

# FUNDAMENTOS DE SISTEMAS DIGITALES

Novena Edición

THOMAS L. FLOYD

**Traducción**

Vuelapluma

**Revisión Técnica**

Eduardo Barrera López de Turiso

*Departamento de Sistemas Electrónicos y de Control*

*Universidad Politécnica de Madrid*



Madrid ● México ● Santa Fe de Bogotá ● Buenos Aires ● Caracas ● Lima  
Montevideo ● San Juan ● San José ● Santiago ● São Paulo ● White Plains ●

# 10

## MEMORIAS Y ALMACENAMIENTO

### CONTENIDO DEL CAPÍTULO

- 10.1 Principios de las memorias semiconductoras
- 10.2 Memorias de acceso aleatorio (RAM)
- 10.3 Memorias de sólo lectura (ROM)
- 10.4 Memorias ROM programables (PROM y EPROM)
- 10.5 Memorias flash
- 10.6 Expansión de memorias
- 10.7 Tipos especiales de memorias
- 10.8 Memorias ópticas y magnéticas
- 10.9 Localización de averías y pruebas

■ ■ ■ Aplicación a los sistemas digitales

### OBJETIVOS DEL CAPÍTULO

- Definir las características básicas de las memorias.
- Explicar qué es una memoria RAM y cómo funciona.
- Explicar la diferencia entre RAM estática (SRAM, *Static RAM*) y RAM dinámica (DRAM, *Dynamic RAM*).
- Explicar qué es una memoria ROM y cómo funciona.
- Describir los distintos tipos de memorias PROM.
- Estudiar las características de las memorias flash.



- Describir la expansión de las memorias ROM y RAM para aumentar la longitud y capacidad de palabra.
- Describir los tipos especiales de memorias, como las FIFO y las LIFO
- Describir la organización básica de las cintas y discos magnéticos.
- Describir el funcionamiento básico de los discos magneto-ópticos y de los discos ópticos.
- Describir los métodos básicos para probar las memorias.
- Desarrollar diagramas de flujo para probar las memorias.
- Aplicar las memorias en un sistema digital.

## PALABRAS CLAVE

- Byte
- Palabra
- Celda
- Dirección
- Capacidad
- Escritura
- Lectura
- RAM
- ROM
- SRAM
- Bus
- DRAM
- PROM
- EPROM
- Memoria flash
- FIFO
- LIFO
- Disco duro

## INTRODUCCIÓN

En el Capítulo 9 se han tratado los registros de desplazamiento, que son un tipo de dispositivos de almacenamiento; de hecho, un registro de desplazamiento es, esencialmente, una memoria a pequeña escala. Los dispositivos de memoria que se cubren en este capítulo se utilizan generalmente para almacenamiento a más largo plazo y de cantidades más grandes de datos de lo que los registros son capaces de permitir.

Las computadoras y otros tipos de sistemas requieren el almacenamiento permanente o semipermanente de un gran número de datos binarios. Los sistemas basados en microprocesador necesitan de los dispositivos de almacenamiento y de las memorias para su funcionamiento, debido a la necesidad de almacenar los programas y mantener los datos generados durante el procesamiento.

En la terminología informática, normalmente el término *memoria* hace referencia a las memorias RAM y ROM y el término *almacenamiento* hace referencia al disco duro, a los discos flexibles y al CD-ROM. En este capítulo se estudian las memorias semiconductoras, y los medios de almacenamiento magnéticos y ópticos.

## ■■■ APLICACIÓN A LOS SISTEMAS DIGITALES

La aplicación a los sistemas digitales del final del capítulo completa el sistema de seguridad del Capítulo 9. Este capítulo se centra en la parte de la lógica de memoria del sistema, que almacena el código de entrada. Una vez que la lógica de memoria se haya desarrollado, se conectará con la lógica de introducción del código del Capítulo 9, para completar el sistema.

## 10.1 PRINCIPIOS DE LAS MEMORIAS SEMICONDUCTORAS

La memoria es la parte de un sistema que almacena datos binarios en grandes cantidades. Las memorias semiconductoras están formadas por matrices de elementos de almacenamiento que pueden ser *latches* o condensadores.

Al finalizar esta sección, el lector deberá ser capaz de:

- Explicar cómo almacena una memoria los datos binarios.
- Exponer la organización básica de una memoria.
- Describir la operación de escritura.
- Describir la operación de lectura.
- Describir la operación de direccionamiento.
- Explicar qué son las memorias RAM y ROM.

### Unidades de datos binarios: bits, bytes, *nibbles* y palabras

Como regla general, las memorias almacenan datos en unidades que tienen de uno a ocho bits. La unidad menor de datos binarios es, como ya sabemos, el **bit**. En muchas aplicaciones, se tratan los datos en unidades de 8 bits, denominadas **bytes** o en múltiplos de unidades de 8 bits. El byte se puede dividir en dos unidades de 4 bits, que reciben el nombre de **nibbles**. Una unidad completa de información se denomina **palabra** y está formada, generalmente, por uno o más bytes. Algunas memorias almacenan datos en grupos de 9 bits; un grupo de 9 bits consta de un byte más un bit de paridad.



#### NOTAS INFORMÁTICAS

La definición general de *palabra* dice que una palabra es una unidad completa de información, consistente en una unidad de datos binarios. Cuando se aplica a las instrucciones de una computadora, una palabra se define de forma más específica como dos bytes (16 bits). Como parte muy importante del lenguaje ensamblador utilizado por las computadoras, la directiva DW (*Define Word*) significa que se definen datos en unidades de 16 bits. Esta definición es independiente del microprocesador o del tamaño de su bus de datos. El lenguaje ensamblador también permite definiciones de bytes (8 bits) con la directiva DB, de palabras dobles (32 bits) con la instrucción DD y de palabras cuádruples (64 bits) con la instrucción QD.

### Matriz de memoria semiconductora básica

Cada elemento de almacenamiento en una memoria puede almacenar un 1 o un 0 y se denomina **celda**. Las memorias están formadas por matrices de celdas, como se ilustra en la Figura 10.1, en la que se utilizan 64

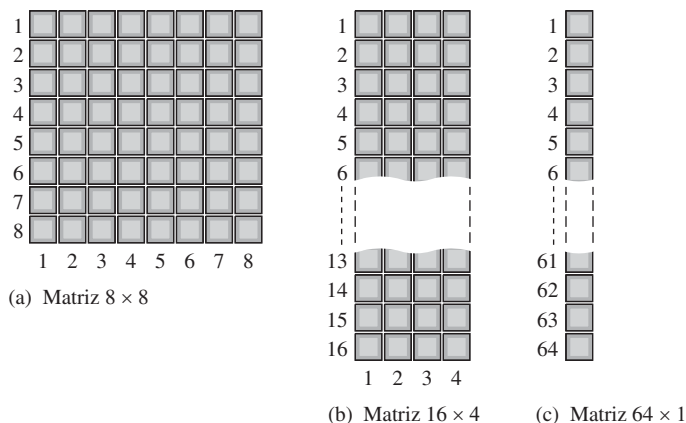


FIGURA 10.1 Matriz de almacenamiento de 64 celdas, organizada de tres formas diferentes.

celdas a modo de ejemplo. Cada bloque de la **matriz de memoria** representa una celda de almacenamiento y su situación se puede especificar mediante una fila y una columna.

La matriz de 64 celdas se puede organizar de muchas maneras en función de las unidades de datos. La Figura 10.1(a) muestra una matriz de  $8 \times 8$ , que se puede entender como una memoria de 64 bits o como una memoria de 8 bytes. La parte (b) nos muestra una matriz de  $16 \times 4$ , que es una memoria de 16 *nibbles* y la parte (c) presenta una matriz de  $64 \times 1$  que es una memoria de 64 bits. Una memoria se identifica mediante el número de palabras que puede almacenar, multiplicado por el tamaño de la palabra. Por ejemplo, una memoria de  $16k \times 8$  puede almacenar 16.384 palabras de ocho bits. La incoherencia en la expresión anterior es común en la terminología de las memorias. En realidad, el número de palabras es siempre una potencia de 2 que, en este caso, es  $2^{14} = 16.384$ . Sin embargo, es una práctica común expresar cada número redondeado al millar más próximo, en este caso 16k.

## Dirección y capacidad de las memorias

La posición de una unidad de datos en una matriz de memoria se denomina **dirección**. Por ejemplo, en la Figura 10.2(a), la dirección de un bit en la matriz de dos dimensiones se especifica mediante la fila y columna en que está, tal como se muestra. En la Figura 10.2(b), la dirección de un byte se especifica únicamente mediante la fila. Como puede ver, la dirección depende de cómo se organice la memoria en unidades de datos. Las computadoras personales disponen de memorias organizadas en bytes. Esto significa que el grupo más pequeño de bits que se puede direccionar es ocho.

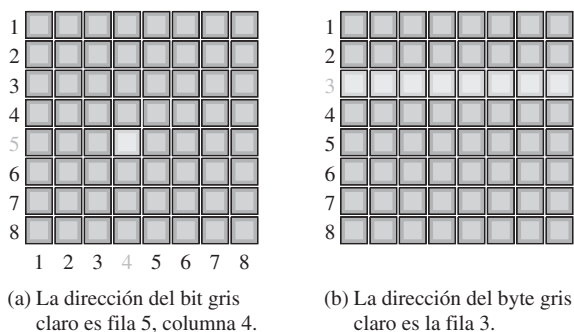


FIGURA 10.2 Ejemplos de direcciones de memoria en una matriz bidimensional.

En la Figura 10.3, la dirección de un byte en una matriz tridimensional se especifica mediante la fila y la columna correspondientes. En este caso, el grupo más pequeño de bits al que se puede acceder es ocho.

La **capacidad** de una memoria es el número total de unidades de datos que puede almacenar. Por ejemplo, en la matriz de memoria organizada en bits de la Figura 10.2(a), la capacidad total es de 64 bits. En la matriz de memoria organizada en bytes de la Figura 10.2(b), la capacidad es de 8 bytes, que es lo mismo que 64 bits. En la Figura 10.3, la capacidad es de 64 bytes. Típicamente, las memorias de computadora disponen de 256 MB (MB es megabyte), o más, de memoria interna.

## Operaciones básicas de las memorias

Puesto que una memoria almacena datos binarios, los datos deben introducirse en la memoria y deben poder recuperarse cuando se necesiten. La operación de **escritura** coloca los datos en una posición específica de la memoria y la operación de **lectura** extrae los datos de una dirección específica de memoria. La operación de

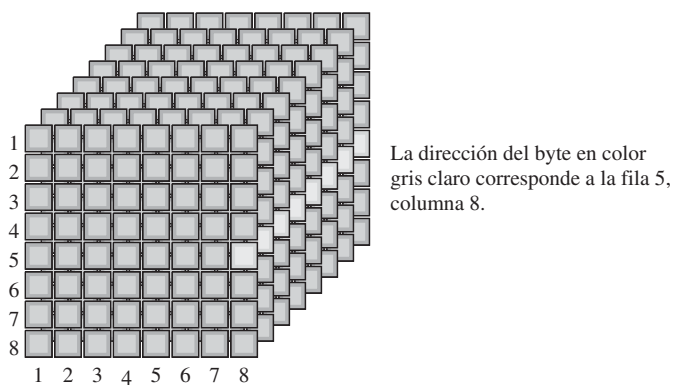
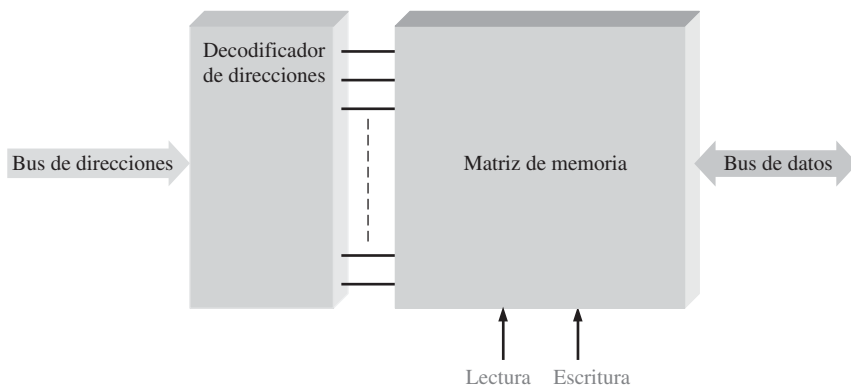
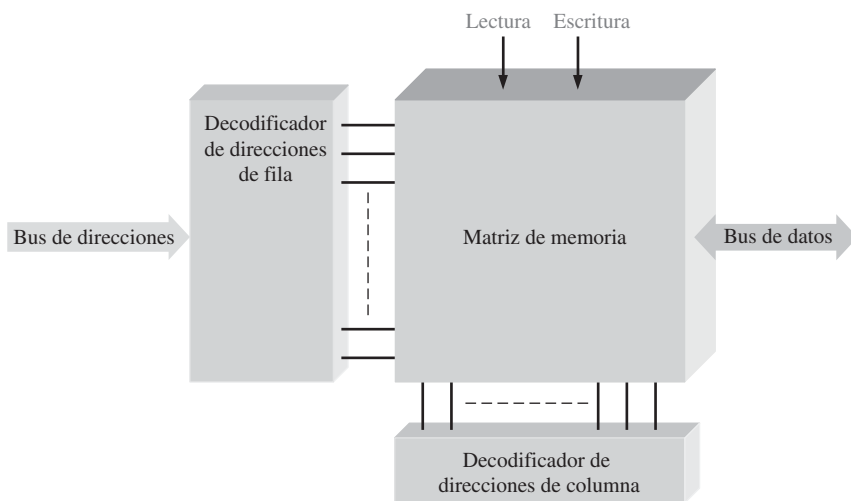


FIGURA 10.3 Ejemplo de dirección de memoria en una matriz de tres dimensiones.



(a) Matriz de memoria bidimensional



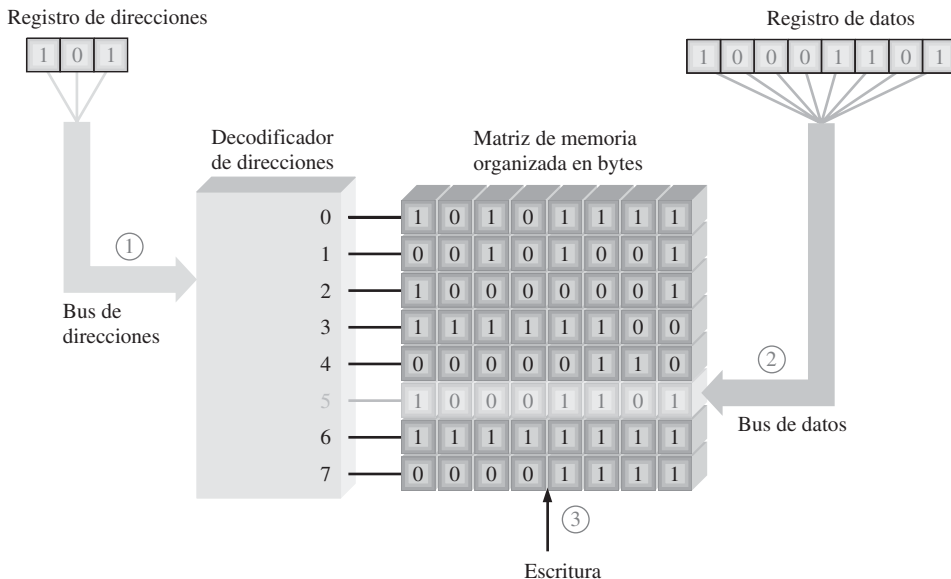
(b) Matriz de memoria de tres dimensiones

FIGURA 10.4 Diagramas de bloques de una memoria de 2 dimensiones y de otra de 3 dimensiones, mostrando el bus de direcciones, el decodificador de direcciones, el bus de datos bidireccional y las entradas de lectura/escritura.

direccionamiento, que forma parte tanto de la operación de lectura como de la de escritura, selecciona la dirección de memoria específica.

Las unidades de datos se introducen en la memoria durante la operación de escritura y se extraen de la memoria durante la operación de lectura a través de un conjunto de líneas que se denominan *bus de datos*. Como se indica en la Figura 10.4, el bus de datos es bidireccional, lo que significa que los datos pueden ir en cualquiera de las dos direcciones (hacia la memoria o desde la memoria). En el caso de una memoria organizada en bytes, el bus de datos tiene al menos ocho líneas, de manera que los ocho bits de una dirección seleccionada se transmiten en paralelo. En una operación de escritura o de lectura, se selecciona una dirección introduciendo un código binario, que representa la dirección deseada, en un conjunto de líneas denominado *bus de direcciones*. El código de dirección se decodifica internamente y de esa forma se selecciona la dirección adecuada. En el caso de la matriz de memoria de 3 dimensiones de la Figura 10.4(b) se usan dos decodificadores, uno para las filas y otro para las columnas. El número de líneas del bus de direcciones depende de la capacidad de la memoria. Por ejemplo, un código de dirección de 15 bits puede seleccionar 32.768 posiciones ( $2^{15}$ ) en la memoria; un código de dirección de 16 bits puede seleccionar 65.536 ( $2^{16}$ ) posiciones de memoria, etc. En las computadoras personales, un bus de direcciones de 32 bits puede seleccionar 4.294.967.296 ( $2^{32}$ ) posiciones, lo que se expresa como 4 G.

**La operación de escritura.** En la Figura 10.5 se muestra la operación de escritura simplificada. Para almacenar un byte de datos en memoria, se introduce en el bus de direcciones un código que se encuentra almacenado en el registro de direcciones. Una vez que el código de dirección está ya en el bus, el decodificador de direcciones decodifica la dirección y selecciona la posición de memoria especificada. La memoria recibe entonces una orden de escritura y los datos almacenados en el registro de datos se introducen en el bus de datos, y se almacenan en la dirección de memoria especificada, completándose así la operación de escritura. Cuando se



- ① El código de dirección 101 se coloca en el bus de direcciones y se selecciona la dirección 5.
- ② El byte de datos se coloca en el bus de datos.
- ③ El comando de escritura hace que el byte de datos se almacene en la dirección 5, reemplazando a los datos anteriores.

FIGURA 10.5 Ilustración de la operación de escritura.

escribe un nuevo byte de datos en una dirección de memoria, se sobrescribe y destruye el byte de datos actualmente almacenado en esa dirección.

**La operación de lectura.** En la Figura 10.6 se muestra la operación de lectura simplificada. De nuevo, se introduce en el bus de direcciones un código almacenado en el registro de direcciones. Una vez que el código de dirección se encuentra en el bus, el decodificador de direcciones decodifica la dirección y selecciona la posición especificada de la memoria. La memoria recibe entonces una orden de lectura, y una “copia” del byte de datos almacenado en la dirección de memoria seleccionada se introduce en el bus de datos y se carga en el registro de datos, finalizando así la operación de lectura. Cuando se lee un byte de datos de una dirección de memoria, éste sigue almacenado en dicha dirección. Esto se denomina *lectura no destructiva*.

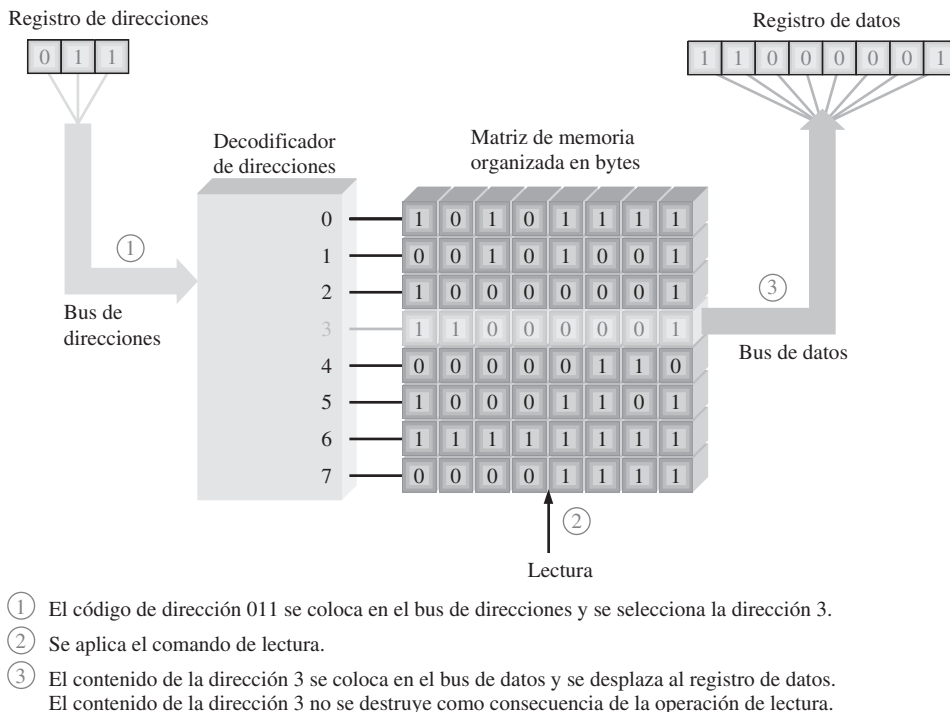


FIGURA 10.6 Ilustración de la operación de lectura.

## Las memorias RAM y ROM

Las dos principales categorías de memorias semiconductoras son las memorias RAM y ROM. La memoria **RAM** (*Random-Access Memory*, memoria de acceso aleatorio) es un tipo de memoria en la que se tarda lo mismo en acceder a cualquier dirección de memoria y éstas se pueden seleccionar en cualquier orden, tanto en una operación de lectura como de escritura. Todas las RAM poseen la capacidad de *lectura* y *escritura*. Debido a que las memorias RAM pierden los datos almacenados cuando se desconecta la alimentación, reciben el nombre de memorias **volátiles**.

La memoria **ROM** (*Read-Only Memory*, memoria de sólo lectura) es un tipo de memoria en la que los datos se almacenan de forma permanente o semipermanente. Los datos se pueden leer de una ROM, pero no existe la operación de escritura como en las RAM. La ROM, al igual que la RAM, es una memoria de acceso aleatorio pero, tradicionalmente, el término RAM se reserva para las memorias de acceso aleatorio de *lec-*



*tura/escritura.* En este capítulo veremos varios tipos de memorias RAM y ROM. Debido a que las ROM mantienen los datos almacenados incluso si se desconecta la alimentación, reciben el nombre de memorias **no volátiles**.

## REVISIÓN DE LA SECCIÓN 10.1

*Las respuestas se encuentran al final del capítulo*

1. ¿Cuál es la unidad más pequeña de datos que se puede almacenar en una memoria?
2. ¿Cuál es la capacidad en bits de una memoria que puede almacenar 256 bytes de datos?
3. ¿En qué consiste la operación de escritura?
4. ¿En qué consiste la operación de lectura?
5. ¿Cómo se localiza una determinada unidad de datos en una memoria?
6. Describir las diferencias entre una RAM y una ROM.

## 10.2 MEMORIAS DE ACCESO ALEATORIO (RAM)

Las RAM son memorias de lectura-escritura en las que los datos se pueden escribir o leer en cualquier dirección seleccionada en cualquier secuencia. Cuando se escriben los datos en una determinada dirección de la RAM, los datos almacenados previamente son reemplazados por la nueva unidad de datos. Cuando una unidad de datos se lee de una determinada dirección de la RAM, los datos de esa dirección permanecen almacenados y no son borrados por la operación de lectura. Esta operación no destructiva de lectura se puede entender como una copia del contenido de una dirección, dejando dicho contenido intacto. La RAM se utiliza habitualmente para almacenamiento de datos a corto plazo, ya que no puede conservar los datos almacenados cuando se desconecta la alimentación.

Al finalizar esta sección, el lector deberá ser capaz de:

- Nombrar las dos categorías de RAM. ■ Explicar qué es una SRAM. ■ Describir la celda de almacenamiento de una SRAM. ■ Explicar la diferencia entre una SRAM asíncrona y una SRAM de ráfaga síncrona. ■ Explicar qué es una DRAM. ■ Describir la celda de almacenamiento de una DRAM. ■ Explicar los tipos de DRAM. ■ Comparar la SRAM con la DRAM.

### La familia de memorias RAM

Las dos categorías de memorias RAM son la *RAM estática* (SRAM) y la *RAM dinámica* (DRAM). Las RAM estáticas utilizan generalmente *latches* como elementos de almacenamiento y, por tanto, pueden almacenar datos de forma indefinida *siempre que se aplique una alimentación continua*. Las RAM dinámicas utilizan condensadores como elemento de almacenamiento y no pueden mantener los datos mucho tiempo sin recargar los condensadores mediante el proceso de **refresco**. Tanto las SRAM como las DRAM perderán los datos cuando se elimine la alimentación continua, por lo que se clasifican como memorias volátiles.

Los datos pueden leerse mucho más rápidamente en una SRAM que en una DRAM. Sin embargo, las DRAM pueden almacenar muchos más datos que las SRAM para un tamaño físico y coste dados, ya que la celda de las DRAM es mucho más sencilla y se pueden incluir muchas más celdas en un área determinada que en una memoria SRAM.

Los tipos básicos de memorias SRAM son las memorias *SRAM asíncronas* y las *SRAM síncronas de ráfaga*. Los tipos básicos de DRAM son la *DRAM con modo página rápido* (*Fast Page Mode, FPM DRAM*), la *DRAM con salida de datos extendida* (*Extended Data Output, EDO DRAM*), la *DRAM con salida de datos extendida en ráfaga* (*Burst Extended Data Output, BEDO DRAM*) y la *DRAM síncrona* (*Synchronous, SDRAM*). Todas ellas se muestran en la Figura 10.7.

