

Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

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

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	5
Program Memory Size	1.75KB (1K 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 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	8-VDFN Exposed Pad
Supplier Device Package	8-DFN (2x3)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic12lf1501-e-mc">https://www.e-xfl.com/product-detail/microchip-technology/pic12lf1501-e-mc</a>

# PIC12(L)F1501

**TABLE 3-5: SPECIAL FUNCTION REGISTER SUMMARY**

Address	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
Bank 0											
00Ch	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--xx xxxx
00Dh to 010h	—	Unimplemented								—	—
011h	PIR1	TMR1GIF	ADIF	—	—	—	—	TMR2IF	TMR1IF	00-- --00	00-- --00
012h	PIR2	—	—	C1IF	—	—	NCO1IF	—	—	--0- -0--	--0- -0--
013h	PIR3	—	—	—	—	—	—	CLC2IF	CLC1IF	---- --00	---- --00
014h	—	Unimplemented								—	—
015h	TMR0	Holding Register for the 8-bit Timer0 Count								xxxx xxxx	uuuu uuuu
016h	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Count								xxxx xxxx	uuuu uuuu
017h	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Count								xxxx xxxx	uuuu uuuu
018h	T1CON	TMR1CS<1:0>		T1CKPS<1:0>		—	T1SYNC	—	TMR1ON	0000 -0-0	uuuu -u-u
019h	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>		0000 0x00	uuuu uxuu
01Ah	TMR2	Timer2 Module Register								0000 0000	0000 0000
01Bh	PR2	Timer2 Period Register								1111 1111	1111 1111
01Ch	T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		-000 0000	-000 0000
01Dh to 01Fh	—	Unimplemented								—	—
Bank 1											
08Ch	TRISA	—	—	TRISA5	TRISA4	— <sup>(2)</sup>	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
08Dh to 090h	—	Unimplemented								—	—
091h	PIE1	TMR1GIE	ADIE	—	—	—	—	TMR2IE	TMR1IE	00-- --00	00-- --00
092h	PIE2	—	—	C1IE	—	—	NCO1IE	—	—	--0- -0--	--0- -0--
093h	PIE3	—	—	—	—	—	—	CLC2IE	CLC1IE	---- --00	---- --00
094h	—	Unimplemented								—	—
095h	OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			1111 1111	1111 1111
096h	PCON	STKOVF	STKUNF	—	RWD $\overline{T}$	RMCLR	R $\overline{I}$	POR	BOR	00-1 11qq	qq-q qquu
097h	WDTCON	—	—	WDTPS<4:0>					SWDTEN	--01 0110	--01 0110
098h	—	Unimplemented								—	—
099h	OSCCON	—	IRCF<3:0>				—	SCS<1:0>		-011 1-00	-011 1-00
09Ah	OSCSTAT	—	—	—	HFIOFR	—	—	LFIOFR	HFIOFS	---0 --00	---q --qq
09Bh	ADRESL	ADC Result Register Low								xxxx xxxx	uuuu uuuu
09Ch	ADRESH	ADC Result Register High								xxxx xxxx	uuuu uuuu
09Dh	ADCON0	—	CHS<4:0>					GO/DONE	ADON	-000 0000	-000 0000
09Eh	ADCON1	ADFM	ADCS<2:0>			—	—	ADPREF<1:0>		0000 --00	0000 --00
09Fh	ADCON2	TRIGSEL<3:0>				—	—	—	—	0000 ----	0000 ----

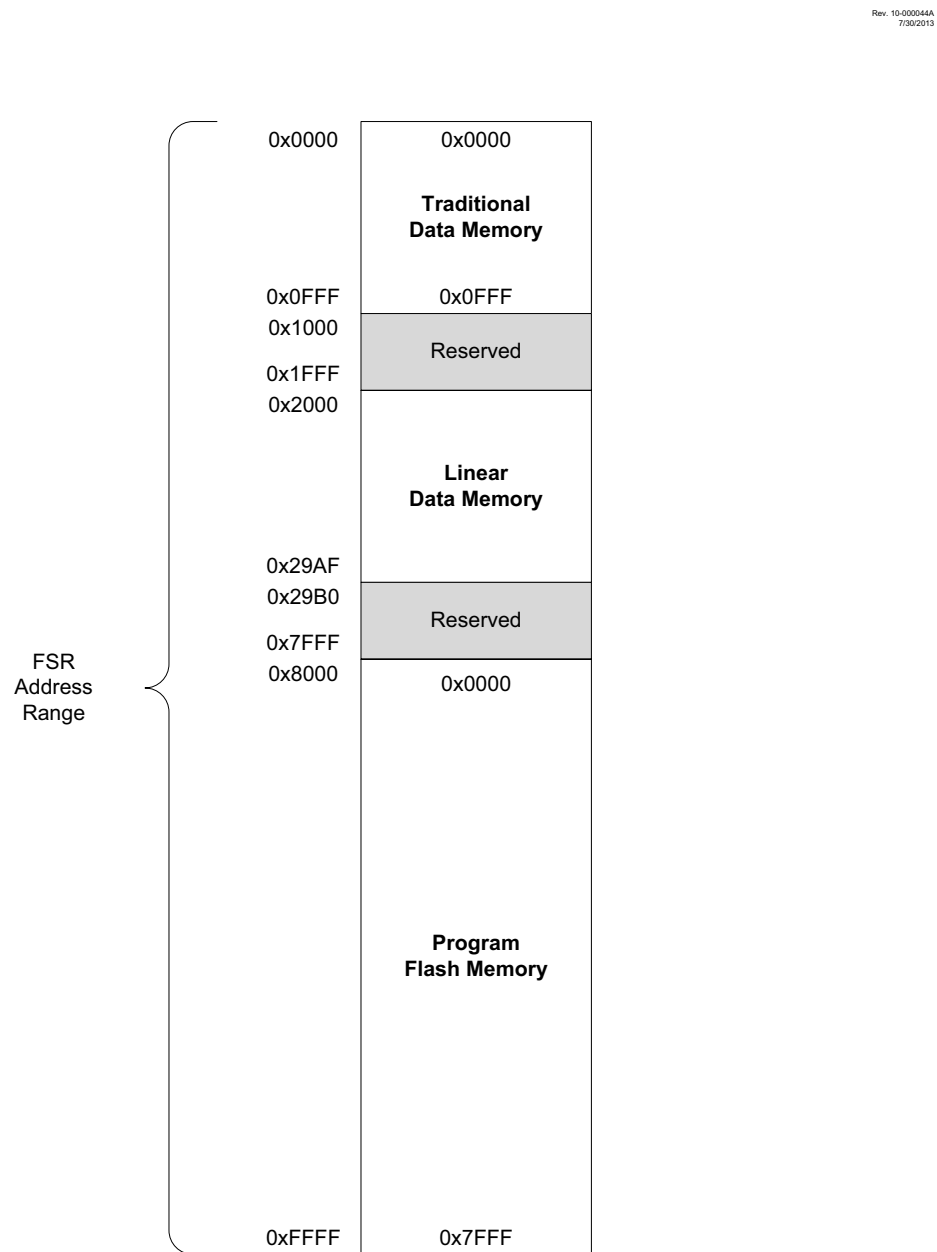
**Legend:** x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, r = reserved. Shaded locations are unimplemented, read as '0'.

**Note 1:** PIC12F1501 only.

**Note 2:** Unimplemented, read as '1'.

# PIC12(L)F1501

### FIGURE 3-8: INDIRECT ADDRESSING



Note: Not all memory regions are completely implemented. Consult device memory tables for memory limits.

