register 2								
bit 7							bit 0		
Legend:									
R = Readable b	bit	W = Writable I	bit	U = Unimpler	nented bit, read	l as '0'			
S = Bit can only	/ be set	x = Bit is unkn	iown	-n/n = Value	at POR and BO	R/Value at all c	other Resets		
'1' = Bit is set		'0' = Bit is clea	ared						

REGISTER 9-6: PMCON2: PROGRAM MEMORY CONTROL 2 REGISTER

bit 7-0 Flash Memory Unlock Pattern bits

To unlock writes, a 55h must be written first, followed by an AAh, before setting the WR bit of the PMCON1 register. The value written to this register is used to unlock the writes. There are specific timing requirements on these writes.

TABLE 9-3: SUMMARY OF REGISTERS ASSOCIATED WITH FLASH PROGRAM MEMORY

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	40
PMCON1	_	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	65
PMCON2	Program Memory Control Register 2								
PMADRL	PMADR<7:0>								64
PMADRH	PMADR8								64
PMDATL	PMDAT<7:0>								63
PMDATH	_	_			PMDAT	<13:8>			63

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Flash program memory module.

TABLE 9-4: SUMMARY OF CONFIGURATION WORD WITH FLASH PROGRAM MEMORY

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
	13:8	—	—	—	WRT<1:0>		BORV	LPBOR	LVP	20
CONFIG	7:0	CP	MCLR	PWRTE	WDTE<1:0>		BORE	N<1:0>	FOSC	20

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Flash program memory.

10.2 Register Definitions: PORTA

REGISTER 10-1: PORTA: PORTA REGISTER

U-0	U-0	U-0	U-0	R-x/x	R/W-x/x	R/W-x/x	R/W-x/x
—	—	—	—	RA3	RA2	RA1	RA0
bit 7							bit 0

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

bit 7-4 Unimplemented: Read as '0'

bit 3-0 RA<3:0>: PORTA I/O Value bits (RA3 is read-only)

Note 1: Writes to PORTx are actually written to the corresponding LATx register. Reads from PORTx register return actual I/O pin values.

REGISTER 10-2: TRISA: PORTA TRI-STATE REGISTER

U-0	U-0	U-0	U-0	U-1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	—	—	(1)	TRISA2	TRISA1	TRISA0
bit 7							bit 0

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

bit 7-4	Unimplemented: Read as '0'.
bit 3	Unimplemented: Read as '1'.
bit 2-0	TRISA<2:0>: RA<2:0> Port I/O Tri-State Control bits
	1 = Port output driver is disabled0 = Port output driver is enabled

Note 1: Unimplemented, read as '1'.

REGISTER 10-3: LATA: PORTA DATA LATCH REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u
_	—	—	—	_	LATA2	LATA1	LATA0
bit 7							bit 0
Legend:							
R = Readable b	oit	W = Writable I	bit	U = Unimpler	mented 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 7-3 Unimplemented: Read as '0'.

bit 2-0 LATA<2:0>: RA<2:0> Output Latch Value bits

Note 1: Writes to PORTx are actually written to the corresponding LATx register. Reads from LATx register return register values, not I/O pin values.

REGISTER 10-4: ANSELA: PORTA ANALOG SELECT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	_		— — ANSA2		ANSA1	ANSA0	
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 7-3 Unimplemented: Read as '0'.

bit 2-0 ANSA<2:0>: Analog Select between Analog or Digital Function on Pins RA<2:0>, respectively

- 1 = Analog input. Pin is assigned as analog input⁽¹⁾. Digital Input buffer disabled.
- 0 = Digital I/O. Pin is assigned to port or Digital special function.
- **Note 1:** Setting a pin to an analog input automatically disables the digital input circuitry. Weak pull-ups, if available, are unaffected. The corresponding TRIS bit must be set to Input mode by the user in order to allow external control of the voltage on the pin.

12.0 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 can be configured to supply a reference voltage to the following:

ADC input channel

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

12.1 Independent Gain Amplifiers

The output of the FVR supplied to the ADC is routed through an independent programmable gain amplifier. The amplifier can be configured to amplify the reference voltage by 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 15.0** "**Analog-to-Digital Converter** (**ADC**) **Module**" for additional information.

To minimize current consumption when the FVR is disabled, the FVR buffers should be turned off by clearing the ADFVR<1:0> bits.

12.2 FVR Stabilization Period

When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the FVRRDY bit of the FVRCON register will be set. See **Section 24.0** "**Electrical Specifications**" for the minimum delay requirement.

FIGURE 12-1: VOLTAGE REFERENCE BLOCK DIAGRAM



TABLE 12-1: PERIPHERALS REQUIRING THE FIXED VOLTAGE REFERENCE (FVR)

Peripheral	Conditions	Description
HFINTOSC	FOSC = 1	EC on CLKIN pin.
	BOREN<1:0> = 11	BOR always enabled.
BOR	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.
IVR	All PIC10F320/322 devices, when VREGPM1 = 1 and not in Sleep	The device runs off of the Power-Save mode regulator when in Sleep mode.

