



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

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

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

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

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	13
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	A/D 4x8b
Oscillator Type	External
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/pic16c716t-20i-ss

TABLE 1-1: PIC16C712/716 PINOUT DESCRIPTION (CONTINUED)

Pin Name	PIC16C712/716		Pin Type	Buffer Type	Description
	DIP, SOIC	SSOP			
RB0/INT RB0 INT	6	7	I/O I	TTL ST	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. Digital I/O External Interrupt
RB1/T1OSO/T1CKI RB1 T1OSO T1CKI	7	8	I/O O I	TTL — ST	Digital I/O Timer1 oscillator output. Connects to crystal in oscillator mode. Timer1 external clock input.
RB2/T1OSI RB2 T1OSI	8	9	I/O I	TTL —	Digital I/O Timer1 oscillator input. Connects to crystal in oscillator mode.
RB3/CCP1 RB3 CCP1	9	10	I/O I/O	TTL ST	Digital I/O Capture1 input, Compare1 output, PWM1 output.
RB4	10	12	I/O	TTL	Digital I/O Interrupt on change pin.
RB5	11	12	I/O	TTL	Digital I/O Interrupt on change pin.
RB6	12	13	I/O I	TTL ST	Digital I/O Interrupt on change pin. ICSP programming clock.
RB7	13	14	I/O I/O	TTL ST	Digital I/O Interrupt on change pin. ICSP programming data.
VSS	5	5, 6	P	—	Ground reference for logic and I/O pins.
VDD	14	15, 16	P	—	Positive supply for logic and I/O pins.

Legend: TTL = TTL-compatible input CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels
OD = Open drain output
SM = SMBus compatible input. An external resistor is required if this pin is used as an output
NPU = N-channel pull-up PU = Weak internal pull-up
No-P diode = No P-diode to VDD AN = Analog input or output
I = input O = output
P = Power L = LCD Driver

2.2.2.1 Status Register

The STATUS register, shown in Figure 2-4, contains the arithmetic status of the ALU, the Reset status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, as with 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 the Z, C or DC bits from the STATUS register. For other instructions, not affecting any Status bits, see the "Instruction Set Summary."

Note 1: These devices do not use bits IRP and RP1 (STATUS<7:6>). Maintain these bits clear to ensure upward compatibility with future products.

2: The C and DC bits operate as a borrow and digit borrow bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

FIGURE 2-4: STATUS REGISTER (ADDRESS 03h, 83h)

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	C
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
-n = Value at POR Reset

bit 7: **IRP:** Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h-1FFh) – not implemented, maintain clear
0 = Bank 0, 1 (00h-FFh) – not implemented, maintain clear

bit 6-5: **RP1:RP0:** Register Bank Select bits (used for direct addressing)
01 = Bank 1 (80h-FFh)
00 = Bank 0 (00h-7Fh)
Each bank is 128 bytes
Note: RP1 = not implemented, maintain clear

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

bit 3: **\overline{PD} :** Power-down bit
1 = After power-up or by the `CLRWDWT` 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/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions) (for borrow the polarity is reversed)
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/borrow bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
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: For borrow the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

PIC16C712/716

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.

EXAMPLE 2-1: INDIRECT ADDRESSING

- Register file 05 contains the value 10h
- Register file 06 contains the value 0Ah
- Load the value 05 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 06)
- A read of the INDF register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although Status bits may be affected).

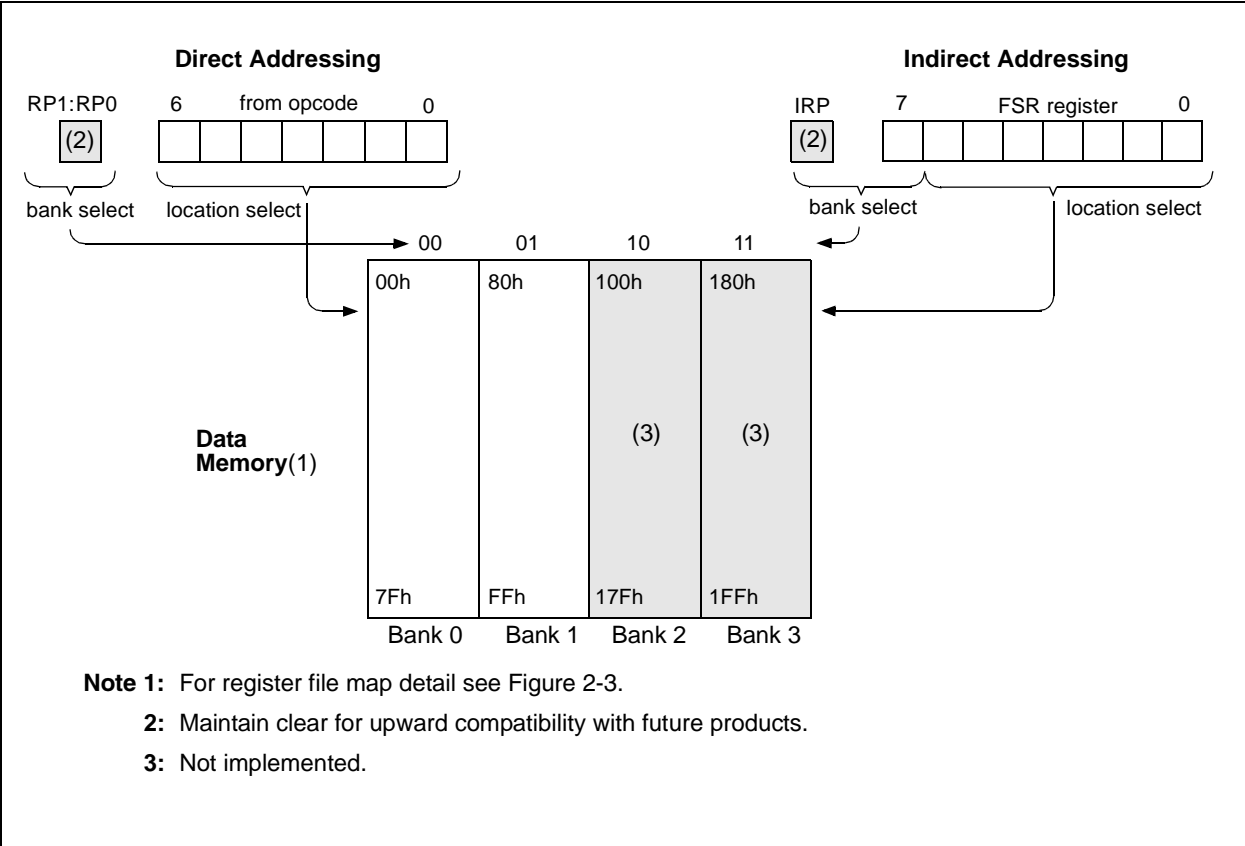
A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-2.

