

Modos de direccionamiento del 8086/8088

Estos procesadores tienen 27 modos de direccionamiento (una cantidad bastante más grande que los microprocesadores anteriores) o reglas para localizar un operando de una instrucción. Tres de ellos son comunes a microprocesadores anteriores:

- 1) **direccionamiento inherente** (el operando está implícito en la instrucción, por ejemplo, en la multiplicación uno de los operandos siempre es AX).
- 2) **direccionamiento a registro** (el operando es un registro del microprocesador)
- 3) **direccionamiento inmediato** (el operando es un número que se encuentra en la misma instrucción),

El resto de los modos sirve para localizar un operando en memoria. Para facilitar la explicación de estos modos, se pueden resumir de la siguiente manera:

Deben sumarse cuatro cantidades:

- dirección de segmento
- dirección base
- una cantidad índice
- un desplazamiento.

La dirección de segmento se almacena en el registro de segmento (DS, ES, SS o CS). En la próxima sección se indica la forma en que se hace esto. Por ahora basta con saber que el contenido del registro de segmento se multiplica por 16 antes de utilizarse para obtener la dirección real. El registro de segmentación siempre se usa para referenciar a memoria.

La base se almacena en el registro base (BX o BP). El índice se almacena en el registro índice (SI o DI). Cualquiera de estas dos cantidades, la suma de las dos o ninguna, pueden utilizarse para calcular la dirección real, pero no pueden sumarse dos bases o dos índices. **Los registros restantes (AX, CX, DX y SP) no pueden utilizarse para direccionamiento indirecto.**

Además del segmento, base e índice, se usa un desplazamiento de 16 bits, 8 bits o 0 bits (sin desplazamiento). Ésta es una cantidad estática que se fija durante el ensamblado.

Todo esto genera los 24 modos de direccionamiento a memoria que se ven a continuación:

Directo:

- 4) [desp16]. Ejemplo: MOV AX, [2000h]

Indirecto:

- 5) [BX], Ejemplo: MOV AX, [BX]
6) [DI],
7) [SI].

Basado:

- 8) [BX+desp8], Ejemplo: MOV AX, [BX+30h]
9) [BP+desp8],
10) [BX+desp16],
11) [BP+desp16].

Indexado:

- 12) [SI+desp8],
13) [DI+desp8],
14) [SI+desp16], Ejemplo: MOV AX, [SI+4000h]
15) [DI+desp16].

Basado-indexado:

- 16) [BX+SI],
17) [BX+DI], Ejemplo: MOV AX, [BX+DI]
18) [BP+SI],
19) [BP+DI].

Basado-indexado con desplazamiento:

- 20) [BX+SI+desp8], Ejemplo: MOV AX, [BX+SI+60h]
- 21) [BX+DI+desp8],
- 22) [BP+SI+desp8],
- 23) [BP+DI+desp8],
- 24) [BX+SI+desp16],
- 25) [BX+DI+desp16],
- 26) [BP+SI+desp16], Ejemplo: MOV AX, [BP+SI+7000h]
- 27) [BP+DI+desp16].

Aquí desp8 indica desplazamiento de 8 bits y desp16 indica desplazamiento de 16 bits. Otras combinaciones no están implementadas en la CPU y generarán error al querer ensamblar, por ejemplo, ADD CL,[DX+SI].

El ensamblador genera el tipo de desplazamiento más apropiado (0, 8 ó 16 bits) dependiendo del valor que tenga la constante: si vale cero se utiliza el primer caso, si vale entre -128 y 127 se utiliza el segundo, y en otro caso se utiliza el tercero. Nótese que [BP] sin desplazamiento no existe. Al ensamblar una instrucción como, por ejemplo, MOV AL,[BP], se generará un desplazamiento de 8 bits con valor cero. Esta instrucción ocupa tres bytes, mientras que MOV AL,[SI] ocupa dos, porque no necesita el desplazamiento.

Estos modos de direccionamiento producen algunos inconvenientes en el 8086/8088. La CPU gasta tiempo calculando una dirección compuesta de varias cantidades. Principalmente esto se debe al hecho de que el cálculo de direcciones está programado en microcódigo (dentro de la CROM del sistema de control de la unidad de ejecución). En las siguientes versiones (a partir del 80186/80188) estos cálculos están cableados en la máquina y, por lo tanto, cuesta mucho menos tiempo el realizarlos.

Veamos un ejemplo: MOV AL, ES:[BX+SI+6]. En este caso, el operando de la izquierda tiene direccionamiento a registro mientras que el de la derecha indica una posición de memoria. Supongamos que los valores actuales de los registros sean: ES = 3200h, BX = 200h, SI = 38h. Entonces, la dirección real de memoria será: $ES * 10h + BX + SI + 6 = 3200h * 10h + 200h + 38h + 6 = 3223Eh$

Estructura de memoria de segmentación: el 8086/8088 usa un esquema de segmentación, para acceder correctamente a un megabyte completo de memoria, con referencias de direcciones de sólo 16 bits.

Veamos cómo funciona. Cualquier dirección tiene dos partes, cada una de las cuales es una cantidad de 16 bits. Una parte es la dirección de segmento y la otra es el offset. A su vez el offset se compone de varias partes: un desplazamiento (un número fijo), una base (almacenada en el registro base) y un índice (almacenado en el registro índice). La dirección de segmento se almacena en uno de los cuatro registros de segmento (CS, DS, ES, SS). El procesador usa estas dos cantidades de 16 bits para calcular la dirección real de 20 bits, según la siguiente fórmula:

$$\text{Dirección real} = 16 * (\text{dirección del segmento}) + \text{offset}$$

Hay dos registros de segmento que tienen usos especiales: el microprocesador utiliza el registro CS (con el offset almacenado en el puntero de instrucción IP) cada vez que se debe acceder a un byte de instrucción de programa, mientras que las instrucciones que utilizan la pila (llamados a procedimientos, retornos, interrupciones y las instrucciones PUSH y POP) siempre utilizan el registro de segmento SS (con el offset almacenado en el registro puntero de pila SP). De ahí los nombres que toman: CS es el segmento de código mientras que SS es el registro segmento de pila.

Para acceder a datos en la memoria se puede utilizar cualquiera de los cuatro registros de segmento, pero uno de ellos provoca que la instrucción ocupe un byte menos de memoria: es el llamado segmento por defecto, por lo que en lo posible hay que tratar de usar dicho segmento para direccionar datos. Este segmento es el DS (registro de segmento de datos) para todos los casos excepto cuando se utiliza el registro base BP. En este caso el segmento por defecto es SS.

Si se utiliza otro registro, el ensamblador genera un byte de prefijo correspondiente al segmento antes de la instrucción: CS → 2Eh, DS → 3Eh, ES → 26h y SS → 36h. El uso de estos diferentes segmentos significa que hay áreas de trabajo separadas para el programa, pila y los datos. Cada área tiene un tamaño máximo de 64 KBytes.

Adaptado de: <http://www.alpertron.com.ar/8088.HTM>