## 4.0 DEVICE CONFIGURATION

Device configuration consists of Configuration Words, Code Protection and Device ID.

### 4.1 Configuration Words

There are several Configuration Word bits that allow different oscillator and memory protection options. These are implemented as Configuration Word 1 at 8007h and Configuration Word 2 at 8008h.

<p><b>Note:</b> The <math>\overline{\text{DEBUG}}</math> bit in Configuration Words is managed automatically by device development tools including debuggers and programmers. For normal device operation, this bit should be maintained as a '1'.</p>
--

## 5.2 Clock Source Types

Clock sources can be classified as external, internal or peripheral.

External clock sources rely on external circuitry for the clock source to function. Examples are: oscillator modules (ECH, ECM, ECL modes).

Internal clock sources are contained within the oscillator module. The internal oscillator block has two internal oscillators that are used to generate the internal system clock sources: the 16 MHz High-Frequency Internal Oscillator (HFINTOSC) and the 31 kHz Low-Frequency Internal Oscillator (LFINTOSC).

The peripheral clock source is a nominal 600 kHz internal RC oscillator, FRC. The FRC is traditionally used with the ADC module, but is sometimes available to other peripherals. See **Section 5.2.2.4 “Peripheral Clock Sources”**.

The system clock can be selected between external or internal clock sources via the System Clock Select (SCS) bits in the OSCCON register. See **Section 5.3 “Clock Switching”** for additional information.

### 5.2.1 EXTERNAL CLOCK SOURCES

An external clock source can be used as the device system clock by performing one of the following actions:

- Program the FOSC<1:0> bits in the Configuration Words to select an external clock source that 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:
  - Secondary oscillator during run-time, or
  - An external clock source determined by the value of the FOSC bits.

See **Section 5.3 “Clock Switching”** for more information.

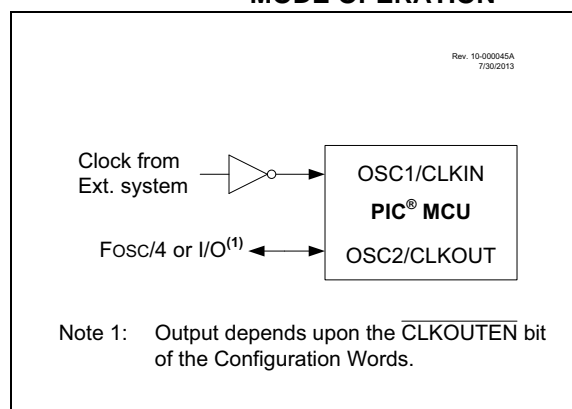
#### 5.2.1.1 EC Mode

The External Clock (EC) mode allows an externally generated logic level signal to be the system clock source. When operating in this mode, an external clock source is connected to the OSC1 input. OSC2/CLKOUT is available for general purpose I/O or CLKOUT. Figure 5-2 shows the pin connections for EC mode.

EC mode has three power modes to select from through the Fosc bits in the Configuration Words:

- ECH – High power, 4-20 MHz
- ECM – Medium power, 0.5-4 MHz
- ECL – Low power, 0-0.5 MHz

**FIGURE 5-2: EXTERNAL CLOCK (EC) MODE OPERATION**



## 10.0 FLASH PROGRAM MEMORY CONTROL

The Flash program memory is readable and writable during normal operation over the full VDD range. Program memory is indirectly addressed using Special Function Registers (SFRs). The SFRs used to access program memory are:

- PMCON1
- PMCON2
- PMDATL
- PMDATH
- PMADRL
- PMADRH

When accessing the program memory, the PMDATH:PMDATL register pair forms a 2-byte word that holds the 14-bit data for read/write, and the PMADRH:PMADRL register pair forms a 2-byte word that holds the 15-bit address of the program memory location being read.

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump rated to operate over the operating voltage range of the device.

The Flash program memory can be protected in two ways; by code protection ( $\overline{CP}$  bit in Configuration Words) and write protection (WRT<1:0> bits in Configuration Words).

Code protection ( $\overline{CP} = 0$ )<sup>(1)</sup>, disables access, reading and writing, to the Flash program memory via external device programmers. Code protection does not affect the self-write and erase functionality. Code protection can only be reset by a device programmer performing a Bulk Erase to the device, clearing all Flash program memory, Configuration bits and User IDs.

Write protection prohibits self-write and erase to a portion or all of the Flash program memory, as defined by the bits WRT<1:0>. Write protection does not affect a device programmers ability to read, write or erase the device.

**Note 1:** Code protection of the entire Flash program memory array is enabled by clearing the  $\overline{CP}$  bit of Configuration Words.

### 10.1 PMADRL and PMADRH Registers

The PMADRH:PMADRL register pair can address up to a maximum of 32K words of program memory. When selecting a program address value, the MSB of the address is written to the PMADRH register and the LSB is written to the PMADRL register.

#### 10.1.1 PMCON1 AND PMCON2 REGISTERS

PMCON1 is the control register for Flash program memory accesses.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared by hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation to occur. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and execute the appropriate error handling routine.

The PMCON2 register is a write-only register. Attempting to read the PMCON2 register will return all '0's.

To enable writes to the program memory, a specific pattern (the unlock sequence), must be written to the PMCON2 register. The required unlock sequence prevents inadvertent writes to the program memory write latches and Flash program memory.

### 10.2 Flash Program Memory Overview

It is important to understand the Flash program memory structure for erase and programming operations. Flash program memory is arranged in rows. A row consists of a fixed number of 14-bit program memory words. A row is the minimum size that can be erased by user software.

After a row has been erased, the user can reprogram all or a portion of this row. Data to be written into the program memory row is written to 14-bit wide data write latches. These write latches are not directly accessible to the user, but may be loaded via sequential writes to the PMDATH:PMDATL register pair.

**Note:** If the user wants to modify only a portion of a previously programmed row, then the contents of the entire row must be read and saved in RAM prior to the erase. Then, new data and retained data can be written into the write latches to reprogram the row of Flash program memory. However, any unprogrammed locations can be written without first erasing the row. In this case, it is not necessary to save and rewrite the other previously programmed locations.

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**

Device	Row Erase (words)	Write Latches (words)
PIC12(L)F1501	16	16

## 10.2.4 WRITING TO FLASH PROGRAM MEMORY

Program memory is programmed using the following steps:

1. Load the address in PMADRH:PMADRL of the row to be programmed.
2. Load each write latch with data.
3. Initiate a programming operation.
4. Repeat steps 1 through 3 until all data is written.

Before writing to program memory, the word(s) to be written must be erased or previously unwritten. Program memory can only be erased one row at a time. No automatic erase occurs upon the initiation of the write.

Program memory can be written one or more words at a time. The maximum number of words written at one time is equal to the number of write latches. See Figure 10-5 (row writes to program memory with 16 write latches) for more details.

The write latches are aligned to the Flash row address boundary defined by the upper 10-bits of PMADRH:PMADRL, (PMADRH<6:0>:PMADRL<7:5>) with the lower five bits of PMADRL, (PMADRL<4:0>) determining the write latch being loaded. Write operations do not cross these boundaries. At the completion of a program memory write operation, the data in the write latches is reset to contain 0x3FFF.

