

MANEJO DE LAS INSTRUCCIONES DE UN PIC

INTRODUCCION

Una instrucci n es un comando, parte del programa, que se le aplica al PIC para que lo interprete y ejecute su cumplimiento. La instrucci n se compone de una serie de bits presentada en secciones o posiciones perfectamente delimitadas, que el procesador interpreta para cumplir una orden deseada, quedando establecidas las variables que se modifican.

Dicho de otra forma, las instrucciones poseen un formato de acuerdo a un sistema de codificaci n. El formato cambia de acuerdo con la operaci n que se va a realizar.

Cada instrucci n, para la l nea baja de los PICs tipo 16C5X, consiste en una palabra de 12 bits que codifica en un solo bloque, la orden, el operador y la ubicaci n del resultado o el salto (en caso de que lo hubiere). En los microcontroladores PICs tipo 16X84 cada instrucci n tiene una longitud de 14 bits.

Los bits que act an como datos de la memoria EPROM se reciben en el decodificador de instrucciones, y operan con el contador de programa y el registro de trabajo W, para acceder a lugares espec ficos del microcontrolador, tales como la ALU, posiciones de memoria, registros, etc.

Como sabemos, los PICs 16X84, entre otros, manejan un set reducido de instrucciones (35 instrucciones en lo que se denomina RISC) que presentan una codificaci n muy particular llamada formato de la instrucci n .

Cada instrucci n posee su formato y es totalmente definido por MICROCHIP.

En la figura 1 de la p gina siguiente podemos apreciar la sintaxis de una instrucci n del set. Se trata de la instrucci n que suma el contenido del registro de trabajo con el contenido de un registro cuya direcci n est determinada por el operador f .

En esta figura, en primer lugar se observa la sintaxis , es decir, la forma en que el programador escribir la instrucci n en el lenguaje entendible por el operador (el programa en .asm). La codificaci n es el formato de la palabra que define la instrucci n y que veremos en detalle luego. Note que para que se complete la instrucci n se necesita una sola palabra y un solo ciclo del contador de programas.

Instrucción: SUMA

Suma el contenido del registro de trabajo con el contenido de la posición de memoria definida por "f"

Sintaxis:	ADDWF f,d
Codificación:	0001 11d1 ffff
Palabras:	1
Ciclos:	1
Operación:	(w) + (f) d
Modifica el Status:	C, DC, Z

Figura 1

Como veremos, el resultado de la operación se guarda en un sitio definido por el programador (dependiendo de qué estado tome el bit *d*) y que esta instrucción modifica los bits C, DC y Z del registro de estado (STATUS).

Esto quiere decir que cuando coloco la instrucción:

ADDWF f,d

El ensamblador generará el código:

0001 11df ffff

En este código, los seis bits de mayor peso (bits 6 al 11 en la figura 2) definen la operación que va a realizar la instrucción de acuerdo a lo requerido por el programador. Aquí se define una operación que el PIC interpreta y ejecuta sobre un operando determinado, cuya dirección la busca según lo

indicado por los 5 bits menos significativos (bits 0 al 4 en la figura 2).

Código de la instrucción ADDWf



Figura 2

El bit 5 es un código de reconocimiento de destino y determina en qué lugar se va a alojar el resultado de la operación. Si este bit es 0 el destino de la operación será puesto en el registro de trabajo (W). Si el bit *d* es puesto a 1 el destino será el de la posición de memoria definido por *f*.

Los cinco bits de menor peso en el formato de la instrucción representan la dirección donde está guardado el operando (*f*), que está en la memoria de datos y por tener cinco bits sí lo podemos direccionar una de las 32 posiciones de memoria de datos.

La mayoría de las instrucciones se realizan en un ciclo de contador de programa (ciclo de instrucción) excepto las instrucciones de salto que necesitan dos ciclos para ejecutarla.

Se determina el ciclo de instrucción dividiendo por cuatro la frecuencia del oscilador, elegida para el funcionamiento del microcontrolador tal como se observa en la figura 3.

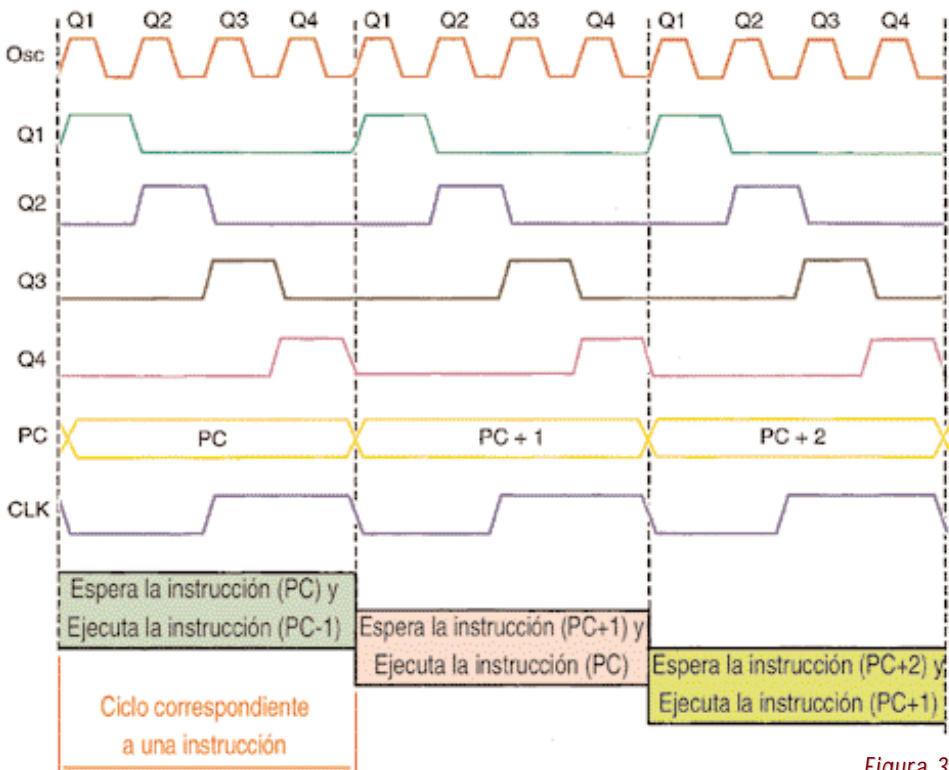
Es decir, la señal que proviene del oscilador externo, conectado a los pines OSC1/CLKIN y OSC2/CLKOUT del microcontrolador, se divide en cua-

Manejo de las Instrucciones de un PIC

tro ciclos, obteniéndose así la señal requerida por el procesador interno para realizar las operaciones. De esta manera se puede realizar la búsqueda y ejecución de la instrucción.

El reloj de instrucción es el ciclo interno que posee el microcontrolador para cronometrar el tiempo de ejecución de las instrucciones.

Los pulsos entrantes del reloj son divididos por 4, generando diferentes señales denominadas Q1, Q2, Q3 y Q4. El estado Q1 hace incrementar el contador de programa, Q2 y Q3, se encargan de la decodificación y ejecución de la instrucción y por último, Q4 es la fase de búsqueda de la instrucción. El código se almacena en el registro de instrucciones.