14.0 TEMPERATURE INDICATOR MODULE

This family of devices is equipped with a temperature circuit designed to measure the operating temperature of the silicon die. The circuit's range of operating temperature falls between of -40° C and $+85^{\circ}$ C. The output is a voltage that is proportional to the device temperature. The output of the temperature indicator is internally connected to the device ADC.

The circuit may be used as a temperature threshold detector or a more accurate temperature indicator, depending on the level of calibration performed. A one-point calibration allows the circuit to indicate a temperature closely surrounding that point. A two-point calibration allows the circuit to sense the entire range of temperature more accurately. Reference Application Note AN1333, *"Use and Calibration of the Internal Temperature Indicator"* (DS01333) for more details regarding the calibration process.

14.1 Circuit Operation

Figure 14-1 shows a simplified block diagram of the temperature circuit. The proportional voltage output is achieved by measuring the forward voltage drop across multiple silicon junctions.

Equation 14-1 describes the output characteristics of the temperature indicator.

EQUATION 14-1: VOUT RANGES

High Range: VOUT = VDD - 4VT

Low Range: VOUT = VDD - 2VT

The temperature sense circuit is integrated with the Fixed Voltage Reference (FVR) module. See **Section 12.0 "Fixed Voltage Reference (FVR)"** for more information.

The circuit is enabled by setting the TSEN bit of the FVRCON register. When disabled, the circuit draws no current.

The circuit operates in either high or low range. The high range, selected by setting the TSRNG bit of the FVRCON register, provides a wider output voltage. This provides more resolution over the temperature range, but may be less consistent from part to part. This range requires a higher bias voltage to operate and thus, a higher VDD is needed.

The low range is selected by clearing the TSRNG bit of the FVRCON0 register. The low range generates a lower voltage drop and thus, a lower bias voltage is needed to operate the circuit. The low range is provided for low voltage operation.

FIGURE 14-1: TEMPERATURE CIRCUIT DIAGRAM



14.2 Minimum Operating VDD vs. Minimum Sensing Temperature

When the temperature circuit is operated in low range, the device may be operated at any operating voltage that is within specifications.

When the temperature circuit is operated in high range, the device operating voltage, VDD, must be high enough to ensure that the temperature circuit is correctly biased.

Table 14-1 shows the recommended minimum $\mathsf{V}\mathsf{D}\mathsf{D}$ vs. range setting.

TABLE 14-1: RECOMMENDED VDD VS. RANGE

Min. VDD, TSRNG = 1	Min. VDD, TSRNG = 0
3.6V	1.8V

14.3 Temperature Output

The output of the circuit is measured using the internal Analog-to-Digital Converter. A channel is reserved for the temperature circuit output. Refer to **Section 15.0 "Analog-to-Digital Converter (ADC) Module"** for detailed information.

14.4 ADC Acquisition Time

To ensure accurate temperature measurements, the user must wait at least 200 μ s after the ADC input multiplexer is connected to the temperature indicator output before the conversion is performed. In addition, the user must wait 200 μ s between sequential conversions of the temperature indicator output.

^{© 2011-2015} Microchip Technology Inc.

PIC10(L)F320/322

						···· ··· ·· ··		•••••••••••••••••••••••••••••••••••••••	•
Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	—	—	ADFV	R<1:0>	78
ADCON		ADCS<2:0>	•		CHS<2:0>		GO/ DONE	ADON	88
ADRES A/D Result Register							89		

TABLE 14-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE TEMPERATURE INDICATOR

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the temperature indicator module.

15.1.5 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:	The ADIF bit is set at the completion of
	every conversion, regardless of whether
	or not the ADC interrupt is enabled.

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 GIE and PEIE bits of the INTCON register must be disabled. If the GIE and PEIE bits of the INTCON register are enabled, execution will switch to the Interrupt Service Routine.

15.2 ADC Operation

15.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON register must be set to a '1'. Setting the GO/ DONE bit of the ADCON register to a '1' will start the Analog-to-Digital conversion.

Note:	The GO/DONE bit should not be set in the					
	same instruction that turns on the ADC.					
	Refer to Section 15.2.5 "A/D Conver-					
	sion Procedure".					

15.2.2 COMPLETION OF A CONVERSION

When the conversion is complete, the ADC module will:

- Clear the GO/DONE bit
- · Set the ADIF Interrupt Flag bit
- Update the ADRES register with new conversion result

15.2.3 TERMINATING A CONVERSION

If a conversion must be terminated before completion, the GO/DONE bit can be cleared in software. The ADRES register will be updated with the partially complete Analog-to-Digital conversion sample. Incomplete bits will match the last bit converted.