The following steps should be completed to load the write latches and program a row of program memory. These steps are divided into two parts. First, each write latch is loaded with data from the PMDATH:PMDATL using the unlock sequence with LWLO = 1. When the last word to be loaded into the write latch is ready, the LWLO bit is cleared and the unlock sequence executed. This initiates the programming operation, writing all the latches into Flash program memory.

**Note:** The special unlock sequence is required to load a write latch with data or initiate a Flash programming operation. If the unlock sequence is interrupted, writing to the latches or program memory will not be initiated.

1. Set the WREN bit of the PMCON1 register.
2. Clear the CFGS bit of the PMCON1 register.
3. Set the LWLO bit of the PMCON1 register. When the LWLO bit of the PMCON1 register is '1', the write sequence will only load the write latches and will not initiate the write to Flash program memory.
4. Load the PMADRH:PMADRL register pair with the address of the location to be written.
5. Load the PMDATH:PMDATL register pair with the program memory data to be written.
6. Execute the unlock sequence (**Section 10.2.2 "Flash Memory Unlock Sequence"**). The write latch is now loaded.
7. Increment the PMADRH:PMADRL register pair to point to the next location.
8. Repeat steps 5 through 7 until all but the last write latch has been loaded.
9. Clear the LWLO bit of the PMCON1 register. When the LWLO bit of the PMCON1 register is '0', the write sequence will initiate the write to Flash program memory.
10. Load the PMDATH:PMDATL register pair with the program memory data to be written.
11. Execute the unlock sequence (**Section 10.2.2 "Flash Memory Unlock Sequence"**). The entire program memory latch content is now written to Flash program memory.

**Note:** The program memory write latches are reset to the blank state (0x3FFF) at the completion of every write or erase operation. As a result, it is not necessary to load all the program memory write latches. Unloaded latches will remain in the blank state.

An example of the complete write sequence is shown in Example 10-3. The initial address is loaded into the PMADRH:PMADRL register pair; the data is loaded using indirect addressing.

## 15.1 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

### 15.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 11.0 “I/O Ports”** for more information.

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

### 15.1.2 CHANNEL SELECTION

There are 7 channel selections available:

- AN<3:0> pins
- Temperature Indicator
- DAC1\_output
- FVR\_buffer1

The CHS bits of the ADCON0 register determine which channel is connected to the sample and hold circuit.

When changing channels, a delay (TACQ) is required before starting the next conversion. Refer to **Section 15.2.6 “ADC Conversion Procedure”** for more information.

### 15.1.3 ADC VOLTAGE REFERENCE

The ADC module uses a positive and a negative voltage reference. The positive reference is labeled ref+ and the negative reference is labeled ref-.

The positive voltage reference (ref+) is selected by the ADPREF bits in the ADCON1 register. The positive voltage reference source can be:

- VREF+ pin
- VDD

The negative voltage reference (ref-) source is:

- VSS

### 15.1.4 CONVERSION CLOCK

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

- Fosc/2
- Fosc/4
- Fosc/8
- Fosc/16
- Fosc/32
- Fosc/64
- FRC (internal RC oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11.5 TAD periods as shown in Figure 15-2.

For correct conversion, the appropriate TAD specification must be met. Refer to the ADC conversion requirements in **Section 27.0 “Electrical Specifications”** for more information. Table 15-1 gives examples of appropriate ADC clock selections.

<b>Note:</b> Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.
--



# PIC12(L)F1501

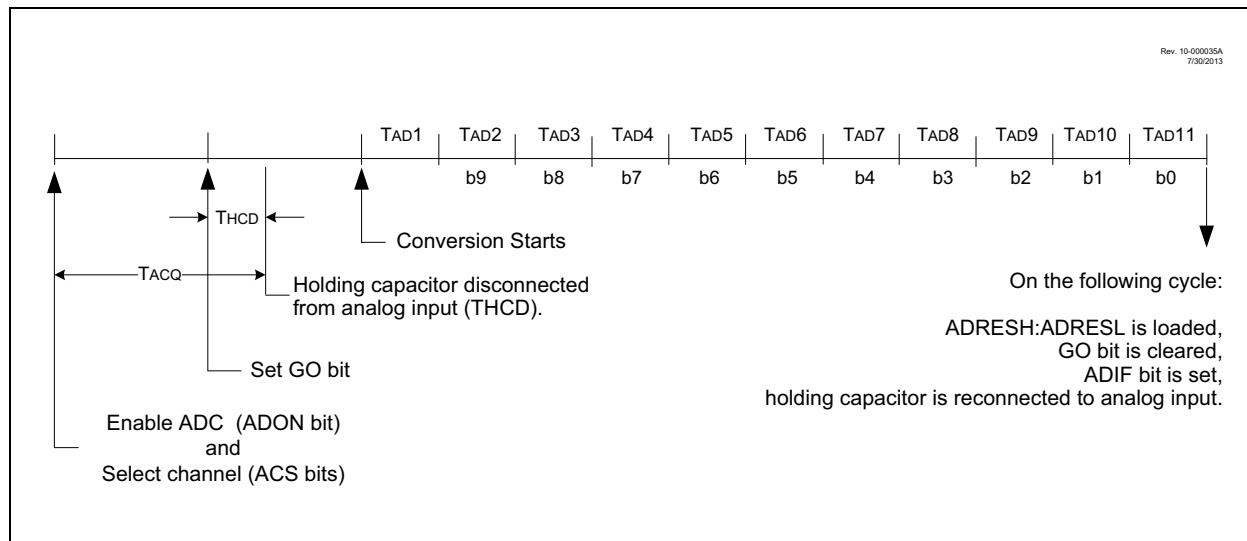
**TABLE 15-1: ADC CLOCK PERIOD (T<sub>AD</sub>) Vs. DEVICE OPERATING FREQUENCIES**

ADC Clock Period (T <sub>AD</sub> )		Device Frequency (F <sub>osc</sub> )				
ADC Clock Source	ADCS<2:0 >	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
F <sub>osc</sub> /2	000	100 ns	125 ns	250 ns	500 ns	2.0 μs
F <sub>osc</sub> /4	100	200 ns	250 ns	500 ns	1.0 μs	4.0 μs
F <sub>osc</sub> /8	001	400 ns	500 ns	1.0 μs	2.0 μs	8.0 μs
F <sub>osc</sub> /16	101	800 ns	1.0 μs	2.0 μs	4.0 μs	16.0 μs
F <sub>osc</sub> /32	010	1.6 μs	2.0 μs	4.0 μs	8.0 μs	32.0 μs
F <sub>osc</sub> /64	110	3.2 μs	4.0 μs	8.0 μs	16.0 μs	64.0 μs
FRC	x11	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs	1.0-6.0 μs

**Legend:** Shaded cells are outside of recommended range.

**Note:** The T<sub>AD</sub> period when using the FRC clock source can fall within a specified range, (see T<sub>AD</sub> parameter). The T<sub>AD</sub> period when using the F<sub>osc</sub>-based clock source can be configured for a more precise T<sub>AD</sub> period. However, the FRC clock source must be used when conversions are to be performed with the device in Sleep mode.