El Set de Instrucciones del 16X84

Vimos cómo es la estructura de una instrucción, razón por la cual le recomendamos que lea nuevamente el comienzo de este capítulo si no entiende lo que a continuación expresaremos.

Los códigos (denominados mnemónicos) que simbolizan un conjunto de instrucciones, representan la tarea que debe hacer el microcontrolador una vez que las analice en función del operando.

Un mnemónico ayuda a recordar el significado que tiene la instrucción.

Para poder analizar al conjunto de instrucciones que conforman el set RISC, se los suele agrupar teniendo en cuenta el tipo de operación que realizan, así es como se las presente bajo cuatro posibles formas, a saber:

1. **Instrucciones orientadas a registros.**
2. **Instrucciones orientadas a bits.**
3. **Instrucciones con literales.**
4. **Instrucciones de control y especiales.**

Figura 4

Instrucciones Orientadas a Registros

CODIGO MNEMONICO	DESCRIPCION	CODIGO DE OPERACION	MODIFICACION DE LOS FLAGS	NOTAS
ADDWF f,d	(W)+(f) → (destino)	00 0111 0FFF FFFF	C, DC, Z	1,2
ANDWF f,d	(W) AND (f) → (destino)	00 0101 0FFF FFFF	Z	1,2
CLRF f	00 → (f)	00 0001 1FFF FFFF	Z	2
CLRW	00 → (W)	00 0001 0000 0011	Z	1,2
COMF f,d	Complemento de f [(f) → (destino)]	00 1001 0FFF FFFF	Z	1,2
DECF f,d	(f)-1 → destino	00 0011 0FFF FFFF	Z	1,2,3
DECFSZ f,d	(f)-1 → destino y si resultado es 0 salta	00 1011 0FFF FFFF	Ninguno	1,2
INCF f,d	(f)+1 → destino	00 1010 0FFF FFFF	Z	1,2,3
INCSZ f,d	(f)+1 → destino y si resultado es 0 salta	00 1111 0FFF FFFF	Ninguno	1,2
IORWF f,d	(W) OR (f) → destino	00 0100 0FFF FFFF	Z	1,2
MOVF f,d	Mueve f → destino	00 1000 0FFF FFFF	Z	
MOVWF f	(W) → (f)	00 0000 1FFF FFFF	Ninguno	1,2
NOP	No operación	00 0000 0xxx0 0000	Ninguno	1,2
RLF f,d	Rota f a la izquierda a través del carry → destino	00 1101 0FFF FFFF	C	1,2
RRF f,d	Rota f a la derecha a través del carry → destino	00 1100 0FFF FFFF	C	1,2
SUBWF f,d	(f)-(W) → (destino)	00 0010 0FFF FFFF	C,DC,Z	1,2
SWAPF f,d	Intercambia los niveles de f → destino	00 1110 0FFF FFFF	Ninguno	
XORWF f,d	(W) XOR (f) → (destino)	00 0110 0FFF FFFF	Z	

Manejo de las Instrucciones de un PIC

Figura 5

Instrucciones orientadas a Bits

CODIGO MNEMONICO	DESCRIPCION	CODIGO DE OPERACION	MODIFICACION DE LAS FLAGS	NOTAS
BCF f,b	Pone a 0 el bit b del registro f	01 00b 8FFF FFFF	Ninguna	1,2
BSF f,b	Pone a 1 el bit b del registro f	01 01b 8FFF FFFF	Ninguna	1,2
BTFSC f,b	Skip si el bit b del reg. f es 0	01 10b 8FFF FFFF	Ninguna	3
BTFSS f,b	Skip si el bit b del reg. f es 1	01 11b 8FFF FFFF	Ninguna	3

En la figura 4 vemos un cuadro donde se reproducen las instrucciones orientadas a registros, la figura 5 muestra las operaciones orientadas a bits y la figura 6 nos permite conocer cuáles son las instrucciones con literales y de control.

Figura 6

Instrucciones con literales y de control

CODIGO MNEMONICO	DESCRIPCION	CODIGO DE OPERACION	MODIFICACION DE LAS FLAGS	NOTAS
ADDLW K	(W)+ K → (W)	11 111X KKKK KKKK	C,D,C,Z	
ANDLW K	(W) AND K → (W)	11 1001 KKKK KKKK	Z	
CALL K	Llamada a subrutina	10 0KKK KKKK KKKK		
CLRWDT	Clear del temporizador del WD	00 0000 0110 0100	Ninguna	
GOTO K	Go To dirección	10 1KKK KKKK KKKK	Z	
IORLW K	(W) OR K → (W)	11 1000 KKKK KKKK	Ninguna	
MOVLW K	K → (W)	11 000X KKKK KKKK	Ninguna	
RETFIE	Retorno de una interrupción	00 0000 0000 1001	Ninguna	
RETLW K	Retorno con un literal en W	11 01XX KKKK KKKK	Ninguna	
RETURN	Retorno de una subrutina	00 0000 0000 1000	C,D,C,Z	
SLEEP	Modo Standby	00 0000 0110 0011	Z	
SUBLW K	K - (W) → W	11 110X KKKK KKKK		
XORLW K	(W) XOR K → (W)	11 1010 KKKK KKKK		

Sobre las notas a las que hacemos referencia en dichas figuras, debe tener en cuenta lo siguiente:

Nota 1. Al modificar un registro de entrada/salida (E/S) con una opera-

ción sobre el mismo (por ejemplo MOVF PORTB,1), el valor utilizado es el que se encuentre presente en las patas del PORTB. Por ejemplo, si el bit tiene un "1" para una pata configurada como entrada y se pone a nivel bajo desde el exterior, el dato se volverá a escribir como un "0".

Nota 2. Si se ejecuta esta instrucción sobre el TMR0 y d=1, se borrará el divisor de frecuencia (preescaler), si está asignado al TMR0.

Nota 3. Si se modifica el Contador de Programa (PC ← CP) o es verdadera una condición de prueba, la instrucción requiere dos ciclos de máquina. El segundo ciclo se ejecuta como un NOP.

Veamos a continuación, cómo deben emplearse cada una de estas instrucciones:

ADDLW	ADD Literal to W			ADDLW
Suma del contenido de W con un número específico				
Operación	$(W) + k \rightarrow (W)$			
Sintaxis	[Etiqueta] ADDLW k			
Operadores	$0 \leq k < 255$			
Código de Operación	0001 11df ffff			
Palabras	1			
Ciclos	1			
Código de Operación	11	111x	kddd	kddd
Descripción	Suma el contenido del registro W al literal k, y almacena el resultado en W. Si se produce acarreo el flag C se pone a "1". Esta instrucción no existe en el PIC 16C5X			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	X	X

C Se pone a 1 si se produce un acarreo desde el bit de mayor peso.

DC Se pone a 1 si se genera un acarreo del bit 3 al bit 4.

Z Se pone a 1 si el resultado de la operación es cero.

