



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

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	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 4x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f716t-i-so

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is give in Table 2-1.

The Special Function Registers can be classified into two sets; core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in that peripheral feature section.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Page
00h	INDF ⁽¹⁾	Addressing	this location	uses conten	ts of FSR to	address data	memory (no	t a physical r	egister)	0000 0000	18
01h	TMR0	Timer0 mod	ule's register	•						xxxx xxxx	27
02h	PCL ⁽¹⁾	Program Co	ounter's (PC)	Least Signif	icant Byte					0000 0000	17
03h	STATUS ⁽¹⁾	IRP ⁽⁴⁾	RP1 ⁽⁴⁾	RP0	TO	PD	Z	DC	С	0001 1xxx	11
04h	FSR ⁽¹⁾	Indirect Data	a Memory Ac	dress Pointe	er					xxxx xxxx	18
05h	PORTA ^(5,6)	_	_	(7)	RA4	RA3	RA2	RA1	RA0	x 0000	19
06h	PORTB ^(5,6) RB7 RB6 RB5 RB4 RB3 RB2 RB1 R		RB0	xxxx xxxx	21						
07h-09h	_	Unimpleme	nimplemented							—	
0Ah	PCLATH ^(1,2)	—	— — Write Buffer for the upper 5 bits of the Program Counter						0 0000	17	
0Bh	INTCON ⁽¹⁾ GIE PEIE TOIE INTE RBIE TOIF INTE		INTF	RBIF	x000 0000	13					
0Ch	PIR1	_	ADIF — ADIF — CCP1IF TMR2IF TMR1IF						TMR1IF	-0000	15
0Dh	_	Unimpleme	Inimplemented								
0Eh	TMR1L	Holding Reg	gister for the	Least Signifi	cant Byte of	the 16-bit TM	IR1 Register			xxxx xxxx	29
0Fh	TMR1H	Holding Reg	gister for the	Most Signific	ant Byte of t	he 16-bit TM	R1 Register			XXXX XXXX	29
10h	T1CON	—		T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR10N	00 0000	32
11h	TMR2				Timer2 Modu	ile's Register	r			0000 0000	35
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	36
13h-14h	—	Unimpleme	nted							—	
15h	CCPR1L	Capture/Co	mpare/PWM	Register 1 (I	LSB)					XXXX XXXX	48
16h	CCPR1H	Capture/Co	mpare/PWM	Register 1 (I	MSB)					xxxx xxxx	48
17h	CCP1CON	P1M1	P1M1 P1M0 DC1B1 DC1B0 CCP1M3 CCP1M2 CCP1M1 CCP1M0						0000 0000	48	
18h	PWM1CON	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	60
19h	ECCPAS	ECCPASE	ECCPASE ECCPAS2 - ⁽⁸⁾ ECCPAS0 PSSAC1 PSSAC0 PSSBD1 PSSBD0						00-0 0000	57	
1Ah-1Dh	—	Unimpleme	Unimplemented								
1Eh	ADRES	A/D Result	Register							xxxx xxxx	37
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	(7)	ADON	0000 0000	41

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY BANK 0

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

Note 1: These registers can be addressed from either bank.

2: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for PC<12:8> whose contents are transferred to the upper byte of the program counter.

3: Other (non Power-up) Resets include: external Reset through MCLR and the Watchdog Timer Reset.

4: The IRP and RP1 bits are reserved. Always maintain these bits clear.

5: On any device Reset, these pins are configured as inputs.

6: This is the value that will be in the PORT output latch.

7: Reserved bits, do not use.

8: ECCPAS1 bit is not used on PIC16F716.

2.2.2.1 STATUS Register

The STATUS register, shown in Register 2-1, 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 TO and 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 uluu (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:	The PIC16F716 does not use bits IRP
	and RP1 of the STATUS register. Main-
	tain 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.

