

### Modo de Uso de la Memoria

El 8086 tiene un bus de direcciones de 20 bits para poder direccionar 1Mb. Pero para disminuir la longitud de las instrucciones limitando el campo de direcciones a 16 bits, se divide la memoria en 4 **segmentos**. Cada segmento tiene una capacidad máxima de 64Kb.

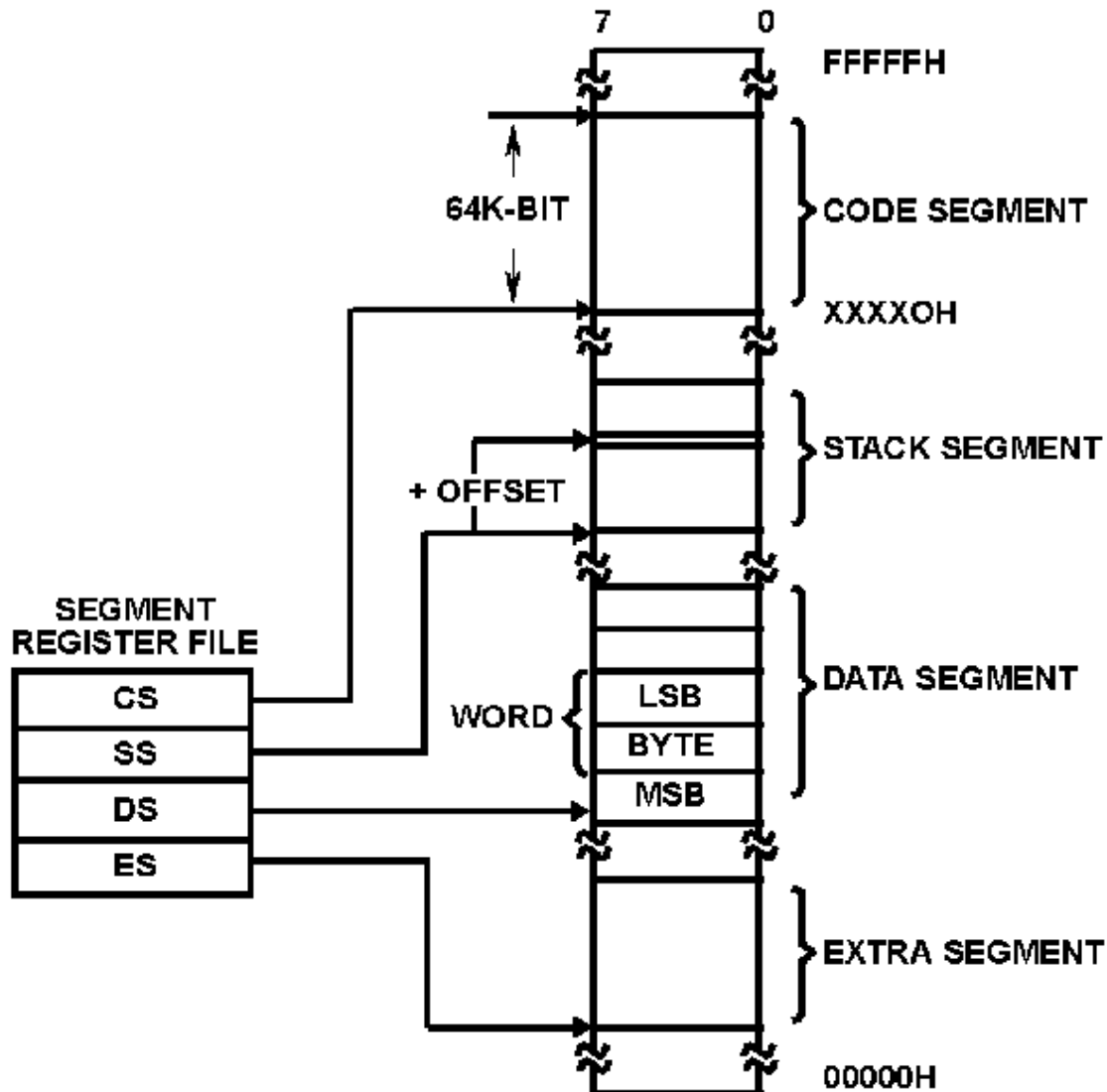


FIGURE 1. MEMORY ORGANIZATION

El cálculo de la dirección física depende de dos registros:

- El registro de segmento que puede estar indicado implícita o explícitamente en la instrucción, al cual se le agregan los últimos cuatro ceros.
- El desplazamiento indicado en el campo de dirección de la instrucción.