EXAMPLE 2-2: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

```
MOV LW 0x20 ;initialize pointer
MOV WF 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 2-10. However, IRP is not used in the PIC16C712/716.

FIGURE 2-10: DIRECT/INDIRECT ADDRESSING



3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PIC® Mid-Range Reference Manual, (DS33023).

3.1 PORTA and the TRISA Register

PORTA is a 5-bit wide bidirectional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input, (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output, (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register 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, the value is modified, and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers.

PORTA pins, RA3:0, are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

Note: On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 3-1: INITIALIZING PORTA

```
BCF    STATUS, RP0    ;
CLRF   PORTA          ; Initialize PORTA by
                        ; clearing output
                        ; data latches
BSF    STATUS, RP0    ; Select Bank 1
MOVLW  0xEF           ; Value used to
                        ; initialize data
                        ; direction
MOVWF  TRISA          ; Set RA<3:0> as inputs
                        ; RA<4> as outputs
BCF    STATUS, RP0    ; Return to Bank 0
```

PIC16C712/716

3.2 PORTB and the TRISB Register

PORTB is an 8-bit wide bidirectional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input, (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output, (i.e., put the contents of the output latch on the selected pin).

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit $\overline{\text{RBPU}}$ (OPTION_REG<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

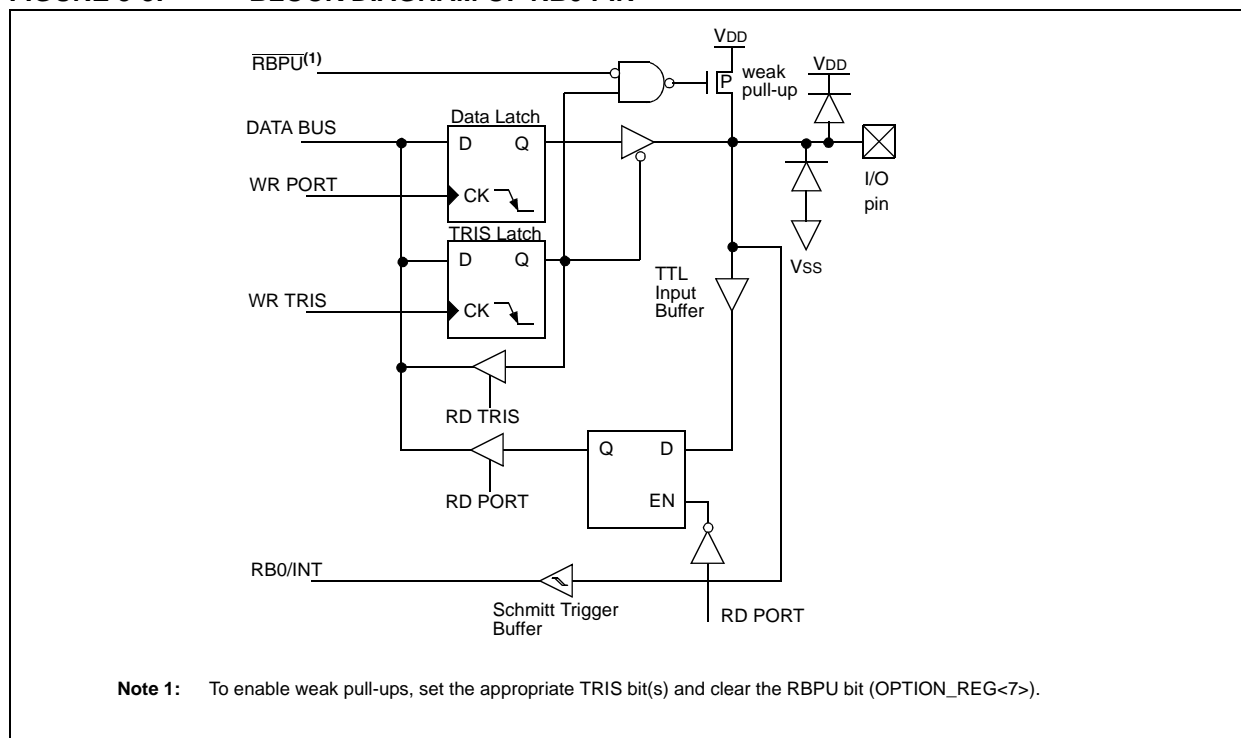
EXAMPLE 3-2: INITIALIZING PORTB

```
BCF    STATUS, RP0 ;
CLRF   PORTB       ; Initialize PORTB by
                   ; clearing output
                   ; data latches

BSF    STATUS, RP0 ; Select Bank 1
MOVLW  0xCF        ; Value used to
                   ; initialize data
                   ; direction

MOVWF  TRISB       ; Set RB<3:0> as inputs
                   ; RB<5:4> as outputs
                   ; RB<7:6> as inputs
```

FIGURE 3-3: BLOCK DIAGRAM OF RB0 PIN



PIC16C712/716

TABLE 3-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
81h	OPTION_REG	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

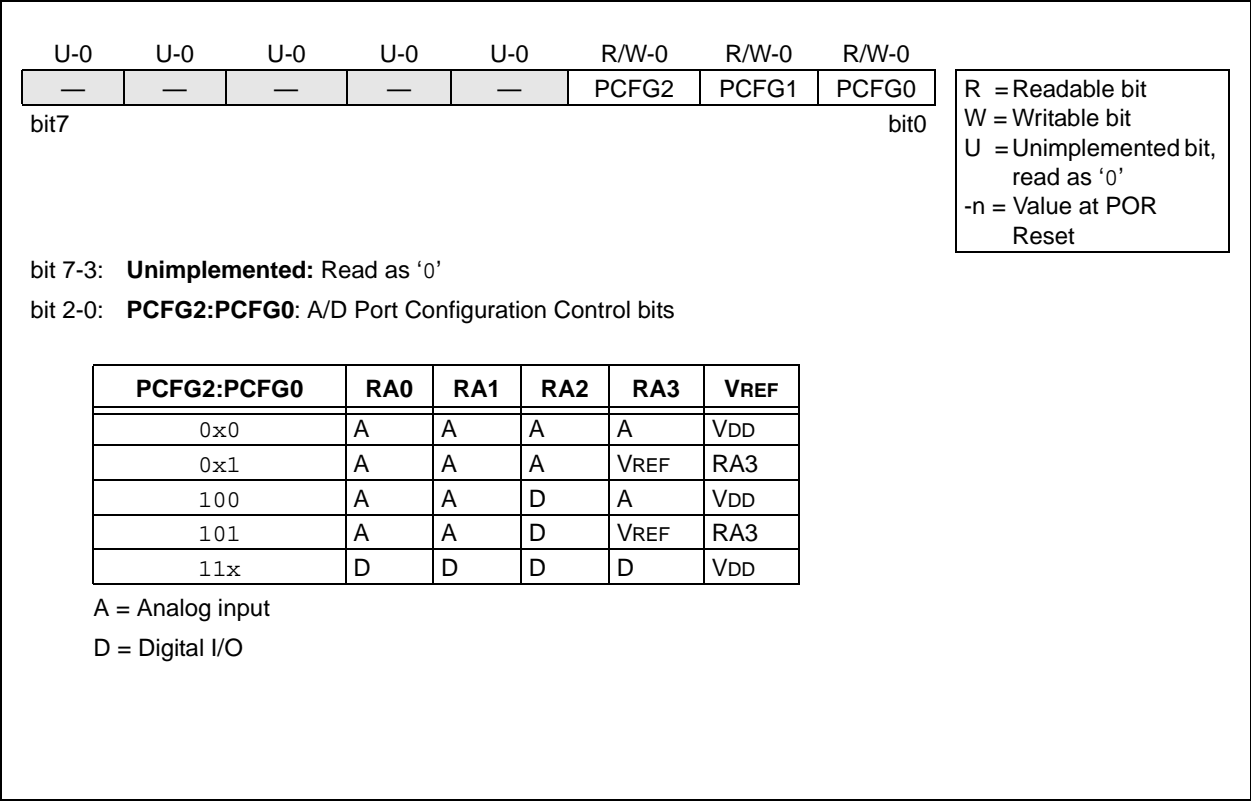
Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

PIC16C712/716

NOTES:

PIC16C712/716

FIGURE 8-2: ADCON1 REGISTER (ADDRESS 9Fh)



PIC16C712/716

FIGURE 9-6: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT

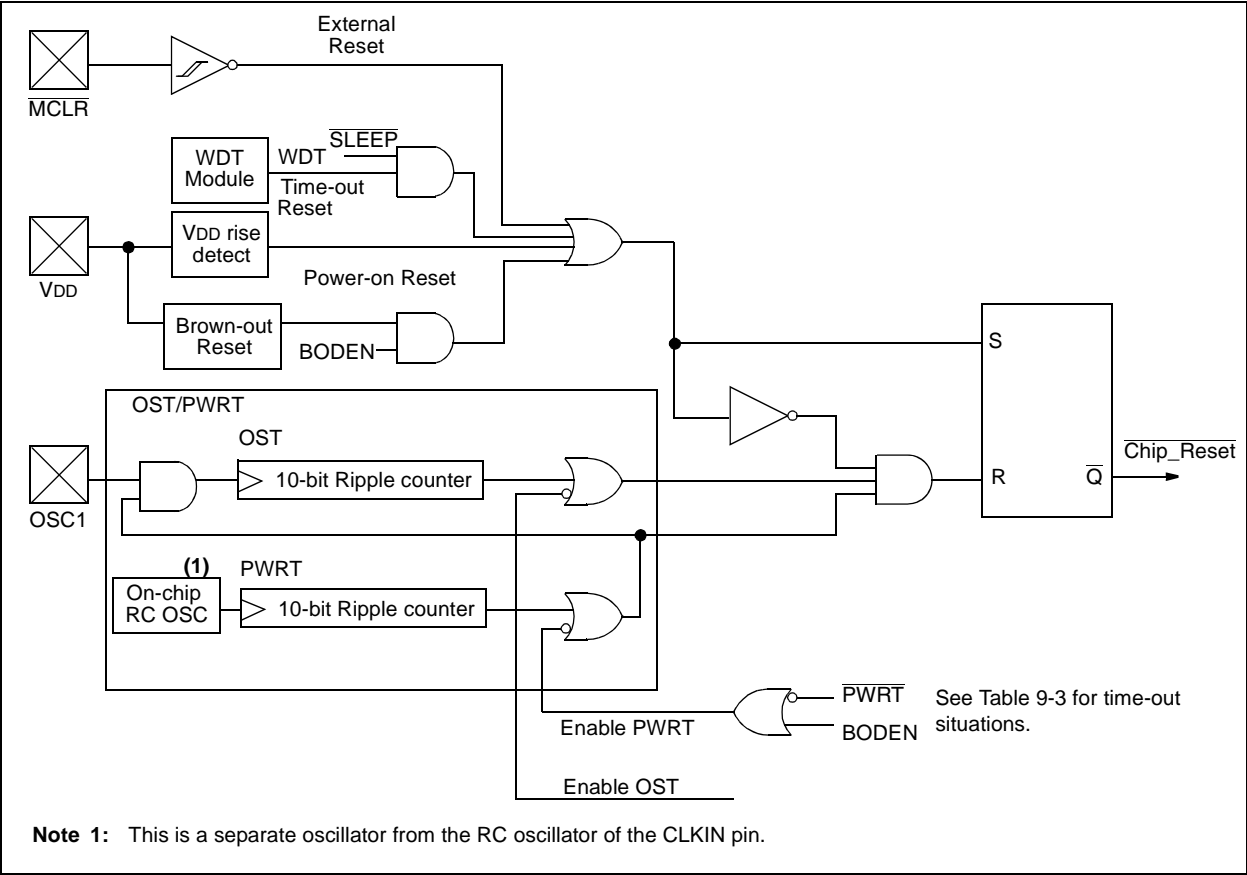
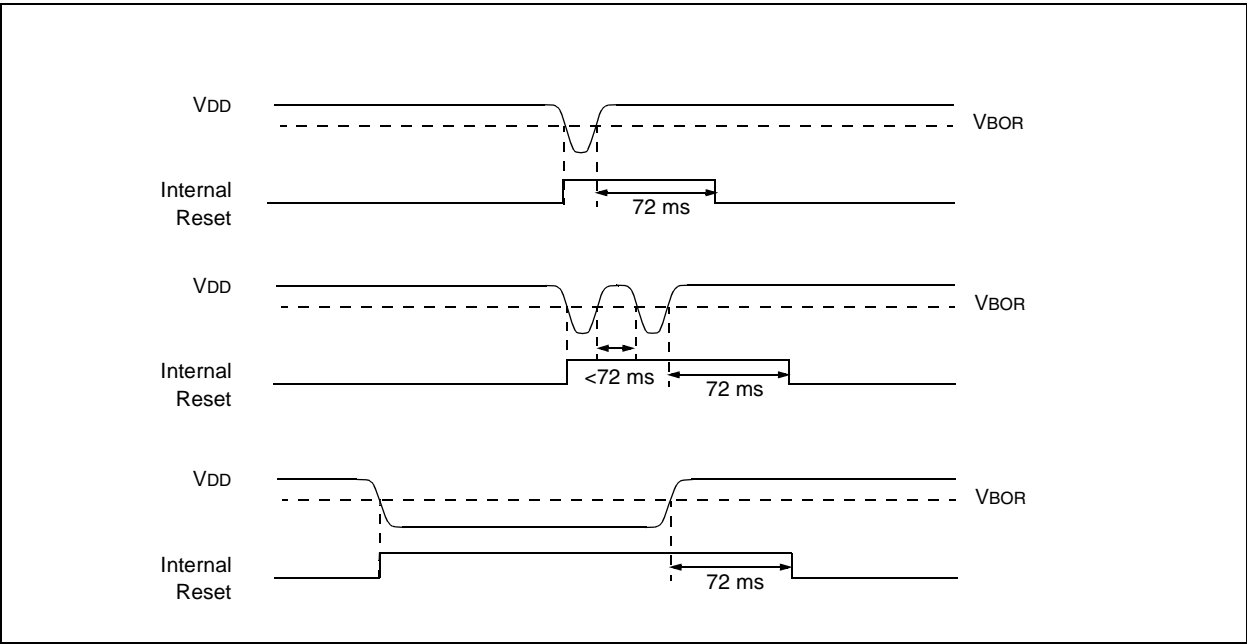


FIGURE 9-7: BROWN-OUT SITUATIONS



PIC16C712/716

FIGURE 9-11: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO V_{DD})

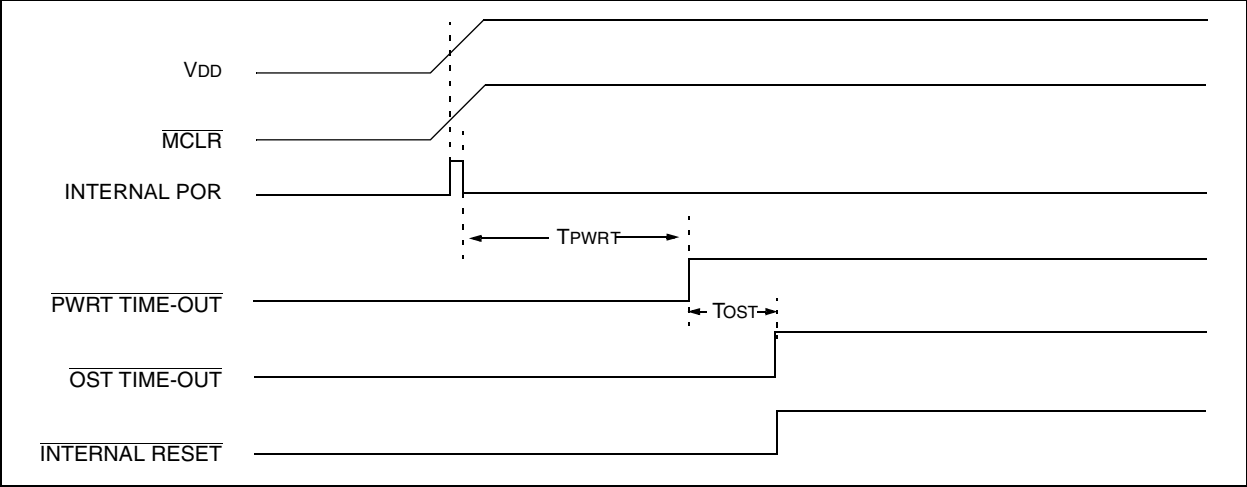


FIGURE 9-12: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 1

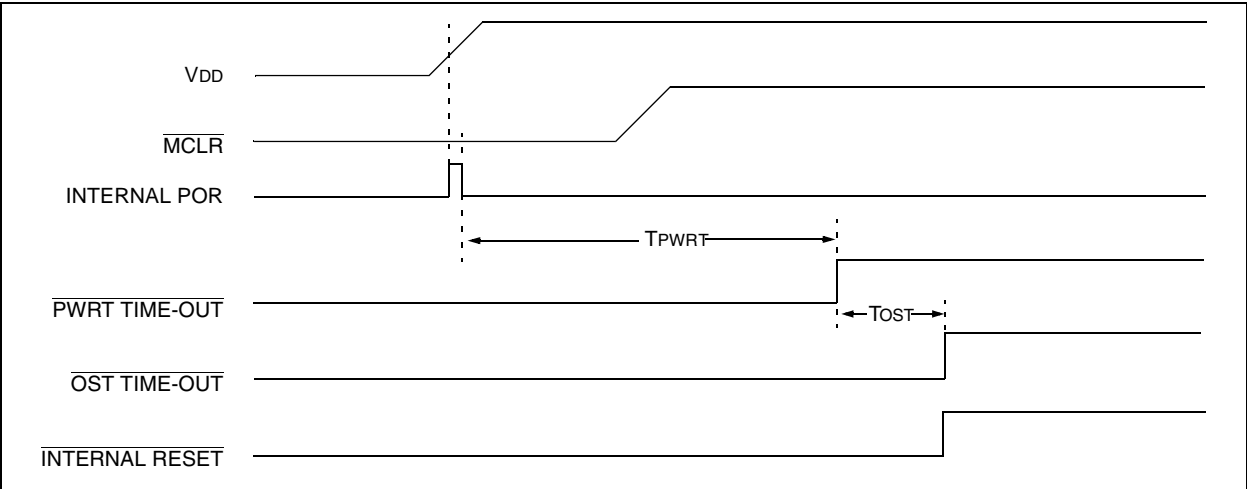
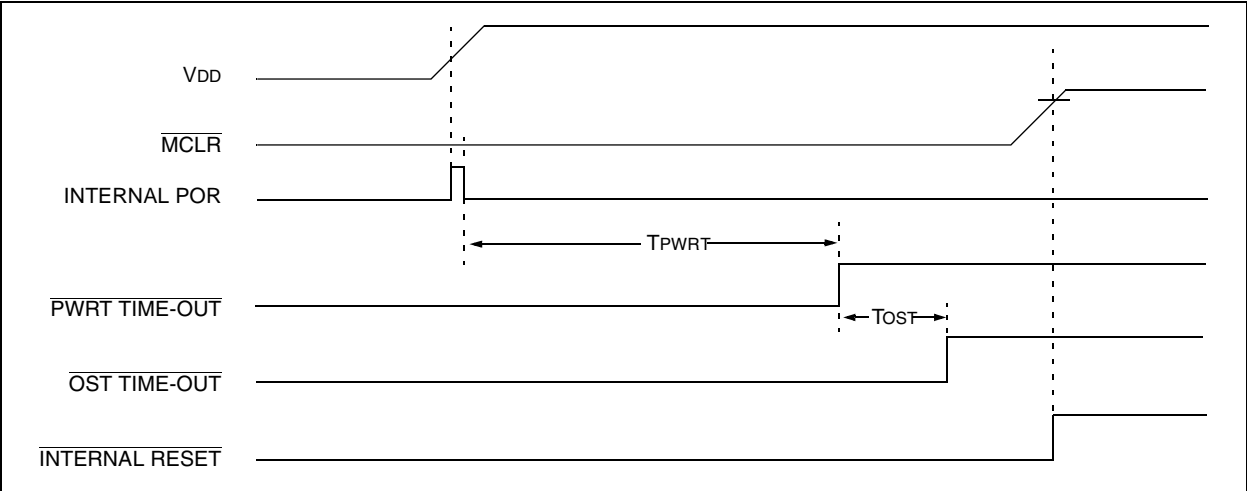


FIGURE 9-13: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 2



9.16 In-Circuit Serial Programming™

PIC16CXXX microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

For complete details on serial programming, please refer to the In-Circuit Serial Programming™ (ICSP™) Guide, (DS30277).

PIC16C712/716

TABLE 10-2: PIC16CXXX INSTRUCTION SET

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	Notes	
			MSb		LSb				
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0000	0011	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDI	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

11.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
 - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
 - MPASM™ Assembler
 - MPLAB C18 and MPLAB C30 C Compilers
 - MPLINK™ Object Linker/
MPLIB™ Object Librarian
 - MPLAB ASM30 Assembler/Linker/Library
- Simulators
 - MPLAB SIM Software Simulator
- Emulators
 - MPLAB ICE 2000 In-Circuit Emulator
 - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
 - MPLAB ICD 2
- Device Programmers
 - PICSTART® Plus Development Programmer
 - MPLAB PM3 Device Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

11.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
 - Simulator
 - Programmer (sold separately)
 - Emulator (sold separately)
 - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Visual device initializer for easy register initialization
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as HI-TECH Software C Compilers and IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PIC MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
 - Source files (assembly or C)
 - Mixed assembly and C
 - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

11.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

11.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 family of microcontrollers and dsPIC30F family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

11.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/librarian features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

11.5 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

11.6 MPLAB SIM Software Simulator

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

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 Assemblers. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

12.4.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 12-4: EXTERNAL CLOCK TIMING

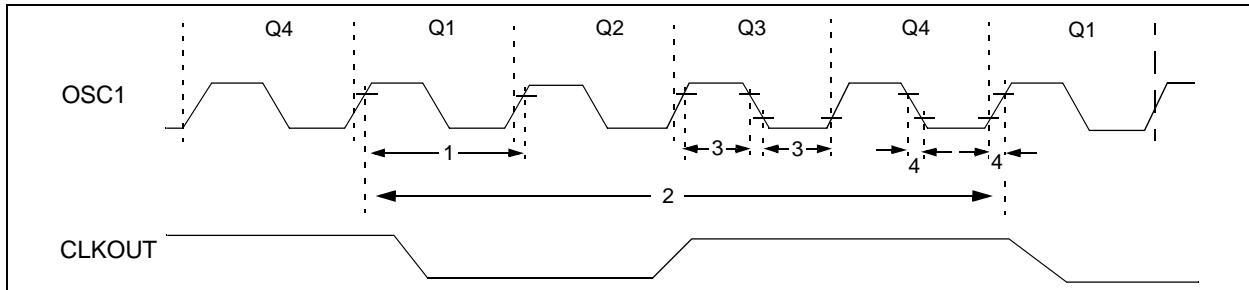


TABLE 12-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
1A	Fosc	External CLKIN Frequency (Note 1)	DC	—	4	MHz	RC and XT osc modes
			DC	—	4	MHz	HS osc mode (-04)
			DC	—	20	MHz	HS osc mode (-20)
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			0.1	—	4	MHz	XT osc mode
			4	—	20	MHz	HS osc mode
			5	—	200	kHz	LP osc mode
1	Tosc	External CLKIN Period (Note 1)	250	—	—	ns	RC and XT osc modes
			250	—	—	ns	HS osc mode (-04)
			50	—	—	ns	HS osc mode (-20)
			5	—	—	μs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			250	—	250	ns	HS osc mode (-04)
			50	—	250	ns	HS osc mode (-20)
			5	—	—	μs	LP osc mode
2	Tcy	Instruction Cycle Time (Note 1)	200	—	DC	ns	Tcy = 4/Fosc
3*	TosL, TosH	External Clock in (OSC1) High or Low Time	100	—	—	ns	XT oscillator
			2.5	—	—	μs	LP oscillator
			15	—	—	ns	HS oscillator
4*	TosR, TosF	External Clock in (OSC1) Rise or Fall Time	—	—	25	ns	XT oscillator
			—	—	50	ns	LP oscillator
			—	—	15	ns	HS oscillator

* 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.

Note1: Instruction cycle period (Tcy) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1/CLKIN pin.

When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

FIGURE 12-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

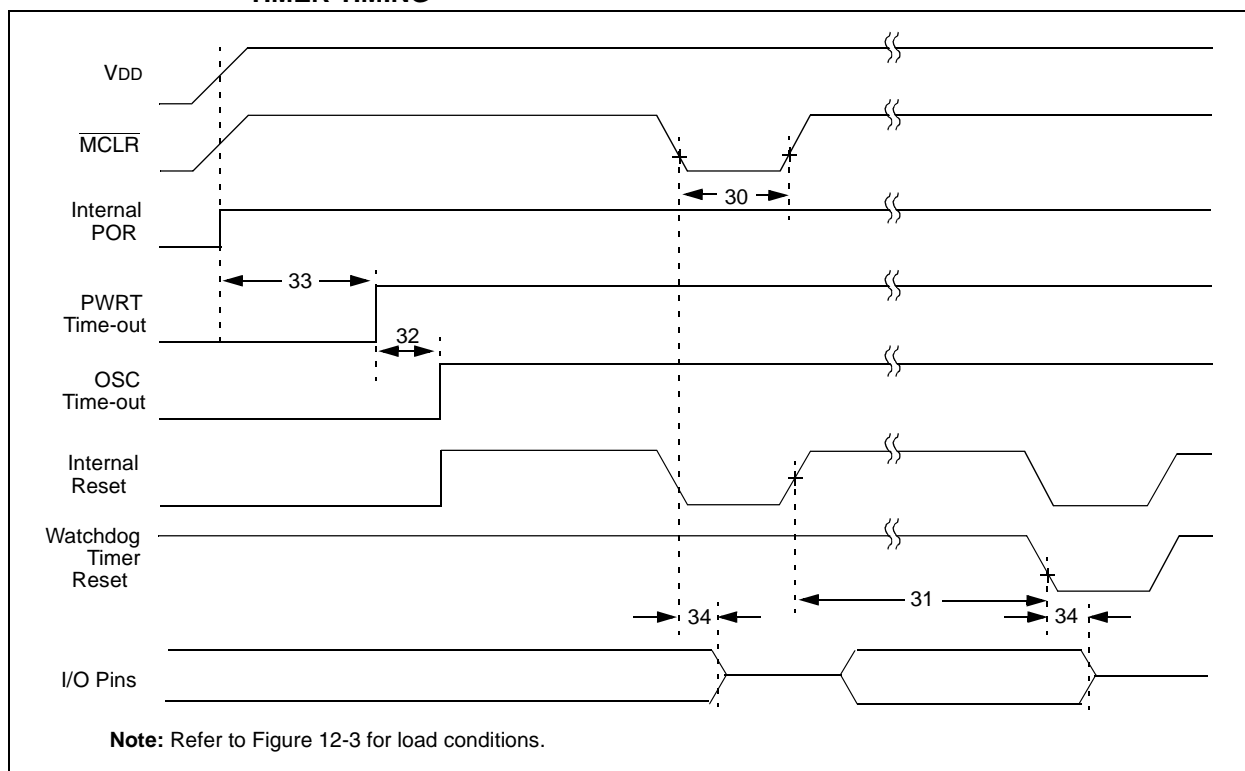


FIGURE 12-7: BROWN-OUT RESET TIMING

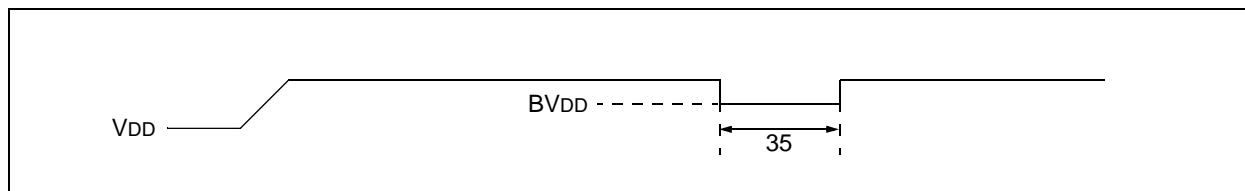


TABLE 12-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS

Parameter No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
30	TmCL	MCLR Pulse Width (low)	2	—	—	μs	VDD = 5V, -40°C to +125°C
31*	TWDT	Watchdog Timer Time-out Period (No Prescaler)	7	18	33	ms	VDD = 5V, -40°C to +125°C
32	TOST	Oscillation Start-up Timer Period	—	1024 TOSC	—	—	TOSC = OSC1 period
33*	TPWRT	Power-up Timer Period	28	72	132	ms	VDD = 5V, -40°C to +125°C
34	TIOZ	I/O High-impedance from MCLR Low or WDT Reset	—	—	2.1	μs	
35	TBOR	Brown-out Reset Pulse Width	100	—	—	μs	VDD ≤ BVDD (D005)

* 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.

PIC16C712/716

FIGURE 12-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS

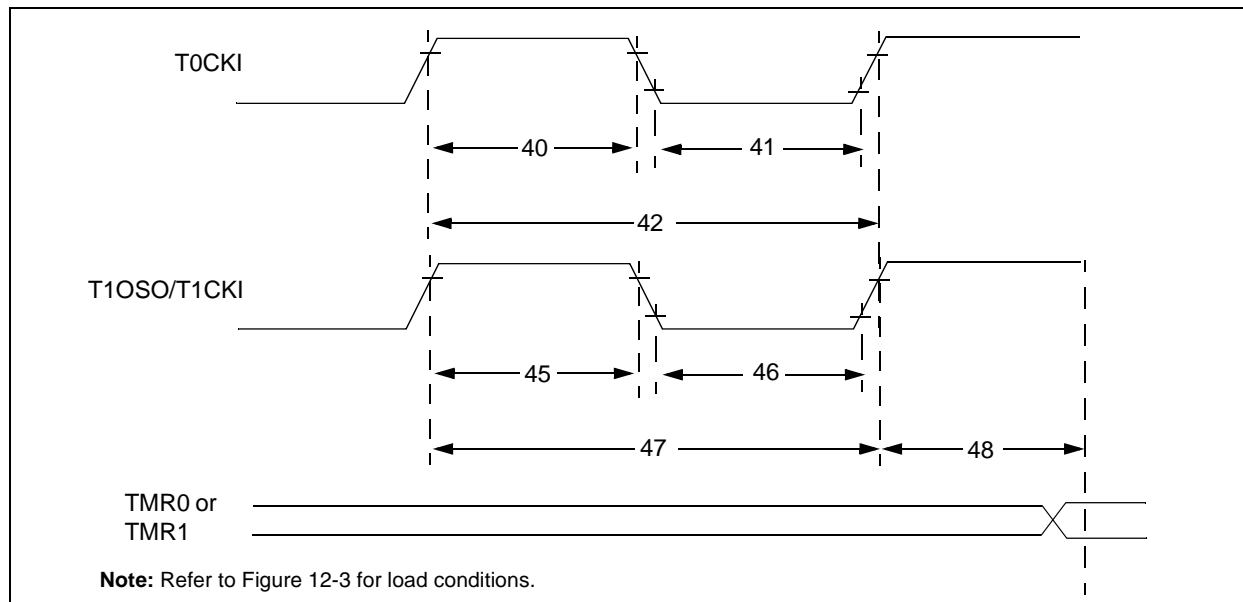


TABLE 12-5: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Sym.	Characteristic		Min.	Typ†	Max.	Units	Conditions	
40*	Tt0H	T0CKI High Pulse Width		No Prescaler 0.5TCY + 20	—	—	ns	Must also meet parameter 42	
		With Prescaler		10	—	—	ns		
41*	Tt0L	T0CKI Low Pulse Width		No Prescaler 0.5TCY + 20	—	—	ns	Must also meet parameter 42	
		With Prescaler		10	—	—	ns		
42*	Tt0P	T0CKI Period		No Prescaler TCY + 40	—	—	ns	N = prescale value (2, 4,..., 256)	
		With Prescaler		Greater of: 20 or $\frac{TCY + 40}{N}$	—	—	ns		
45*	Tt1H	T1CKI High Time	Synchronous, Prescaler = 1		0.5TCY + 20	—	—	ns	Must also meet parameter 47
			Synchronous, Prescaler = 2,4,8	Standard	15	—	—	ns	
				Extended (LC)	25	—	—	ns	
			Asynchronous	Standard	30	—	—	ns	
				Extended (LC)	50	—	—	ns	
46*	Tt1L	T1CKI Low Time	Synchronous, Prescaler = 1		0.5TCY + 20	—	—	ns	Must also meet parameter 47
			Synchronous, Prescaler = 2,4,8	Standard	15	—	—	ns	
				Extended (LC)	25	—	—	ns	
			Asynchronous	Standard	30	—	—	ns	
				Extended (LC)	50	—	—	ns	
47*	Tt1P	T1CKI input period	Synchronous	Standard	Greater of: 30 OR $\frac{TCY + 40}{N}$	—	—	ns	N = prescale value (1, 2, 4, 8)
				Extended (LC)	Greater of: 50 OR $\frac{TCY + 40}{N}$				N = prescale value (1, 2, 4, 8)
			Asynchronous	Standard	60	—	—	ns	
				Extended (LC)	100	—	—	ns	
	Ft1		Timer1 oscillator input frequency range (oscillator enabled by setting bit T1OSCEN)			DC	—	200	kHz
48	TCKEZtmr1	Delay from external clock edge to timer increment			2Tosc	—	7Tosc	—	

* 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.

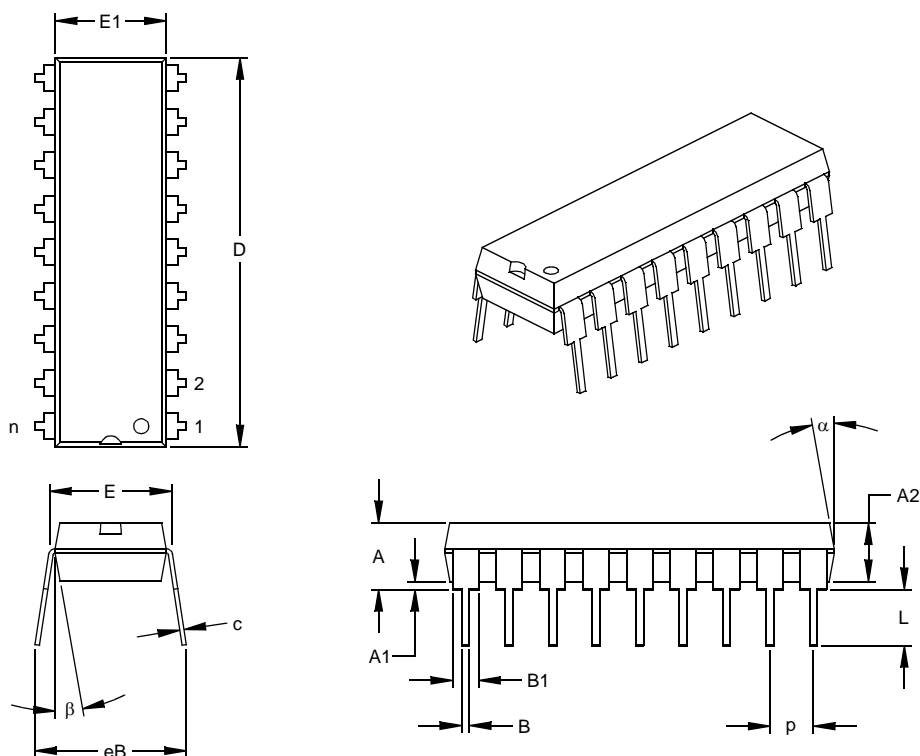
PIC16C712/716

13.2 Package Details

The following sections give the technical details of the packages.

18-Lead Plastic Dual In-line (P) – 300 mil (PDIP)

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



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.890	.898	.905	22.61	22.80	22.99
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter

§ Significant Characteristic

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" (0.254mm) per side.

JEDEC Equivalent: MS-001

Drawing No. C04-007

PIC16C712/716

Interrupt Sources.....	51, 61
A/D Conversion Complete	47
Block Diagram.....	61
Capture Complete (CCP).....	40
Compare Complete (CCP).....	41
Interrupt-on-Change (RB7:RB4)	24
RB0/INT Pin, External.....	62
TMR0 Overflow	30, 62
TMR1 Overflow	31, 34
TMR2 to PR2 Match	37
TMR2 to PR2 Match (PWM)	36, 42
Interrupts, Context Saving During	62
Interrupts, Enable Bits	
A/D Converter Enable (ADIE Bit)	16
CCP1 Enable (CCP1IE Bit).....	16, 40
Global Interrupt Enable (GIE Bit)	15, 61
Interrupt-on-Change (RB7:RB4) Enable (RBIE Bit).....	15, 62
Peripheral Interrupt Enable (PEIE Bit)	15
RB0/INT Enable (INTE Bit)	15
TMR0 Overflow Enable (T0IE Bit).....	15
TMR1 Overflow Enable (TMR1IE Bit)	16
TMR2 to PR2 Match Enable (TMR2IE Bit)	16
Interrupts, Flag Bits	
A/D Converter Flag (ADIF Bit)	17, 47
CCP1 Flag (CCP1IF Bit)	17, 40, 41
Interrupt-on-Change (RB7:RB4) Flag (RBIF Bit)	15, 24, 62
RB0/INT Flag (INTF Bit).....	15
TMR0 Overflow Flag (T0IF Bit)	15, 62
TMR1 Overflow Flag (TMR1IF Bit)	17
TMR2 to PR2 Match Flag (TMR2IF Bit).....	17
M	
Master Clear (MCLR)	
MCLR Reset, Normal Operation	54, 58, 59
MCLR Reset, Sleep	59
MCLR Reset, Sleep	54, 58
Memory Organization	
Data Memory	10
Program Memory	9
Microchip Internet Web Site	101
MPLAB ASM30 Assembler, Linker, Librarian	70
MPLAB ICD 2 In-Circuit Debugger.....	71
MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator	71
MPLAB ICE 4000 High-Performance Universal In-Circuit Emulator	71
MPLAB Integrated Development Environment Software ...	69
MPLAB PM3 Device Programmer.....	71
MPLINK Object Linker/MPLIB Object Librarian	70
O	
OPCODE Field Descriptions	67
OPTION_REG Register	12, 14
INTEDG Bit	14
PS2:PS0 Bits	14, 29
PSA Bit.....	14, 29
RBPB Bit.....	14
T0CS Bit.....	14, 29
T0SE Bit.....	14, 29
Oscillator Configuration.....	51, 53
HS	53, 58
LP.....	53, 58
RC.....	53, 54, 58
Selection (FOSC1:FOSC0 Bits).....	52

XT	53, 58
Oscillator, Timer1	31, 34
Oscillator, WDT.....	63

P

Packaging	89
Details.....	90
Paging, Program Memory.....	9, 19
PCON Register	18, 58
BOR Bit.....	18
POR Bit.....	18
PICSTART Plus Development Programmer	72
PIE1 Register.....	12, 16
ADIE Bit	16
CCP1IE Bit	16
TMR1IE Bit	16
TMR2IE Bit	16
Pin Functions	
MCLR/VPP	6
RA0/AN0	6
RA1/AN1	6
RA2/AN2.....	6
RA3/AN3/VREF.....	6
RA4/T0CKI	6
RB0/INT	7
RB1	7
RB2.....	7
RB3.....	7
RB4.....	7
RB5.....	7
RB6.....	7
RB7	7
VDD	7
Vss	7
Pinout Descriptions	
PIC16C712/716 Pinout Description	6
PIR1 Register	11, 17
ADIF Bit	17
CCP1IF Bit.....	17
TMR1IF Bit.....	17
TMR2IF Bit	17
Pointer, FSR	20
POR. See Power-on Reset	
PORTA	
Initialization.....	21
PORTA Register	11, 21
RA3:RA0 Port Pins	21
RA4/T0CKI Pin	22
TRISA Register.....	12, 21
PORTB	
Block Diagram of RB1/T1OSO/T1CKI Pin.....	24
Block Diagram of RB2/T10SI Pin.....	25
Block Diagram of RB3/CCP1 Pin	25
Initialization.....	23
PORTB Register	11, 23
Pull-up Enable (RBPB Bit).....	14
RB0/INT Edge Select (INTEDG Bit)	14
RB0/INT Pin, External.....	62
RB3:RB0 Port Pins	23
RB7:RB4 Interrupt-on-Change	62
RB7:RB4 Interrupt-on-Change Enable (RBIE Bit)	15, 62
RB7:RB4 Interrupt-on-Change Flag (RBIF Bit)	15, 24, 62
RB7:RB4 Port Pins	26
TRISB Register.....	12, 23

PIC16C712/716

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

TO: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___ Y ___ N

Device: PIC16C712/716

Literature Number: DS41106C

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?
