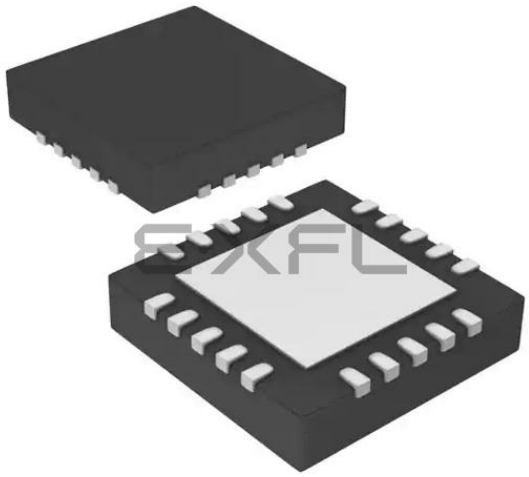


Welcome to E-XFL.COM

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	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 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-VFQFN Exposed Pad
Supplier Device Package	20-QFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf721-i-ml

2.0 MEMORY ORGANIZATION

2.1 Program Memory Organization

The PIC16(L)F720/721 has a 13-bit program counter capable of addressing a 8K x 14 program memory space. Table 2-1 shows the memory sizes implemented. 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.

TABLE 2-1: DEVICE SIZE AND ADDRESSES

Device	Program Memory Size (Words)	Last Program Memory Address	High-Endurance Flash Memory Address Range (1)
PIC16F720 PIC16LF720	2048	07FFh	0780h-07FFh
PIC16F721 PIC16LF721	4096	0FFFh	0F80h-0FFFh

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

FIGURE 2-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16(L)F720

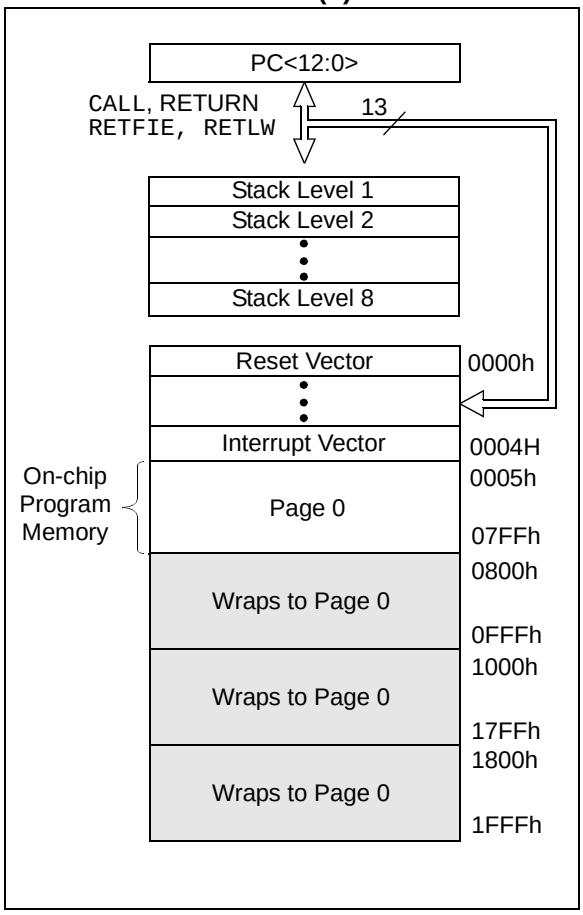
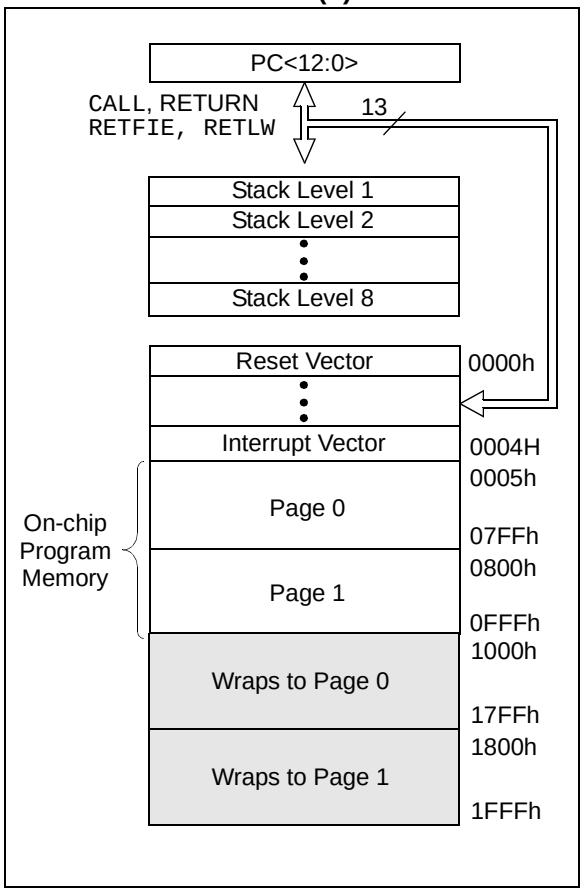


FIGURE 2-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16(L)F721



2.2.2.1 STATUS Register

The STATUS register, shown in Register 2-1, contains:

- the arithmetic status of the ALU
- the Reset status
- the bank select bits for data memory (SRAM)

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as '`000u u1uu`' (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits (Refer to **Section 21.0 "Instruction Set Summary"**).

Note 1: The C and DC bits operate as $\overline{Borrow}$ and $\overline{Digit\ Borrow}$ out bits, respectively, in subtraction.

REGISTER 2-1: STATUS: STATUS REGISTER

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC(1)	C(1)
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 **IRP:** Register Bank Select bit (used for indirect addressing)
`1` = Bank 2, 3 (100h-1FFh)
`0` = Bank 0, 1 (00h-FFh)

bit 6-5 **RP<1:0>:** Register Bank Select bits (used for direct addressing)
`00` = Bank 0 (00h-7Fh)
`01` = Bank 1 (80h-FFh)
`10` = Bank 2 (100h-17Fh)
`11` = Bank 3 (180h-1FFh)

bit 4 **$\overline{TO}$:** Time-out bit
`1` = After power-up, `CLRWDT` instruction or `SLEEP` instruction
`0` = A WDT time out occurred

bit 3 **$\overline{PD}$:** Power-down bit
`1` = After power-up or by the `CLRWDT` instruction
`0` = By execution of the `SLEEP` instruction

bit 2 **Z:** Zero bit
`1` = The result of an arithmetic or logic operation is zero
`0` = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/$\overline{Digit\ Borrow}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)(1)
`1` = A carry out from the 4th low-order bit of the result occurred
`0` = No carry out from the 4th low-order bit of the result

bit 0 **C:** Carry/$\overline{Borrow}$ bit(1) (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)(1)
`1` = A carry out from the Most Significant bit of the result occurred
`0` = No carry out from the Most Significant bit of the result occurred

Note 1: For $\overline{Borrow}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate instructions (`RRF`, `RLF`), this bit is loaded with either the high-order or low-order bit of the source register.