Tipo de acceso a memoria	Registro de segmento por defecto	Otro posible registro de segmento	Registro que contiene el desplazamiento
Búsqueda de la instrucción	CS	Ninguno	IP
Operación sobre la pila	SS	Ninguno	SP

Acceso al dato fuente o destino de una cadena	ES	Ninguno	DI
Acceso a un dato	DS	CS, ES, SS	Dir Efectiva
Acceso a un dato con BP como registro base	SS	CS, DS, ES	Dir Efectiva

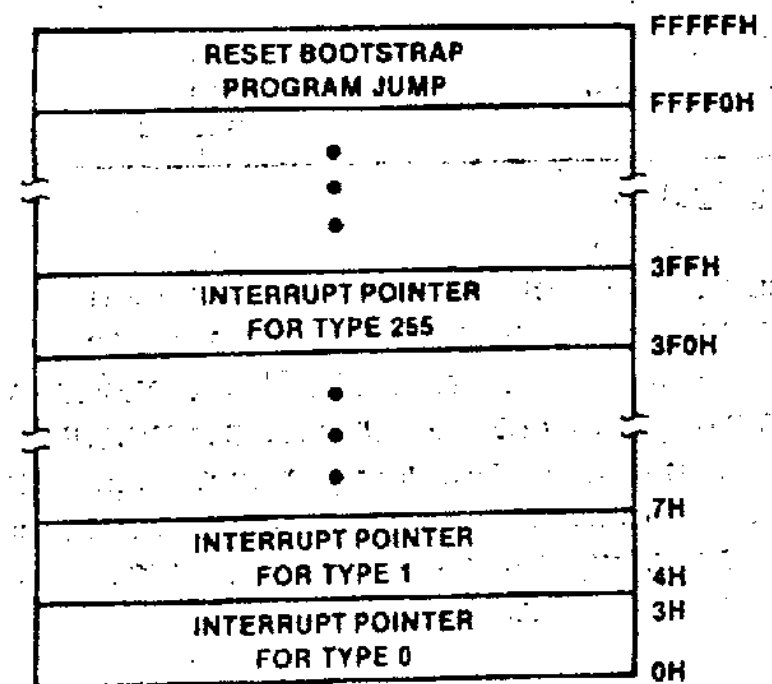
Cuando el operando es una palabra (16 bits), se pueden poner en direcciones pares o impares linderas. Tanto para direcciones como para datos, el byte menos significativo de la palabra se guarda en la dirección más chica y el byte mas significativo en la dirección superior siguiente.

**La BIU ejecutará automáticamente dos ciclos de búsqueda o escritura en caso de operandos de 16-bits.**

Los segmentos de memoria pueden estar separados, solapados o superpuestos.

**Relocalización:** cuando un programa no contiene instrucciones de carga o modificación de los registros de segmento ni instrucciones de salto a un segmento distinto del segmento de programa, puede ser relocalizado. Para ello es necesario que todos los desplazamientos sean relativos a las direcciones bases contenidas en los registros de segmento.

Algunas direcciones de memoria están reservadas para los vectores de interrupción y el de RESET.



Tipo de referencia a memoria	Registro de segmento usado	Regla de seleccione de segmento
Instrucciones	CS (Code)	Automática con todas las instrucciones
Pila	SS (Stack)	Todos los pushes y pops a la pila. Referencias a memoria relativas al BP

		excepto en referencias de datos
Local Data	DS (Data)	Referencias a datos cuando: relativo a la pila, operación con cadenas
External (Global) Data	ES (Extra)	Uso con cadenas

**La pila** se desarrolla hacia direcciones decrecientes de memoria. El SP contiene la dirección de la última palabra de la pila. La pila trabaja con palabras y no con bytes.

### **Modo Mínimo y Modo Máximo**

- **Modo Mínimo:** todas las señales para la memoria y E/S se generan dentro del 8088.
- **Modo Máximo:** algunas de las señales de control deben decodificarse externamente, normalmente se utiliza en sistemas grandes con coprocesadores.

### **Direcciones de I/O**

Existe un máximo de 64K de registros de I/O. Las direcciones de I/O aparecen de la misma forma que las de memoria en las líneas del bus de direcciones de A0 a A15. En las operaciones de I/O las líneas de direcciones de A16 a A19 son cero.

