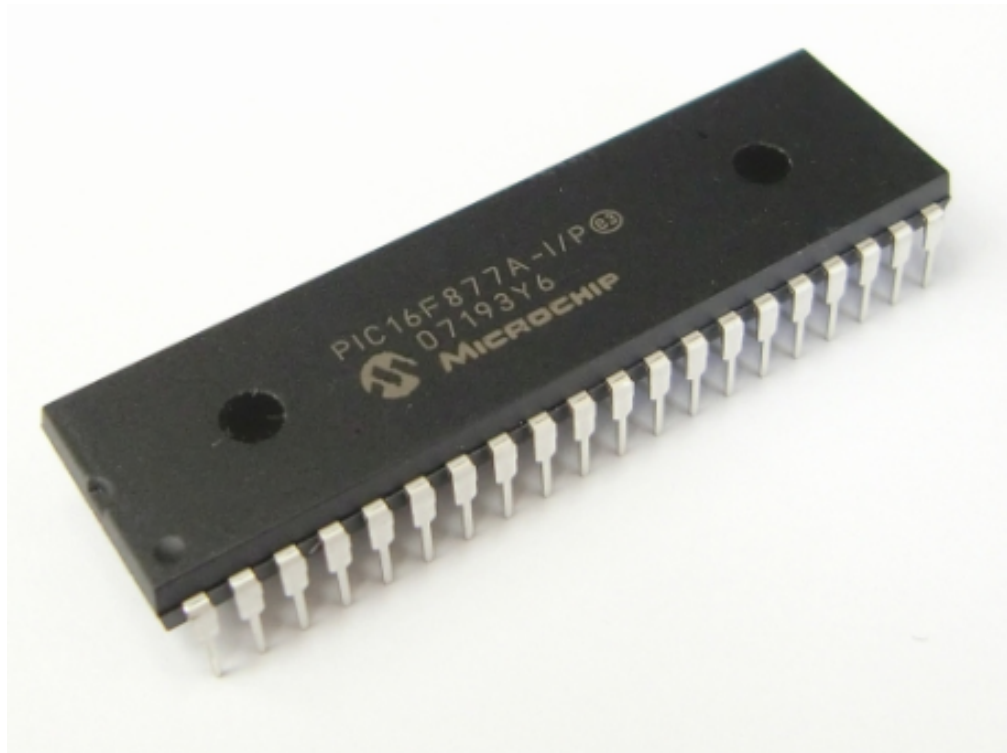


APROBADO

08/12/2014 06:26:21 P.M.



Instrucciones para programar el PIC16F877



PARA DIFUSIÓN PÚBLICA

Estos microcontroladores responden a una serie de instrucciones o códigos que se deben grabar en su memoria de programa, en total son 35. A continuación se encuentra una tabla con la lista completa y después una descripción de cada una de ellas con el fin de facilitar su aprendizaje.