EJEMPLO:

Queremos sumar el contenido del registro de trabajo con el número 10 expresado en hexadecimal, la instrucción será:

ADDLW H«10»

La notación H«10» significa que el número está expresado en hexadecimal. Si la notación fuese D«10» significa que el número está expresado en decimal y si fuese B«10» entonces el número está en notación binaria.

Volviendo a nuestro ejemplo:

Si antes de la instrucción tenía en cada registro:

W = 21h (21h significa que el número está expresado en hexadecimal, que es lo mismo que 0010 0001 b —expresado en binario—)

Al ejecutarse la instrucción, se suman ambos números y el resultado se guarda en el registro W, es decir:

W = 21 h + 10 h = 31 h (en hexadecimal),

W = 0001 0000 b + 0001 0000 b = 0010 0000 b (en binario)

ADDWF	ADD W to F			ADDWF
Suma entre el contenido de W y el contenido de un registro f				
Operación	(W) + (f) ----> (destino)			
Sintaxis	[Etiqueta] ADDWF f,d			
Operadores	0 < f < 127 d [0,1]			
Palabras	1			
Ciclos	1			
Código de Operación	00	0111	dfff	fff
Descripción	Suma el contenido del registro W al contenido del registro f, y almacena el resultado en W si d = 0, o sino lo almacena en el registro f si d = 1. Si se produce un acarreo el flag C se pone a "1"			

Registro de STATUS

PA0	PA1	PA0	TC#	PD#	Z	DC	C
-	-	-	-	-	X	X	X

C Se pone a 1 si se produce un acarreo desde el bit de mayor peso.

DC Se pone a 1 si se genera un acarreo del bit 3 al bit 4.

Z Se pone a 1 si el resultado de la operaci n es cero.

EJEMPLO:

Supongamos querer sumar el contenido del registro de trabajo con el contenido de un registro llamado REG, y queremos guardar el resultado en el registro de trabajo, la notaci n ser :

ADDWF REG,0

Si antes de la instrucc i n W = 10 h y REG = 21 h, el resultado de la suma ser 31 h que se guardar en el registro de trabajo, es decir, luego de la suma el contenido de los registros ser :

W = 31 h REG = 21 h

Note que el contenido del registro REG no se modific luego de la operaci n. Si la instrucc i n hubiese sido:

ADDWF REG,1

Entonces el resultado se hubiese guardado en REG, por lo tanto, luego de la operaci n el contenido de los registros ser a:

W = 10 h REG = 31 h

ANDLW	AND Literal and W			ANDLW
Operaci3n AND entre un n3mero y el contenido de W				
Operaci3n	(W) AND. (k) ----> (W)			
Sintaxis	[Etiqueta] ANDLW k			
Operadores	0 < f < 255			
Palabras	1			
Ciclos	1			
C3digo de Operaci3n	11	1001	kkkk	kkkk
Descripci3n	Efect3a la operaci3n AND l3gico entre el contenido del registro W y el literal k, y almacena el resultado en W.			

Manejo de las Instrucciones de un PIC

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	-	-

Z Se pone a 1 si el resultado de la operación es cero.

EJEMPLO:

Supongamos querer hacer la función lógica AND entre el número 0001 0010 b que está almacenado en el registro de trabajo y el número 0011 1001 b. La instrucción será:

ANDLW B«00111001»

Luego, antes de la instrucción tenemos:

W = 0001 0010 b

Se realiza la operación:

(0001 0010) AND (0011 1001) = 0001 0000

Luego de la instrucción, en W tendremos:

W = 0001 0000 b

ANDWF	AND W src F		ANDWF	
Operación lógica AND entre el contenido de un registro f y el contenido del registro de trabajo (W)				
Operación	(W) AND (f) → (destino)			
Sintaxis	[Etiqueta] ANDWF f,d			
Operadores	0 < f < 127 d [0,1]			
Palabras	1			
Ciclos	1			
Código de Operación	00	0101	dfff	ffff
Descripción	Efectúa la operación AND lógica entre el contenido del registro W y el contenido del registro f, y almacena el resultado en W si d = 0, y en f si d = 1.			

Registro de STATUS

PAZ	PAI	PAO	TMF	PMF	Z	DC	C
-	-	-	-	-	X	-	-

Z Se pone a 1 si el resultado de la operación es cero

EJEMPLO:

Queremos realizar la operación AND entre el número 0001 1110 b que está almacenado en el registro ST y el número 1001 0110 b que está almacenado en el registro de trabajo, la instrucción será:

ANDWF ST,1

Antes de ejecutarse la instrucción tenemos:

W = 1001 0110 b

ST = 0001 1110 b

Se realizará la operación:

(1001 0110) AND (0001 1110) = 0001 0110

Es decir que en los registros, luego de ejecutada la instrucción tendremos:

W = 1001 0110 b (no ha cambiado)

ST = 0001 0110 (se ha almacenado el resultado).

BCF	Bit Clear F			BCF
Borra un bit de un registro determinado				
Operación	0 --> (f)			
Sintaxis	[Etiqueta] BCF f,b			
Operadores	0 < f < 127 0 < b < 7			
Palabras	1			
Ciclos	1			
Código de Operación	01	00bb	bfff	ffff
Descripción	Pone a cero el bit número b del registro f			

Manejo de las Instrucciones de un PIC

Registro de STATUS

PA0	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	-

EJEMPLO:

Queremos poner a 0 el bit N... 5 del registro DESC, luego la instrucción ser :

BCF DESC, 5

Si antes de la instrucción el registro DESC tenía:

DESC = 1111 0111 b

Al ejecutarse la instrucción, el registro queda con el valor:

DESC = 1110 0111 b

Nota: Recuerde que el bit menos significativo es el cero y el más significativo es el 7, es decir, los ocho bits de un registro se numeran de 0 a 7.

BSF	Bit Set F			BSF
Pone un "1" en un bit de un registro				
Operación	1 --> (f)			
Sintaxis	[Etiqueta] BSF f,b			
Operadores	0 < f < 127 0 < b < 7			
Palabras	1			
Ciclos	1			
Código de Operación	01	11bb	bfff	ffff
Descripción	Pone a 1 el bit b del registro f			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	-

EJEMPLO:

Queremos poner a 1 el bit N... 0 del registro DESC, luego la instrucción ser:

BSF DESC, 0

Si antes de la instrucción el registro DESC tenía:

DESC = 1111 0111 b

Al ejecutarse la instrucción, el registro queda con el valor:

DESC = 1111 0110 b

BTFSC	Bit Test, Skip if Clear			BTFSC
Que el programa salte una instrucción si el bit b de un registro es "0"				
Operación	skip if (f) = 0			
Sintaxis	[Etiqueta] BTFSC f,b			
Operadores	0 < f < 127 0 < b < 7			
Palabras	1			
Ciclos	1 (2)			
Código de operación	01	10bb	bfff	ffff
Descripción	Si el bit número b del registro f es cero, la instrucción que sigue a ésta se ignora y se trata como un NOP (skip). En este caso, y solo en este caso, la instrucción BTFSC precisa dos ciclos para ejecutarse.			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	-

EJEMPLO:

RUTINA	BTFSC	REG,2
	GOTO	DEC
	GOTO	SIM

Bien, aquí hemos introducido algunas variables a saber:

RUTINA es una etiqueta, de modo tal que cuando alguna instrucción haga que el contador de programa apunte a dicha etiqueta, entonces se ejecutará la instrucción BTFSC.