2.5 Indirect Addressing, INDF and FSR Registers

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

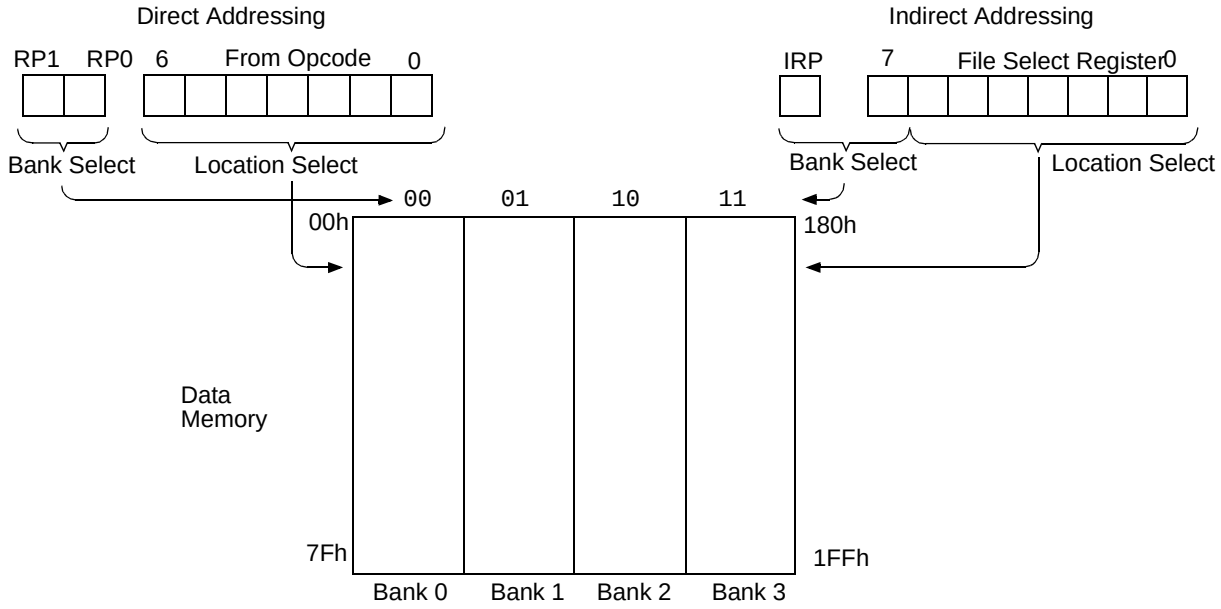
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit of the STATUS register, as shown in Figure 2-6.

A simple program to clear the RAM location 020h-02Fh using indirect addressing is shown in Example 2-2.

EXAMPLE 2-2: INDIRECT ADDRESSING

```
    MOVLW  020h  ;initialize pointer
    MOVWF  FSR   ;to RAM
    BANKISEL 020h
NEXT CLRF  INDF  ;clear INDF register
     INCF  FSR   ;inc pointer
     BTFSS FSR,4 ;all done?
     GOTO  NEXT  ;no clear next
CONTINUE         ;yes continue
```

FIGURE 2-6: DIRECT/INDIRECT ADDRESSING



Note: For memory map detail, refer to Figures 2-3 and 2-4.

3.0 RESETS

The PIC16(L)F720/721 differentiates between various kinds of Reset:

a) Power-on Reset (POR)
b) WDT Reset during normal operation
c) WDT Reset during Sleep
d) $\overline{\text{MCLR}}$ Reset during normal operation
e) $\overline{\text{MCLR}}$ Reset during Sleep
f) Brown-out Reset (BOR)

Some registers are not affected in any Reset condition; their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on:

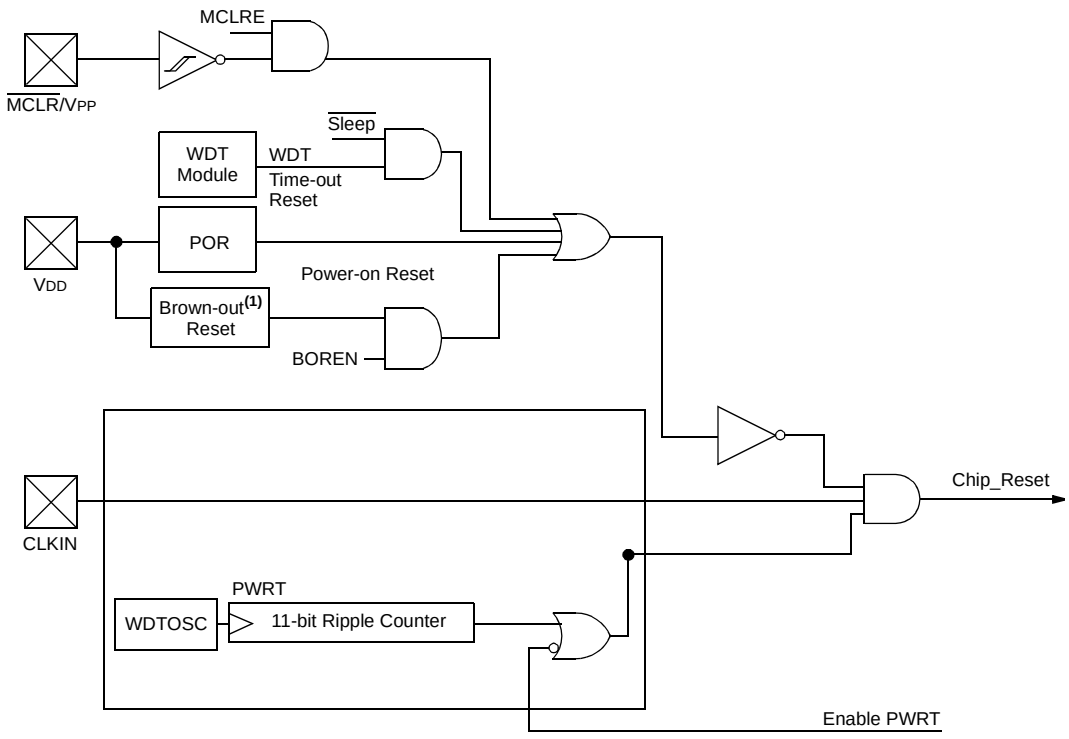
- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Reset
- Brown-out Reset (BOR)

Most registers are not affected by a WDT wake-up since this is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different Reset situations, as indicated in Table 3-5. These bits are used in software to determine the nature of the Reset.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 3-1.

The $\overline{\text{MCLR}}$ Reset path has a noise filter to detect and ignore small pulses. See **Section 23.0 "Electrical Specifications"** for pulse-width specifications.

FIGURE 3-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: Refer to the Configuration Word Register 1 (Register 8-1).

4.1 Operation

Interrupts are disabled upon any device Reset. They are enabled by setting the following bits:

- GIE bit of the INTCON register
- Interrupt Enable bit(s) for the specific interrupt event(s)
- PEIE bit of the INTCON register (if the Interrupt Enable bit of the interrupt event is contained in the PIE1 register)

The INTCON and PIR1 registers record individual interrupts via interrupt flag bits. Interrupt flag bits will be set, regardless of the status of the GIE, PEIE and individual Interrupt Enable bits.

The following events happen when an interrupt event occurs while the GIE bit is set:

- Current prefetched instruction is flushed
- GIE bit is cleared
- Current Program Counter (PC) is pushed onto the stack
- PC is loaded with the interrupt vector 0004h

The ISR determines the source of the interrupt by polling the interrupt flag bits. The interrupt flag bits must be cleared before exiting the ISR to avoid repeated interrupts. Because the GIE bit is cleared, any interrupt that occurs while executing the ISR will be recorded through its Interrupt Flag, but will not cause the processor to redirect to the interrupt vector.

The `RETFIE` instruction exits the ISR by popping the previous address from the stack and setting the GIE bit.

