

LABORATORIO DE SISTEMAS DIGITALES CON MICROPROCESADORES

Modos de Direccionamiento

¿Qué son los Modos de Direcccionamiento?

- ❑ Los **Modos de Direcccionamiento** son las diversas formas con las que se puede indicar a un μP dónde encontrar o depositar un dato, en una instrucción. Identifica los operandos de la operación a realizar, fuente y destino, de los datos sobre los que se operará.
 - ❑ Los operandos se pueden especificar por: un registro de la CPU, una localidad de memoria, un puerto de Entrada/Salida o un dato inmediato.
 - ❑ Existen dos grupos principales de Modos de Direcccionamiento, según los operandos se encuentren en registros o en memoria serán: modos de *Registro* o modos de *Memoria*.
 - ❑ La dirección física de memoria o registro donde se encuentra realmente el operando se llama *Dirección Efectiva*, y se representa $\langle ea \rangle$. Según el modo de direccionamiento utilizado, la CPU tendrá que realizar unos cálculos distintos hasta obtener el valor de dicha dirección efectiva.
 - ❑ Las instrucciones pueden contener hasta dos operandos que se denominan *Operando Fuente*, el del lado derecho, y *Operando Destino*, el del lado izquierdo. En estos casos, cada operando tendrá su propio modo de direccionamiento.
-

Direcccionamiento inmediato

- ❑ Transfiere un byte o palabra de datos inmediato hacia el operando destino. Este modo es usado para inicializar registros o localidades de memoria y para operar sobre ellos con valores constantes de datos.
 - ❑ Ejemplo: `MOV AX,0ABCDH`
`MOV BL,12H`
 - ❑ Las instrucciones que usan el modo de direccionamiento inmediato obtienen el dato como parte de la instrucción.
`B8 00 10 MOV AX,1000H`
 - ❑ Este modo no opera con registros de segmento, por lo que **no se puede cargar un registro de segmento de manera inmediata.**
-

Direccionamiento por Registro

- ❑ Transfiere un byte o palabra desde un registro fuente hasta un registro destino.
- ❑ Ej. MOV AX,CX
- ❑ INC BX
- ❑ El operando no requiere ninguna referencia de memoria.

Nota: Los dos operandos no pueden ser registros de segmento.

- ❑ El registro de segmento de código (CS) nunca puede utilizarse como destino.
 - ❑ No se permite el acceso entre registros de segmento ni de distintos tamaños.
-

Direccionamiento Directo

- ❑ Transfiere un byte o palabra, contenido en una localidad de memoria en el segmento DS, a un registro de 8 o 16 bits. La localidad de memoria puede ser el operando fuente o destino.
 - ❑ Ej: **MOV AL,[1234H]**
 - ❑ En el modo de direccionamiento directo, la dirección de memoria se proporciona directamente como parte de la instrucción (Puede ser a través de etiquetas, en las cuales el programador no necesita conocer la dirección numérica)
 - ❑ **MOV AH, MEMBDS**
 - ❑ 8A 00 10 **MOV AH,[MEMBDS]** AH←[1000H]
-

Direccionamiento Indirecto

- El modo de direccionamiento directo se usa para acceder localidades de memoria de manera no frecuente. Sin embargo cuando una localidad de memoria debe ser leída o escrita varias veces dentro de un programa, la búsqueda repetida de la dirección lógica hace este modo ineficiente. El modo de **direccionamiento indirecto** resuelve este problema almacenando esta dirección de memoria en un registro base (BX, BP) o un registro índice (SI o DI)
- MOV [DI],BH
- MOV [BP],DL
- 8B 04 MOV AX, [SI] $AL \leftarrow [SI]; AH \leftarrow [SI+1]$
- FF 25 JMP [DI] $IP \leftarrow [DI+1:DI]$
- FE 46 00 INC BYTE PTR[BX] $[BP] \leftarrow [BP] + 1$
- FF 0F DEC WORD PTR[BX] $[BX+1:BX] \leftarrow [BX+1:BX] - 1$

Direccionamiento Base más Índice

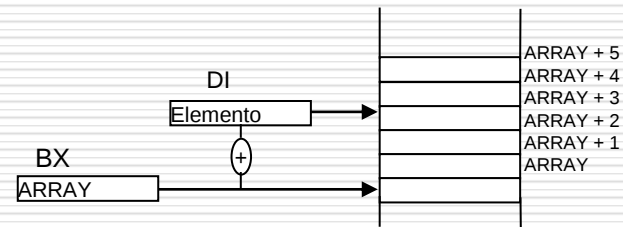
- ❑ Transfiere un byte o palabra entre un registro y una localidad de memoria direccionada por la suma de un registro base más un registro índice.
- ❑ Este modo es usado para el acceso a tablas.
- ❑ Ej. MOV AX,[BX+DI]
- ❑ En muchos casos, el registro base retiene la dirección de inicio de un arreglo de memoria, y el registro índice retiene la posición relativa de un dato en un arreglo.