REG es un registro que yo deba haber postulado con anterioridad en el programa.

DEC y *SIM* son etiquetas, de modo que cuando el contador de programa esté apuntando a la línea:

GOTO DEC

Entonces el programa se seguirá ejecutando desde la línea donde está la etiqueta DEC.

Si antes de la instrucción el contador de programa (PC - CP) está en la línea donde se encuentra la etiqueta RUTINA, entonces se ejecutará la instrucción BTFSC. Al ejecutarse, si el BIT 2 del registro REG es igual a cero, el contador de programa saltará una línea e irá a ejecutar la instrucción:

GOTO SIM

Por último, note que en esta instrucción tenemos tres columnas y es la forma en que, en general, vamos a escribir un programa en lenguaje de usuario (editar un programa). En la primera columna se colocan las etiquetas, en la segunda columna se escribe siempre la instrucción y en la tercera columna se escribe el operando de la instrucción. Normalmente va una cuarta columna precedida del signo ; (punto y coma) que se utiliza para escribir observaciones que no serán tenidas en cuenta por el compilador o el ensamblador que convertirá al programa que estaba en lenguaje de usuario en un programa en lenguaje de máquina (para los PICs es un lenguaje hexadecimal o .hex).

La misma instrucción de nuestro ejemplo la podremos escribir de la siguiente manera:

RUTINA	BTFSC	REG,2	;comienzo de una rutina
---------------	--------------	--------------	--------------------------------

La cuarta columna no ser tenida en cuenta dado que est precedida del signo ;

BTFSS		Bit Test Skip if Set		BTFSS	
Que el programa salte una instrucción si el bit b de un registro es " 1"					
Operación	skip if (<i>f</i> -< <i>b</i> >) = 1				
Sintaxis	[Etiqueta] BTFSS <i>f</i> , <i>b</i>				
Operadores	0 < <i>f</i> < 127 0 < <i>b</i> < 7				
Palabras	1				
Ciclos	1 (2)				
Código de Operación	01	11 <i>bb</i>	<i>b</i> <i>fff</i>	<i>ffff</i>	
Descripción	Si el bit número <i>b</i> del registro <i>f</i> está a 1, la instrucción que sigue a ésta se ignora y se trata como un NOP (skip). En este caso, y sólo en este caso, la instrucción BTFSS precisa dos ciclos para ejecutarse.				

Registro de STATUS

PAC	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	-

EJEMPLO:

RUTINA	BTFSS	REG,0
	GOTO	DEC
	GOTO	SIM

Igual que en el caso anterior hemos introducido algunas variables a saber:

RUTINA es una etiqueta, de modo tal que cuando alguna instrucción haga que el contador de programa apunte a dicha etiqueta, entonces se ejecutará la instrucción BTFSS.

REG es un registro que yo deba haber postulado con anterioridad en el programa.

Manejo de las Instrucciones de un PIC

DEC y *SIM* son etiquetas, de modo que cuando el contador de programa est apuntando a la l nea:

GOTO DEC

Entonces el programa se seguir ejecutando desde la l nea donde est la etiqueta DEC.

Si antes de la instrucc i n el contador de programa PC est en la l nea donde se encuentra la etiqueta RUTINA, entonces se ejecutar la instrucc i n BTFSS. Al ejecutarse, si el BIT 0 del registro REG es igual a uno, el contador de programa saltar una l nea e ir a ejecutar la instrucc i n:

GOTO SIM

CALL	Subrutina Call			CALL
Llamada a Subrutina				
Operador:	0 = k = 2047			
Sintaxis:	[Etiqueta] CALL k			
Operaci3n:	(PC)+1 \rightarrow Top of Stack k \rightarrow PC <10:0>; PCLATCH(<4:3>) \rightarrow PC <12,11>			
Palabra:	1			
Ciclos:	2			
C3digo de Operaci3n:	10	0kkk	kkkk	kkkk
Descripci3n:	Concerna la direcci3n de retorno al programa principal en la Pila y despu3s llama a la subrutina situada en la direcci3n cargada en el PC. El modo de c3dculo de la direcci3n efectiva difiere segun la familia de PICs utilizada. Tambi3n hay que posicionar PA2, PA1 y PA0 (PIC 16C53) o el registro PCLATCH (En los dem3s PIC) antes de ejecutarse la instrucc i n.			

Registro de STATUS

PA2	PA1	PA0	TOV	PD#	Z	DC	C
-	-	-	-	-	-	-	-

EJEMPLO:

DIV CALL RUTIN

En la línea DIV se encuentra la instrucción CALL. Cuando el PC apunta a esta línea, automáticamente va a ejecutar un programa que se encuentra en la línea que posee la etiqueta RUTIN. Cuando termina de ejecutarse la subrutina, se vuelve al programa principal a la línea siguiente a DIV (se dice DIV+1).

CLRF	Clear f			CLRF
Borra el contenido de un registro				
Operación	00h --> f 1 ---> Z			
Sintaxis	[Etiqueta] CLRF f			
Operadores	0 < f < 127			
Palabras	1			
Ciclos	1			
Código de Operación	00	0001	1fff	ffff
Descripción	Se borra el contenido del registro f y el flag Z se activa			

Registro de STATUS

PA0	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	1	-	-

Z Se pone a 1 si el resultado de la operación es cero.

EJEMPLO:

CLRF REG

Si antes de la instrucción:

REG = 5A h

Al ejecutarse:

REG = 00 h

flag Z = 1

CLRW	Clear W			CLRW
Borra el contenido del registro de trabajo (W)				
Operación	00h -->(W) 1 ---> Z			
Sintaxis	[Etiqueta] CLRW			
Operadores	No tiene			
Palabras	1			
Ciclos	1			
Código de Operación	00	0001	0000	0011
Descripción	El registro de trabajo W se carga con 00h y el flag Z se pone a 1			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	1	-	-

Z Se pone a 1 si el resultado de la operación es cero.

EJEMPLO:

CLRW

Si antes de la instrucción:

W= 0011 1111 b

Al ejecutarse:

W = 00

flag Z = 1

En definitiva, es una instrucción que se utiliza cuando debemos asegurarnos que el contenido del registro de trabajo es igual a 0 .

CLRWDT	Clear Watchdog Timer			CLRWDT
Se borra el contenido del perro guardián				
Operación	00h --> WDT 1 --> T0# 1 --> PD#			
Sintaxis	[Etiqueta] CLRWDT			
Operadores	No tiene			
Palabras	1			
Ciclos	1			
Código de Operación	00	0000	0110	0100
Descripción	Se borra todo el registro WDT (Watchdog) como su preescaler. Los bits T0# y PD# del registro de estado se ponen a "1".			

Registro de STATUS

PA0	PA1	PA0	T0#	PD#	Z	DC	C
-	-	-	1	1	-	-	-

Cuando se ejecutan las instrucciones CLRWDT o SLEEP, T0# se pone a 1. Se pone a 0 si el temporizador Watchdog se desborda.

