Instrucciones de Movimiento MOV de byte (8 bits) o word (palabras, 16 bits)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nemónico** | **Significado** | **Formato** | **Operación** | **Flags afect.** |
| MOV | Movimiento | MOV D,F | (F) (D) | Ninguno |

y la dirección física se obtiene sumando la dirección de segmento multiplicada por 16 (10H) más el desplazamiento.

Dirección Física = Dirección de Segmento x10H + DESPL

Ejemplo

MOV [1234H], 56H

Mueve el operando inmediato 56H a al posición de memoria DS:1234H.

Si DS=2A00H, la dirección efectiva es 2B234H pues

Direcc. Efectiva = DS\*10H + Despl.

= 2A00H\*10H + 1234 H

= 2A000H + 1234 H = 2B234H

##### **Dirección Efectiva (física) para diferentes modos de direccionamiento de la instrucción MOV**

**NOTA**: Considerar BX=0300H SI=0200H DS=1000H Y MATRIZ=1000H

##### **Tipo Instrucción Fuente Generación Dirección Destino**

BX AX

Registro

Registro

Registro MOV AX,BX

3AH BL

Datos

Registro

Inmediato MOV BL,3AH

##### 

##### AX DSx10H + DESPL. 11234H

10000 + 1234

Registro

Memoria

Directo MOV 1234H,AX

AX DSx10H + (BX) 10300H

10000H + 0300H

Registro

Memori

Indirecto por MOV[BX],AX

Registro

AX DSx10H + (BX) + (SI) 10500H

10000H + 0300H+0200H

Registro

MemoriA

Base más MOV[BX+SI],AX

Indice

AX DSx10H + (BX) + 4 10304H

Relativo por MOV [BX+4],AX 10000 + 0300H + 4

Registro

Memori

Registro

DSx10H + ARRAY + (BX) + (SI) 11500H

AX 10000H + 1000H + 0300H + 0200H

Registro

Memori

Relativo a la MOV ARRAY[BX+SI],AX

Base más Indice

**Formato de las Instrucciones y Tipos de Datos**

1. **Instrucciones sin Operandos**

Se usa 1 byte para el código de operación.

Ej. CLI (borrar señalizador de interrupción)

1. **Instrucciones con un solo Operando**

###### Se utiliza un byte para el código de operación y un byte para el operando.

Ej. JE 05H (si el señalizador Z=1, salta 5 posiciones el IP)

1. **Instrucciones con dos Operandos**

###### Es el formato característico de la mayor parte de las operaciones

15 10 9 8 7 6 5 4 3 2 1 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Código Op. | d | W | Mod. | Registro | R/m |

Los 6 bits de mayor peso indican el código de operación.

1. el destino se especifica en los campos mod.

y r/m

d =

1. el destino es un registro especificado en el

campo mod

w especifica la longitud del operando. Si

0 indica que la long. de los operandos es de 8

bits.

w =

1. indica que la long. de los operandos es de 16 bits.

Mod. junto con r/m indica si uno de los operandos es un registro o una posición de memoria.

mod = 00 indica que el operando está en memoria y el

desplazamiento es 0.

mod = 01 el operando está en memoria y el desplazamiento es un valor de 8 bits, que para convertirse en 16, se extiende el bit de signo, o sea el 8vo. se repite en los 8 bits de más peso.

mod = 10 el operando está en memoria y el desplazamiento se expresa con 16 bits.

Mod = 11 indica que el operando está en un registro.

Reg. Consta de 3 bits que definen que definen el registro de propósito gral. que interviene en la instrucción.

000 AX o AL, según el operando sea de 16 o de 8 bits.

###### 001 CX o CL “ “ “

010 DX o DL “ “ “

###### 011 BX o BL “ “ “

1. SP o AH “ “ “
2. BP o CH “ “ “
3. SI o DH “ “ “
4. DI o BH “ “ “

R/m = 000 EA = (BX) + (SI) + DESPL (basado + indexado)

R/m = 001 EA = (BX) + (DI) + DESPL “

###### R/m = 010 EA = (BP) + (DI) + DESPL “

R/m = 011 EA = (BP) + (DI) + DESPL “

R/m = 100 EA = 0 + (SI) + DESPL indexado

R/m = 101 EA = 0 + (DI) + DESPL indexado

R/m = 110 EA = (BP) + 0 + DESPL basado

R/m = 111 EA = (BX) + 0 + DESPL basado

#### Modos de direccionamiento de 16 bits de memoria RAM

|  |  |
| --- | --- |
| **Código** | **Función** |
| 000 | DS:[BX+SI] |
| 001 | DS:[BX+DI] |
| 010 | SS:[BP+SI] |
| 011 | SS:[BP+DI] |
| 100 | DS:[SI] |
| 101 | DS:[DI] |
| 110 | SS:[BP] |
| 111 | DS:[BX] |