

PRÁCTICA 1. Programación en Ensamblador para Procesamiento de la Información

TEORÍA:

TEMA 1: INTRODUCCIÓN A LA ARQUITECTURA VON NEUMANN

TEMA 2: REPRESENTACIÓN DE LA INFORMACIÓN

TEMA 3: ALMACENAMIENTO DE LA INFORMACIÓN

TEMA 4: OPERACIONES CON LOS DATOS

Esta práctica tiene dos objetivos diferenciados. El primero de ellos está relacionado con la programación en ensamblador del microprocesador 8086. El segundo, pretende familiarizar al alumno con la arquitectura de procesador estudiada en el Tema 1, los sistemas de representación que se estudian en el Tema 2, mapa de memoria estudiado en el Tema 3 y las distintas operaciones con los datos que se estudian en el Tema 4.

1.1 Programación en ensamblador para 8086

Elegir uno de entre los ejercicios propuestos, describir su desarrollo y justificar el correcto funcionamiento del mismo.

Ejercicio 1

Desarrollar un programa en ensamblador del 8086 que a partir de una cadena numérica, que viene definida por 4 dígitos (cada dígito va ocupar una posición de memoria y con valores 0 ó 1), introducida por teclado, calcule, almacene en memoria y muestre en pantalla el valor decimal equivalente de la cadena según los siguientes sistemas de representación: Binario natural sin signo y Complemento a 1. Mostrar un esquema de los mapas de direcciones del sistema computador, indicando sus partes claramente diferenciadas.

Ejercicio 2

Desarrollar un programa en ensamblador del 8086 que a partir de una cadena numérica, que viene definida por 4 dígitos (cada dígito va ocupar una posición de memoria y con valores 0 ó 1), introducida por teclado, calcule, almacene en memoria y muestre en pantalla el valor decimal equivalente de la cadena según los siguientes sistemas de representación: Binario natural con signo y Complemento a 2. Mostrar un esquema de los mapas de direcciones del sistema computador, indicando sus partes claramente diferenciadas.

Ejercicio 3

Desarrollar un programa en ensamblador del 8086 que a partir de una cadena numérica, que viene definida por 4 dígitos (cada dígito va ocupar una posición de memoria y con valores 0 ó 1), introducida por teclado, calcule, almacene en memoria y muestre en pantalla el valor decimal equivalente de la cadena según los siguientes sistemas de representación: Complemento a 1 y Complemento a 2. Mostrar un esquema de los mapas de direcciones del sistema computador, indicando sus partes claramente diferenciadas.

Ejercicio 4

Desarrollar un programa en ensamblador del 8086 que a partir de una cadena numérica, que viene definida por 4 dígitos (cada dígito va ocupar una posición de memoria y con valores 0 ó 1), introducida por teclado, calcule, almacene en memoria y muestre en pantalla el valor decimal equivalente de la cadena según los siguientes sistemas de representación: Binario natural sin signo y Complemento a 2. Mostrar un esquema de los mapas de direcciones del sistema computador, indicando sus partes claramente diferenciadas.

Ejercicio 5

Desarrollar un programa en ensamblador del 8086 que a partir de una cadena numérica, que viene definida por 4 dígitos (cada dígito va ocupar una posición de memoria y con valores 0 ó 1), introducida por teclado, calcule, almacene en memoria y muestre en pantalla el valor decimal equivalente de la cadena según los siguientes sistemas de representación: Complemento a 2 y Exceso $Z = 2^{n-1}$. Mostrar un esquema de los mapas de direcciones del sistema computador, indicando sus partes claramente diferenciadas.

Ejercicio 6

Desarrollar un programa en ensamblador del 8086 que a partir de una cadena numérica, que viene definida por 4 dígitos (cada dígito va ocupar una posición de memoria y con valores 0 ó 1), introducida por teclado, calcule, almacene en memoria y muestre en pantalla el valor decimal equivalente de la cadena según los siguientes sistemas de representación: Binario natural sin signo y Exceso $Z = 2^{n-1}$. Mostrar un esquema de los mapas de direcciones del sistema computador, indicando sus partes claramente diferenciadas.

Ejercicio 7

Desarrollar un programa en ensamblador del 8086 que a partir de una cadena numérica, que viene definida por 4 dígitos (cada dígito va ocupar una posición de memoria y con

valores 0 ó 1), introducida por teclado, calcule, almacene en memoria y muestre en pantalla el valor decimal equivalente de la cadena según los siguientes sistemas de representación: Binario natural con signo y Complemento a 1. Mostrar un esquema de los mapas de direcciones del sistema computador, indicando sus partes claramente diferenciadas.

Ejercicio 8

Desarrollar un programa en ensamblador del 8086 que a partir de una cadena numérica, que viene definida por 4 dígitos (cada dígito va ocupar una posición de memoria y con valores 0 ó 1), introducida por teclado, calcule, almacene en memoria y muestre en pantalla el valor decimal equivalente de la cadena según los siguientes sistemas de representación: Binario natural con signo y Hexadecimal. Mostrar un esquema de los mapas de direcciones del sistema computador, indicando sus partes claramente diferenciadas.