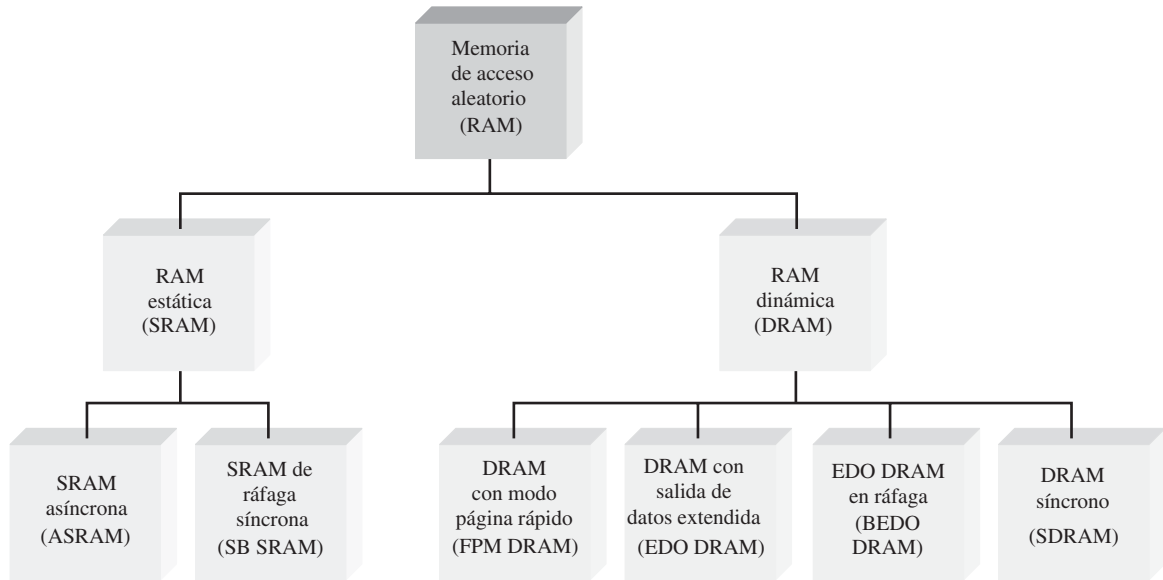
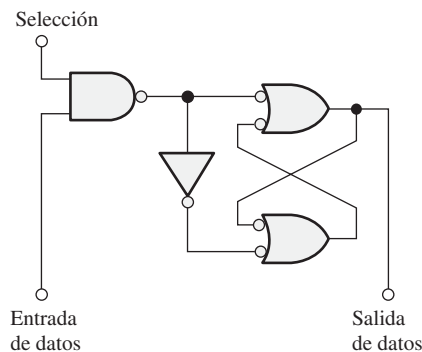


FIGURA 10.7 La familia de memorias RAM.

## RAM estática (SRAM)

**Celda de memoria.** Todas las RAM estáticas se caracterizan por las celdas de memoria *latch*. Cuando se aplica alimentación continua a una celda de **memoria estática** se puede mantener un estado 1 o 0 indefinidamente. Si se retira la alimentación, el bit de datos almacenado se perderá.

La Figura 10.8 muestra una celda de memoria de tipo *latch* para la **SRAM**. La celda se selecciona mediante un nivel activo en la línea Selección de bit y un bit de datos (1 o 0) se escribe en la celda colocándolo en la línea Entrada de datos. Un bit de datos se puede leer extrayéndolo de la línea Salida de datos.

FIGURA 10.8 Celda típica de memoria *latch* de una SRAM.

**Matriz básica de celdas de memoria estáticas.** Las celdas de almacenamiento en una SRAM se organizan en filas y columnas, como se ilustra en la Figura 10.9 para el caso de una matriz  $n \times 4$ . Todas las celdas de una misma fila comparten la misma línea Seleccionar Fila. Cada conjunto de líneas Entrada de datos y Salida de datos

van a cada celda situada en una determinada columna y se conectan a una única línea de datos, que sirve como entrada y salida (E/S datos), a través de los buffers de entrada y salida de datos.

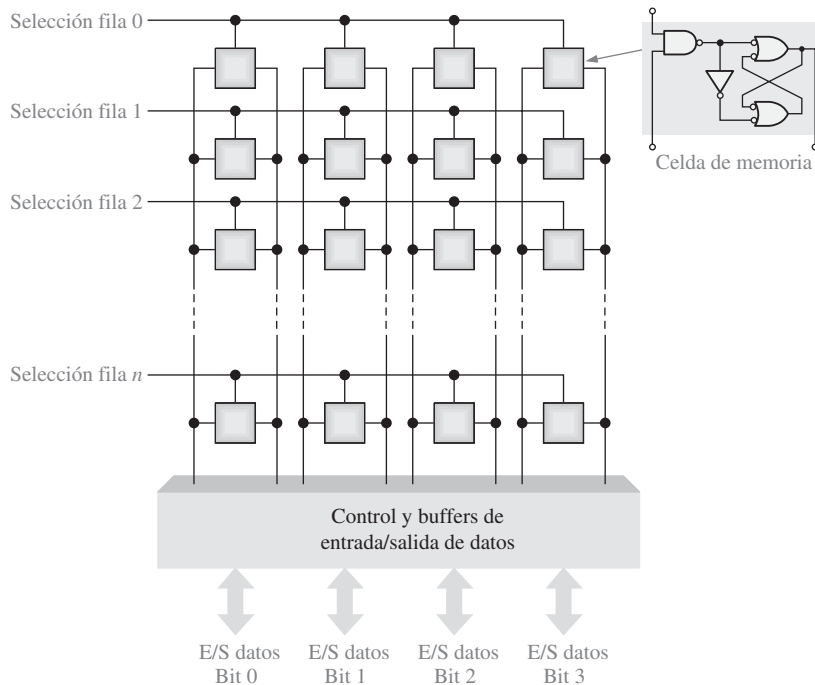


FIGURA 10.9 Matriz SRAM básica.

Para escribir una unidad de datos, en este caso un grupo de 4 bits, en una fila de celdas determinada de la matriz de memoria, la línea Selección Fila se pone en su estado activo y los cuatro bits de datos se colocan en las líneas de entrada de datos. La línea de escritura (*write*) se pone entonces en estado activo, lo que da lugar a que cada bit de datos se almacene en una celda seleccionada en la columna asociada. Para leer una unidad de datos, se pone en estado activo la línea de lectura (*read*), lo que hace que los cuatro bits de datos almacenados en la fila seleccionada aparezcan en las líneas de E/S de datos.

## Organización de la SRAM asíncrona básica

Una SRAM asíncrona es aquella en la que su funcionamiento no está sincronizado con un reloj de sistema. Para ilustrar la organización general de una SRAM, vamos a utilizar una memoria de  $32\text{ K} \times 8$  bits. En la Figura 10.10 se muestra el símbolo lógico de esta memoria.

En el modo de lectura (READ), los 8 bits de datos que se almacenan en una dirección determinada aparecen en las líneas de salida de datos. En el modo de escritura (WRITE), los 8 bits de datos que se aplican a las líneas de entrada de datos se almacenan en la dirección seleccionada. Las líneas de entrada y salida de datos ( $E/S_1$  a  $E/S_8$ ) son las mismas líneas. Durante la operación de lectura, éstas actúan como líneas de salida ( $S_1$  a  $S_8$ ) y durante la operación de escritura actúan como líneas de entrada ( $E_1$  a  $E_8$ ).

**Salidas tri-estado y buses.** Los *buffers* tri-estado en una memoria permiten que las líneas de datos actúen como líneas de entrada o salida y conectan la memoria con el bus de datos en una computadora. Estos buffers tienen tres posibles estados de salida: ALTO (1), BAJO (0) y ALTA-Z (alta impedancia, abierto). Las salidas

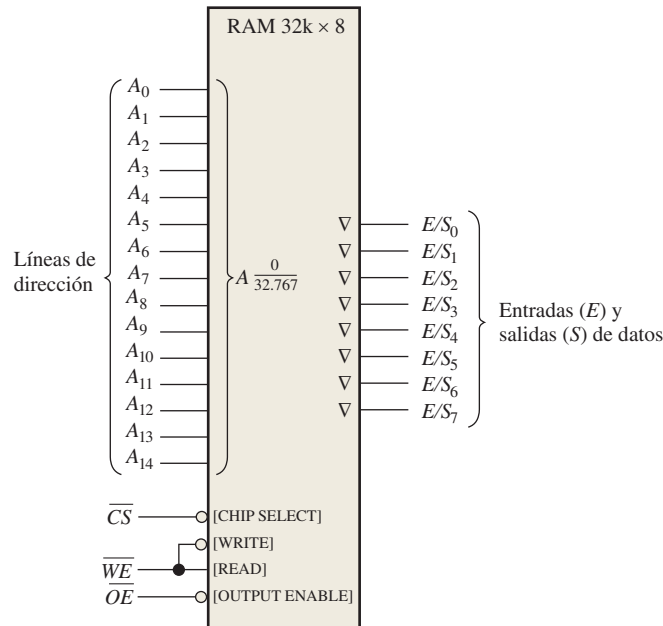


FIGURA 10.10 Diagrama lógico de una SRAM asíncrona de 32 k × 8.

tri-estado se indican en los símbolos lógicos mediante un pequeño triángulo invertido ( $\nabla$ ), como se muestra en la Figura 10.9, y se utilizan por compatibilidad con las estructuras de bus, como las que se encuentran en los sistemas basados en microprocesador.

Físicamente, un bus es un conjunto de caminos conductores que sirven para interconectar dos o más componentes funcionales de un sistema o de varios sistemas diferentes. Eléctricamente, un bus es una colección de señales y de niveles de tensión y/o corriente específicos que permiten a los distintos dispositivos conectados al bus comunicarse y funcionar correctamente.

Por ejemplo, un microprocesador se conecta a las memorias y dispositivos de entrada/salida mediante determinadas estructuras de bus. Un bus de direcciones permite al microprocesador direccionar las memorias, y el bus de datos sirve para la transferencia de datos entre el microprocesador, las memorias y los dispositivos de entrada/salida, tales como monitores, impresoras, teclados y modems. El bus de control permite al microprocesador controlar la temporización y la transferencia de datos entre los distintos componentes.

**Matriz de memoria.** Los chips SRAM se pueden organizar en bits, *nibbles* (4 bits), bytes (8 bits) o múltiplos de bytes (16, 24 o 32 bits).

La Figura 10.11 muestra la organización de una SRAM típica de 32 k × 8. La matriz de celdas de memoria está organizada en 256 filas y 128 columnas, cada una de 8 bits, como se muestra en la parte (a) de la figura. En la práctica, dispone de  $2^{15} = 32.768$  direcciones, y cada dirección contiene 8 bits. La capacidad de esta memoria ejemplo es entonces de 32.768 bytes (lo que normalmente se expresa como 32 kB).

La SRAM de la Figura 10.11(b) trabaja del siguiente modo. En primer lugar, la entrada de habilitación del chip,  $\overline{CS}$ , debe estar a nivel BAJO para que la memoria funcione. Ocho de las quince líneas de dirección se decodifican en el decodificador de filas, de modo que se selecciona una de las 256 filas. Las restantes siete líneas de dirección las decodifica el decodificador de columnas, de modo que se selecciona una de las 128 columnas de 8 bits.

**Lectura.** En el modo lectura (READ), la entrada de habilitación de escritura  $\overline{WE}$  está a nivel ALTO y la salida de habilitación  $\overline{OE}$  está a nivel BAJO. La puerta  $G_1$  desactiva los buffers de entrada, y la puerta  $G_2$  activa

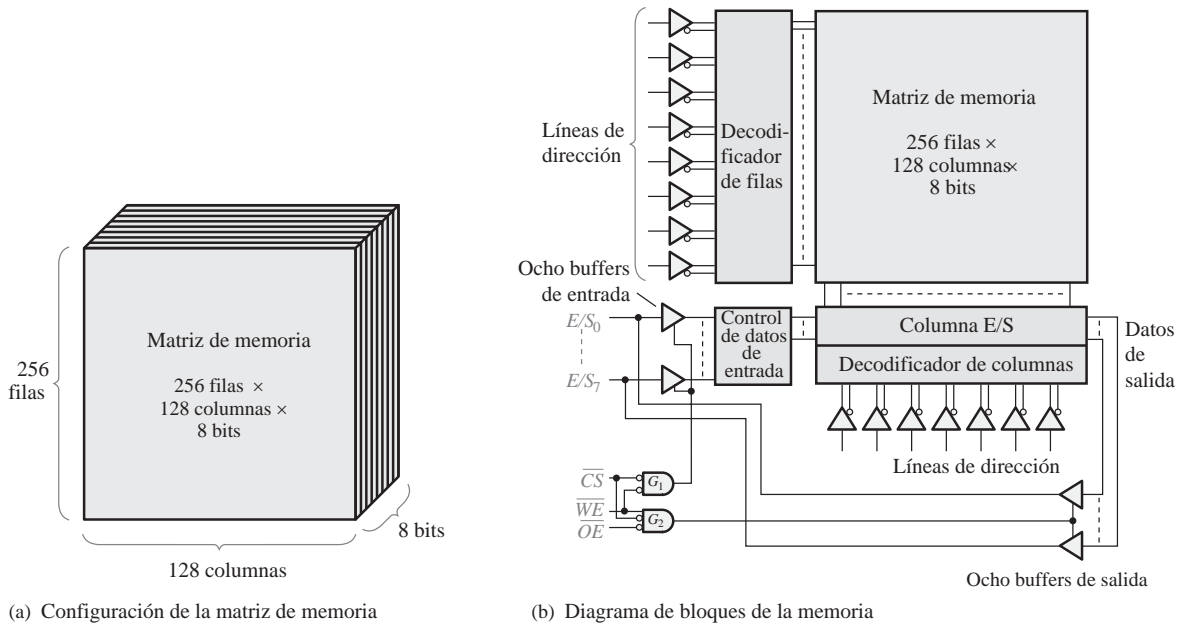


FIGURA 10.11 Organización básica de una SRAM asíncrona de 32k × 8.

los buffers de tres estados de salida de las columnas. Por tanto, los ocho bits de datos almacenados en la dirección seleccionada se llevan a través de las E/S de las columnas hasta las líneas de datos ( $E/S_1$  a  $E/S_8$ ), que actúan como líneas de salida de datos.

**Escritura.** En el modo escritura (WRITE),  $\overline{WE}$  está a nivel BAJO y  $\overline{OE}$  está a nivel ALTO. La puerta  $G_1$  activa los buffers de entrada, y la puerta  $G_2$  desactiva los buffers de salida. Por tanto, los ocho bits de datos de entrada de las líneas de datos se llevan a través del control de datos de entrada y de la E/S de columna a la dirección seleccionada, y se almacenan.

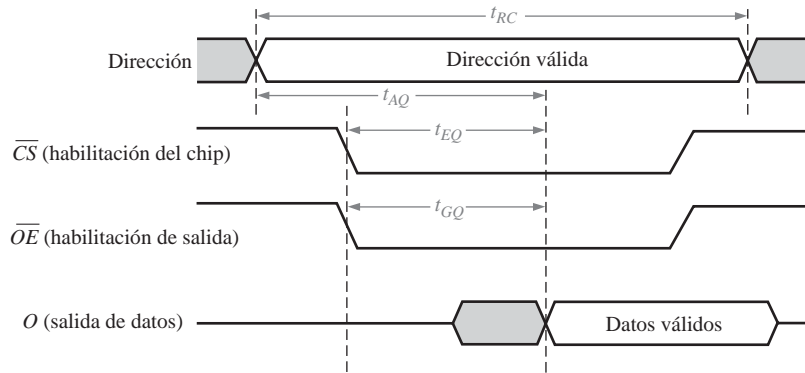
**Ciclos de lectura y escritura.** La Figura 10.12 muestra un diagrama de tiempos típico para un ciclo de lectura y un ciclo de escritura de una memoria. En la parte (a) se presenta el ciclo de lectura; se aplica un código de dirección válido a las líneas de dirección durante un intervalo de tiempo especificado, que se denomina *período del ciclo de lectura*,  $t_{RC}$ . A continuación, las entradas de habilitación del chip ( $\overline{CS}$ ) y de habilitación de salida ( $\overline{OE}$ ) pasan a nivel BAJO. Un intervalo de tiempo después de que la entrada  $\overline{OE}$  haya pasado a nivel BAJO, un byte de datos válido procedente de la dirección seleccionada se presenta en las líneas de datos. Este intervalo de tiempo se denomina *tiempo de acceso de la habilitación de salida*,  $t_{GQ}$ . Existen otros dos tiempos de acceso en el ciclo de lectura: el *tiempo de acceso de dirección*,  $t_{AQ}$ , que se mide desde el principio de una dirección válida hasta que los datos válidos aparecen en las líneas de datos, y el *tiempo de acceso de la habilitación del chip*,  $t_{EQ}$ , que se mide desde la transición de nivel ALTO a nivel BAJO de  $\overline{CS}$  hasta que los datos válidos aparecen en las líneas de datos.

En cada ciclo de lectura, se lee de la memoria una unidad de datos, en este caso un byte.

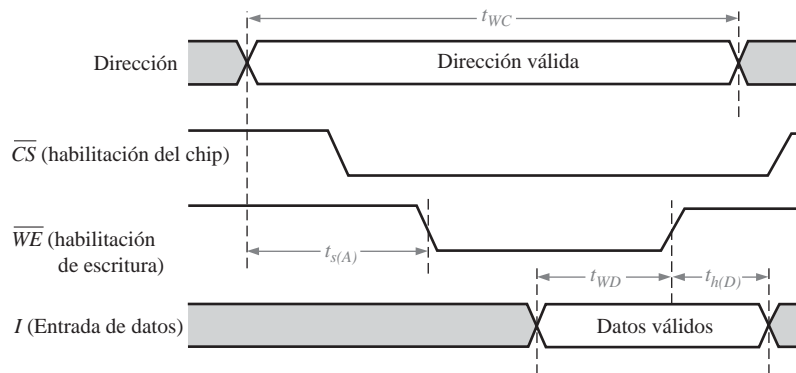
La Figura 10.12(b) muestra el ciclo de escritura. Un código de dirección válido se aplica a las líneas de dirección durante un intervalo de tiempo especificado que se denomina *período del ciclo de escritura*,  $t_{WC}$ . A continuación, las entradas de habilitación del chip ( $\overline{CS}$ ) y de habilitación de escritura ( $\overline{WE}$ ) pasan a nivel BAJO. El intervalo de tiempo requerido desde el inicio de una dirección válida hasta que la entrada  $\overline{WE}$  pasa a nivel BAJO se denomina *tiempo de establecimiento de dirección*,  $t_{S(A)}$ . El tiempo que la entrada  $\overline{WE}$  debe estar a nivel BAJO es la anchura del impulso de escritura. El tiempo que la entrada  $\overline{WE}$  debe permanecer a

nivel BAJO después de haber aplicado los datos válidos en las entradas de datos se designa por  $t_{WD}$ . El tiempo que los datos de entrada válidos deben permanecer en las líneas de entrada después de que la entrada  $\overline{WE}$  pase a nivel ALTO es el *tiempo de mantenimiento de datos*,  $t_{h(D)}$

En cada ciclo de escritura, se escribe en la memoria una unidad de datos.



(a) Ciclo de lectura ( $\overline{WE}$  a nivel ALTO).



(b) Ciclo de escritura ( $\overline{WE}$  a nivel BAJO)

FIGURA 10.12 Diagramas de tiempos de los ciclos de lectura y escritura básicos para la SRAM de la Figura 10.11.

## SRAM síncrona de ráfaga

A diferencia de la SRAM asíncrona, una SRAM síncrona está sincronizada con el reloj del sistema. Por ejemplo, en un sistema informático, la SRAM síncrona opera con la misma señal de reloj que el microprocesador, de modo que el microprocesador y la memoria están sincronizados para conseguir una operación más rápida.

El concepto fundamental en que se basa la naturaleza síncrona de una SRAM puede ilustrarse con la Figura 10.13, que es un diagrama de bloques simplificado de una memoria de  $32\text{ k} \times 8$ , con propósitos ilustrativos. La SRAM síncrona es muy similar a la SRAM asíncrona, en términos de la matriz de memoria, del decodificador de direcciones y de las entradas de lectura/escritura y activación. La diferencia fundamental es que la SRAM síncrona utiliza registros con señal de reloj para sincronizar todas las entradas con el reloj del sistema. Tanto la dirección, como la entrada de lectura/escritura, la señal de activación del chip y los datos de

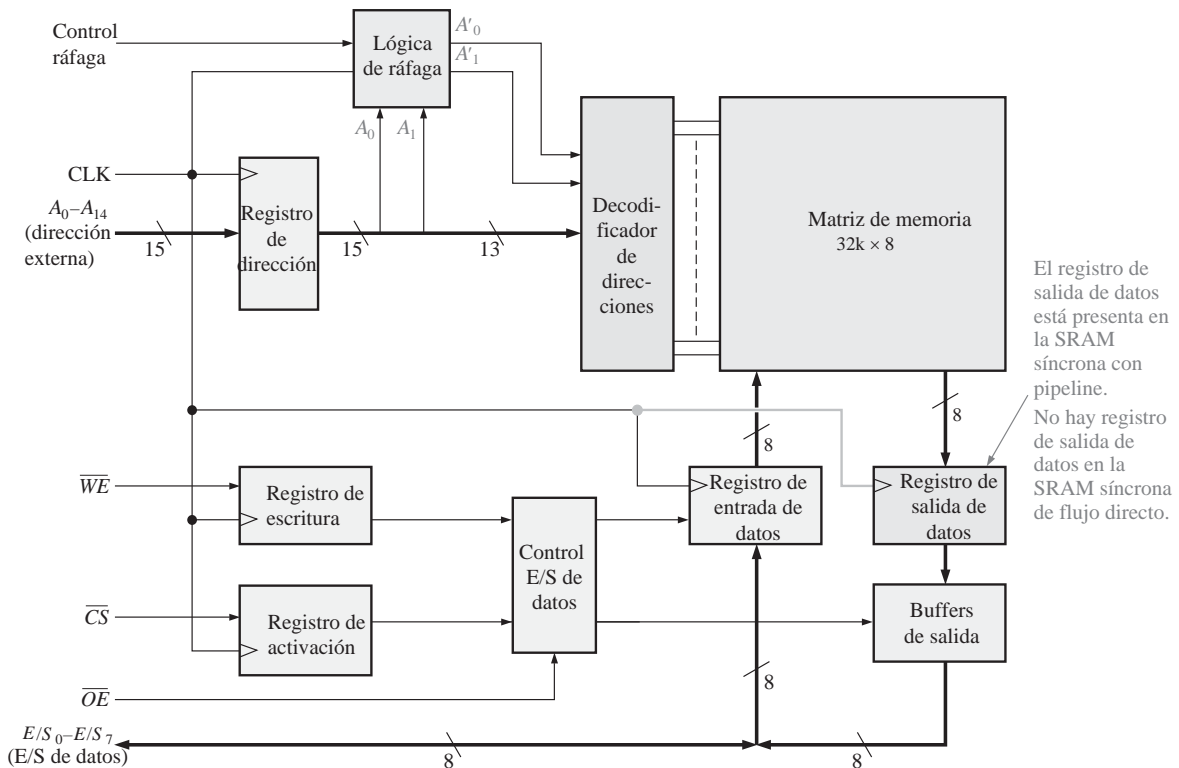
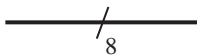


FIGURA 10.13 Diagrama de bloques básico de una SRAM de ráfaga síncrona.

entrada se enclavan en sus respectivos registros con el flanco activo del pulso de reloj. Una vez enclavada esta información, la operación de la memoria estará en sincronía con el reloj.

Para simplificar, se presenta en la Figura 10.13 una notación para expresar múltiples líneas paralelas o líneas de bus, como alternativa a dibujar cada línea por separado. Un conjunto de líneas paralelas puede expresarse mediante una sola línea gruesa atravesada por una barra y con una indicación del número de líneas distintas que forman el conjunto. Por ejemplo, la siguiente notación representa un conjunto de ocho líneas paralelas:



Los bits de dirección  $A_0$  a  $A_{14}$  se enclavan en el registro de dirección con el flanco positivo de un pulso de reloj. En el mismo pulso de reloj, el estado de las líneas de activación de escritura ( $\overline{WE}$ ) y de selección de chip ( $\overline{CS}$ ) se enclava en el registro de escritura y el registro de activación, respectivamente. Éstos son registros de un único bit o simplemente flip-flops. Asimismo, los datos de entrada son enclavados con el mismo pulso de reloj en el registro de datos de entrada para las operaciones de escritura, y los datos existentes en una dirección de memoria seleccionada se enclavan en el registro de salida de datos para las operaciones de lectura, según determine el control de E/S de datos, basándose en las entradas procedentes del registro de escritura, del registro de activación y de la línea de activación de salida ( $\overline{OE}$ ).

Existen dos tipos básicos de memoria SRAM síncronas: de *flujo directo* y con *pipeline*. La SRAM síncrona de flujo directo no dispone de un registro de salida de datos, por lo que los datos de salida fluyen asincrónicamente hacia las líneas de E/S de datos a través de los búferes de salida. La SRAM síncrona con *pipeline*

dispone de un registro de salida de datos, como se muestra en la Figura 10.13, por lo que los datos de salida se presentan síncronamente en las líneas de E/S de datos.

**Operación en modo ráfaga.** Como muestra la Figura 10.13, las memorias SRAM síncronas tienen normalmente una función de ráfaga de direcciones, que permite a la memoria leer o escribir en hasta cuatro posiciones utilizando una única dirección. Cuando se enclava una dirección externa en el registro de direcciones, los dos bits menos significativos de la dirección,  $A_0$  y  $A_1$ , se aplican al circuito de la lógica de ráfaga. Éste produce una secuencia de cuatro direcciones internas añadiendo 00, 01, 10 y 11 a los dos bits de dirección menos significativos en sucesivos pulsos de reloj. La secuencia comienza siempre con la dirección base, que es la dirección externa almacenada en el registro de dirección.

La lógica de la ráfaga de direcciones en una SRAM síncrona típica está compuesta por un contador binario y puertas OR-exclusiva, como muestra la Figura 10.14. Para una lógica de ráfaga de 2 bits, la secuencia interna de ráfaga de direcciones se forma a partir de los bits  $A_2$ - $A_{14}$  de la dirección base, más los dos bits de la dirección de ráfaga,  $A_1'$  y  $A_0'$ .