**FIGURE 15-2: ANALOG-TO-DIGITAL CONVERSION T<sub>AD</sub> CYCLES**



# PIC12(L)F1501

**REGISTER 17-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 1**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
CxINTP	CxINTN	CxPCH<1:0>		—	CxNCH<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                      **CxINTP:** Comparator Interrupt on Positive Going Edge Enable bits  
1 = The CxIF interrupt flag will be set upon a positive going edge of the CxOUT bit  
0 = No interrupt flag will be set on a positive going edge of the CxOUT bit
- bit 6                      **CxINTN:** Comparator Interrupt on Negative Going Edge Enable bits  
1 = The CxIF interrupt flag will be set upon a negative going edge of the CxOUT bit  
0 = No interrupt flag will be set on a negative going edge of the CxOUT bit
- bit 5-4                      **CxPCH<1:0>:** Comparator Positive Input Channel Select bits  
11 = CxVP connects to Vss  
10 = CxVP connects to FVR Voltage Reference  
01 = CxVP connects to DAC Voltage Reference  
00 = CxVP connects to CxIN+ pin
- bit 3                      **Unimplemented:** Read as '0'
- bit 2-0                      **CxNCH<2:0>:** Comparator Negative Input Channel Select bits  
111 = Reserved  
110 = Reserved  
101 = Reserved  
100 = CxVN connects to FVR Voltage reference  
011 = Reserved  
010 = Reserved  
001 = CxVN connects to CxIN1- pin  
000 = CxVN connects to CxIN0- pin

**REGISTER 17-3: CMOUT: COMPARATOR OUTPUT REGISTER**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0/0
—	—	—	—	—	—	—	MC1OUT
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 7-1                      **Unimplemented:** Read as '0'
- bit 0                      **MC1OUT:** Mirror Copy of C1OUT bit

## 18.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A software programmable prescaler is available for exclusive use with Timer0. The prescaler is enabled by clearing the PSA bit of the OPTION\_REG register.

**Note:** The Watchdog Timer (WDT) uses its own independent prescaler.

There are eight prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits of the OPTION\_REG register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be disabled by setting the PSA bit of the OPTION\_REG register.

The prescaler is not readable or writable. All instructions writing to the TMR0 register will clear the prescaler.

## 18.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The TMR0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The TMR0IF bit can only be cleared in software. The Timer0 interrupt enable is the TMR0IE bit of the INTCON register.

**Note:** The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep.

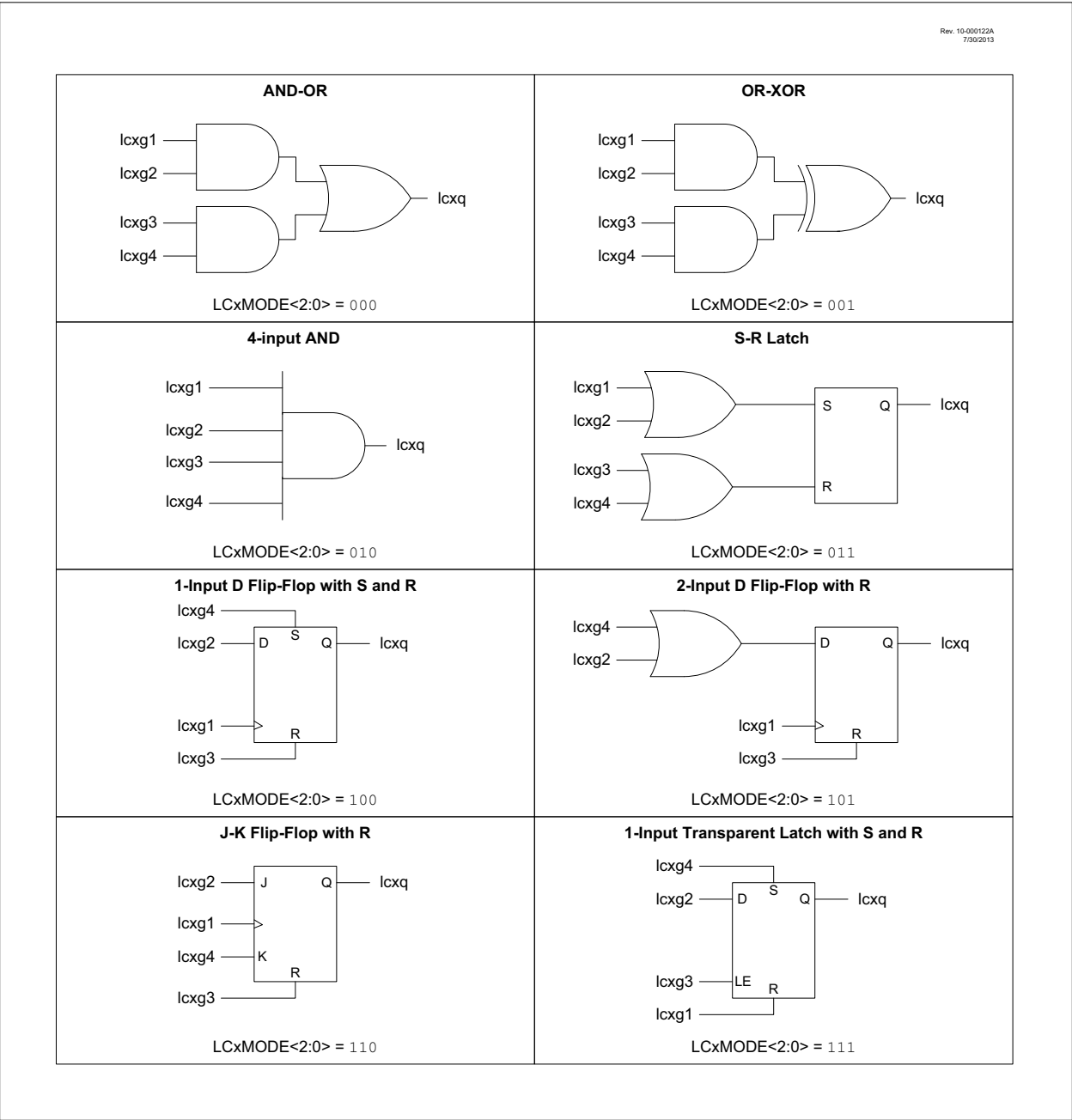
## 18.1.5 8-BIT COUNTER MODE SYNCHRONIZATION

When in 8-Bit Counter mode, the incrementing edge on the T0CKI pin must be synchronized to the instruction clock. Synchronization can be accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the instruction clock. The high and low periods of the external clocking source must meet the timing requirements as shown in **Section 27.0 “Electrical Specifications”**.

## 18.1.6 OPERATION DURING SLEEP

Timer0 cannot operate while the processor is in Sleep mode. The contents of the TMR0 register will remain unchanged while the processor is in Sleep mode.

