MODOS DE DIRECCIONAMIENTO DE LOS DATOS

Los DSC disponen de 4 modos para direccionar los datos u operandos cuando se ejecutan las instrucciones.

1. Direccionamiento inmediato
2. Direccionamiento directo por registro
3. Direccionamiento directo.
4. Direccionamiento indirecto por registro

Direccionamiento inmediato

En este caso el operando que maneja la instrucción es un valor literal constante predefinido que esta contenido en el formato de la propia instrucción. Se usa de forma independiente y combinada con los restantes modos de direccionamiento. El tamaño del operando inmediato puede abarcar desde 1 a 16 Bits y puede tener o no bit de signo.

EJEMPLO 1 (LITERAL DE 16 BITS)

```
MOV #13645, W1          ;W1 = 13645
```

Se mueve el número inmediato 13645 al registro W1.

EJEMPLO 2 (LITERAL DE 5 BITS)

```
MOV #0X1234, W7         ;W7 = 0X1234
MOV #0X3675, W6         ;W6 = 0X3675
AND.B W6, #0x0F, W7     ;W7<7:0> = W6<7:0> AND 0x0F
                        ;W7<7:0> = 0x75 AND 0x0F
                        ;W7<7:0> = 05
                        ;W7 = 0x1205
```

Se mueve el número hexadecimal 0X1234 al registro W7 y el número 0x3675 al registro W6. Después, se realiza la operación lógica AND, entre el valor inmediato expresado por el byte 0x0F y el byte de menos significativo de W6, depositándose el resultado en el byte menos significativo de W7. El byte más significativo de W7 no se modifica.

EJEMPLO 3 (LITERAL DE 10 BITS)

```
MOV #56, W2             ;W2 = 56
ADD #1000, W2           ;W2 = W2 + 1000
                        ;W2 = 56 + 1000 = 1056
```

Se mueve el número 56 al registro W2. Después, se suma el valor del registro W2 con el operando inmediato 1000, depositándose el resultado en el registro W2.

Direccionamiento directo por registro

Se emplea para acceder al contenido de los 16 registros de trabajo (W0 – W15) que actúan como operando en la instrucción. El acceso puede ser un dato de tamaño byte o palabra. La ventaja de este modo de direccionado es la rapidez con que se acceden a los registros internos de la MCU.

EJEMPLO 1

```
MOV #34, W0          ;W0 = 34
MOV #67, W1          ;W1 = 67
ADD W0, W1, W2       ;W2 = W0 + W1
                     ;W2 = 34 + 67
                     ;W2 = 101
```

Se mueve el número 34 al registro W0 y el número 67 al registro W1. Después, se suma el valor del registro W0 con el valor del registro W1, depositándose el resultado en el registro W2.

EJEMPLO 2

```
MOV #0XC3F7, W2      ;W2 = 0XC3F7
MOV #5, W3           ;W3 = 5
SL W2, W3, W4        ;W4 = W2 << W3
                     ;W4 = 0XC3F7 << 5
                     ;W4 = 1100 0011 1111 0111 << 5
                     ;W4 = 0111 1110 1110 0000 = 0X7EE0
```

Se mueve el número 0XC3F7 al registro W2 y el número 5 al registro W3. Después, se desplaza a la izquierda el contenido del registro W2 el número de veces que indica W3. El resultado se carga en W4.

EJEMPLO 3

```
MOV #0XFC83, W1      ;W1 = 0XFC83
MOV #0X4018, W3      ;W3 = 0X4018
MOV #0X2931, W2      ;W2 = 0X2931
IOR.B W1, W3, W2     ;W2 <7:0> = W1 <7:0> OR W3<7:0>
                     ;W2 <7:0> = 0X83 OR 0X18
                     ;W2 <7:0> = 1000 0011 OR
                     ;                0001 1000
                     ;W2 <7:0> = 1001 1011 = 0X9B
                     ;W2 = 0X299B
```

Se mueve el número 0XFC83 al registro W1, el número 0X4018 al registro W3 y el número 0X2931 al registro W2. Se realiza la operación lógica OR entre el byte menos significativo de W1 y W3, depositándose el resultado en el byte menos significativo de W2. El byte más significativo de W2 no se modifica.

Direccionamiento directo

Lo emplean las instrucciones que tienen como operando la dirección de la memoria de datos donde reside el contenido de una VARIABLE. Para este direccionamiento es frecuente utilizar los 8 KB inferiores del espacio de la memoria de datos que se denomina **“RAM cercana”**. Sin embargo, con la instrucción MOV es posible acceder a los 64 KB de la memoria de datos con este direccionamiento.