Ejercicio 9

Desarrollar un programa en ensamblador del 8086 que a partir de una cadena numérica, que viene definida por 4 dígitos (cada dígito va ocupar una posición de memoria y con valores 0 ó 1), introducida por teclado, calcule, almacene en memoria y muestre en pantalla el valor decimal equivalente de la cadena según los siguientes sistemas de representación: Complemento a 2 y Hexadecimal. Mostrar un esquema de los mapas de direcciones del sistema computador, indicando sus partes claramente diferenciadas.

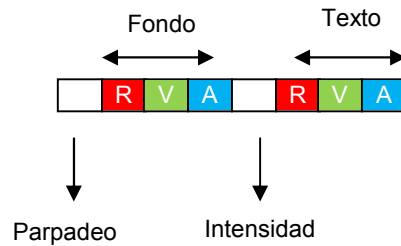
Ejercicio 10

Desarrollar un programa en ensamblador del 8086 que a partir de una cadena numérica, que viene definida por 4 dígitos (cada dígito va ocupar una posición de memoria y con valores 0 ó 1), introducida por teclado, calcule, almacene en memoria y muestre en pantalla el valor decimal equivalente de la cadena según los siguientes sistemas de representación: Exceso 2^{n-1} y Hexadecimal. Mostrar un esquema de los mapas de direcciones del sistema computador, indicando sus partes claramente diferenciadas.

1.2 Información útil

- Dirección de la Memoria de Video en modo texto: B800:0000h.
- Organización de la pantalla en modo texto: 25 filas x 80 columnas.
- Interrupción software para establecer el modo de pantalla:
 - AL = modo (texto = 03h, gráfico =13h).
 - AH = 00h
 - INT 10h

- Asignación de atributos a caracteres en memoria de video en modo texto:



Ejemplo:

```
MOV AX, 0B800h
```

```
MOV ES, AX
```

```
MOV AH, 00001111b ; carácter impreso en pantalla con fondo negro y texto blanco
```

```
MOV AL, carácter_ascii
```

```
MOV ES:[DI], AX ; DI = posición de la memoria donde se escribe el carácter
```

- Interrupción software para capturar una tecla **introducida por teclado**:

```
AH = 00h
```

```
INT 16h
```

Se almacena en AL el código ascii de la tecla pulsada

- Interrupción software para capturar una cadena de caracteres introducida por teclado:

```
AH = 0Ah
```

```
INT 21h
```

Lee por el teclado hasta que se pulsa *intro*. En este momento se ejecuta un Retorno de Carro (CR) que sitúa el cursor al comienzo de la línea, almacenando en memoria los caracteres leídos + el CR. Esta interrupción necesita que se reserve los siguientes espacios:

- a) Una posición de memoria que indica el **número máximo** de caracteres a teclear + 1 (el CR). El valor de este byte es establecido por el programador.
- b) Otra posición que indicará el **número real** de caracteres tecleados. El valor de este byte será establecido por la interrupción después de pulsarse el CR.
- c) **(Nmax+1)** posiciones donde se almacenarán los caracteres tecleados y el CR (se almacenará el código Ascii de cada carácter).

Ejemplo:

```
.DATA
```

```
...
```

```
CADENA DB 5, 0, 0, 0, 0, 0 ; cadena donde se almacenan los caracteres introducidos por teclado
```

```
...
```

```
.CODE
```

```
...
```

```
MOV DX, Offset CADENA ; Indica la posición de memoria a partir de la que se almacenará la CADENA
```

```
MOV AH, 0Ah ; código de función
```

```
INT 21h
```

1.3 Evaluación

Al finalizar la práctica el alumno debe entender y dominar los siguientes conceptos:

En relación al microprocesador 8086:

- Bus de direcciones y máximo espacio de memoria direccionable.
- Bus de datos: ancho de palabra.
- Organización de la memoria: segmento y desplazamiento. Tipos de segmentos, tamaño de los segmentos.
- Registros: tamaño, propósito; cuándo y para qué se usan.

En relación a la programación en ensamblador:

- Lenguaje Ensamblador y Código Máquina: definición y diferencias.
- Paso del lenguaje ensamblador a código máquina: ensamblado, linkado y depuración.
- Esquema de un programa en ensamblador: sentencias (directivas e instrucciones).
- Formato de las instrucciones: código de operación, operando fuente, operando destino.
- Modos de direccionamiento: Definición.
- Declaración de variables y constantes: directivas usadas (DB, DW), ubicación en memoria.
- Instrucción de transferencia de datos: MOV. Formato y restricciones.
- Instrucciones aritmético-lógicas: ADD, SUB, etc. Formato y restricciones
- Instrucciones de salto: bucles, uso de etiquetas.
- Tablas: definición e inicialización, ubicación en memoria, acceso a los datos, registros usados para recorrer tablas.

En relación a la Representación de la Información:

- Sistemas de representación: Rango. Resolución. Procedimiento para calcular el valor equivalente de una representación binaria dependiendo del sistema de representación.

En relación a las Operaciones con los Datos:

- Operaciones de extensión de signo para cadenas binarias según el sistema de representación empleado.

1.4 Referencias bibliográficas

- Morgan C. L., Waite M. (1988). Introducción al microprocesador 8086/8088 (16 Bit), Madrid, McGraw-Hill.
- Charte Ojeda F. (2003). Programación en ensamblador, Madrid, Anaya.
- García de Celis, Ciriaco. El universo digital del IBM PC, AT y PS/2
<http://meltingpot.fortunecity.com/uruguay/978/>

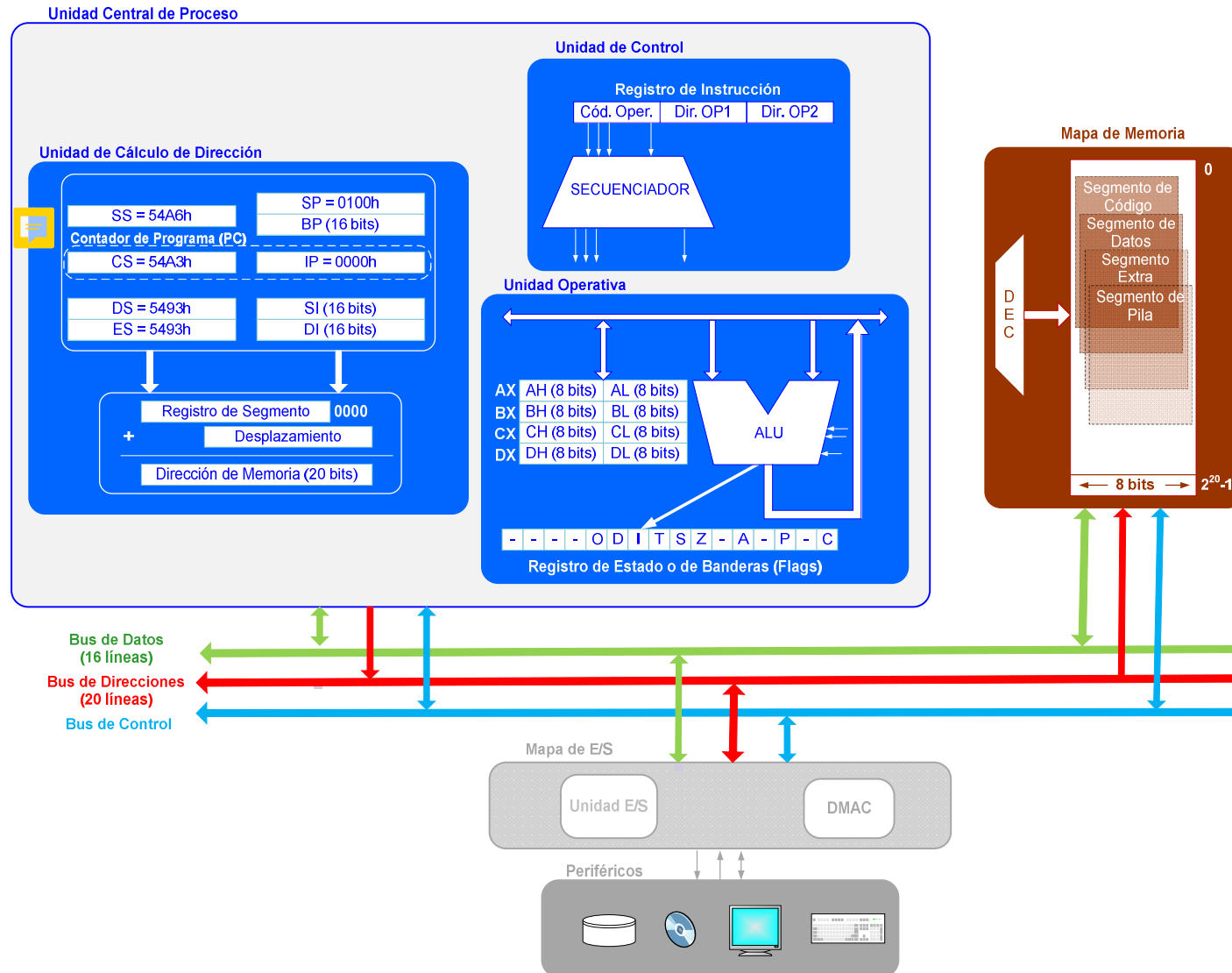


Figura 1.1. Esquema básico del microprocesador 8086.

Descripción de ejecución de instrucciones