PD# se pone a 1 cuando se ejecuta la instrucción CLRWDT o SLEEP.

EJEMPLO:

CLRWDT

Si antes de ejecutarse la instrucción:

WDT = 0000 0100 b

Al ejecutarse:

WDT = 0000 0000 b

Preescaler WDT = 0

bit de estado T0 = 1

bit de estado PD = 1

Manejo de las Instrucciones de un PIC

COMF	Complement f			COMF
Realiza el complemento del contenido de un registro				
Operación	(f#) -----> (dest)			
Sintaxis	[Etiqueta] COMF f,d			
Operadores	0 < f < 127 d [0,1]			
Palabras	1			
Ciclos	1			
Código de Operación	00	1001	dfff	ffff
Descripción	Hace el complemento del contenido del registro f bit a bit. El resultado se almacena en el registro f si d=1 y en el registro W si d=0, en este caso f no vaia.			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	-	-

Z Se pone a 1 si el resultado de la operaci n es cero.

EJEMPLO:

COMF DIS,0

Realiza el complemento del n mero almacenado en el registro DIS y guarda el resultado en el registro W. Si antes de la instrucc i n:

DIS = 0011 1110 b
W = 0000 0000 b

Al ejecutarse:

DIS = 0011 1110 b
W = 1100 0001 b
flag Z = 0

DECF	Decrement f			DECF
Se decrementa en una unidad el contenido de un registro				
Operación	(f)-1 --> (dest)			
Sintaxis	[Etiqueta] DECF f,d			
Operadores	0 < f < 127 d [0,1]			
Palabras	1			
Ciclos	1			
Código de Operación	00	0011	dfff	ffff
Descripción	Se decrementa el contenido del registro f en una unidad. El resultado se almacena en f si d=1 y en W si d=0 , en este caso f no vaía.			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	-	-

Z Se pone a 1 si el resultado de la operaci n es cero.

EJEMPLO:

DECF SINT,1

Cuando el PC llega a esta l nea, se decrementa el contenido del registro SINT y en dicho registro se guarda el resultado. El registro de trabajo no interviene para nada. Si antes de la instruccin:

SINT = 0011 0111 b
Z = 0

Al ejecutarse:

SINT = 0011 0110 b
Z = 0

DECFSZ	Decrement f, Skip if 0			DECFSZ
Decrementa el contenido de un registro y cuando éste es igual a cero, el contador de programa salta una línea o instrucción				
Operación	(f) - 1 --> (dest) : skip if result =0			
Sintaxis	[Etiqueta] DECFSZ f,d			
Operadores	0 < f < 127 d [0..1]			
Palabras	1			
Ciclos	1 (C)			
Código de Operación	00	1011	dfff	ffff
Descripción	Decrementa el contenido del registro f en una unidad, el resultado se almacena en f si d=1 y en W si d=0, en este caso, f no vale. Si el resultado es cero, se ignora la siguiente instrucción y, en ese caso la instrucción tiene una duración de dos ciclos.			

Registro de STATUS

PA0	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	-

EJEMPLO:

```

TEMP          DECFSZ    DEL,1
              GOTO      TEMP
              CLRF      REG
    
```

Con estas líneas de programa estamos realizando una temporización, consumiendo ciclos de reloj. Cuando el PC se ubica en la línea TEMP, se decrementa el contenido del registro DEL. Supongamos que el registro DEL estaba cargado con el número 50 expresado en notación decimal, al ejecutarse la instrucción, ahora el contenido de dicho registro será 49 (distinto de cero) por lo tanto el PC se ubicará en la siguiente línea (GOTO TEMP). Cuando esto ocurre, la instrucción está enviando al PC nuevamente a la etiqueta TEMP, con lo cual se volverá a decrementar el contenido del registro DEL que ahora tendrá cargado el número 48. Este ciclo se repite constantemente hasta que en el registro DEL se encuentre el número

00 en cuyo caso el CP saltar una línea y ejecutar la siguiente instrucción que en este caso es CLR F REG.

GOTO	Unconditional Branch			GOTO
Salto incondicional a la dirección especificada por el operando				
Operación	$K \rightarrow PC <10:0>$ $(PCLATH <4:3>) \rightarrow (PC <12:11>)$			
Sintaxis	[Etiqueta] GOTO k			
Operadores	0 < k < 2047			
Palabras	1			
Ciclos	2			
Código de Operación	10	kkkkk	kkkk	kkkk
Descripción	Salto incondicional, normalmente se utiliza para dirigirse a una dirección determinada.			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	-

EJEMPLO:

GOTO TEMP

Al ejecutarse la instrucción, el contador de programa se dirige a la dirección donde se encuentra la etiqueta TEMP.

INCF	Increment f	INCF
Incrementa en una unidad el contenido de un registro		
Operación	$(f) + 1 \rightarrow (dest)$	
Sintaxis	[Etiqueta] INCF f,d	

INCF	Increment f			INCF
Incrementa en una unidad el contenido de un registro				
Operadores	$0 < f < 127$ $d [0,1]$ $(f) + 1 \rightarrow (dest)$			
Palabras	1			
Ciclos	1			
Código de Operación	00	1010	dfff	ffff
Descripción	Se incrementa en una unidad el contenido del registro f , si d=1 el resultado se almacena en f , si d=0 el resultado se almacena en W , en este caso el resultado de f no vaia.			

Registro de STATUS

PA0	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	-	-

Z Se pone a 1 si el resultado de la operaci n es cero al haber desbordamiento.

EJEMPLO:

INCF REG,1

Cuando el PC llega a esta instrucc i n se incrementa el contenido del registro REG y el resultado se guarda en el mismo registro, en este caso el registro de trabajo no interviene para nada. Si antes de la instrucc i n:

REG = FF h
 flag Z = 0

Al ejecutarse:

REG = 00 (FF h +1 h = 00 h)
 flag Z = 1

INCFSZ	Increment f, SKIP if 0			INCFSZ
Incrementa en una unidad el contenido de un registro y cuando éste es 00 entonces el PC salta una instrucción				
Operación	(f) + 1 --> (dest) , skip if result = 0			
Sintaxis	[Etiqueta] <INCFSZ f,d			
Operadores	0 < f < 127 d [0,1]			
Palabras	1			
Ciclos	1 (C)			
Código de Operación	00	1111	dfff	ffff
Descripción	Incrementa el contenido del registro f en una unidad, el resultado se almacena de nuevo en f si d=1, y en W si d=0, en este caso, f no valía. Si el resultado es cero, se ignora la siguiente instrucción y, en ese caso la instrucción tiene una duración de dos ciclos.			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	-

EJEMPLO:

```

ABA          INCFSZ      DEN,1
             GOTO        ABA
             GOTO        SIGA
    
```

Para entender como es que si incrementamos un número en algún momento este puede llegar a cero, hay que decir que cuando se llega a la capacidad máxima de un registro, el próximo incremento hará que este se coloque en cero (en todos sus bits) produciéndose un desborde.

