



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	16MHz
Connectivity	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 12x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf720t-i-ss

2.2.2.3 PCON Register

The Power Control (PCON) register contains flag bits (refer to Table 3-4) to differentiate between a:

- Power-on Reset ($\overline{\text{POR}}$)
- Brown-out Reset ($\overline{\text{BOR}}$)
- Watchdog Timer Reset (WDT)
- External MCLR Reset

The PCON register also controls the software enable of the BOR.

The PCON register bits are shown in Register 2-3.

REGISTER 2-3: PCON: POWER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-q	R/W-q
—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7						bit 0	

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

q = Value depends on condition

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **$\overline{\text{POR}}$:** Power-on Reset Status bit

1 = No Power-on Reset occurred

0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{\text{BOR}}$:** Brown-out Reset Status bit

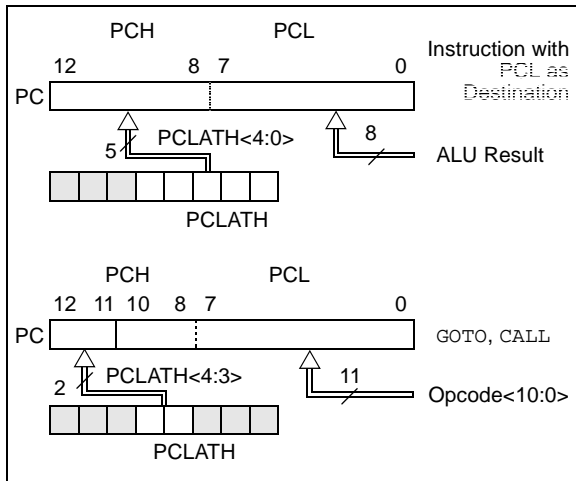
1 = No Brown-out Reset occurred

0 = A Brown-out Reset occurred (must be set in software after a Power-on Reset or Brown-out Reset occurs)

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 2-5 shows the two situations for the loading of the PC. The upper example in Figure 2-5 shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in Figure 2-5 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> → PCH).

FIGURE 2-5: LOADING OF PC IN DIFFERENT SITUATIONS



2.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When performing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to the Application Note AN556, "Implementing a Table Read" (DS00556).

2.3.2 STACK

All devices have an 8-level x 13-bit wide hardware stack (refer to Figures 2-1 and 2-2). 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 Program Memory Paging

All devices are capable of addressing a continuous 8K word block of program memory. The CALL and GOTO instructions provide only 11 bits of address to allow branching within any 2K program memory page. When doing a CALL or GOTO instruction, the upper two bits of the address are provided by PCLATH<4:3>. When doing a CALL or GOTO instruction, the user must ensure that the page Select bits are programmed so that the desired program memory page is addressed. If a return from a CALL instruction (or interrupt) is executed, the entire 13-bit PC is POPed off the stack. Therefore, manipulation of the PCLATH<4:3> bits is not required for the RETURN instructions (which POPs the address from the stack).

Note: The contents of the PCLATH register are unchanged after a RETURN or RETFIE instruction is executed. The user must rewrite the contents of the PCLATH register for any subsequent subroutine calls or GOTO instructions.

Example 2-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that PCLATH is saved and restored by the Interrupt Service Routine (if interrupts are used).

EXAMPLE 2-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

```

ORG 500h
PAGESEL SUB_P1 ;Select page 1
                ;(800h-FFFh)
CALL SUB1_P1 ;Call subroutine in
:            ;page 1 (800h-FFFh)
:
ORG 900h ;page 1 (800h-FFFh)
SUB1_P1
:            ;called subroutine
:            ;page 1 (800h-FFFh)
:
RETURN ;return to
        ;Call subroutine
        ;in page 0
        ;(000h-7FFh)
    
```

4.5.1 INTCON REGISTER

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, PORTB change and external RA2/INT pin interrupts.

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 4-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	TMR0IE	INTE	RABIE ⁽¹⁾	TMR0IF ⁽²⁾	INTF	RABIF
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts
- bit 6 **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
- bit 5 **TMR0IE:** Timer0 Overflow Interrupt Enable bit
1 = Enables the Timer0 interrupt
0 = Disables the Timer0 interrupt
- bit 4 **INTE:** INT External Interrupt Enable bit
1 = Enables the INT external interrupt
0 = Disables the INT external interrupt
- bit 3 **RABIE:** PORTA or PORTB Change Interrupt Enable bit⁽¹⁾
1 = Enables the PORTA or PORTB change interrupt
0 = Disables the PORTA or PORTB change interrupt
- bit 2 **TMR0IF:** Timer0 Overflow Interrupt Flag bit⁽²⁾
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow
- bit 1 **INTF:** INT External Interrupt Flag bit
1 = The INT external interrupt occurred (must be cleared in software)
0 = The INT external interrupt did not occur
- bit 0 **RABIF:** PORTA or PORTB Change Interrupt Flag bit
1 = When at least one of the PORTA or PORTB general purpose I/O pins changed state (must be cleared in software)
0 = None of the PORTA or PORTB general purpose I/O pins have changed state

Note 1: The appropriate bits in the IOCB register must also be set.

2: TMR0IF bit is set when Timer0 rolls over. Timer0 is unchanged on Reset and should be initialized before clearing TMR0IF bit.

PIC16(L)F720/721

6.0 I/O PORTS

There are as many as 18 general purpose I/O pins available. Depending on which peripherals are enabled, some or all of the pins may not be available as general purpose I/O. In general, when a peripheral is enabled, the associated pin may not be used as a general purpose I/O pin.

6.1 PORTA and TRISA Registers

PORTA is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISA (Register 6-2). Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., disable the output driver). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., enables output driver and puts the contents of the output latch on the selected pin). Example 6-1 shows how to initialize PORTA.

Reading the PORTA register (Register 6-1) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch.

The TRISA register (Register 6-2) controls the PORTA pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISA register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

Note: The ANSELA register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

EXAMPLE 6-1: INITIALIZING PORTA

```
BANKSEL PORTA      ;
CLRF  PORTA         ;Init PORTA
BANKSEL ANSELA      ;
CLRF  ANSELA        ;digital I/O
BANKSEL TRISA       ;
MOVLW 0Ch           ;Set RA<3:2> as inputs
MOVWF TRISA         ;and set RA<5:4,1:0>
                   ;as outputs
```

6.1.1 WEAK PULL-UPS

Each of the PORTA pins has an individually configurable internal weak pull-up. Control bits WPUA<5:0> enable or disable each pull-up (see Register 6-5). Each weak pull-up is automatically turned off when the port pin is configured as an output. All pull-ups are disabled on a Power-on Reset by the RABPU bit of the OPTION_REG register.

6.1.2 INTERRUPT-ON-CHANGE

All of the PORTA pins are individually configurable as an interrupt-on-change pin. Control bits IOCA<5:0> enable or disable the interrupt function for each pin (see Register 6-6). The interrupt-on-change feature is disabled on a Power-on Reset.

For enable interrupt-on-change pins, the present value is compared with the old value latched on the last read of PORTA to determine which bits have changed or mismatched the old value. The 'mismatch' outputs of the last read are OR'd together to set the PORTA Change Interrupt Flag bit (RABIF) in the INTCON register. This interrupt can wake the device from Sleep. The user, in the Interrupt Service Routine, clears the interrupt by:

1. Any read or write of PORTA. This will end the mismatch condition.
2. Clear the flag bit RABIF.

A mismatch condition will continue to set flag bit RABIF. Reading or writing PORTA will end the mismatch condition and allow flag bit RABIF to be cleared. The latch holding the last read value is not affected by a MCLR or Brown-out Reset. After these Resets, the RABIF flag will continue to be set if a mismatch is present.

Note: When a pin change occurs at the same time as a read operation on PORTA, the RABIF flag will always be set. If multiple PORTA pins are configured for the interrupt-on-change, the user may not be able to identify which pin changed state.

7.2 Clock Source Modes

Clock source modes can be classified as external or internal.

- Internal clock source (INTOSC) is contained within the oscillator module and derived from a 500 kHz high-precision oscillator. The oscillator module has eight selectable output frequencies, with a maximum internal frequency of 16 MHz.
- The External Clock mode (EC) relies on an external signal for the clock source.

The system clock can be selected between external or internal clock sources via the FOSC bits of the Configuration Word 1.

7.3 Internal Clock Modes

The oscillator module has eight output frequencies derived from a 500 kHz high-precision oscillator. The IRCF bits of the OSCCON register select the postscaler applied to the clock source dividing the frequency by 1, 2, 4 or 8. Setting the PLLEN bit of the Configuration Word 1 locks the internal clock source to 16 MHz before the postscaler is selected by the IRCF bits. The PLLEN bit must be set or cleared at the time of programming; therefore, only the upper or low four clock source frequencies are selectable in software.

The internal oscillator block has one internal oscillator and a dedicated Phase-Locked Loop that are used to generate two internal system clock sources: the 16 MHz High-Frequency Internal Oscillator (HFINTOSC) and the 500 kHz (MFINTOSC). Both can be user-adjusted via software using the OSCTUNE register (Register 7-2).

7.3.1 INTOSC AND INTOSCIO MODES

The INTOSC and INTOSCIO modes configure the internal oscillators as system clock source when the device is programmed using the oscillator selection or the FOSC<1:0> bits in the CONFIG1 register. See **Section 8.0 “Device Configuration”** for more information.

In INTOSC mode, CLKIN is available for general purpose I/O. CLKOUT outputs the selected internal oscillator frequency divided by 4. The CLKOUT signal may be used to provide a clock for external circuitry, synchronization, Calibration, test or other application requirements.

In INTOSCIO mode, CLKIN and CLKOUT are available for general purpose I/O.

7.3.2 FREQUENCY SELECT BITS (IRCF)

The output of the 500 kHz MFINTOSC and 16 MHz HFINTOSC, with Phase-Locked Loop enabled, connect to a postscaler and multiplexer (see Figure 7-1). The Internal Oscillator Frequency Select bits (IRCF) of the OSCCON register select the frequency output of the internal oscillator. Depending upon the PLLEN bit, one of four frequencies of two frequency sets can be selected via software:

If PLLEN = 1, HFINTOSC frequency selection is as follows:

- 16 MHz
- 8 MHz (default after Reset)
- 4 MHz
- 2 MHz

If PLLEN = 0, MFINTOSC frequency selection is as follows:

- 500 kHz
- 250 kHz (default after Reset)
- 125 kHz
- 62.5 kHz

Note: Following any Reset, the IRCF<1:0> bits of the OSCCON register are set to '10' and the frequency selection is set to 8 MHz or 250 kHz. The user can modify the IRCF bits to select a different frequency.

There is no start-up delay before a new frequency selected in the IRCF bits takes effect. This is because the old and new frequencies are derived from INTOSC via the postscaler and multiplexer.

Start-up delay specifications are located in the Table 23-2 in **Section 23.0 “Electrical Specifications”**.

7.3.3 INTERNAL OSCILLATOR STATUS BITS

The internal oscillator (500 kHz) is a factory-calibrated internal clock source. The frequency can be altered via software using the OSCTUNE register (Register 7-2).

The Internal Oscillator Status Locked bit (ICSL) of the OSCCON register indicates when the internal oscillator is running within 2% of its final value.

The Internal Oscillator Status Stable bit (ICSS) of the OSCCON register indicates when the internal oscillator is running within 0.5% of its final value.

PIC16(L)F720/721

7.6 External Clock Modes

7.6.1 EC MODE

The External Clock (EC) mode allows an externally generated logic level as the system clock source. When operating in this mode, an external clock source is connected to the CLKIN input and the CLKOUT is available for general purpose I/O. Figure 7-2 shows the pin connections for EC mode.

FIGURE 7-2: EXTERNAL CLOCK (EC) MODE OPERATION

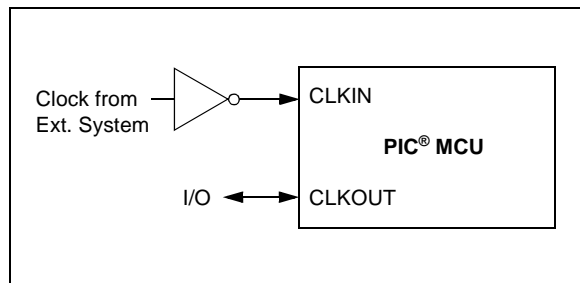


TABLE 7-1: SUMMARY OF REGISTERS ASSOCIATED WITH CLOCK SOURCES

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
OSCCON	—	—	IRCF1	IRCF0	ICSL	ICSS	—	—	64
OSCTUNE	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	65

Legend: x = unknown, u = unchanged, — = unimplemented locations read as '0'. Shaded cells are not used by clock sources.

TABLE 7-2: SUMMARY OF CONFIGURATION WORD WITH CLOCK SOURCES

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
CONFIG1	13:8	—	—	—	PLLEN	—	—	BOREN1	BOREN0	68
	7:0	—	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTEN	—	FOSC1	FOSC0	
CONFIG2	13:8	—	—	—	—	—	—	—	—	69
	7:0	—	—	—	—	—	—	WRT1	WRT0	

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by clock sources.

12.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A single software programmable prescaler is available for use with either Timer0 or the Watchdog Timer (WDT), but not both simultaneously. The prescaler assignment is controlled by the PSA bit of the OPTION_REG register. To assign the prescaler to Timer0, the PSA bit must be cleared to a '0'.

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 assigned to the WDT module.

The prescaler is not readable or writable. When assigned to the Timer0 module, all instructions writing to the TMR0 register will clear the prescaler.

Note: When the prescaler is assigned to WDT, a CLRWDI instruction will clear the prescaler along with the WDT.

12.1.4 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.
--

12.1.5 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 23.0 “Electrical Specifications”**.

13.5.2.1 T1G Pin Gate Operation

The T1G pin is one source for Timer1 gate control. It can be used to supply an external source to the Timer1 gate circuitry.

13.5.2.2 Timer0 Overflow Gate Operation

When Timer0 increments from FFh to 00h, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 gate circuitry.

13.5.2.3 Timer2 Match Gate Operation

The TMR2 register will increment until it matches the value in the PR2 register. On the very next increment cycle, TMR2 will be reset to 00h. When this Reset occurs, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 gate circuitry.

13.5.2.4 Watchdog Overflow Gate Operation

The Watchdog Timer oscillator, prescaler and counter will be automatically turned on when TMR1GE = 1 and T1GSS selects the WDT as a gate source for Timer1 (T1GSS = 11).

TMR1ON does not factor into the oscillator, prescaler and counter enable (see Table 13-5).

The PSA and PS bits of the OPTION_REG register still control what time-out interval is selected. Changing the prescaler during operation may result in a spurious capture.

Enabling the Watchdog Timer oscillator does not automatically enable a Watchdog Reset or Wake-up from Sleep upon counter overflow.

Note: When using the WDT as a gate source for Timer1, operations that clear the Watchdog Timer (CLRWDT, SLEEP instructions) will affect the time interval being measured. This includes waking from Sleep. All other interrupts that might wake the device from Sleep should be disabled to prevent them from disturbing the measurement period.

As the gate signal coming from the WDT counter will generate different pulse widths depending on if the WDT is enabled, when the CLRWDT instruction is executed, and so on, Toggle mode must be used. A specific sequence is required to put the device into the correct state to capture the next WDT counter interval.

TABLE 13-5: WDT/TIMER1 GATE INTERACTION

WDTEN	TMR1GE = 1 and T1GSS = 11	WDT Oscillator Enable	WDT Reset	Wake-up	WDT Available for T1G Source
1	N	Y	Y	Y	N
1	Y	Y	Y	Y	Y
0	Y	Y	N	N	Y
0	N	N	N	N	N

PIC16(L)F720/721

14.0 TIMER2 MODULE

The Timer2 module is an 8-bit timer with the following features:

- 8-bit timer register (TMR2)
- 8-bit period register (PR2)
- Interrupt on TMR2 match with PR2
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)

See Figure 14-1 for a block diagram of Timer2.

14.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock ($F_{osc}/4$). The clock is fed into the Timer2 prescaler, which has prescale options of 1:1, 1:4 or 1:16. The output of the prescaler is then used to increment the TMR2 register.

The values of TMR2 and PR2 are constantly compared to determine when they match. TMR2 will increment from 00h until it matches the value in PR2. When a match occurs, two things happen:

- TMR2 is reset to 00h on the next increment cycle.
- The Timer2 postscaler is incremented.

The match output of the Timer2/PR2 comparator is then fed into the Timer2 postscaler. The postscaler has postscale options of 1:1 to 1:16 inclusive. The output of the Timer2 postscaler is used to set the TMR2IF interrupt flag bit in the PIR1 register.

The TMR2 and PR2 registers are both fully readable and writable. On any Reset, the TMR2 register is set to 00h and the PR2 register is set to FFh.

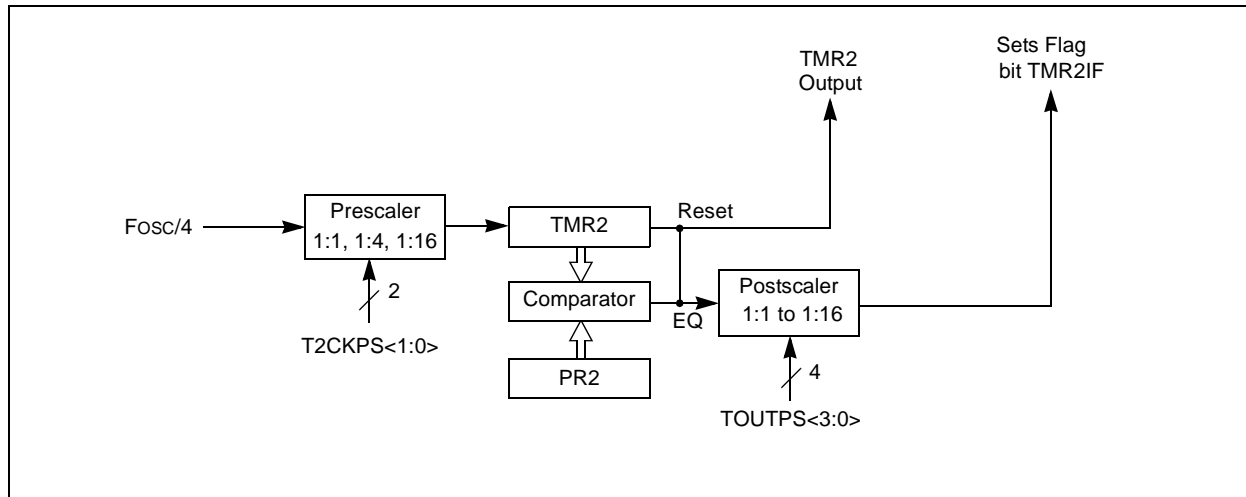
Timer2 is turned on by setting the TMR2ON bit in the T2CON register to '1'. Timer2 is turned off by clearing the TMR2ON bit to '0'.

The Timer2 prescaler is controlled by the T2CKPS bits in the T2CON register. The Timer2 postscaler is controlled by the TOUTPS bits in the T2CON register. The prescaler and postscaler counters are cleared when:

- A write to TMR2 occurs.
- A write to T2CON occurs.
- Any device Reset occurs (Power-on Reset, \overline{MCLR} Reset, Watchdog Timer Reset, or Brown-out Reset).

Note: TMR2 is not cleared when T2CON is written.

FIGURE 14-1: TIMER2 BLOCK DIAGRAM



14.2 Timer2 Control Register

REGISTER 14-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **TOUTPS<3:0>:** Timer2 Output Postscaler Select bits

0000 = 1:1 Postscaler
 0001 = 1:2 Postscaler
 0010 = 1:3 Postscaler
 0011 = 1:4 Postscaler
 0100 = 1:5 Postscaler
 0101 = 1:6 Postscaler
 0110 = 1:7 Postscaler
 0111 = 1:8 Postscaler
 1000 = 1:9 Postscaler
 1001 = 1:10 Postscaler
 1010 = 1:11 Postscaler
 1011 = 1:12 Postscaler
 1100 = 1:13 Postscaler
 1101 = 1:14 Postscaler
 1110 = 1:15 Postscaler
 1111 = 1:16 Postscaler

bit 2 **TMR2ON:** Timer2 On bit

1 = Timer2 is On
 0 = Timer2 is Off

bit 1-0 **T2CKPS<1:0>:** Timer2 Clock Prescale Select bits

00 = Prescaler is 1
 01 = Prescaler is 4
 1x = Prescaler is 16

TABLE 14-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2

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	RABIE	TMR0IF	INTF	RABIF	37
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	38
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	39
PR2	Timer2 module Period Register								98
TMR2	Timer2 module Register								98
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	99

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used for Timer2 module.

PIC16(L)F720/721

17.2.6 TRANSMISSION

When the $\overline{R/W}$ bit of the received address byte is set and an address match occurs, the $\overline{R/W}$ bit of the SSPSTAT register is set and the slave will respond to the master by reading out data. After the address match, an \overline{ACK} pulse is generated by the slave hardware and the SCL pin is held low (clock is automatically stretched) until the slave is ready to respond. See **Section 17.2.7 “Clock Stretching”**. The data the slave will transmit must be loaded into the SSPBUF register, which sets the BF bit. The SCL line is released by setting the CKP bit of the SSPCON register.

An SSP interrupt is generated for each transferred data byte. The SSPIF flag bit of the PIR1 register initiates an SSP interrupt, and must be cleared by software before the next byte is transmitted. The BF bit of the SSPSTAT register is cleared on the falling edge of the eighth received clock pulse. The SSPIF flag bit is set on the falling edge of the ninth clock pulse.

Following the eighth falling clock edge, control of the SDA line is released back to the master so that the master can acknowledge or not acknowledge the response. If the master sends a not acknowledge, the slave's transmission is complete and the slave must monitor for the next Start condition. If the master acknowledges, control of the bus is returned to the slave to transmit another byte of data. Just as with the previous byte, the clock is stretched by the slave, data must be loaded into the SSPBUF and CKP must be set to release the clock line (SCL).

FIGURE 17-12: I²C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)

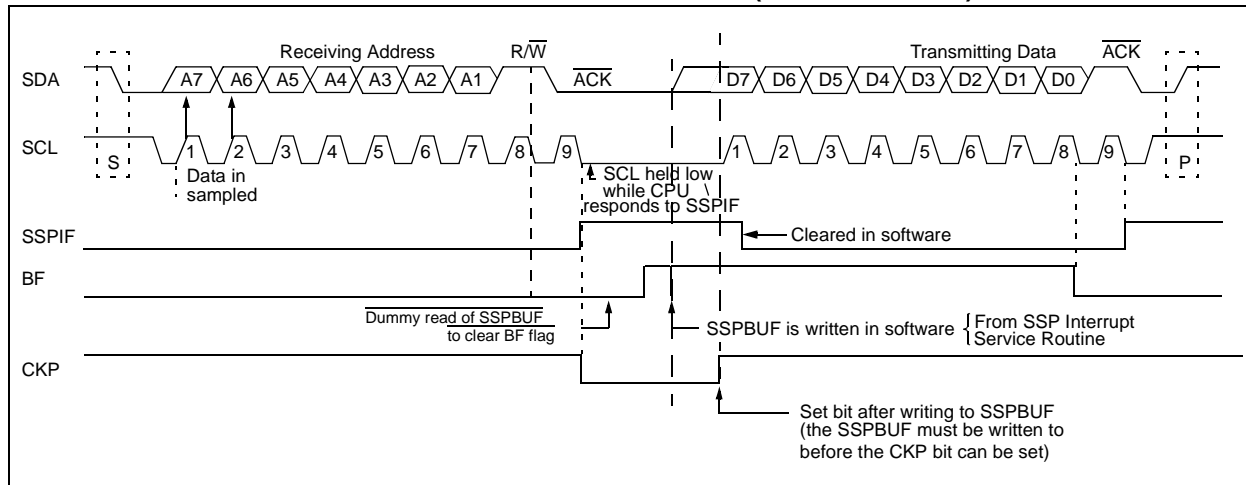
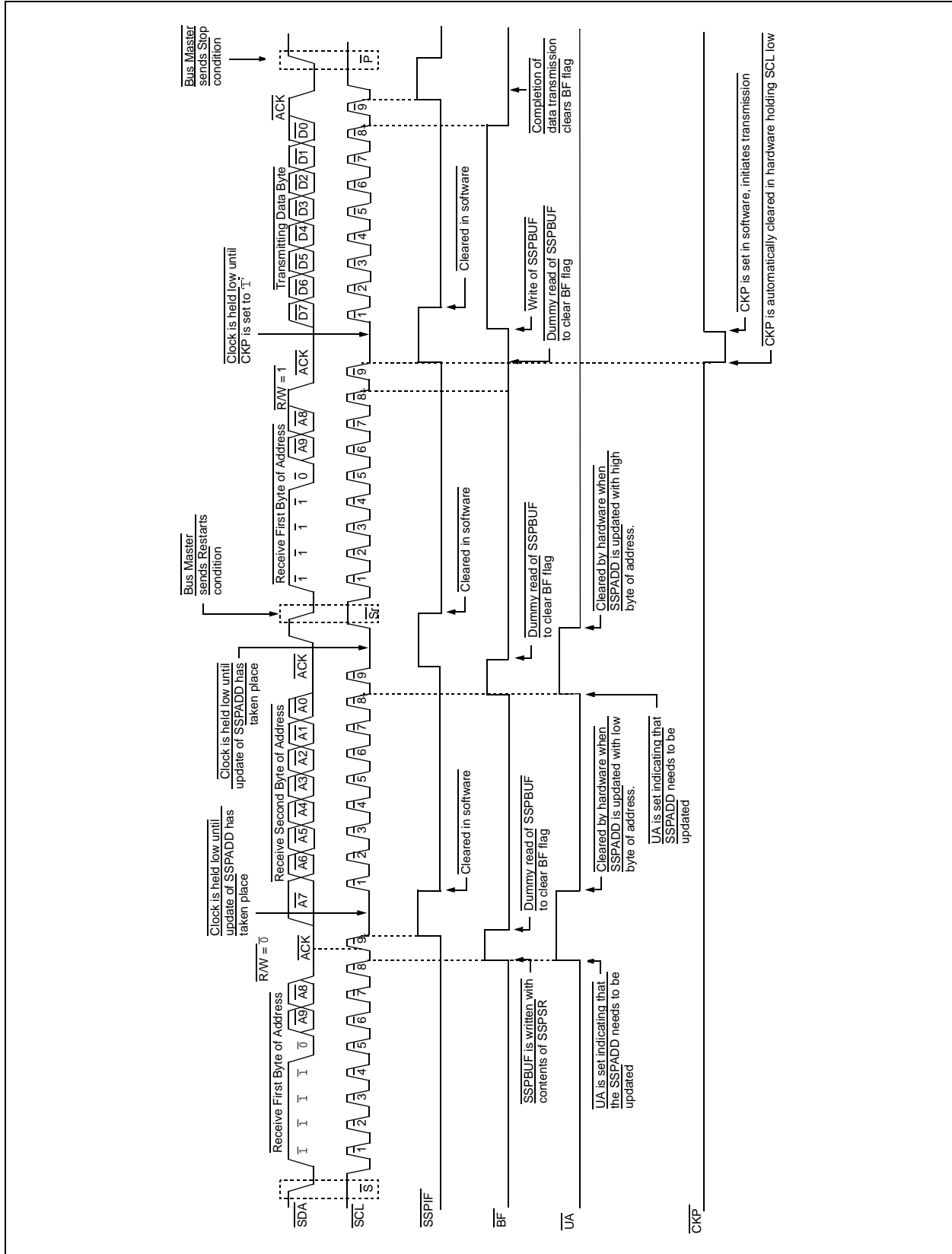