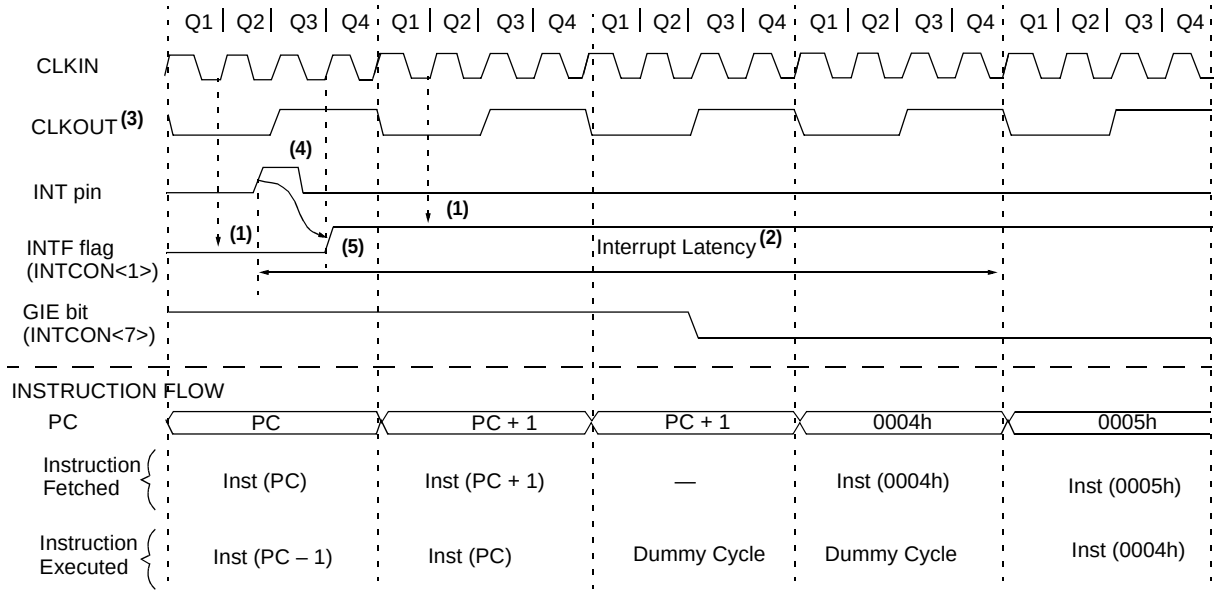
For additional information on a specific interrupt's operation, refer to its peripheral chapter.

Note 1:	Individual interrupt flag bits are set, regardless of the state of any other enable bits.
2:	All interrupts will be ignored while the GIE bit is cleared. Any interrupt occurring while the GIE bit is clear will be serviced when the GIE bit is set again.

4.2 Interrupt Latency

Interrupt latency is defined as the time from when the interrupt event occurs to the time code execution at the interrupt vector begins. The latency for synchronous interrupts is three instruction cycles. For asynchronous interrupts, the latency is three to four instruction cycles, depending on when the interrupt occurs. See Figure 4-2 for timing details.

FIGURE 4-2: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).

2: Asynchronous interrupt latency = 3-4 TCY. Synchronous latency = 3 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

3: CLKOUT is available only in INTOSC and RC Oscillator modes.

4: For minimum width of INT pulse, refer to AC specifications in **Section 23.0 "Electrical Specifications"**.

5: INTF is enabled to be set any time during the Q4-Q1 cycles.

TABLE 9-2: SUMMARY OF ASSOCIATED ADC REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	—	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	75
ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	76
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	44
ANSELB	—	—	ANSB5	ANSB4	—	—	—	—	53
ANSELC	ANSC7	ANSC6	—	—	ANSC3	ANSC2	ANSC1	ANSC0	58
ADRES	ADC Result Register								76
FVRCON	FVRRDY	FVREN	TSEN	TSRNG	—	—	ADFVR1	ADFVR0	81
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
TRISA	—	—	TRISA5	TRISA4	—	TRISA2	TRISA1	TRISA0	43
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	52
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	58

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

15.3.2 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 15-1.

EQUATION 15-1: PWM PERIOD

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

Note: TOSC = 1/FOSC

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

- TMR2 is cleared
- The CCP1 pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is latched from CCPR1L into CCPR1H.

Note: The Timer2 postscaler (refer to **Section 14.1 "Timer2 Operation"**) is not used in the determination of the PWM frequency.

15.3.3 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to multiple registers: CCPR1L register and DC1 and B1 bits of the CCP1CON register. The CCPR1L contains the eight MSbs and the DC1 and B1 bits of the CCP1CON register contain the two LSbs. CCPR1L and DC1 and B1 bits of the CCP1CON register can be written to at any time. The duty cycle value is not latched into CCPR1H until after the period completes (i.e., a match between PR2 and TMR2 registers occurs). While using the PWM, the CCPR1H register is read-only.

Equation 15-2 is used to calculate the PWM pulse width.

Equation 15-3 is used to calculate the PWM duty cycle ratio.

EQUATION 15-2: PULSE WIDTH

$$Pulse\ Width = (CCPR1L{:}CCP1CON\langle 5{:}4\rangle) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

Note: TOSC = 1/FOSC

EQUATION 15-3: DUTY CYCLE RATIO

$$Duty\ Cycle\ Ratio = \frac{(CCPR1L{:}CCP1CON\langle 5{:}4\rangle)}{4(PR2 + 1)}$$

The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

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

When the 10-bit time base matches the CCPR1H and 2-bit latch, then the CCP1 pin is cleared (refer to Figure 15-3).

TABLE 15-6: SUMMARY OF REGISTERS ASSOCIATED WITH PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	—	—	—	—	53
CCP1CON	—	—	DC1	B1	CCP1M3	CCP1M2	CCP1M1	CCP1M0	100
CCPR1L	Capture/Compare/PWM Register Low Byte								—
CCPR1H	Capture/Compare/PWM Register High Byte								—
PR2	Timer2 module Period Register								98
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	99
TMR2	Timer2 module Register								98
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	52
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	58

Legend: - = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the PWM.

16.2 AUSART Baud Rate Generator (BRG)

The Baud Rate Generator (BRG) is an 8-bit timer that is dedicated to the support of both the asynchronous and synchronous AUSART operation.

The SPBRG register determines the period of the free running baud rate timer. In Asynchronous mode, the multiplier of the baud rate period is determined by the BRGH bit of the TXSTA register. In Synchronous mode, the BRGH bit is ignored.

Table 16-3 contains the formulas for determining the baud rate. Example 16-1 provides a sample calculation for determining the baud rate and baud rate error.

Typical baud rates and error values for various Asynchronous modes have been computed for your convenience and are shown in Table 16-5. It may be advantageous to use the high baud rate (BRGH = 1), to reduce the baud rate error.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

EXAMPLE 16-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, and Asynchronous mode with SYNC = 0 and BRGH = 0 (as seen in Table 16-5):

$$Desired\ Baud\ Rate = \frac{FOSC}{64(SPBRG + 1)}$$

Solving for SPBRG:

$$SPBRG = \left(\frac{FOSC}{64(Desired\ Baud\ Rate)}\right) - 1$$

$$= \left(\frac{16000000}{64(9600)}\right) - 1$$

$$= [25.042] = 25$$

$$Actual\ Baud\ Rate = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\%\ Error = \left(\frac{Actual\ Baud\ Rate - Desired\ Baud\ Rate}{Desired\ Baud\ Rate}\right) 100$$

$$= \left(\frac{9615 - 9600}{9600}\right) 100 = 0.16\%$$

TABLE 16-3: BAUD RATE FORMULAS

Configuration Bits		AUSART Mode	Baud Rate Formula
SYNC	BRGH		
0	0	Asynchronous	FOSC/[64 (n+1)]
0	1	Asynchronous	FOSC/[16 (n+1)]
1	x	Synchronous	FOSC/[4 (n+1)]

Legend: x = Don't care, n = value of SPBRG register

TABLE 16-4: REGISTERS ASSOCIATED WITH THE BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	118
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	119
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	117

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for the Baud Rate Generator.