Si d = 0 el resultado de la operación se almacena en el registro W				
Si d = 1 el resultado se almacena en el registro utilizado				
Operaciones orientadas a registros				
Nemotécnico	Operación	Cód. de operación (14 bits)		Estados afectados
ADDWF f,d	Sumar W y f	00 0111	dfff ffff	C,DC,Z Z Z Z
ANDWF f,d	AND entre W y f	00 0101	dfff ffff	
CLRF f	Limpiar f	00 0001	1fff ffff	
CLRWF	Limpiar w	00 0001	0XXX XXXX	
COMF f,d	Complementar f	00 1001	dfff ffff	Z
DECF f,d	Decrementar f	00 0011	dfff ffff	Z
DECFSZ f,d	Decrementar f, saltar si cero	00 1011	dfff ffff	Z
INCF f,d	Incrementar f	00 1010	dfff ffff	Z
INCFSZ f,d	Incrementar f, saltar si cero	00 1111	dfff ffff	
IORWF f,d	OR entre W y f	00 0100	dfff ffff	Z
MOVF f,d	Mover f	00 1000	dfff ffff	Z
MOVWF f	Mover W a f	00 0000	1fff ffff	
NOP	No operación	00 0000	0XX0 0000	
RLF f,d	Rotar a la izquierda a través del carry	00 1101	dfff ffff	C
RRF f,d	Rotar a la derecha a través del carry	00 1100	dfff ffff	C
SUBWF f,d	Restar W de f	00 0010	dfff ffff	C,DC,Z
SWAPF f,d	Intercambiar nibbles de f	00 1110	dfff ffff	Z
XORWF f,d	OR exclusiva entre W y f	00 0110	dfff ffff	
Operaciones orientadas a bits				
BCF f,b	Limpiar bit b de f	01 00bb	bfff ffff	
BSF f,b	Activar bit b de f	01 01bb	bfff ffff	
BTFSC f,b	Probar bit b de f, saltar si es cero	01 10bb	bfff ffff	
BTFSS f,b	Probar bit b de f, saltar si es uno	01 11bb	bfff ffff	
Operaciones con constantes y de control				
ADDLW k	Sumar literal k a W	11 111X	kkkk kkkk	C,DC,Z
ANDLW k	AND entre k y W	11 1001	kkkk kkkk	Z
CALL k	Llamar subrutina	10 0kkk	kkkk kkkk	
CLRWDT	Limpiar WDT	00 0000	0110 0100	T0,PD
GOTO k	Salta a dirección k	10 1kkk	kkkk kkkk	
IORLW k	OR entre k y W	11 1000	kkkk kkkk	Z
MOVLW k	Cargar a W con literal k	11 00XX	kkkk kkkk	
RETFIE	Retornar de interrupción	00 0000	0000 1001	
RETLW k	Retornar y cargar a W con k	11 01XX	kkkk kkkk	
RETURN	Retornar de subrutina	00 0000	0000 1000	T0,PD
SLEEP	Ir al modo de bajo consumo	00 0000	0110 0011	C,DC,Z
SUBLW k	Restarle k a W	11 110X	kkkk kkkk	Z
XORLW k	OR exclusiva entre k y W	11 1010	kkkk kkkk	

Instrucción ADDLW: Suma un valor constante k al contenido del registro W, el resultado se guarda en el registro W.

Sintaxis: ADDLW k
 Operación: (W) + k
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: C, DC, Z

Ejemplo : ADDLW d'15'
 Antes de la instrucción registro W = d'10'
 Después de la instrucción registro W = d'25'

Instrucción ADDWF: Suma el contenido de un registro f al contenido del registro W, el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis: ADDWF f,d
 Operación: (W) + (f)
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: C, DC, Z

Ejemplo: ADDWF 15h,0
 Antes de la instrucción registro W = 17h registro 15h = 0C2h
 Después de la instrucción registro W = 0D9h registro 15h = 0C2h

Instrucción ANDLW: Operación lógica AND entre una constante k y el registro W, la operación se hace bit a bit, el resultado queda en el registro W.

Sintaxis : ANDLW k
 Operación : (W) AND (k)
 Ciclos de instrucción : 1
 Bits del registro de estados que se afectan: Z

Ejemplo: ANDLW b'10101010'
 Antes de la instrucción registro W = b'11110000'
 Después de la instrucción registro W = b'10100000'

Instrucción ANDWF: Operación lógica AND entre un registro f y el registro W, el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis : ANDWF f,d
 Operación : (W) AND (f)
 Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Z

Ejemplo : ANDWF FSR,1

Antes de la instrucción registro W = b'00010111' registro FSR = b'11000010'

Después de la instrucción registro W = b'00010111' registro FSR = b'00000010'

Instrucción BCF: Pone en cero el bit b del registro f.

Sintaxis : BCF f,b

Operación : 0 (f)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Ninguno

Ejemplo : BCF regis,7

Antes de la instrucción regis = b'11111111'

Después de la instrucción regis = b'01111111'

Instrucción BSF: Pone en uno el bit b del registro f.

Sintaxis : BSF f,b

Operación : 1 (f)

Ciclos de instrucción : 1

Bits del registro de estados que se afectan: Ninguno

Ejemplo : BSF regis,2

Antes de la instrucción regis = b'00000000'

Después de la instrucción regis = b'00000100'

Instrucción BTFSC: Pregunta por el bit b del registro f, si dicho bit está en cero el programa se salta una línea y ejecuta la instrucción que allí se encuentre, si el bit estaba en uno no salta una línea sino que ejecuta la instrucción inmediatamente siguiente.