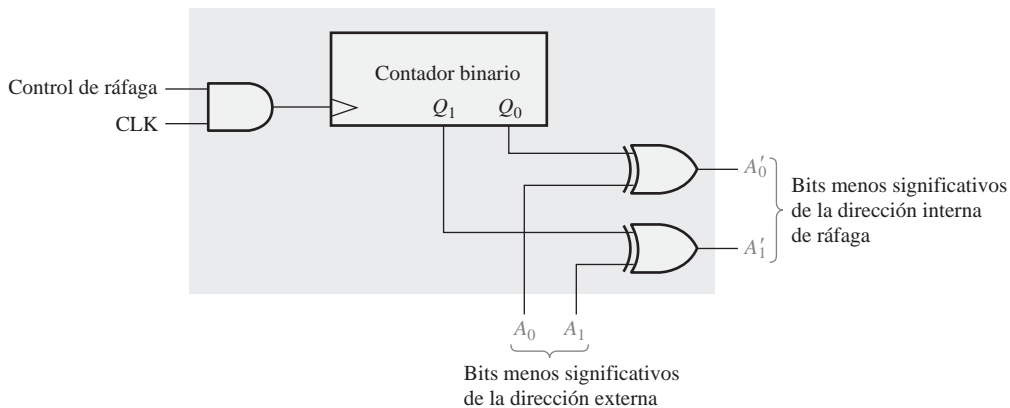


FIGURA 10.14 Lógica de la ráfaga de direcciones.

Al comenzar la secuencia de ráfaga, el contador se encuentra en su estado 00 y los dos bits menos significativos de la dirección se aplican a las entradas de las puertas XOR. Suponiendo que tanto  $A_0$  como  $A_1$  sean 0, los dos bits menos significativos de la secuencia de direcciones interna serían 00, 01, 10 y 11.

## Memoria caché

Una de las principales aplicaciones de las memorias SRAM es la implementación de memorias caché en computadoras. La **memoria caché** es una memoria de alta velocidad y relativamente pequeña que almacena los datos o instrucciones más recientemente utilizados de la memoria principal, más grande pero más lenta. La memoria caché puede también utilizar memoria RAM dinámica (DRAM), de la que hablaremos a continuación. Normalmente, la memoria SRAM es varias veces más rápida que la memoria DRAM. En conjunto, la memoria caché hace que el microprocesador pueda acceder a la información almacenada mucho más rápido que si sólo se empleara memoria DRAM de alta capacidad. La memoria caché es, básicamente, un método eficiente en términos de coste para mejorar el rendimiento del sistema sin tener que incurrir en el gasto de hacer que toda la memoria sea más rápida.

El concepto de memoria caché se basa en la idea de que los programas informáticos tienden a obtener instrucciones o datos de un área de la memoria principal antes de pasar a otra área. Básicamente, el controlador de la caché “adivina” qué área de la lenta memoria dinámica necesitará a continuación la unidad central de



proceso (CPU), y mueve el contenido de dicha área a la memoria caché, para que esté listo cuando sea necesario. Si el controlador de caché ha realizado una estimación correcta, los datos están disponibles de manera inmediata para el microprocesador. Si la estimación del controlador de caché es errónea, la CPU debe acudir a la memoria principal y esperar mucho más tiempo para obtener las instrucciones o datos correctos. Afortunadamente, el controlador de caché tiene razón la mayor parte de las veces.

**Analogía de la caché.** Hay muchas analogías que pueden usarse para describir una memoria caché, pero tal vez lo más efectivo sea compararla con una nevera doméstica. Una nevera doméstica puede considerarse como una especie de “caché” para determinados productos alimenticios, mientras que el supermercado es la memoria principal donde se almacena toda la comida. Cada vez que deseamos comer o beber algo, podemos ir primero a la nevera (caché), para ver si contiene el producto que buscamos. Si es así, nos ahorramos un montón de tiempo. Si el producto no se encuentra allí, tendremos que invertir un tiempo adicional en obtenerlo del supermercado (memoria principal).

**Cachés L1 y L2.** Las cachés de nivel 1 (caché L1) están usualmente integradas en el chip del procesador y tienen una capacidad de almacenamiento muy limitada. La caché L1 se conoce también con el nombre de caché *primaria*. Una caché de nivel 2 (caché L2) es un chip o conjunto de chips de memoria independiente, externo al procesador, y usualmente dispone de una capacidad de almacenamiento mayor que una caché L1. La caché L2 también se conoce con el nombre de caché *secundaria*. Algunos sistemas pueden tener cachés de nivel superior (L3, L4, etc.), pero L1 y L2 son los más comunes. Asimismo, algunos sistemas emplean una caché de disco, ubicada en la memoria DRAM principal y utilizada para mejorar el rendimiento del disco duro, porque la DRAM, aunque mucho más lenta que la memoria SRAM, sigue siendo mucho más rápida que la unidad de disco duro. La Figura 10.15 ilustra dos memorias caché L1 y L2 en un sistema informático.

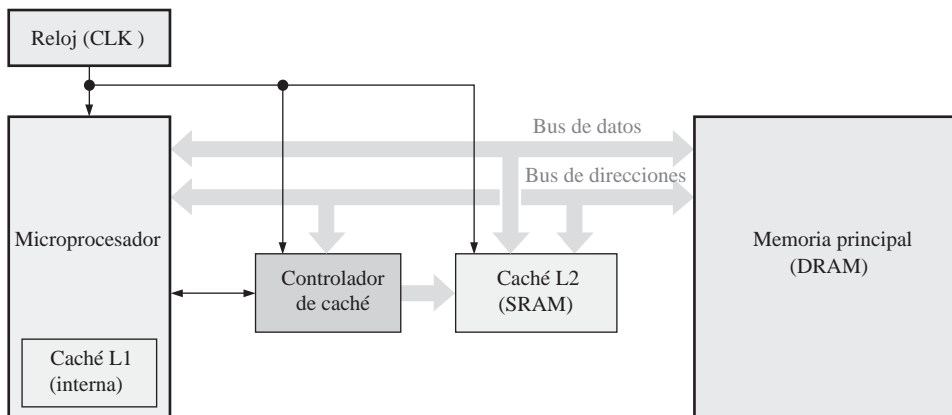


FIGURA 10.15 Diagrama de bloques mostrando memorias caché L1 y L2 en un sistema informático.

## Las celdas de almacenamiento de la RAM dinámica (DRAM)

Las celdas de las **memorias dinámicas** almacenan un bit de datos en un condensador en lugar de en un *latch*. La ventaja de este tipo de celda es que es muy sencilla, lo que permite construir matrices de memoria muy grandes en un chip, a un coste por bit más bajo que el de las memorias estáticas. La desventaja es que el condensador de almacenamiento no puede mantenerse cargado más que un período de tiempo, y el dato almacenado se pierde a no ser que su carga se refresque periódicamente. La operación de refresco requiere circuitería de memoria adicional y complica el funcionamiento de la **DRAM**. La Figura 10.16 presenta una celda típica de una DRAM, formada por un único transistor MOS (MOSFET) y un condensador.

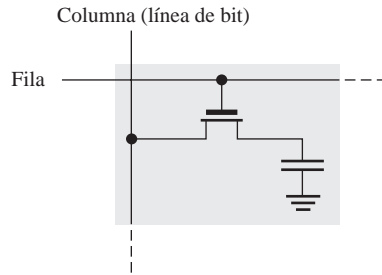


FIGURA 10.16 Celda de una RAM dinámica MOS.

En este tipo de celda, el transistor actúa como un interruptor. El funcionamiento básico simplificado se ilustra en la Figura 10.17, y es el siguiente. Un nivel BAJO en la línea  $R/\bar{W}$  (modo escritura) activa el buffer de tres estados de entrada y desactiva el *buffer* de salida. Para escribir un 1 en la celda, la línea  $D_{IN}$  debe estar a nivel ALTO, y el transistor debe ser puesto en conducción poniendo un nivel ALTO en la línea de fila. El transistor opera como un interruptor cerrado, que conecta el condensador a la línea de bit. Esta conexión permite al condensador cargarse con una tensión positiva, como muestra la Figura 10.17(a). Cuando se almacena un 0, se aplica un nivel BAJO a la línea  $D_{IN}$ . Si el condensador almacenaba un 0, permanece descargado; ahora bien, si almacenaba un 1, se descarga como se indica en la Figura 10.17(b). Cuando la línea de fila vuelve al nivel BAJO, el transistor no conduce y desconecta el condensador de la línea de bit, con lo que la carga (1 o 0) “queda atrapada” en el condensador.

Para leer una celda, la línea  $R/\bar{W}$  (*Read/Write*, lectura/escritura) se pone a nivel ALTO, lo que activa el buffer de salida y desactiva el *buffer* de entrada. Cuando la línea de fila se pone a nivel ALTO, el transistor conduce y conecta el condensador a la línea de bit y, por tanto, al buffer de salida (amplificador). De esta manera, el bit de datos aparece en la línea de salida de datos ( $D_{OUT}$ ). Este proceso se ilustra en la Figura 10.17(c).

Para refrescar la celda de memoria, la línea  $R/\bar{W}$ , la línea de fila y la línea de refresco se ponen a nivel ALTO. El transistor conduce, conectando el condensador a la línea de bit. El *buffer* de salida se activa y el bit de datos almacenado se aplica a la entrada del *buffer* de refresco, el cual se activa mediante el nivel ALTO de la entrada de refresco. Esto da lugar a una tensión en la línea de bit que corresponde al bit almacenado, recargando el condensador como se ilustra en la Figura 10.17(d).

## Organización básica de una DRAM

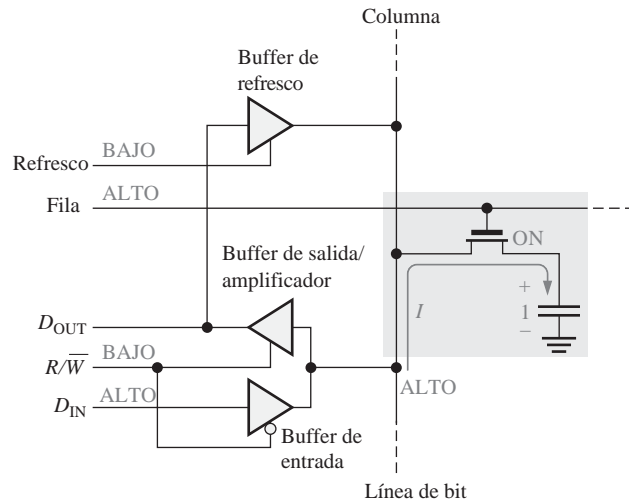
La principal aplicación de las DRAM se encuentra en la memoria principal de las computadoras. La diferencia principal entre las DRAM y las SRAM es el tipo de celda de memoria. Como se ha visto, la celda de la memoria DRAM está formada por un transistor y un condensador, y es mucho más sencilla que la celda de la SRAM. Esto permite densidades mucho mayores en las DRAM, lo que da lugar a mayores capacidades de bits para una determinada área de chip, aunque el tiempo de acceso es mucho mayor.

De nuevo, dado que la carga almacenada en un condensador tiende a perderse, las celdas de una DRAM requieren una operación de refresco frecuente para conservar los bits de datos almacenados. Este requisito da lugar a una circuitería más compleja que en la SRAM. A continuación se exponen varias funciones comunes en la mayor parte de las DRAM, utilizando como ejemplo una DRAM genérica de  $1\text{ M} \times 1$  bit.

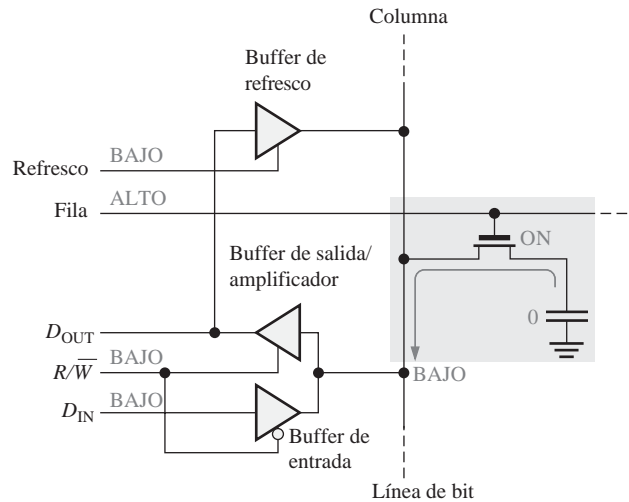
**Multiplexación de direcciones.** Las DRAM utilizan una técnica denominada *multiplexación de direcciones* que sirve para reducir el número de líneas de dirección. La Figura 10.18 muestra el diagrama de bloques de una DRAM de 1.048.576 bits (1 Mbit) con una organización de  $1\text{ M} \times 1$ . Vamos a centrarnos en los bloques de color gris oscuro para ilustrar la multiplexación de direcciones. Los bloques en color gris claro representan la lógica de refresco.

Las diez líneas de dirección se multiplexan en el tiempo al comienzo de un ciclo de memoria mediante la validación de dirección de fila ( $\overline{RAS}$ ) y la validación de dirección de columna ( $\overline{CAS}$ ), en dos campos de dirección separados. En primer lugar, la dirección de fila de 10 bits se pasa al *latch* de direcciones de fila. Después, la dirección de columna de 10 bits se pasa al *latch* de direcciones de columna. Las direcciones de fila y columna se decodifican para seleccionar una de las 1.048.576 ( $2^{20} = 1.048.576$ ) direcciones de la matriz de memoria. En la Figura 10.19 se presenta el diagrama de tiempos básico para la operación de multiplexación de direcciones.

**Ciclos de lectura y escritura.** Al inicio de cada ciclo de memoria de lectura o escritura,  $\overline{RAS}$  y  $\overline{CAS}$  se activan (nivel BAJO) para multiplexar las direcciones de fila y columna hacia los *latches* y decodificadores. Durante

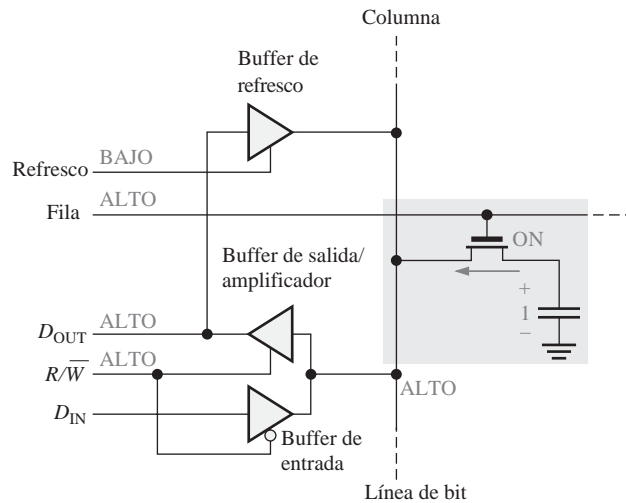


(a) Escritura de un 1 en la celda de memoria

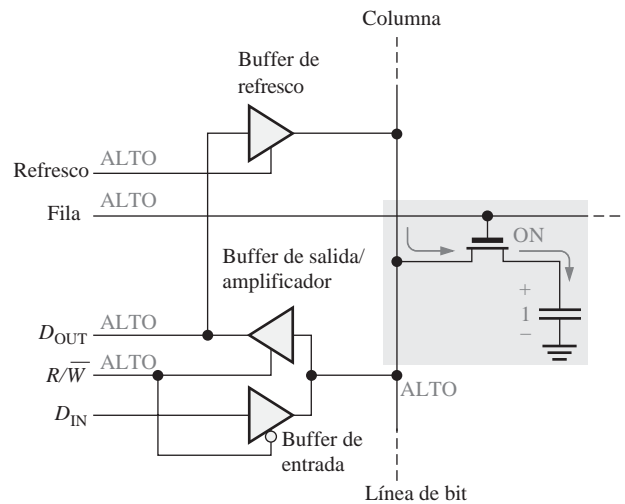


(b) Escritura de un 0 en la celda de memoria

FIGURA 10.17 Funcionamiento básico de una celda de una DRAM. (Continúa)



(c) Lectura de un 1 desde la celda de memoria



(d) Refresco de un 1 almacenado

FIGURA 10.17 Funcionamiento básico de una celda de una DRAM. (Continuación)

el ciclo de lectura, la entrada  $R/\bar{W}$  está a nivel ALTO. Durante el ciclo de escritura, la entrada  $R/\bar{W}$  está a nivel BAJO. Esto se ilustra en la Figura 10.20.

**Modo página rápido.** En los ciclos de lectura y escritura normales descritos anteriormente, primero se carga la dirección de fila de una posición de memoria concreta mediante la entrada activa a nivel BAJO  $\bar{RAS}$ , y luego se carga la dirección de columna de esa posición mediante la entrada activa a nivel BAJO  $\bar{CAS}$ . Después se selecciona la siguiente posición mediante otra entrada  $\bar{RAS}$  seguida de  $\bar{CAS}$ , y así sucesivamente.

Una “página” es una sección de memoria disponible en una misma dirección de fila y que consta de todas las columnas de dicha fila. El modo página rápido permite operaciones de lectura y escritura sucesivas en cada una de las direcciones de columna de una fila seleccionada. En primer lugar, se carga una dirección de fila

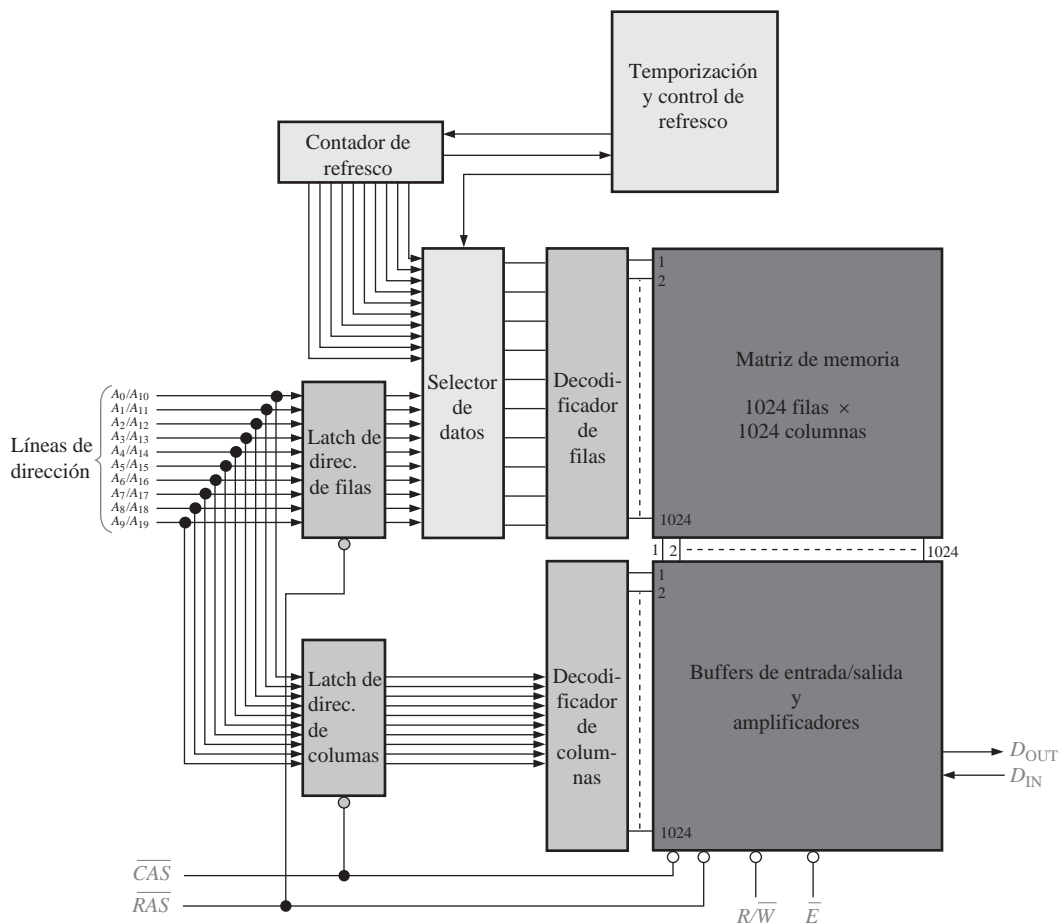


FIGURA 10.18 Diagrama de bloques simplificado de una DRAM de  $1M \times 1$ .

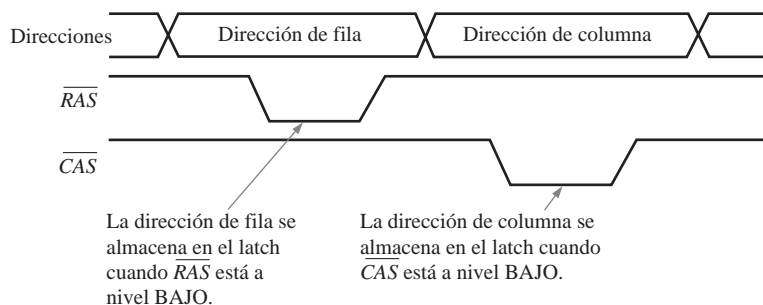
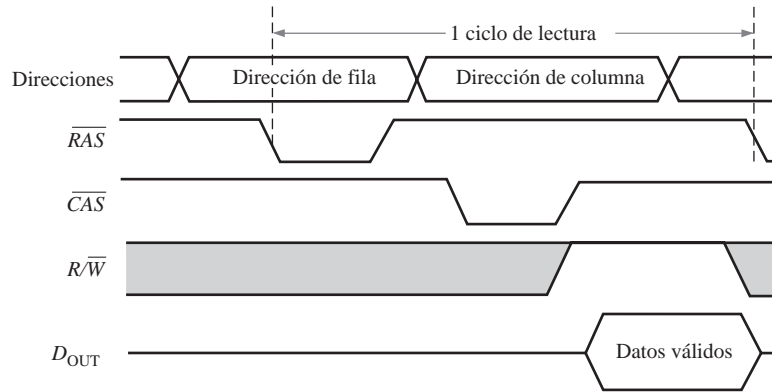
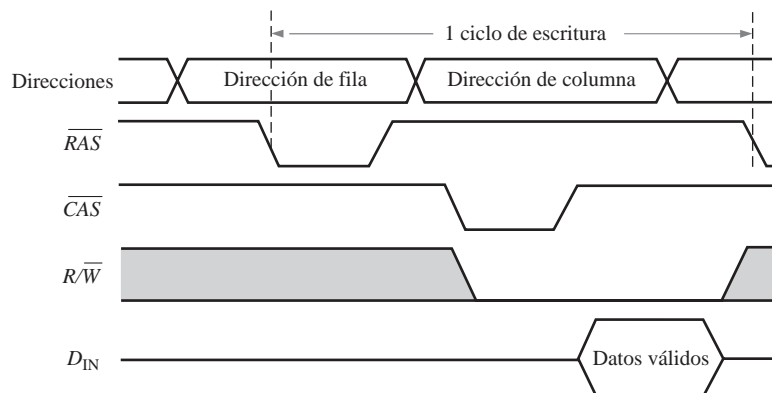


FIGURA 10.19 Diagrama de tiempos básico para la multiplexación de direcciones.

pasando la entrada  $\overline{RAS}$  a nivel BAJO y permaneciendo en este nivel, mientras que  $\overline{CAS}$  bascula entre los niveles ALTO y BAJO. Sólo se selecciona una dirección de fila, que permanece seleccionada mientras que  $\overline{RAS}$  esté activa. Cada  $\overline{CAS}$  sucesiva selecciona una columna de la fila especificada. De este modo, después



(a) Ciclo de lectura



(b) Ciclo de escritura

**FIGURA 10.20** Diagrama de tiempos para los ciclos de lectura y escritura normales.

de un ciclo en modo página rápida, todas las direcciones de la fila seleccionada se habrán leído o escrito, dependiendo del nivel de  $R/\overline{W}$ . Por ejemplo, un ciclo en modo página rápida para la DRAM de la Figura 10.18 requiere que la señal  $\overline{CAS}$  se active 1024 veces para cada fila seleccionada mediante la señal  $\overline{RAS}$ .

En la Figura 10.21 se presenta un diagrama de tiempos que ilustra el funcionamiento básico en modo página rápida para la operación de lectura. Cuando  $\overline{CAS}$  pasa a su estado no activo (ALTO), desactiva las salidas de datos. Por tanto, la transición de la señal  $\overline{CAS}$  al nivel ALTO sólo debe producirse después de que el sistema externo almacene los datos válidos en un *latch*.

**Ciclos de refresco.** Como ya sabe, las DRAM se basan en el almacenamiento de carga en un condensador para cada bit de memoria de la matriz. Esta carga se degrada (se pierde) con el tiempo y la temperatura, por lo que cada bit se debe refrescar (recargar) periódicamente para mantener el estado correcto del bit. Típicamente, una DRAM se debe refrescar cada 8 ms o 16 ms, aunque en algunos dispositivos el período de refresco puede exceder 100 ms.

Una operación de lectura refresca automáticamente todas las direcciones de la fila seleccionada. Sin embargo, en aplicaciones típicas, no siempre se puede predecir cuán a menudo se producirá un ciclo de lectura y, por tanto, no se puede depender de que un ciclo de lectura se efectúe frecuentemente para evitar la pérdida de datos. En consecuencia, en los sistemas DRAM se deben implementar ciclos de refresco especiales.

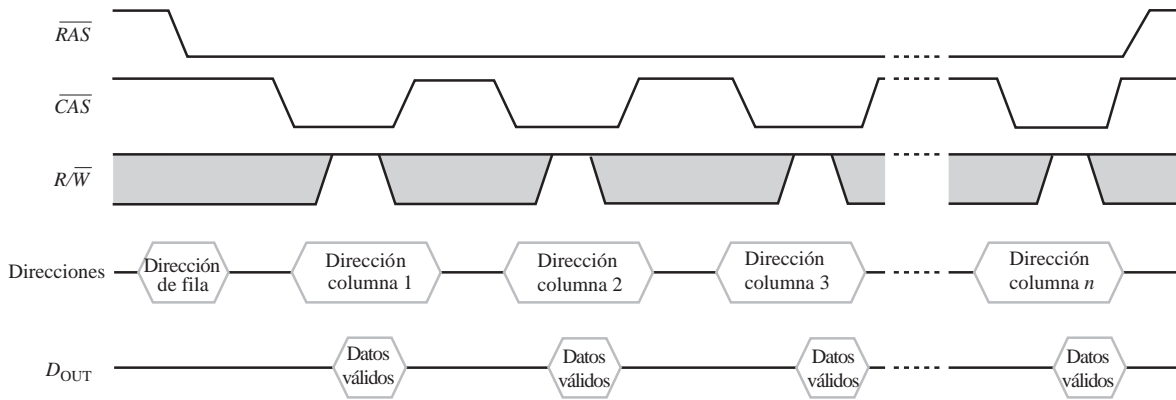


FIGURA 10.21 Cronograma del modo página rápida para la operación de lectura.