Si el registro DEN tenía el número:

DEN = 0111 0001 b

Al ejecutarse la instrucción INCFSZ:

Manejo de las Instrucciones de un PIC

DEN = 0111 0010 b (se incrementa en una unidad)

Entonces el PC va a la dirección GOTO ABA lo que hace que se vuelva a incrementar nuevamente el contenido de DEN. Este ciclo se repite hasta que el contenido de DEN sea cero en cuyo caso el PC salta una instrucción y ejecuta la línea GOTO SIGA (SIGA es el nombre que le hemos dado a otra variable o registro).

IORLW	Inclusive OR Literal with W			IORLW
Realiza la operación lógica OR entre el contenido del registro de trabajo y un número determinado, bit a bit				
Operación	(W) OR k ---> (W)			
Sintaxis	[Etiqueta] IORLW k			
Operadores	0 < k < 255			
Palabras	1			
Ciclos	1			
Código de Operación	11	1000	kkkk	kkkk
Descripción	Se realiza la operación lógica OR entre el registro W y el literal k . El resultado se almacena en el registro W .			

Registro de STATUS

PA0	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	-	-

Z Se pone a 1 si el resultado de la operación es cero

EJEMPLO:

IORLW 0x32

Esto quiere decir que se realiza la operación OR entre el contenido del registro de trabajo y el número 32 expresado en hexadecimal, luego:

32 h = 0011 0010 b

Si en W est almacenado el n mero:

W = 0001 1110 b, luego de la instruccin, en el registro de trabajo se almacenar el resultado es decir:

W = 0010 1100 b

IORWF				IORWF
Inclusive OR W with f				
Realiza la operaci3n l3gica OR entre el contenido del registro de trabajo y el contenido de un registro determinado, bit a bit				
Operaci3n	(W) .OR.(f)--> (dest)			
Sintaxis	[Etiqueta] IORWF f,d			
Operadores	0 < f < 127 d [0,1]			
Palabras	1			
Ciclos	1			
C3digo de Operaci3n	00	0100	dfff	ffff
Descripci3n	Efectu3a la operaci3n l3gica OR entre el contenido del registro W y el contenido del registro f , y almacena el resultado en f si d=1 y en W si d=0 .			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	-	-

Z Se pone a 1 si el resultado de la operaci3n es cero.

EJEMPLO:

IORWF RES,1

Si antes de la instruccin:

RES = 13 h = 0001 0011 b

W = 91 h = 1001 0001 b

Manejo de las Instrucciones de un PIC

Al ejecutarse la instrucción:

RES = 0001 0011 b OR 1001 0001 b = 1001 0011 b = 93 h

El contenido del registro de trabajo no se altera.

MOVLW	Move literal to W			MOVLW
Carga al registro de trabajo con un número especificado en el operando				
Operación	k --> (W)			
Sintaxis	[Etiqueta] MOVLW k			
Operadores	0 < f < 255			
Palabras	1			
Ciclos	1			
Código de Operación	11	00xx	kkkk	kkkk
Descripción	El registro W se carga con el valor de 8 bits del literal k			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	-	-

EJEMPLO:

MOVLW 0x50

Al ejecutarse la instrucción, en el registro de trabajo tendremos:

W = 50 h

La instrucción MOVLW suele emplearse en la construcción de las denominadas tablas que se construyen mediante la comparación de números. En ese caso se carga directamente en el registro de trabajo el número de referencia que se comparará con cantidades almacenadas en memoria (datos) mediante una instrucción posterior.

MOVF	Move f			MOVF
“Mueve el contenido de un registro”				
Operación	(f) --> (dest)			
Sintaxis	[Etiqueta] MOVF f,d			
Operadores	0 < f < 127 d [0,1]			
Palabras	1			
Ciclos	1			
Código de Operación	00	10000	dfff	ffff
Descripción	El contenido del registro f se carga en el registro destino dependiendo del valor de d . Si d=0 el destino es el registro W . Si d=1 el destino es el propio registro f . Esta instrucción permite verificar dicho registro ya que el flag Z queda afectado.			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	-	-

Z Se pone a 1 si el resultado de la operación es cero.

EJEMPLO:

MOVF SER,0

Si el contenido del registro SER es:

SER = 12 h

Luego de ejecutarse la instrucción se tendrá :

W = 12 h

SER = 12 h

Note que en este caso si el operando hubiese sido SER,1, entonces el propio contenido del registro se hubiese cargado en el mismo registro, lo que se emplea para corroborar que realmente esa posición de memoria tiene el dato esperado.

MOVWF	Move W to f			MOVWF
Carga a un registro con el contenido de W				
Operación	(W)--> (f)			
Sintaxis	[Etiqueta] MOVWF f			
Operadores	0 < f < 127			
Palabras	1			
Ciclos	1			
Código de Operación	00	0000	1fff	ffff
Descripción	Mueve el contenido del registro W al registro f			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	-

EJEMPLO:

MOVWF DIR

En este caso se carga al registro DIR con el contenido del registro de trabajo (W). Si antes de la instrucción:

W = 4C h

Al ejecutarse:

DIR = 4C h

W = 4C h

MOVWF es una de las instrucciones más empleadas para la construcción de programas dado que suele ser el medio empleado para cargar datos en posiciones de memoria RAM. Para que tenga una idea, es muy común encontrar la siguiente secuencia:

MOVLW d 22
MOVWF DIR

Aquí hemos cargado al registro DIR con el número 22 expresado en decimal.

NOP	No operación				NOP
Sin Operación					
Operación	no operación				
Sintaxis	[Etiqueta] NOP				
Operadores	No tiene				
Palabras	1				
Ciclos	1				
Código de Operación	00	0000	0000	0000	
Descripción	No realiza operación alguna. En realidad, se consume un ciclo de instrucción sin hacer nada.				

Registro de STATUS

PAZ	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	-

EJEMPLO:

NOP

RETFIE	Return from Interrupt				RETFIE
Retorno de Interrupción					
Operación	TOS --> CP 1 --> GIE				
Sintaxis	[Etiqueta] RETFIE				
Operadores	No tiene				
Palabras	1				
Ciclos	2				
Código de Operación	00	0000	0000	1001	
Descripción	Carga el CP con el valor que se encuentra en la parte alta de la Pila, asegurando así la vuelta de la interrupción. Pone a 1 el bit GIE , con el fin de autorizar de nuevo que se tengan en cuenta las interrupciones.				

EJEMPLO:

RETFIE

Al ejecutarse esta instrucción el contador de programa se dirige a la dirección cargada en la posición más alta de la pila y el bit GIE se pone a 1 con el objeto de habilitar nuevamente la posibilidad de interrupción.

RETLW	Return with Literal in W			RETLW
Retorno de subrutina con un literal cargado en W				
Operación	k --> (W); TOS ---> PC			
Sintaxis	[Etiqueta] RETLW k			
Operadores	0 < K < 255			
Palabras	1			
Ciclos	2			
Código de Operación	11	01xx	kkkk	kkkk
Descripción	Carga el registro W con el literal k, y después carga el PC con el valor que se encuentra en la parte superior de la PILA, efectuando así un retorno de subrutina.			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	-

EJEMPLO:

RETLW k

Como es lógico, esta instrucción está ligada con las subrutinas, dado que nos indica que ésta ha terminado y el retorno inmediato con la carga en el registro de trabajo W de un número que está expresado en la siguiente columna de la instrucción. Además carga al CP con el valor que previamente almacenó en la PILA (STACK).