Sintaxis : BTFSC f,b

Operación : salta si (f) = 0

Ciclos de instrucción : 1 ó 2

Bits del registro de estados que se afectan: Ninguno

Ejemplo : línea 1 BTFSC regis,0

 Línea 2 GOTO inicio

 Línea 3

Antes de la instrucción	apuntador de programa = línea 1
Después de la instrucción	si el bit 0 del registro regis = 0 apuntador de programa = línea 3
	si el bit 0 del registro regis = 1 apuntador de programa = línea 2

Instrucción BTFSS: Pregunta por el bit b del registro f, si dicho bit está en uno el programa se salta una línea y ejecuta la instrucción que allí se encuentre, si el bit estaba en cero no salta una línea sino que ejecuta la instrucción inmediatamente siguiente.

Sintaxis : BTFSS f,b
 Operación : salta si (f) = 1
 Ciclos de instrucción : 1 ó 2
 Bits del registro de estados que se afectan: Ninguno

Ejemplo : línea 1 BTFSS regis,0
 línea 2 GOTO inicio
 línea 3

Antes de la instrucción	apuntador de programa = línea 1
Después de la instrucción	si el bit 0 del registro regis = 1 apuntador de programa = línea 3
	si el bit 0 del registro regis = 0 apuntador de programa = línea 2

Instrucción CALL: Llama una subrutina que está ubicada en la posición de memoria o etiqueta "k".

Sintaxis : CALL k
 Operación : carga el apuntador de programa con la dirección "K"
 Ciclos de instrucción : 2
 Bits del registro de estados que se afectan: Ninguno

Ejemplo : aquí CALL rutina

Antes de la instrucción	contador de programa = aquí
Después de la instrucción	contador de programa = rutina En la pila se guarda la dirección "aquí" para regresar

Instrucción CLRF: Borra el contenido del registro "f", lo carga con 00.

Sintaxis : CLRF f
Operación : borra el contenido del registro f (lo carga con 00)
Ciclos de instrucción : 1
Bits del registro de estados que se afectan: Z

Ejemplo : CLRF regis

Antes de la instrucción	regis = 5Ah
Después de la instrucción	regis = 00

Instrucción CLRW: Borra el contenido del registro W (lo carga con 00)

Sintaxis: CLRW
Operación: Registro W = 00
Ciclos de instrucción: 1
Bits del registro de estados que se afectan: Ninguno

Ejemplo : CLRW

Antes de la instrucción	registro W = 5Ah
Después de la instrucción	registro W = 00

Instrucción CLRWDT: Reinicia en cero el conteo del temporizador interno *Watchdog Timer*, para que no sea reseteado el microcontrolador.

Sintaxis: CLRWDT
Operación: contador del temporizador *watchdog timer* = 00
Ciclos de instrucción : 1
Bits del registro de estados que se afectan: TO, PD

Ejemplo : CLRWDT

Antes de la instrucción	Contador WDT = ?
Después de la instrucción	Contador WDT = 00

Bits del registro de estados: TO = 1, PD = 1

Instrucción COMF: Complementa el contenido del registro "f" (cambia unos por ceros y viceversa), el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis: COMF f,d
Operación: (complementar registro f)
Ciclos de instrucción: 1

Bits del registro de estados que se afectan: Z

Ejemplo : COMF regis,0
 Antes de la instrucción regis = b'00101100', W = ?
 Después de la instrucción regis = b'00101100', W = b'11010011'

Instrucción DECF: Decrementa en uno el contenido del registro "f", el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis: DECF f,d
 Operación: (f) - 1
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: Z

Ejemplo: DECF regis,1
 Antes de la instrucción regis = d'13'
 Después de la instrucción regis = d'12'

Instrucción DECFSZ: Decrementa en uno el contenido del registro "f", si el contenido queda en 00 el micro salta una línea del programa, el resultado obtenido se guarda en el registro escogido según el valor del bit d.