El *refresco a ráfagas* y el *refresco distribuido* son los dos modos básicos de las operaciones de refresco. En el refresco a ráfagas, todas las filas de la matriz de memoria se recargan consecutivamente en cada período de refresco. Para una memoria con un período de refresco de 8 ms, se produce una vez un refresco a ráfagas de todas las filas cada 8 ms. Las operaciones de lectura y escritura normales se suspenden durante un ciclo de refresco de ráfaga. En el refresco distribuido, cada fila se refresca en intervalos entremezclados con los ciclos de lectura y escritura normales. Por ejemplo, en la memoria de la Figura 10.18, que tiene 1024 filas, un periodo de refresco de 8 ms exige que se recargue una fila cada  $8\text{ms}/1024 = 7,8 \mu\text{s}$ , cuando se utiliza el refresco distribuido.

Los dos tipos de operaciones de refresco son: *refresco sólo con RAS* (*RAS-only refresh*) y *refresco CAS antes de RAS* (*CAS before RAS refresh*). El refresco sólo *RAS* consiste en una transición de la señal *RAS* a nivel BAJO (estado activo), que almacena la dirección de la fila en el *latch* para realizar el refresco, mientras que la línea *CAS* permanece a nivel ALTO (estado inactivo) a lo largo del ciclo. Se utiliza un contador externo para proporcionar las direcciones de fila en este tipo de operación.

El refresco *CAS* antes de *RAS* se inicia cuando la línea *CAS* pasa a nivel BAJO y a continuación la línea *RAS* pasa a nivel BAJO. Esta secuencia activa un contador de refresco interno que genera la dirección de fila para la que se debe realizar la recarga. Esta dirección se conmuta mediante el selector de datos hacia el decodificador de filas.

## Tipos de memoria DRAM

Ahora que hemos aprendido los conceptos básicos de una memoria DRAM, echemos un breve vistazo a los tipos principales existentes. Dichos tipos son la *DRAM con modo página rápida* (*Fast Page Mode, FPM DRAM*), la *DRAM con salida de datos extendida* (*Extended Data Output, EDO DRAM*), la *DRAM con salida de datos extendida en ráfaga* (*Burst Extended Data Output, BEDO DRAM*) y la *DRAM síncrona* (*Synchronous DRAM, SDRAM*).

**FPM DRAM.** Ya hemos descrito anteriormente la operación en modo de página rápida. Este tipo de DRAM ha sido tradicionalmente el más común y es el que se ha usado en las computadoras hasta el desarrollo de la EDO DRAM. Recuerde que una página de la memoria está formada por todas las direcciones de columna contenidas en una misma dirección de fila.

Como ya hemos visto, la idea básica de la **FPM DRAM** se basa en la probabilidad de que las siguientes direcciones de memoria a las que haya que acceder se encuentren en la misma fila (en la misma página). Afortunadamente, esto sucede en un gran porcentaje de las veces. El modo FPM ahorra tiempo, con respecto

al acceso puramente aleatorio, porque en el modo FPM la dirección de fila se especifica una única vez para acceder a varias direcciones de columna sucesivas, mientras que en el acceso aleatorio puro, hay que especificar una dirección de fila para cada dirección de columna.

Recuerde que en una operación de lectura en modo de página rápido, la señal  $\overline{CAS}$  tiene que esperar hasta que los datos válidos correspondientes a una dirección dada sean aceptados (enclavados) por el sistema externo (UCP), antes de volver a su estado inactivo. Cuando  $\overline{CAS}$  pasa a su estado inactivo, se desactivan las salidas de datos. Esto significa que la siguiente dirección de columna no puede ser generada hasta que los datos correspondientes a la dirección de columna actual sean transferidos a la UCP. Esto limita la velocidad de acceso a las columnas situadas dentro de una página.

**EDO DRAM.** La memoria DRAM con salida de datos extendida, algunas veces denominada *DRAM con modo hiperpágina*, es muy similar a la FPM DRAM. La diferencia fundamental es que la señal  $\overline{CAS}$  en la **EDO DRAM**, no desactiva los datos de salida cuando pasa a su estado de inactividad, porque se pueden mantener los datos válidos correspondientes a la dirección actual hasta que  $\overline{CAS}$  vuelva a activarse. Esto significa que se puede acceder a la siguiente dirección de columna antes de que el sistema externo acepte los datos válidos actuales. La idea es acelerar el tiempo de acceso.

**BEDO DRAM.** La DRAM con salida de datos extendida en ráfaga es una EDO DRAM con la capacidad de generar ráfagas de direcciones. Recuerde, de nuestra explicación sobre la SRAM síncrona de ráfaga, que la función de ráfaga de direcciones permite generar internamente hasta cuatro direcciones a partir de una única dirección externa, lo que ahorra tiempo de acceso. Este mismo concepto se aplica a la **BEDO DRAM**.

**SDRAM.** Para poder estar a la altura de la siempre creciente velocidad de los microprocesadores, son necesarias memorias DRAM más rápidas. La DRAM síncrona es uno de los esfuerzos más recientes en este sentido. Al igual que la RAM estática síncrona explicada anteriormente, la operación de la memoria **SDRAM** está sincronizada con el reloj del sistema, con el que también opera el microprocesador de un sistema informático. Las mismas ideas básicas descritas en relación con la SRAM síncrona de ráfaga se pueden aplicar a la memoria SDRAM.

Esta operación de tipo síncrono hace que la memoria SDRAM sea totalmente diferente de los otros tipos de DRAM asíncrona previamente mencionados. Con las memorias asíncronas, el microprocesador se ve obligado a esperar a que la DRAM complete sus operaciones internas. Sin embargo, con la operación de tipo síncrono, la DRAM enclava las direcciones, los datos y la información de control generados por el procesador, bajo control del reloj del sistema. Esto permite al procesador gestionar otras tareas mientras se están realizando las operaciones de lectura o escritura en memoria, en lugar de tener que esperar a que la memoria realice su tarea, como es el caso en los sistemas asíncronos.

### REVISIÓN DE LA SECCIÓN 10.2

1. Enumerar dos tipos de SRAM.
2. ¿Qué es una memoria caché?
3. Explicar en qué se diferencian las SRAM y las DRAM.
4. Describir la operación de refresco de una DRAM.
5. Enumerar cuatro tipos de DRAM.

## 10.3 MEMORIAS DE SÓLO LECTURA (ROM)

---

Una ROM mantiene de forma permanente o semipermanente los datos almacenados, que pueden ser leídos de la memoria pero, o no se pueden cambiar en absoluto, o se requiere un equipo especial para ello. Una ROM almacena datos que se utilizan repetidamente en las aplicaciones, tales como tablas,



conversiones o instrucciones programadas para la inicialización y el funcionamiento de un sistema. Las ROM mantienen los datos almacenados cuando se desconecta la alimentación y son, por tanto, memorias no volátiles.

Al finalizar esta sección, el lector deberá ser capaz de:

- Enumerar los tipos de ROM. ■ Describir una celda básica de almacenamiento ROM de máscara.
- Explicar cómo se leen los datos de una ROM. ■ Estudiar la organización interna de una ROM típica. ■ Estudiar algunas aplicaciones de las ROM.

## La familia de las memorias ROM

La Figura 10.22 muestra cómo se clasifican las memorias ROM semiconductoras. La ROM de máscara es un tipo de memoria en la que los datos se almacenan permanentemente en la memoria durante el proceso de fabricación. La **PROM**, o ROM programable, es aquel tipo de ROM en la que el usuario, con ayuda de equipos especializados, almacena eléctricamente los datos. Tanto la ROM de máscara como la PROM pueden ser de cualquier tecnología MOS o bipolar. La **EPROM**, o memoria PROM borrable (*erasable PROM*) es exclusivamente un dispositivo MOS. La **UV EPROM** puede ser programada eléctricamente por el usuario, pero los datos almacenados deben borrarse mediante la exposición a la luz ultravioleta durante un período de varios minutos. La PROM borrable eléctricamente (**EEPROM** o  $E^2$ PROM, *Electrically Erasable PROM*) se puede borrar en unos pocos milisegundos.

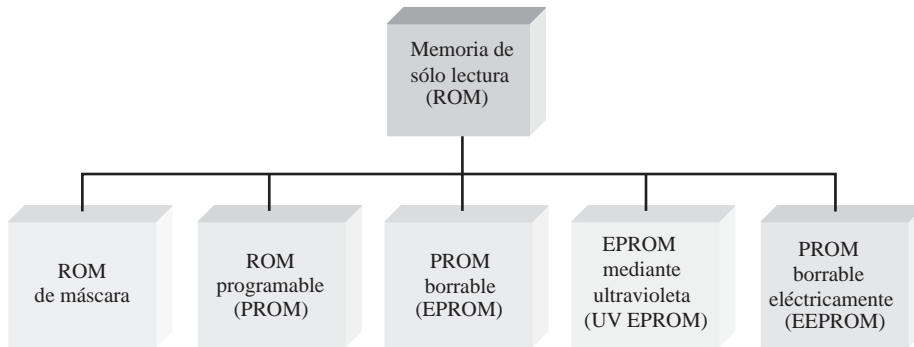


FIGURA 10.22 La familia de memorias ROM.

## La ROM de máscara

Normalmente, la ROM de máscara se denomina simplemente ROM. Es una memoria programada de forma permanente durante el proceso de fabricación, para proporcionar funciones estándar de uso extendido, tales como conversiones populares, o para proporcionar funciones especificadas por el usuario. Una vez que se programa la memoria, ésta no puede cambiarse. La mayoría de los circuitos integrados ROM utilizan la presencia o ausencia de una conexión de transistor en una unión fila/columna para representar un 1 o un 0.

La Figura 10.23 muestra celdas MOS de una ROM. La presencia de una conexión desde una línea de fila a la puerta de un transistor representa un 1 en esa posición, ya que, cuando la línea de fila está a nivel ALTO, todos los transistores con conexión de puerta a esa línea de fila conducen, y ponen a nivel ALTO (1) a las líneas de columna asociadas. En las uniones de fila/columna en las que no existe conexión de puerta, las líneas de columna permanecen a nivel BAJO (0) cuando se direcciona la fila.

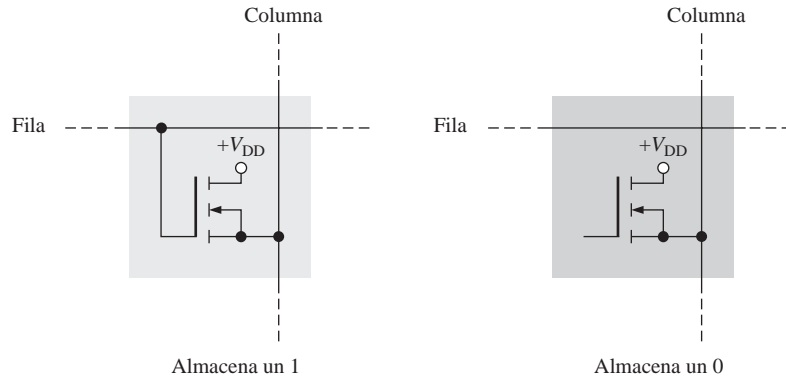


FIGURA 10.23 Celdas ROM.

### Una ROM básica

Para ilustrar el concepto de ROM, la Figura 10.24 muestra una matriz ROM simplificada. Los cuadrados en color gris claro representan los 1s almacenados y los cuadrados en gris oscuro representan los 0s almacenados. El procedimiento de lectura básico es el siguiente: cuando se aplica un código de dirección binario a las líneas de entrada de dirección, la línea de la fila correspondiente se pone a nivel ALTO. Este nivel ALTO se conecta a las líneas de las columnas a través de los transistores en cada unión (celda) donde se almacena un 1. En cada celda en la que se almacena un 0, la línea de columna permanece a nivel BAJO, debido a la

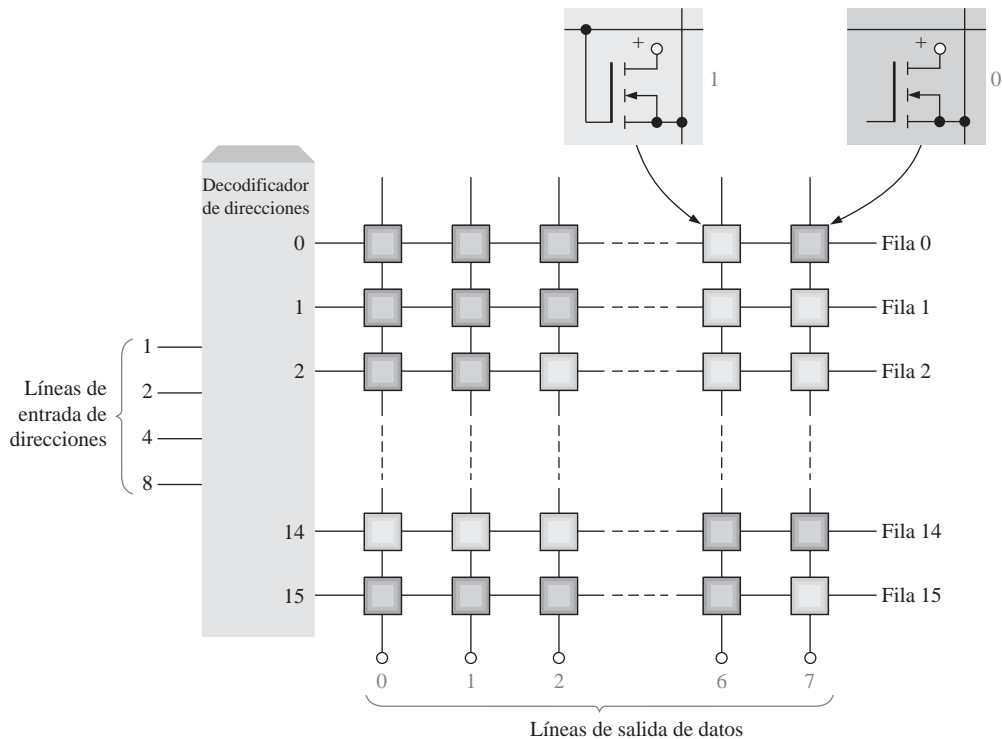


FIGURA 10.24 Representación de una matriz ROM de 16 × 8 bits.

resistencia de terminación. Las líneas de las columnas constituyen la salida de datos. Los ocho bits de datos almacenados en la fila seleccionada se presentan en las líneas de salida.

Como puede ver, la ROM de ejemplo de la Figura 10.24 está organizada en 16 direcciones, cada una de las cuales almacena 8 bits de datos. Por tanto, se trata de una ROM de  $16 \times 8$  (16 por 8) y su capacidad total es de 128 bits, o 16 bytes. Las ROM puede utilizar tablas de búsqueda (LUT, *Look-Up Table*) para realizar conversiones de códigos y generación de función lógicas.

### EJEMPLO 10.1

Dibujar una ROM similar a la de la Figura 10.24, programada para convertir a código Gray números binarios de 4 bits.

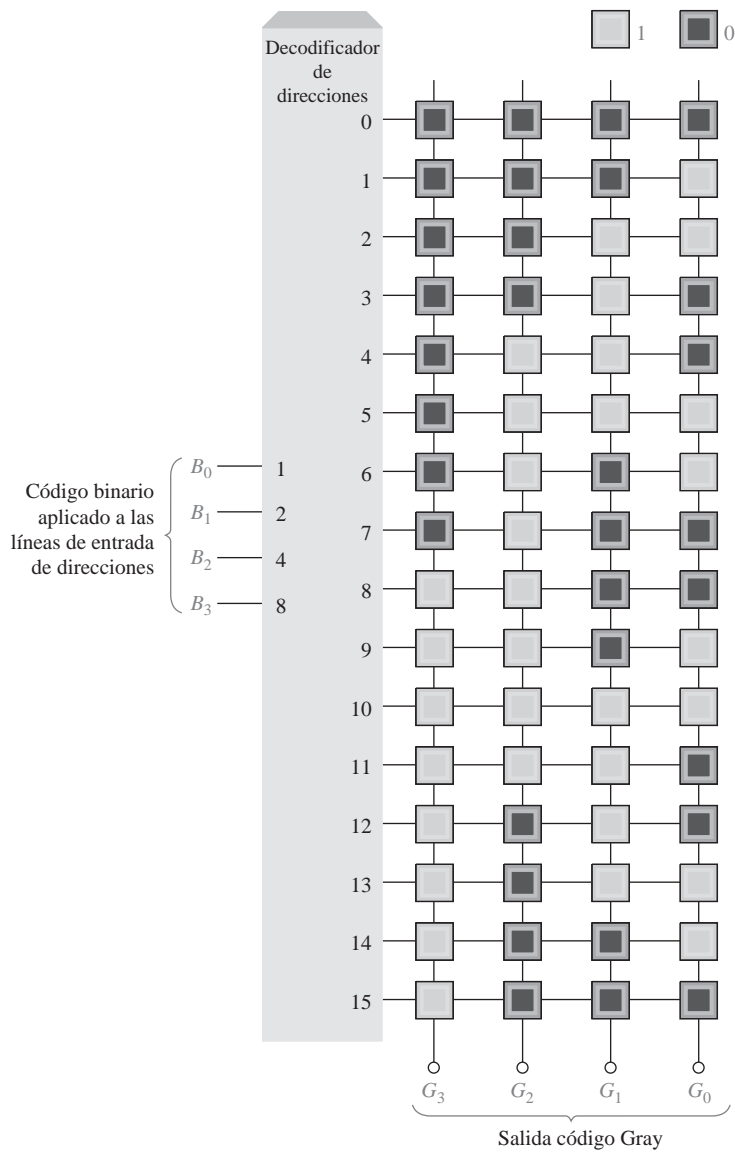
#### Solución

Repase, en el Capítulo 2, el código Gray. Se ha desarrollado la Tabla 10.1 con el fin de utilizarla para programar la ROM.

| Binario |       |       |       | Gray  |       |       |       |
|---------|-------|-------|-------|-------|-------|-------|-------|
| $B_3$   | $B_2$ | $B_1$ | $B_0$ | $G_3$ | $G_2$ | $G_1$ | $G_0$ |
| 0       | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| 0       | 0     | 0     | 1     | 0     | 0     | 0     | 1     |
| 0       | 0     | 1     | 0     | 0     | 0     | 1     | 1     |
| 0       | 0     | 1     | 1     | 0     | 0     | 1     | 0     |
| 0       | 1     | 0     | 0     | 0     | 1     | 1     | 0     |
| 0       | 1     | 0     | 1     | 0     | 1     | 1     | 1     |
| 0       | 1     | 1     | 0     | 0     | 1     | 0     | 1     |
| 0       | 1     | 1     | 1     | 0     | 1     | 0     | 0     |
| 1       | 0     | 0     | 0     | 1     | 1     | 0     | 0     |
| 1       | 0     | 0     | 1     | 1     | 1     | 0     | 1     |
| 1       | 0     | 1     | 0     | 1     | 1     | 1     | 1     |
| 1       | 0     | 1     | 1     | 1     | 1     | 1     | 0     |
| 1       | 1     | 0     | 0     | 1     | 0     | 1     | 0     |
| 1       | 1     | 0     | 1     | 1     | 0     | 1     | 1     |
| 1       | 1     | 1     | 0     | 1     | 0     | 0     | 1     |
| 1       | 1     | 1     | 1     | 1     | 0     | 0     | 0     |

TABLA 10.1

La matriz ROM resultante de  $16 \times 4$  se muestra en la Figura 10.25. Se puede ver que un código binario en las líneas de dirección de entrada produce el correspondiente código Gray en las líneas de salida (columnas). Por ejemplo, cuando se aplica el número binario 0110 a las líneas de dirección de entrada, se selecciona la dirección 6, que almacena el código Gray 0101.



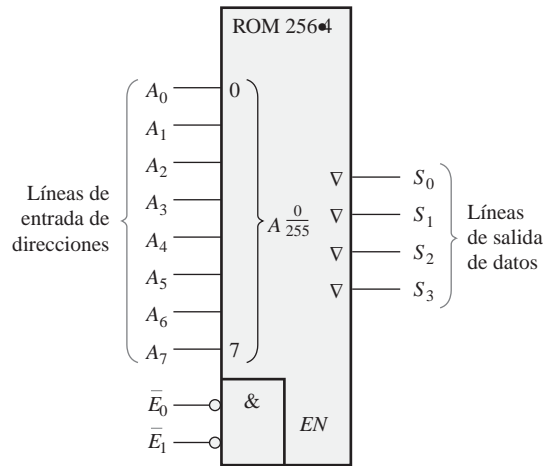
**FIGURA 10.25** Representación de una ROM programada como convertidor de código binario - Gray.

**Problema relacionado\*** Utilizando la Figura 10.25, determinar el código Gray de salida cuando se aplica el código binario 1011 a las líneas de dirección de entrada.

\*Las respuestas se encuentran al final del capítulo.

## Organización interna de la ROM

La mayoría de los circuitos integrados ROM tienen una organización interna algo más compleja que la de la ROM básica del ejemplo que acabamos de presentar. Para ilustrar cómo se estructura un CI ROM se utiliza



**FIGURA 10.26** Símbolo lógico de una ROM de  $256 \times 4$ . El identificativo  $A_{\frac{0}{255}}$  significa que el código de dirección de 8 bits selecciona direcciones de 0 a 255.

un dispositivo de 1024 bits organizado en una matriz de  $256 \times 4$ . El símbolo lógico se muestra en la Figura 10.26. Cuando se aplica cualquiera de los 256 códigos binarios (ocho bits) a las líneas de dirección, se presentan cuatro bits de datos en las salidas si las entradas de habilitación del chip se encuentran a nivel BAJO (256 direcciones requieren ocho líneas de dirección).

Aunque la estructura de  $256 \times 4$  de este dispositivo parece indicar que tiene 256 filas y 4 columnas en la matriz de memoria, éste no es el caso en realidad. Realmente la matriz de memoria es una matriz de  $32 \times 32$  (32 filas y 32 columnas), como muestra el diagrama de bloques de la Figura 10.27

La ROM de la Figura 10.27 funciona de la manera siguiente: cinco de las ocho líneas de dirección ( $A_0$  hasta  $A_4$ ) se decodifican mediante el decodificador de filas (comúnmente denominado decodificador  $Y$ ) para seleccionar una de las 32 filas. Tres de las ocho líneas de dirección ( $A_5$  hasta  $A_7$ ) se decodifican mediante el decodificador de columnas (denominado comúnmente decodificador  $X$ ) para seleccionar cuatro de las 32 columnas. En realidad, el decodificador de columnas está formado por cuatro decodificadores 1-de-8 (seletores de datos), como se muestra en la Figura 10.27.

El resultado de esta estructura es que, al aplicar un código de dirección de 8 bits ( $A_0$  hasta  $A_7$ ), aparece una palabra de datos de 4 bits en las salidas de datos cuando las líneas de habilitación del chip ( $\bar{E}_0$  y  $\bar{E}_1$ ) deben estar a nivel BAJO para activar los buffers de salida. Este tipo de organización interna (arquitectura) es típica de diversos circuitos integrados ROM, de distintas capacidades.



### NOTAS INFORMÁTICAS

La memoria ROM se usa en las computadoras personales para almacenar lo que se denomina BIOS (*Basic Input/Output Services*, servicios básicos de entrada/salida). Se trata de programas que se emplean para llevar a cabo funciones fundamentales de soporte y supervisión en la computadora. Por ejemplo, los programas de BIOS almacenados en la ROM controlan determinadas funciones de vídeo, proporcionan la función de formateo de discos, exploran el teclado para detectar las pulsaciones y controlan ciertas funciones de impresión.

## Tiempo de acceso de la ROM

En la Figura 10.28 se presenta un diagrama de tiempos típico que ilustra el tiempo de acceso a la ROM. El **tiempo de acceso** de una ROM,  $t_a$ , es el tiempo que transcurre desde que se aplica un código de dirección válido

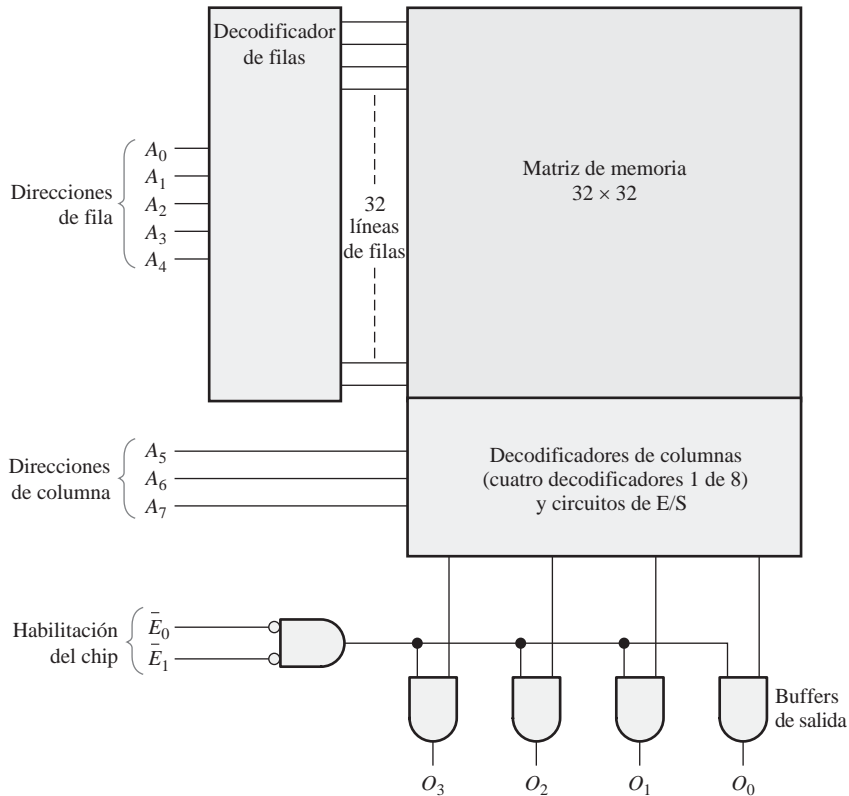


FIGURA 10.27 ROM de 1024 bits con una organización de  $256 \times 4$  basada en una matriz de  $32 \times 32$ .

do en las líneas de entrada hasta que aparecen los datos válidos en las líneas de salida. El tiempo de acceso se puede también medir desde que se activa la entrada de habilitación del chip ( $\bar{E}$ ) hasta que aparecen los datos válidos en la salida, cuando ya se encuentra una dirección válida en las líneas de entrada.