Reserved	Reserved	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	TO	PD	Z	DC	С
bit 7							bit 0

REGISTER 2-1: STATUS: STATUS REGISTER

r				
Legend:				
R = Readabl	e 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: This bit	is reserved and should	be maintained as '0'	
bit 6	RP1: This bit	is reserved and should	be maintained as '0'	
bit 5	RP0: Registe	er Bank Select bit (used	for direct addressing)	
	1 = Bank 1 (8	30h-FFh)		
	0 = Bank 0 (00h-7Fh)		
bit 4	TO: Time-ou	t bit		
	1 = After pov	ver-up, CLRWDT instructi	on or SLEEP instruction	
	o = A WDT t	me-out occurred		
bit 3	PD: Power-d	own bit		
	1 = After pov	ver-up or by the CLRWDT	' instruction	
	0 = By execu	Ition of the SLEEP instru	ction	
bit 2	Z: Zero bit			
	1 = The resu	It of an arithmetic or logi	c operation is zero	
	0 = The resu	It of an arithmetic or logi	ic operation is not zero	
bit 1	DC: Digit Ca reversed.	rry/Borrow bit (Addwf, A	DDLW, SUBLW, SUBWF instruction	ons), For Borrow, the polarity is
	1 = A carry-c	out from the 4th low-orde	er bit of the result occurred	
	0 = No carry	out from the 4th low-ord	ler bit of the result	
bit 0	C: Carry/Bor	row bit ⁽¹⁾ (ADDWF, ADDLI	N, SUBLW, SUBWF instruction	ns)
	1 = A carry-c	out from the Most Signific	cant bit of the result occurred	
	0 = No carry	out from the Most Signi	ficant bit of the result occurred	
Note 1: Fo	or Borrow, the p	plarity is reversed. A sub	otraction is executed by adding	the two's complement of the sec-
or	nd operand. For	rotate (RRF, RLF) instruc	tions, this bit is loaded with eith	ner the high-order or low-order bit

of the source register.

2.3 PCL and PCLATH

The Program Counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 2-3 shows the two situations for the loading of the PC. The upper example in Figure 2-3 shows how the PC is loaded on a write to PCL (PCLATH<4:0> \rightarrow PCH). The lower example in Figure 2-3 shows how the PC is loaded during a CALL or GOTO instruction (PCLATH<4:3> \rightarrow PCH).

2.3.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<12:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by writing the desired upper 5 bits to the PCLATH register. When the lower 8 bits are written to the PCL register, all 13 bits of the program counter will change to the values contained in the PCLATH register.

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). Care should be exercised when jumping into a look-up table or program branch table (computed GOTO) by modifying the PCL register. Assuming that PCLATH is set to the table start address, if the table length is greater than 255 instructions or if the lower 8 bits of the memory address rolls over from 0xFF to 0x00 in the middle of the table, then PCLATH must be incremented for each address rollover that occurs between the table beginning and the target location within the table.

For more information refer to Application Note AN556, *"Implementing a Table Read"* (DS00556).

2.3.2 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 bit of the address is provided by PCLATH<3>. When doing a CALL or GOTO instruction, the user must ensure that the page select bit is 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<3> bit is not required for the RETURN instructions (which POPs the address from the stack).

FIGURE 2-3: LOADING OF PC IN DIFFERENT SITUATIONS



2.4 Stack

The stack allows a combination of up to 8 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 PUSHed onto the stack when a CALL instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a RETURN, RETLW or a RETFIE instruction execution. PCLATH is not modified when the stack is PUSHed or POPed.

After the stack has been PUSHed 8 times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

FIGURE 3-2: BLOCK DIAGRAM OF RA4/T0CKI PIN



TABLE 3-1:	SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

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
PORTA	_	_	_	RA4	RA3	RA2	RA1	RA0	x 0000	u uuuu
TRISA	_	_	_	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1 1111	1 1111
ADCON1	_	_	_	_		PCFG2	PCFG1	PCFG0	000	000

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

4.1.3 SOFTWARE PROGRAMMABLE PRESCALER

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

There are 8 prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits of the OPTION register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be assigned to the WDT module.

