



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	I ² C, SPI
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 5x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f72-i-ss

2.2.2.6 PCON Register

Note: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

Note: $\overline{\text{BOR}}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent RESETS to see if $\overline{\text{BOR}}$ is clear, indicating a brown-out has occurred. The $\overline{\text{BOR}}$ status bit is a 'don't care' and is not necessarily predictable if the brown-out circuit is disabled (by clearing the BOREN bit in the Configuration word).

The Power Control (PCON) register contains a flag bit to allow differentiation between a Power-on Reset (POR), a Brown-out Reset, an external $\overline{\text{MCLR}}$ Reset and WDT Reset.

REGISTER 2-6: PCON: POWER CONTROL REGISTER (ADDRESS 8Eh)

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

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

bit 1 **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 **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 Brown-out Reset occurs)

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

2.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). When doing 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, "Implementing a Table Read" (AN556).

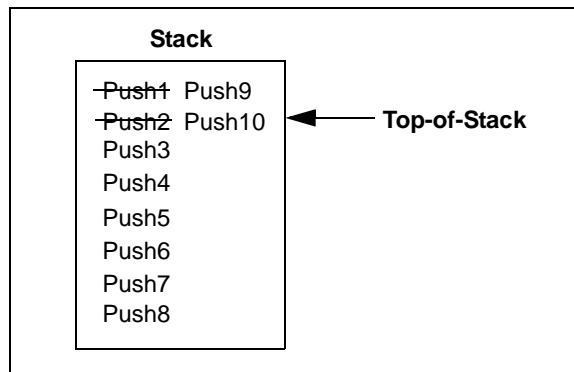
2.3.2 STACK

The stack allows a combination of up to eight program calls and interrupts to occur. The stack contains the return address from this branch in program execution.

Mid-range devices have an 8-level deep x 13-bit wide hardware stack. The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSH'd onto the stack when a CALL instruction is executed, or an interrupt causes a branch. The stack is POP'd in the event of a RETURN, RETLW or RETFIE instruction execution. PCLATH is not modified when the stack is PUSH'd or POP'd.

After the stack has been PUSH'd 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). An example of the overwriting of the stack is shown in Figure 2-4.

FIGURE 2-4: STACK MODIFICATION



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

The CALL and GOTO instructions provide 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 pushed onto 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 PIC16F72 device ignores the paging bit PCLATH<4:3>. The use of PCLATH<4:3> as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

2.5 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a pointer). This is indirect addressing.

A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-1.

EXAMPLE 2-1: INDIRECT ADDRESSING

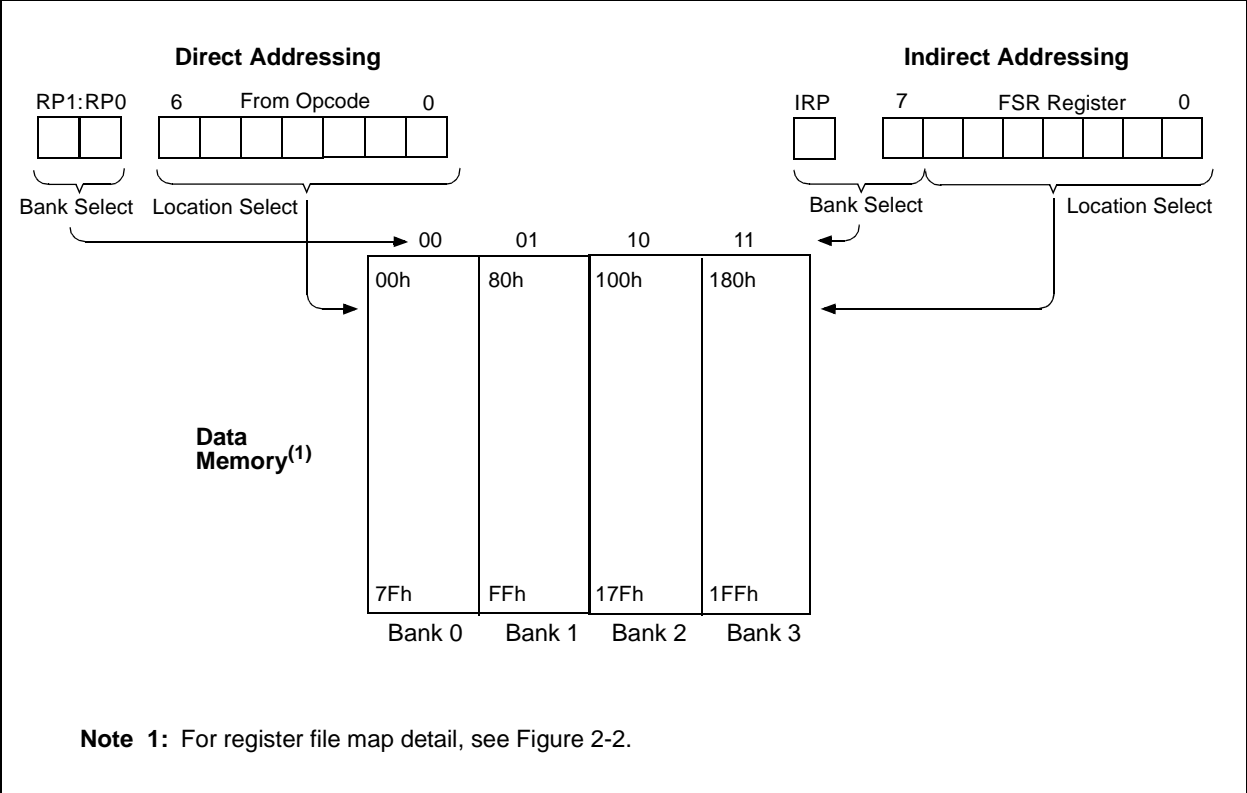
```

        movlw 0x20 ;initialize pointer
        movwf FSR ;to RAM
NEXT    clrf  INDF ;clear INDF register
        incf  FSR ;inc pointer
        btfss FSR,4 ;all done?
        goto  NEXT ;NO, clear next
CONTINUE
        :          ;YES, continue
    
```

An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 2-5.

PIC16F72

FIGURE 2-5: DIRECT/INDIRECT ADDRESSING



4.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 4-1 is a block diagram of the Timer0 module and the prescaler shared with the WDT.

Additional information on the Timer0 module is available in the PIC™ Mid-Range MCU Family Reference Manual (DS33023).