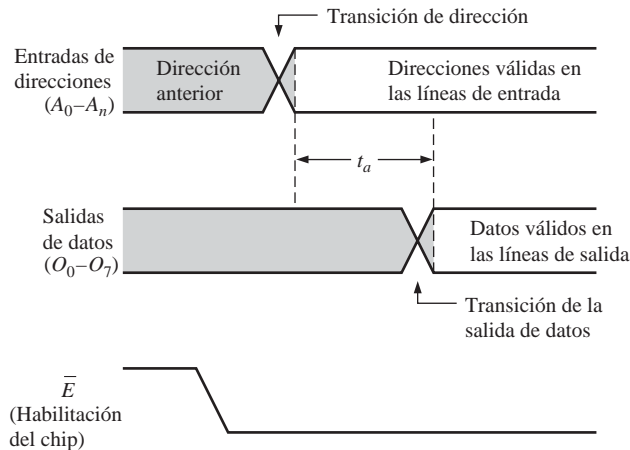


FIGURA 10.28 Tiempo de acceso de una ROM ( $t_a$ ) desde el cambio de dirección hasta la salida de datos, con la entrada de habilitación del chip activa.

## REVISIÓN DE LA SECCIÓN 10.3

1. ¿Cuál es la capacidad de almacenamiento en bits de una ROM con una estructura de  $512 \times 8$ ?
2. Enumerar los tipos de las memorias de sólo lectura.
3. ¿Cuántos bits de dirección son necesarios en una memoria de 2048 bits organizada como una memoria de  $256 \times 8$ ?

## 10.4 MEMORIAS ROM PROGRAMABLES (PROM Y EPROM)

Las PROM son básicamente iguales que las ROM de máscara, una vez que han sido programadas. Como ya hemos visto, las ROM son un tipo de dispositivo lógico programable. La diferencia consiste en que las PROM salen de fábrica sin estar programadas y se programan a medida para satisfacer las necesidades del usuario.

Al finalizar esta sección, el lector deberá ser capaz de:

- Distinguir entre una ROM de máscara y una PROM.
- Describir una celda básica de almacenamiento de una PROM.
- Utilizar las EPROM, incluyendo las UV EPROM y EEPROM.
- Analizar un ciclo de programación de una EPROM.

### Memorias PROM

Las **PROM** utilizan algún tipo de mecanismo de fundición para almacenar bits, donde un *hilo* de memoria se funde o queda intacto para representar un 0 o un 1. El proceso de fundición es irreversible; una vez que una PROM ha sido programada no puede cambiarse.

La Figura 10.29 muestra una matriz MOS de una PROM con hilos fusibles. Los fusibles se introducen en la PROM (durante el proceso de fabricación) entre la fuente del transistor de cada celda y su línea de columna. Durante el proceso de programación, se introduce una corriente adecuada a través del hilo fusible para fundirlo y que permanezca abierto, almacenando de esta manera un 0. El fusible se deja intacto para almacenar un 1.

Los tres tipos básicos de tecnologías de fusibles utilizados en las PROM son las conexiones de metal, las conexiones de silicio y las uniones *pn*. A continuación, se proporciona una breve descripción de cada una de ellas.

1. Las conexiones de metal se realizan con materiales como el nicromo. Cada bit de la matriz de memoria se representa mediante una conexión separada. Durante la programación, la conexión puede fundirse o quedar intacta. Básicamente, esto se realiza direccionando primero una determinada celda, y luego aplicando una cantidad de corriente suficientemente alta como para hacer que la conexión se abra.
2. Las conexiones de silicio están constituidas por tiras estrechas y alargadas de silicio policristalino. La programación de estos fusibles requiere que las conexiones se fundan por el paso de una cantidad de corriente adecuada a su través. Esta cantidad de corriente hace que aumente la temperatura en el fusible, lo que origina que se oxide el silicio, formando un aislante alrededor de la conexión que ahora está abierta.
3. La tecnología de uniones cortocircuitadas, o de migración inducida por avalancha, consiste básicamente en dos uniones *pn* dispuestas una frente a la otra. Durante el proceso de programación, una de las uniones de los diodos entra en avalancha, y el voltaje y el calor resultantes hacen que los iones de aluminio migren y cortocircuiten la unión. La unión restante se utiliza posteriormente como diodo polarizado en directa para representar un bit de datos.

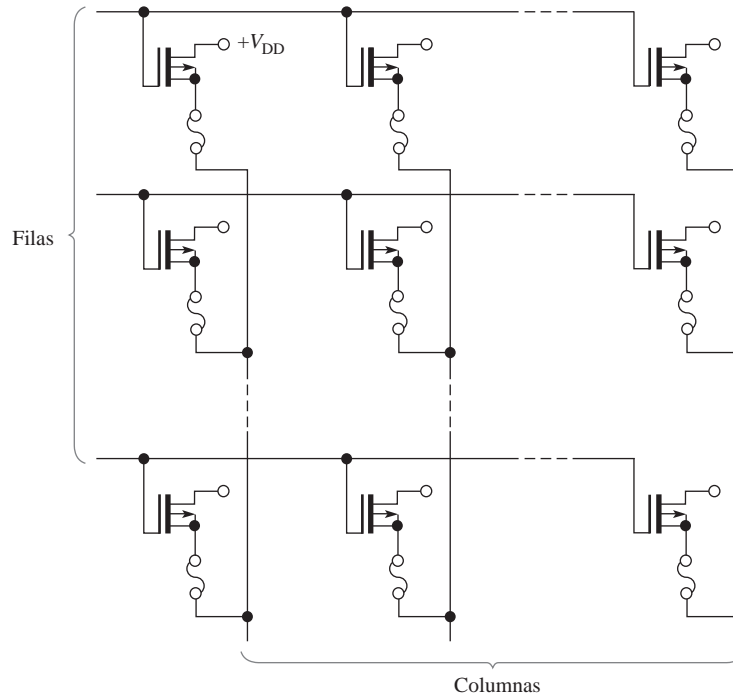


FIGURA 10.29 Matriz PROM MOS con hilos fusibles. Generalmente, todos los drenadores se conectan a  $V_{DD}$ .

## Memorias EPROM

Una **EPROM** es una PROM borrable. A diferencia de una PROM ordinaria, una EPROM puede ser reprogramada si antes se borra el programa existente en la matriz de memoria.

Una EPROM utiliza una matriz NMOSFET con una estructura de puerta aislada. La puerta del transistor aislada no tiene ninguna conexión eléctrica y puede almacenar una carga eléctrica durante un período de tiempo indefinido. Los bits de datos en este tipo de matriz se representan mediante la presencia o ausencia de una carga almacenada en la puerta. El borrado de un bit de datos es un proceso que elimina la carga de la puerta.

Los dos tipos fundamentales de memorias PROM borrables son las PROM borrables por rayos ultravioleta (UV EPROM) y las PROM borrables eléctricamente (EEPROM).

**UV EPROM.** Una UV EPROM se puede reconocer por la ventana de cuarzo transparente de su encapsulado, como se muestra en la Figura 10.30. La puerta aislada del **FET** de una EPROM ultravioleta está “flotando” dentro de un material óxido aislante. El proceso de programación hace que los electrones sean

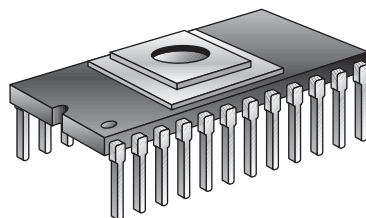


FIGURA 10.30 Encapsulado de una PROM borrable por rayos ultravioleta.



eliminados de la puerta flotante. El borrado se realiza mediante la exposición del chip de la matriz de memoria a una radiación ultravioleta de alta intensidad, a través de la ventana de cuarzo en la parte superior del encapsulado. La carga positiva almacenada en la puerta se neutraliza después de un período de tiempo de entre unos minutos y una hora de exposición.

En la Figura 10.31 se representa una UV EPROM típica mediante un diagrama lógico. Su funcionamiento es representativo de otras memorias UV EPROM típicas, de distintos tamaños. Como muestra el símbolo lógico de la Figura 10.31, este dispositivo tiene 2048 direcciones ( $2^{11} = 2048$ ), cada una con ocho bits. Observe que las ocho salidas son tri-estado ( $\nabla$ ).

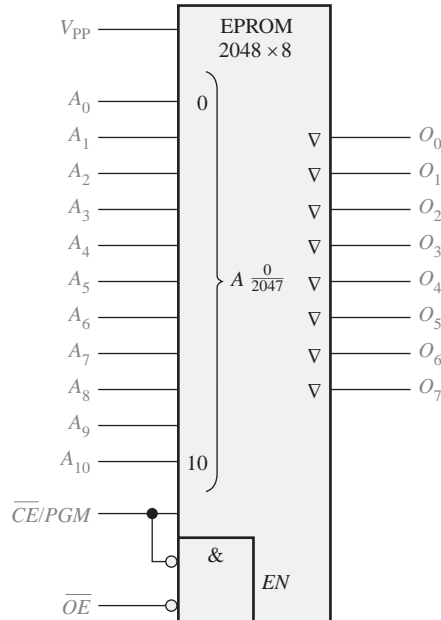


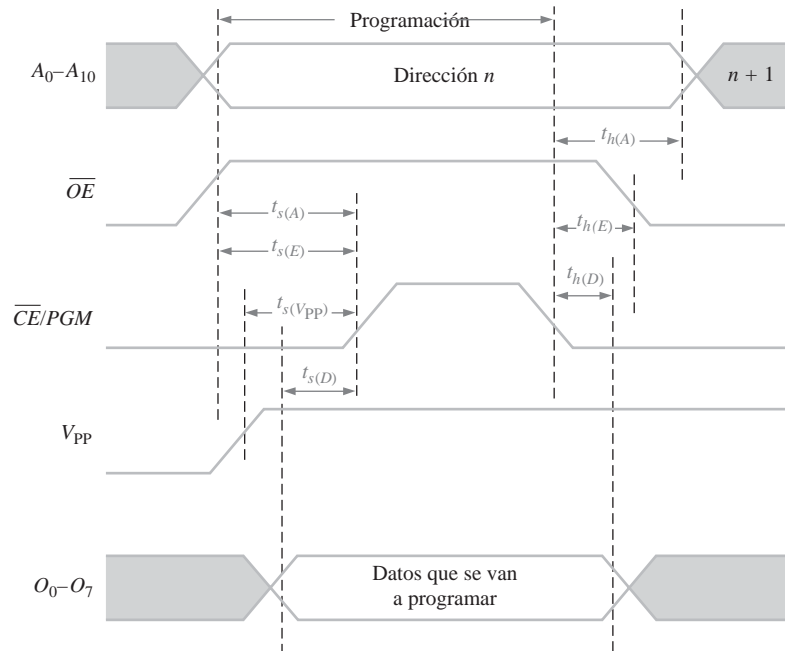
FIGURA 10.31 Símbolo lógico de una UV EPROM 2048 × 8.

Para leer la memoria, la entrada de habilitación de salida ( $\overline{OE}$ ) tiene que estar a nivel BAJO y la entrada de habilitación de chip/programación ( $\overline{CE}/PGM$ , *chip enable/program*) también a nivel BAJO. Para borrar los datos almacenados, el dispositivo se expone a una luz ultravioleta de alta intensidad que pasa a través de la tapa transparente. Una típica lámpara UV borrará los datos en unos 20 ó 25 minutos. Al igual que en la mayoría de las EPROM, tras el borrado todos los bits son 1. La luz ambiente normal contiene la longitud de onda ultravioleta adecuada para hacer que se realice el borrado tras un período de tiempo suficiente. Por tanto, la ventana transparente se ha de mantener tapada.

Para programar el dispositivo, se aplica una tensión continua alta a  $V_{pp}$ , estando  $\overline{OE}$  a nivel ALTO. Los ocho bits de datos que van a ser programados en una dirección determinada se aplican a las salidas ( $Q_0$  hasta  $Q_7$ ) y se selecciona la dirección en las entradas  $A_0$  hasta  $A_{10}$ . A continuación, se aplica un impulso a nivel ALTO a la entrada  $\overline{CE}/PGM$ . Las direcciones se pueden programar en cualquier orden.

En la Figura 10.32 se muestra un cronograma para el modo de programación. Normalmente, estas señales las produce un programador de dispositivos EPROM.

**EEPROM.** Las PROM borrables eléctricamente se pueden borrar y programar mediante impulsos eléctricos. Ya que se pueden grabar y borrar eléctricamente, las EEPROM se pueden programar y borrar rápidamente dentro del propio circuito final con fines de reprogramación.



**FIGURA 10.32** Diagrama de tiempos de un ciclo de programación de una UV EPROM 2048 × 8, que indica los tiempos de establecimiento ( $t_s$ ) y de mantenimiento ( $t_h$ ) críticos.

Los dos tipos de EEPROM son la MOS de puerta flotante y la de silicio óxido nitroso metal (MNOS, *Metal Nitride-Oxide Silicon*). La aplicación de una tensión en la puerta de control de la estructura de puerta flotante permite la eliminación y el almacenamiento de la carga en la puerta flotante.

### REVISIÓN DE LA SECCIÓN 10.4

1. ¿En qué se diferencian las memorias PROM de las ROM?
2. Después del borrado, ¿todos los bits de una EPROM típica son 1 ó 0?
3. ¿Cuál es el modo normal de funcionamiento de una PROM?

## 10.5 MEMORIAS FLASH

La memoria ideal debería tener una alta capacidad de almacenamiento, ser no volátil, disponer de capacidad de lectura y escritura en el propio sistema, tener una velocidad de operación comparativamente rápida y ser efectiva en términos de coste. Las tecnologías de memorias tradicionales, como ROM, PROM, EPROM, EEPROM, SRAM y DRAM exhiben, cada una de ellas, una o más de estas características, pero ninguna tecnología las tiene todas, salvo la memoria flash.

Al finalizar esta sección, el lector deberá ser capaz de:

- Explicar las características básicas de una memoria flash.
- Describir la operación básica de una célula de memoria flash.
- Comparar las memorias flash con otros tipos de memoria

Las *memorias flash* son memorias de lectura/escritura de alta densidad (alta densidad equivale a gran capacidad de almacenamiento de bits) no volátiles, lo que significa que pueden almacenarse los datos indefinida-

mente en ausencia de alimentación. Estas memorias se utilizan frecuentemente en lugar de las unidades de disquete o de las unidades de disco duro de baja capacidad en las computadoras portátiles.

La característica de alta densidad significa que puede incluirse un gran número de celdas en un área de superficie dada del chip; es decir, cuanto más alta sea la densidad, más bits podrán almacenarse en un chip de un tamaño determinado. Esta alta densidad se consigue en las memorias flash con una célula de almacenamiento compuesta por un único transistor MOS de puerta flotante. El bit de datos se almacena como una carga o una ausencia de carga en la puerta flotante, dependiendo de si se desea almacenar un 0 o un 1.

## Célula de memoria flash

La Figura 10.33 representa una célula monotransistor de una memoria flash. El transistor MOS de puerta apilada consta de una puerta de control y una puerta flotante, además del drenador y la fuente. La puerta flotante almacena electrones (carga) si se aplica la suficiente tensión a la puerta de control. *Se almacena un 0 cuando existe una cantidad significativa de carga y un 1 cuando la carga es menor, o inexistente.* La cantidad de carga presente en la puerta flotante determina si el transistor se activará y conducirá corriente del drenador a la fuente cuando se aplique una tensión de control durante una operación de lectura.

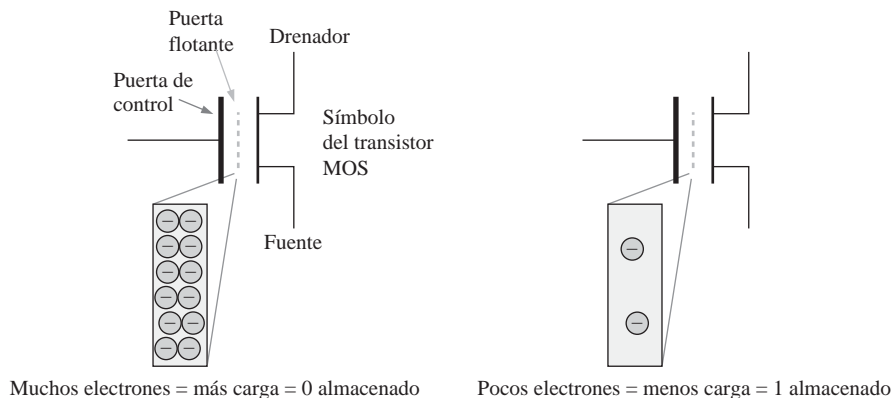


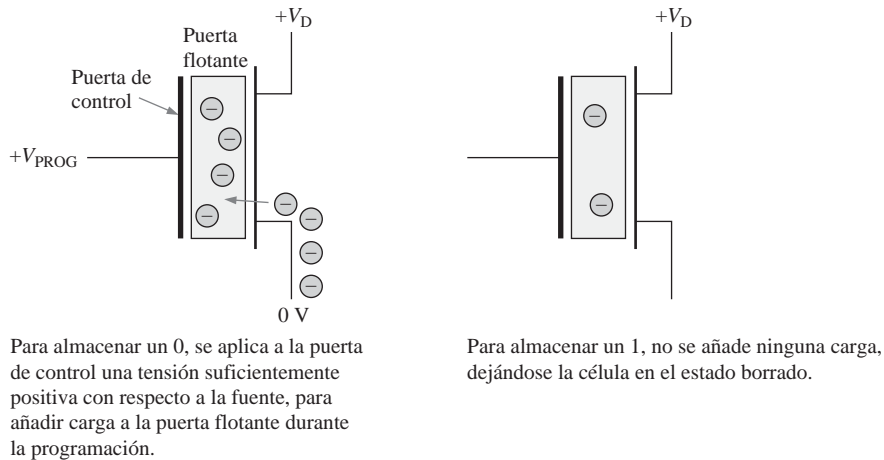
FIGURA 10.33 La célula de almacenamiento de una memoria flash.

## Funcionamiento básico de la memoria flash

Hay tres operaciones principales en una memoria flash: la operación de *programación*, la operación de *lectura* y la operación de *borrado*.

**Programación.** Inicialmente, todas las células se encuentran en el estado 1, porque la carga fue eliminada de las células en una operación previa de borrado. La operación de programación añade electrones (carga) a la puerta flotante de aquellas células que deban almacenar un 0. No se añade carga a aquellas células que deban almacenar un 1. La aplicación a la puerta de control de una tensión suficientemente positiva con respecto a la fuente, durante la programación, atrae electrones a la puerta flotante, como indica la Figura 10.34. Una vez programada, una célula puede conservar la carga durante 100 años sin necesidad de aplicar una alimentación externa.

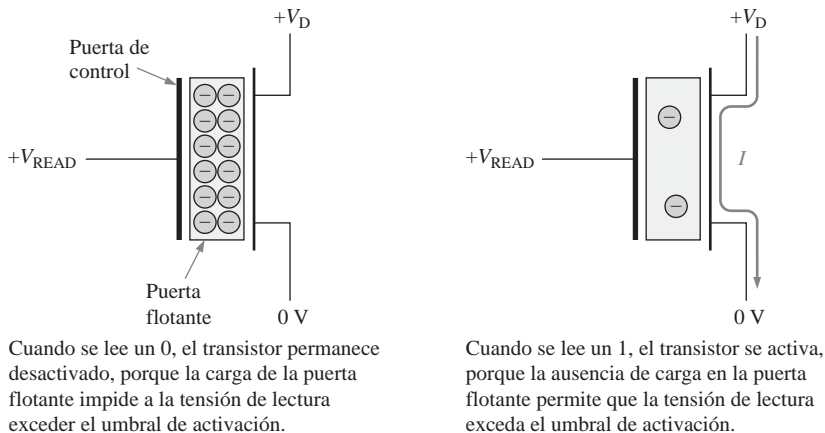
**Lectura.** Durante una operación de lectura, se aplica una tensión positiva a la puerta de control. La cantidad de carga presente en la puerta flotante de una célula determina si la tensión aplicada a la puerta de control activará, o no, el transistor. Si hay almacenado un 1, la tensión de la puerta de control es suficiente para activar el transistor. Si hay almacenado un 0, el transistor no se activará, porque la tensión de la puerta de control no



**FIGURA 10.34** Ilustración simplificada del proceso de almacenamiento de un 0 o un 1 en una célula flash durante la operación de programación.

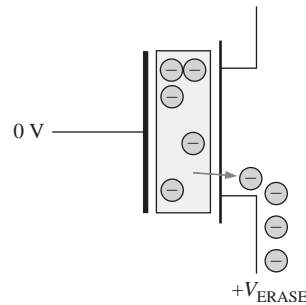
es suficiente para contrarrestar la carga negativa almacenada en la puerta flotante. Piense en la carga de la puerta flotante como en una fuente de tensión que se opone a la tensión aplicada a la puerta de control durante la lectura. Desde este punto de vista, la carga de la puerta flotante asociada a un 0 almacenado evita que la tensión de la puerta de control alcance el umbral de activación, mientras que la carga pequeña o nula asociada con un 1 almacenado permite a la tensión de la puerta de control exceder dicho umbral de activación.

Cuando el transistor se activa, existe corriente desde el drenador hacia la fuente del transistor de la célula. La presencia o ausencia de esta corriente es detectada para indicar un 1 o un 0, respectivamente. Esta idea básica se ilustra en la Figura 10.35.



**FIGURA 10.35** La operación de lectura de una célula flash de una matriz.

**Borrado.** Durante una operación de borrado, se elimina la carga de todas las células de memoria. Para ello, se aplica a la fuente del transistor una tensión suficientemente positiva con respecto a la puerta de control. Esta polaridad es precisamente la opuesta a la utilizada durante la programación. Esta tensión atrae a los electrones de la puerta flotante y hace que ésta se vacíe de carga, como se ilustra en la Figura 10.36. Las memorias flash siempre se borran antes de volver a ser programadas.



Para borrar una célula, se aplica a la fuente una tensión suficientemente positiva con respecto a la puerta de control, con el fin de extraer la carga de la puerta flotante durante la operación de borrado.

**FIGURA 10.36** Ilustración simplificada del proceso de eliminación de la carga de una célula durante la operación de borrado.

## Matriz básica de una memoria flash

La Figura 10.37 muestra una matriz simplificada de células de memoria flash. Sólo se accede a una fila cada vez. Cuando una celda de una línea de bit dada se activa (un 1 almacenado) durante una operación de lectura, existirá corriente a través de la línea de bit, lo que producirá una caída de tensión a través de la carga activa. Esta caída de tensión se compara con una tensión de referencia mediante un circuito comparador, generándose un nivel de salida que indica que hay un 1. Si hay un 0 almacenado, no hay corriente en la línea de bit o ésta es muy pequeña, generándose un nivel opuesto a la salida del comparador.

## Comparación de las memorias flash con otras memorias

Vamos a comparar las memorias flash con los otros tipos de memorias con los que ya estamos familiarizados.

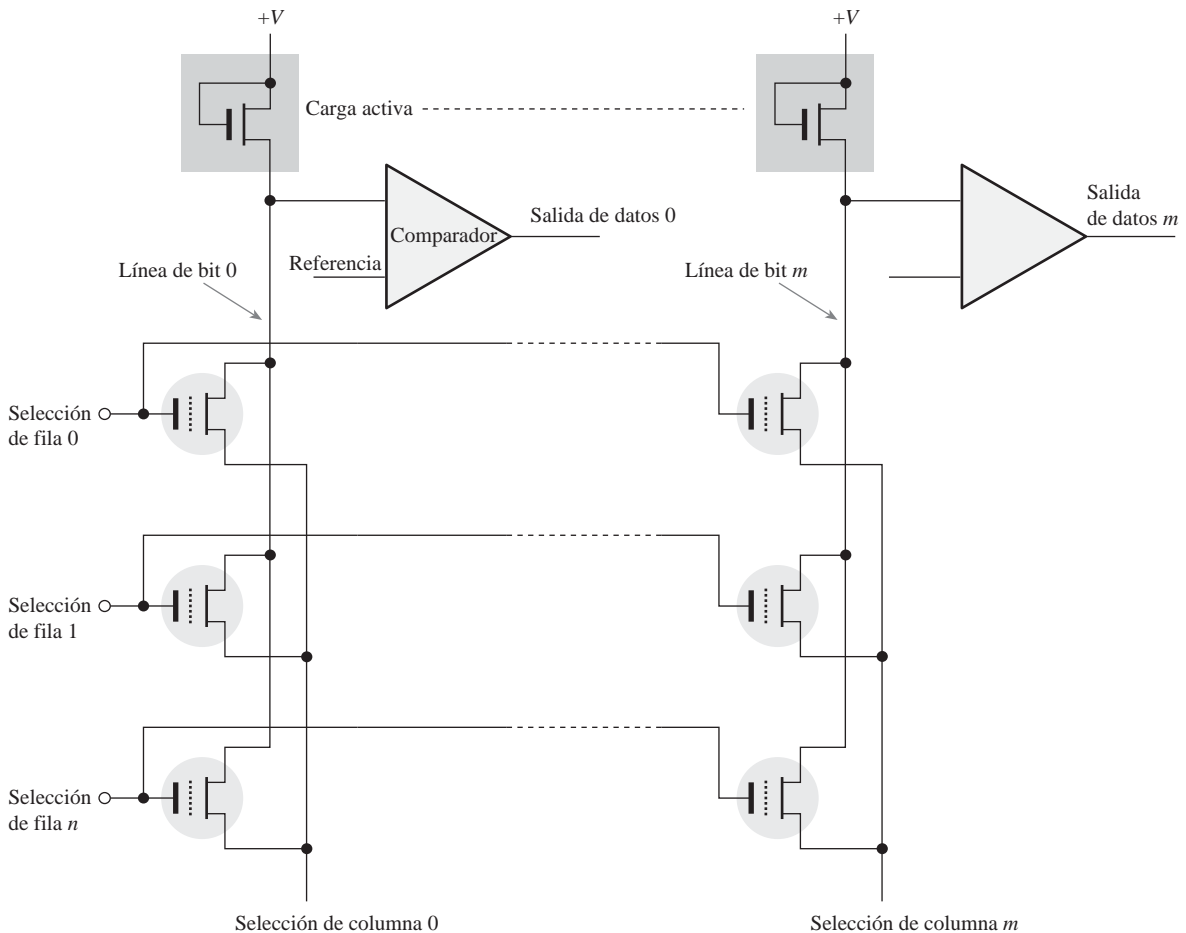
**Flash frente a ROM, EPROM y EEPROM.** Las memorias de sólo lectura son dispositivos de alta densidad y no volátiles. Sin embargo, una vez que se ha programado una ROM, su contenido no puede nunca alterarse. También la programación inicial es un proceso costoso y que consume tiempo.