8B 00	MOV AX,[BX+SI]	AH←[BX+SI+1], AL←[BX+SI];
FF 21	JMP [BX+DI]	IP←[BX+DI+1:BX+DI];
FE 02	INC BYTE PTR[BP+SI]	[BP+SI]←[BP+SI]+1 ;
FF 0B	DEC WORD PTR[BP+DI]	[BP+DI+1:BP+DI]←[BP+DI+1:BP+DI]-1

Ejemplo Direcccionamiento Base más Índice

- Ej. Si se desea direccionar los elementos en un arreglo de datos localizados en el segmento de datos en la localidad ARRAY. Se requiere cargar BX con la dirección ARRAY y DI con el número del elemento del arreglo que se desea acceder.

- `MOV BX, OFFSET ARRAY`
- `MOV DI, 3`
- `MOV AL, [BX+DI]`

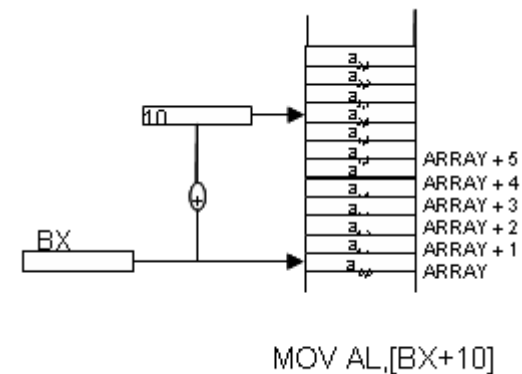


`MOV AL,[BX+DI]`

Direcccionamiento Relativo a Registro

- ❑ Transfiere un byte o palabra entre un registro y una localidad de memoria direccionada por **un registro** base o un registro índice **más un desplazamiento**.
- ❑ Si la localidad de memoria se direcciona por la **suma de un registro base y un desplazamiento**, también se conoce como **direccionamiento basado**.
- ❑ Ej. `MOV AX,[BX+10H]`
- ❑ Si la localidad de memoria se direcciona por la **suma de un registro índice y un desplazamiento**, también se conoce como **direccionamiento indexado**.
- ❑ Ej. `MOV AX,[SI+500H]`

- `MOV AX,[BX+4]`
- `MOV AX,ARRAY [SI]`
- `MOV LIST[BP],CL`
- `MOV ARRAY[DI],AL`



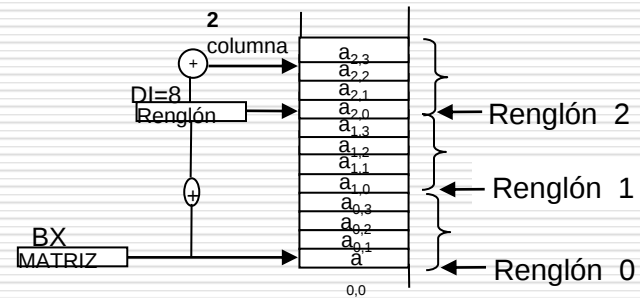
Direccionamiento Relativo Base más Índice (1)

- ❑ Transfiere un byte o palabra entre un registro y la localidad de memoria direccionada por un registro base más un registro índice más un desplazamiento.

 - ❑ Ej. `MOV AX, [BX+SI+100H]`
 - ❑ `MOV AX, ARRAY[BX+DI]`
 - ❑ `MOV LIST[BP+DI],CL`
-

Direccionamiento Relativo Base más Índice (2)

Este tipo de direccionamiento es usado comúnmente para direccionar arreglos de datos en memoria de dos dimensiones (matrices).