4.1 Timer0 Operation

Timer mode is selected by clearing bit T0CS (OPTION<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0 register is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register.

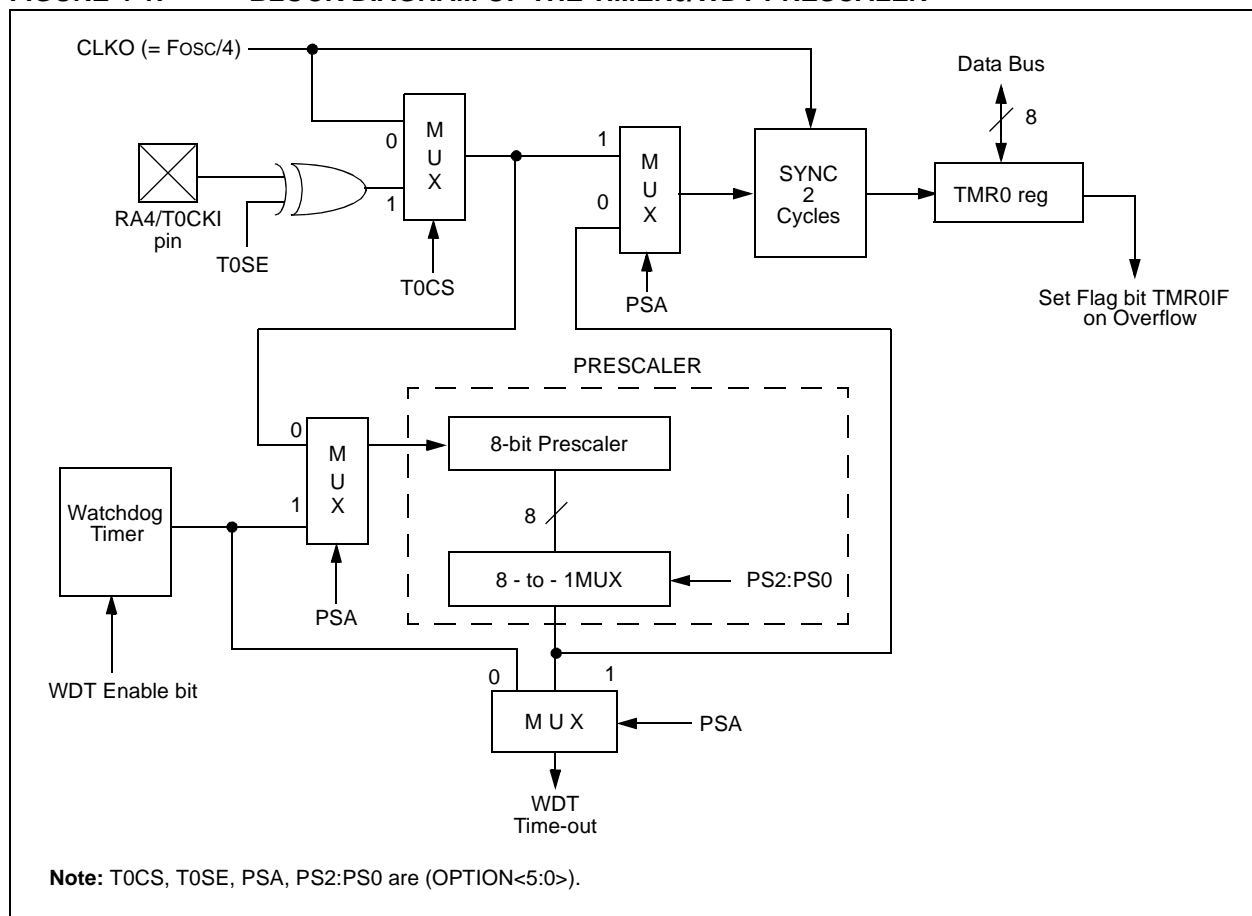
Counter mode is selected by setting bit T0CS (OPTION<5>). In Counter mode, Timer0 will increment, either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit T0SE (OPTION<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 4.3.

The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler is not readable or writable. Section 4.4 details the operation of the prescaler.

4.2 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit TMR0IF (INTCON<2>). The interrupt can be masked by clearing bit TMR0IE (INTCON<5>). Bit TMR0IF must be cleared in software by the Timer0 module Interrupt Service Routine, before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from SLEEP, since the timer is shut-off during SLEEP.

FIGURE 4-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER



7.0 READING PROGRAM MEMORY

The FLASH Program Memory is readable during normal operation over the entire VDD range. It is indirectly addressed through Special Function Registers (SFR). Up to 14-bit wide numbers can be stored in memory for use as calibration parameters, serial numbers, packed 7-bit ASCII, etc. Executing a program memory location containing data that forms an invalid instruction results in a NOP.

There are five SFRs used to read the program and memory:

- PMCON1
- PMDATL
- PMDATH
- PMADRL
- PMADRH

The program memory allows word reads. Program memory access allows for checksum calculation and reading calibration tables.

When interfacing to the program memory block, the PMDATH:PMDATL registers form a two-byte word, which holds the 14-bit data for reads. The PMADRH:PMADRL registers form a two-byte word, which holds the 13-bit address of the FLASH location being accessed. This device has up to 2K words of program FLASH, with an address range from 0h to 07FFh. The unused upper bits PMDATH<7:6> and PMADRH<7:5> are not implemented and read as zeros.

7.1 PMADR

The address registers can address up to a maximum of 8K words of program FLASH.

When selecting a program address value, the MSByte of the address is written to the PMADRH register and the LSByte is written to the PMADRL register. The upper MSbits of PMADRH must always be clear.

7.2 PMCON1 Register

PMCON1 is the control register for memory accesses.

The control bit RD initiates read operations. This bit cannot be cleared, only set, in software. It is cleared in hardware at the completion of the read operation.

REGISTER 7-1: PMCON1: PROGRAM MEMORY CONTROL REGISTER 1 (ADDRESS 18Ch)

R-1	U-0	U-0	U-0	U-0	U-0	U-0	R/S-0
reserved	—	—	—	—	—	—	RD
bit 7							bit 0

bit 7 **Reserved:** Read as '1'

bit 6-1 **Unimplemented:** Read as '0'

bit 0 **RD:** Read Control bit

1 = Initiates a FLASH read, RD is cleared in hardware. The RD bit can only be set (not cleared) in software.

