



UNIVERSIDAD NACIONAL ABIERTA
VICERRECTORADO ACADÉMICO
AREA: INGENIERÍA / CARRERA: INGENIERÍA DE SISTEMAS

MATERIAL INSTRUCCIONAL DE APOYO

NOMBRE: **MICROPROCESADORES**
Código: 316
U.C.: 4

CARRERA: Ingeniería de Sistemas
Código: 236

SEMESTRE: VII

AUTOR: Ing. Olga Esther Rodríguez Durán (Especialista de Contenido)
Ing. Judit Carballo (Coordinadora Ingeniería de Sistemas)
Lic. Carmen Velásquez de Rojas (Evaluadora)
Lic. Antonio Alfonzo MsC (Diseñador Instruccional)

COMITÉ TÉCNICO: Dra. Egleé Arellano de Rojas (Diseñadora)

Caracas, Octubre 2005

INTRODUCCIÓN

El estudio a distancia e incluso el presencial en algunos momentos es una forma de aprendizaje que se basa específicamente en el trabajo personal de los estudiantes con mayor o menor independencia respecto de la guía directa de los profesores. El estudiante, al estar provisto de material autoinstruccional, puede de esta forma acceder a un mayor grado de aprendizaje, puesto que logra profundizar o aprender según sus ritmos y conocimientos previos. Aquí radica el potencial que tiene la elaboración del presente Material Instruccional de Apoyo (M.I.A.), el cual junto con el libro *Los Microprocesadores Intel*, de Barry B. Brey y el Software Simulador del Microprocesador 80286 se complementan para conformar el paquete instruccional del curso **Microprocesadores**, aunado a ello el estudiante cuenta con la asesoría en los centros locales para aclarar dudas, tanto de manera presencial como a través del correo electrónico.










El Material Instruccional de Apoyo, se ha elaborado tomando en cuenta el nivel de conocimientos que ha obtenido el estudiante en los cursos anteriores, particularmente en el curso Arquitectura de Computadoras el cual define las conductas de entrada necesarias para el estudio de los microprocesadores y la implementación de programas, para resolver un problema específico. También se ha considerado la complejidad de cada tópico y el uso del software simulador, éste software apoya el estudio de aquellas unidades en las cuales se realiza la descripción y programación de un microprocesador comercial.

El Material Instruccional de Apoyo se ha organizado atendiendo a la Estructura del Curso. **El Módulo I:** El Microprocesador, Arquitectura e Interfaces, consta de cuatro unidades. **El Módulo II:** Familia de Microprocesadores, el Coprocesador Aritmético y la Tecnología MMX, incluye dos unidades, por su parte en el **Módulo III:** Programación del Microprocesador, se compone del estudio de dos unidades. Para cada módulo se presenta: una breve introducción, el objetivo del módulo y las unidades que lo componen. Por su parte cada unidad incluye: una breve presentación, el objetivo de la unidad, una sinopsis del contenido y una serie de actividades que el estudiante debe realizar para complementar su estudio, dentro de estas actividades se hace énfasis en la presentación de aspectos teóricos, ejemplos, ejercicios resueltos, ejercicios de autoevaluación, consultas a material bibliográfico y consultas a direcciones electrónicas en la Web. Cabe destacar que algunas de estas direcciones pueden no estar activadas al momento de la consulta, por lo que se recomienda al estudiante acceder a otras direcciones relativas al tema a través de un buscador (Browser) en la Web.

El estudio del curso Microprocesadores, exige el uso simultáneo del Plan de Curso y el paquete instruccional. Estos dos componentes se interrelacionan para proporcionarle una visión general del curso y las herramientas necesarias para el logro del objetivo terminal.

Iconos empleados en el material instruccional

A lo largo de la lectura de este material encontrará diversos íconos, cuyo significado se explica a continuación:

	<p>Ampliación de conocimientos: Está dirigido al estudiante que desea profundizar más en sus conocimientos en determinado tema.</p>
	<p>Atención: Se presenta cuando se quiere hacer una aclaratoria, una advertencia o una reflexión sobre algún aspecto del contenido.</p>
	<p>Consulta en la Web: Indica referencias a páginas Web</p>
	<p>Consulta en otros libros: Se refiere a un llamado a consulta en libros que no figuran como textos de carácter obligatorio para el curso.</p>
	<p>Dirección de Correo electrónico: Indica direcciones de correo electrónico</p>
	<p>Ejercicios propuestos: son ejercicios sugeridos a manera de práctica sobre algún tema de la unidad.</p>
	<p>Ejercicios de autoevaluación: Ejercicios que debe realizar el estudiante y posteriormente verificar contra los resultados aquí presentados.</p>
	<p>Ejemplo: Es la exposición de un caso alusivo al tema en cuestión y su resolución.</p>
	<p>Recordatorio: Indica algún aspecto a enfatizar, relacionado con los conocimientos adquiridos previamente por el estudiante.</p>

MÓDULO I

EL MICROPROCESADOR, ARQUITECTURA E INTERFACES.

En este módulo se estudia la configuración interna del microprocesador visible al programador, la manera como se direcciona la memoria y como funciona cada instrucción en la familia de los microprocesadores Intel. El establecimiento de las interfaces apropiadas para la conexión de los sistemas de memoria y de los dispositivos de entrada-salida (E/S) también son objetos de estudio en este módulo. Finalmente se profundiza en la implementación de interfaces sencillas que se conectan a los sistemas de bus ubicados en la computadora.

Objetivo del Módulo I: Resolver con sentido lógico y creativo, problemas inherentes a la arquitectura del microprocesador y al establecimiento de interfaces.

El módulo I está estructurado en cuatro unidades:

Unidad 1: Arquitectura del Microprocesador.

Unidad 2: Interfaz con la memoria.

Unidad 3: Interfaz de E/S Básica, Interrupciones y E/S Controlada por DMA.

Unidad 4: Interfaz del Bus.

UNIDAD 1

Arquitectura del Microprocesador.

El estudio del modelo de programación de los microprocesadores, puede realizarse través de la descripción de la **arquitectura interna visible** para el programa. Esta arquitectura comprende los registros que son utilizados por el programador en el desarrollo de aplicaciones, mediante instrucciones que hacen uso de ellos. En la Unidad I, se presentan los modelos de programación de los microprocesadores 8086 al Pentium II, pertenecientes éstos a la familia Intel, también su **modo de funcionamiento** en modo real y protegido.

Las características de diseño de los microprocesadores, establecen sus capacidades y limitaciones, por ejemplo es conveniente para el programador, que el microprocesador tenga diversos **modos de direccionamiento**, ya que eso le permitirá desarrollar software más eficiente. La Unidad I presenta los modos de direccionamiento de datos, de la memoria y de la pila, para la familia de microprocesadores Intel. Finalmente se incluye en esta unidad el estudio de las **instrucciones** con que cuentan los miembros de esta familia, este estudio contempla entre otros aspectos la sintaxis del lenguaje ensamblador y algunas de sus directivas, así como también la función de cada instrucción, lo que permite sentar bases para la programación de los microprocesadores en unidades posteriores.

Objetivo de la Unidad 1: Aplicar los conceptos de arquitectura de los microprocesadores en la descripción de su estructura y/o funcionamiento, dada ó no una situación.

Contenido de la Unidad 1: El contenido de la unidad contempla el estudio de los siguientes temas:

- El microprocesador y su arquitectura.
- Direccionamiento de memoria.
- Paginación de memoria.
- Modos de direccionamiento. Direccionamiento de la pila de memoria.
- Instrucciones.

Actividades recomendadas para el estudio del contenido de la unidad 1.



Atención: Considerar que en el **Plan de Curso de Microprocesadores**, hay un error en la sección **V. Diseño de la Instrucción**, en las secciones 2.1, 3.2, 4.2, 5.2 y 6.2, correspondientes a las unidades: 2, 3, 4, 5 y 6 respectivamente. El error se encuentra en la indicación del Texto a utilizar a utilizar, donde dice: “ Texto *Organización y Arquitectura de Computadoras* “. Debe decir : “ Texto: “ *Los Microprocesadores Intel.* “.

- 1.- Examine la tabla 1.1, en ella puede identificar las lecturas de los temas contentivos de la unidad 1:

TÍTULO	MATERIAL DE REFERENCIA	CONTENIDO	CAPITULO
El microprocesador y su arquitectura.	El texto Los Microprocesadores Intel , B. B. Brey.	Arquitectura interna del microprocesador.	2
		Direccionamiento de memoria en modo real.	
		Introducción al direccionamiento de memoria en modo protegido.	
		Paginación de memoria.	
Modos de direccionamiento.	El texto Los Microprocesadores Intel , B. B. Brey.	Modos de direccionamiento de datos.	3
		Modos de direccionamiento de la memoria del programa.	
		Modos de direccionamiento de la memoria de la pila.	
Instrucciones.	El texto Los Microprocesadores Intel , B. B. Brey.	Instrucciones para movimiento de datos.	4
		Instrucciones aritméticas y lógicas.	5
		Instrucciones de control de programa.	6

Tabla 1.1



Atención: Muchos de los ejemplos que presenta el texto Los microprocesadores Intel de B.B. Brey, usan el Microsoft MACRO Assembler (MAMS) como herramienta de desarrollo, pero la mayoría de ellos pueden ser ensamblados con otros ensambladores, por ejemplo ASM (Intel Assembler) y TASM (Borland Turbo Assembler). En el curso Microprocesadores se utilizará el Software Simulador incluido en el paquete instruccional. No obstante se recomienda al estudiante investigar sobre el uso del MAMS, con el objeto de ampliar sus conocimientos.

2.- Una vez leído el contenido indicado en la tabla 1.1, usted estará en capacidad de responder lo siguiente:

- 2.1 ¿Qué es el modelo interno de programación de un microprocesador?
- 2.2 ¿Por qué se puede afirmar que el modelo de programación de los microprocesadores Intel, es visible para el programa?
- 2.3 ¿Qué microprocesadores Intel contienen registros de 8 y 16 bits?
- 2.4 ¿Cuál es la función de los registros de propósito específico para los microprocesadores Intel?
- 2.5 ¿Cuál es la función y propósito de los registros de segmento: CS, DS, ES y SS?
- 2.6 ¿Cuáles son las principales ventajas de los siguientes direccionamientos de datos: inmediato, directo, indirecto, registro, indirecto con registro?
- 2.7 ¿Cuáles son los tres modos de direccionamiento de memoria de programa?
- 2.8 ¿Cuáles registros controlan la memoria de la pila?
- 2.9 ¿Cuáles son los códigos de operación que transfieren datos entre la pila y las banderas?
- 2.10 ¿Cuáles son los registros que emplean las estructuras de transferencia de cadenas de datos para direccionar memoria?
- 2.11 ¿Qué es una directiva de lenguaje ensamblador?
- 2.12 ¿Para qué son empleadas las directivas PROC y ENDP?
- 2.13 Un microprocesador 80386 opera en el modo protegido y utiliza un registro de 32 bits, ¿Es necesario que el código de operación esté precedido del prefijo de tamaño de registro? Razone su respuesta.
- 2.14 ¿Cuál es la función de la directiva .MODEL?