Aunque la EPROM es una memoria de alta densidad no volátil, sólo se puede borrar extrayéndola del sistema y utilizando luz ultravioleta. Sólo se puede reprogramar utilizando un equipo especial.

La EEPROM tiene una estructura de celda más compleja que la ROM y la EPROM y su densidad no es tan alta, aunque puede reprogramarse sin sacarse del sistema. Debido a su densidad mucho menor, el coste por bit es mayor que en las ROM y EPROM.

Una memoria flash se puede reprogramar fácilmente dentro del sistema ya que, esencialmente, es un dispositivo de LECTURA/ESCRITURA. La densidad de una memoria flash es comparable a la de la ROM y la EPROM, ya que ambas utilizan celdas de un único transistor. Una memoria flash (al igual que una ROM, EPROM o EEPROM) es no volátil, lo que permite almacenar los datos indefinidamente sin alimentación.

**Flash frente a SRAM.** Como se ha explicado, las memorias estáticas de acceso aleatorio son dispositivos de LECTURA/ESCRITURA volátiles. Una SRAM requiere una alimentación constante para mantener los datos almacenados. En muchas aplicaciones, se utiliza una batería de reserva para evitar la pérdida de datos, si la fuente de alimentación principal se apaga. Sin embargo, puesto que siempre existe la posibilidad de que la batería falle, el mantenimiento indefinido de los datos en una SRAM no se puede garantizar. Como las celdas de memoria de una SRAM son, básicamente, un *latch* formado por varios transistores, la densidad es relativamente baja.



**FIGURA 10.37** Matriz básica de una memoria flash.

Una memoria flash también es una memoria de LECTURA/ESCRITURA, pero, a diferencia de la SRAM, no es volátil. También, una memoria flash tiene una densidad mucho más alta que una SRAM.

**Flash frente a DRAM.** Las memorias de acceso aleatorio son dispositivos de LECTURA/ESCRITURA volátiles de alta densidad. Las DRAM no sólo requieren una alimentación constante para mantener los datos, sino que también los datos almacenados deben refrescarse frecuentemente. En muchas aplicaciones, debe utilizarse con la DRAM un almacenamiento de reserva, como por ejemplo un disco duro.

Las memorias flash tienen densidades más altas que las DRAM, porque una celda de memoria está formada por un transistor y no es necesario realizar el refresco, mientras que una celda DRAM es un transistor más un condensador que tiene que recargarse. Típicamente, una memoria flash consume menos potencia que una DRAM equivalente, y en muchas aplicaciones puede usarse para reemplazar un disco duro.

La Tabla 10.2 facilita un resumen de la comparación de las tecnologías de memorias.

### REVISIÓN DE LA SECCIÓN 10.5

1. ¿Cuáles son los tipos de memoria no volátiles?
2. ¿Cuál es la ventaja principal de una memoria flash sobre una SRAM o una DRAM?
3. Enumerar los tres modos de operación de una memoria flash.

| Tipo de memoria | No volátil | Alta densidad | Celda con un sólo transistor | Re-escribible en el sistema final |
|-----------------|------------|---------------|------------------------------|-----------------------------------|
| Flash           | Sí         | Sí            | Sí                           | Sí                                |
| SRAM            | No         | No            | No                           | Sí                                |
| DRAM            | No         | Sí            | Sí                           | Sí                                |
| ROM             | Sí         | Sí            | Sí                           | No                                |
| EPROM           | Sí         | Sí            | Sí                           | No                                |
| EEPROM          | Sí         | No            | No                           | Sí                                |

TABLA 10.2 Comparación de los tipos de memoria.

## 10.6 EXPANSIÓN DE MEMORIAS

Las memorias disponibles se pueden ampliar para incrementar la longitud de palabra (número de bits en cada dirección) o la capacidad de palabra (número de direcciones diferentes), o ambas. La expansión de memoria se consigue añadiendo el número apropiado de chips de memoria a los buses de dirección, datos y control. También se presentan los módulos de expansión de memoria SIMM, DIMM y RIMM.

Al finalizar este capítulo, el lector deberá ser capaz de:

- Definir la expansión de la longitud de palabra.
- Mostrar cómo se amplía la longitud de palabra de una memoria.
- Definir la expansión de la capacidad de palabra.
- Mostrar cómo se amplía la capacidad de palabra de una memoria.

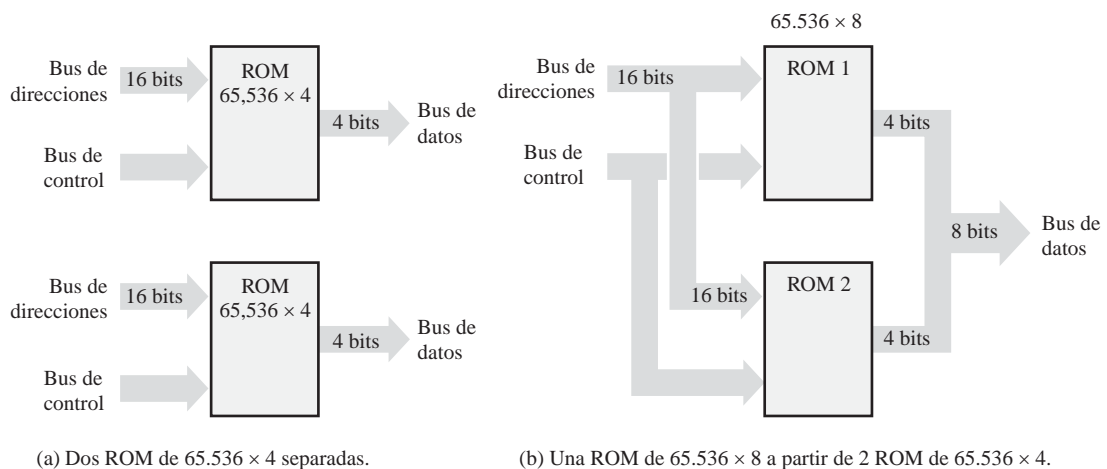


FIGURA 10.38 Expansión de dos ROM de  $65.536 \times 4$  a una ROM de  $65.536 \times 8$ , que ilustra la expansión de la longitud de palabra.

## Expansión de la longitud de palabra

Para aumentar la **longitud de palabra** de una memoria, el número de bits del bus de datos debe aumentarse. Por ejemplo, se puede conseguir una longitud de palabra de 8 bits utilizando dos memorias, teniendo cada una de ellas palabras de 4 bits, como se ilustra en la Figura 10.38(a). Como puede ver en la parte (b), el bus de direcciones de 16 bits se conecta normalmente a ambas memorias, de modo que la memoria combinada tenga el mismo número de direcciones ( $2^{16} = 65.536$ ) que cada memoria individual. Los buses de datos de 4 bits de las dos memorias se combinan para formar un bus de datos de 8 bits. De este modo, cuando se selecciona una dirección, se producen ocho bits en el bus de datos, cuatro para cada memoria. El Ejemplo 10.2 muestra los detalles de la expansión de  $65.536 \times 4$  a  $65.536 \times 8$ .

### EJEMPLO 10.2

Expandir la ROM de  $65.536 \times 4$  ( $64k \times 4$ ) de la Figura 10.39 para obtener una ROM de  $64k \times 8$ . Observe que “64k” es una abreviatura aceptada para designar 65.536. Pero, ¿por qué no se emplea “65k”? Posiblemente, porque 64 es una potencia de dos.

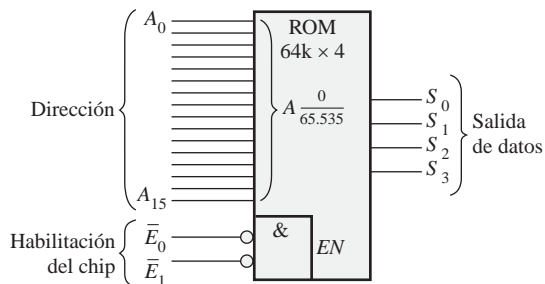


FIGURA 10.39 Una ROM de  $64k \times 4$ .

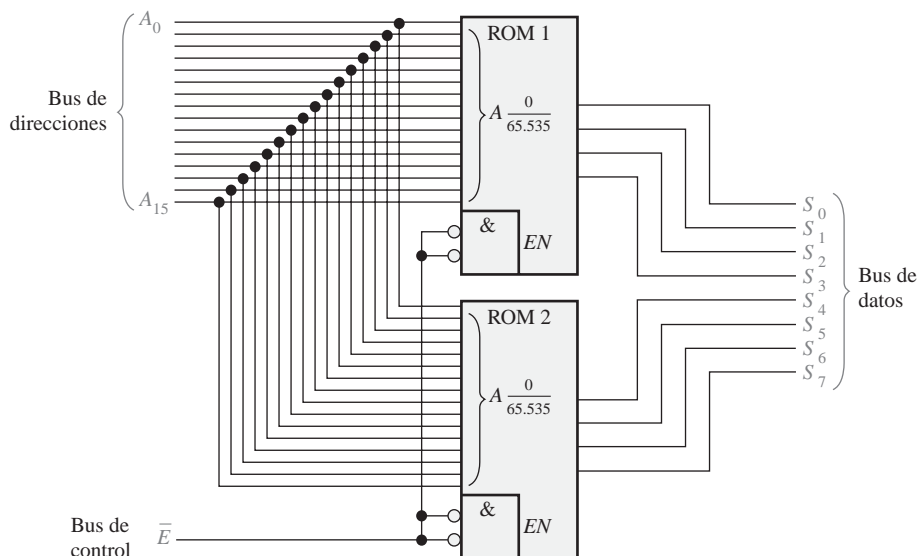


FIGURA 10.40



**Solución**

Las dos ROM de  $64k \times 4$  se conectan como se muestra en la Figura 10.40. Observe que se accede a una dirección específica en la ROM 1 y en la ROM 2 a la vez. Los cuatro bits de la dirección seleccionada de la ROM 1 y los cuatro bits correspondientes de la ROM 2 se llevan en paralelo para formar una palabra de 8 bits en el bus de datos. Observe también que un nivel BAJO en la línea de habilitación de chip  $\bar{E}$ , que actúa como un bus de control simple, activa *ambas* memorias.

**Problema relacionado**

Describir la manera de expandir una ROM de  $64k \times 1$  para obtener una ROM de  $64k \times 8$ .

**EJEMPLO 10.3**

Utilizar las memorias del Ejemplo 10.2 para formar una ROM de  $64k \times 16$ .

**Solución**

En este caso se necesita una memoria que almacene 65.356 palabras de 16 bits. Se requieren cuatro ROM de  $64k \times 4$  para realizar el trabajo, como se muestra en la Figura 10.41.

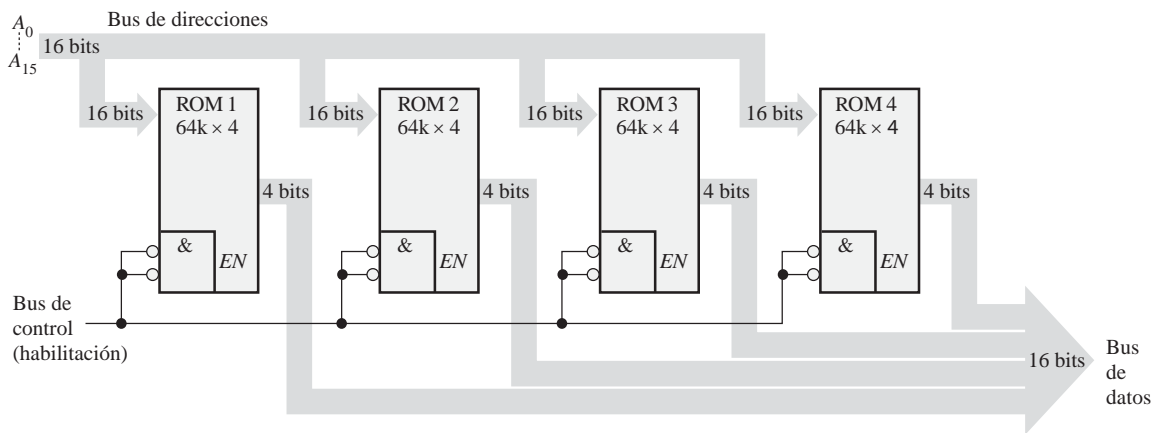


FIGURA 10.41

**Problema relacionado**

¿Cuántas ROM de  $64k \times 1$  se requerirían para implementar la memoria mostrada en la Figura 10.41?

Una ROM sólo tiene salidas de datos, pero una RAM tiene entradas y salidas de datos. Para realizar la expansión de la longitud de palabra en una RAM (SRAM o DRAM), las entradas y salidas de datos forman el bus de datos. Puesto que las líneas de entrada de datos y las correspondientes líneas de salida de datos deben conectarse juntas, se requieren buffers triestado. La mayoría de las RAM proporcionan circuitería de tres estados interna. La Figura 10.42 ilustra la expansión de la RAM para incrementar la longitud de palabra.

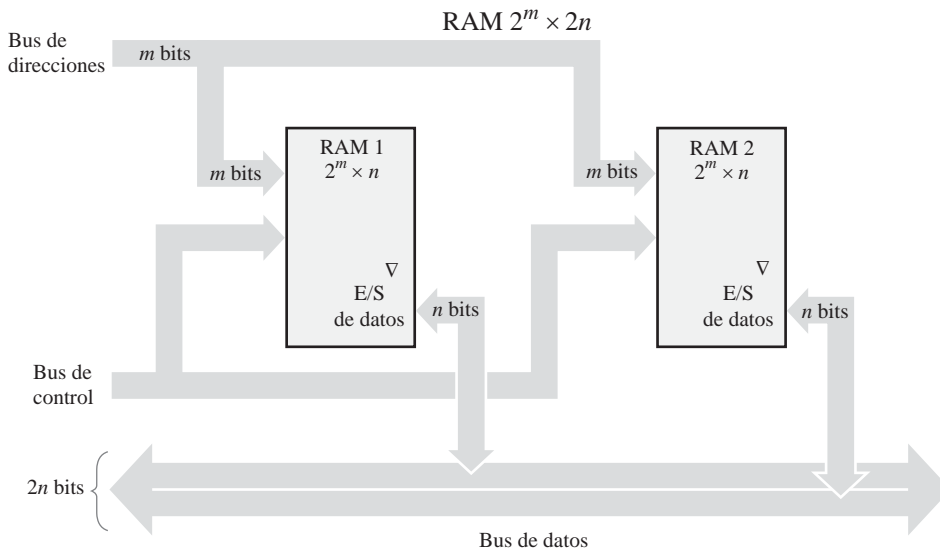


FIGURA 10.42 Ilustración de la expansión de palabra, con dos RAM de  $2^m \times n$  unidas para formar una RAM de  $2^m \times 2n$ .

### EJEMPLO 10.4

Utilizar memorias SRAM de  $1M \times 4$  para crear una SRAM de  $1M \times 8$ .

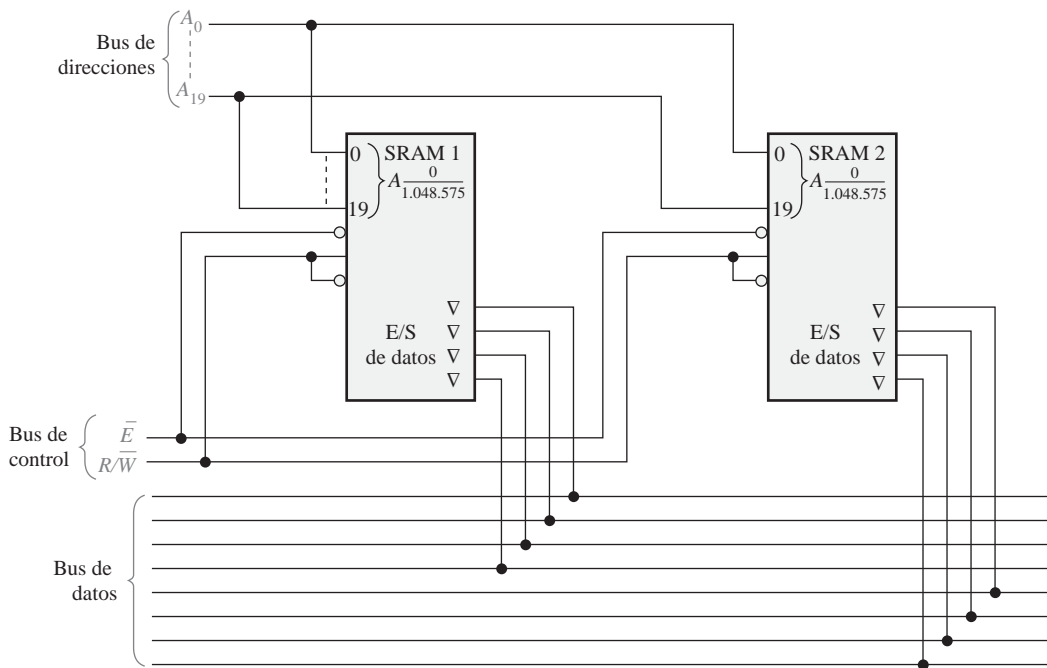
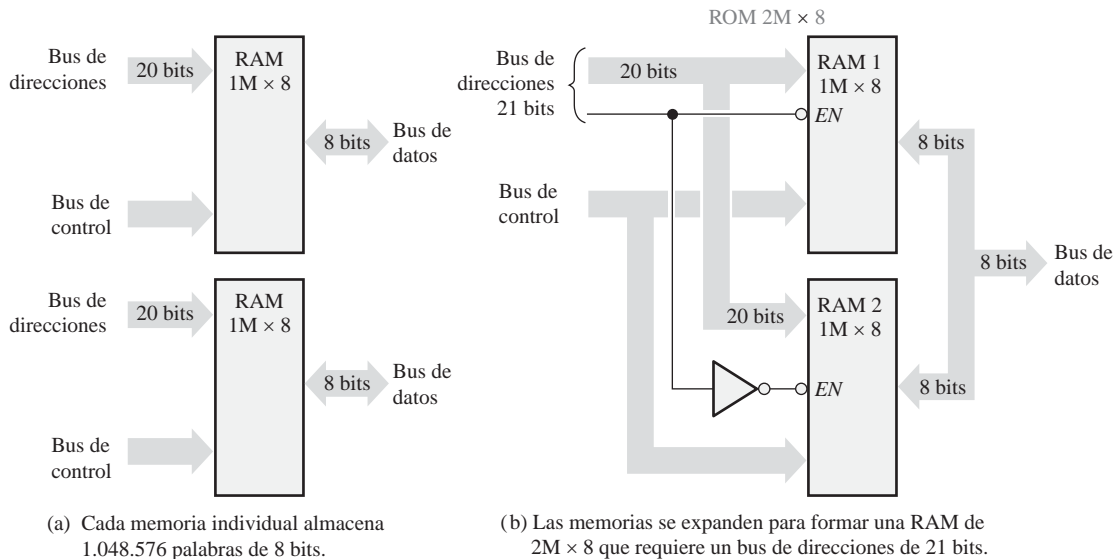


FIGURA 10.43

|                             |  |
|-----------------------------|--|
| <b>Solución</b>             | Se conectan dos SRAM de $1M \times 4$ como muestra el diagrama de bloques simplificado de la Figura 10.43. |
| <b>Problema relacionado</b> | Utilizar memorias SRAM de $1M \times 8$ para crear una SRAM de $1M \times 16$ .                            |

### Expansión de la capacidad de palabra

Cuando las memorias se amplían para incrementar la **capacidad de palabra**, el *número de direcciones se aumenta*. Para conseguir este incremento, el número de bits de dirección se debe aumentar como se ilustra en la Figura 10.44, en la que se expanden dos RAM de  $1M \times 8$  para formar una memoria de  $2M \times 8$ .



**FIGURA 10.44** Expansión de la capacidad de palabra.

Cada memoria individual tiene 20 bits de dirección para seleccionar 1.048.576 direcciones, como se muestra en la parte (a). La memoria expandida tiene 2.097.152 direcciones y, por tanto, requiere 21 bits de dirección, como se muestra en la parte (b). El bit de dirección número 21 se utiliza para activar el chip de memoria adecuado. El bus de datos de la memoria expandida resultante sigue siendo de ocho bits. En el Ejemplo 10.5 se ilustran los detalles de esta expansión.

|  |   |
|--|---|
| <b>EJEMPLO 10.5</b>  |   |
| Utilizar memorias RAM de $512k \times 4$ para implementar una memoria de $1M \times 4$ . |   |
| <b>Solución</b>  | La expansión del direccionamiento se consigue conectando la entrada de habilitación de chip ( $\bar{E}_0$ ) al vigésimo bit de dirección ( $A_{19}$ ) como muestra la Figura 10.45. La entrada $\bar{E}_1$ se usa como entrada de habilitación común a las dos memorias. Cuando el vigésimo bit de dirección ( $A_{19}$ ) está a nivel BAJO, se selecciona la RAM 1 (la RAM 2 está desactivada), y los 19 bits de dirección de menor orden ( $A_0-A_{18}$ ) permiten acceder a las direcciones de la RAM 1. |

Cuando el vigésimo bit de dirección ( $A_{19}$ ) está a nivel ALTO, la RAM 2 se activa debido al nivel BAJO en la salida del inversor (la RAM 1 está desactivada), y los 19 bits de dirección de menor orden ( $A_0-A_{18}$ ) permiten acceder a cada una de las direcciones de la RAM 2.

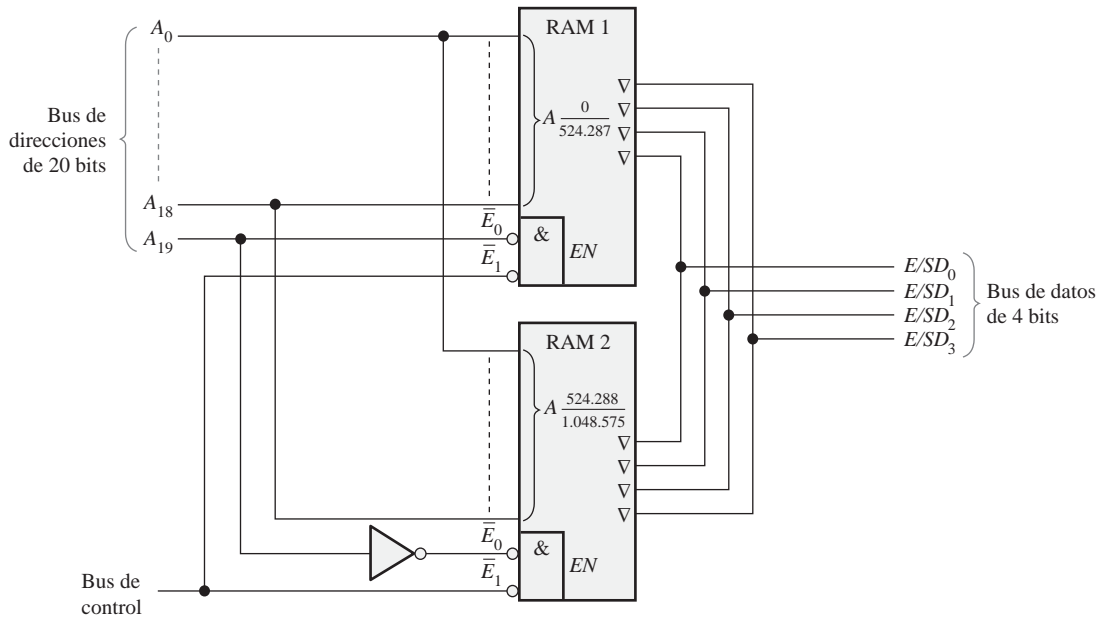


FIGURA 10.45

**Problema relacionado** ¿Cuáles son los rangos de direcciones de la RAM 1 y de la RAM 2 de la Figura 10.45?

## Módulos de memoria

Las memorias se suelen suministrar en forma de módulos de memoria de terminal simple (*single in-line memory module*, **SIMM**) o como módulos de memoria de terminal doble (*dual in-line memory module*, **DIMM**). Los módulos SIMM y los más recientes módulos DIMM son pequeñas tarjetas de circuito impreso en las que se montan chips de memoria con las entradas y salidas conectadas a un conector de borde situado en la parte inferior de la tarjeta de circuito. Los módulos DIMM son generalmente más rápidos, pero sólo pueden ser instalados en las máquinas más modernas, que hayan sido diseñadas para admitirlos.

Los módulos SIMM se clasifican en módulos de 30 y de 72 pines. Ambos tipos se ilustran en la Figura 10.46. Aunque las capacidades de memoria disponibles para los módulos SIMM pueden variar entre 256 KB y 32 MB, la diferencia fundamental entre las dos configuraciones de pines es el tamaño del bus de datos. Generalmente, los módulos SIMM de 30 pines están diseñados para buses de datos de 8 bits, necesiándose más módulos SIMM para poder manejar más bits de datos. Los módulos SIMM de 72 pines admiten un bus de datos de 32 bits, por lo que hacen falta un par de módulos SIMM para los buses de datos de 64 bits.