Sintaxis: DECFSZ f,d
 Operación : (f) - 1 , salta una línea si el resultado es cero
 Ciclos de instrucción : 1 (2)
 Bits del registro de estados que se afectan: Ninguno

Ejemplo : aquí DECFSZ regis, 1
 GOTO ciclo
 continúa

Antes de la instrucción contador de programa = aquí
 Después de la instrucción regis = regis - 1
 si regis = 0, entonces
 contador de programa = continúa
 si regis no es 0, entonces
 contador de programa = aquí + 1 (ejecuta GOTO ciclo)

Instrucción GOTO: El contador de programa salta a la dirección "k"

Sintaxis: GOTO k
 Operación: El contador de programa salta a la dirección k
 Ciclos de instrucción: 2
 Bits del registro de estados que se afectan: Ninguno

Ejemplo: GOTO ciclo
 Antes de la instrucción contador de programa = ?
 Después de la instrucción contador de programa = ciclo

Instrucción INCF: Incrementa en uno el contenido del registro "f", el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis: INCF f,d
 Operación: (f) + 1
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: Z

Ejemplo : INCF regis,1

 Antes de la instrucción regis = d'24'
 Después de la instrucción regis = d'25'

Instrucción INCFSZ: Incrementa en uno el contenido del registro "f", si el contenido de "f" queda en 00 el micro salta una línea del programa, el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis: INCFSZ f,d
 Operación: (f) + 1 , salta si el resultado es cero
 Ciclos de instrucción: 1 (2)
 Bits del registro de estados que se afectan: Ninguno

Ejemplo: aquí DECFSZ regis, 1
 GOTO ciclo
 continúa.....

 Antes de la instrucción contador de programa = aquí
 Después de la instrucción regis = regis + 1
 si regis = 0, entonces
 contador de programa = continua
 si regis no es 0, entonces
 contador de programa = aquí + 1 (ejecuta GOTO ciclo)

Instrucción IORLW: Operación lógica OR entre el contenido del registro W y la constante k, el resultado queda en el registro W.

Sintaxis: IORLW k
 Operación: (W) OR (k)
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: Z

Ejemplo: IORLW b'10101100'

Antes de la instrucción registro W = b'00001111'

Después de la instrucción registro W = b'10101111'

Instrucción IORWF: Operación lógica OR entre el registro W y el registro "f", el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis: IORWF f,d

Operación: (W) OR (f)

Ciclos de instrucción: 1

Bits del registro de estados que se afectan: Z

Ejemplo: IORWF regis,0

Antes de la instrucción regis = b'00110011 , W = b'11110000'

Después de la instrucción regis = b'00110011' , W = b'11110011'

Instrucción MOVLW: Carga el registro W con el valor constante k.

Sintaxis: MOVLW k

Operación: (W) se carga con el valor k

Ciclos de instrucción: 1

Bits del registro de estados que se afectan: Ninguno

Ejemplo: MOVLW 5Ah

Antes de la instrucción W = ?

Después de la instrucción W = 5Ah

Instrucción MOVF: Mueve el contenido del registro "f" hacia el registro W.

Sintaxis: MOVF f

Operación: (W) se carga con (f)

Ciclos de instrucción: 1

Bits del registro de estados que se afectan: Z

Ejemplo : MOVF regis,0

Antes de la instrucción registro W = ?, regis = d'32'

Después de la instrucción registro W = d'32'

Instrucción MOVWF: Mueve el contenido del registro W al registro "f".

Sintaxis: MOVWF f
 Operación: (W) se mueve al registro (f)
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: Ninguno

Ejemplo: MOVWF 20h

Antes de la instrucción registro 20h = ? , registro W = d'48'
 Después de la instrucción registro 20h = d'48' , registro W = d'48'

Instrucción NOP: No hace nada.

Sintaxis: NOP
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: Ninguno

Instrucción RETURN: Retorno desde una subrutina.

Sintaxis: RETURN
 Operación: carga el contador de programa con la dirección donde debe regresar luego de ejecutar la subrutina, la cual estaba guardada en la pila o stack.
 Ciclos de instrucción: 2
 Bits del registro de estados que se afectan: Ninguno

Ejemplo : RETURN
 Después de la interrupción, contador de programa = dato de la pila

Instrucción RETLW: Retorno desde una subrutina y adicionalmente carga el registro W con el valor constante k.