0 = Does not initiate a FLASH read

Legend:

W = Writable bit

R = Readable bit

'1' = Bit is set

U = Unimplemented bit, read as '0'

S = Settable bit

'0' = Bit is cleared

-n = Value at POR

x = Bit is unknown

9.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

9.1 SSP Module Overview

The Synchronous Serial Port (SSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be Serial EEPROMs, shift registers, display drivers, A/D converters, etc. The SSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I²C)

An overview of I²C operations and additional information on the SSP module can be found in the PICTM Mid-Range MCU Family Reference Manual (DS33023).

Refer to Application Note AN578, *"Use of the SSP Module in the I²C Multi-Master Environment."*

9.2 SPI Mode

This section contains register definitions and operational characteristics of the SPI module.

SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) RC5/SDO
- Serial Data In (SDI) RC4/SDI/SDA
- Serial Clock (SCK) RC3/SCK/SCL

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select (\overline{SS}) RA5/AN4/ \overline{SS}

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits in the SSPCON register (SSPCON<5:0>) and SSPSTAT<7:6>. These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (IDLE state of SCK)
- Clock edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

FIGURE 9-2: SPI MODE TIMING, MASTER MODE

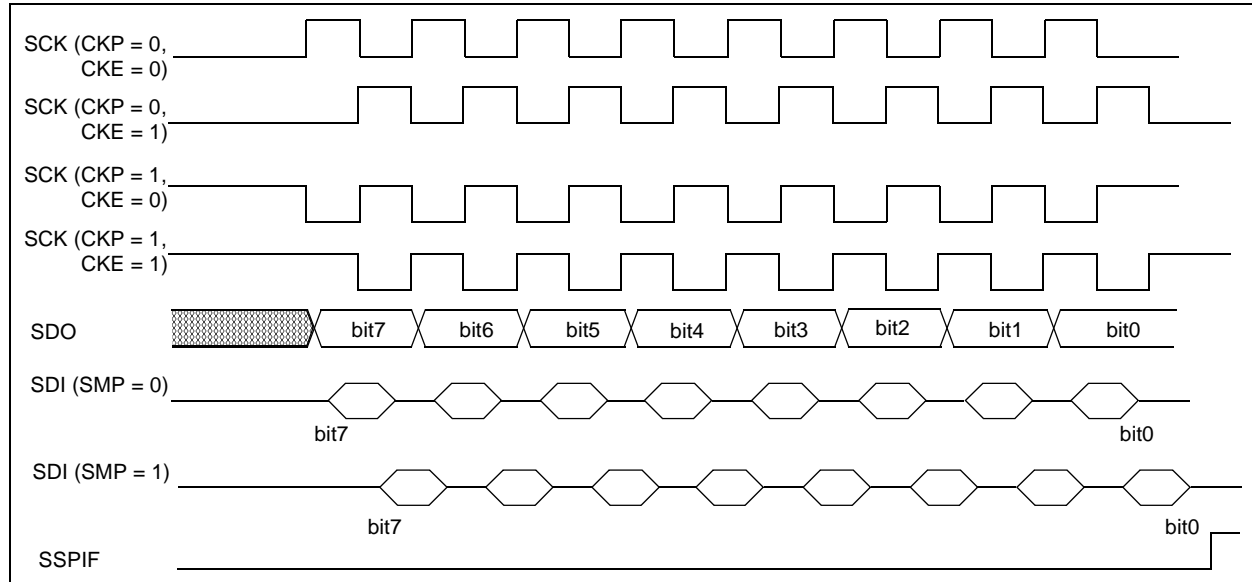


FIGURE 9-3: SPI MODE TIMING (SLAVE MODE WITH CKE = 0)

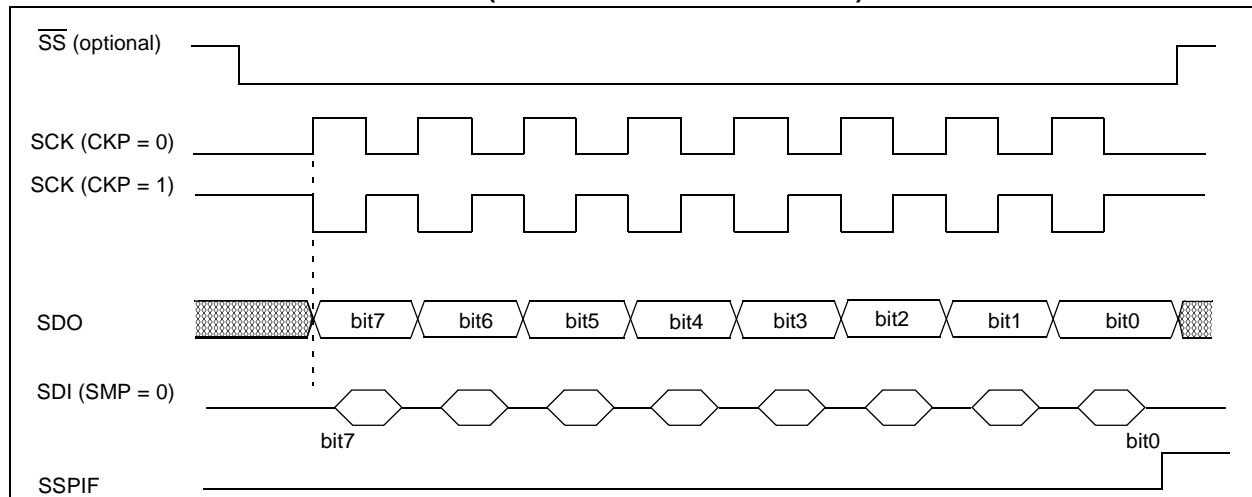


FIGURE 9-4: SPI MODE TIMING (SLAVE MODE WITH CKE = 1)