3.- Estudie el ejemplo 1.1, en el cual se presentan los modelos de programación de dos microprocesadores de 8 bits.



Ejemplo 1.1: Modelo de programación de los microprocesadores de 8 bits: 8080 y Z80.

El microprocesador Z80 de Zilog es una versión apreciablemente mejorada tanto en circuitería como en características de programación del modelo INTEL 8080, el Z80 resulta ser un microprocesador más rápido y sencillo en el desarrollo de

sistemas ya que solo usa una fuente de alimentación de +5 Volts, contiene íntegramente todo el conjunto de instrucciones del 8080, lo cual le permite ejecutar todos los programas escritos para el CPU 8080, contiene el Z80 una expansión adicional de 80 instrucciones de ahí se deriva su nombre, su repertorio suma un total de 156 instrucciones. Esta significativa mejora puede apreciarse si se comparan los modelos de programación de cada uno de ellos.

Se presentan a continuación los modelos de programación de los microprocesadores 8080 de Intel y el Z80 de Zilog, así como también la descripción y funcionamiento de cada uno de los registros que los conforman.

MODELO DE PROGRAMACIÓN DEL MICROPROCESADOR 8080

En la figura 1.1 se presenta el modelo de programación de este microprocesador, el cual dispone de los siguientes registros:

- Un registro acumulador de 8 bits (también llamado registro A).
- Seis registros de propósito general de 8 bits (B, C, D, E H y L.), que con algunas instrucciones pueden usarse como registros pares de 16 bits (B-C, D-E y H-L).
- Dos registros de 16 bits de propósito específico: un puntero de Stack (SP, siglas en inglés de Stack Pointer)) y un contador de programa (PC, siglas en inglés de Counter Program).
- Cinco flags (banderas) de 1 bit cada una que posee la unidad aritmético lógica y cuyo contenido es modificado después de la ejecución de numerosas instrucciones lógicas y aritméticas. Estas flags son: S, C, A, P y C.

Registro acumulador



Registros de propósito general

B	C
D	E
H	L

Registros de propósito específico

SP (Stack Pointer)
PC (Program Counter)

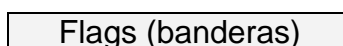


Figura 1.1 Modelo de programación del microprocesador Intel 8080



Atención: El registro Flags contiene 8 bits, pero solo se utilizan los cinco bits menos significativos, que corresponden a las cinco banderas: S, C, A, P y C.

MODELO DE PROGRAMACIÓN DEL MICROPROCESADOR Z80

El modelo de programación de este microprocesador (ver figura 1.2), está compuesto por un banco principal, otro alternativo y por último un banco compuesto por registros especiales. La existencia del banco alternativo mejora la velocidad ante la presencia de las interrupciones ya que permite cambiar desde el banco principal al alternativo. Los registros son:

- Hay 2 grupos de registros de propósito general. Cada grupo contiene 6 registros de 8 bits, que pueden ser usados individualmente como registros de 8 bits o en parejas como registros de 16 bits. Un grupo se llama BC, DE y HL y el otro BC', DE' y HL'. El programador puede cambiar a cualquiera de los dos grupos de registros con una simple instrucción de intercambio.
- Registros de Acumuladores y Banderas: El Z80 tiene 2 registros acumuladores independientes de 8 bits (comúnmente llamados A y A') y 2 registros de banderas asociados a los 2 acumuladores (F y F'). Los acumuladores almacenan los resultados de operaciones aritméticas o lógicas de 8 bits, y los registros de banderas indican condiciones para operaciones de 8 y 16 bits. Estas banderas son S, Z, A, P, N y C. El programador puede usar el par acumulador-bandera que desee y puede cambiar al otro par con una simple instrucción de intercambio.
- Registros de propósito específico:
 - Dispone del registro Program Counter (PC) de 16 bits, que contiene la dirección de la instrucción que se debe buscar en la memoria.
 - El Stack Pointer (SP) de 16 bits, el cual contiene la dirección del tope actual de la pila.
 - Dos Registros Índice (IX, IY), son dos registros independientes que contienen los 16 bits de direcciones que se usan en modos de direccionamiento indexado. En este modo, un registro índice se usa como base para apuntar a una región en memoria en la cual el dato será almacenado o recuperado.
 - El Registro de Interrupciones (I): El Z80 puede operar en un modo donde cualquier posición de memoria puede lograrse en respuesta de una interrupción. Este registro se usa para almacenar los 8 bits más significativos de la dirección, mientras el periférico que interrumpe provee los 8 bits menos significativos. Esto da la posibilidad de que las rutinas de interrupción sean localizadas dinámicamente en cualquier parte de la memoria con un mínimo tiempo de acceso a la rutina.
 - El Registro de Refresco de Memoria (R), de 8 bits que almacena el bloque de memoria a cuyo refresco se va a proceder.

Registros del banco principal

A	F
B	C
D	E
H	L

Registros del banco alternativo

A'	F'
B'	C'
C'	E'
H'	L'

Registros Índice

IX
IY

Registros de pila y contador del programa

SP
PC

Registro de interrupciones y de refresco de memoria

I
R

Figura 1.2 Modelo de programación de microprocesador Zilog Z80.

Atención: El texto utilizado para el estudio del Curso Microprocesadores se basa en los modelos fabricados por la compañía Intel, sin embargo se utiliza en este ejemplo, el microprocesador Z80 desarrollado por la compañía Zilog, con el objeto de que el alumno amplíe sus conocimientos.

- 4.- Lea el siguiente contenido teórico, el cual le aportará conocimientos necesarios para describir cómo acceder a la memoria de un computador, usando la técnica de direccionamiento de memoria en modo real.

Una de las funciones básicas que debe implementar un SO (Sistema Operativo) es la Administración de la Memoria, para tener un control sobre los lugares donde están almacenados los procesos y datos que actualmente se están utilizando. El direccionamiento de la memoria puede considerarse desde dos puntos de vista: físico y lógico. El primero se refiere a los medios electrónicos utilizados en el computador para acceder a las diversas posiciones de memoria, el segundo, a la forma en que se expresan y guardan las direcciones.

El sistema de numeración utilizado para representar las direcciones de memoria no suele ser el decimal, sino el hexadecimal. La razón es que los números

hexadecimales guardan cierta concordancia con las potencias de 2. Por ejemplo, un bus de direcciones de 8 bits puede acceder a 256 posiciones ($2^8 = 256$), expresado en hexadecimal es el rango 00-FF H. En caso de direcciones de 16 bits, se puede acceder 65.536 posiciones (es el rango 0000-FFFF H). El bus de direcciones del PC XT, tiene 20 bits, las posibilidades son 1.048.576 (es el rango 00000-FFFFF H).

Como veremos a continuación, puede utilizarse una doble notación: direcciones absolutas, indicadas por un hexadecimal de cinco dígitos, en el rango 00000-FFFFF H, o en direcciones segmentadas, que utiliza dos grupos de cuatro dígitos hexadecimales.

El direccionamiento segmentado, consiste en la asignación de bloques de memoria de tamaño variable, llamados segmentos. El tamaño de cada segmento será el requerido según la petición, por ejemplo el tamaño del proceso a cargar. El tamaño máximo para un segmento estará determinado por la capacidad de direccionamiento del hardware de la computadora, esto es, de cuantos bits se dispone para almacenar una dirección. El acceso a cada elemento individual (byte) en la memoria se hace mediante una dirección de memoria que se integra por dos elementos: una dirección de segmento y una de desplazamiento. La combinación (suma) de la dirección de segmento y la de desplazamiento generan la dirección de memoria absoluta a acceder, tal como se muestra en la figura 1.3, mostrada a continuación:

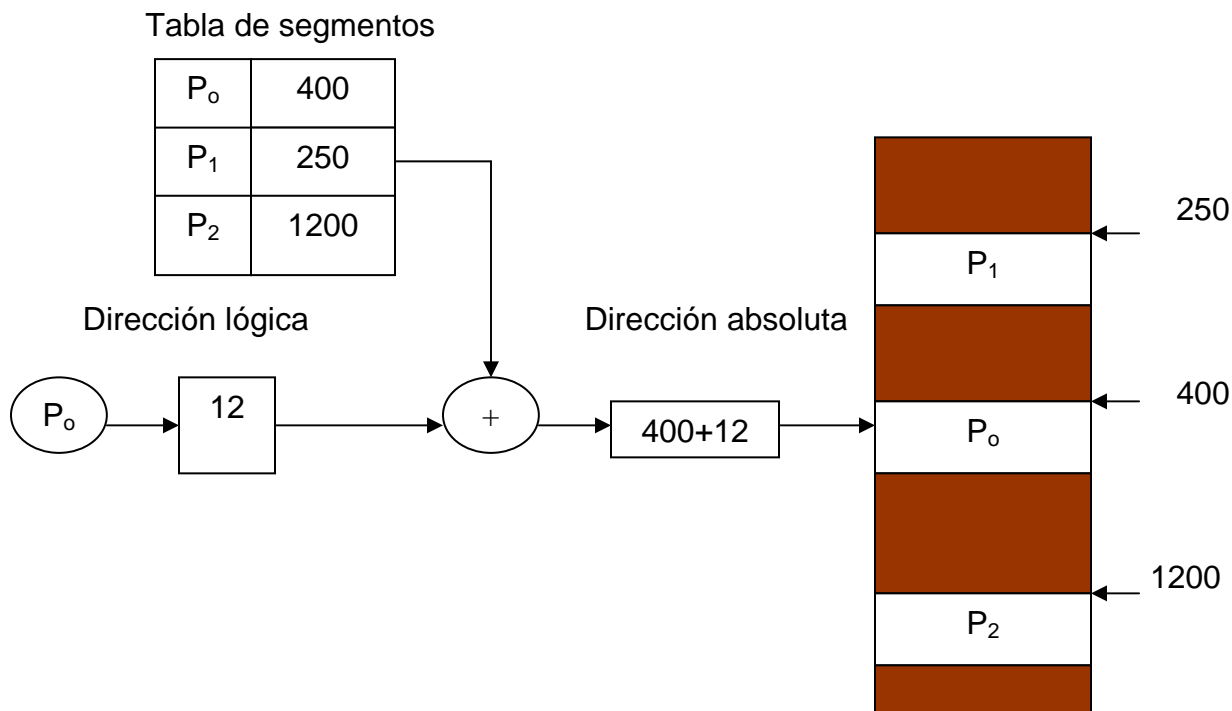


Figura 1.3 Esquema de direccionamiento de memoria en modo real



Atención: Una **dirección lógica** indica una posición relativa al comienzo de un programa, las instrucciones de programas contienen solo direcciones lógicas. Una **dirección física** es, por supuesto, la posición actual en la memoria principal. Cuando el procesador ejecuta un proceso, automáticamente convierte las direcciones lógicas en físicas.

- 5.- Estudie el ejemplo 1.2, en el cual se muestra un programa de aplicación que contiene un segmento de código, datos y pila cargado en un sistema DOS de memoria.



Ejemplo 1.2: Almacenamiento de un programa de aplicación en el sistema DOS de memoria.