Sintaxis: RETLW k
 Operación: contador de programa se carga con el valor de la pila, además (W) = k
 Ciclos de instrucción: 2
 Bits del registro de estados que se afectan: Ninguno

Ejemplo : CALL tabla
 .
 .
 tabla NOP
 NOP
 RETLW k1
 Antes de la instrucción registro W = ?

Después de la instrucción registro W = k1, contador de programa = dato de la pila

Instrucción RETFIE: Retorna al programa luego de ejecutar una interrupción.

Sintaxis: RETFIE
 Operación: pila k contador de programa , 1 k intcon,gie
 Ciclos de instrucción: 2
 Bits del registro de estados que se afectan: Ninguno

Ejemplo : RETFIE

Antes de la instrucción contador de programa = ?
 Después de la instrucción contador de programa = dato de la pila, las interrupciones se habilitan nuevamente

Instrucción RLF: Rotar el contenido del registro "f" un bit a la izquierda, usando el carry como bit intermedio, el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis: RLF f,d
 Operación: Rota el contenido del registro "f" a la izquierda a través del carry
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: C

Ejemplo : RLF regis,0

Antes de la instrucción bit carry = 0, regis = 1110 0110, W=?
 Después de la instrucción bit carry = 1, regis = 1110 0110
 W = 11001100

Instrucción RRF: Rotar el contenido del registro "f" un bit a la derecha, usando el carry como bit intermedio, el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis: RRF f,d
 Operación: Rota el contenido del registro "f" a la derecha a través del carry
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: C

Ejemplo : RRF regis,0

Antes de la instrucción bit carry = 0, regis = 1110 0101, W=?
 Después de la instrucción bit carry = 1, regis = 1110 0101
 W = 0111 0010

Instrucción SLEEP: El micro entra en un estado de *standby* (dormido), de ese modo solamente sale si ocurre una interrupción.

Sintaxis: SLEEP
 Operación: bit TO = 1, bit PD = 0
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: TO, PD

Ejemplo : SLEEP

Instrucción SUBLW: Resta el contenido del registro W de la constante k usando el método de complemento a dos, el resultado se guarda en el registro W, el bit de carry indica el signo de la respuesta.

Sintaxis: SUBLWk
 Operación: k - (W)
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: C, DC, Z

Ejemplo 1: SUBLW 5h
 Antes de la instrucción W = 1h, bit de carry = ?
 Después de la instrucción W = 4h, bit de carry = 1 (positivo)

Ejemplo 2: SUBLW 5h
 Antes de la instrucción W = 5h, bit de carry = ?
 Después de la instrucción W = 0, bit de carry = 1 (positivo)

Ejemplo 3: SUBLW 5h
 Antes de la instrucción W = 6h, bit de carry = ?
 Después de la instrucción W = 0FFh , bit de carry = 0 (negativo)

Instrucción SUBWF: Resta el contenido del registro W del registro "f" usando el método de complemento a dos, el bit de carry indica el signo de la respuesta, el resultado se guarda en el registro escogido según el valor del bit d.

Sintaxis: SUBWF f,d
 Operación: (f) - (W)
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: C, DC, Z

Ejemplo 1: SUBWF regis,1
 Antes de la instrucción regis = 5h, W = 2h, bit de carry = ?

Después de la instrucción regis = 3h, W = 2h, bit de carry = 1 (positivo)

Ejemplo 2: SUBWF regis,1
 Antes de la instrucción regis = 2h, W = 2h, bit de carry = ?
 Después de la instrucción regis = 0, W = 2h, bit de carry = 1 (positivo)

Ejemplo 3: SUBWF regis,1
 Antes de la instrucción regis = 5h, W = 6h, bit de carry = ?
 Después de la instrucción regis = 0FFh, W = 6h, bit de carry = 0 (negativo)

Instrucción SWAPF: Intercambia los cuatro bits altos y los cuatro bits bajos del registro "f", el nuevo dato obtenido se guarda en el registro seleccionado según el valor del bit d.