Esta instrucción tarda dos ciclos de instrucción.

Siempre W debe regresar con un literal (muchas veces ese literal es cero). Muchas veces es til esta instruccin, especialmente cuando deseamos regresar al programa principal con alg n valor obtenido de la subrutina.

RETURN	Return from Subroutine			RETURN
Retorno de Subrutina				
Operaci3n	TOS ----> PC			
Sintaxis	[Etiqueta] RETURN			
Operadores	No tiene			
Palabras	1			
Ciclos	2			
C3digo de Operaci3n	00	0000	0000	1000
Descripci3n	Carga el PC con el valor que se encuentra en la parte superior de la PILA, efectuando as3 un retorno de subrutina			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	-

Esta instruccin no amerita ejemplo alguno, dado que cuando sta se ejecuta, se vuelve al programa principal a una direcci3n que est almacenada en el STACK.

RLF			RLF
Rotate Left f through Carry			
Operaci3n			
Sintaxis	[Etiqueta] RLF f,d		
Operadores	$0 < f < 127$ d [0,1]		

RLF				RLF
Rotate Left f through Carry				
Palabras	1			
Ciclos	1			
Código de Operación	00	1101	dfff	ffff
Descripción	Rotación de un bit a la izquierda del contenido del registro f pasando por el bit de acarreo C . Si d=1 el resultado se almacena en f , si d=0 el resultado se almacena en W .			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	X

EJEMPLO:

RLF REG,0

Si antes de la instrucció n:

REG = 1110 0110 b
flag C = 0

Como d = 0 el resultado se guardar en W.
Al ejecutarse la instrucció n:

REG = 1110 0110 b
W = 1100 1100 b
flag C = 1

RRF				RRF
Rotate Right f through Carry				
Operación				
Sintaxis	[Etiqueta] RRF f,d			
Operadores	0 < f < 127 d [0,1]			

RRF				RRF
Rotate Right f through Carry				
Palabras	1			
Ciclos	1			
Código de Operación	00	1100	dfff	ffff
Descripción	Rotación de un bit a la derecha del contenido del registro f pasando por el bit de acarreo C . Si d=1 el resultado se almacena en f si d=0 el resultado se almacena en W			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	X

EJEMPLO:

RRF REG,0

Si antes de la instrucción:

REG1 = 1110 0110 b
flag C = 1

Como d = 0 el resultado se guarda en W.
Al ejecutarse la instrucción:

REG = 1110 0110 b
W = 0111 0011 b
flag C = 0

SLEEP	Sleep	SLEEP
Modo bajo Consumo		
Operación	00h → WDT 0 → WDT prescaler 1 → TO# 0 → PD#	

SLEEP	Sleep			SLEEP
Modo bajo Consumo				
Sintaxis	[Etiqueta] SLEEP			
Operadores	No tiene			
Palabras	1			
Ciclos	1			
Código de Operación	00	0000	0110	0011
Descripción	<p>Pone al circuito en modo Sleep (bajo consumo) con parada del oscilador. Pone a 0 el flag PD# (Power Down) y el flag TO# (Timer Out) se pone a 1. Se puede salir de este estado por:</p> <ol style="list-style-type: none"> 1. Activación de MCLR para provocar un Reset 2. Desbordamiento del Watchdog si quedó operativo en el modo reposo 3. Generación de una interrupción que no sea TMR0 ya que ésta se desactiva con la instrucción SLEEP. 			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z#	DC	C
-	-	-	1	0	-	-	-

TO Se pone a 1 al ejecutar la instrucción SLEEP o CLRWDT.

PD Se pone a 0 al ejecutar la instrucción SLEEP.

Es una instrucción muy usada en programas almacenados en los microprocesadores empleados en controles remoto. Normalmente el circuito está en modo SLEEP mientras no se detecte la acción de una tecla, pero cuando no hay ninguna orden, vuelve a este estado.

SUBLW	Subtract W from Literal	SUBLW
Resta a un número expresado en el operando el contenido del registro W		
Operación	$k - (W) \rightarrow (W)$	
Sintaxis	[Etiqueta] SUBLW k	
Operadores	$0 < k < 255$	

SUBLW	Subtract W from Literal			SUBLW
Resta a un número expresado en el operando el contenido del registro W				
Palabras	1			
Ciclos	1			
Código de Operación	11	110x	kkkk	kkkk
Descripción	Resta en complemento a dos del contenido del literal k el contenido del registro W , y almacena el resultado en W .			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	X	X

Z Se pone a 1 si el resultado de la operación es cero.

DC Se pone a 1 si se genera un acarreo del bit 3 al grupo de 4 bits superior.

C Se pone a 1 si se genera un acarreo del bit de mayor peso.

EJEMPLO:

SUBLW 0x 05

a) Si antes de la instrucción:

W = 01 h

Al ejecutarse la instrucción:

W = (05 — 01) = 04

flag C = 1 ; el resultado es positivo

b) Si antes de la instrucción: W = 05 h

Al ejecutarse la instrucción: W = (05 — 05) = 00 h

flag C = 1 ; el resultado es cero

flag Z = 1

Manejo de las Instrucciones de un PIC

c) Si antes de la instrucción. $W = 07\text{ h}$

Al ejecutarse: $W = (05 - 07) = FE\text{ h}$

flag $C = 0$; el resultado es negativo.

SUBWF	Subtract W from f			SUBWF
Resta al contenido de un registro el contenido del registro W				
Operación	$(f) - (W) \rightarrow (\text{dest})$			
Sintaxis	[Etiqueta] SUBW f,d			
Operadores	$0 < f < 127$ d [0,1]			
Palabras	1			
Ciclos	1			
Código de Operación	00	0010	dfff	ffff
Descripción	Resta en complemento a dos el contenido del registro f menos el contenido del registro W almacena el resultado en W si $d=0$ y en f si $d=1$.			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	X	X

Z Se pone a 1 si el resultado de la operación es cero.

DC Se pone a 1 si se genera un acarreo del bit 3 al grupo de 4 bits superior.

C Se pone a 1 si se genera un acarreo del bit de mayor peso.

Tenga presente que el complemento a 2 de un registro (sustraendo), equivale a restar dicho registro de 256 ($255+1$). Si el resultado se lo debemos sumar a otro registro (minuendo), si el minuendo (que lo vamos a sumar), es mayor o igual que el sustraendo (que se lo restamos a 256), el resultado será mayor que 255 y por ende se producirá un desborde u overflow con lo que $C="1"$. Esto quiere decir que si $C=1$ entonces el resultado de la resta es positivo.

Para el acarreo decimal, el bit DC se coloca en "1" cuando el valor del nibble menos significativo (desde el bit 0 al 3) del registro W, sea menor que el nibble menos significativo del registro REG (registro fuente).