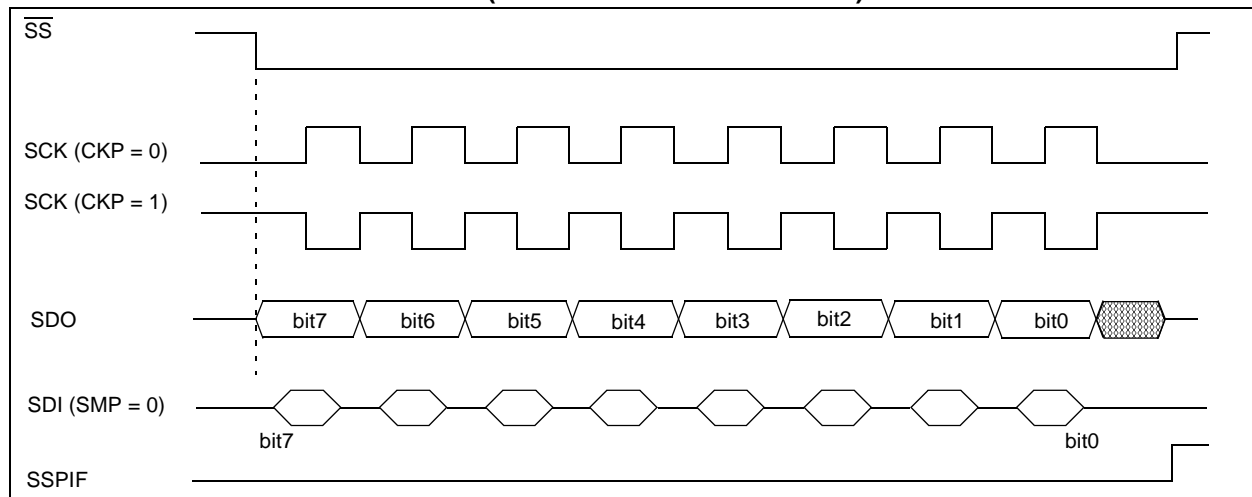


FIGURE 9-6: I²C WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)

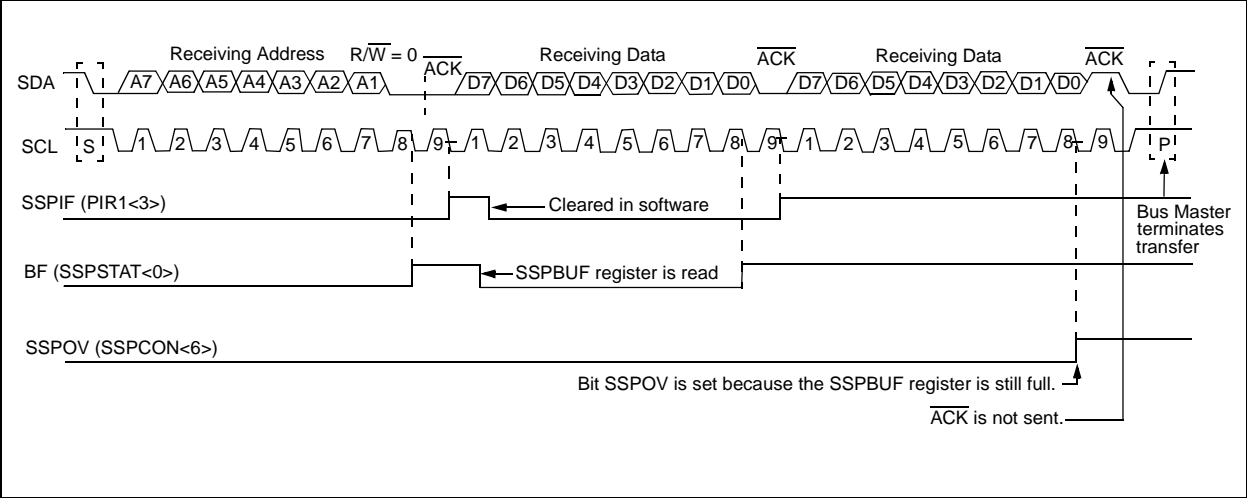
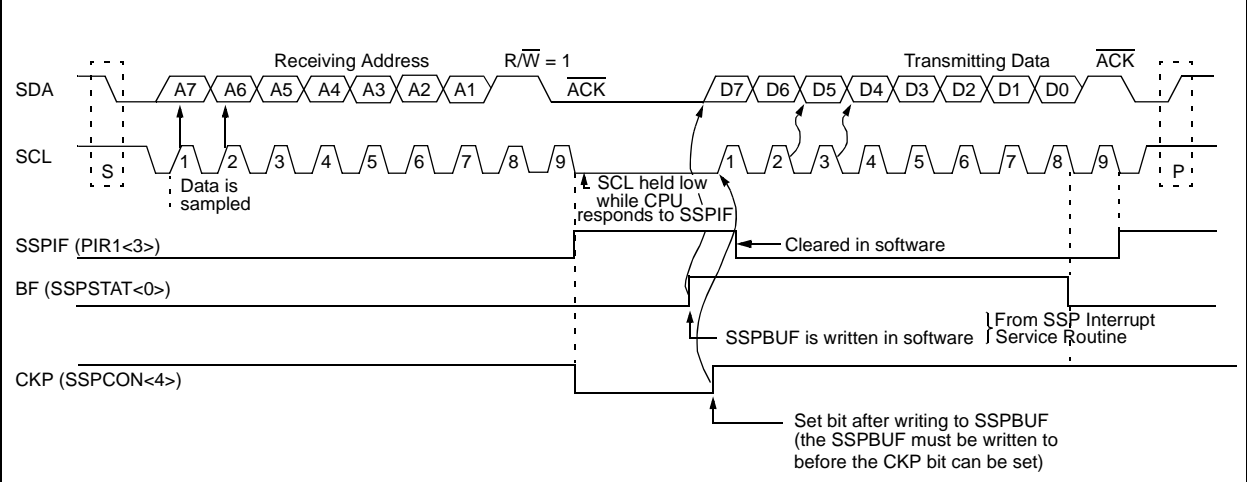


FIGURE 9-7: I²C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)



11.11.1 INT INTERRUPT

External interrupt on the RB0/INT pin is edge triggered, either rising, if bit INTEDG (OPTION<6>) is set, or falling, if the INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, flag bit INTF (INTCON<1>) is set. This interrupt can be disabled by clearing enable bit INTE (INTCON<4>). Flag bit INTF must be cleared in software in the Interrupt Service Routine before re-enabling this interrupt. The INT interrupt can wake-up the processor from SLEEP, if bit INTE was set prior to going into SLEEP. The status of global interrupt enable bit GIE decides whether or not the processor branches to the interrupt vector following wake-up. See Section 11.14 for details on SLEEP mode.

11.11.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set flag bit TMR0IF (INTCON<2>). The interrupt can be enabled/disabled by setting/clearing enable bit TMR0IE (INTCON<5>) (see Section 4.0).

11.11.3 PORTB INTCON CHANGE

An input change on PORTB<7:4> sets flag bit RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit RBIE (INTCON<4>) (see Section 3.2).