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

When the prescaler is assigned to WDT, a CLRWDT instruction will clear the prescaler along with the WDT.

4.1.3.1 Switching Prescaler Between Timer0 and WDT Modules

As a result of having the prescaler assigned to either Timer0 or the WDT, it is possible to generate an unintended device Reset when switching prescaler values. When changing the prescaler assignment from Timer0 to the WDT module, the instruction sequence shown in Example 4-1, must be executed.

EXAMPLE 4-1: CHANGING PRESCALER (TIMER0 \rightarrow WDT)

Г			
	BANKSEL	TMR0	;
	CLRWDT		;Clear WDT
	CLRF	TMR0	;Clear TMR0 and
			;prescaler
	BANKSEL	OPTION_REG	;
	BSF	OPTION_REG, PSA	;Select WDT
	CLRWDT		;
			;
	MOVLW	b'11111000'	;Mask prescaler
	ANDWF	OPTION_REG,W	;bits
	IORLW	b'00000101'	;Set WDT prescaler
	MOVWF	OPTION_REG	;to 1:32
l		—	

When changing the prescaler assignment from the WDT to the Timer0 module, the following instruction sequence must be executed (see Example 4-2).

EXAMPLE 4-2:	CHANGING PRESCALER
	(WDT \rightarrow TIMER0)

CLRWDT		;Clear WDT and ;prescaler
BANKSEL	OPTION_REG	;
MOVLW	b'11110000'	;Mask TMR0 select and
ANDWF	OPTION_REG,W	;prescaler bits
IORLW	b'00000011'	;Set prescale to 1:16
MOVWF	OPTION_REG	;

4.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The T0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The T0IF bit must be cleared in software. The Timer0 interrupt enable is the T0IE bit of the INTCON register.

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

4.1.5 USING TIMER0 WITH AN EXTERNAL CLOCK

When Timer0 is in Counter mode, the synchronization of the T0CKI input and the Timer0 register is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks. Therefore, the high and low periods of the external clock source must meet the timing requirements as shown in the **Section 12.0 "Electrical Characteristics"**.

TABLE 4-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER0

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
TMR0	Timer0 M	odule Regis		xxxx xxxx	uuuu uuuu					
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
OPTION_REG	RBPU	INTEDG	TOCS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
TRISA		_	_	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	1 1111	1 1111

Legend: -= Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the Timer0 module.

5.8 ECCP Capture/Compare Time Base

The ECCP module uses the TMR1H:TMR1L register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMR1H:TMR1L register pair is copied into the CCPR1H:CCPR1L register pair on a configured event.

In Compare mode, an event is triggered when the value CCPR1H:CCPR1L register pair matches the value in the TMR1H:TMR1L register pair. This event can be a Special Event Trigger.

For more information, see Section 8.0 "Enhanced Capture/Compare/PWM Module".

5.9 ECCP Special Event Trigger

If a ECCP is configured to trigger a special event, the trigger will clear the TMR1H:TMR1L register pair. This special event does not cause a Timer1 interrupt. The ECCP module may still be configured to generate a ECCP interrupt.

In this mode of operation, the CCPR1H:CCPR1L register pair effectively becomes the period register for Timer1.

Timer1 should be synchronized to the FOSC to utilize the Special Event Trigger. Asynchronous operation of Timer1 can cause a Special Event Trigger to be missed.

In the event that a write to TMR1H or TMR1L coincides with a Special Event Trigger from the ECCP, the write will take precedence.

For more information, see Section 8.0 "Enhanced Capture/Compare/PWM Module".



FIGURE 5-2: TIMER1 INCREMENTING EDGE

6.0 TIMER2 MODULE

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

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

See Figure 6-1 for a block diagram of Timer2.