El bus del PC XT contiene 20 líneas, estas líneas contienen las direcciones que identifican la parte del computador que comunica. Las direcciones son de dos tipos; unas se refieren a direcciones de memoria; otras a direcciones de puertos de E/S. Con los datos anteriores, resulta evidente que el PC XT puede manejar 1.048.576 direcciones (2^{20}). Sin embargo, el microprocesador 8088 presente en el XT dispone de registros de 16 bits que solo pueden albergar 65.536 posiciones (2^{16}), muy alejado del valor anterior. Para resolver el problema, se utilizan las direcciones segmentadas, compuestas por dos palabras de 16 bits que se componen para formar una dirección de 20 bits (esta dirección es denominada también como dirección absoluta). Para generar estas direcciones el 8088 dispone de cuatro registros de segmento: CS, SS, DS y ES.

Los aspectos concretos de almacenamiento de un programa dependen de la plataforma; no existe un modelo único. En la figura 1.4 se muestra como se utilizan los registros de segmento para señalar las direcciones de inicio de los diversos módulos de un programa cargado en memoria, dentro del espacio de memoria direccionable con 20 bits (00000-FFFFFF H).

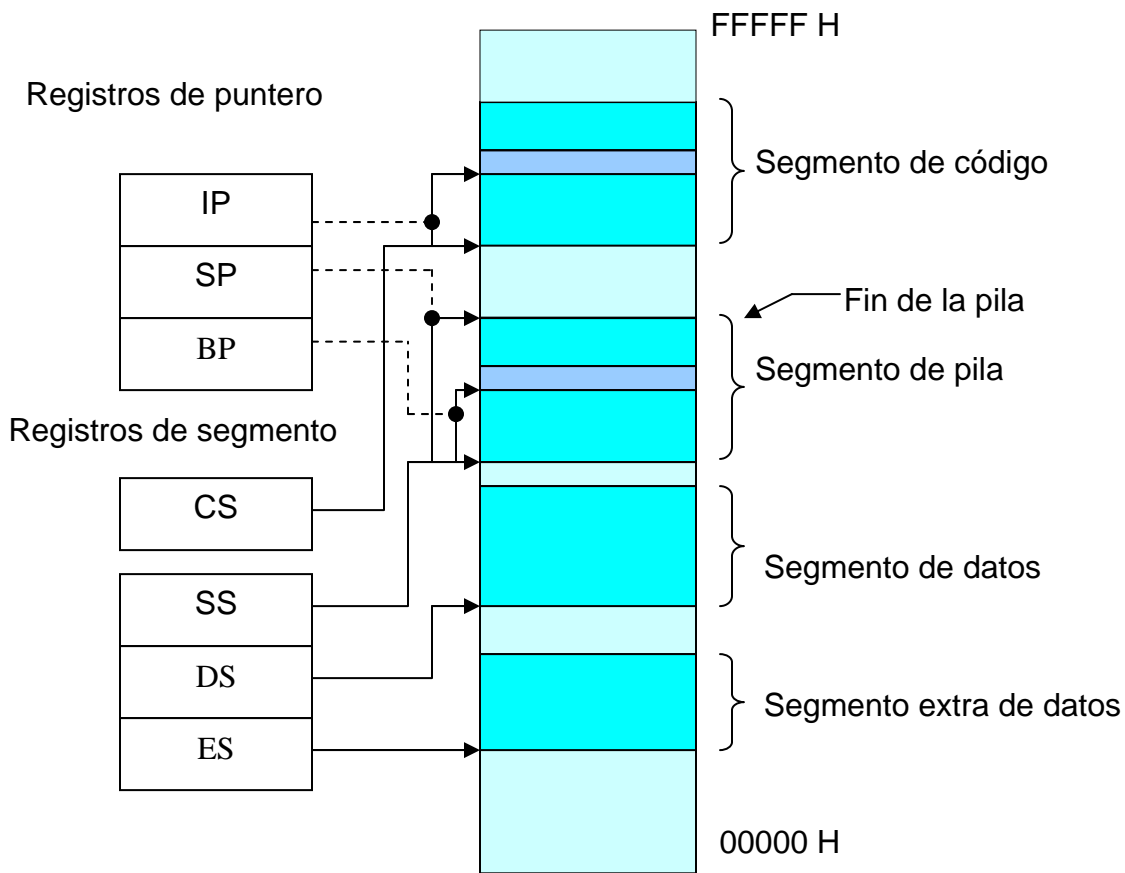



Figura 1.4 Programa de aplicación cargado en un sistema de memoria DOS

6.- Estudie el ejemplo 1.3, en el cual se aplica el manejo de memoria en modo protegido, para describir cómo el microprocesador 80386 asigna las localidades de memoria a un segmento de datos.

 **Ejemplo 1.3:** Manejo de memoria en modo protegido.

En la ejecución de un programa en un sistema basado en microprocesador 80386, se requiere en un instante dado almacenar datos en la memoria, bajo estas condiciones el contenido del registro de segmento de datos DS es el mostrado en la figura 1-5:

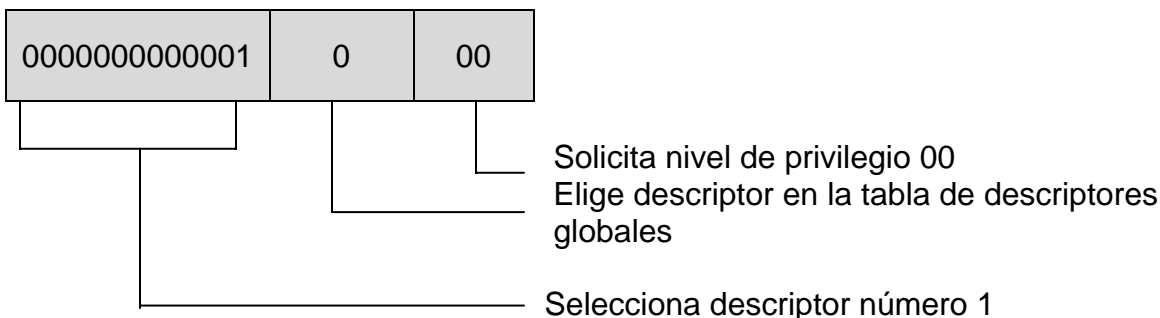


Figura 1.5 Contenido del registro de segmento DS

Y el formato del descriptor número 1 contenido en la tabla de descriptores globales es el siguiente:

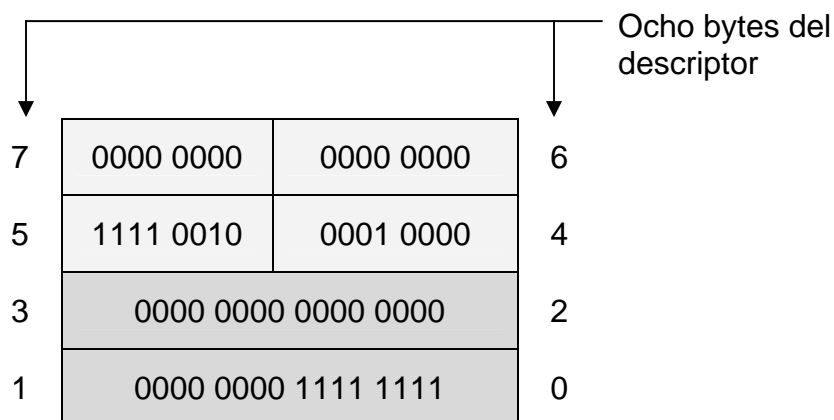


Figura 1.6 Formato del descriptor número 1

El inicio y fin del segmento de datos será el siguiente:

Base = 00100000 H

Límite = 000FF H

Final = Base + Límite = 00100000 H + 000FF H

Final = 001000FF H.

Se puede concluir entonces, que el microprocesador utilizará las localidades de memoria comprendidas en el rango: 00100000 H – 001000FF H para almacenar el segmento de datos.

- 7.- Estudie el ejemplo 1.4, en el cual se presenta uno de los cuatro mecanismos de manejo de la memoria disponible en el microprocesador Intel Pentium II.



Ejemplo 1.4: Gestión de memoria segmentada paginada en el Pentium II.

El Pentium II incluye hardware, tanto para segmentación como para paginación. Cuando ambos mecanismos están activados, se puede ver la memoria segmentada paginada.

La segmentación define particiones lógicas de memoria en el control de acceso. Por su parte con la paginación, se gestiona la asignación de memoria dentro de las particiones.

Cuando se utiliza segmentación las direcciones utilizadas en los programas son direcciones virtuales (llamadas también direcciones lógicas), estas direcciones constan de una referencia al segmento de 16 bits y un desplazamiento de 32 bits. El mecanismo de traducción de dirección para la segmentación implica hacer corresponder una dirección lógica con la dirección lineal. El siguiente paso es la traducción de una dirección lineal a una dirección física de 32 bits. La figura

1.7 se ilustra la combinación de los mecanismos de segmentación y paginación disponibles en el Pentium II.