Sintaxis: SWAPF f,d
 Operación: los bits f<3:0> se intercambian de posición con los bits f<7:4>
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: Ninguno

Ejemplo 1: SWAPF regis,0

Antes de la instrucción regis = b'10101100', W = ?
 Después de la instrucción regis = b'10101100', W = b'11001010'

Instrucción XORLW: Operación lógica XOR entre el registro W y la constante k, el resultado se guarda en el registro W.

Sintaxis: XORLW k
 Operación: (W) XOR (k)
 Ciclos de instrucción: 1
 Bits del registro de estados que se afectan: Z

Ejemplo: XORLW b'10101100'

Antes de la instrucción registro W = b'111110000'
 Después de la instrucción registro W = b'01011100'

Instrucción XORWF: Operación lógica XOR entre el registro W y el registro "f", el resultado obtenido se guarda en el registro seleccionado según el bit d.

Sintaxis: XORWF f,d

APROBADO

08/12/2014 06:52:38 P.M.

Instrucciones para programar el PIC16F877

Operación: (W) XOR (f)

Ciclos de instrucción: 1

Bits del registro de estados que se afectan: Z

Ejemplo: XORWF regis,1

Antes de la instrucción regis = b'11110000', W = b'10101010'

Después de la instrucción regis = b'01011010', W = b'10101010'

Sintaxis	Descripción	Ciclos de Instrucción
Instrucciones de carga		
CLRF f	Borra el contenido del registro f	1
CLRW	Borra el contenido del registro W	1
MOVF f ,d	Mueve el contenido del registro f	1
MOVWF f	Mueve el contenido del registro W al registro f	1
MOVLW k	Guarda en el registro W el valor k	1
Instrucciones de bit		
BCF f, b	Pone en "0" el bit b del registro f	1
BSF f, b	Pone en "1" el bit b del registro f	1
Instrucciones aritméticas		
ADDLW k	Suma el valor k al registro W	1
ADDWF f, d	Suma el contenido del registro W con el de f	1
DECF f, d	Decrementa en una unidad el contenido del registro f	1
INCF f, d	Incrementa en una unidad el contenido del registro f	1
SUBLW k	Resta el valor k al registro W	1
SUBWF f, d	Resta el contenido del registro W al registro f	1
Instrucciones lógicas		
ANDLW k	Realiza la operación lógica AND entre el valor k y el registro W	1
ANDWF f, d	Realiza la operación lógica AND entre el contenido del registro W y el de f	1
COMF f, d	Niega el valor del registro f	1
IORLW k	Realiza la operación lógica OR entre el valor k y el registro W	1
IORWF f, d	Realiza la operación lógica OR entre el contenido del registro W y el de f	1
RLF f, d	Rota el contenido del registro f hacia la izquierda a través del carry bit	1
RRF f, d	Rota el contenido del registro f hacia la derecha a través del carry bit	1
SWAPF f, d	Intercambia los cuatro primeros bits con los cuatro últimos del registro f	1
XORLW	Realiza la operación lógica XOR entre el valor k y el registro W	1
XORWF f, d	Realiza la operación lógica XOR entre el contenido del registro W con el de f	1
Instrucciones de salto		
BTFSC f, b	Lee el bit b del registro f y salta la instrucción siguiente si está en "0"	1 (2 si salta)
BTFSS f, b	Lee el bit b del registro f y salta la instrucción siguiente si está en "1"	1 (2 si salta)
DECFSZ f, d	Decrementa en una unidad el contenido del registro f y salta la instrucción siguiente si es cero	1 (2 si salta)
INCFSZ f, d	Incrementa en una unidad el contenido del registro f y salta la instrucción siguiente si es cero	1 (2 si salta)
GOTO k	Salto incondicional a la etiqueta k	2
Instrucciones de manejo de subrutinas		
CALL k	Llamada a la subrutina k	2
RETFIE	Retorno de una interrupción	2
RETLW	Retorno de una subrutina con el valor k en el registro W	2
RETURN	Retorno de una subrutina	2

Instrucciones especiales		
CLRWDT	Borra el temporizador del Watchdog	1
NOP	No realiza ninguna operación (pero consume tiempo de ejecución)	1
SLEEP	Se pone el PIC en estado de reposo (dormido)	1