6.1 Timer2 Operation

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

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

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

The match output of the Timer2/PR2 comparator is then fed into the Timer2 postscaler. The postscaler has postscale options of 1:1 to 1:16 inclusive. The output of the Timer2 postscaler is used to set the TMR2IF interrupt flag bit in the PIR2 register.

FIGURE 6-1: TIMER2 BLOCK DIAGRAM

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

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

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

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

Note: TMR2 is not cleared when T2CON is written.



7.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- · Port configuration
- · Channel selection
- · ADC voltage reference selection
- · ADC conversion clock source
- Interrupt control

7.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ADCON1 bits. See the corresponding Port section for more information.

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

7.1.2 CHANNEL SELECTION

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

When changing channels, a delay is required before starting the next conversion. Refer to **Section 7.2 "ADC Operation"** for more information.

7.1.3 ADC VOLTAGE REFERENCE

The PCFG bits of the ADCON0 register provide independent control of the positive voltage reference. The positive voltage reference can be either VDD or an external voltage source.

7.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS bits of the ADCON0 register. There are four possible clock options:

- Fosc/2
- Fosc/8
- Fosc/32
- FRC (dedicated internal oscillator)

The time to complete one bit conversion is defined as TAD. One full 8-bit conversion requires 9.5 TAD periods.

For correct conversion, the appropriate TAD specification must be met. See A/D conversion requirements in **Section 12.0 "Electrical Characteristics"** for more information. Table 7-1 gives examples of appropriate ADC clock selections.

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

TABLE 7-1: TAD vs. DEVICE OPERATING FREQUENCIES

AD Clock S	Source (TAD)	Device Frequency			
Operation	ADCS<1:0>	20 MHz	5 MHz	1.25 MHz	333.33 kHz
2 Tosc	00	100 ns ⁽²⁾	400 ns ⁽²⁾	1.6 μs	6 μs
8 Tosc	01	400 ns ⁽²⁾	1.6 μs	6.4 μs	24 μs ⁽³⁾
32 Tosc	10	1.6 μs	6.4 μs	25.6 μs ⁽³⁾	96 μs ⁽³⁾
RC	11	2-6 μs ^{(1), (4)}	2-6 μs ^{(1), (4)}	2-6 μs ^{(1), (4)}	2-6 μs ⁽¹⁾

Legend: Shaded cells are outside of recommended range.

Note 1: The RC source has a typical TAD time of 4 μ s.

2: These values violate the minimum required TAD time.

3: For faster conversion times, the selection of another clock source is recommended.

4: When device frequency is greater than 1 MHz, the RC A/D conversion clock source is recommended for Sleep operation only.

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
CCPR1L	Capture/Compare/PWM Register 1 (LSB)						xxxx xxxx	xxxx xxxx		
CCPR1H	Capture/Cor	mpare/PWM I	Register 1 (M	SB)					xxxx xxxx	XXXX XXXX
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	x000 0000	x000 0000
PIE1	_	ADIE	_	_	_	CCP1IE	TMR2IE	TMR1IE	-0000	-0000
PIR1	_	ADIF	_	_	_	CCP1IF	TMR2IF	TMR1IF	-0000	-0000
PR2	Timer2 Period Register						1111 1111	1111 1111		
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register						xxxx xxxx	XXXX XXXX		
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register						xxxx xxxx	XXXX XXXX		
TMR2	Timer2 module's register						0000 0000	0000 0000		
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111

TABLE 8-2: REGISTERS ASSOCIATED WITH CAPTURE

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

8.3.8 ENHANCED PWM AUTO-SHUTDOWN MODE

The PWM mode supports an Auto-Shutdown mode that will disable the PWM outputs when an external shutdown event occurs. Auto-Shutdown mode places the PWM output pins into a predetermined state. This mode is used to help prevent the PWM from damaging the application.