FIGURE 22-3: PROGRAMMABLE LOGIC FUNCTIONS



## REGISTER 24-3: CWGxCON2: CWG CONTROL REGISTER 2

R/W-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
GxASE	GxARSEN	—	—	—	GxASDSC1	GxASDSFLT	GxASDSCLC2
bit 7						bit 0	

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

u = Bit is unchanged

x = Bit is unknown

-n/n = Value at POR and BOR/Value at all other Resets

'1' = Bit is set

'0' = Bit is cleared

q = Value depends on condition

- bit 7      **GxASE:** Auto-Shutdown Event Status bit  
1 = An auto-shutdown event has occurred  
0 = No auto-shutdown event has occurred
- bit 6      **GxARSEN:** Auto-Restart Enable bit  
1 = Auto-restart is enabled  
0 = Auto-restart is disabled
- bit 5-3    **Unimplemented:** Read as '0'
- bit 2      **GxASDSC1:** CWG Auto-shutdown on Comparator C1 Enable bit  
1 = Shutdown when Comparator C1 output (C1OUT\_async) is high  
0 = Comparator C1 output has no effect on shutdown
- bit 1      **GxASDSFLT:** CWG Auto-shutdown on FLT Enable bit  
1 = Shutdown when  $\overline{\text{CWG1FLT}}$  input is low  
0 =  $\overline{\text{CWG1FLT}}$  input has no effect on shutdown
- bit 0      **GxASDSCLC2:** CWG Auto-shutdown on CLC2 Enable bit  
1 = Shutdown when CLC2 output (LC2\_out) is high  
0 = CLC2 output has no effect on shutdown

## MOVIW Move INDFn to W

**Syntax:** [ *label* ] MOVIW ++FSRn  
[ *label* ] MOVIW --FSRn  
[ *label* ] MOVIW FSRn++  
[ *label* ] MOVIW FSRn--  
[ *label* ] MOVIW k[FSRn]

**Operands:**  $n \in [0,1]$   
 $mm \in [00,01,10,11]$   
 $-32 \leq k \leq 31$

**Operation:** INDFn  $\rightarrow$  W  
Effective address is determined by

- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:

- FSR + 1 (all increments)
- FSR - 1 (all decrements)
- Unchanged

**Status Affected:** Z

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

**Description:** This instruction is used to move data between W and one of the indirect registers (INDFn). Before/after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

**Note:** The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h - FFFFh. Incrementing/decrementing it beyond these bounds will cause it to wrap-around.

## MOVLB Move literal to BSR

**Syntax:** [ *label* ] MOVLB k

**Operands:**  $0 \leq k \leq 31$

**Operation:**  $k \rightarrow$  BSR

**Status Affected:** None

**Description:** The 5-bit literal 'k' is loaded into the Bank Select Register (BSR).

## MOVLP Move literal to PCLATH

**Syntax:** [ *label* ] MOVLP k

**Operands:**  $0 \leq k \leq 127$

**Operation:**  $k \rightarrow$  PCLATH

**Status Affected:** None

**Description:** The 7-bit literal 'k' is loaded into the PCLATH register.

## MOVLW Move literal to W

**Syntax:** [ *label* ] MOVLW k

**Operands:**  $0 \leq k \leq 255$

**Operation:**  $k \rightarrow$  (W)

**Status Affected:** None

**Description:** The 8-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.

**Words:** 1

**Cycles:** 1

**Example:** MOVLW 0x5A  
After Instruction  
W = 0x5A

## MOVWF Move W to f

**Syntax:** [ *label* ] MOVWF f

**Operands:**  $0 \leq f \leq 127$

**Operation:** (W)  $\rightarrow$  (f)

**Status Affected:** None

**Description:** Move data from W register to register 'f'.

**Words:** 1

**Cycles:** 1

**Example:** MOVWF OPTION\_REG  
Before Instruction  
OPTION\_REG = 0xFF  
W = 0x4F  
After Instruction  
OPTION\_REG = 0x4F  
W = 0x4F

**TABLE 27-5: MEMORY PROGRAMMING SPECIFICATIONS**

Standard Operating Conditions (unless otherwise stated)							
Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
<b>Program Memory Programming Specifications</b>							
D110	VIHH	Voltage on $\overline{\text{MCLR}}$ /VPP pin	8.0	—	9.0	V	(Note 2)
D112	VPBE	VDD for Bulk Erase	2.7	—	VDDMAX	V	
D113	VPEW	VDD for Write or Row Erase	VDDMIN	—	VDDMAX	V	
D114	IPPPGM	Current on $\overline{\text{MCLR}}$ /VPP during Erase/Write	—	1.0	—	mA	
D115	IDDPGM	Current on VDD during Erase/Write	—	5.0	—	mA	
<b>Program Flash Memory</b>							
D121	EP	Cell Endurance	10K	—	—	E/W	-40°C ≤ TA ≤ +85°C (Note 1)
D122	VPRW	VDD for Read/Write	VDDMIN	—	VDDMAX	V	
D123	TIW	Self-timed Write Cycle Time	—	2	2.5	ms	
D124	TRETD	Characteristic Retention	—	40	—	Year	
D125	EHEFC	High-Endurance Flash Cell	100K	—	—	E/W	0°C ≤ TA ≤ +60°C, lower byte last 128 addresses

† 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:** Self-write and Block Erase.

**2:** Required only if single-supply programming is disabled.

**TABLE 27-6: THERMAL CONSIDERATIONS**

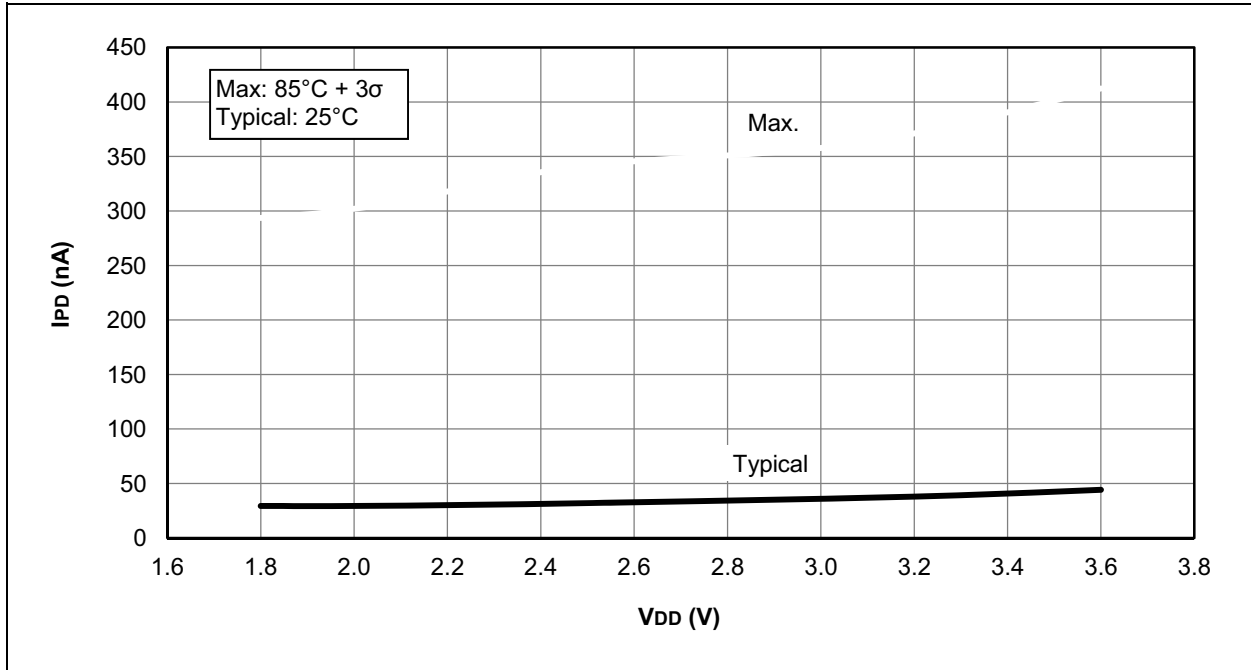
Standard Operating Conditions (unless otherwise stated)					
Operating temperature -40°C ≤ TA ≤ +125°C					
Param No.	Sym.	Characteristic	Typ.	Units	Conditions
TH01	θJA	Thermal Resistance Junction to Ambient	89.3	°C/W	8-pin PDIP package
			149.5	°C/W	8-pin SOIC package
			211	°C/W	8-pin MSOP package
			56.7	°C/W	8-pin DFN 3X3mm package
			68	°C/W	8-pin DFN 2X3mm package
			60	°C/W	8-pin UDFN 2X3mm package
TH02	θJC	Thermal Resistance Junction to Case	43.1	°C/W	8-pin PDIP package
			39.9	°C/W	8-pin SOIC package
			39	°C/W	8-pin MSOP package
			10.7	°C/W	8-pin DFN 3X3mm package
			12.7	°C/W	8-pin DFN 2X3mm package
			11	°C/W	8-pin UDFN 2X3mm package
TH03	TJMAX	Maximum Junction Temperature	150	°C	
TH04	PD	Power Dissipation	—	W	PD = PINTERNAL + PI/O
TH05	PINTERNAL	Internal Power Dissipation	—	W	PINTERNAL = IDD × VDD <sup>(1)</sup>
TH06	PI/O	I/O Power Dissipation	—	W	PI/O = Σ (IOL × VOL) + Σ (IOH × (VDD - VOH))
TH07	PDER	Dertated Power	—	W	PDER = PDMAX (TJ - TA)/θJA <sup>(2)</sup>