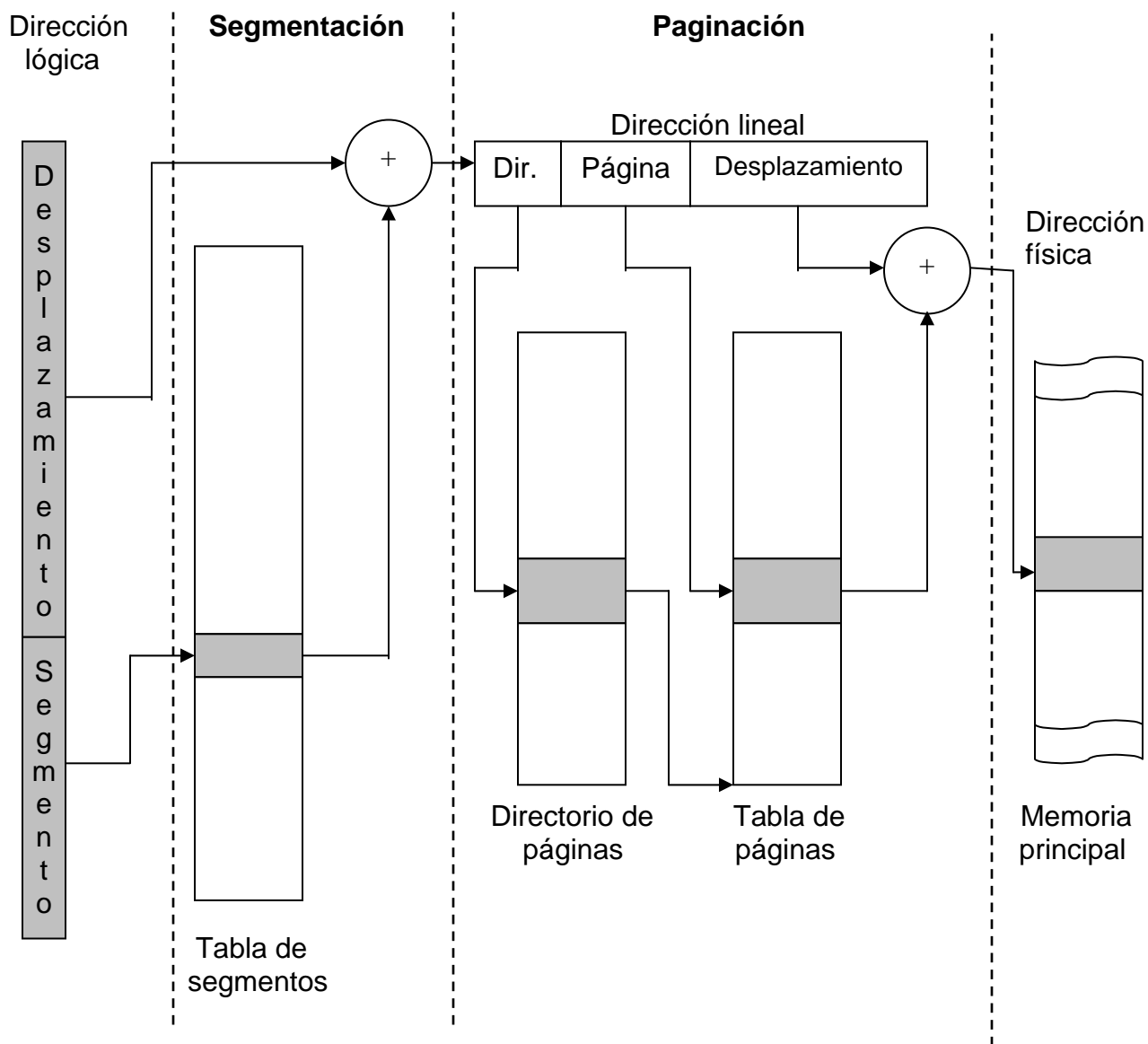


Figura 1.7 Mecanismos de traducción de una dirección de memoria en el Pentium II



Atención: Las **particiones** son bloques de memoria que pueden tener tamaño fijo o tamaño variable.

- 8.- Estudie el contenido de la Tabla 1.2, en la cual se presentan los registros de 8, 16 y 32 bits utilizados por la familia de microprocesadores Intel, para el direccionamiento de registro.

REGISTROS DE 8 BITS	REGISTROS DE 16 BITS	REGISTROS DE 32 BITS
AH AL BH BL CH CL DH DL	AX BX CX DX SP BP DI SI CS ES DS SS FS GS } *	EAX EBX ECX EDX ESP EBP EDI ESI

* **Nota:** Son utilizados por algunas instrucciones MOV, así como también por las instrucciones PUSH y POP.

Tabla 1.2

9.- Estudie el ejemplo 1.5, en el cual se describen los modos de direccionamiento disponibles en el Pentium II.



Ejemplo 1.5: Modos de direccionamiento del Pentium II.

En el mecanismo de traducción de direcciones del Pentium II presentado en la figura 1.8, se obtiene una dirección lineal que pasa por un mecanismo de traducción de páginas para producir una dirección física. En el ejemplo que se presenta a continuación se ignora el paso de paginación, ya que éste es transparente para el repertorio de instrucciones y para el programador.

El Pentium II está equipado con diversos modos de direccionamiento, ideados para permitir la ejecución eficiente de lenguajes de alto nivel. En la figura 1.8 se muestra el hardware involucrado y en la Tabla 1.3 los modos de direccionamiento disponibles.

Registro de segmento

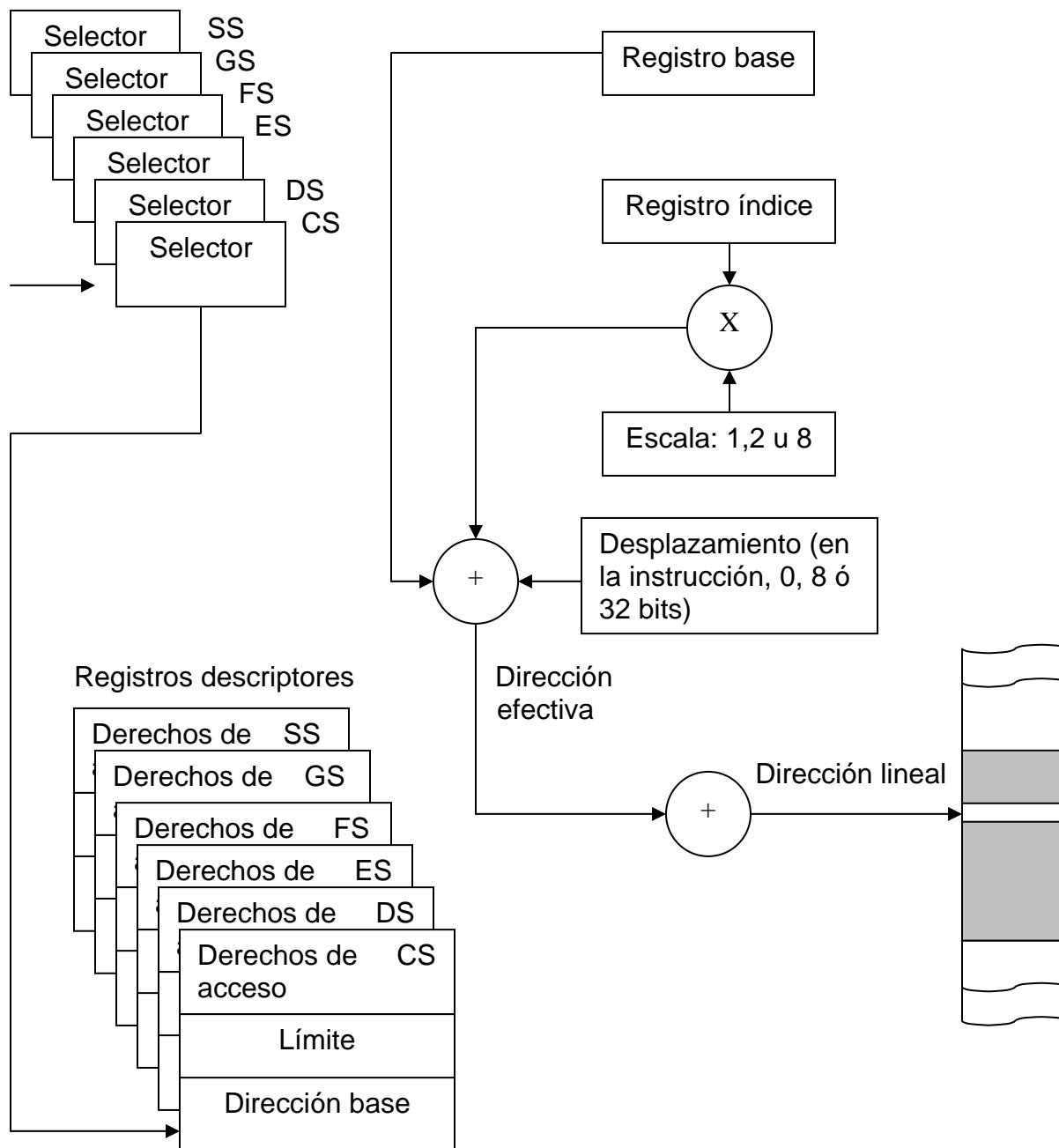


Figura 1.8 Cálculos en el modo de direccionamiento del Pentium II

MODO	ALGORITMO	DESCRIPCIÓN
Inmediato	Operando = A	El operando se incluye en la instrucción. El operando puede ser un byte, una palabra o una palabra doble de datos. Este modo puede utilizarse para definir y utilizar constantes, o para fijar valores iniciales de variables.
Registro	LA = R	El operando está situado en un registro. Si se hace un uso masivo de este direccionamiento en un repertorio de instrucciones, los registros de la CPU se emplearán intensamente, y debido al número limitado de registros en comparación con el número de posiciones de la memoria principal, se recomienda usarlo eficientemente.
Con desplazamiento	LA = (SR) + A	El desplazamiento del operando está incluido, formando parte de la instrucción. Debido a que implica instrucciones largas, se puede encontrar en pocas máquinas, en el Pentium II, el valor del desplazamiento puede ser de 32 bits, haciendo que la instrucción tenga 6 bytes.
Base	LA = (SR) + (B)	Especifica que uno de los registros de 8, 16 o 32 bits contiene la dirección efectiva
Base con desplazamiento	LA = (SR) + (B) + A	La instrucción incluye un desplazamiento que hay que sumar a un registro base, que puede ser cualquiera de los registros de uso general. Un ejemplo de uso de este modo es: utilización por un compilador para apuntar al comienzo de una zona de variables, en este caso el registro base apunta al comienzo de un marco de pila, que contiene las variables locales para el procedimiento correspondiente.
Índice escalado con desplazamiento	LA = (SR) + (I) x S + A	La instrucción incluye un desplazamiento a sumar a un registro, llamado en este caso registro índice. El registro índice puede ser cualquiera de los registros de uso general excepto el ESP. Este modo es muy conveniente para indexar matrices.

MODO	ALGORITMO	DESCRIPCIÓN
Base con índice y desplazamiento	$LA = (SR) + (B) + (I) + A$	Suma los contenidos de los registros base e índice, y un desplazamiento para formar la dirección efectiva. Como ejemplo, este modo de direccionamiento podría emplearse para manejar una matriz bidimensional; en este caso el desplazamiento apunta al inicio de la matriz, y cada registro gestiona una dimensión de la misma.
Base con índice escalado y desplazamiento	$LA = (SR) + (I) \times S + (B) + A$	Suma el contenido del registro índice, multiplicado por un factor de escala, con el contenido del registro base y el desplazamiento. Es útil Este modo permite la indexación eficiente de una matriz bidimensional cuando los elementos de la misma tienen longitudes de 2, 4 u 8 bytes.
Relativo	$LA = (PC) + A$	Se suma un desplazamiento al valor del contador del programa que apunta a la siguiente instrucción. Puede emplearse en instrucciones de transferencia del control (control de flujo).

LA = dirección lineal
 SR = Registro de segmento
 A = Contenido de un campo de dirección de la instrucción
 I = Registro índice
 (X) = contenido de X
 PC = Contador de programa
 R = Registro
 B = Registro Base
 S = Factor de escala

Tabla 1.3 Modos de direccionamiento del Pentium II.

10.- Estudie el ejemplo 1.6, en el cual se ilustra el uso de la pila para el paso de parámetros entre funciones, en el lenguaje C de alto nivel.



Ejemplo 1.6: Uso de la pila cuando se invoca una función en lenguaje C.

En el lenguaje C normalmente se utiliza la pila para pasar parámetros entre funciones. Antes de realizar la llamada a una función se insertan en la pila todos los parámetros luego se invoca a la función y ésta lee los parámetros de la pila.