The auto-shutdown sources are selected using the ECCPASx bits of the ECCPAS register. A shutdown event may be generated by:

- A logic '0' on the INT pin
- · Setting the ECCPASE bit in firmware

A shutdown condition is indicated by the ECCPASE (Auto-Shutdown Event Status) bit of the ECCPAS register. If the bit is a '0', the PWM pins are operating normally. If the bit is a '1', the PWM outputs are in the shutdown state. Refer to Figure 8-5.

When a shutdown event occurs, two things happen:

The ECCPASE bit is set to '1'. The ECCPASE will remain set until cleared in firmware or an auto-restart occurs (see Section 8.3.9 "Auto-Restart Mode").

The enabled PWM pins are asynchronously placed in their shutdown states. The PWM output pins are grouped into pairs [P1A/P1C] and [P1B/P1D]. The state of each pin pair is determined by the PSSAC and PSSBD bits of the ECCPAS register. Each pin pair may be placed into one of three states:

- Drive logic '1'
- Drive logic '0'
- Tri-state (high-impedance)



FIGURE 8-5: AUTO-SHUTDOWN BLOCK DIAGRAM

9.0 SPECIAL FEATURES OF THE CPU

The PIC16F716 device has a host of features intended to maximize system reliability, minimize cost through elimination of external components, provide power-saving operating modes and offer code protection. These are:

- OSC Selection
- Reset
 - Power-on Reset (POR)
 - Power-up Timer (PWRT)
 - Oscillator Start-up Timer (OST)
 - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Sleep
- Code protection
- ID locations
- In-Circuit Serial Programming[™] (ICSP[™])

The PIC16F716 device has a Watchdog Timer, which can be shut off only through Configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay on power-up only and is designed to keep the part in Reset while the power supply stabilizes. With these two timers on-chip, most applications need no external Reset circuitry.

Sleep mode is designed to offer a very low-current Power-Down mode. The user can wake-up from Sleep through external Reset, Watchdog Timer Wake-up, or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost, while the LP crystal option saves power. A set of Configuration bits are used to select various options.

9.1 Configuration Bits

The Configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special configuration memory space (2000h-3FFFh), which can be accessed only during programming.

9.5 Power-up Timer (PWRT)

The Power-up Timer provides a fixed nominal time-out, on power-up only, from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in Reset as long as the PWRT is active. The PWRT's time delay allows VDD to rise to an acceptable level. The power-up timer enable Configuration bit, PWRTE, is provided to enable/disable the PWRT.

The power-up time delay will vary from chip-to-chip due to VDD, temperature and process variation. See AC parameters for details.

9.6 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This ensures that the crystal oscillator or resonator has started and stabilized. See AC parameters for details.

The OST time-out is invoked only for XT, LP and HS modes and only on Power-on Reset or wake-up from Sleep.

9.7 Programmable Brown-Out Reset (PBOR)

The PIC16F716 has on-chip Brown-out Reset circuitry. A Configuration bit, BOREN, can disable (if clear/programmed) or enable (if set) the Brown-out Reset circuitry.

The BORV Configuration bit selects the programmable Brown-out Reset threshold voltage (VBOR). When BORV is 1, VBOR IS 4.0V. When BORV is 0, VBOR is 2.5V

A Brown-out Reset occurs when VDD falls below VBOR for a time greater than parameter TBOR (see Table 12-4). A Brown-out Reset is not guaranteed to occur if VDD falls below VBOR for less than parameter TBOR.

On any Reset (Power-on, Brown-out, Watchdog, etc.) the chip will remain in Reset until VDD rises above VBOR. The Power-up Timer will be invoked and will keep the chip in Reset an additional 72 ms only if the Power-up Timer enable bit in the Configuration register is set to 0 ($\overline{PWRTE} = 0$).

If the Power-up Timer is enabled and VDD drops below VBOR while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above VBOR, the Power-up Timer will execute a 72 ms Reset. See Figure 9-6.