16.3 AUSART Synchronous Mode

Synchronous serial communications are typically used in systems with a single master and one or more slaves. The master device contains the necessary circuitry for baud rate generation and supplies the clock for all devices in the system. Slave devices can take advantage of the master clock by eliminating the internal clock generation circuitry.

There are two signal lines in Synchronous mode: a bidirectional data line and a clock line. Slaves use the external clock supplied by the master to shift the serial data into and out of their respective receive and transmit shift registers. Since the data line is bidirectional, synchronous operation is half-duplex only. Half-duplex refers to the fact that master and slave devices can receive and transmit data but not both simultaneously. The AUSART can operate as either a master or slave device.

Start and Stop bits are not used in synchronous transmissions.

16.3.1 SYNCHRONOUS MASTER MODE

The following bits are used to configure the AUSART for Synchronous Master operation:

- SYNC = 1
- CSRC = 1
- SREN = 0 (for transmit); SREN = 1 (for receive)
- CREN = 0 (for transmit); CREN = 1 (for receive)
- SPEN = 1

Setting the SYNC bit of the TXSTA register configures the device for synchronous operation. Setting the CSRC bit of the TXSTA register configures the device as a master. Clearing the SREN and CREN bits of the RCSTA register ensures that the device is in the Transmit mode, otherwise the device will be configured to receive. Setting the SPEN bit of the RCSTA register enables the AUSART.

16.3.1.1 Master Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a master transmits the clock on the TX/CK line. The TX/CK pin output driver is automatically enabled when the AUSART is configured for synchronous transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One clock cycle is generated for each data bit. Only as many clock cycles are generated as there are data bits.

16.3.1.2 Synchronous Master Transmission

Data is transferred out of the device on the RX/DT pin. The RX/DT and TX/CK pin output drivers are automatically enabled when the AUSART is configured for synchronous master transmit operation.

A transmission is initiated by writing a character to the TXREG register. If the TSR still contains all or part of a previous character, the new character data is held in TXREG until the last bit of the previous character has been transmitted. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXREG is immediately transferred to the TSR. The transmission of the character commences immediately following the transfer of the data to the TSR from the TXREG.

Each data bit changes on the leading edge of the master clock and remains valid until the subsequent leading clock edge.

Note:	The TSR register is not mapped in data memory, so it is not available to the user.

16.3.1.3 Synchronous Master Transmission Setup:

1. Initialize the SPBRG register and the BRGH bit to achieve the desired baud rate (refer to **Section 16.2 “AUSART Baud Rate Generator (BRG)”**).
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Disable Receive mode by clearing bits SREN and CREN.
4. Enable Transmit mode by setting the TXEN bit.
5. If 9-bit transmission is desired, set the TX9 bit.
6. If interrupts are desired, set the TXIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
7. If 9-bit transmission is selected, the ninth bit should be loaded in the TX9D bit.
8. Start transmission by loading data to the TXREG register.

FIGURE 17-4: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

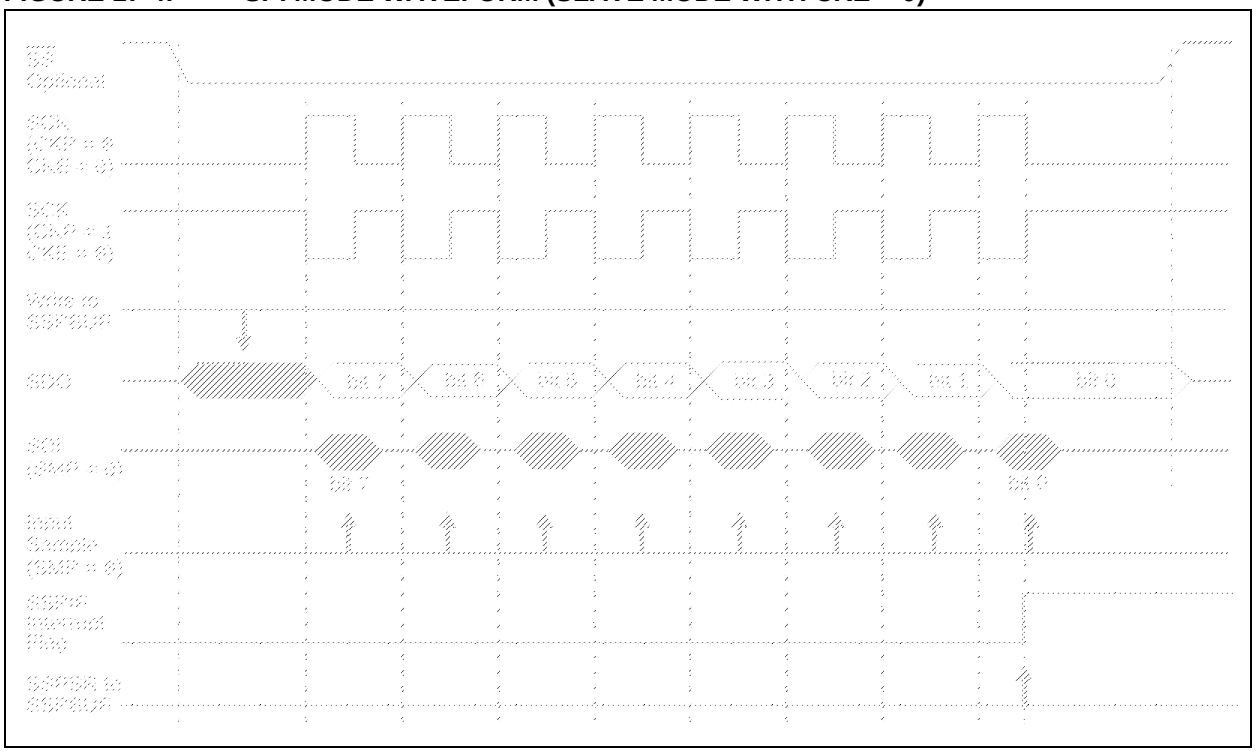
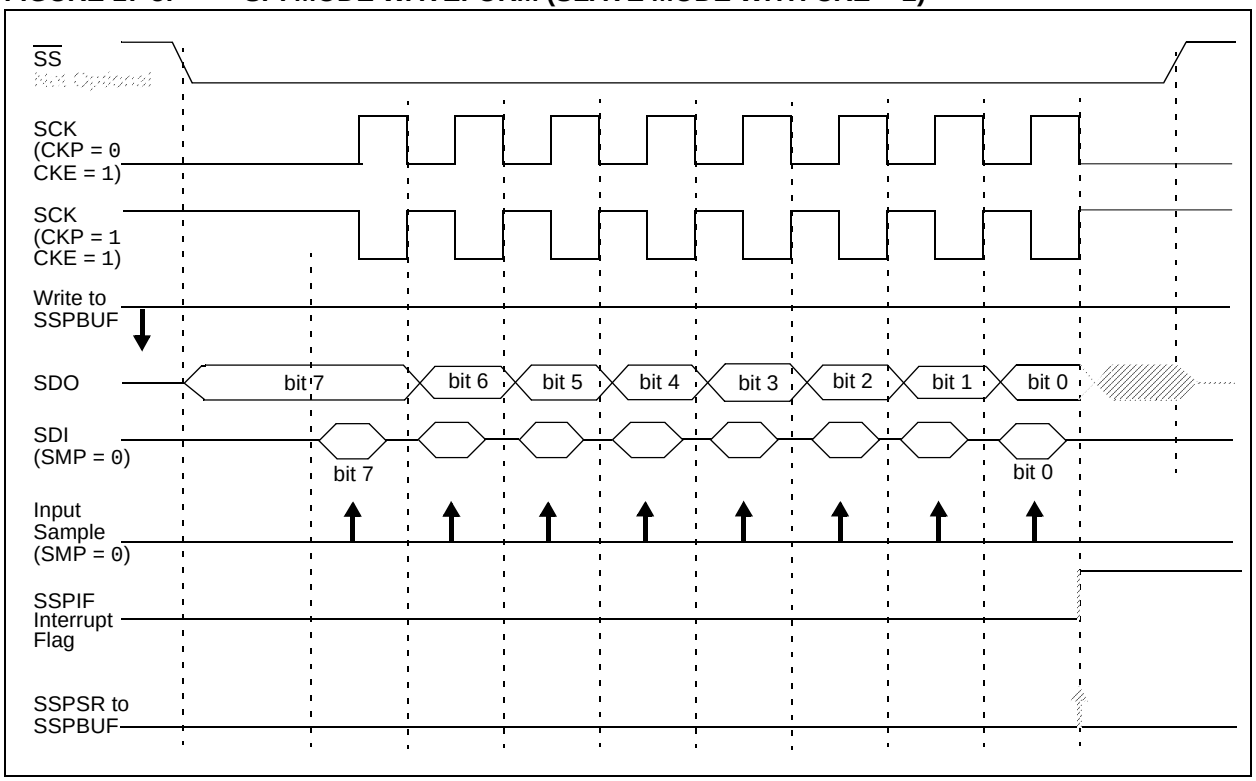