Cuando se realiza la invocación de una función utilizando las instrucciones CALL y RET, la estructura típica de la pila es la que se muestra en la figura 1.9:

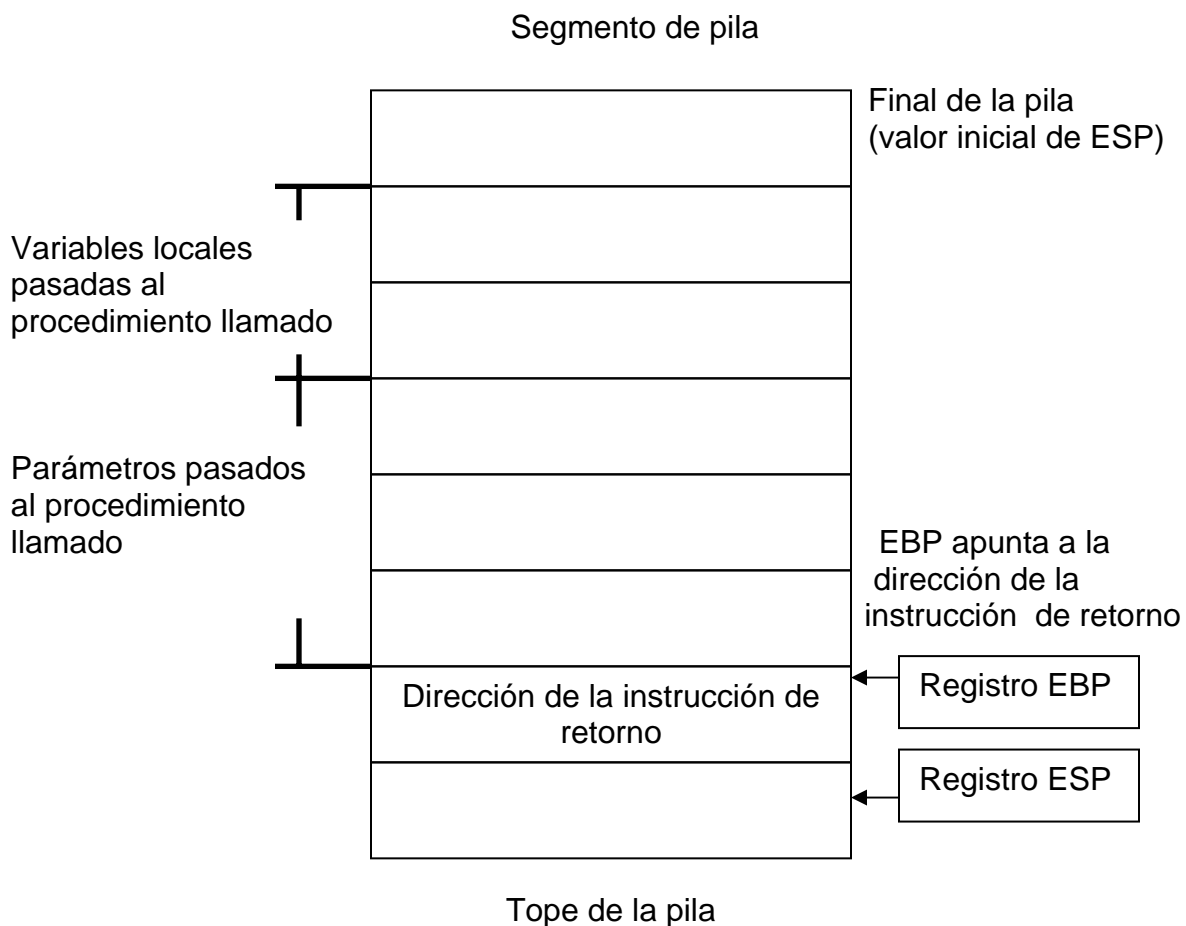


Figura 1.9 Uso de la pila cuando se invoca un función en lenguaje C

El orden en el que se insertan los parámetros en la pila es importante. Los últimos parámetros en insertarse en la pila son los primeros parámetros en la declaración de la función C.

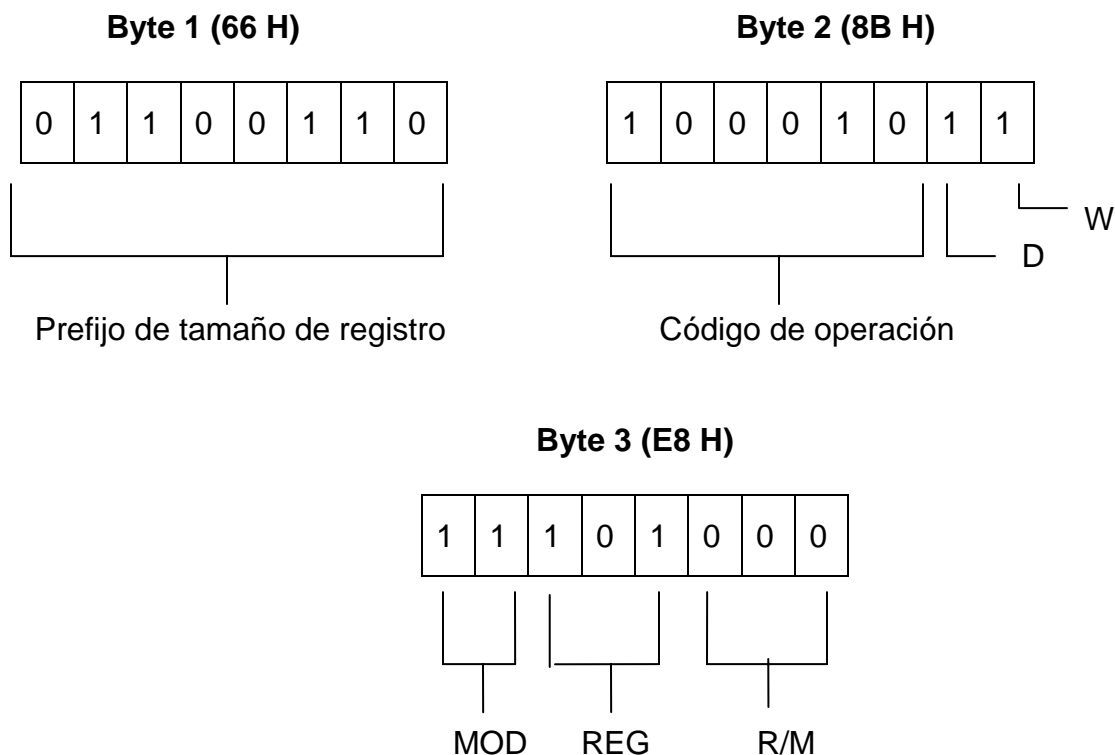
11.-Estudie el ejemplo mostrado a continuación en el cual se convierte una instrucción escrita en lenguaje de máquina a lenguaje ensamblador, para los casos en que un microprocesador 80386 o posteriores, opera en el modo de instrucciones de 16 bits y de 32 bits.



Ejemplo 1.7: Instrucción 668BEC H convertida a su forma de lenguaje ensamblador, cuando es ejecutada en un microprocesador 80386 o posteriores, operando en el modo de instrucciones de 16 bits.

El microprocesador opera en el modo de instrucciones de 16 bits.

La instrucción consta de tres bytes: 66 H, 8B H y EC H y el formato de la instrucción en binario será el siguiente:



El byte 1: Selecciona operandos de registro de 32 bits.

El byte 2: Contiene el código de operación conformado por los seis bits más significativos (100010), que corresponden a la operación MOV. El bit de dirección D indica que la palabra doble ($W = 1$) será movida al registro destino.

El byte 3: Puesto que el campo MOD contiene 11, el campo R/M indica que es un registro. Las asignaciones de REG y R/M serán las siguientes:

- REG = 101, que corresponde al registro de 32 bits EBP (operando destino)
- R/M = 000, que corresponde al registro de 32 bits EAX (operando fuente)

La instrucción escrita en lenguaje ensamblador es MOV EBP, EAX. Esta instrucción mueve la palabra doble desde el registro EAX al registro EBP.



Atención: El ejemplo 1.7 está incluido en el texto Los microprocesadores Intel, pero es explicado con más detalle en el presente material.

- 12.-Estudie el ejemplo 1.8, en el cual se presenta un pequeño programa escrito en lenguaje ensamblador que muestra el uso de la instrucción para transferencia de datos MOV y el empleo de algunas directivas.



Ejemplo 1.8: Aplicación de la instrucción MOV de transferencia de datos y las directivas ORG, DB y DW.

```

ORG 1000 H
    var_byte    DB    20 H
    var_word    DW    ?
ORG 2000 H
    MOV AX, 1000 H
    MOV BX, AX
    MOV BL, var_byte
    MOV var_word, BX
    END

```

En el programa anterior, se observa el uso el uso más común de la directiva ORG. La idea es separar las variables del programa. En el programa, las variables serán almacenadas a partir de la dirección 1000 H mientras que las instrucciones del programa estarán a partir de la dirección 2000 H.

La primera instrucción MOV asigna el valor inmediato 1000 H al registro AX. Esta instrucción emplea el modo de direccionamiento conocido como “inmediato”. Como ambos son valores de 16 bits, no hay inconveniente en esa asignación.

El siguiente MOV asigna el contenido del registro AX al registro BX. De nuevo, como ambos son de 16 bits, es una asignación válida. El modo de direccionamiento que usa es el denominado “registro”.

El tercer MOV asigna el contenido de la variable “var_byte” (que es 20 H) al registro BL. Como BL es la parte baja de BX, pasa de 1000 H a 1020 H. Como BL y “var_byte” son ambos de 8 bits, es una asignación permitida. Además, se emplea el modo de direccionamiento “directo”.

El último MOV asigna el valor contenido en el registro BX a la dirección de memoria a la que “var_word” hace referencia. Ahora, “var_word” contiene el valor 1020 H.

- 13.-Estudie el ejemplo 1.9, en el cual se presenta un pequeño programa escrito en lenguaje ensamblador que muestra el uso de las instrucciones aritméticas ADD y ADC.



Ejemplo 1.9: Aplicación de las instrucciones aritméticas ADD (suma) y ADC (suma con acarreo).

```

ORG 1000 H
    dato1_l    DW    0FFFF H
    dato1_h    DW    0015 H
    dato2_l    DW    0011 H
    dato2_h    DW    0002 H
ORG 2000 H
    MOV        AX, dato1_l
    ADD        AX, dato2_l
    MOV        BX, dato1_h
    ADC        BX, dato2_h
    END

```

Note que la segunda suma es realizada usando un ADC y no un ADD. De esta manera, si en el ADD se produce un acarreo, éste es sumado junto a dato1_h y dato2_h durante el ADC, produciendo el resultado correcto.

- 14.-Estudie el ejemplo 1.10, en el cual se emula una instrucción de un lenguaje de alto nivel en lenguaje ensamblador, utilizando instrucciones aritméticas, de salto (condicional e incondicional) y de transferencia de datos.



Ejemplo 1.10: Aplicación de instrucciones del lenguaje ensamblador para codificar la estructura de control IF THEN ELSE del lenguaje Pascal.

No existe una instrucción en el lenguaje ensamblador que sea capaz de hacer lo que hace la estructura IF-THEN-ELSE de Pascal. Sin embargo, es posible emularla mediante la combinación de instrucciones CMP y saltos condicionales e incondicionales.

Por ejemplo, para simular el siguiente código de Pascal:

```

IF AL = 4 THEN
  BEGIN
    BL = 1;
    CL = CL + 1;
  END;

```

Se debe comenzar calculando la condición del IF, en este caso, comparar AL con 4. Eso se logra con una instrucción CMP:

```
CMP AL, 4
```

Esta instrucción alterará los flags y en particular, interesa ver al flag Z, ya que si dicho flag está en 1, implica que al restar AL con 4, el resultado dio 0, por lo que AL tiene que valer 4. Entonces, si esa condición es verdadera, se debería ejecutar las instrucciones que están dentro del THEN. Si no, se evita ejecutarlas.