For operations where the desired brown-out voltage is other than 4.0V or 2.5V, an external brown-out circuit must be used. Figure 9-8, Figure 9-9 and Figure 9-10 show examples of external Brown-out Protection circuits.

ADDLW	Add literal and W		
Syntax:	[<i>label</i>] ADDLW k		
Operands:	$0 \le k \le 255$		
Operation:	$(W) + k \to (W)$		
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.		

BCF	Bit Clear f		
Syntax:	[<i>label</i>]BCF f,b		
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$		
Operation:	0 → (f)		
Status Affected:	None		
Description:	Bit 'b' in register 'f' is cleared.		

ADDWF	Add W and f		
Syntax:	[<i>label</i>] ADDWF f,d		
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$		
Operation:	(W) + (f) \rightarrow (destination)		
Status Affected:	C, DC, Z		
Description:	Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.		

BSF	Bit Set f
Syntax:	[<i>label</i>]BSF f,b
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$
Operation:	$1 \rightarrow (f \le b >)$
Status Affected:	None
Description:	Bit 'b' in register 'f' is set.

ANDLW	AND literal with W		
Syntax:	[<i>label</i>] ANDLW k		
Operands:	$0 \le k \le 255$		
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.		

BTFSC	Bit Test f, Skip if Clear
Syntax:	[label] BTFSC f,b
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ 0 \leq b \leq 7 \end{array}$
Operation:	skip if (f) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '1', the next instruction is executed. If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a two-cycle instruction.

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

SUBWF	Subtract W from f		
Syntax:	[label] SL	JBWF f,d	
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in \ensuremath{\left[0,1\right]} \end{array}$		
Operation:	(f) - (W) \rightarrow (destination)		
Status Affected:	C, DC, Z		
Description:	Subtract (2' W register f '0', the resu register. If '0 stored back	s complement method) rom register 'f'. If 'd' is It is stored in the W d' is '1', the result is in register 'f.	
	C = 0	W > f	
	1		

C = 0	W > f
C = 1	$W \leq f$
DC = 0	W<3:0> > f<3:0>
DC = 1	W<3:0> ≤ f<3:0>

XORLW	Exclusive OR literal with W						
Syntax:	[<i>label</i>] XORLW k						
Operands:	$0 \le k \le 255$						
Operation:	(W) .XOR. $k \rightarrow (W)$						
Status Affected:	Z						
Description:	The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.						

SWAPF	Swap Nibbles in f
Syntax:	[<i>label</i>] SWAPF f,d
Operands:	$\begin{array}{l} 0 \leq f \leq 127 \\ d \in [0,1] \end{array}$
Operation:	$(f<3:0>) \rightarrow (destination<7:4>),$ $(f<7:4>) \rightarrow (destination<3:0>)$
Status Affected:	None
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed in register 'f'.

XORWF	Exclusive OR W with f					
Syntax:	[<i>label</i>] XORWF f,d					
Operands:	$0 \le f \le 127$ $d \in [0,1]$					
Operation:	(W) .XOR. (f) \rightarrow (destination)					
Status Affected:	Z					
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.					

PIC16F716

NOTES:

11.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers. Software control of the MPLAB ICE 2000 In-Circuit Emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The architecture of the MPLAB ICE 2000 In-Circuit Emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE 2000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft[®] Windows[®] 32-bit operating system were chosen to best make these features available in a simple, unified application.

11.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC[®] and MCU devices. It debugs and programs PIC[®] and dsPIC[®] Flash microcontrollers with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The MPLAB REAL ICE probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with the popular MPLAB ICD 2 system (RJ11) or with the new high speed, noise tolerant, lowvoltage differential signal (LVDS) interconnection (CAT5).