Usando este modo de direccionamiento se puede cargar cualquier posición de la memoria de datos con el contenido de un registro de trabajo W y viceversa. Los operandos pueden tener tamaño byte o palabra, excepto MOV que sólo trabaja con palabras.

EJEMPLO 1

```
MOV #45, W0          ;W0 = 45
MOV W0, 0X2200   ;[0x2200] = W0 = 45
DEC 0x2200   ;[0x2200] = [0x2200] – 1
                  ;[0x2200] = 45 - 1
                  ;[0x2200] = 44
```

```
MOV #45, W0      ;W0 = 45
MOV W0, VAR1   ;VAR1 = W0 = 45
DEC VAR1      ;VAR1 = VAR1 – 1
                  ;VAR1 = 45 - 1
                  ;VAR1 = 44
```

Se inicializa la dirección de memoria 0x2200 (VAR1) con 45, para esto primero se mueve el número 45 al registro W0 y después se mueve a la dirección de memoria 0x2200 (VAR1). **NO SE PUEDE REALIZAR DIRECTAMENTE UN MOVIMIENTO DE UN NÚMERO INMEDIATO A UNA DIRECCIÓN DE MEMORIA (VARIABLE), SE TIENE QUE REALIZAR A TRAVÉS DE UN REGISTRO Wn**. Posteriormente se realiza el decremento del contenido de la dirección de memoria 0x2200 (VAR1).

EJEMPLO 2

La mayoría de las instrucciones que usan el direccionamiento directo toman al registro de trabajo WREG como operando implícito, actuando W0 como WREG. Este tipo de instrucciones comienzan leyendo los operandos, uno de los cuales implícitamente reside en W0, luego se realiza una operación con ellos y, finalmente, el resultado se escribe en WREG o en una dirección de memoria. Son instrucciones de lectura – modificación – escritura.

```
MOV #23, W0          ;W0 = 23
MOV W0, 0X3311     ;[0X3311]= W0 = 23
MOV #74, W0          ;W0 = 74
ADD 0X3311
              ;[0X3311] = [0X3311] + W0
              ;[0X3311] = 23 + 74 = 97
```

```
MOV #23, W0          ;W0 = 23
MOV W0, VAR2        ;VAR2 = W0 = 23
MOV #74, W0          ;W0 = 74
ADD VAR2
              ;VAR2 = VAR2 + W0
              ;VAR2 = 23 + 74 = 97
```

Se inicializa la dirección de memoria 0x3311 (VAR2) con 23, para esto primero se mueve el número 23 al registro W0 y después se mueve a la dirección de memoria 0x3311 (VAR2). **NO SE PUEDE REALIZAR DIRECTAMENTE UN MOVIMIENTO DE UN NÚMERO INMEDIATO A UNA DIRECCIÓN DE MEMORIA (VARIABLE), SE TIENE QUE REALIZAR A TRAVÉS DE UN**

REGISTRO Wn. Después se mueve el número 74 a W0. Posteriormente se realiza la suma del contenido de W0 con el contenido de la dirección de memoria 0x3311 (VAR2). El resultado se guarda en VAR2.

EJEMPLO 3

MOV #0x9876, W0
;W0 = 0x9876
MOV W0, 0X1100
;[0x1100] = W0 = 0x9876
MOV #0XFF0F, W0
;W0 = 0XFF0F
AND 0x1100, WREG
;W0 = [0x1100] AND W0
;W0 = 0x9876 AND 0XFF0F
;W0 = 0X9806
	MOV #0x9876, W0
	;W0 = 0x9876
	MOV W0, VAR3
	;VAR3 = W0 = 0x9876
	MOV #0XFF0F, W0
	;W0 = 0XFF0F
	AND VAR3, WREG
	;W0 = VAR3 AND W0
	;W0 = 0x9876 AND 0XFF0F
	;W0 = 0X9806

Se inicializa la dirección de memoria 0x1100 (VAR3) con 0X9876, para esto primero se mueve el número 0X9876 al registro W0 y después se mueve a la dirección de memoria 0x1100 (VAR3). **NO SE PUEDE REALIZAR DIRECTAMENTE UN MOVIMIENTO DE UN NÚMERO INMEDIATO A UNA DIRECCIÓN DE MEMORIA (VARIABLE), SE TIENE QUE REALIZAR A TRAVÉS DE UN REGISTRO Wn**. Después se mueve el número 0XFF0F a W0. Posteriormente se realiza la operación AND lógica contenido de W0 con el contenido de la dirección de memoria 0x1100 (VAR3). El resultado se guarda en W0 (WREG).