FIGURE 17-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)

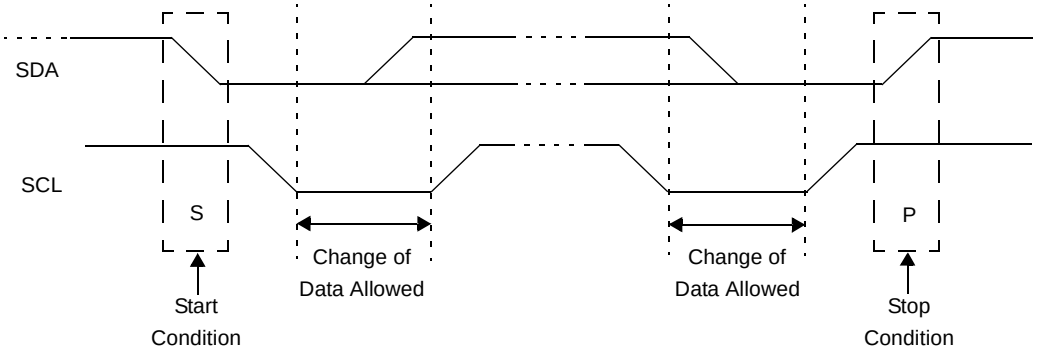


17.2.2 START AND STOP CONDITIONS

During times of no data transfer (Idle time), both the clock line (SCL) and the data line (SDA) are pulled high through external pull-up resistors. The Start and Stop conditions determine the start and stop of data transmission. The Start condition is defined as a high-to-low transition of the SDA line while SCL is high. The Stop condition is defined as a low-to-high transition of the SDA line while SCL is high.

Figure 17-9 shows the Start and Stop conditions. A master device generates these conditions for starting and terminating data transfer. Due to the definition of the Start and Stop conditions, when data is being transmitted, the SDA line can only change state when the SCL line is low.

FIGURE 17-9: START AND STOP CONDITIONS



17.2.3 ACKNOWLEDGE

After the valid reception of an address or data byte, the hardware automatically will generate the Acknowledge ($\overline{ACK}$) pulse and load the SSPBUF register with the received value currently in the SSPSR register. There are certain conditions that will cause the SSP module not to generate this $\overline{ACK}$ pulse. They include any or all of the following:

- The Buffer Full bit, BF of the SSPSTAT register, was set before the transfer was received.
- The SSP Overflow bit, SSPOV of the SSPCON register, was set before the transfer was received.
- The SSP module is being operated in Firmware Master mode.

In such a case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF of the PIR1 register is set. Table 17-2 shows the results of when a data transfer byte is received, given the status of bits BF and SSPOV. Flag bit BF is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

TABLE 17-2: DATA TRANSFER RECEIVED BYTE ACTIONS

Status Bits as Data Transfer is Received		SSPSR → SSPBUF	Generate $\overline{ACK}$ Pulse	Set bit SSPIF (SSP Interrupt occurs if enabled)
BF	SSPOV			
0	0	Yes	Yes	Yes
1	0	No	No	Yes
1	1	No	No	Yes
0	1	No	No	Yes

Note 1: Shaded cells show the conditions where the user software did not properly clear the overflow condition.

18.5 Writing to Flash Program Memory

A word of the Flash program memory may only be written to if the word is in an unprotected segment of memory.

Flash program memory may only be written to if the destination address is in a segment of memory that is not write-protected, as defined in bits WRT<1:0> of the Configuration Word Register 2. Flash program memory must be written in 32-word rows. See Figure 18-2 for more details. A row consists of 32 words with sequential addresses, with a lower boundary defined by an address, where PMADR<4:0>= 00000. All row writes to program memory are done as 32-word erase and one to 32-word write operations. The write operation is edge-aligned. Crossing boundaries is not recommended, as the operation will only affect the new boundary, wrapping the data values at the same time. Once the write control bit is set, the Program Memory (PM) controller will immediately write the data. Program execution is stalled while the write is in progress.

To erase a program memory row, the address of the row to erase must be loaded into the PMADRH:PMADRL register pair. A row consists of 32 words so, when selecting a row, PMADR<4:0> are ignored. After the Address has been set up, then the following sequence of events must be executed:

1. Set the WREN and FREE control bits of the PMCON1 register.
2. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
3. Set the WR control bit of the PMCON1 register.

To write program data, it must first be loaded into the buffer latches (see Figure 18-2). This is accomplished by first writing the destination address to PMADRL and PMADRH and then writing the data to PMDATA and PMDATH. After the address and data have been set up, then the following sequence of events must be executed:

1. Set the WREN control bit of the PMCON1 register.
2. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
3. Set the WR control bit of the PMCON1 register.

All 32 buffer register locations should be written to with correct data. If less than 32 words are being written to in the block of 32 words, then a read from the program memory location(s) not being written to must be performed. This takes the data from the program location(s) not being written and loads it into the PMDATL and PMDATH registers. Then, the sequence of events to transfer data to the buffer registers must be executed.