**Note 1:** IDD is current to run the chip alone without driving any load on the output pins.

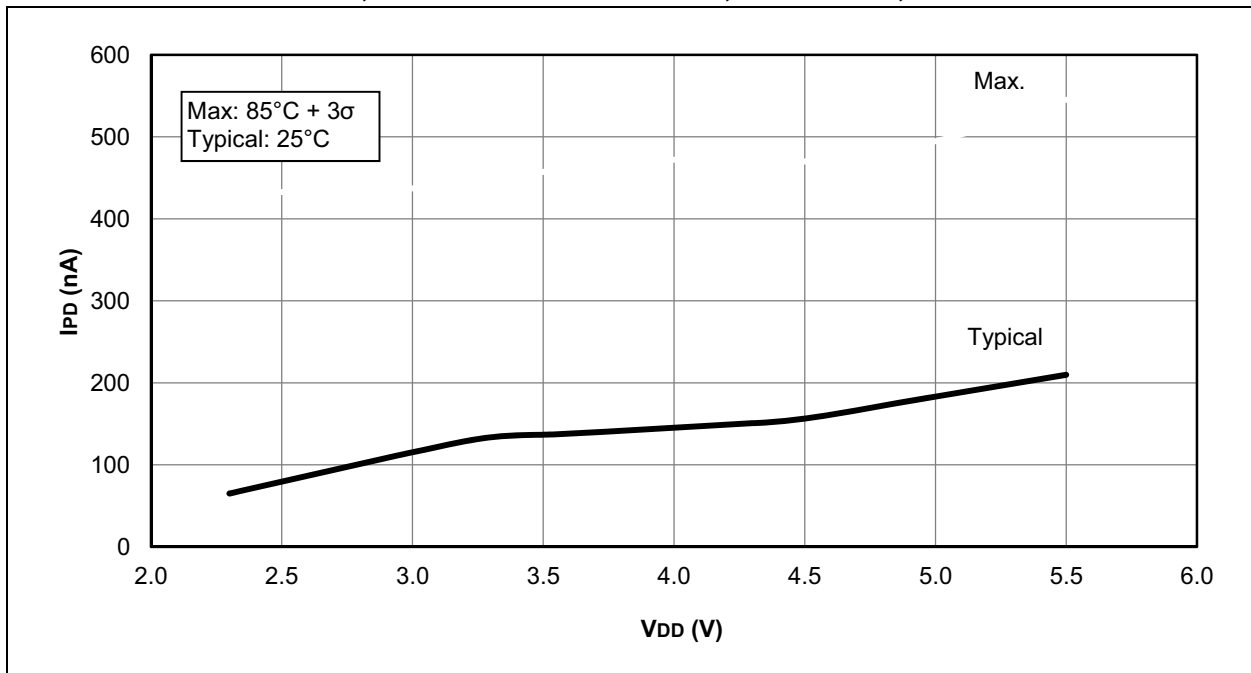
**2:** TA = Ambient Temperature.

**3:** TJ = Junction Temperature.

**FIGURE 28-21: I<sub>PD</sub> BASE, LOW-POWER SLEEP MODE, PIC12LF1501 ONLY**



**FIGURE 28-22: I<sub>PD</sub> BASE, LOW-POWER SLEEP MODE, VREGPM = 1, PIC12F1501 ONLY**





# PIC12(L)F1501

FIGURE 28-27: I<sub>PD</sub>, BROWN-OUT RESET (BOR), BORV = 0, PIC12LF1501 ONLY

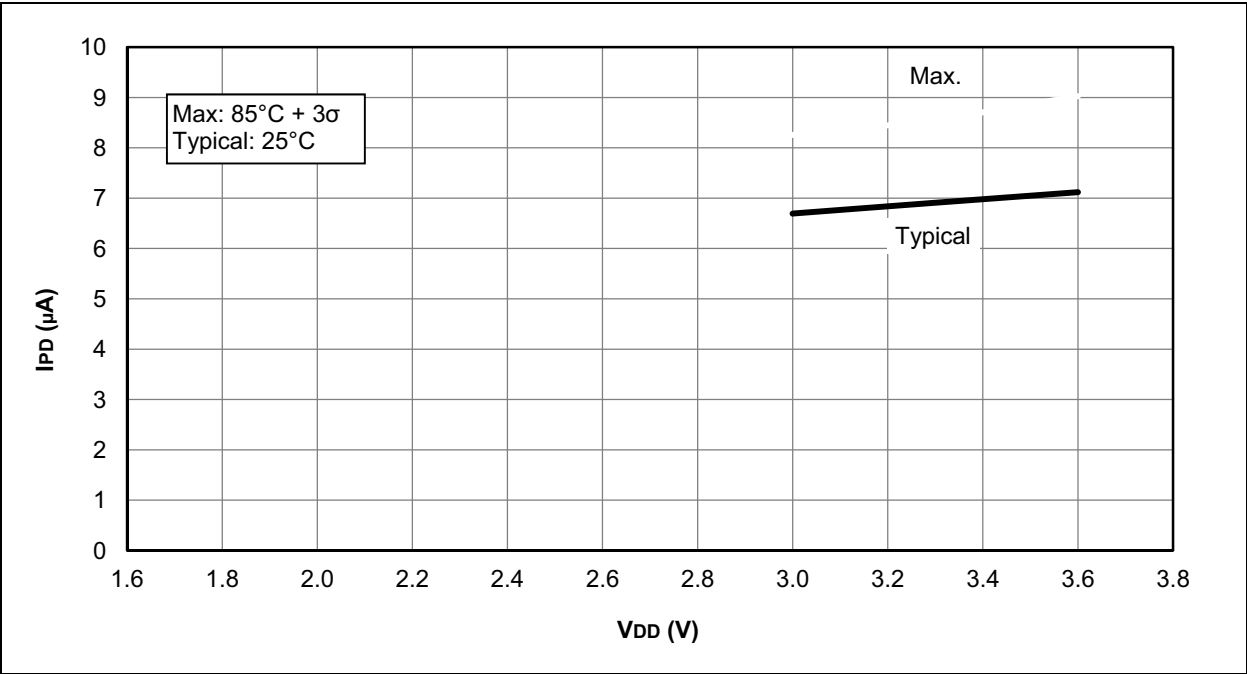
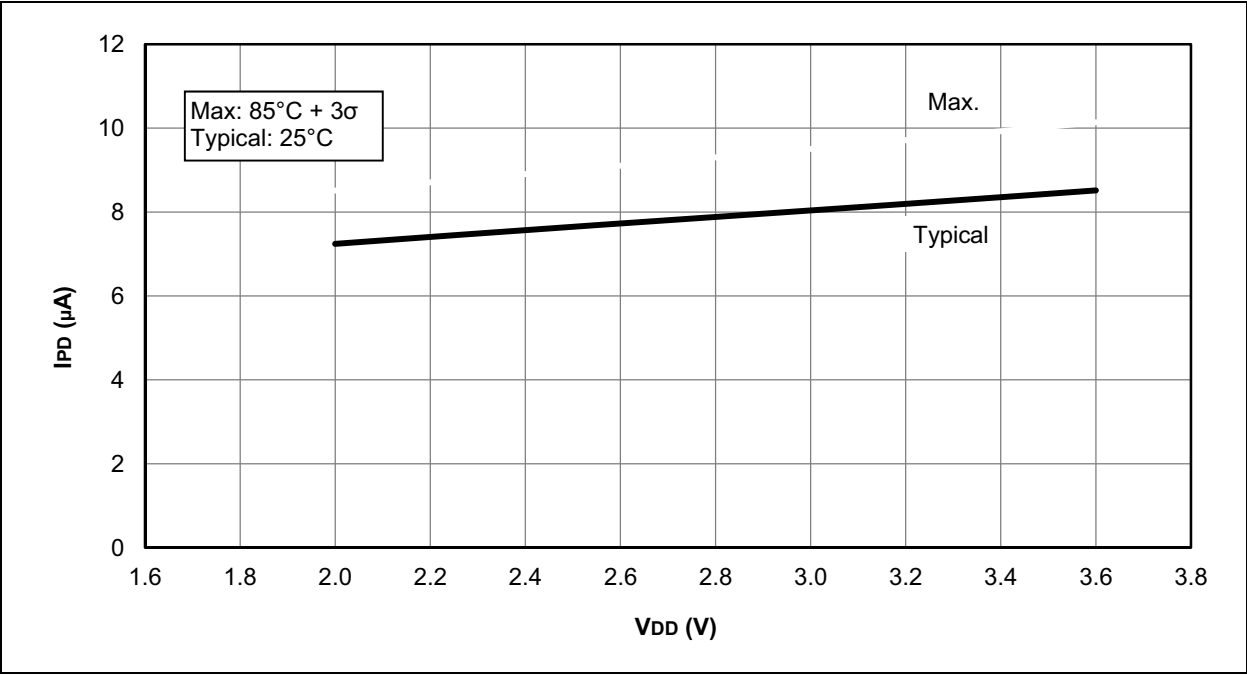


FIGURE 28-28: I<sub>PD</sub>, BROWN-OUT RESET (BOR), BORV = 1, PIC12LF1501 ONLY



# PIC12(L)F1501

FIGURE 28-43: BROWN-OUT RESET VOLTAGE, BORV = 1, PIC12LF1501 ONLY

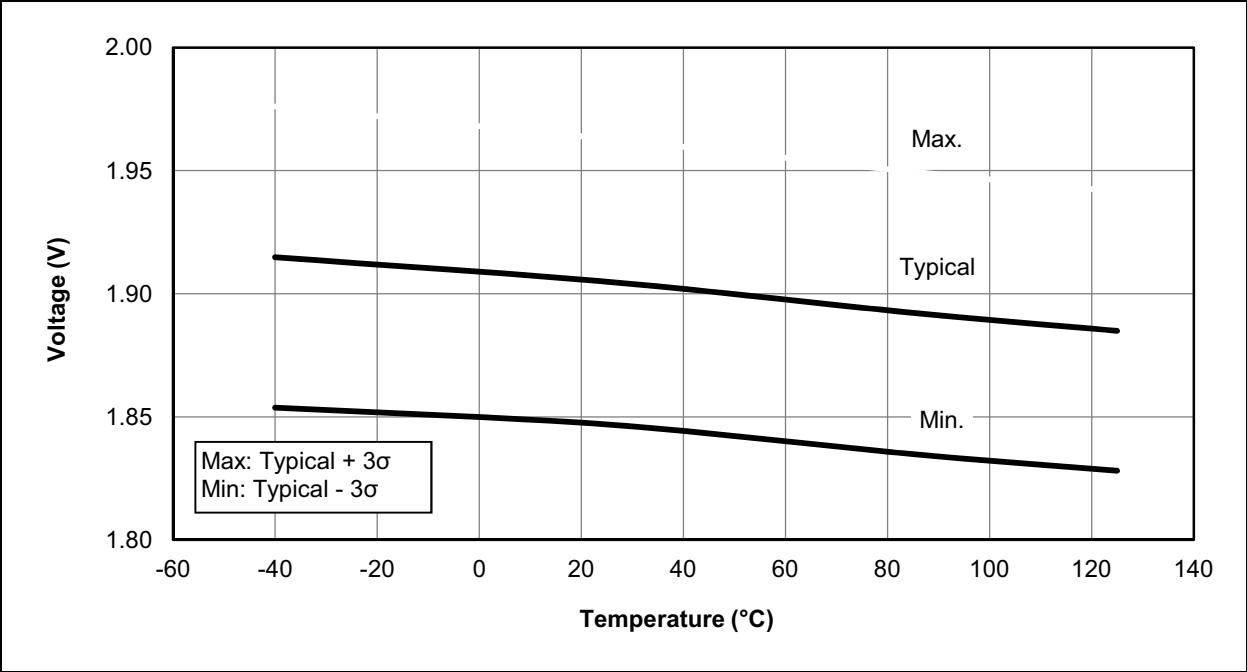
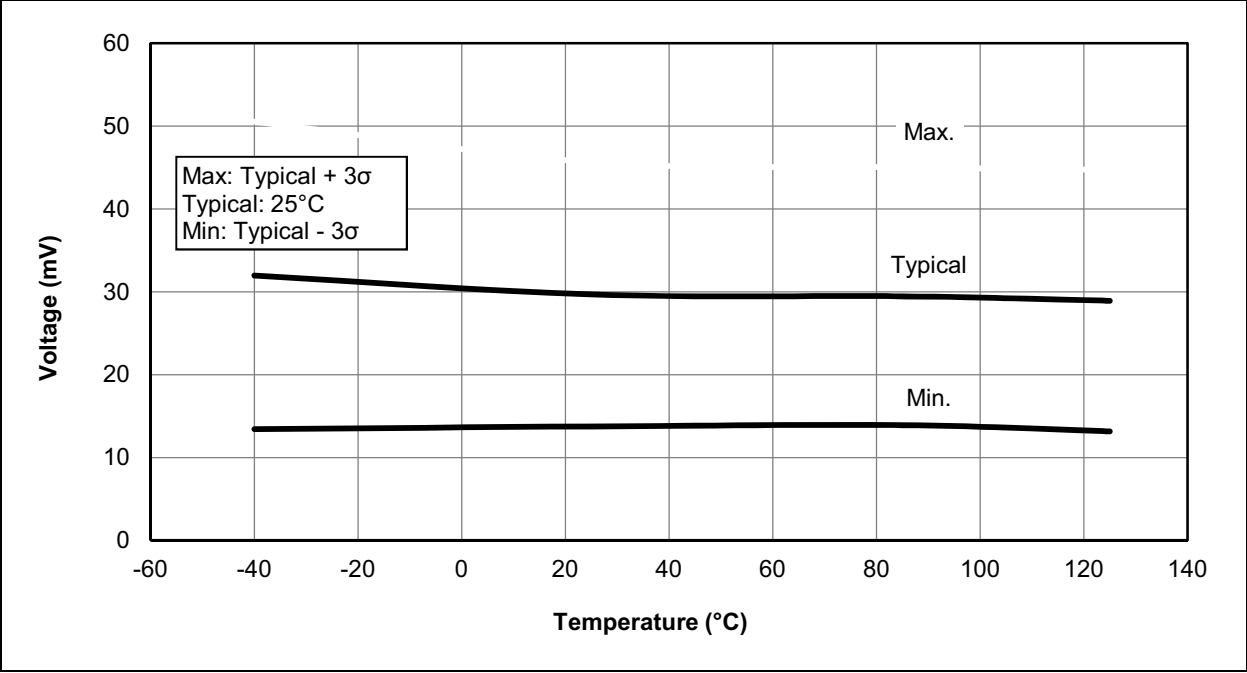
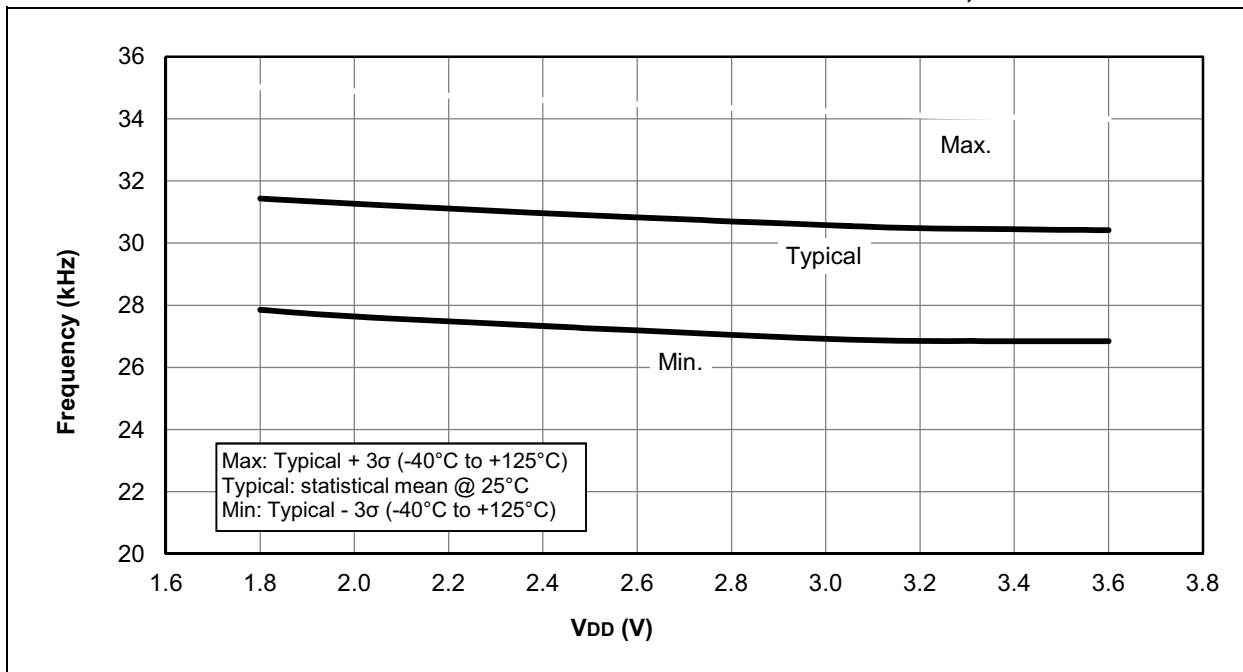


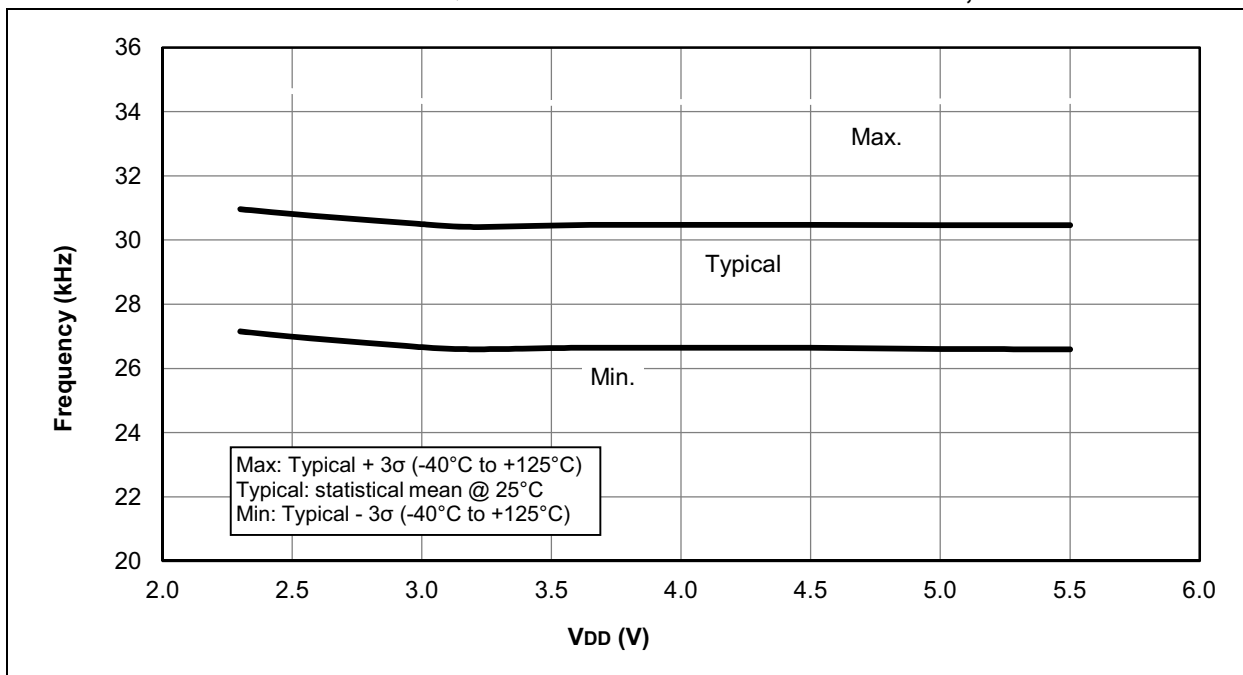
FIGURE 28-44: BROWN-OUT RESET HYSTERESIS, BORV = 1, PIC12LF1501 ONLY



**FIGURE 28-58: LFINTOSC FREQUENCY OVER  $V_{DD}$  AND TEMPERATURE, PIC12LF1501 ONLY**



**FIGURE 28-59: LFINTOSC FREQUENCY OVER  $V_{DD}$  AND TEMPERATURE, PIC12F1501 ONLY**



## 29.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

## 29.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 29.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

## 29.9 PICKit 3 In-Circuit Debugger/Programmer

The MPLAB PICKit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICKit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

## 29.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

**Trademarks**

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Klear, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC<sup>32</sup> logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KlearNet, KlearNet logo, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2011-2015, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-63277-915-1

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949 ==**

*Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*