Direccionamiento indirecto por registro

El contenido de los registros de trabajo se utilizan para guardar la dirección efectiva del operando en la memoria de datos. Los registros W actúan como apuntadores, permitiendo incrementar o decrementar automáticamente el contenido de un registro antes o después de ejecutar la instrucción. Con esto se logra la exploración secuencial de arreglos o bancos de datos en el sentido que se desee, según el tamaño de los datos direccionados por los registros W sea de uno o de dos bytes o de mas , los incrementos o decrementos son de una o dos unidades o mas. En la tabla 1 se muestran las modalidades del direccionamiento indirecto por registro.

El direccionamiento indirecto con registro de desplazamiento situado en la última línea de la tabla 1, emplea un registro Wb que suma el contenido con el registro Wn que contiene la dirección efectiva. Wb actúa con el desplazamiento de Wn y permite acceder a los 64 KB de la memoria RAM, aunque carece de posibilidades de incremento y decremento.

Modo indirecto por registro	Sintaxis	Dato tamaño byte	Dato tamaño palabra	Modificación de Wn
Sin modificación	[Wn] *Wn	EA=[Wn]	EA=[Wn]	El contenido de Wn contiene la dirección efectiva y no varía
Pre-incremento	[++Wn] *(++Wn)	EA=[Wn+=1]	EA=[Wn+=2]	Wn es pre-incrementado para formar la dirección efectiva del operando
Pre-decremento	[--Wn] *(--Wn)	EA=[Wn-=1]	EA=[Wn-=2]	Wn es pre-decrementado para formar la dirección efectiva del operando
Post-incremento	[Wn++] *(Wn++)	EA=[Wn]+=1	EA=[Wn]+=2	Wn contiene la dirección del operando y luego es post-incrementado
Post-decremento	[Wn--] *(Wn--)	EA=[Wn]-=1	EA=[Wn]-=2	Wn contiene la dirección del operando y luego es post-decrementado
Registro Desplazamiento	[Wn+Wb] *(Wn+wb)	EA=[Wn+Wb]	EA=[Wn+Wb]	La suma de Wn y Wb forma la dirección efectiva del operando. Wn y Wb no cambian

Tabla 1 Modalidades que admiten el modo de direccionamiento indirecto por registro

EJEMPLO1

```
MOV #0X0206, W0          ;W0 = 0X0206
MOV #0X0100, W3          ;W3 = 0X0100
MOV [W3++], [W0--]       ;[W0] = [W3]
                         ;W3 = W3+2 = 0X0100 + 2 = 0X0102
                         ;W0 = W0–2 = 0X0206 – 2 = 0X0204
```

Con esta instrucción se mueve la palabra apuntada por el contenido de W3 a la posición de memoria que apunta W0. Después W3 se incrementa en 2 unidades y W0 se decrementa en dos unidades también. Todo esto ocurre en un ciclo de instrucción.

EJEMPLO2

```
MOV #0X0308, W4
MOV #0X0100, W5
MOV #0X0020, W6
MOV.B [--W4], [W5+W6]    ;W4 = W4 – 1 = 0X0308 – 1 = 0X0307
                         ;[W5+W6] <7:0>  = [W4]<7:0>
                         ;[0X0100+0X0020] <7:0>  = [W4]<7:0>
                         ;[0X0120] <7:0>  = [W4]<7:0>
```

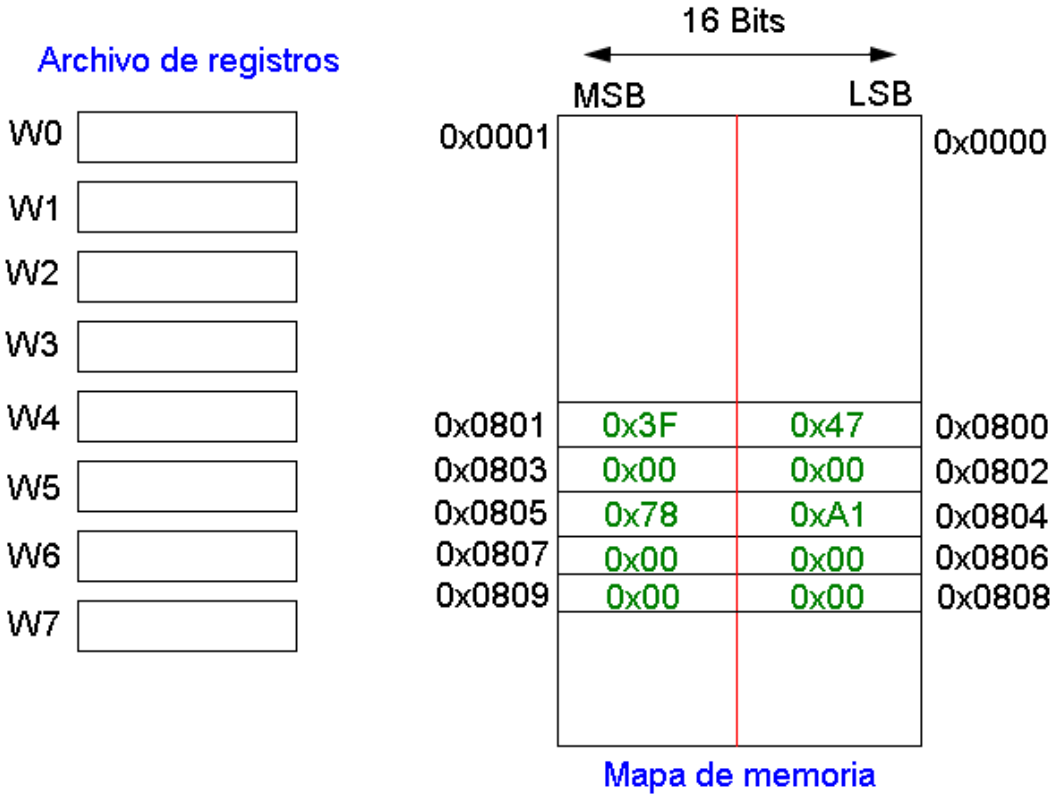
En este ejemplo primero se decrementa el apuntador (W4) en una unidad, esto es, porque se utiliza el especificador “.B” (Byte). Luego con el apuntador modificado de W4 se accede al contenido de la memoria, donde se lee el byte de menos significativo y se coloca en el byte menos significativo de la dirección formada por W5+W6. Todo esto ocurre en un ciclo de instrucción.

EJEMPLO3

```
MOV #0X0106, W5
MOV #0X0200, W7
MOV #0X0300, W8
ADD W7, [++W8], [W5--]   ;W8 = W8 + 2 = 0X0300 + 2 = 0X0302
                         ;[W5] = W7 + [W8]
                         ;W5 = W5 – 2 = 0X0106 – 2 = 0X0104
```

En primer lugar se incrementa en dos unidades el apuntador W8 luego, se suma el contenido de W7 con el contenido de la dirección apuntada por W8 y el resultado se coloca en la dirección apuntada por W5. Finalmente, el contenido de W5 se decrementa en 2 unidades. Todo esto ocurre en un ciclo de instrucción.

Ejercicios. Considere los valores mostrados en los registros y localidades de memoria de la siguiente figura:



Obtener los valores de los registros y/o localidades de memoria después de ejecutar las siguientes instrucciones:

1	2	3

MOV 0x0800, W2
INC W2, W2
	MOV #0X0800, W1
	MOV [W1], W7
	AND.B W7, #0X0F, [++W1]
		MOV #0X0A0A, W6
		SL W6, #3, W6
		MOV W6, 0X0808

4	5	6

MOV #0X034, W4
MOV 0X0804, W6
ADD W4, W6, W6
	MOV #0X0808, W7
	MOV #0X0802, W3
	MOV #0X0800, W5
	ADD W5, [++W3], [w7--]
		MOV #0X0802, W2
		MOV #0X0804, W3
		MOV #0X0802, W7
		SUB.B W7, [W3++], [--W2]

7	8	9

INC.B 0X0804
CLR.B 0x0804
	MOV 0X0800, W0
	MOV #0X0800, W5
	MOV.B W0, [W5+0X03]
		MOV #0X1245, W0
		CLR.B W0
		ADD #0X0378, W0

10	11	12

MOV #0X0800, W5
CLR W6
REPEAT #4
MOV W6, [W5++]
	MOV #1, W1
	MOV #1, W2
	DO #5, CICLO
	ADD W1, W2, W3
	MOV W2, W1
	CICLO:MOV W3, W2
		MOV #0X0800, W1
		CLR W0
		REPEAT #3
		ADD W0, [W1++], W0
		LSR W0, #2, W0