

## EL MICROPROCESADOR 8086

Uno de los propósitos de los lenguajes de alto nivel es de ocultarle al programador los detalles del hardware de la computadora en la que se ejecutarán los programas que éste escribe. Sin embargo al programar en un lenguaje de bajo nivel o ensamblador se debe tener un conocimiento mayor del hardware. Dos de los aspectos que debe conocer el programador de ensamblador sobre la computadora en que se va a ejecutar sus programas es la organización de la memoria y el modelo de programación de su microprocesador.

### Organización de la Memoria

Podemos visualizar a la memoria principal de una computadora como una colección de localidades de memoria cada una de un byte. Cada localidad de memoria tiene su propia dirección. Dado que cada dirección es un número entero, hay una relación directa entre el número de localidades de memoria y el tamaño que debe tener una dirección en bits. Una dirección de  $N$  bits nos permite direccionar  $2^N$  direcciones diferentes.

En la memoria principal de una computadora se almacenan tanto los programas como los datos con que operan esos programas. Al estar ejecutando un programa, se requiere calcular las direcciones en las que están almacenadas las instrucciones y datos. El cálculo de las direcciones de memoria depende de cómo el procesador visualiza u organiza la memoria. Dos de las formas en las que el microprocesador visualiza u organiza la memoria en una computadora son las llamadas memoria lineal y memoria segmentada.

#### Memoria Lineal

El esquema de memoria lineal es el más simple de todos ya que bajo éste el microprocesador ve a la memoria como un arreglo lineal de localidades de memoria de un byte y en el que las direcciones de las localidades de memoria son los índices de los elementos del arreglo. El esquema de memoria lineal es usado por lo general en microprocesadores con capacidad de direccionamiento reducido, por lo general aquellos en los que la memoria no excede a los 64 KB.

#### Memoria Segmentada

En el esquema de memoria segmentada el microprocesador visualiza a la memoria como dividida en segmentos de memoria y a cada segmento lo visualiza como un arreglo lineal de localidades de memoria de un byte. Cada segmento tiene una dirección llamada dirección de segmento. La dirección de una localidad de memoria se forma a partir de la **dirección del segmento** que contiene a la localidad de memoria y al índice de la localidad de memoria dentro del segmento. A este índice se le conoce como **desplazamiento**.

El esquema de memoria segmentada es usado por lo general en microprocesadores con capacidad de direccionamiento que excede a los 64 KB. Uno de estos procesadores es el procesador 8086 de Intel.

#### Segmentación de la memoria en el Microprocesador 8086

El CPU empleado en las primeras computadoras IBM PC y compatibles, el microprocesador 8086 de Intel, nos permite direccionar 1 MB (MegaByte) =  $2^{20} = 1,048,576$  bytes. Los procesadores más modernos de Intel de la familia 80x86 (80286, 80386, 80486 y Pentium) empleados en las computadoras PC pueden direccionar un mayor número de localidades (hasta 16 MB =  $2^{24} = 16,777,216$  bytes para el 80286 y hasta 4 GB =  $2^{32} = 4,294,967,296$  bytes para el 80386, 80486 y Pentium) pero para mantener compatibilidad con el microprocesador 8086 tienen dos formas de operación: una que imita al microprocesador 8086 llamado **modo real** y otro que explota todo el poder del procesador llamado **modo protegido**. El sistema operativo MSDOS a fin de mantener compatibilidad con el software existente, sólo permite la operación de los microprocesadores en modo real y para fines prácticos una computadora PC bajo ese sistema operativo se comportará como si tuviese un procesador 8086 (aunque más rápido).

En el microprocesador 8086 (o en los microprocesadores más modernos de la familia 80x86 operando en el modo real) visualiza la memoria como dividida en segmentos los cuales tienen las siguientes características:

- Los segmentos no son divisiones físicas. Un segmento es una ventana lógica a través de la cual el programa visualiza porciones de la memoria.
- Los segmentos empiezan cada 16 bytes. Como a un bloque de memoria de 16 bytes se le conoce como **párrafo**, los segmentos empiezan a cada párrafo.
- Hay hasta 65,536 segmentos ( $1,048,576/16 = 65,536$ ).
- El primer segmento es el segmento 0, el siguiente es el segmento 1, etc. El número del segmento se conoce como la **dirección del segmento**.
- La **dirección real**, también llamada **dirección efectiva**, en la que empieza un segmento se obtiene multiplicando la dirección del segmento por 16.
- Los segmentos pueden ser tan pequeños como 16 bytes y tan grandes como 64 KB = 65,536 bytes.

## Modelo de programación del Microprocesador 8086

Los registros de un microprocesador son componentes dentro del microprocesador que nos permiten almacenar datos. Estos datos pueden representar valores sobre los cuales se van a realizar operaciones, resultados de las operaciones, direcciones de localidades de memoria donde se encuentran datos e instrucciones, direcciones de los dispositivos de entrada/salida sobre los que deseamos escribir o leer, o los datos a escribir o leer de esos dispositivos.

El número, tamaño y uso de los registros de un microprocesador así como su conjunto de instrucciones determina la eficiencia con que el microprocesador realiza una tarea. A la descripción del número, tamaño y uso de los registros de un microprocesador se le conoce como el modelo de programación del microprocesador.

Los microprocesadores más poderosos de la familia del 8086 tienen otros registros adicionales pero no forman parte del estudio de este curso. Los registros del 8086 pueden clasificarse en tres grupos: registros de uso general, registros apuntadores y de índice y registros de segmentos. Adicionalmente tienen un registro de banderas que señala las condiciones respecto al funcionamiento de la unidad aritmética y lógica (ALU).

### Registros de propósito general

Los registros de propósito general, como su nombre lo indica, se utilizan en la forma en que lo desee el programador. Cada uno de estos registros se puede direccionar como un registro de 16 bits (AX, BX, CX, DX) o como un registro de 8 bits (AH, AL, BH, BL, CH, CL, DH, DL). Cada uno de los registros de 16 bits está formado por la concatenación de dos registros de 8 bits: AX = AH:AL, BX = BH:BL, CX = CH:CL y DX = DH:DL, donde el bit 0 del registro AH es el bit 8 del registro AX, etc.

Adicionalmente, algunas de las instrucciones del microprocesador emplean los registros de propósito general para tareas específicas. Por esta razón, a cada uno se le da su nombre (Acumulador, Base, Contador y Datos). Las funciones primarias de los registros de propósito general incluyen:

- AX, AH, AL (Acumulador): a menudo conserva el resultado temporal después de una operación aritmética o lógica.
- BX, BH, BL (Base): Se utiliza para guardar la dirección base de listas de datos en la memoria.
- CX, CH, CL (Contador): Contiene el conteo para ciertas instrucciones de corrimientos y rotaciones, de iteraciones en el ciclo loop y operaciones repetidas de cadenas.
- DX, DH, DL (Datos): Contiene la parte más significativa de un producto después de una multiplicación; la parte más significativa del dividendo antes de la división.

### Registros apuntadores y de índices

Aunque los registros apuntadores y de índices (a excepción del registro apuntador de instrucciones) son también de uso general, se utilizan principalmente para formar la dirección efectiva o real de una localidad de memoria. Los registros apuntadores y de índice contienen el desplazamiento con respecto a un segmento de un dato o una instrucción. Las funciones primarias de los registros apuntadores y de índice incluyen:

- SP (Apuntador de pila): Asociado con el registro SS para operaciones con la pila.
- BP (Apuntador de base): Asociado con el registro SS para operaciones con la pila.
- SI (Índice fuente): Asociado con el registro DS para operaciones con cadenas.
- DI (Índice destino): Asociado con el registro ES para operaciones con cadenas.
- IP (Apuntador de instrucciones): contiene la dirección de la siguiente instrucción que va a ejecutar el microprocesador.

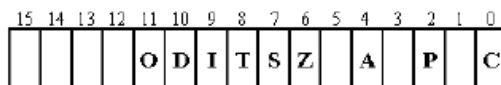
### Registros de segmentos

Los registros de segmentos se utilizan para formar la dirección efectiva o real de una localidad de memoria. Los registros de segmentos contienen la dirección de segmento de un dato o una instrucción. El código y los datos de un programa en ejecución se encuentran cada uno en uno o más segmentos de memoria. La pila del programa ocupa otro segmento. Los registros de segmento son:

- CS (Código): Tiene la dirección lógica del segmento en que se encuentra el código de un programa.
- DS (Datos): Tiene la dirección lógica del segmento en que se encuentran los datos estáticos de un programa.
- ES (Extra): Se utiliza en ciertas operaciones del microprocesador para el manejo de cadenas.
- SS (Pila): Tiene la dirección lógica del segmento en que se encuentran la pila del sistema.

## El registro de banderas

Cada bandera es un bit en el registro de banderas, también llamado registro de código de condiciones. En la siguiente figura se ilustra el registro de banderas del microprocesador 8086. Aunque el registro de banderas es de 16 bits sólo hay nueve banderas: O, D, I, T, S, Z, A, P, C. Los otros 7 bits no son usados.



Registro de banderas del microprocesador 8086

De estas nueve banderas, seis de ellas cambian después de ejecutar muchas de las instrucciones aritméticas y lógicas. Estas seis banderas son:

- C (Acarreo): Indica un acarreo después de una suma o un préstamo después de una resta.
- P (Paridad): Es un cero para una paridad impar y un 1 para una paridad par. La paridad es un conteo de unos expresada como un número par e impar.
- A (Acarreo auxiliar): Indica un acarreo después de una suma o un préstamo después de una resta del bit 3 al bit 4 en el resultado. Esta bandera sólo se utiliza en las operaciones **daa** y **das** para hacer ajustes.
- Z (Cero): Indica que el resultado de una operación aritmética o lógica es cero. Si Z = 1, el resultado es cero y si Z = 0, el resultado no es cero.
- S (Signo): Indica el signo aritmético del resultado después de una suma o una resta. Si S = 1, el resultado es negativo. Si S = 0, el resultado es positivo.
- O (Sobreflujo): Es una condición que ocurre cuando se suman o se restan números con signo. Un sobreflujo indica que el resultado ha excedido la capacidad de la máquina. Para operaciones sin signo esta bandera se ignora.

Las otras tres banderas se utilizan para controlar ciertas características del microprocesador. Esas tres banderas y su uso son:

- T (Trampa): Activa, si se pone a 1, o desactiva, si se pone a 0, el modo de ejecución paso a paso. Este modo es utilizado por los depuradores para ejecutar las instrucciones una a la vez y permitir observar el efecto de la instrucción sobre los registros y la memoria (modo debug).
- I (Interrupción): Habilita, si se pone a 1, o deshabilita, si se pone a 0, las interrupciones al microprocesador. El estado de esta bandera se controla con las instrucciones **sti** (habilitar interrupciones) y **cli** (desactivar las interrupciones).
- D (Dirección): Controla la dirección para mover o comparar cadenas (izquierda o derecha).

## Programa "Hello World"

Estudie este programa que le servirá más adelante para realizar un ejercicio práctico. Identifique las diferentes partes del código.

```
; Programa ejemplo: Hello world
; Las instrucciones resaltadas no pueden omitirse, sino pueden ocurrir errores inesperados
.MODEL SMALL          ; modelo de memoria pequeño
.STACK 64             ; segmento de stack
.DATA                 ; segmento de datos
; Definición de datos
CADENA DB 'Hello world !!$'
;-----
.CODE                 ; segmento de código
MAIN PROC FAR
    MOV AX,@data      ; inicializa segmento de datos
    MOV DS,AX         ; puede inicializarse con .STARTUP
    MOV AH, 09h       ; prepara los registros AH y DX
    LEA DX, CADENA    ; para mostrar mensaje en pantalla
    INT 21h           ; y llamar a la interrupción para mostrar el mensaje
    MOV AH, 4Ch       ; prepara el registro AH
    INT 21h           ; para llamar interrupción de salida al DOS
MAIN ENDP
END MAIN
```

## Programa

Tomando como base este ejemplo, modifíquelo para que sume dos números de 1 byte almacenados en memoria y muestre su resultado en pantalla. Investigue las instrucciones necesarias para sumar.

## Instrucciones

La computadora portátil se rota entre los estudiantes cada 7 minutos, cada uno tiene el siguiente rol:

- (1) Escribir código (primer nombre ordenado alfabéticamente)
- (2) Documentación y estilo (segundo nombre ordenado alfabéticamente)
- (3) Ensamblar y probar (tercer nombre ordenado alfabéticamente)

## Ayudas importantes

- Para definir números de 1 byte se utiliza la siguiente instrucción:

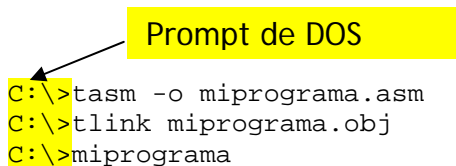
```
NUM1    DB    12    ; guarda un 12 (decimal)
```

- Para desplegar un carácter en pantalla utilice las siguientes instrucciones:

```
MOV AH, 02H    ; Petición de la función para desplegar
MOV DL, 'a'     ; caracter a desplegar (entre comillas o el
                ; valor ASCII equivalente)
INT 21H        ; llama a la interrupción para desplegar
```

- Pasos para escribir y ejecutar un programa en Assembler:
  1. **Editor:** Bloc de notas, Word Pad, Edit
  2. **Ensamblador:** tasm
    - Produce .obj
  3. **Enlazador (link):** tlink
    - Produce .exe ó .com
  4. **Ejecutar**
    - Nombre del programa con extensión .exe

Desde el Command Prompt de DOS:



```
C:\>tasm -o miprograma.asm
C:\>tlink miprograma.obj
C:\>miprograma
```

## Bibliografía

1. Abel, Peter. Lenguaje Ensamblador y Programación para PC IBM y Compatibles. Tercera Edición. Prentice-Hall Hispanoamericana, S. A. México. 1996.
2. Brey, Barry B. Los microprocesadores Intel: 8086/8088, 80186, 80286, 80386 y 80486. Arquitectura, programación e interfaces. Tercera Edición. Prentice-Hall Hispanoamericana, S. A. México. 1995.
3. Godfrey, J. Terry. Lenguaje Ensamblador para Microcomputadoras IBM para Principiantes y Avanzados. Prentice-Hall Hispanoamericana, S. A. México. 1991.
4. Hyde, Randall. The Art of Assembly Language Programming. Este libro se encuentra como una serie de documentos PDF en el siguiente servidor FTP: <ftp.cs.ucr.edu/pub/pc/ibmpcdir>
5. Morgan, Christopher L, y Waite Mitchel. Introducción al microprocesador 8086/8088 (16 bit). McGraw-Hill/Interamericana de México, S.A de C. V. México. 1988.
6. Swan, Tom. Mastering Turbo Assembler. Hayden Books. Indiana, U.S.A. 1989.