Una solución que permite esto es la siguiente:

```

        CMP AL, 4      ;(1)
        JZ   Then     ;(2)
        JMP  Fin_IF   ;(3)
Then:   MOV  BL, 1    ;(4)
        INC CL       ;(5)
Fin_IF: HLT          ;(6)

```

Analizando el código se puede establecer lo siguiente para cada uno de los comentarios:

- Si la comparación en (1) establece el flag Z en 1, el salto en (2) se produce, haciendo que la ejecución continúe en la etiqueta Then. Ahí, se ejecutan las instrucciones (4) y (5) que hacen lo que se encuentra en el THEN del IF y continúa la ejecución en la instrucción apuntada por la etiqueta Fin_IF
- Si el flag Z quedó en 0 en (1), el salto en (2) no se produce, por lo que la ejecución continúa en la próxima instrucción, el JMP en (3), que saltea las instrucciones y continúa la ejecución en la instrucción apuntada por la etiqueta Fin_IF , que señala el final del IF.
- En el final del IF, se ejecuta un HLT (6) para terminar la ejecución del programa.

15.- Resuelva las preguntas y los problemas presentados en el texto indicado en la bibliografía obligatoria.



Ejercicios propuestos

- 1.- Aplique los conceptos de modelo de programación estudiados y describa el modelo interno de programación de los microprocesadores de 16 y 32 bits de la familia Intel, utilizando el mismo esquema presentado en el ejemplo 1.1.
- 2.- Se propone al estudiante realizar el ejemplo 1.3, utilizando el sistema de codificación hexadecimal para especificar el contenido del descriptor.
- 3.- Determine la dirección de memoria a la cual accede cada una de las siguientes instrucciones con el modo de operación real (asuma para ello que DS = 0200 H, BX = 0300 H y DI = 400 H):
 - a) MOV EAX,[BX]
 - b) MOV [DI],AL
- 4.- Se propone al estudiante realizar el ejemplo 1.7, pero considerando que el modo de instrucciones es de 16 bits.
- 5.- Desarrolle una secuencia de instrucciones que intercambie el contenido de AX con BX, el de ECX con EDX y el de SI con DI.
- 6.- Explique cómo se convierte la instrucción AAM de binario a BCD.



Consulta en otros libros

[Sayers 1995] trata arquitecturas prácticas de microprocesadores de 8, 16 y 32 bits.

[Stallings 2000] Incluye material sobre: la organización de los registros visibles al usuario (modelo de programación), los modos de direccionamiento y el repertorio de instrucciones.

[Lameda 1984] presenta el estudio de microprocesadores de 8 bits.

[Flynn85] Flynn, M.; Jonson, J.; and Wakefield, S. On Instruction Sets and Their Formats IEEE Transactions on Computer, September 1987. Contiene aspectos de diseño de repertorios de instrucciones, particularmente aquellos relativos a los formatos.

[Tanenbaum 2000] en el cual se presentan los foatos de las intrucciones, los modos de direccionamiento y los tipos de instrucciones. Tambien presenta el nivel del lenguaje ensamblador.

Además se encuentra información sobre la arquitectura del microprocesador en cualquiera de los textos recomendados



Consulta en la Web

<http://www.itlp.edu.mx/publica/tutoriales/ensamblador/indice5.htm> : Contiene información sobre los elementos básicos del lenguaje ensamblador.

<http://weblidi.info.unlp.edu.ar/catedras/organiza/download/apunte4.pdf> : Contiene información sobre las instrucciones y directivas del lenguaje ensamblador.

<http://www.tau.org.ar/base/lara.pue.udlap.mx/sistoper/capitulo4.html>: Incluye información interesante sobre la segmentación y la paginación de memoria.

<http://futura.disca.upv.es/~eso/es/t2-arquitectura/gen-t2-arquitectura.html>: Presenta una introducción a la arquitectura IA-32 de los microprocesadores Intel.



Ejercicios de Autoevaluación

- 1.- Responda con sus propias palabras los siguientes cuestionamientos:
 - a) ¿Cuál es la función de un registro de segmento en el direccionamiento de memoria en modo protegido?
 - b) ¿Cuál es la función del TLB (buffer de referencia de traducción), ubicado dentro del microprocesador 80486?
- 2.- ¿Cuáles son las localidades de inicio y final que direcciona un descriptor del microprocesador 80286 con una dirección de base de A00000 H y un límite de 1000 H?
- 3.- Represente en forma gráfica un descriptor que describa un segmento de memoria bajo las siguientes condiciones:
 - Inicie en la localidad 03000000 H y termine en la localidad 05FFFFFF H.
 - El segmento de memoria es un segmento de datos que crece hacia arriba en el sistema de memoria y puede ser escrito.
 - El descriptor es para un microprocesador 80386.
- 4.- Genere en forma gráfica la dirección de memoria a la cual accede la instrucción MOV AL,[1234 H] con el modo de operación real (suponga para ello que DS = 0200 H y que en la localidad de memoria 3134 H se encuentra almacenado el valor 8A H).
- 5.- Desarrolle una secuencia corta que incluya instrucciones de transferencia de datos y lógicas, la cual permita establecer en 1 los cinco bits del extremo derecho de DI, sin cambiar los bits restantes de éste. La secuencia guardará el resultado en SI, con el objeto de no cambiar el contenido original de DI.



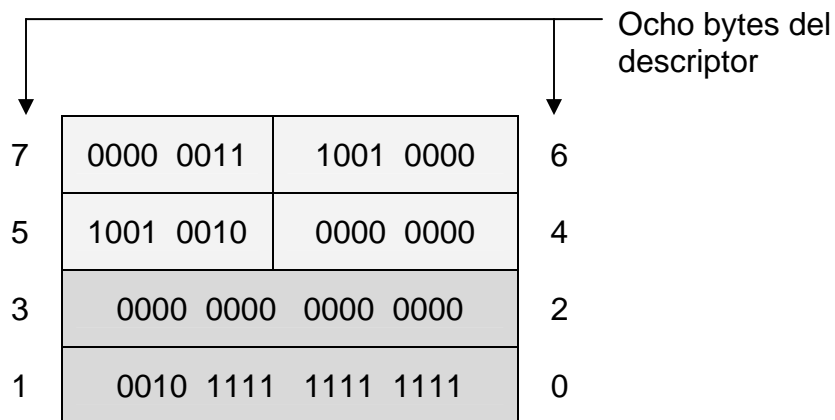
Respuesta a los Ejercicios de Autoevaluación

- 1.- A continuación las respuestas:
 - a) El registro de segmento contiene un selector que elige a un descriptor de la tabla de descriptores ya sea local, o global, también contiene el nivel de privilegio solicitado.
 - b) El TLB almacena las 32 últimas traducciones de la dirección lineal a la física de la unidad de paginación.

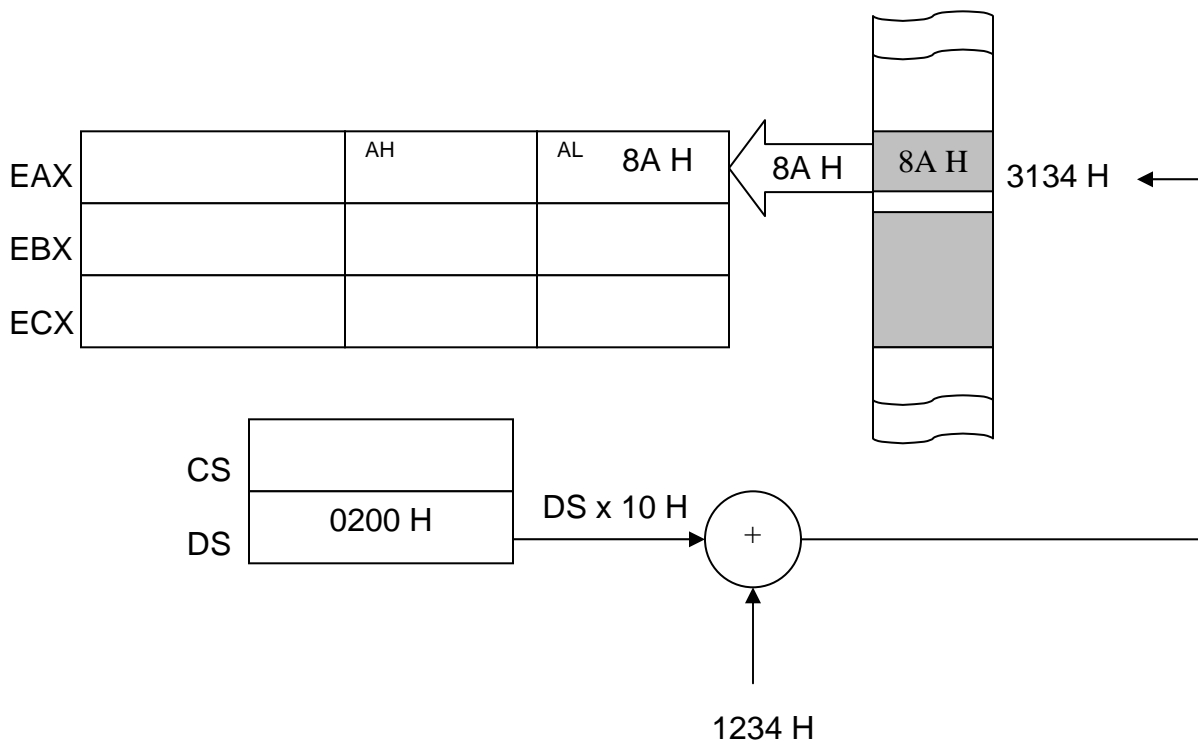
2.- El inicio y fin del segmento será el siguiente:

Base = Inicio = A00000 H
 Final = Base + límite = A00000 + 1000 h
 Final = A01000 H

3.- Utilizando el formato presentado el ejemplo 1.3:



4.- A continuación se genera la dirección para la instrucción MOV AL,[1234 H]:



5.- La secuencia será la siguiente:

```
MOV     SI, DI
OR      SI, 1F H
```

UNIDAD 2

Interfaz con la Memoria.

En computadores antiguos, la forma más común de almacenamiento de acceso aleatorio para la memoria principal consistía en una matriz de pequeños anillos ferromagnéticos denominados núcleos. Pero con el advenimiento de la microelectrónica y sus ventajas, nació el uso de las memorias semiconductoras para la memoria principal. En esta unidad se exploran aspectos claves de esta tecnología, se inicia con el estudio de la organización y funcionamiento de los dispositivos semiconductores de memoria, específicamente los cuatro tipos más comunes: ROM (memoria de sólo lectura), EEPROM (memoria de sólo lectura, borrrable y programable eléctricamente), SRAM (memoria estática de acceso aleatorio) y DRAM (memoria dinámica de acceso aleatorio).