Note:	A device Reset forces all registers to their						
	Reset state. Thus, the ADC module is						
	turned off and any pending conversion is						
	terminated.						

15.2.4 ADC OPERATION DURING SLEEP

The ADC module can operate during Sleep. This requires the ADC clock source to be set to the FRC option. When the FRC clock source is selected, the ADC waits one additional instruction before starting the conversion. This allows the SLEEP instruction to be executed, which can reduce system noise during the conversion. If the ADC interrupt is enabled, the device will wake-up from Sleep when the conversion completes. If the ADC interrupt is disabled, the ADC module is turned off after the conversion completes, although the ADON bit remains set.

When the ADC clock source is something other than FRC, a SLEEP instruction causes the present conversion to be aborted and the ADC module is turned off, although the ADON bit remains set.

TABLE 15-2: SUMMARY OF REGISTERS ASSOCIATED WITH ADC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON	ADCS<2:0>			CHS<2:0>			GO/DONE	ADON	88
ADRES	ADRES<7:0>						89		
ANSELA	—	—	—	—	—	ANSA2	ANSA1	ANSA0	70
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	_	—	ADFVR<1:0>		78
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	40
PIE1	—	ADIE	—	NCO1IE	CLC1IE	—	TMR2IE	-	41
PIR1	—	ADIF	—	NCO1IF	CLC1IF	—	TMR2IF	_	42
TRISA	_	_	_	_	_	TRISA2	TRISA1	TRISA0	69

Legend: x = unknown, u = unchanged, — = unimplemented read as '0', q = value depends on condition. Shaded cells are not used for ADC module.

19.0 CONFIGURABLE LOGIC CELL (CLC)

The Configurable Logic Cell (CLCx) provides programmable logic that operates outside the speed limitations of software execution. The logic cell selects any combination of the eight input signals and through the use of configurable gates reduces the selected inputs to four logic lines that drive one of eight selectable single-output logic functions.

Input sources are a combination of the following:

- Two I/O pins
- Internal clocks
- · Peripherals
- Register bits

The output can be directed internally to peripherals and to an output pin.

Refer to Figure 19-1 for a simplified diagram showing signal flow through the CLCx.

Possible configurations include:

- Combinatorial Logic
 - AND
 - NAND
 - AND-OR
 - AND-OR-INVERT
 - OR-XOR
 - OR-XNOR
- Latches
 - S-R
 - Clocked D with Set and Reset
 - Transparent D with Set and Reset
 - Clocked J-K with Reset



FIGURE 19-1: CLCx SIMPLIFIED BLOCK DIAGRAM

20.1 NCOx OPERATION

The NCOx operates by repeatedly adding a fixed value to an accumulator. Additions occur at the input clock rate. The accumulator will overflow with a carry periodically, which is the raw NCOx output. This effectively reduces the input clock by the ratio of the addition value to the maximum accumulator value. See Equation 20-1.

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

The NCOx output creates an instantaneous frequency, which may cause uncertainty. This output depends on the ability of the receiving circuit (i.e., CWG or external resonant converter circuitry) to average the instantaneous frequency to reduce uncertainty.

20.1.1 NCOx CLOCK SOURCES

Clock sources available to the NCOx include:

- HFINTOSC
- Fosc
- LC1OUT
- NCO1CLK pin

The NCOx clock source is selected by configuring the NxCKS<1:0> bits in the NCOxCLK register.

20.1.2 ACCUMULATOR

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

- NCOxACCL
- NCOxACCH
- NCOxACCU

EQUATION 20-1:



2'

n = Accumulator width in bits

20.1.3 ADDER

The NCOx Adder is a full adder, which operates asynchronously to the clock source selected. The addition of the previous result and the increment value replaces the accumulator value on the rising edge of each input clock.

20.1.4 INCREMENT REGISTERS

The Increment value is stored in two 8-bit registers making up a 16-bit increment. In order of LSB to MSB they are:

- NCOxINCL
- NCOxINCH

Both of the registers are readable and writable. The Increment registers are double-buffered to allow for value changes to be made without first disabling the NCOx module.

The buffer loads are immediate when the module is disabled. Writing to the MS register first is necessary because then the buffer is loaded synchronously with the NCOx operation after the write is executed on the lower increment register.

Note: The increment buffer registers are not useraccessible.

REGISTER 21-3: CWGxCON2: CWG CONTROL REGISTER 2

R/W/HC/HS-0/0	R/W-0/0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
GxASE	GxARSEN	_	_	_	—	GxASDCLC1	GxASDFLT
bit 7							bit 0

Legend:					
HC = Bit is cleared by hardw	are	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			

utputs 5xOEA
IXOEA

THE MICROCHIP WEBSITE

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

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

CUSTOMER CHANGE NOTIFICATION SERVICE

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

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

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

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

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

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