EJEMPLO:

SUBWF REG,1

a) Si antes de la instrucción:

REG = 04 h

W = 02 h

Al ejecutarse la instrucción:

REG = (04 — 02) = 02h

W = 02 h

flag C = 1 ; el resultado es positivo

b) Si antes de la instrucción:

REG = 04 h

W = 04 h

Al ejecutarse la instrucción:

REG = (04 — 04) = 00h

W = 04 h

flag C = 1 ; el resultado es cero

flag Z = 1 ; el resultado es cero

c) Si antes de la instrucción:

REG = 0100 0110 b (70 en decimal)

W = 1110 0000 b (224 en decimal)

Aquí ocurre un caso especial, veamos por qué. La operación es la siguiente:

¥ Se realiza el complemento a 2 del sustraendo (complemento a 2 de W)

¥ Se suma el contenido de REG con dicho complemento a 2 de W.

¥ El resultado de esta adición, será el resultado de la resta.

Manejo de las Instrucciones de un PIC

Registro w	1110 0000
Complemento a 1	0001 1111
Sumando 1	+ 1
Resultado	0010 0000 (complemento a 2 del contenido de W = 32 en decimal)

Ahora debemos sumar 32 más 70 que es el contenido de REG, el resultado ser :

$$32 + 70 = 102$$

∓ El resultado no excede de 255, por lo que no hay bit de acarreo y en consecuencia "C" es igual a 0 (lo que indica resultado negativo).

Cuando el minuendo sea menor que el sustraendo, el número que debemos restar de 256, es mayor que el que habrá que sumar (REG), por lo tanto no habrá desbordamiento y el bit Carry C será siempre igual a "0".

El acarreo decimal DC se ha colocado en 1, ya que el valor del pack menos significativo (del bit 0 al 3), del registro de trabajo W (es 0000) y es menor que el del registro REG (registro fuente, es 0110). Por último el bit Z es 0, ya que el resultado de la operación no es cero.

Como consecuencia de un resultado negativo, el registro Reg permanece inalterable y en el registro W estará el resultado de la operación:

W = 0010 0110 b (decimal 102)
 REG = 0100 0110 b (decimal 70)
 Flag C = 1
 Flag DC = 0
 Flag Z = 0

SWAPF	Swap Nibbles in f	SWAPF
Intercambia los "packs", es decir los cuatro bits más significativos pasan a ser los menos significativos y viceversa.		
Operación	(f<3:0>) ---> (dest <7:4>) (f<7:4>) ---> (dest <3:0>)	
Sintaxis	[Etiqueta] S WAPF f,d	

SWAPF	Swap Nibbles in f			SWAPF
Intercambia los “packs”, es decir los cuatro bits más significativos pasan a ser los menos significativos y viceversa.				
Operadores	$0 < f < 127$ $d [0,1]$			
Palabras	1			
Ciclos	1			
Código de Operación	00	1110	dfff	ffff
Descripción	Los cuatro bits de más peso del registro f se intercambian con los 4 bits de menos peso del mismo registro. Si d=0 el resultado se almacena en W , si d=1 el resultado se almacena en f .			

Registro de STATUS

PA2	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	-	-	-

EJEMPLO:

SWAPF INDI,0

Si antes de la instrucción:

INDI = 1010 0101 h

Como d=0 el resultado se almacenar en W, luego de la ejecución de la instrucción tendremos:

INDI = 1010 0101 b

W = 0101 1010 b

XORLW	Exclusive OR Literal With k	XORLW
Realiza la operación lógica OR EXCLUSIVA entre el contenido del registro de trabajo y un valor expresado en el operando		
Operación	$(W) XOR k \rightarrow (W)$	

Manejo de las Instrucciones de un PIC

XORLW	Exclusive OR Literal With k			XORLW
Realiza la operación lógica OR EXCLUSIVA entre el contenido del registro de trabajo y un número expresado en el operando				
Operación	$(W).XOR.k \rightarrow (W)$			
Sintaxis	[Etiqueta] XORLW k			
Operadores	$0 < f < 255$			
Palabras	1			
Ciclos	1			
Código de Operación	11	1010	kkkk	kkkk
Descripción	Realiza la función OR-Exclusiva entre el contenido del registro W y la constante k de 8 bits. El resultado se almacena en W			

Registro de STATUS

PA0	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	-	-

Z Se pone a 1 si el resultado de la última operación es cero.

EJEMPLO:

XORLW B 0011 1000

Si antes de la instrucción:

W = 1011 0101 b

Al ejecutarse la instrucción:

W = 1011 0101 b \oplus 0011 1000 b = 1000 1101

XORWF	Exclusive OR With f		XORWF
Realiza la operación lógica OR EXCLUSIVA entre el contenido de un registro y el contenido del registro de trabajo			
Operación	$(W).XOR.(f) \rightarrow (dest)$		
Sintaxis	[Etiqueta] XORWF f,d		

XORWF	Exclusive OR With f	XORWF
Realiza la operación lógica OR EXCLUSIVA entre el contenido de un registro y el contenido del registro de trabajo		
Operadores	0 < f < 127 d [0,1]	
Palabras	1	
Ciclos	1	
Descripción	Realiza la función OR-Exclusiva entre el contenido del registro W y el contenido del registro f y almacena el resultado en f si d=1 y en W si f=0	

Registro de STATUS

PA0	PA1	PA0	TO#	PD#	Z	DC	C
-	-	-	-	-	X	-	-

EJEMPLO:

XORWF REG,1

Si antes de la instrucción:

REG = 0011 1000 b

W = 1011 0101 b

Como d=1, el resultado se almacena en REG

Al ejecutarse la instrucción:

REG = 1011 0101 b \oplus 0011 1000 b = 1000 1101

W = 1011 0101 b

Nota 1:

En las instrucciones CALL y GOTO los 11 <10:0> bits de menos peso corresponden al código de operación, mientras que los 2 bits de mayor peso <12:11> los suministran respectivamente los bits 4 y 3 del registro PCLATH que realmente apuntan a una de las 4 páginas del mapa de memoria que puede direccionar el PC.

Palabras de Configuración e Identificación

Los PIC de la gama media disponen de una palabra de configuración de 14 bits que se escribe durante el proceso de grabación del dispositivo y que debe hacerse de acuerdo con el sistema en el que se va a insertar. Dichos bits ocupan la posición reservada de memoria de programa 2007h. La estructura de la palabra de configuración es la siguiente:

Posición de Memoria 2007h
Reservada para la palabra de configuración

									FC	PWRTE	WDT	FOSC1	FOSC2
Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 4: CP, bit de configuración de protección de código

- 1= Protección de código desactivado
- 0= Protección de código activado

Bit 3: PWRTE, Activación del temporizador "Power-Up"

- 1= Desactivado
- 0= Activado

Bit 2: bit de configuración habilitación del Watchdog (WDT)

- 1: WDT activado
- 0: WDT desactivado

Bit 1 y Bit 0: FOSC<1:0>, tipo de oscilador empleado

- 11: Oscilador RC
- 10: Oscilador HS (8 - 20MHz)
- 01: Oscilador XT (100 KHz- 4MHz)
- 00: Oscilador LP (Bajo consumo 32- 200Hz)