Una vez estudiados los componentes que conforman la memoria principal del microprocesador, es necesario que se ensamblen estos componentes en forma correcta a objeto de producir un sistema de memoria adecuado para un microprocesador. Debe considerarse la cantidad de memoria requerida y el método óptimo de conectar a toda la memoria del microprocesador. Por esto se incluye en la unidad 2, la solución de problemas de conexión de la memoria al microprocesador mediante la interfaz adecuada, específicamente para la familia de los microprocesadores Intel.

Objetivo de la Unidad 2: Resolver problemas de conexión de la memoria al microprocesador, mediante el uso de la interfaz apropiada.

Contenido de la Unidad 2: El contenido de la unidad 2 incluye el estudio de los siguientes temas:

- 2.1 Dispositivos de memoria
- 2.2 Decodificación de la dirección
- 2.3 Interfaz de memoria para los microprocesadores de 8, 16, 32 y 64 bits
- 2.4 RAM dinámica

Recomendaciones para el estudio del contenido de la unidad 2.

- 1.- Examine la tabla 2.1, en ella puede identificar las lecturas de los temas contentivos de la unidad 2.

TÍTULO	MATERIAL DE REFERENCIA	CAPITULO	CONTENIDO	SECCIÓN
Interfaz con la memoria.	El texto Los Microprocesadores Intel , B. B. Brey.	10	Dispositivos de memoria.	10.1
			Decodificación de la dirección.	10.2
			Interfaz de la memoria de los microprocesadores de 8, 16, 32 y 64 bits.	10.3 -10.6
			RAM dinámica. Impresoras.	10.7

Tabla 2.1

- 2.- Una vez leído el contenido indicado en la tabla 2.1, usted estará en capacidad de responder lo siguiente:

- 2.1 ¿Cuáles son los terminales de conexión comunes a todos los dispositivos de memoria?
- 2.2 ¿Por qué es necesario decodificar las líneas de dirección del microprocesador?
- 2.3 ¿Qué es una interfaz de memoria?
- 2.4 ¿Qué tipos de conexión son comunes para todos los dispositivos de memoria?
- 2.5 ¿Cuántas palabras se encuentran en un dispositivo de memoria que tenga 8 conexiones de dirección?

- 3.- Estudie la figura 2.1, la cual ilustra una jerarquía de memoria típica, como la que se podría encontrar en una computadora de tamaño moderado. Tome en cuenta que no es la única configuración que se podría usar; pero sirve para ilustrar lo que podría esperarse encontrar cuando se estudia una computadora real.

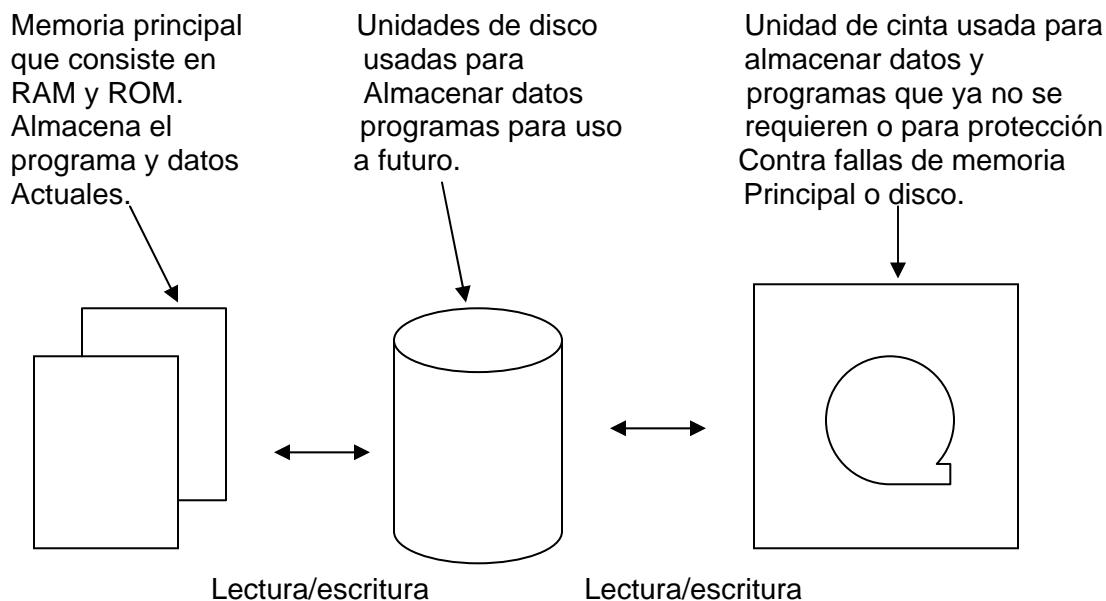


Figura 2.1 Jerarquía típica de memoria.



Atención Las unidades de almacenamiento de disco y de cinta no son objeto de estudio del curso Microprocesadores.

- 4.- Examine la figura 2.2 presentada en el ejemplo 2.1, esta arquitectura ilustra una arquitectura típica de un arreglo de memoria. En el arreglo, m bits de dirección se dividen en t bits de dirección de fila y s bits de dirección de columna, para simplificar los decodificadores a utilizar.



Ejemplo 2.1: Arquitectura típica de un arreglo de memoria.

La memoria de la figura 2.2 tiene n bits de capacidad, o sea que puede almacenar n bits de datos. Para acceder a cualquiera de estas n celdas de memoria es necesario suministrar una dirección única para identificar la localización requerida dentro del arreglo de memoria.

El acceso o decodificación de la dirección requerida se facilita dividiéndola en direcciones de fila y columna para aplicarlas al arreglo de memoria. Los m bits de dirección se dividen en t bits de dirección de fila y s de dirección de columna, lo que simplifica mucho los decodificadores requeridos en el chip de memoria. En vez de utilizar un decodificador de línea m a línea 2^m , basta con dos decodificadores de línea más pequeños y simples, t a línea 2^t y línea s a línea 2^s . Además si m es un número par, es decir s y t son iguales, se puede simplificar todavía más el chip de memoria ya que solo se requiere un tipo de decodificador aunque se necesiten dos

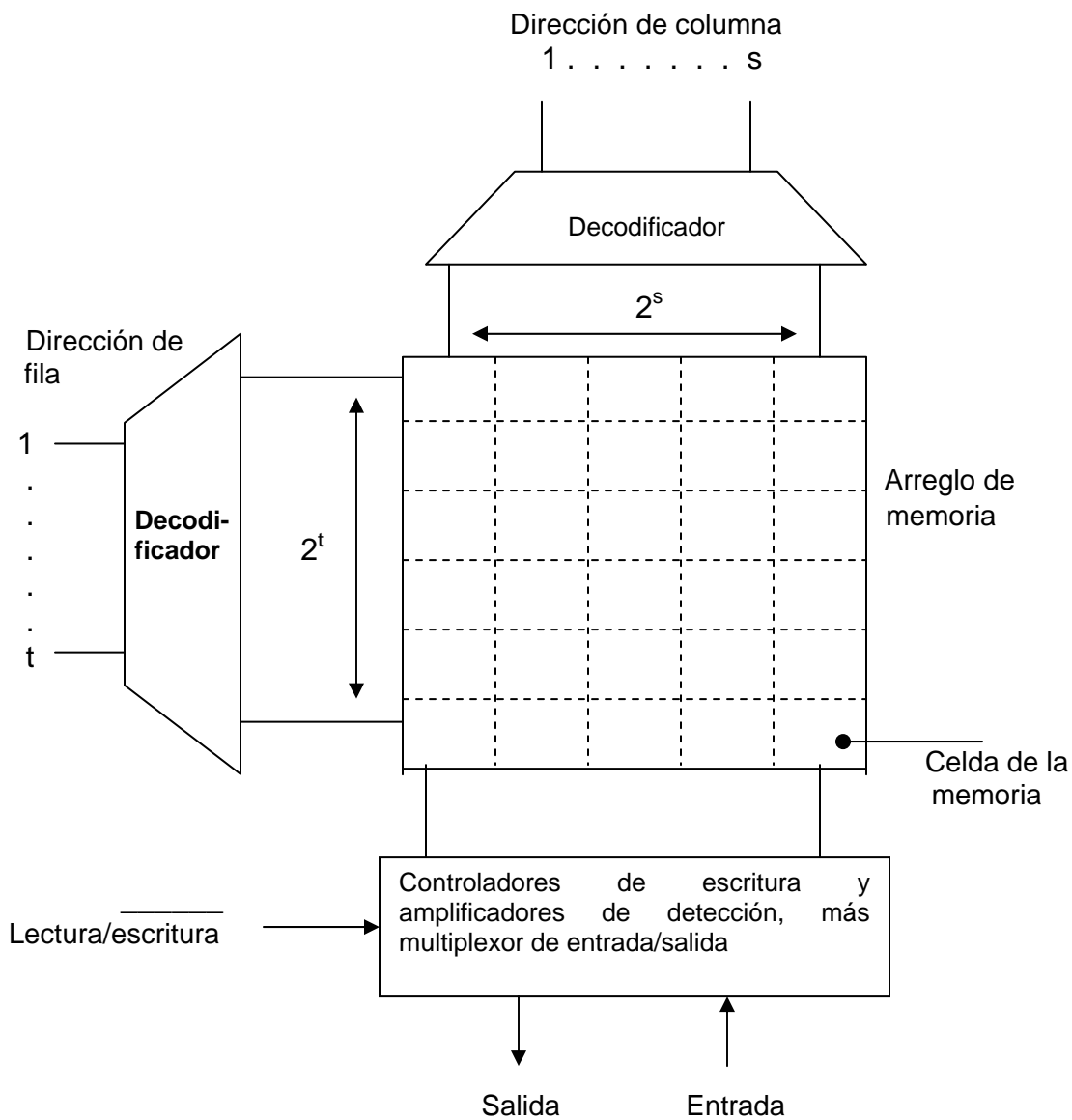


Figura 2.2 Arquitectura típica de un arreglo de memoria de 2^n celdas.

5.- Estudie el ejemplo 2.2, el cual explica en términos generales la forma en que se lleva a cabo la corrección de errores en las memoria semiconductoras.



Ejemplo 2.2: Código corrector de errores.

“Un error en un sistema digital producirá que un 1 se reemplace con un 0 o que un 0 se reemplace con un 1. Es esencial que los errores se detecten de manera que se haga algo para rectificarlos” [Sayers,1995].

Las memorias semiconductoras están sujetas a errores. Es por esto que la mayoría de los sistemas de memoria modernos incluyen lógica para detectar y corregir errores. La figura 2.3, ilustra en términos generales como se lleva a cabo este proceso.