MPLAB REAL ICE is field upgradeable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added, such as software breakpoints and assembly code trace. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, real-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

11.9 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PIC MCUs and can be used to develop for these and other PIC MCUs and dsPIC DSCs. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming[™] (ICSP[™]) protocol, offers costeffective, in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single stepping and watching variables, and CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real time. MPLAB ICD 2 also serves as a development programmer for selected PIC devices.

11.10 MPLAB PM3 Device Programmer

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

12.2 DC Characteristics: PIC16F716 (Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$						
Param No.	Sym	Characteristic	Min	Тур†	Max	Units	Vdd	Conditions
	Vdd	Supply Voltage						
D001			2.0	_	5.5	V		
	Idd	Supply Current						
D010			_	14	17	μA	2.0	Fosc = 32 kHz
DUIU				23	28	μA	3.0	LP Oscillator mode
			_	45	63.7	μA	5.0	
			_	120	160	μA	2.0	Fosc = 1 MHz
D011			_	180	250	μA	3.0	XT Oscillator mode
			_	290	370	μA	5.0	
			_	220	300	μA	2.0	Fosc = 4 MHz
D012			_	350	470	μA	3.0	XT Oscillator mode
			_	600	780	μA	5.0	
			_	2.1	2.9	mA	4.5	Fosc = 20 MHz
D013				2.5	3.3	mA	5.0	HS Oscillator mode
	IPD	Power-down Base Current						
0000			—	0.1	0.8	μA	2.0	WDT, BOR and T1OSC:
D020			—	0.1	0.85	μA	3.0	disabled
			—	0.2	2.7	μA	5.0	
		Peripheral Module Current ⁽¹⁾	1					
			—	1	2.0	μA	2.0	WDT Current
D021			—	2	3.5	μA	3.0	
			—	9	13.5	μA	5.0	
			-	37	50	μA	3.0	BOR Current
D022			—	40	55	μA	4.5	
			—	45	60	μA	5.0	
D025			—	1.8	6	μA	2.0	T1osc Current
			—	2.6	7.5	μA	3.0	
			—	3.0	9	μA	5.0	

† 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: The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral "Δ" current can be determined by subtracting the base IDD or IPD current from this limit.

12.5.2 TIMING CONDITIONS

The temperature and voltages specified in Table 12-1 apply to all timing specifications, unless otherwise noted. Figure 12-3 specifies the load conditions for the timing specifications.

TABLE 12-1: TEMPERATURE AND VOLTAGE SPECIFICATIONS - AC

	Standard Operating Conditions (unless otherwise stated)				
	Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for industrial				
	$-40^{\circ}C \le TA \le +125^{\circ}C$ for extended				
AC CHARACTERISTICS	Operating voltage VDD range as described in DC spec Section 12.1 "DC Character-				
	istics: PIC16F716 (Industrial, Extended)" and Section 12.4 "DC Characteristics:				
	PIC16F716 (Industrial, Extended)". LC parts operate for commercial/industrial				
	temp's only.				

FIGURE 12-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS



12.5.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 12-4: EXTERNAL CLOCK TIMING







FIGURE 13-5: TYPICAL IDD vs. VDD OVER Fosc (XT MODE)



14.2 Package Details

The following sections give the technical details of the packages.

18-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

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



	INCHES					
Dimension	Dimension Limits		NOM	MAX		
Number of Pins	Ν	18				
Pitch	е	.100 BSC				
Top to Seating Plane	А	-	.210			
Molded Package Thickness	A2	.115	.130	.195		
Base to Seating Plane	A1	.015	-	-		
Shoulder to Shoulder Width	E	.300	.310	.325		
Molded Package Width	E1	.240	.250	.280		
Overall Length	D	.880	.900	.920		
Tip to Seating Plane	L	.115	.130	.150		
Lead Thickness	С	.008	.010	.014		
Upper Lead Width	b1	.045	.060	.070		
Lower Lead Width	b	.014	.018	.022		
Overall Row Spacing §	eB	_	-	.430		

Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. § Significant Characteristic.
- 3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- 4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-007B