11.12 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (i.e., W, STATUS registers). This will have to be implemented in software, as shown in Example 11-1.

For the PIC16F72 device, the register W_TEMP must be defined in both banks 0 and 1 and must be defined at the same offset from the bank base address (i.e., if W_TEMP is defined at 20h in bank 0, it must also be defined at A0h in bank 1). The register STATUS_TEMP is only defined in bank 0.

EXAMPLE 11-1: SAVING STATUS, W AND PCLATH REGISTERS IN RAM

```

MOVWF  W_TEMP          ;Copy W to TEMP register
SWAPF  STATUS,W        ;Swap status to be saved into W
CLRF   STATUS           ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF  STATUS_TEMP     ;Save status to bank zero STATUS_TEMP register
:
: (ISR)                 ;Insert user code here
:
SWAPF  STATUS_TEMP,W   ;Swap STATUS_TEMP register into W
                        ; (sets bank to original state)
MOVWF  STATUS          ;Move W into STATUS register
SWAPF  W_TEMP,F        ;Swap W_TEMP
SWAPF  W_TEMP,W        ;Swap W_TEMP into W
    
```

PIC16F72

11.13 Watchdog Timer (WDT)

The Watchdog Timer is a free running, on-chip RC oscillator that does not require any external components. This RC oscillator is separate from the RC oscillator of the OSC1/CLKI pin. That means that the WDT will run, even if the clock on the OSC1/CLKI and OSC2/CLKO pins of the device has been stopped, for example, by execution of a `SLEEP` instruction.

During normal operation, a WDT time-out generates a device RESET (Watchdog Timer Reset). If the device is in `SLEEP` mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer Wake-up). The `TO` bit in the STATUS register will be cleared upon a Watchdog Timer time-out.

The WDT can be permanently disabled by clearing configuration bit `WDTEN` (see Section 11.1).

WDT time-out period values may be found in the Electrical Specifications section under parameter #31. Values for the WDT prescaler (actually a postscaler, but shared with the Timer0 prescaler) may be assigned using the `OPTION` register.

Note 1: The `CLRWDT` and `SLEEP` instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET condition.

2: When a `CLRWDT` instruction is executed and the prescaler is assigned to the WDT, the prescaler count will be cleared, but the prescaler assignment is not changed.

FIGURE 11-11: WATCHDOG TIMER BLOCK DIAGRAM

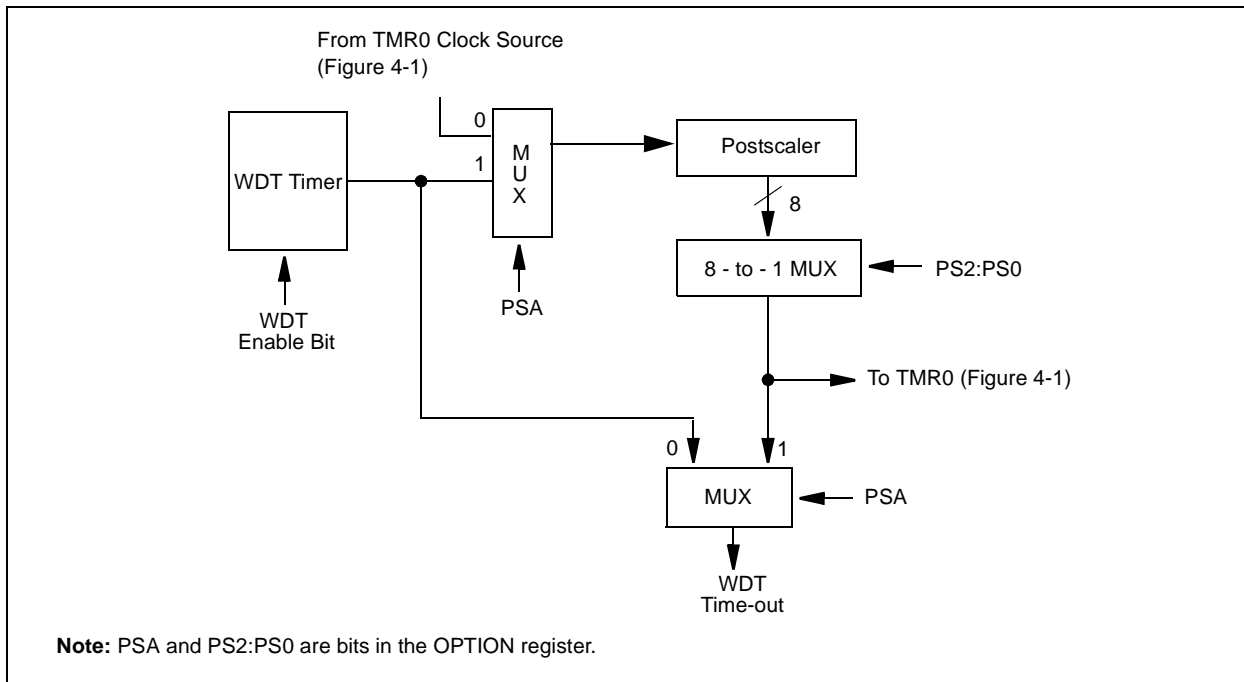


TABLE 11-7: SUMMARY OF WATCHDOG TIMER REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	(1)	BOREN ⁽¹⁾	—	CP	PWRTEN ⁽¹⁾	WDTEN	FOSC1	FOSC0
81h,181h	OPTION	RBPUP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Legend: Shaded cells are not used by the Watchdog Timer.

Note 1: See Register 11-1 for operation of these bits.

12.1 Instruction Descriptions

ADDLW **Add Literal and W**

Syntax: `[label] ADDLW k`

Operands: $0 \leq k \leq 255$

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

Status Affected: C, DC, Z

Description: The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

ANDWF **AND W with f**

Syntax: `[label] ANDWF f,d`

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: $(W) .AND. (f) \rightarrow (\text{destination})$

Status Affected: Z

Description: AND the W register with register 'f'. If 'd' = '0', the result is stored in the W register. If 'd' = '1', the result is stored back in register 'f'.

ADDWF **Add W and f**

Syntax: `[label] ADDWF f,d`

Operands: $0 \leq f \leq 127$
 $d \in [0,1]$

Operation: $(W) + (f) \rightarrow (\text{destination})$

Status Affected: C, DC, Z

Description: Add the contents of the W register with register 'f'. If 'd' = '0', the result is stored in the W register. If 'd' = '1', the result is stored back in register 'f'.

BCF **Bit Clear f**

Syntax: `[label] BCF f,b`

Operands: $0 \leq f \leq 127$
 $0 \leq b \leq 7$

Operation: $0 \rightarrow (f)$

Status Affected: None

Description: Bit 'b' in register 'f' is cleared.

ANDLW **AND Literal with W**

Syntax: `[label] ANDLW k`

Operands: $0 \leq k \leq 255$

Operation: $(W) .AND. (k) \rightarrow (W)$

Status Affected: Z

Description: The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register.

BSF **Bit Set f**

Syntax: `[label] BSF f,b`

Operands: $0 \leq f \leq 127$
 $0 \leq b \leq 7$

Operation: $1 \rightarrow (f)$

Status Affected: None

Description: Bit 'b' in register 'f' is set.

COMF	Complement f
Syntax:	[<i>label</i>] COMF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) \rightarrow (destination)
Status Affected:	Z
Description:	The contents of register 'f' are complemented. If 'd' = '0', the result is stored in W. If 'd' = '1', the result is stored back in register 'f'.

GOTO	Unconditional Branch
Syntax:	[<i>label</i>] GOTO k
Operands:	$0 \leq k \leq 2047$
Operation:	$k \rightarrow PC<10:0>$ $PCLATH<4:3> \rightarrow PC<12:11>$
Status Affected:	None
Description:	GOTO is an unconditional branch. The eleven-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two-cycle instruction.

DECF	Decrement f
Syntax:	[<i>label</i>] DECF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) - 1 \rightarrow (destination)
Status Affected:	Z
Description:	Decrement register 'f'. If 'd' = '0', the result is stored in the W register. If 'd' = '1', the result is stored back in register 'f'.

INCF	Increment f
Syntax:	[<i>label</i>] INCF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) + 1 \rightarrow (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' = '0', the result is placed in the W register. If 'd' = '1', the result is placed back in register 'f'.

DECFSZ	Decrement f, Skip if 0
Syntax:	[<i>label</i>] DECFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) - 1 \rightarrow (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' = '0', the result is placed in the W register. If 'd' = '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2 Tcy instruction.

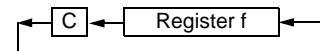
INCFSZ	Increment f, Skip if 0
Syntax:	[<i>label</i>] INCFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) + 1 \rightarrow (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' = '0', the result is placed in the W register. If 'd' = '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2 Tcy instruction.

RETFIE Return from Interrupt

Syntax: [*label*] RETFIE
 Operands: None
 Operation: TOS → PC,
 1 → GIE
 Status Affected: None

RLF Rotate Left f through Carry

Syntax: [*label*] RLF f,d
 Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
 Operation: See description below
 Status Affected: C
 Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' = '0', the result is placed in the W register. If 'd' = '1', the result is stored back in register 'f'.

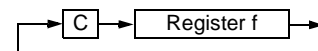


RETLW Return with Literal in W

Syntax: [*label*] RETLW k
 Operands: $0 \leq k \leq 255$
 Operation: k → (W);
 TOS → PC
 Status Affected: None
 Description: The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.

RRF Rotate Right f through Carry

Syntax: [*label*] RRF f,d
 Operands: $0 \leq f \leq 127$
 $d \in [0,1]$
 Operation: See description below
 Status Affected: C
 Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' = '0', the result is placed in the W register. If 'd' = '1', the result is placed back in register 'f'.



RETURN Return from Subroutine

Syntax: [*label*] RETURN
 Operands: None
 Operation: TOS → PC
 Status Affected: None
 Description: Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.

SLEEP

Syntax: [*label*] SLEEP
 Operands: None
 Operation: 00h → WDT,
 0 → WDT prescaler,
 1 → \overline{TO} ,
 0 → \overline{PD}
 Status Affected: \overline{TO} , \overline{PD}
 Description: The power-down status bit, \overline{PD} is cleared. Time-out status bit, \overline{TO} is set. Watchdog Timer and its prescaler are cleared. The processor is put into SLEEP mode with the oscillator stopped.

TABLE 14-6: SPI MODE REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
70*	TssL2scH, TssL2scL	$\overline{SS}\downarrow$ to SCK \downarrow or SCK \uparrow input	T _{CY}	—	—	ns	
71*	TscH	SCK input high time (Slave mode)	T _{CY} + 20	—	—	ns	
72*	TscL	SCK input low time (Slave mode)	T _{CY} + 20	—	—	ns	
73*	TdiV2scH, TdiV2scL	Setup time of SDI data input to SCK edge	100	—	—	ns	
74*	Tsch2diL, TscL2diL	Hold time of SDI data input to SCK edge	100	—	—	ns	
75*	TdoR	SDO data output rise time	—	10	25	ns	Standard(F) Extended(LF)
76*	TdoF	SDO data output fall time	—	10	25	ns	
77*	TssH2doZ	$\overline{SS}\uparrow$ to SDO output hi-impedance	10	—	50	ns	
78*	TscR	SCK output rise time (Master mode)	—	10	25	ns	Standard(F) Extended(LF)
79*	TscF	SCK output fall time (Master mode)	—	10	25	ns	
80*	Tsch2doV, TscL2doV	SDO data output valid after SCK edge	—	—	50 145	ns ns	Standard(F) Extended(LF)
81*	TdoV2scH, TdoV2scL	SDO data output setup to SCK edge	T _{CY}	—	—	ns	
82*	TssL2doV	SDO data output valid after $\overline{SS}\downarrow$ edge	—	—	50	ns	
83*	Tsch2ssH, TscL2ssH	$\overline{SS}\uparrow$ after SCK edge	1.5 T _{CY} + 40	—	—	ns	

* 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.

FIGURE 14-14: I²C BUS START/STOP BITS TIMING

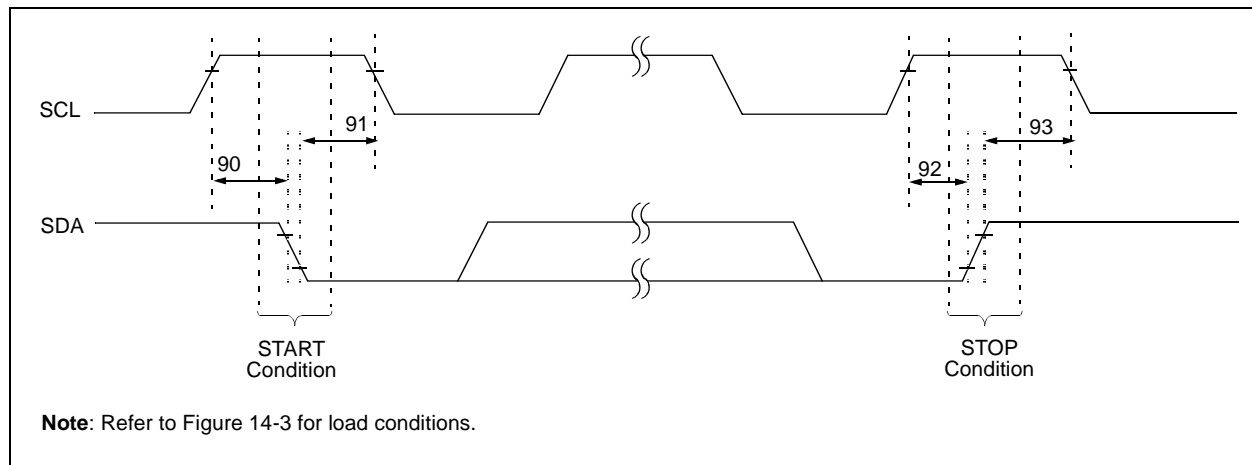


TABLE 14-8: I²C BUS DATA REQUIREMENTS

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
100*	THIGH	Clock High Time	100 kHz mode	4.0	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5 Tcy	—		
101*	TLOW	Clock Low Time	100 kHz mode	4.7	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μs	Device must operate at a minimum of 10 MHz
			SSP Module	1.5 Tcy	—		
102*	TR	SDA and SCL Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 C _B	300	ns	C _B is specified to be from 10 - 400 pF
103*	TF	SDA and SCL Fall Time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1 C _B	300	ns	C _B is specified to be from 10 - 400 pF
90*	TSU:STA	START Condition Setup Time	100 kHz mode	4.7	—	μs	Only relevant for Repeated START condition
			400 kHz mode	0.6	—	μs	
91*	THD:STA	START Condition Hold Time	100 kHz mode	4.0	—	μs	After this period, the first clock pulse is generated
			400 kHz mode	0.6	—	μs	
106*	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107*	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	(Note 2)
			400 kHz mode	100	—	ns	
92*	TSU:STO	STOP Condition Setup Time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109*	TAA	Output Valid from Clock	100 kHz mode	—	3500	ns	(Note 1)
			400 kHz mode	—	—	ns	
110*	TBUF	Bus Free Time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
	CB	Bus Capacitive Loading		—	400	pF	

* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

2: A Fast mode (400 kHz) I²C bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement TSU:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line T_R max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

PIC16F72

**TABLE 14-9: A/D CONVERTER CHARACTERISTICS: PIC16F72 (INDUSTRIAL)
PIC16LF72 (INDUSTRIAL)**

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
A01	NR	Resolution	PIC16F72	—	—	8 bits	bit	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
			PIC16LF72	—	—	8 bits	bit	VREF = VDD = 2.2V
A02	EABS	Total Absolute Error		—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A03	EIL	Integral Linearity Error		—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A04	EDL	Differential Linearity Error		—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A05	EFS	Full Scale Error		—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A06	EOFF	Offset Error		—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A10	—	Monotonicity (Note 3)		—	guaranteed	—	—	VSS ≤ VAIN ≤ VREF
A20	VREF	Reference Voltage		2.5	—	VDD+0.3	V	-40°C to +85°C
				2.2	—	VDD+0.3	V	0°C to +85°C
A25	VAIN	Analog Input Voltage		VSS - 0.3	—	VREF + 0.3	V	
A30	ZAIN	Recommended Impedance of Analog Voltage Source		—	—	10.0	kΩ	
A40	IAD	A/D Conversion Current (VDD)	PIC16F72	—	180	—	μA	Average current consumption when A/D is on (Note 1).
			PIC16LF72	—	90	—	μA	
A50	IREF	VREF input current (Note 2)		N/A	—	± 5	μA	During VAIN acquisition. During A/D Conversion cycle.
				—	—	500	μA	

* 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: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.

2: VREF current is from the RA3 pin or the VDD pin, whichever is selected as a reference input.

3: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

FIGURE 15-5: TYPICAL I_{DD} vs. F_{OSC} OVER V_{DD} (LP MODE)

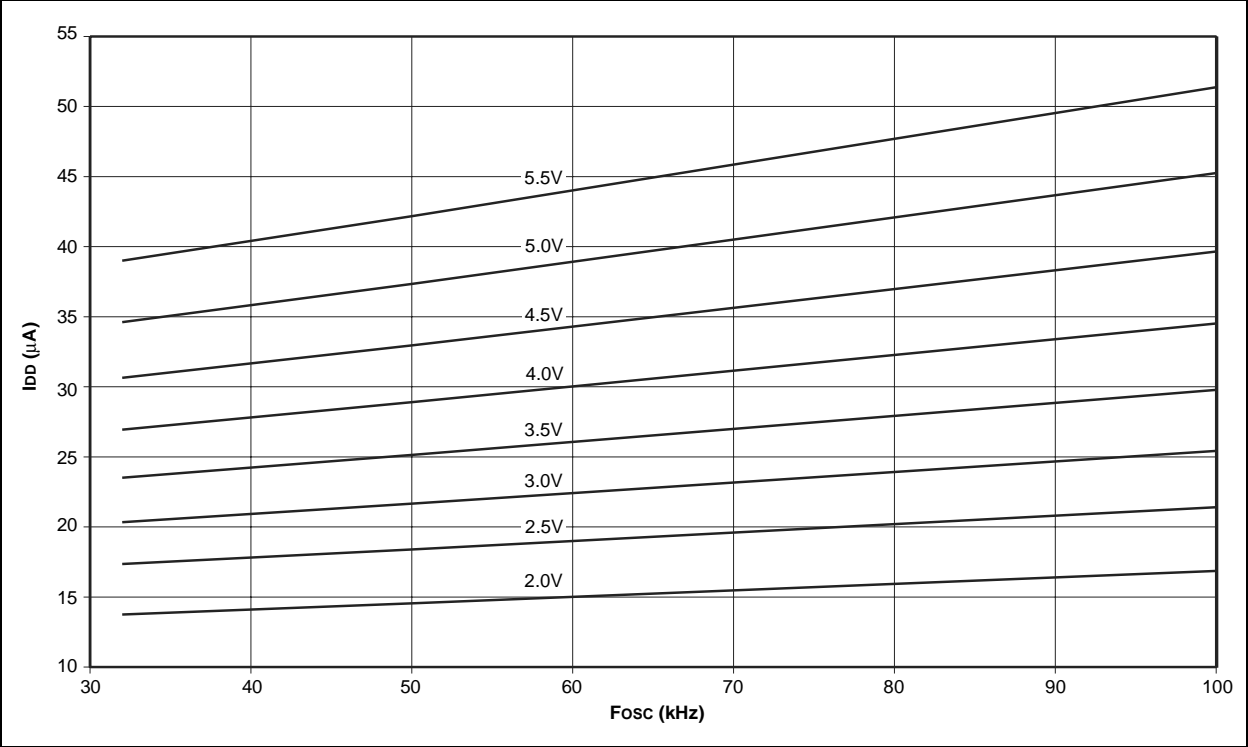
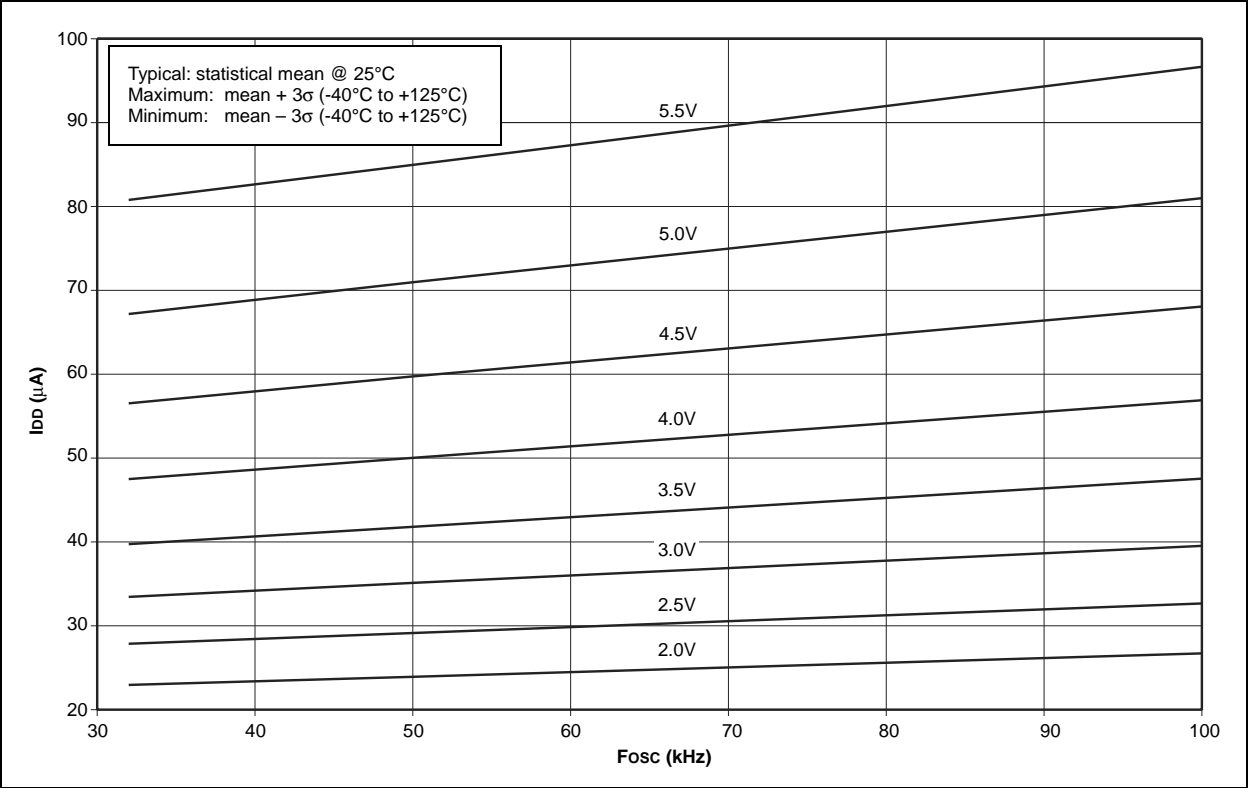


FIGURE 15-6: MAXIMUM I_{DD} vs. F_{OSC} OVER V_{DD} (LP MODE)



PIC16F72

NOTES:

Timing Diagrams

A/D Conversion	105
Brown-out Reset	96
Capture/Compare/PWM (CCP1)	98
CLKO and I/O	95
External Clock	94
I ² C Bus Data	102
I ² C Bus START/STOP bits	101
I ² C Reception (7-bit Address)	50
I ² C Transmission (7-bit Address)	50
RESET, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer	96
Slow Rise Time (MCLR Tied to VDD Through RC Network)	68
SPI Master Mode	47
SPI Master Mode (CKE = 0, SMP = 0)	99
SPI Master Mode (CKE = 1, SMP = 1)	99
SPI Slave Mode (CKE = 0)	47, 100
SPI Slave Mode (CKE = 1)	47, 100
Time-out Sequence on Power-up (MCLR Tied to VDD Through Pull-up Resistor)	67
Time-out Sequence on Power-up (MCLR Tied to VDD Through RC Network): Case 1	67
Time-out Sequence on Power-up (MCLR Tied to VDD Through RC Network): Case 2	67
Timer0 and Timer1 External Clock	97
Wake-up from SLEEP through Interrupt	72
Timing Parameter Symbolology	93
TMR1H Register	9
TMR1L Register	9
TMR2 Register	9
TMR2ON bit	36
TOUTPS0 bit	36
TOUTPS1 bit	36
TOUTPS2 bit	36
TOUTPS3 bit	36
TRISA Register	10, 21
TRISB Register	10, 23
TRISC Register	10, 25

U

UA	44
Update Address bit, UA	44

W

Wake-up from SLEEP	59, 71
Interrupts	65, 66
MCLR Reset	66
WDT Reset	66
Watchdog Timer (WDT)	59, 70
Associated Registers	70
Enable (WDTEN bit)	70
Postscaler. See Postscaler, WDT	
Programming Considerations	70
RC Oscillator	70
Time-out Period	70
WDT Reset, Normal Operation	62, 65, 66
WDT Reset, SLEEP	62, 65, 66
WCOL	45
Write Collision Detect bit, WCOL	45
WWW, On-Line Support	3