When the LWLO bit is '1', the write sequence will only load the buffer register and will not actually initiate the write to program Flash:

1. Set the WREN and LWLO bits of the PMCON1 register.
2. Write 55h, then AAh, to PMCON2 (Flash programming unlock sequence).
3. Set control bit WR of the PMCON1 register to begin the write operation.

Note:	Self-write execution to Flash memory cannot be done while running in low power PFM and Voltage Regulator modes. Therefore, executing a self-write will put the PFM and voltage regulator into High Power mode for the duration of the sequence.

To transfer data from the buffer registers to the program memory, the last word to be written should be written to the PMDATH:PMDATL register pair. Then, the following sequence of events must be executed:

1. Clear the LWLO bit of the PMCON1 Register.
2. Write 55h, then AAh, to PMCON2 (Flash programming sequence).
3. Set control bit WR of the PMCON1 register to begin the write operation.
4. Two NOP instructions must follow the setting of the WR bit.

This is necessary to provide time for the address and to be provided to the Program Flash Memory to be put in the write latches.

Note:	An ICD break that occurs during the 55h - AAh – Set WR bit sequence will interrupt the timing of the sequence and prevent the unlock sequence from occurring. In this case, no write will be initiated, as there was no operation to complete.

No automatic erase occurs upon the initiation of the write; if the program Flash needs to be erased before writing, the row (32 words) must be previously erased.

After the "BSF PMCON1, WR" instruction, the processor requires two cycles to set up the erase/write operation.

The user must place two NOP instructions after the WR bit is set. These two instructions will also be forced in hardware to NOP, but if an ICD break occurs at this point, the forcing to NOP will be lost.

21.0 INSTRUCTION SET SUMMARY

The PIC16(L)F720/721 instruction set is highly orthogonal and is comprised of three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

Each PIC16 instruction is a 14-bit word divided into an **opcode,** which specifies the instruction type and one or more **operands,** which further specify the operation of the instruction. The formats for each of the categories is presented in Figure 21-1, while the various opcode fields are summarized in Table 21-1.

Table 21-2 lists the instructions recognized by the MPASM™ assembler.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator, which selects the bit affected by the operation, while 'f' represents the address of the file in which the bit is located.

For **literal and control** operations, 'k' represents an 8-bit or 11-bit constant, or literal value.

One instruction cycle consists of four oscillator periods; for an oscillator frequency of 4 MHz, this gives a nominal instruction execution time of 1 μs. All instructions are executed within a single instruction cycle, unless a conditional test is true, or the program counter is changed as a result of an instruction. When this occurs, the execution takes two instruction cycles, with the second cycle executed as a NOP.

All instruction examples use the format '0xhh' to represent a hexadecimal number, where 'h' signifies a hexadecimal digit.

21.1 Read-Modify-Write Operations

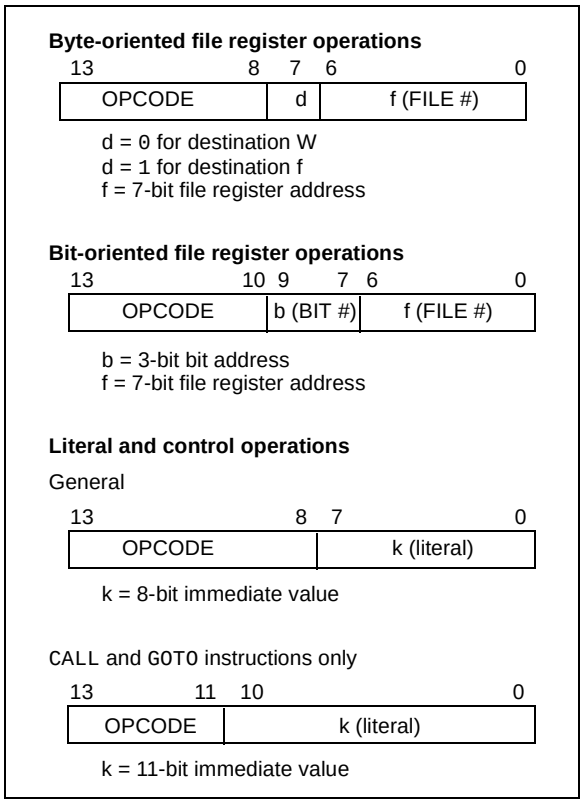
Any instruction that specifies a file register as part of the instruction performs a Read-Modify-Write (R-M-W) operation. The register is read, the data is modified, and the result is stored according to either the instruction, or the destination designator 'd'. A read operation is performed on a register even if the instruction writes to that register.

For example, a CLRF PORTB instruction will read PORTB, clear all the data bits, then write the result back to PORTB. This example would have the unintended consequence of clearing the condition that set the RABIF flag.

TABLE 21-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1.
PC	Program Counter
$\overline{TO}$	Time-out bit
C	Carry bit
DC	Digit carry bit
Z	Zero bit
$\overline{PD}$	Power-down bit

FIGURE 21-1: GENERAL FORMAT FOR INSTRUCTIONS



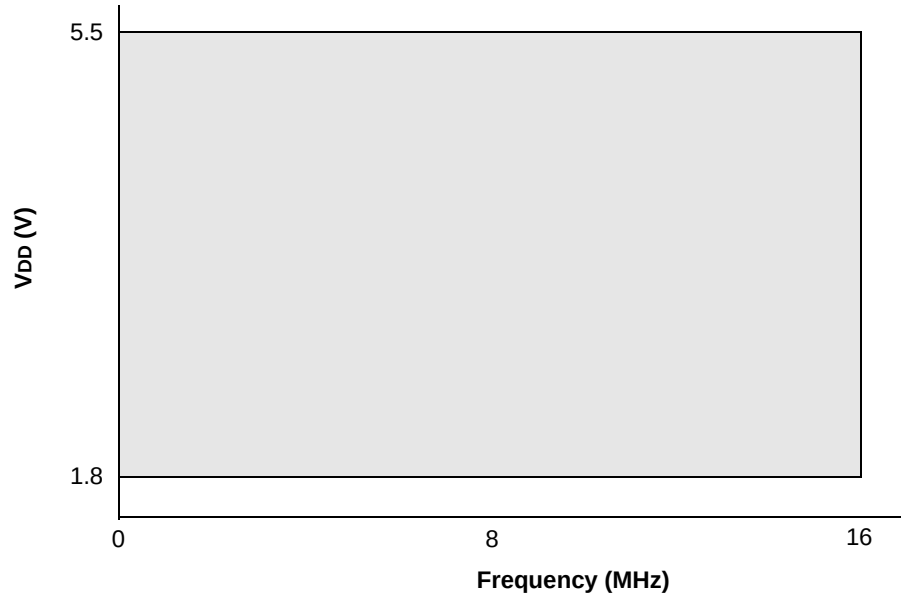
23.5 Thermal Considerations

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	62.2	°C/W	20-pin PDIP package
			75.0	°C/W	20-pin SOIC package
			89.3	°C/W	20-pin SSOP package
			43.0	°C/W	20-pin QFN 4x4mm package
TH02	θJC	Thermal Resistance Junction to Case	27.5	°C/W	20-pin PDIP package
			23.1	°C/W	20-pin SOIC package
			31.1	°C/W	20-pin SSOP package
			5.3	°C/W	20-pin QFN 4x4mm 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 x VDD(1)
TH06	PI/O	I/O Power Dissipation	—	W	PI/O = Σ (IOL * VOL) + Σ (IOH * (VDD - VOH))
TH07	PDER	Derated Power	—	W	PDER = PDMAX (TJ - TA)/θJA(2)