FIGURE 17-13: I²C SLAVE MODE TIMING (TRANSMISSION 10-BIT ADDRESS)



PIC16(L)F720/721

REGISTER 18-2: PMDATH: PROGRAM MEMORY DATA HIGH REGISTER

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	PMD13	PMD12	PMD11	PMD10	PMD9	PMD8
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **PMD<13:8>:** The value of the program memory word pointed to by PMADRH and PMADRL after a program memory read command.

REGISTER 18-3: PMDATL: PROGRAM MEMORY DATA LOW REGISTER

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
PMD7	PMD6	PMD5	PMD4	PMD3	PMD2	PMD1	PMD0
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-0 **PMD<7:0>:** The value of the program memory word pointed to by PMADRH and PMADRL after a program memory read command.

REGISTER 18-4: PMADRH: PROGRAM MEMORY ADDRESS HIGH REGISTER

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	PMA12	PMA11	PMA10	PMA9	PMA8
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **PMA<12:8>:** Program Memory Read Address bits

PIC16(L)F720/721

MOVF	Move f
Syntax:	[<i>label</i>] MOVF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) \rightarrow (dest)
Status Affected:	Z
Description:	The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.
Words:	1
Cycles:	1
Example:	<pre>MOVF FSR, 0</pre> <p>After Instruction</p> <p>W = value in FSR register</p> <p>Z = 1</p>

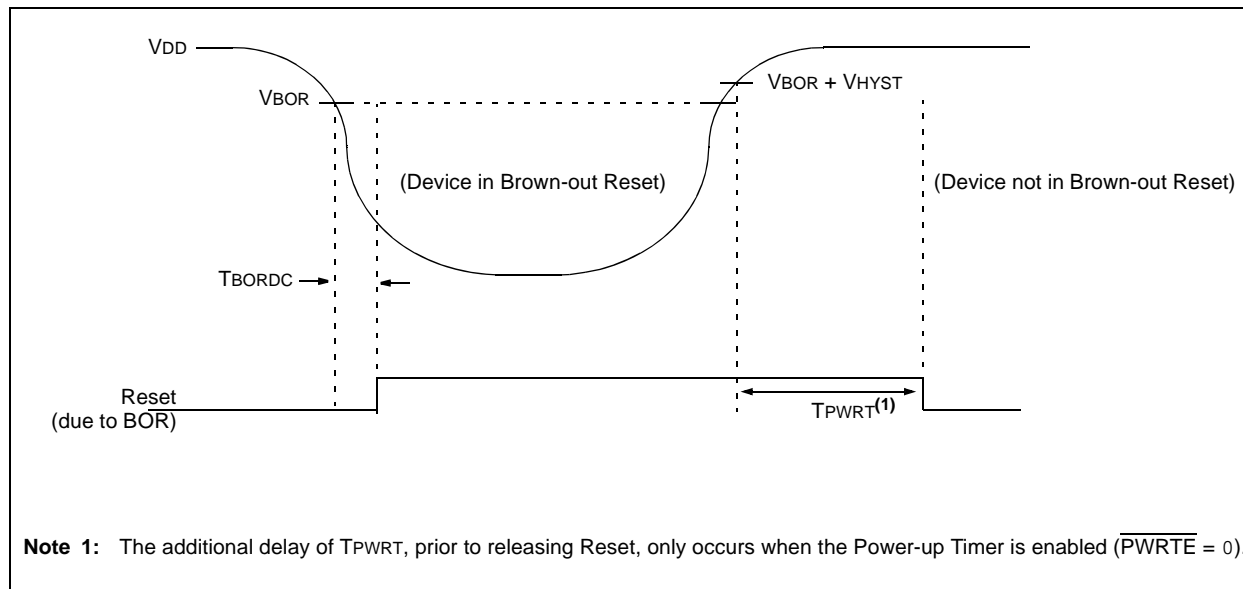
MOVLW	Move literal to W
Syntax:	[<i>label</i>] MOVLW k
Operands:	$0 \leq k \leq 255$
Operation:	$k \rightarrow (W)$
Status Affected:	None
Description:	The 8-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.
Words:	1
Cycles:	1
Example:	<pre>MOVLW 0x5A</pre> <p>After Instruction</p> <p>W = 0x5A</p>

MOVWF	Move W to f
Syntax:	[<i>label</i>] 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:	<pre>MOVW OPTION F</pre> <p>Before Instruction</p> <p>OPTION = 0xFF W = 0x4F</p> <p>After Instruction</p> <p>OPTION = 0x4F W = 0x4F</p>

NOP	No Operation
Syntax:	[<i>label</i>] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.
Words:	1
Cycles:	1
Example:	<pre>NOP</pre>

PIC16(L)F720/721

FIGURE 23-8: BROWN-OUT RESET TIMING AND CHARACTERISTICS



PIC16(L)F720/721

TABLE 23-8: PIC16F720/721 A/D CONVERSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
Param. No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
AD130*	TAD	A/D Clock Period	1.0	—	9.0	μS	$V_{DD} > 2.0\text{V}^{(2)}$
			4.0	—	16.0	μS	$V_{DD} \leq 2.0\text{V}^{(2)}$
		A/D Internal RC Oscillator Period	1.0	2.0	6.0	μS	(ADRC mode)
AD131	TCNV	Conversion Time (not including Acquisition Time) ⁽¹⁾	—	10.5	—	TAD	Set GO/DONE bit to new data in A/D Result register
AD132*	TACQ	Acquisition Time		2	—	μS	$V_{DD} = 3.0\text{V}$, EC or INTOSC Clock mode ⁽³⁾

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The ADRES register may be read on the following TCY cycle.

Note 2: Setting of 16.0 μS TAD not recommended for temperature $> 85^{\circ}\text{C}$.

Note 3: If ADRC mode is selected for use with $V_{DD} \leq 2.0\text{V}$, longer acquisition times will be required (see **Section 9.3 "A/D Acquisition Requirements"**)

FIGURE 23-11: PIC16F720/721 A/D CONVERSION TIMING (NORMAL MODE)