Los módulos DIMM tienen un aspecto similar a los módulos SIMM, pero proporcionan una mayor densidad de memoria con sólo un incremento relativamente pequeño en el tamaño físico. La diferencia clave

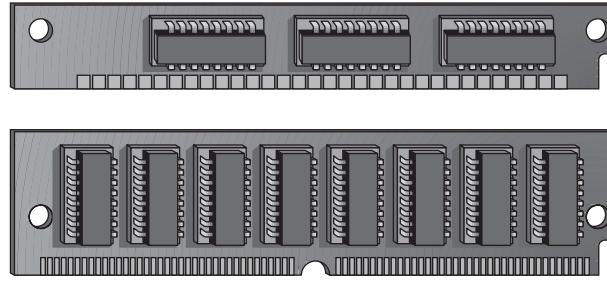


FIGURA 10.46 Módulos SIMM de 30 y de 72 pines.

estriba en que los módulos DIMM distribuyen los pines de entrada y salida en ambos lados de la tarjeta de circuito impreso, mientras que los módulos SIMM sólo emplean uno de los lados. Las configuraciones normales de los módulos DIMM son las de 72 pines, 100 pines, 144 pines y 168 pines, con las que se pueden emplear buses de datos de 32 y de 64 bits. Por regla general, la capacidad de los módulos DIMM va de 4 a 512 MB.

Los módulos SIMM y DIMM se enchufan en zócalos situados en una tarjeta de sistema, como los ilustrados en la Figura 10.47, siendo lo normal que haya varios zócalos disponibles para la expansión de memoria. Por supuesto, los zócalos para los módulos SIMM y DIMM son diferentes y no son intercambiables.

Otro módulo estándar de memoria, similar al módulo DIMM, pero con un bus de mayor velocidad, es el módulo RIMM (*Rambus In Line Memory Module*). Muchas computadoras portátiles utilizan una variante del módulo DIMM llamada SODIMM, que es de menor tamaño, tiene 144 pines y una capacidad de hasta 256 MB.

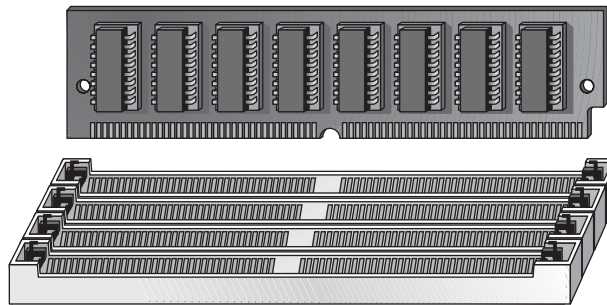


FIGURA 10.47 Un módulo SIMM/DIMM insertado en un zócalo de una tarjeta de un sistema.

### CONSEJOS PRÁCTICOS

Los componentes de memoria son extremadamente sensibles a la electricidad estática. Tome las siguientes precauciones cuando esté manejando chips de memoria o módulos de tipo SIMM y DIMM:

- Antes de manejar los componentes, descargue la electricidad estática de su cuerpo tocando una superficie puesta a masa, o lleve una muñequera de puesta a tierra con una resistencia de alto valor, si dispone de una. Una toma de tierra cómoda y fiable es la que puede encontrarse en los enchufes eléctricos.
- No extraiga los componentes de sus bolsas antiestáticas hasta que esté listo para instalarlos.
- No deposite componentes sobre las bolsas antiestáticas, porque sólo el interior de las mismas es antiestático.

**CONSEJOS  
PRÁCTICOS**

- Cuando esté manejando módulos SIMM o DIMM, sosténgalos por los bordes o por la abrazadera metálica de montaje. No toque los componentes de las placas ni los terminales del conector de borde.
- No deslice nunca un componente sobre ningún tipo de superficie.
- Evite la presencia de plástico, vinilo, espuma de poliestireno y nilón en el área de trabajo.

Cuando instale módulos SIMM o DIMM, siga estos pasos:

1. Alinee las muescas de las tarjetas del módulo SIMM o DIMM con las muescas del zócalo de memoria.
2. Presione firmemente el módulo hasta haberlo encajado en el zócalo.
3. Generalmente, los cierres en ambos lados del zócalo se ajustarán al insertar completamente el módulo. Estos cierres también permiten liberar el módulo, para poderlo sacar del zócalo.

**REVISIÓN DE  
LA SECCIÓN 10.6**

1. ¿Cuántas RAM de  $16k \times 1$  se requieren para conseguir una memoria con una capacidad de palabra de  $16k$  y una longitud de palabra de ocho bits?
2. Para expandir la memoria de  $16k \times 8$  de la cuestión anterior en una organización de  $32k \times 8$ , ¿cuántas RAM de  $16k \times 1$  se requieren?
3. ¿Qué significa SIMM?
4. ¿Qué significa DIMM?
5. ¿Qué significa el término RIMM?

## 10.7 TIPOS ESPECIALES DE MEMORIAS

---

En esta sección, se cubren la memoria FIFO (*First In-First Out*, primero en entrar primero en salir), la memoria LIFO (*Last In-First Out*, último en entrar primero en salir), la pila de memoria y el dispositivo de memoria de acoplamiento carga.

Al finalizar esta sección, el lector deberá ser capaz de:

- Describir una memoria FIFO. ■ Describir una memoria LIFO. ■ Utilizar las pilas de memoria.
- Explicar cómo se usa una parte de una RAM como pila de memoria. ■ Describir una memoria básica CCD.

### Memorias FIFO (*First In-First Out*)

Este tipo de memoria está formado por una disposición de registros de desplazamiento. El término **FIFO** hace referencia al funcionamiento básico de este tipo de memoria, en la que el primer bit de datos que se escribe es el primero que se lee.

En la Figura 10.48 se ilustra una diferencia importante entre un registro de desplazamiento convencional y un registro FIFO. En un registro convencional, un bit de datos se desplaza a través del registro sólo cuando se introducen nuevos datos; en un registro FIFO, un bit de datos atraviesa el registro hasta situarse en la posición de bit más a la derecha que esté vacía.

La Figura 10.49 es el diagrama de bloques de una memoria serie FIFO. Esta memoria en particular tiene cuatro registros de datos serie de 64 bits y un registro de control de 64 bits (registro de marca). Cuando los datos se introducen mediante un impulso de desplazamiento de entrada, automáticamente, bajo el control del

| Registro de desplazamiento convencional |   |   |   |   |        |
|---|---|---|---|---|--------|
| Entrada                                 | X | X | X | X | Salida |
| 0                                       | 0 | X | X | X | →      |
| 1                                       | 1 | 0 | X | X | →      |
| 1                                       | 1 | 1 | 0 | X | →      |
| 0                                       | 0 | 1 | 1 | 0 | →      |

X = bits de datos desconocidos.

En un registro de desplazamiento convencional, los datos permanecen a la izquierda hasta que son desplazados por medio de datos adicionales.

| Registro de desplazamiento FIFO |   |   |   |   |        |
|---------------------------------|---|---|---|---|--------|
| Entrada                         | — | — | — | — | Salida |
| 0                               | — | — | — | 0 | →      |
| 1                               | — | — | 1 | 0 | →      |
| 1                               | — | 1 | 1 | 0 | →      |
| 0                               | 0 | 1 | 1 | 0 | →      |

— = posiciones vacías.

En un registro de desplazamiento FIFO, los datos "van cayendo" hacia la derecha.

FIGURA 10.48 Comparación del funcionamiento de un registro convencional y uno FIFO.

registro de marca, se mueven a la posición vacía más próxima a la salida. Los datos no pueden avanzar a las posiciones que están ocupadas. Sin embargo, cuando un bit de datos se desplaza mediante un impulso de desplazamiento de salida, los bits de datos que están en los registros automáticamente se mueven a la posición siguiente hacia la salida. En una memoria FIFO asincrónica, los datos se desplazan hacia fuera independientemente de la entrada de datos, utilizando dos relojes separados.

## Aplicaciones de una FIFO

Un área de aplicación importante del registro FIFO es el caso en que dos sistemas con velocidades diferentes tienen que comunicarse. Los datos pueden entrar en un registro FIFO a una velocidad y salir a otra velocidad distinta. La Figura 10.50 muestra cómo debe emplearse un registro FIFO en estas situaciones.

## Memorias LIFO (Last In-First Out)

Las memorias **LIFO** se encuentran en aplicaciones que utilizan microprocesadores y otros sistemas de computación. Permiten almacenar datos y luego extraerlos en orden inverso; es decir, el último byte de datos almacenado es el primer byte de datos que se recupera.

**Pilas de registros.** Comúnmente, una memoria LIFO se denomina pila *push-down*. En algunos sistemas, se implementa con un grupo de registros, como muestra la Figura 10.51. Una pila puede estar formada por cualquier número de registros, pero el registro superior se denomina *tope de la pila*.

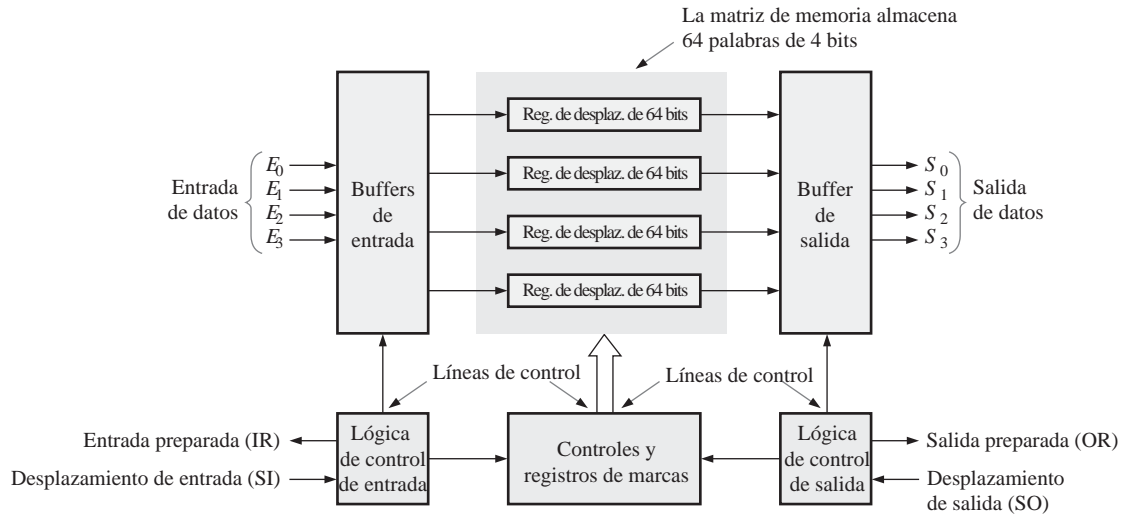


FIGURA 10.49 Diagrama de bloques de una memoria FIFO serie típica.

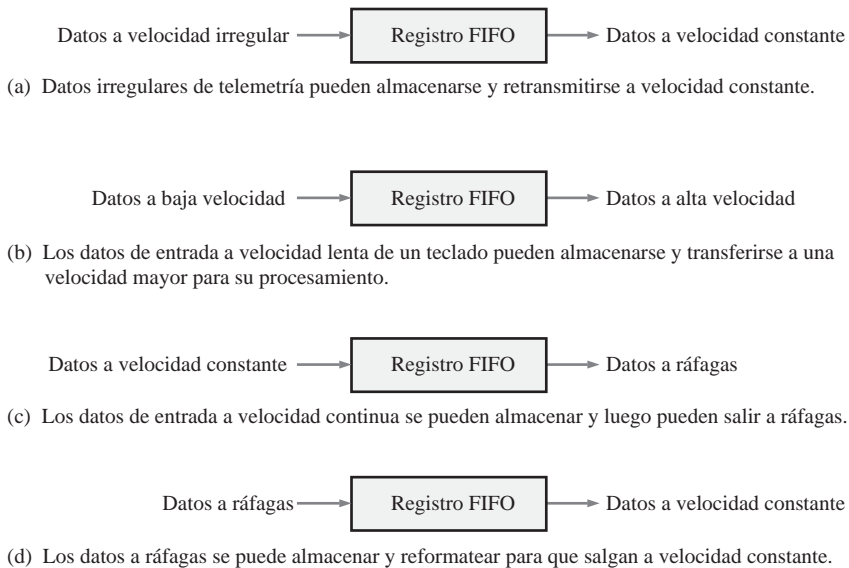


FIGURA 10.50 Ejemplos de registro FIFO para aplicaciones de control de la velocidad de transmisión de datos.

Para ilustrar el principio de estas memorias, se carga en paralelo un byte de datos en el tope de la pila. Cada sucesivo byte empuja al anterior al registro siguiente. Este proceso se muestra en la Figura 10.52. Observe que cada nuevo dato siempre se carga en el registro superior, y los bytes previamente almacenados son empujados hacia abajo dentro de la pila. El nombre de *pila push-down* (pila de empuje hacia abajo) viene de esta característica.

Los bytes de datos se recuperan en orden inverso. El último byte introducido siempre está en el registro superior de la pila, por lo que, cuando sale de la pila, los demás bytes saltan a las siguientes posiciones superiores. Este proceso se ilustra en la Figura 10.53.



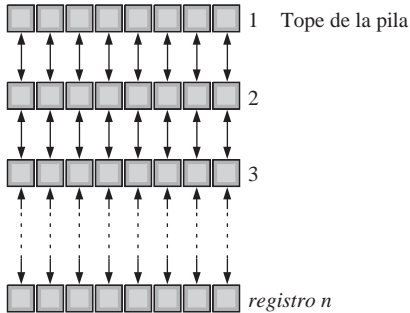


FIGURA 10.51 Pila de registros.

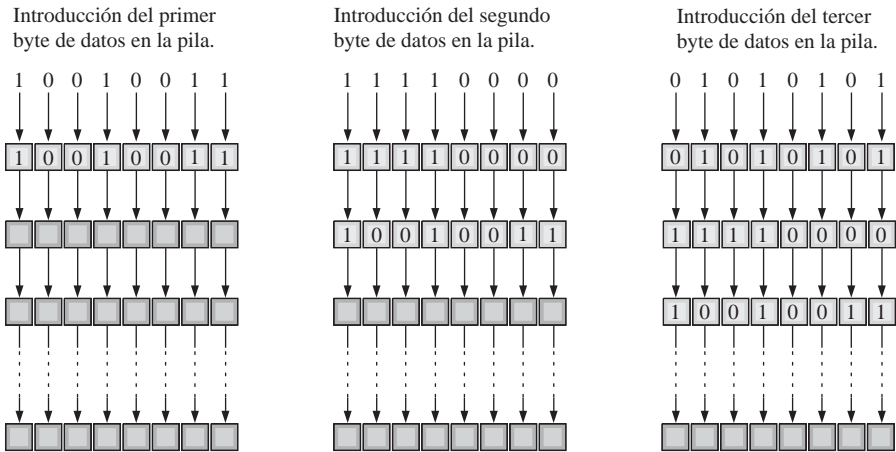


FIGURA 10.52 Ilustración simplificada del proceso de introducción de datos en la pila.

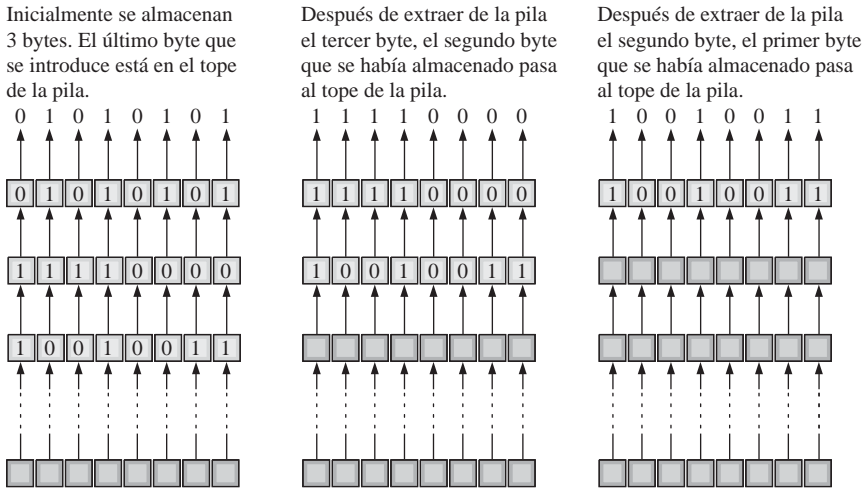
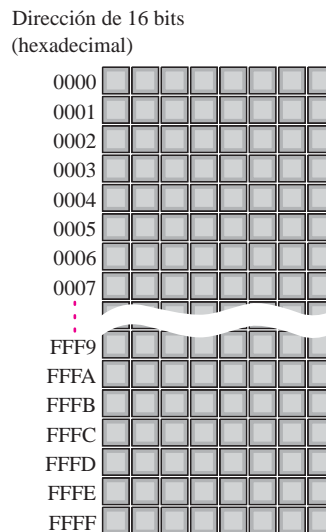


FIGURA 10.53 Ilustración simplificada de extracción de datos de la pila

**Pila RAM.** Otra forma de implementar una memoria LIFO, en algunos sistemas basados en microprocesador, es disponer una sección de la RAM como pila, mejor que utilizar un conjunto de registros dedicados. Como hemos visto, en un registro de pila, los datos se desplazan de arriba a abajo desde una posición a la siguiente. En una pila RAM, los datos en sí no se desplazan, sino que es el tope de la pila lo que se mueve bajo el control de un registro denominado puntero de pila.

Consideremos una memoria de acceso aleatorio organizada en bytes, es decir, en la que cada dirección contiene ocho bits, como muestra la Figura 10.54. La dirección binaria 0000000000001111, por ejemplo, puede escribirse en hexadecimal como 000F. Una dirección de 16 bits puede tener un valor hexadecimal *mínimo* de  $0000_{16}$  y un valor *máximo* de  $FFFF_{16}$ . Utilizando esta notación, una matriz de memoria de 64 kB se puede representar como se muestra en la Figura 10.54. La dirección de memoria más baja es  $0000_{16}$  y la dirección más alta es  $FFFF_{16}$ .

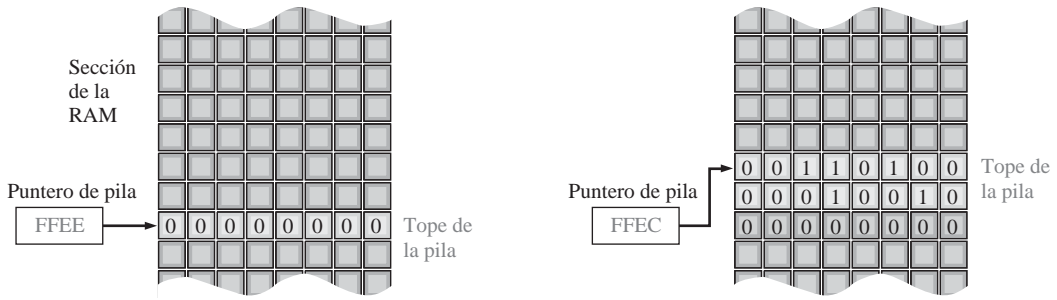


**FIGURA 10.54** Representación de una memoria de 64 kB con direcciones de 16 bits expresadas en hexadecimal.

Ahora, supongamos que se reserva una sección de la RAM para utilizarla como pila. Un registro independiente especial, denominado puntero de pila, contiene la dirección del registro superior de la pila, tal y como se muestra en la Figura 10.55. Se utiliza una representación hexadecimal de 4 dígitos para las direcciones binarias. En la figura, las direcciones se han elegido arbitrariamente, con el único fin de servir de ilustración.

Ahora vamos a ver cómo se introducen los datos en la pila. Inicialmente, el puntero de pila se encuentra en la dirección  $FFEE_{16}$ , que es el tope de la pila, como se indica en la Figura 10.55(a). El puntero de pila se decrementa en dos unidades y toma el valor  $FFEC_{16}$ . Esto hace que el tope de la pila se mueva a una dirección de memoria inferior, como se ve en la Figura 10.55(b). Observe que el tope de la pila no es estacionario como en el caso de la pila de registros fijos, sino que se mueve hacia abajo (direcciones más bajas) por la RAM, según se almacenan las palabras de datos. La Figura 10.55(b) muestra dos bytes (una palabra de datos) que se introducen en la pila. Después de que se almacena la palabra, el tope de la pila está en  $FFEC_{16}$ .

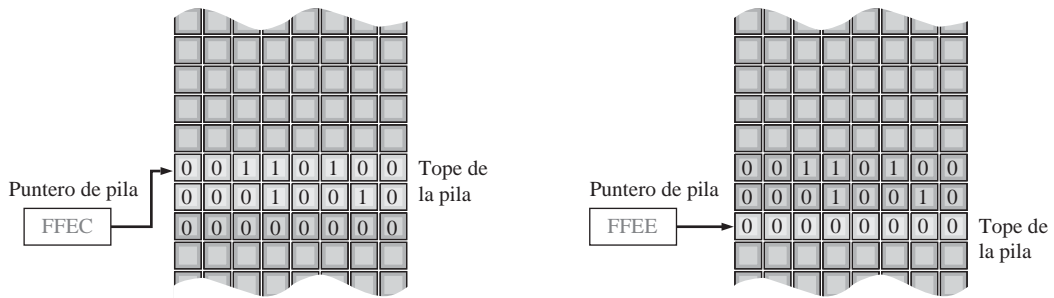
La Figura 10.56 ilustra la operación de extracción de los datos de la pila RAM. La última palabra de datos almacenada en la pila se lee en primer lugar. El puntero de pila, que está en la posición  $FFEC$ , se incrementa en dos unidades para apuntar a la dirección  $FFEE_{16}$  y se realiza una operación de extracción, como puede verse en la parte (b) de la figura. Tenga en mente que las memorias RAM, cuando se leen, no son destructivas, por lo que los datos permanecerán almacenados en la memoria después de la operación de extracción. Una palabra de datos sólo se destruye cuando se escribe sobre ella.



(a) Inicialmente, el puntero de pila está en FFEE antes de que la palabra de datos 0001001000110100 (1234) se introduzca en la pila.

(b) El puntero de pila se decrementa en dos unidades y la palabra de datos 0001001000110100 se coloca en las dos posiciones anteriores al valor original del puntero de pila.

FIGURA 10.55 Ilustración de la operación de introducción de datos en una pila RAM.



(a) El puntero de pila se encuentra en la posición FFEC antes de que se copie (extraiga) la palabra de datos de la pila.

(b) El puntero de pila se incrementa en dos unidades y la última palabra de datos almacenada se copia (se extrae) de la pila.

FIGURA 10.56 Ilustración de la operación de extracción de datos de la pila RAM.

## Memorias CCD

La memoria **CCD** (*charge-coupled device*, dispositivo de acoplamiento de carga) almacena los datos como cargas de condensador. Sin embargo, a diferencia de la RAM dinámica, la celda de almacenamiento no incluye un transistor. La principal ventaja de estas memorias CCD es su alta densidad.

La memoria CCD está formada por largas filas de condensadores semiconductores, denominados *canales*. Los datos se introducen en serie en el canal, depositando una pequeña carga en el condensador si se trata de un 0, y una carga grande si es un 1. Después, estas cargas se desplazan a lo largo del canal mientras que se introducen más datos, de acuerdo con las señales de reloj.

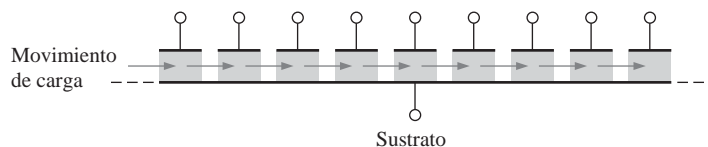


FIGURA 10.57 Un canal CCD (charge-coupled device).

Como en el caso de la DRAM, las cargas se deben refrescar periódicamente. Este proceso se realiza desplazando las cargas en serie a través de un circuito de refresco. La Figura 10.57 muestra el concepto básico de un canal CCD. Como los datos se desplazan en serie a través de los canales, estas memorias tienen un tiempo de acceso relativamente grande. Las matrices CCD se usan en algunas cámaras modernas, para captura de imágenes de vídeo en forma de carga de luz inducida.

### REVISIÓN DE LA SECCIÓN 10.7

1. ¿Qué es una memoria FIFO?
2. ¿Qué es una memoria LIFO?
3. Explicar la operación de introducción de datos en una pila de memoria.
4. Explicar la operación de extracción de datos de una pila de memoria.
5. ¿De qué es abreviatura el término CCD?

## 10.8 MEMORIAS ÓPTICAS Y MAGNÉTICAS

En esta sección nos ocuparemos de los fundamentos de los discos magnéticos, cintas magnéticas, discos magneto-ópticos y discos ópticos. Estos medios de almacenamiento son muy importantes, especialmente en aplicaciones de computadoras, en las que se utiliza almacenamiento masivo no volátil de datos y programas.

Al finalizar esta sección, el lector deberá ser capaz de:

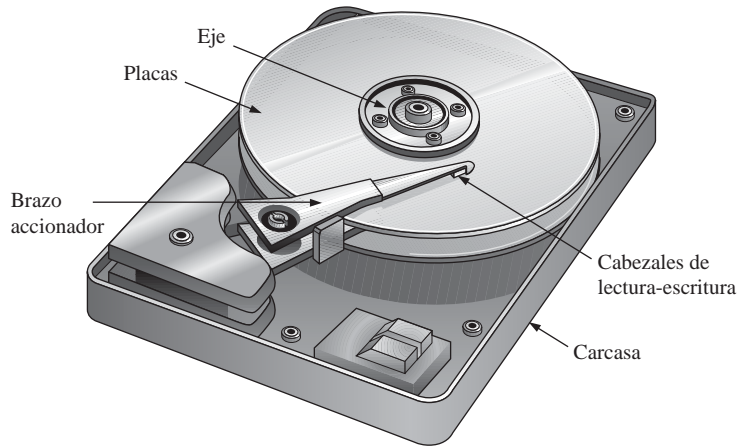
- Describir un disco duro magnético. ■ Describir los discos flexibles. ■ Describir los discos duros extraíbles. ■ Explicar el principio básico de los discos magneto-ópticos. ■ Definir los discos CD-ROM, CD-R y CD-RW. ■ Describir una memoria WORM. ■ Describir el DVD-ROM.