Note 1: IDD is current to run the chip alone without driving any load on the output pins.
2: TA = Ambient Temperature; TJ = Junction Temperature

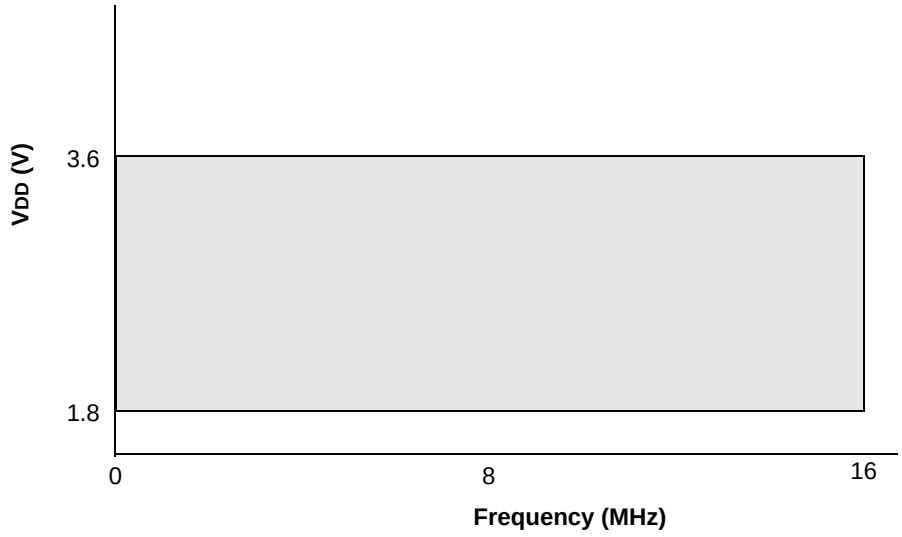
23.7 AC Characteristics: PIC16F720/721-I/E

FIGURE 23-3: PIC16F720/721 VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

FIGURE 23-4: PIC16LF720/721 VOLTAGE FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

TABLE 23-4: RESET, WATCHDOG TIME, POWER-UP TIMER, AND BROWN-OUT RESET PARAMETERS

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ TA ≤ +125°C							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
30*	TMCL	$\overline{MCLR}$ Pulse Width (low)	2 5	— —	— —	µs µs	VDD = 5V, -40°C to +85°C VDD = 5V(1)
31	TWDT	Standard Watchdog Timer Time-out Period (No Prescaler)(2)	10 10	18 18	27 33	ms ms	VDD = 3.3V-5V, -40°C to +85°C VDD = 3.3V-5V(1)
33*	TPWRT	Power-up Timer Period, $\overline{PWRTE}$ = 0	40	65	140	ms	
34*	TIOZ	I/O high-impedance from $\overline{MCLR}$ Low or Watchdog Timer Reset	—	—	2.0	µs	
35	VBOR	Brown-out Reset Voltage	1.80	1.9	2.1	V	
36*	VHYST	Brown-out Reset Hysteresis	0	25	50	mV	
37*	TBORDC	Brown-out Reset DC Response Time	1	3	5 10	µs	VDD ≤ VBOR, -40°C to +85°C VDD ≤ VBOR

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Voltages above 3.6V require that the regulator be enabled.

2: Design Target. If unable to meet this target, the maximum can be increased, but the minimum cannot be changed.