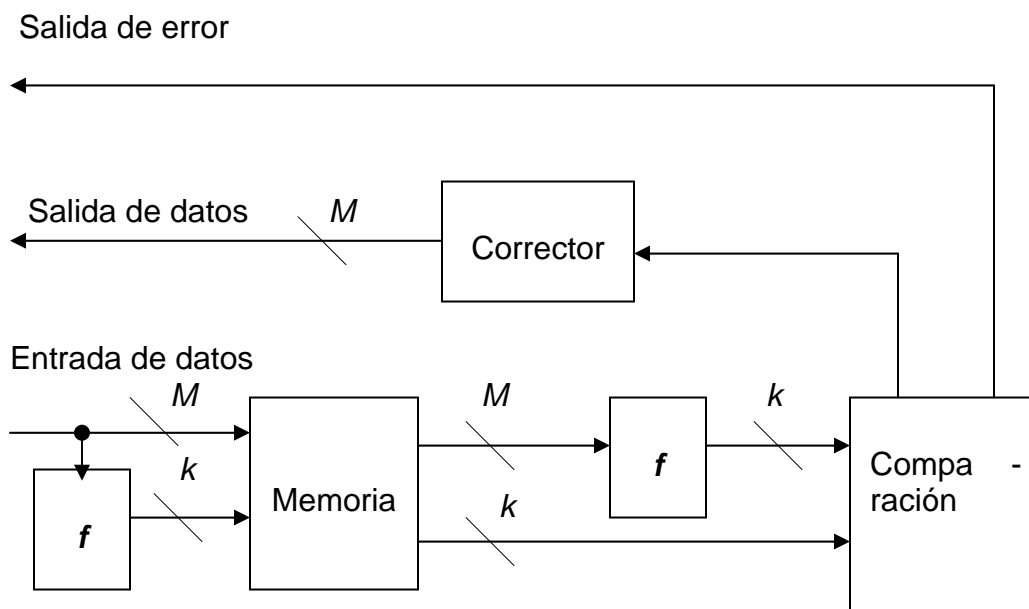


Figura 2.3 Código corrector de errores.

Cuando se van a escribir datos en memoria, se realiza un cálculo con los datos representado por la función f , para producir un código. Se almacenan tanto los datos como el código en la memoria. De esta manera se va a almacenar una palabra de datos de M bits, y el código tiene una longitud de k bits, el tamaño real de la palabra almacenada es de $M + k$ bits.

Cuando se va a leer una palabra previamente almacenada, se utiliza el código para detectar errores, se genera un nuevo código de k bits a partir de los M bits de datos, que se comparará con los bits de código captados de memoria. De esta comparación se obtiene uno de los tres resultados posibles:

- No se detectan errores. Los bits de datos se envían al exterior.
- Se detecta un error y es posible corregirlo. Se dan a un corrector los bits de datos más los bits de corrección de error, lo que produce el conjunto reducido de M bits a ser enviados fuera.

- Se detecta un error, pero no es posible corregirlo. Se envía una señal de error.



Ejercicios propuestos

- 1.- Describa cómo se realiza la selección de la dirección requerida en el arreglo de memoria presentado en la figura 2.2, para el caso en que la memoria sea del tipo RAM.
- 2.- ¿Cuántos dispositivos de memoria de 4 K x 1 se requieren para proporcionar un complemento total de memoria para un procesador de 8 bits con 16 líneas de dirección? Represente gráficamente la conexión adecuada de la memoria al microprocesador.



Consulta en otros libros

[Sayers 1995] Incluye información sobre arreglos de memoria para un microprocesador.

[Satallings 2000] Proporciona un tratamiento amplio sobre las memorias semiconductoras, incluyendo la interfaz RAMLink, la cual se centra en la interfaz procesador/memoria, en lugar de en la arquitectura interna de los chips DRAM.

[Lameda 1984] presenta variados ejemplos sobre el diseño de los sistemas de memoria.



Consulta en la Web

<http://ortihuela.galeon.com/ram.htm>: Incluye información sobre los diversos tipos de memoria RAM.

http://es.wikipedia.org/wiki/Memoria_RAM: Presenta información relevante sobre los tipos de memoria RAM y sus aplicaciones.

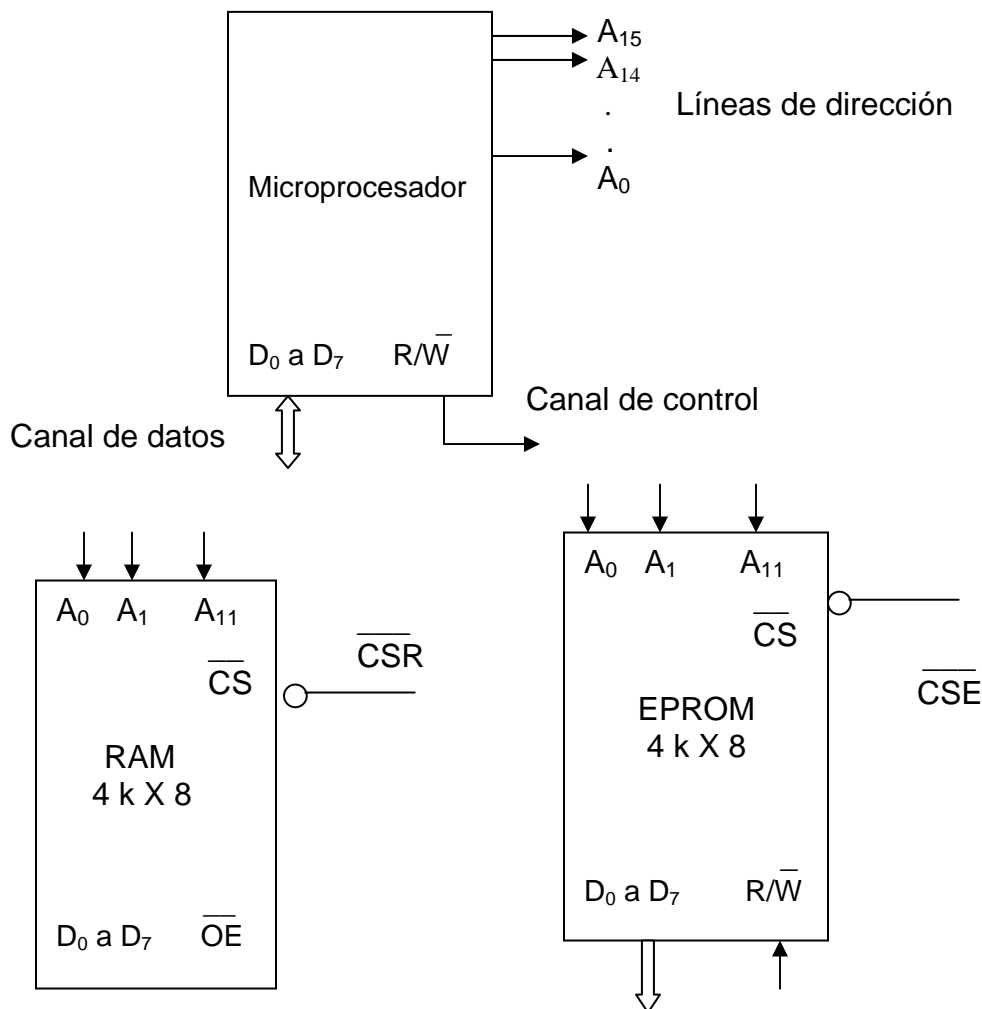


Ejercicios de Autoevaluación

- 1.- Configure la memoria de un sistema con una longitud de palabra de 16 bits. La cantidad de memoria RAM requerida es de 1 K empezando en la posición 000000 H. Y una cantidad de memoria ROM (EPROM) de 4K empezando en la

posición 30F000 H. Para ello se dispone de circuitos EPROM de 2K x 8 y circuitos RAM de 256 x 8. Presente también la tabla de decodificación necesaria.

- 2.- Se pide diseñar un sistema de microprocesador que pueda direccionar 64 K localizaciones de memoria. La memoria deberá estar conformada por 4 K x 8 de EPROM y 4 K x 8 de RAM. El microprocesador a utilizar dispone de 16 líneas de dirección, y las señal de control R/W (cuando se desea leer un dato de memoria se activa la señal R mediante un 1 lógico, por el contrario si se desea escribir en la memoria RAM la señal W deberá tomar el valor de 0 lógico). A continuación se presenta el microprocesador y los dispositivos de memoria a utilizar:



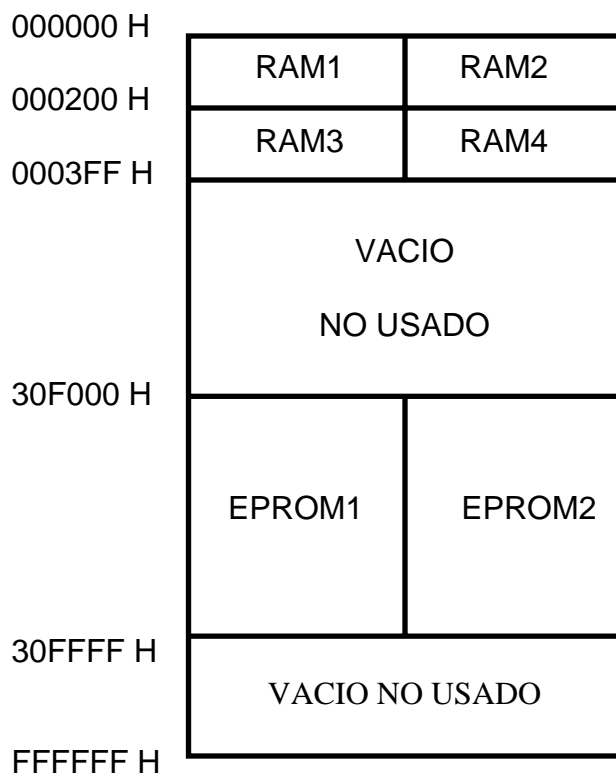
En base al planteamiento dado anteriormente presenta la conexión adecuada de los dispositivos de memoria al microprocesador para lograr la configuración deseada.



Respuesta a los Ejercicios de Autoevaluación

- 1.- Como las direcciones contienen tres bytes, se requieren 24 líneas de dirección (desde A₀ hasta A₂₄). La cantidad de dispositivos RAM a utilizar es de 4 (ya que

4 x 256 bytes = 1024 bytes = 1k) y dos dispositivos EPROM. El sistema de memoria así configurado será el siguiente:



Y la tabla de decodificación será la siguiente:

A ₃	A ₂	A ₂	A ₂	A ₁	A ₁	A ₁	A ₁	A ₁	A ₁	A ₁	A ₁	A ₁	A ₁	A ₁	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	0	RAM1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	1	RAM2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	X	X	X	X	X	X	X	X	X	0	RAM3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	X	X	X	X	X	X	X	X	X	1	RAM4
0	0	1	1	0	0	0	0	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	0	EPRO M1
0	0	1	1	0	0	0	0	1	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	1	EPRO M2

- 2.- Para direccionar la memoria sólo se requieren utilizar 12 líneas de dirección ($2^{12} = 4\text{ K}$) y para evitar que los dispositivos RAM y ROM coloquen información en el canal de datos, cuando las señales CSE y CSR toman el valor de cero lógico, es necesario distinguir entre las dos memorias mediante estas señales. Por medio de las líneas de dirección no usadas A₁₂ a A₁₅ se puede determinar el estado de las líneas de selección de los chips. Debe obtenerse una relación entre las líneas de dirección restantes de modo que ambas líneas de selección de chip jamás estén bajas al mismo tiempo. Esto se logra mediante la decodificación de direcciones. La decodificación se puede obtener de acuerdo a lo siguiente:

