

## SET DE INSTRUCCIONES RISC PARA MICROCONTROLADORES PIC MIDRANGE

Existen un total de 35 instrucciones. Todas las instrucciones son palabras de 14 bits, divididas en dos partes: el código de operación y el operando. Los operandos, pueden ser bytes o bits de memoria o registros. De esta forma se puede hablar de instrucciones “orientadas a bytes” u “orientadas a bits”.

Todas las instrucciones, con excepción de las que modifican el contenido del contador del programa (como son los saltos y llamados a subrutina) se ejecutan en un ciclo de instrucción, es decir, 4 ciclos de reloj. Para un sistema funcionando a 4 Mhz, cada instrucción se ejecuta en 1 microsegundo. Si la instrucción modifica el contenido del contador del programa, entonces el tiempo de ejecución es de 2 ciclos de instrucción ó 2 microsegundos para el ejemplo a 4 Mhz. El grupo de 35 instrucciones es el siguiente:

### MOVE GROUP

<b>movf</b>	<b>f,d</b>	<b>move f</b>
<b>movwf</b>	<b>f</b>	<b>move w to f</b>
<b>movlw</b>	<b>k</b>	<b>move literal to w</b>
<b>clrf</b>	<b>f</b>	<b>clear f</b>
<b>clrw</b>		<b>clear w</b>
<b>swapf</b>	<b>f,d</b>	<b>swap nibbles in f</b>

### ARITHMETIC GROUP

<b>addwf</b>	<b>f,d</b>	<b>add w and f</b>
<b>addlw</b>	<b>k</b>	<b>add literal to w</b>
<b>subwf</b>	<b>f,d</b>	<b>subtract w from f</b>
<b>sublw</b>	<b>k</b>	<b>subtract w from literal</b>
<b>incf</b>	<b>f,d</b>	<b>increment f</b>
<b>incfsz</b>	<b>f,d</b>	<b>increment f, skip if 0</b>
<b>decf</b>	<b>f,d</b>	<b>decrement f</b>
<b>decfsz</b>	<b>f,d</b>	<b>decrement f, skip if 0</b>

### LOGIC GROUP

<b>andwf</b>	<b>f,d</b>	<b>and w and f</b>
<b>andlw</b>	<b>k</b>	<b>and literal to w</b>
<b>iorwf</b>	<b>f,d</b>	<b>inclusive or w and f</b>
<b>iorlw</b>	<b>k</b>	<b>inclusive or literal to w</b>
<b>xorwf</b>	<b>f,d</b>	<b>exclusive or w and f</b>
<b>xorlw</b>	<b>k</b>	<b>exclusive or literal to w</b>
<b>comf</b>	<b>f,d</b>	<b>complement f</b>
<b>rlf</b>	<b>f,d</b>	<b>rotate left f, through carry</b>
<b>rrf</b>	<b>f,d</b>	<b>rotate right f, through carry</b>

### BIT GROUP

<b>bcf</b>	<b>f,b</b>	<b>bit clear in f</b>
<b>bsf</b>	<b>f,b</b>	<b>bit set in f</b>
<b>btfsc</b>	<b>f,b</b>	<b>bit test in f, skip if clear</b>
<b>btfss</b>	<b>f,b</b>	<b>bit test in f, skip if set</b>

### CONTROL GROUP

<b>clrwdt</b>		<b>clear watchdog timer</b>
<b>sleep</b>		<b>go into sleep mode</b>
<b>nop</b>		<b>no operation</b>

### BRANCH GROUP

<b>goto</b>	<b>k</b>	<b>goto address</b>
<b>call</b>	<b>k</b>	<b>call subrutine</b>
<b>return</b>		<b>return from subrutine</b>
<b>retlw</b>	<b>k</b>	<b>return with literal in w</b>
<b>retfie</b>		<b>return from interrupt</b>
<b>incfsz</b>	<b>f,d</b>	<b>increment f, skip if 0</b>

<b>decfsz</b>	<b>f,d</b>	<b>decrement f, skip if 0</b>
<b>btfsc</b>	<b>f,b</b>	<b>bit test in f, skip if clear</b>
<b>btfss</b>	<b>f,b</b>	<b>bit test in f, skip if set</b>

### **OPERANDOS:**

Los operandos pueden asignarse con las letras f, w, b, k, d. Cada letra tiene el significado siguiente:

- f*: designa alguna localidad de memoria (file register), de alguno de los 4 bancos de los llamados “registros especiales”, o bien, alguna de las localidades de memoria RAM.
- w*: designa el acumulador del microcontrolador.
- b*: designa alguno de los 8 bits del registro especial o localidad de memoria elegido.
- k*: designa una constante ó una dirección.
- d*: designa el destino en donde se almacenará el resultado de la operación. Si d=0, el destino es el registro w. Si d=1, entonces el destino es el registro ó localidad de memoria f.