`MOV AL,[BX+DI+2]`

Selecciona elemento $a_{2,2}$

MOV BX, OFFSET MATRIZ

MOV DI,8

MOV AL,[BX+DI+2]

$$\begin{pmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \end{pmatrix}$$

Direccionamiento de string

- ❑ En computación un string (cadena) es una secuencia de bytes o palabras almacenadas en memoria. Una tabla de datos es un ejemplo de string. Debido a su importancia el 8086 tiene algunas instrucciones diseñadas específicamente para manejar strings (cadenas) de caracteres.
 - ❑ Estas instrucciones tienen un modo de direccionamiento especial y usan a DS:SI para apuntar al string fuente y a ES:DI para apuntar al string destino.
 - ❑ MOVSB mueve el byte del dato fuente a la localidad destino. SI y DI se incrementan o decrementan automáticamente dependiendo del valor de la bandera D.
-

Instrucciones de cadena

Instrucciones de Cadena (string)					
Mnemónico general Op-code Operando	Código Objeto	Mnemónico	Segmento de memoria	Operación simbólica	Descripción
STOSB	AA	STOSB	Extra	ES:[DI] ← AL Si DF=0, DI ← DI+1 Si DF=1, DI ← DI-1	Transfiere un byte o palabra del registro AL o AX a la cadena direccionada por DI en el segmento extra; Si DF=0, incrementa DI, de lo contrario decrementa DI; las banderas no son afectadas.
STOSW	AB	STOSW	Extra	ES:[DI] ← AL ES:[DI+1] ← AH Si DF=0, DI ← DI+2 Si DF=1, DI ← DI-2	
LODSB	AC	LODSB	Datos	AL ← DS:[SI] Si DF=0, SI ← SI+1 Si DF=1, SI ← SI-1	Transfiere un byte o palabra de la cadena direccionada por SI en el segmento de datos al registro AL o AX; Si DF=0, incrementa SI, de lo contrario decrementa SI; las banderas no son afectadas.
LODSW	AD	LODSW	Datos	AL ← DS:[SI] AH ← DS:[SI+1] Si DF=0, SI ← SI+2 Si DF=1, SI ← SI-2	

Tabla de Modos de Direcccionamiento

Direcccionamiento	Cod. Ob.	Mnemónico	Segmento	Operación simbólica
Inmediato	B8 00 10	MOV AX,1000H	Código	AH←10H; AL ←00
Registro	8B D1	MOV DX,CX	Dentro del CPU	DX←CX
Directo	8A 00 10	MOV AH,[MEMBDS]	Datos	AH←[1000H]
Indirecto a registro	8B 04	MOV AX, [SI]	Datos	AL←[SI]; AH←[SI+1]
	FF 25	JMP [DI]	Datos	IP←[DI+1:DI]
	FE 46 00	INC BYTE PTR[BP]	Stack	[BP]←[BP] + 1
	FF 0F	DEC WORD PTR[BX]	Datos	[BX+1:BX]←[BX+1:BX]-1
Indexado	8B 44 06	MOV AX,[SI+6]	Datos	AL←[SI+6]; AH←[SI+7]
	FF 65 06	JMP [DI+6]	Datos	IP ← [DI+7:DI+6]
Basado	8B 46 02	MOV AX,[BP+2]	Stack	AL←[BP+2]; AH←[BP+3]
	FF 67 02	JMP [BX+2] ^c	Datos	IP←[BX+3:BX+2]
Base mas índice	8B 00	MOV AX,[BX+SI]	Datos	AL←[BX+SI]; AH←[BX+SI+1]
	FF 21	JMP [BX+DI]	Datos	IP←[BX+DI+1:BX+DI]
	FE 02	INC BYTE PTR[BP+SI]	Stack	[BP+SI]←[BP+SI]+1
	FF 0B	DEC WORD PTR[BP+DI]	Stack	[BP+DI+1:BP+DI]←[BP+DI+1:BP+DI]-1
Relativo base mas índice	8B 40 05	MOV AX,[BX+SI+5]	Datos	AL←[BX+SI+5]; AH←[BX+SI+6]
	FF 61 05	JMP [BX+DI+5]	Datos	IP←[BX+DI+6:BX+DI+5]
	FE 42 05	INC BYTE PTR[BP+SI+5]	Stack	[BP+SI+5]←[BP+SI+5]+1
	FF 4B 05	DEC WORD PTR[BP+DI+5]	Stack	[BP+DI+6:BP+DI+5]←[BP+DI+6:BP+DI+5]-1
String	A4	MOVSB	Extra, Datos	ES:[DI]←DS:[SI] Si DF=0, entonces SI←SI+1; DI←DI+1 Si DF=1, entonces SI←SI-1; DI←DI-1