El 8088 tiene instrucciones especiales que direccionan uno o dos de estos puertos.

La diferencia entre dirección de memoria o de I/O se hace notar mediante la línea IO/M.

### **Modos de Direccionamiento**

Para describir los modos de direccionamiento usaremos la instrucción MOV destino, origen.

### **Modos de Direccionamiento de Datos**

#### **Direccionamiento por Registro**

Transfiere el contenido de un registro a otro del mismo tamaño. El único MOV que no se permite es el que se hace entre un registro de segmento y otro. Tampoco se puede modificar el registro de segmento de código. Es rápido.

MOV AX,BX (AX <- BX)

#### **Direccionamiento Inmediato**

Transfiere el dato que sigue al código de operación al registro especificado. Es muy rápido porque se hace con información de la cola.

MOV AX,3456H (AX <- 3456H)

#### **Direccionamiento Directo**

Se suma el desplazamiento dado al segmento de datos.

- Direccionamiento directo que solo se aplica a una MOV entre una localidad de memoria y AL ó AX. Son instrucciones muy comunes y tienen 3 bytes de longitud
- Direccionamiento directo por desplazamiento: ídem anterior pero de 4 bytes  
MOV AX,1234H (AX <- [DS x 10H + 1234H])

### **Direccionamiento indirecto por registro**

Transfiere un dato entre un registro y una dirección de memoria direccionada por un registro índice o base (BP, EX, DI, SI)

MOV AX,[BX] ( $AX \leftarrow [DS \times 10H + BX]$ )

Cuando se utiliza este modo de direccionamiento y se emplean los registros BX, DI o SI, se asume que el dato está en el segmento de datos.

Si se usa BP se toma que el dato está en el segmento de pila.

### ***Direccionamiento Base Más Índice***

Direcciona en forma indirecta los datos de memoria. Se usa un registro base (BP o BX) más un registro índice (DI o SI). Si se usa BP se toma como dato de pila

MOV AX,[BX+SI] ( $AX \leftarrow [BX+SI+DS \times 10H]$ )

Se usa para direccionar tablas en memoria.

### ***Direccionamiento Relativo por Registro***

Para ubicar el dato se agrega un desplazamiento al contenido del índice (BP, BX, DI o SI). También BP direcciona en pila y los otros en datos.

MOV [BX+4],AX ( $AX \rightarrow BX+4+DS \times 10H$ )

### ***Direccionamiento Relativo Base Más Índice***

Transfiere entre un registro y una dirección dada por un registro base más índice más desplazamiento.

MOV ARRAY[BX+SI],AX ( $BX+ARRAY+SI+DS \times 10H \leftarrow AX$ )

Se usa para direccionar un arreglo bidireccional.

## **Modos de Direccionamiento de Memoria de Programa**

Usados en instrucciones de salto (JMP) y llamada (CALL)

### ***Direccionamiento Directo a la Memoria del Program***

Se almacena la dirección con el código de operación. Se usa para saltos largos entre segmentos.

JMP 10000H CS  $\leftarrow$  10000H, IP  $\leftarrow$  0000H

### ***Direccionamiento Relativo de Memoria de Programa***

Se modifica el contenido del IP para continuar la ejecución en otro lado, según el valor especificado luego del código de operación. Se usa para saltos cortos intrasegmento.

JMP 2 (IP  $\leftarrow$  IP+2)

### ***Direccionamiento Indirecto de Memoria de Programa***

La dirección está contenida en un registro relativo.

JMP [BX] (IP  $\leftarrow$  BX)

## **Direccionamiento a la Pila de Memoria**

Los datos se ponen en la pila por PUSH y se recuperan por POP.

La instrucción CALL aprovecha la pila para salvar la dirección de retorno y la RET para recuperarla.

La pila se mantiene con 2 registros:

- SP: Stack Pointer
- SS: Stack Segment
- PUSH: El SP-1 apunta a donde se almacenará la parte alta del próximo dato. La parte baja se almacena en SP-2. Luego decrecienta 2.

- POP: LA parte alta del dato se recupera de SP, la parte baja se recupera de SP+1. Luego incrementa 2.

NOTA: PUSH A y POP A salva o recupera todos los registros (excepto los de segmento) en la pila.

### **Direccionamiento Implícito**

El operando está implícito. HLT, LOCK, WAIT, DAA.