### **Uso de la directiva EQU:**

Dentro del archivo del programa fuente, es decir del programa escrito en lenguaje ensamblador, se puede, con ayuda de la directiva EQU, definir previamente valores para la facilidad de su identificación. Algunas de las definiciones más utilizadas son las siguientes:

w	EQU	H'0000'	PORTA	EQU	H'0005'
f	EQU	H'0001'	PORTB	EQU	H'0006'
STATUS	EQU	H'0003'	TRISA	EQU	H'0085'
RP1	EQU	H'0006'	TRISB	EQU	H'0086'
RP0	EQU	H'0005'	PCLATH	EQU	H'000A'

Estas declaraciones, junto con las del resto de los registros, así como los bits individuales de cada registro, se almacenan en un solo archivo que es parte de las librerías ya incluidas en el programa MPLAB-IDE, del cual se habla más adelante. Entonces, es suficiente escribir, dentro del programa fuente en lenguaje ensamblador, la directiva: **include <p16F88.inc>** ó el nombre del archivo correspondiente al microcontrolador empleado.

Una vez establecidas estas equivalencias, podemos poner algunos ejemplos con instrucciones. Observe que, en todas ellas es posible usar las equivalencias o bien escribir directamente el valor numérico. Por ejemplo:

<b><u>Etiqueta</u></b>	<b><u>Instrucción</u></b>	<b><u>Operando</u></b>	<b><u>Forma general:</u></b>
EJEMPLO1	bcf	STATUS,RP0	bcf            f,d

Resultado: Bit Clear f. Pon en cero el bit RP0 del registro STATUS.

EJEMPLO2	bcf	3,5
----------	-----	-----

Resultado: mismo que en el ejemplo anterior, pero usando ahora las constantes directamente al escribir la instrucción. Observe como el hecho de escribir directamente palabras como “STATUS” en lugar del número 3, facilitan mucho la comprensión.

EJEMPLO3	btfss	STATUS,RP1	btfss            f,b
----------	-------	------------	----------------------

Resultado: Bit Test F, Skip if Set. Si el bit RP1 del registro designado es cero, ejecuta la siguiente instrucción, si el bit es 1, entonces no ejecuta la siguiente instrucción, pero sí la que sigue a ésta.

EJEMPLO4	addlw	H'3A'	addlw            k
----------	-------	-------	--------------------

Resultado: Add literal to w. Suma el registro w con la constante 3AH. El resultado lo pone en w.

EJEMPLO5	decfsz	H'20',w	decfsz            f,d
----------	--------	---------	-----------------------

Resultado: Decrement f, Skip if Zero. Decrementa el valor de la localidad 20H. El resultado lo almacena en el registro w. Si el resultado es 0, no ejecuta la siguiente instrucción, sino la inmediata después de ésta. Si el resultado es diferente de 0, entonces ejecuta la siguiente instrucción.

EJEMPLO6	decf	H'2A',w	decf            f,d
----------	------	---------	---------------------

Resultado: Decrementa la localidad 2AH, el resultado lo pone en el registro w.



Es decir que el registro W contiene, antes de la instrucción de “call”, el desplazamiento (offset) deseado sobre la dirección de inicio de la tabla, (que en el ejemplo, puede ser un valor del 1 a 4). Aquí es muy importante señalar que en la instrucción (addwf PCL), se está afectando la parte baja PCL (8 bits) y también la alta (3 bits) del contador del programa, la cual se carga con el contenido de PCLATH. Es necesario entonces inicializar también el registro PCLATH para que contenga el valor de la página en donde se encuentra la tabla, que pueden ser los valores 0 a 7.

```

...main...
movlw 1           ;en este ejemplo, la tabla se encuentra en la página 1.
movwf PCLATH     ;carga el número de página en PCLATH
.....          ;en esta zona del programa, debe de inicializarse w con el desplazamiento.
movlw H'1'       ;en este ejemplo w=1
call TABLA
.....          ;en ésta línea regresa la subrutina TABLA con el dato de la tabla en w.
.....
org H'100'       ;origen de la tabla en la página 1. (100H ... 1FFH)
TABLA addwf PCL  ;ésta instrucción suma a PCL el contenido de w, y carga PCLATH en la parte al-
                ;ta del contador del programa.
retlw 'H'        ;regresa de la subrutina con el código ASCII de la “H” en w.
retlw 'O'
retlw 'L'
retlw 'A'

```