FIGURE 23-9: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

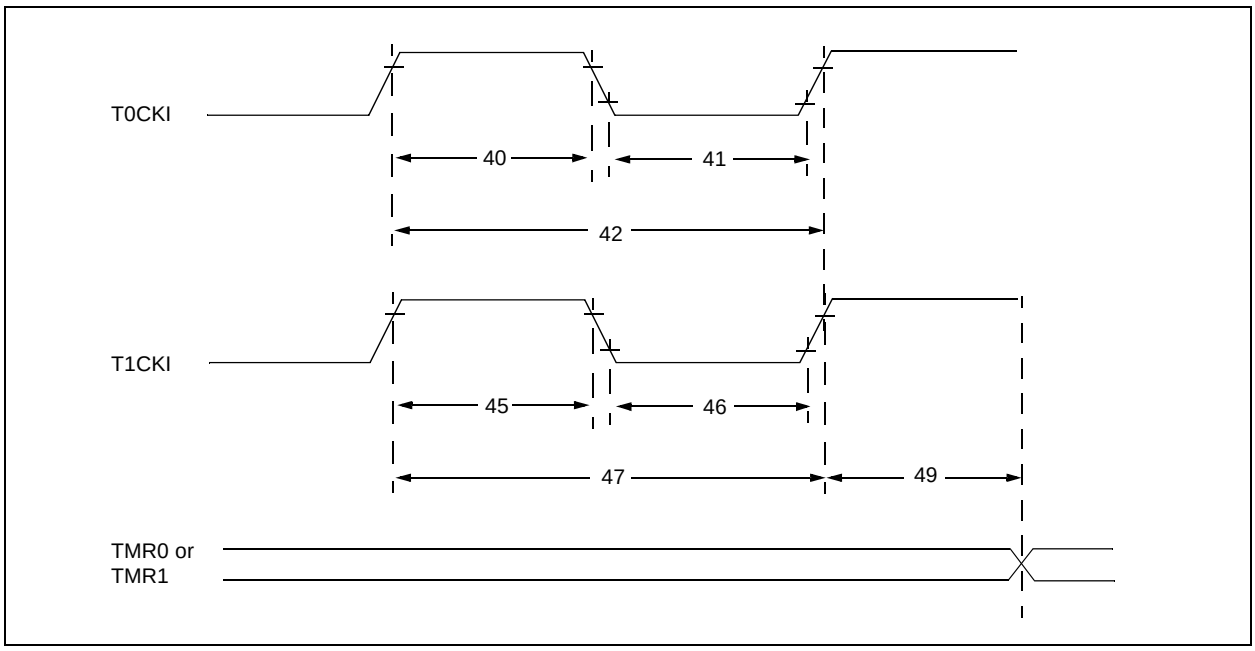


FIGURE 24-7: PIC16LF720/721 MAX. IDD vs. FOSC OVER VDD, MFINTOSC

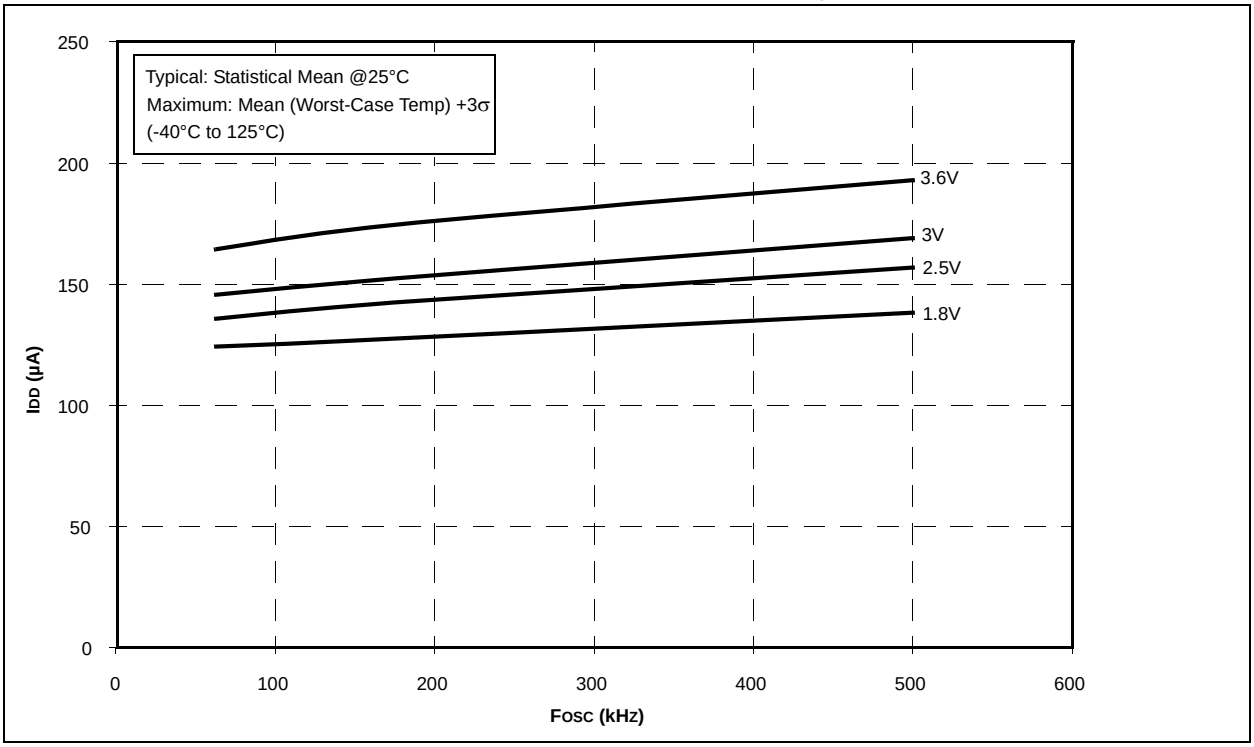
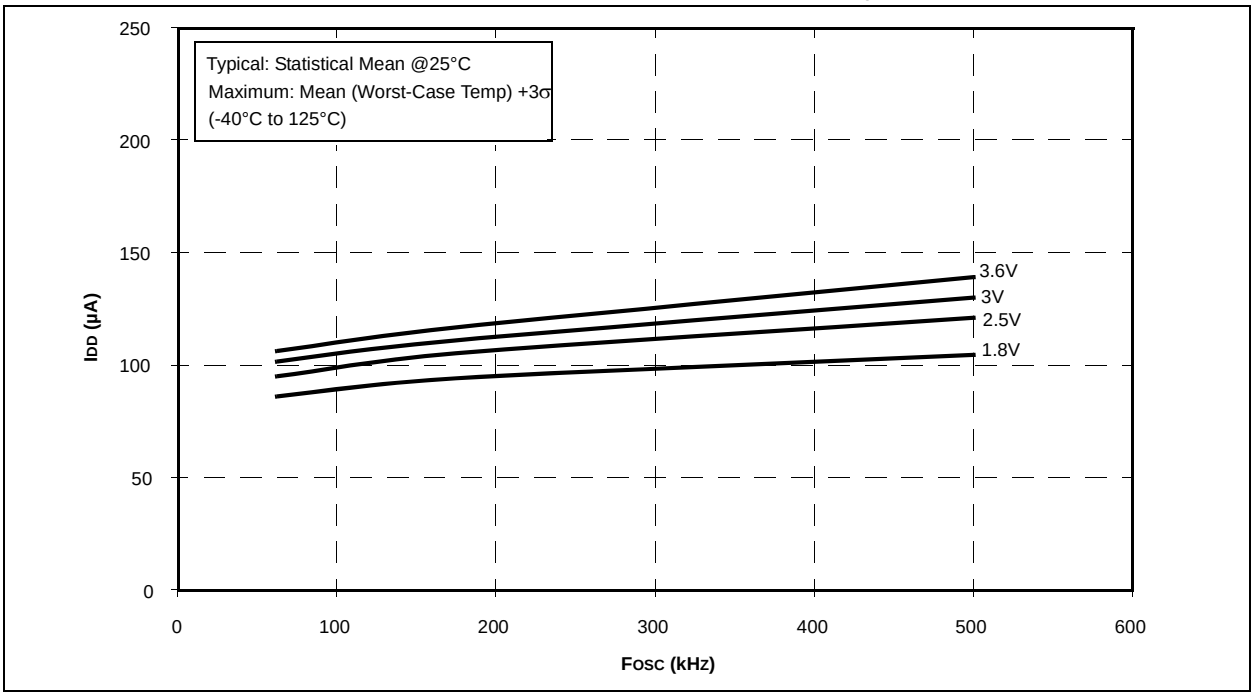
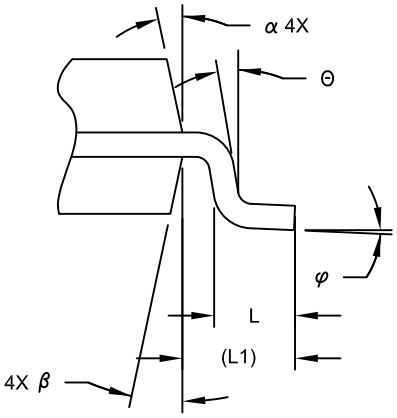


FIGURE 24-8: PIC16LF720/721 TYPICAL IDD vs. FOSC OVER VDD, MFINTOSC

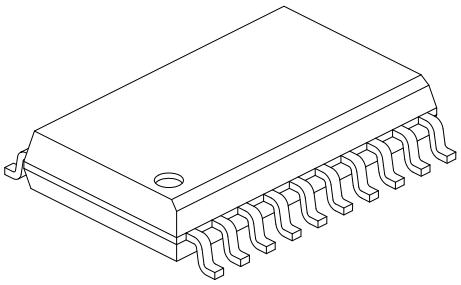


20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		12.80 BSC	
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1		1.40 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.20	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

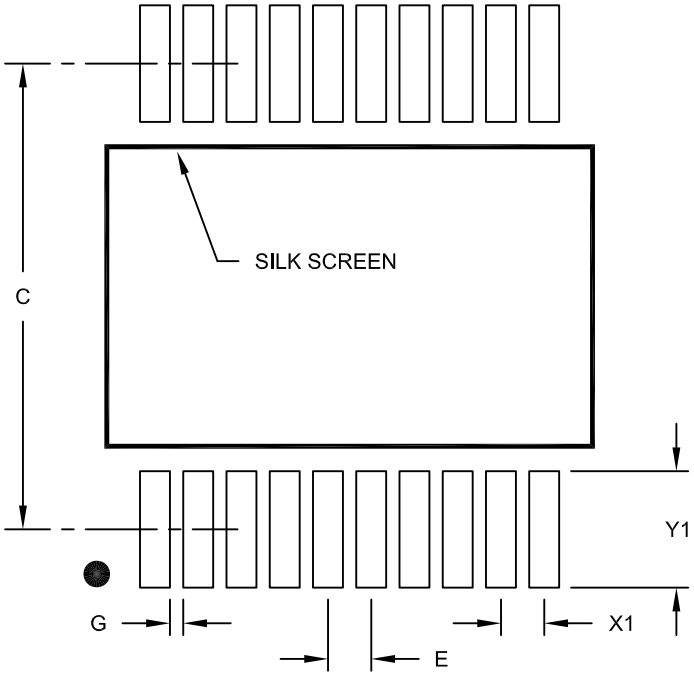
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2

20-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X20)	X1			0.45
Contact Pad Length (X20)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072A