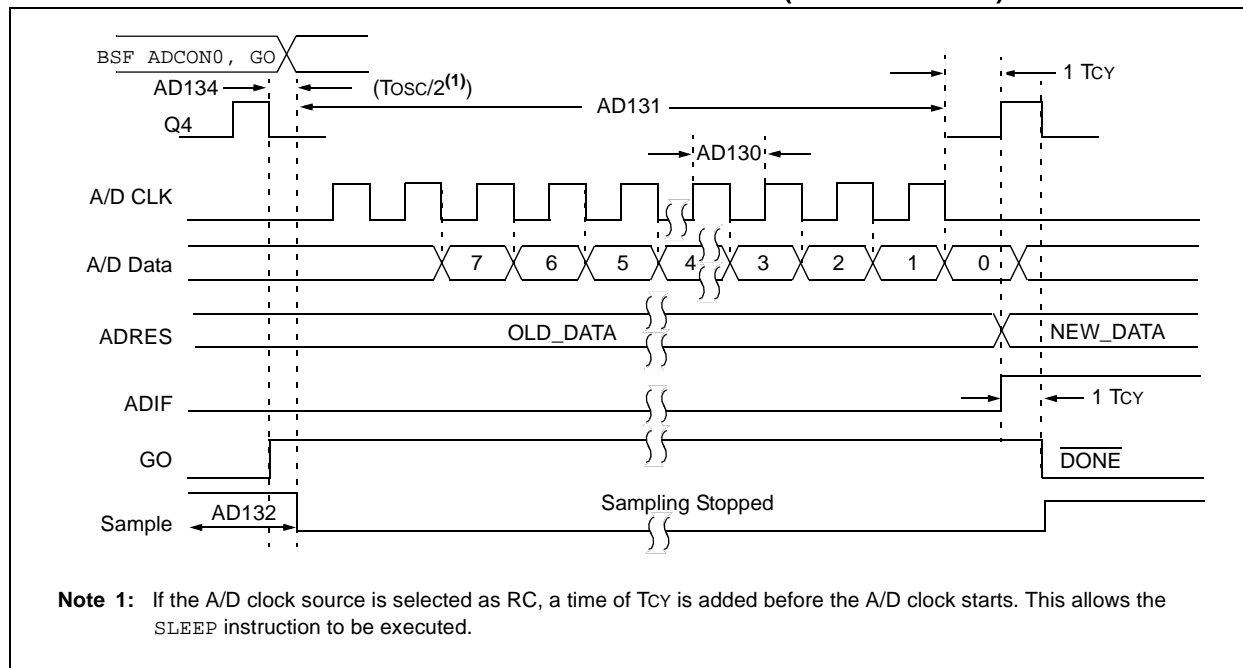


FIGURE 24-30: Vol vs. IoL OVER TEMPERATURE, VDD = 1.8V

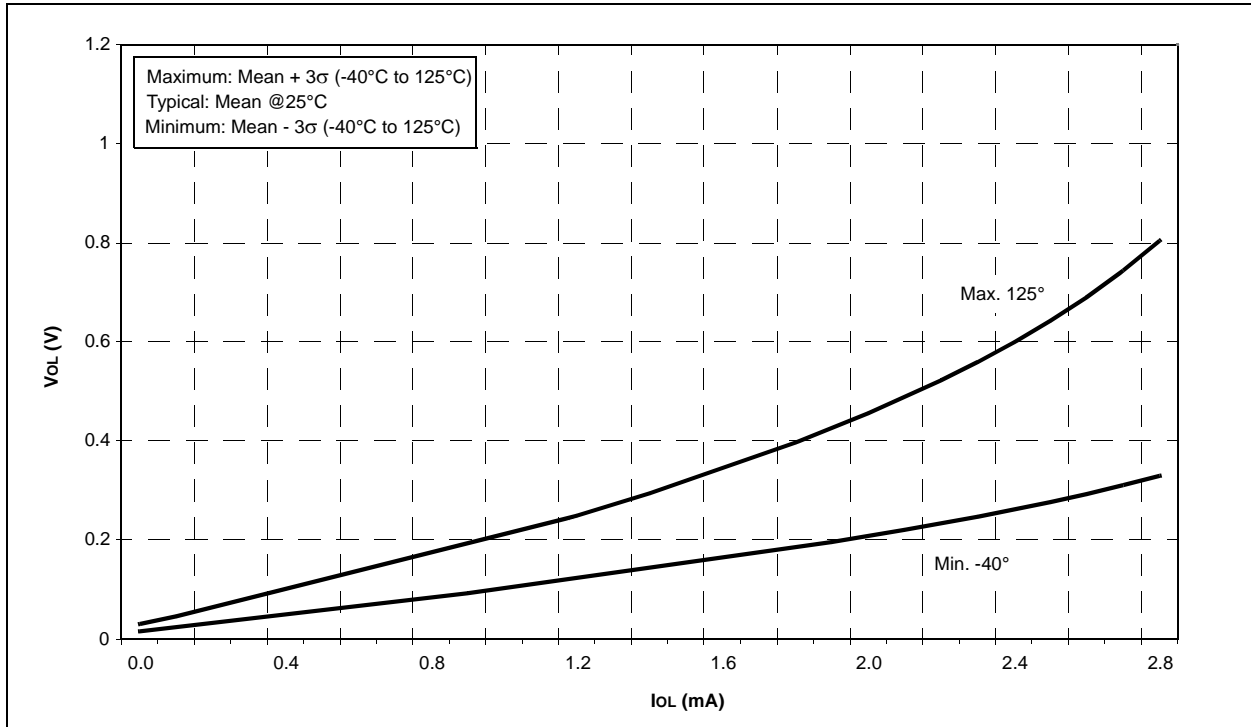


FIGURE 24-31: PIC16F720/721 PWRT PERIOD

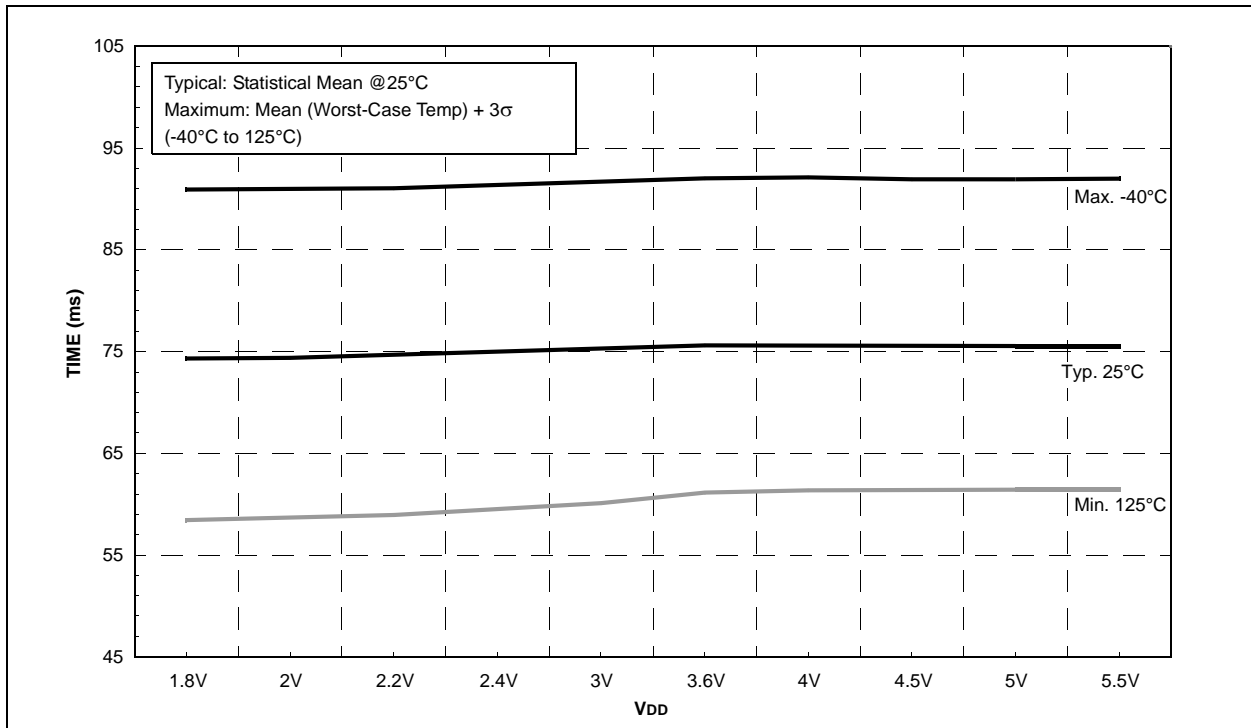


FIGURE 24-34: PIC16F720/721 A/D INTERNAL RC OSCILLATOR PERIOD

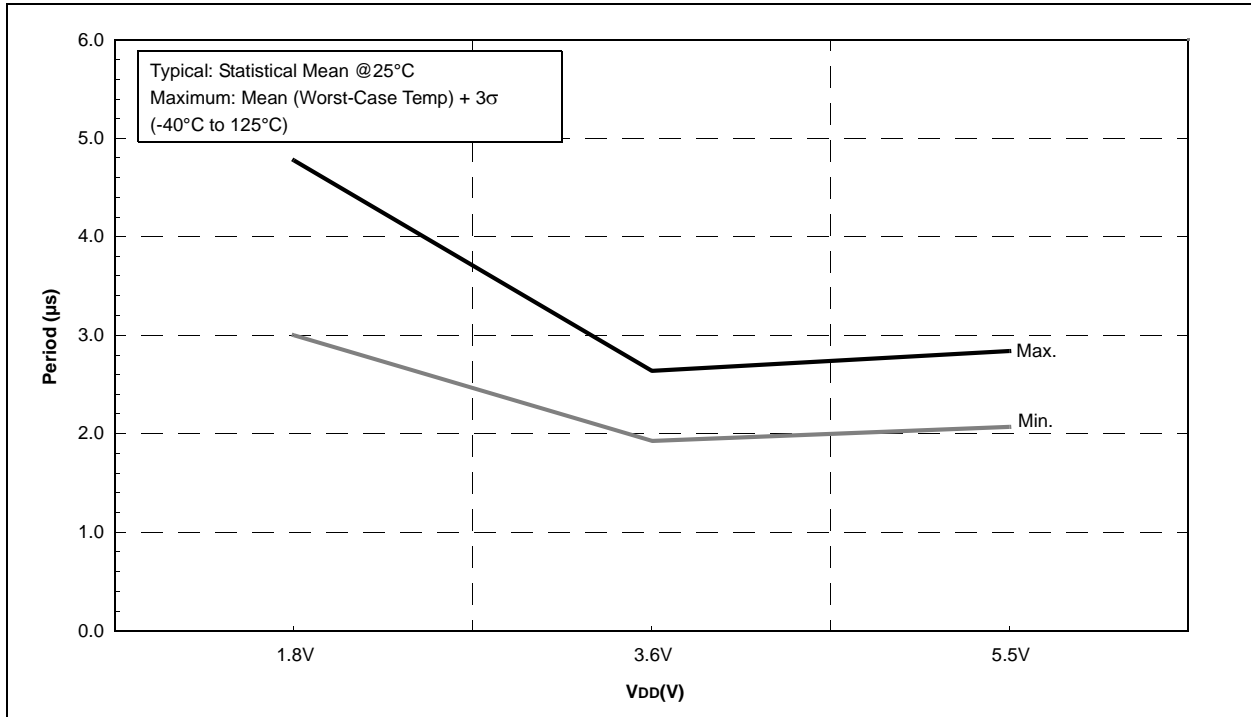
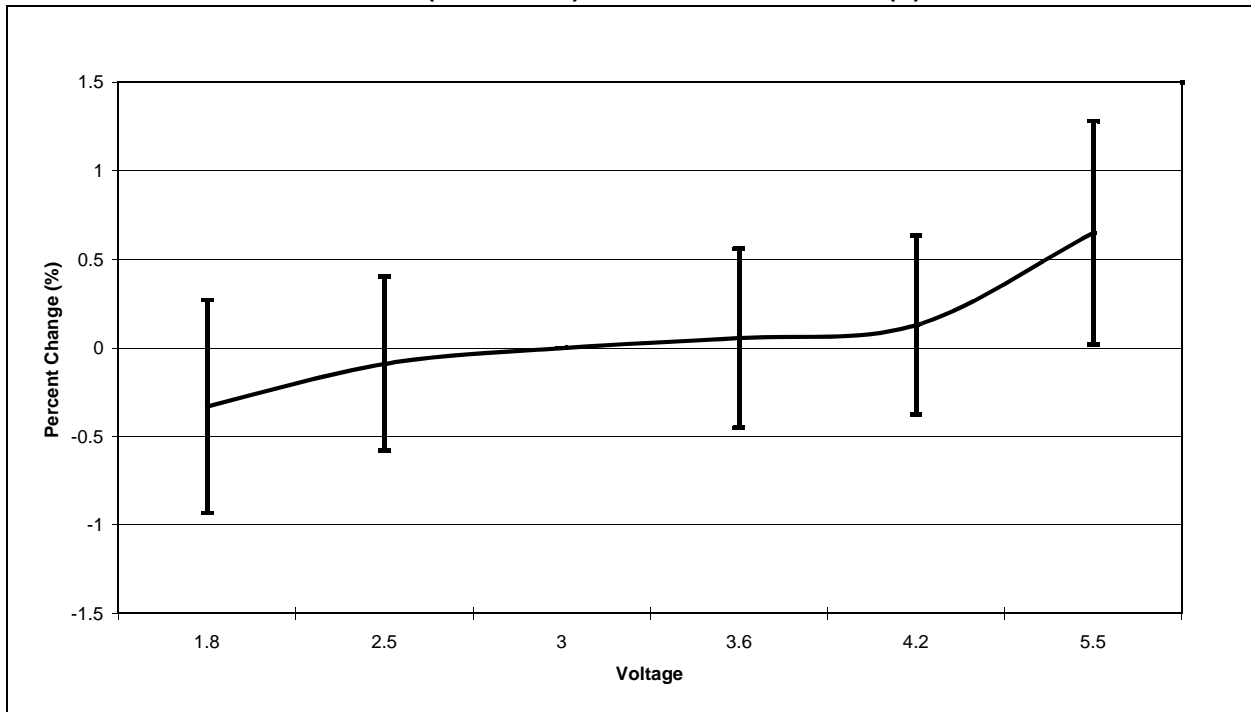


FIGURE 24-35: TYPICAL FVR (X1 AND X2) VS. SUPPLY VOLTAGE (V) NORMALIZED AT 3.0V

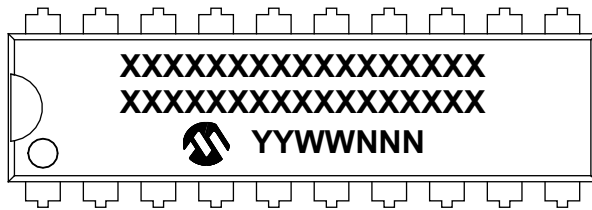


PIC16(L)F720/721

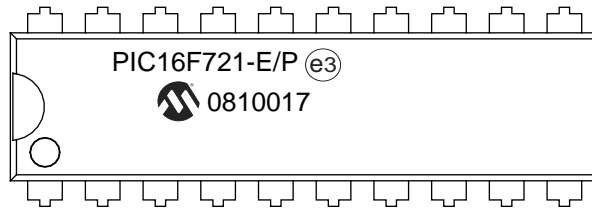
25.0 PACKAGING INFORMATION

25.1 Package Marking Information

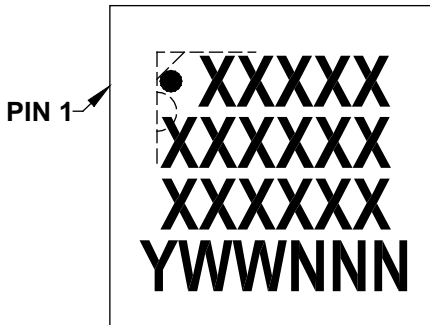
20-Lead PDIP (300 mil)



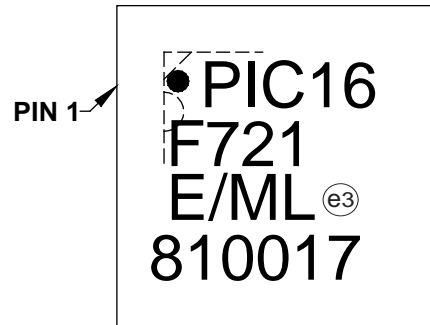
Example



20-Lead QFN (4x4x0.9 mm)



Example



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

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

- * Standard PICmicro® device marking consists of Microchip part number, year code, week code and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

PIC16(L)F720/721

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

<u>PART NO.</u>	<u>IXI⁽¹⁾</u>	<u>X</u>	<u>XX</u>	<u>XXX</u>	Examples: a) PIC16F720-E/P 301 = Extended Temp., PDIP package, QTP pattern #301 b) PIC16F721T-I/SO = Tape and Reel, Industrial Temp., SOIC package
Device	Tape and Reel Option	Temperature Range	Package	Pattern	
Device: PIC16F720, PIC16LF720, PIC16F721, PIC16LF721					
Temperature Range: I = -40°C to +85°C E = -40°C to +125°C					
Package: ML = Micro Lead Frame (QFN) P = Plastic DIP SO = SOIC SS = SSOP					
Pattern: 3-Digit Pattern Code for QTP (blank otherwise)					

Note 1: T= Available in tape and reel for all industrial devices except PDIP
2: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.