### Almacenamiento magnético

**Disco duro magnético.** Las computadoras emplean el disco duro como dispositivo interno de almacenamiento masivo. Los *discos duros* son “placas” rígidas de aleación de aluminio o de una mezcla de vidrio y cerámica recubiertos con una capa magnética. Hay disponibles unidades de disco duro, con dos tamaños principales de diámetro, 5,25 y 3,5 pulgadas, aunque también existen de 2,5 y 1,75 pulgadas. Las unidades de disco duro se sellan herméticamente para mantener al disco libre de polvo.

Normalmente, se apilan dos o más discos sobre un eje o pivote común, que hace que el conjunto gire a una velocidad de miles de revoluciones por minuto (rpm). Existe una separación entre cada disco, con el fin de permitir el montaje de un cabezal de lectura-escritura en el extremo del brazo accionador, como se muestra en la Figura 10.58. Hay un cabezal de lectura-escritura en cada cara del disco, ya que los datos se graban en ambas caras de la superficie del disco. El brazo accionador de la unidad sincroniza todos los cabezales de lectura-escritura para mantenerlos perfectamente alineados cuando se desplazan por la superficie del disco, con una separación de sólo una fracción de milímetro con respecto al disco. Una pequeña partícula de polvo podría hacer que un cabezal se rompiera, dañando como consecuencia la superficie del disco.

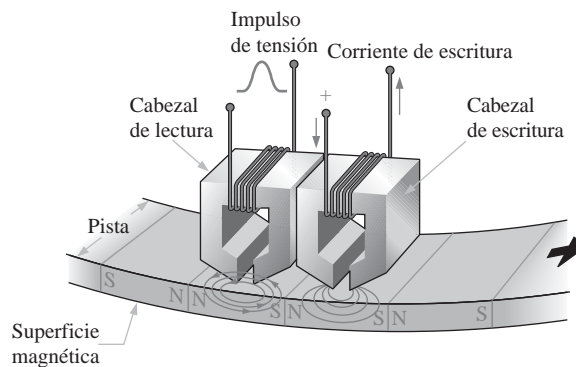
**Principios básicos del cabezal de lectura-escritura.** El disco duro es un dispositivo de acceso aleatorio, ya que puede recuperar datos almacenados en cualquier lugar del disco, en cualquier orden. En la Figura 10.59 se presenta un diagrama simplificado de la operación de lectura-escritura en la superficie magnética. La dirección o polarización de las partículas magnéticas sobre la superficie del disco se controla mediante la dirección de las líneas de flujo magnético (campo magnético) producidas por el cabezal de escritura, según la dirección de un impulso de corriente en el devanado. Este flujo magnético magnetiza un pequeño punto de la superficie del disco en la dirección del campo magnético. Un punto magnetizado con una cierta polaridad representa un



**FIGURA 10.58** Esquema simplificado de una unidad de disco duro.

1 binario, y uno de polaridad opuesta representa un 0 binario. Una vez que se ha magnetizado un punto de la superficie, permanece en dicho estado hasta que se escribe sobre él un campo magnético opuesto.

Cuando un cabezal de lectura pasa por una superficie magnetizada, los puntos magnetizados generan campos magnéticos en el cabezal de lectura, lo que provoca impulsos de tensión en el devanado. La polaridad de estos impulsos depende de la dirección del punto magnetizado e indica si el bit almacenado es un 1 o un 0. A menudo, los cabezales de lectura y de escritura se combinan en una única unidad.



**FIGURA 10.59** Funcionamiento simplificado del cabezal de lectura/escritura.

**Formato del disco duro.** Un disco duro está organizado en pistas y sectores, como muestra la Figura 10.60(a). Cada pista está dividida en una serie de sectores, y cada pista y sector tienen una dirección física que el sistema operativo utiliza para localizar un determinado registro de datos. Normalmente, los discos duros tienen desde unos pocos cientos hasta unos pocos miles de pistas. Como puede ver en la figura, hay un número constante de pistas por sector, utilizando los sectores externos una superficie mayor que los sectores internos. La disposición de pistas y sectores en un disco se denomina *formato*.

La Figura 10.60(b) muestra una pila de disco duro. Las unidades de disco duro difieren en el número de placas apiladas, aunque siempre hay un mínimo de dos. El conjunto de todas las pistas correspondientes de cada placa constituyen lo que se conoce colectivamente como cilindro, como se indica en la figura.



### NOTAS INFORMÁTICAS

Los datos se almacenan en un disco duro en forma de archivos. Llevar la cuenta de la ubicación de los archivos es el trabajo del controlador de dispositivos que gestiona el disco duro (algunas veces denominado BIOS de la unidad de disco duro). El controlador de dispositivo y el sistema operativo de la computadora pueden acceder a dos tablas que se utilizan para controlar la ubicación y los nombres de los archivos. La primera tabla se denomina FAT (*File Allocation Table*, tabla de asignación de archivos). La FAT muestra qué partes se han asignado a archivos específicos y mantiene un registro de los sectores abiertos y los sectores defectuosos. La segunda tabla es el directorio raíz, que contiene el nombre de los archivos, el tipo de archivo, la fecha y hora de creación, el número del *cluster* inicial y otras informaciones referentes al archivo.

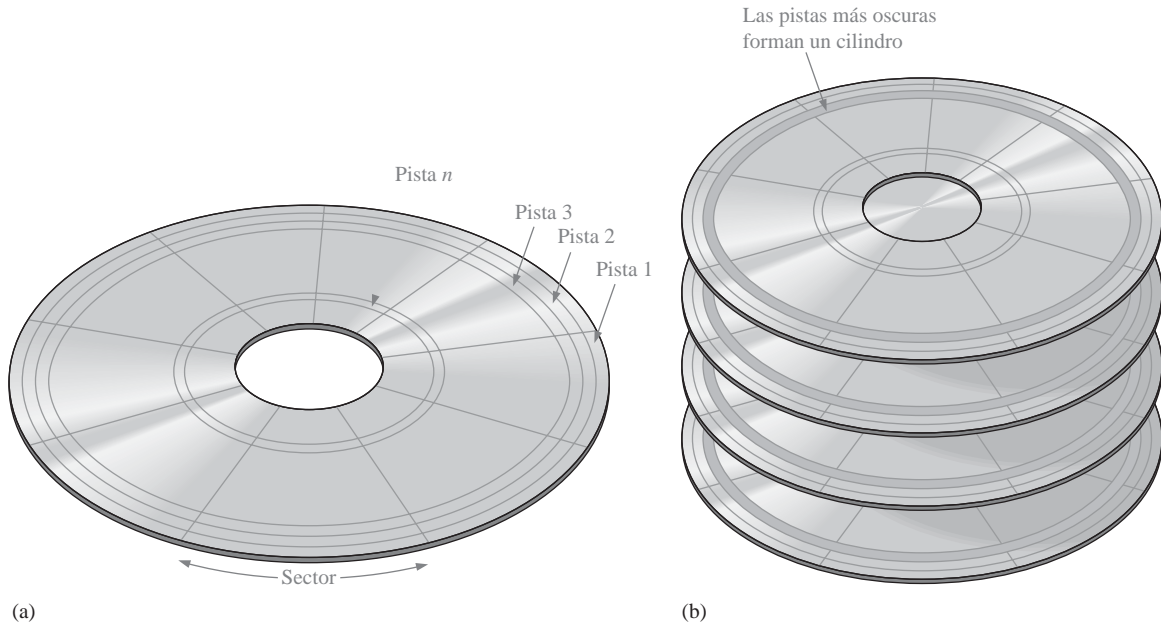


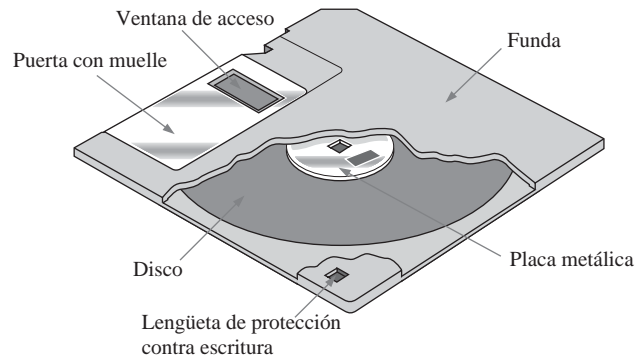
FIGURA 10.60 Organización y formato de un disco duro.

**Funcionamiento del disco duro.** El rendimiento de una unidad de disco duro específica viene determinado por varios parámetros básicos. Una operación de *búsqueda* consiste en el movimiento del cabezal de lectura-escritura hasta la pista deseada. El **tiempo de búsqueda** es el tiempo medio de realización de dicha operación. Normalmente, las unidades de disco duro tienen un tiempo medio de búsqueda de varios milisegundos, dependiendo de cada unidad concreta.

El **período de latencia** es el tiempo que tarda el sector deseado en colocarse debajo del cabezal, una vez que éste se ha posicionado en la pista deseada. El caso peor se produce cuando el sector deseado acaba de pasar la posición del cabezal y está girando, alejándose de la misma. El sector tiene que girar casi una revolución completa hasta alcanzar la posición del cabezal. El *período de latencia medio* supone que el disco debe recorrer media revolución. Obviamente, el período de latencia depende de la velocidad constante de rotación del disco. Las velocidades de rotación de disco son diferentes para distintas unidades de disco aunque, típicamente, son 3.600 rpm, 4.500 rpm, 5.400 rpm y 7.200 rpm. Algunas unidades de disco recientes giran a una velocidad de 10.033 rpm y tienen un período medio de latencia inferior a 3 ms.

La suma del tiempo medio de búsqueda y del período medio de latencia es el *tiempo de acceso* de la unidad de disco.

**Discos flexibles.** El nombre de disco flexible se debe a que este tipo de discos está hecho de un material de poliéster flexible, cubierto por ambas caras con una capa magnética. Los primeros discos flexibles tenían un diámetro de 5,25 pulgadas y estaban contenidos en una funda semiflexible. Los actuales **discos flexibles** o disquetes tienen un diámetro de 3,5 pulgadas y disponen de una funda de plástico rígido, como se muestra en la Figura 10.61. Una puerta con muelle cubre la ventana de acceso, y permanece cerrada hasta que el disquete se introduce en la unidad. El disco dispone de una placa metálica con un agujero para centrar el disquete y otro para hacerlo rotar dentro de la funda protectora. Obviamente, los disquetes son disquetes extraíbles, mientras que los discos duros no. Los discos flexibles se formatean en pistas y sectores de forma similar a los discos duros, excepto por el número de pistas y sectores. Los disquetes de alta densidad de 1,44 MB tienen 80 pistas por cada lado, con 18 sectores.



**FIGURA 10.61** El disco flexible de 3,5 pulgadas (disquete).

**Zip™.** La unidad Zip es un tipo de dispositivo de almacenamiento magnético extraíble que parece ser el posible recambio de los disquetes de capacidad limitada. Al igual que el disco flexible, el cartucho del **disco Zip** es un disco flexible incluido en una carcasa rígida aproximadamente del mismo tamaño que el disquete, aunque más grueso. La unidad Zip es mucho más rápida que la unidad de disquetes, dado que tiene una velocidad de giro de 3.000 rpm frente a las 300 rpm de la unidad de disquete. La unidad Zip tiene una capacidad de almacenamiento de 250 MB, lo que es aproximadamente 173 veces mayor que la capacidad de 1,44 MB del disquete.

**Jaz™.** Otro tipo de dispositivo de almacenamiento magnético es la unidad Jaz, que es similar a una unidad de disco duro excepto en que las dos placas se encuentran dentro de un cartucho extraíble protegido por un obturador a prueba de polvo. Los **cartuchos Jaz** están disponibles con capacidades de almacenamiento de 1 o 2 GB.

**Disco duro extraíble.** Además de las populares unidades extraíbles Zip y Jaz, hay disponibles discos duros extraíbles con capacidades de 80 GB a 250 GB. Tenga presente que la tecnología está cambiando de forma tan rápida que probablemente ya habrá más avances en el momento en que se ponga a leer este libro.

**Cinta magnética.** La cinta se utiliza para realizar copias de seguridad de datos desde dispositivos de almacenamiento masivo y, normalmente, es mucho más lenta en términos de tiempo de acceso, ya que se accede a los datos en serie en lugar de mediante una selección aleatoria. Hay disponibles diversos tipos de cintas, entre los que se incluyen QIC, DAT, 8 mm y DLT.

**QIC** es la abreviatura de *quarter-inch cartridge* (cartucho de cuarto de pulgada) y se parece bastante a un casete de audio con dos carretes en el interior. Los distintos estándares QIC varían desde 36 a 72 pistas que pueden almacenar desde 80 MB hasta 1,2 GB. Las actualizaciones más recientes del estándar Travan han aumentado la longitud de la cinta y la anchura de la misma, permitiendo capacidades de almacenamiento de

hasta 4 GB. Las unidades de cinta QIC utilizan cabezales de lectura-escritura con un único cabezal de escritura y un cabezal de lectura en cada lado. Esto permite que la unidad de cinta compruebe los datos que se acaban de escribir, cuando la cinta está girando en cualquier dirección. En el modo de registro, la cinta se desliza bajo los cabezales de lectura-escritura aproximadamente a una velocidad de 100 pulgadas por segundo (25,4 cm/s), como se indica en la Figura 10.62.

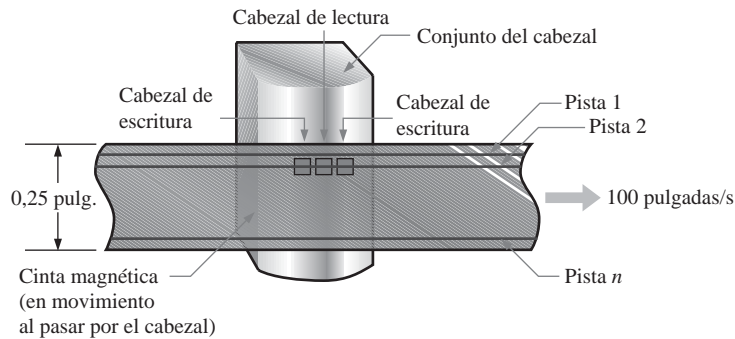


FIGURA 10.62 Cinta QIC.

**DAT**, que son las siglas de Digital Audio Tape (cinta digital de audio), utiliza una técnica denominada grabación por exploración helicoidal. Las cintas DAT ofrecen capacidades de almacenamiento en el rango de los 12 GB, pero son más caras que las QIC.

Un tercer tipo de formato de cinta es la cinta de 8 mm que, originalmente, se diseñó para la industria de vídeo, pero que ha sido adoptada por la industria informática como una forma fiable de almacenar grandes cantidades de datos de computadora. La cinta de 8 mm es similar a la DAT, pero ofrece capacidades de almacenamiento de hasta 25 GB.

**DLT** corresponde a las siglas de *Digital Linear Tape* (cinta digital lineal). La cinta DLT es una cinta de una anchura de media pulgada (1,25 cm), lo que significa que es un 60% más ancha que la cinta de 8 mm y, por tanto, dos veces más ancha que una cinta estándar QIC. Básicamente, la cinta DLT difiere en la forma en que trabaja el mecanismo de la unidad de cinta, para minimizar el deterioro de la cinta en comparación con otros sistemas. La cinta DLT ofrece la mayor capacidad de almacenamiento de todos los soportes físicos de este tipo, con capacidades que alcanzan hasta los 35 GB.

## Almacenamiento magneto-óptico

Como su nombre indica, los dispositivos de almacenamiento magneto-óptico combinan las tecnologías magnética y óptica (láser). Un **disco magneto-óptico** se formatea en pistas y sectores de forma similar a los discos magnéticos.

La diferencia básica entre un disco puramente magnético y un disco magneto-óptico es que la capa magnética utilizada en los discos magneto-ópticos requiere calor para alterar la polarización magnética. Por tanto, el disco magneto-óptico es extremadamente estable a temperatura ambiente, haciendo que los datos no cambien. Para escribir un bit de datos, se enfoca un haz láser de alta potencia sobre un punto muy pequeño del disco, y la temperatura de dicho punto se eleva por encima de un nivel de temperatura denominado punto de Curie (aproximadamente 200°C). Una vez caliente, las partículas magnéticas en dicho punto pueden fácilmente ver cambiada su dirección (polarización) debido al efecto del campo magnético generado por el cabezal de escritura. La información se lee del disco mediante un láser de menor potencia que el que se emplea para escribir, utilizando el efecto de Kerr, según el cual la polaridad de la luz del láser reflejado se altera dependiendo de la orientación de las partículas magnéticas. Los puntos con cierta polaridad representan los ceros y los pun-



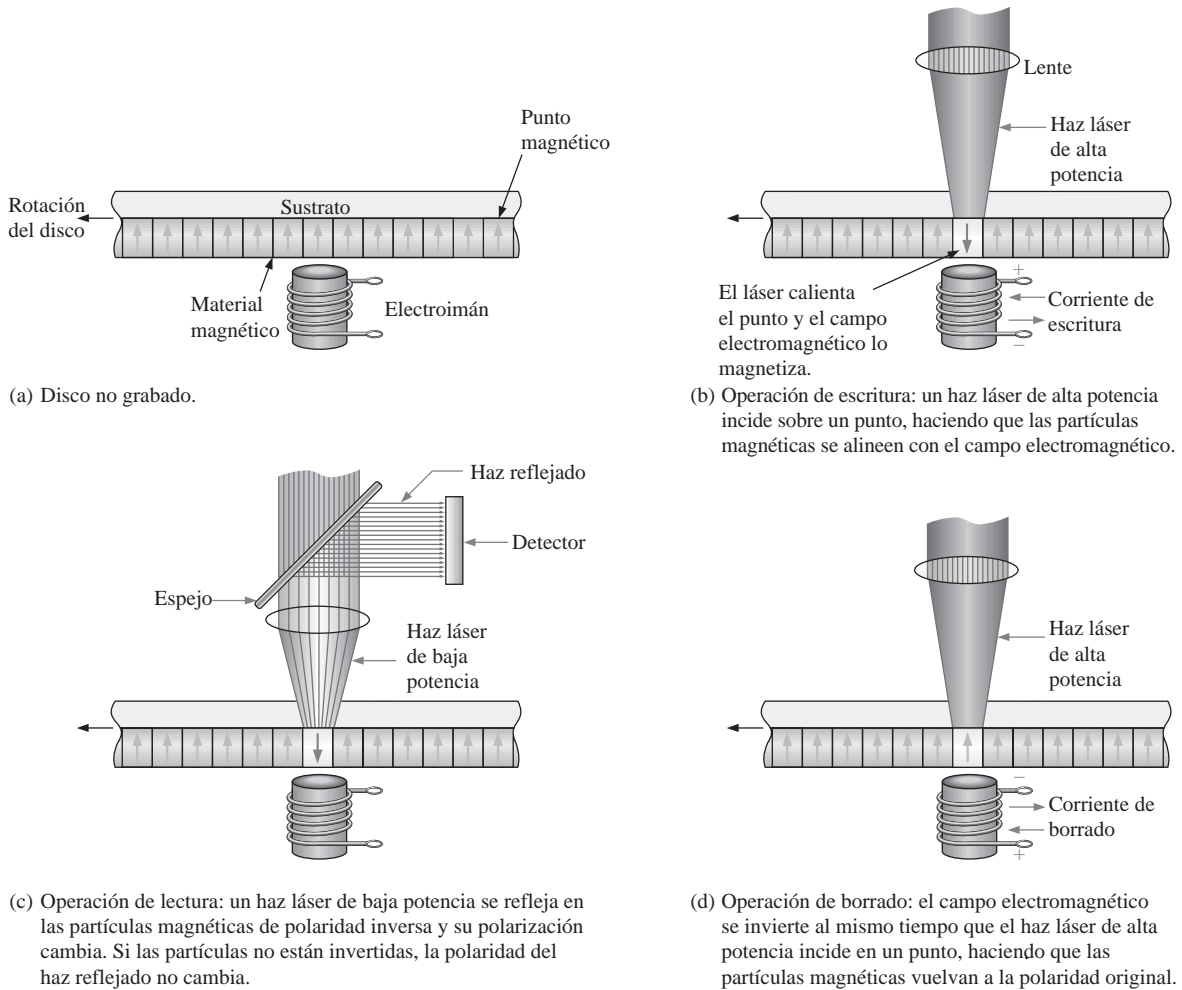


FIGURA 10.63 Principios básicos de un disco magneto-óptico.

tos con la polaridad opuesta representan los unos. El funcionamiento básico magneto-óptico se muestra en la Figura 10.63, que representa una pequeña sección transversal de un disco.

## Almacenamiento óptico

**CD-ROM.** La memoria de sólo lectura de disco compacto (*compact-disc read-only memory*) es un disco de 120 mm de diámetro con tres capas dispuestas en forma de sandwich: la capa inferior de plástico de policarbonato, una hoja delgada de aluminio para la reflectividad y una capa superior de laca para protección. El disco **CD-ROM** se formatea con una única pista en forma de espiral, con sectores secuenciales de 2 kB y tiene una capacidad de 680 MB. Los datos se pregrababan en fábrica en forma de agujeros microscópicos denominados *muestras* y el área plana que rodea a estos agujeros se denomina *planicie*. Las muestras se imprimen en la capa de plástico y no pueden borrarse.

Un reproductor de CD-ROM lee los datos en la pista espiral mediante un haz láser de infrarrojos de baja potencia, como se muestra en la Figura 10.64. Los datos están codificados mediante las muestras y planicies,

como muestra la figura. La luz del láser reflejada desde una muesca tiene un desfase de  $180^\circ$  respecto de la luz reflejada desde las planicies. Cuando el disco gira, el estrecho haz de láser incide sobre las muescas y planicies de longitudes variables, y un fotodiodo detecta la diferencia en la luz reflejada. El resultado es una serie de unos y ceros, que corresponde a la configuración de las muescas y de las planicies a lo largo de la pista.

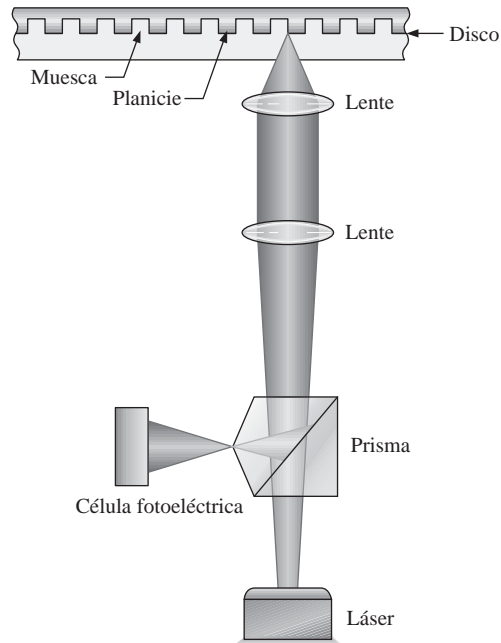


FIGURA 10.64 Operación básica de lectura de datos en un CD-ROM.

**WORM.** Los discos de única escritura- múltiples lecturas (*write once -read many*) son un tipo de dispositivo de almacenamiento óptico en los que se puede escribir una sola vez, después de lo cual no se pueden borrar los datos, aunque sí se pueden leer muchas veces. Para escribir los datos se utiliza un haz láser de baja potencia, que crea agujeros microscópicos en la superficie del disco. Los unos y los ceros quedan representados por las áreas en las que hay y no hay agujero.

**CD-R.** Prácticamente, es un tipo de WORM. La diferencia se encuentra en que el CD grabable permite múltiples sesiones de escritura en diferentes áreas del disco. El disco CD-R tiene una pista en forma de espiral como el CD-ROM, pero, en lugar de hacerse agujeros mecánicamente sobre el disco para representar los datos, el CD-R emplea un láser para quemar agujeros microscópicos en una superficie con tinte orgánico. Cuando se calienta más allá de una temperatura crítica con un haz láser durante la operación de lectura, los puntos quemados cambian de color y reflejan menos luz que las áreas no quemadas. Por tanto, los unos y los ceros se representan en un CD-R mediante las áreas quemadas y no quemadas, mientras que un CD-ROM se representan mediante las muescas y las planicies. Al igual que con el CD-ROM, los datos no pueden borrarse una vez que se han escrito.

**CD-RW.** El disco CD regrabable puede utilizarse para leer y escribir datos. En lugar de la capa de grabación con tinte que se emplea en el CD-R, el CD-RW normalmente utiliza un compuesto cristalino con una propiedad especial. Cuando se calienta a una cierta temperatura, al enfriarse se vuelve cristalino, pero si se calienta a una temperatura superior, se funde y se vuelve amorfo al enfriarse. Para escribir datos, el haz láser enfocado calienta el material a la temperatura de fundido dando lugar al estado amorfo. Las áreas amorfas resultan-

tes reflejan menos la luz que las áreas cristalinas, permitiendo que la operación de lectura detecte los unos y los ceros. Los datos se pueden borrar o sobrescribir calentando las áreas amorfas a una temperatura superior a la temperatura de cristalización, pero inferior a la temperatura de fusión, lo que hace que el material amorfo pase de nuevo al estado cristalino.

**DVD-ROM.** Originalmente, DVD eran las siglas correspondientes a Digital Video Disk (videodisco digital) pero, actualmente, corresponden a *Digital Versatile Disk*. Al igual que en el CD-ROM, en el **DVD-ROM** los datos se pregrababan en el disco. Sin embargo, el tamaño de las muescas es menor que en el CD-ROM, lo que permite almacenar más datos en una pista. La diferencia principal entre el CD-ROM y el DVD-ROM es que el CD-ROM tiene una única cara mientras que el DVD almacena datos por las dos caras. También, además de los discos DVD de dos caras, hay disponibles discos de múltiples capas que utilizan capas de datos semitransparentes colocadas sobre las capas de datos principales, proporcionando capacidades de almacenamiento de decenas de gigabytes. Para acceder a todas las capas, hay que cambiar el enfoque del haz láser para pasar de una capa a otra.

### REVISIÓN DE LA SECCIÓN 10.8

1. Enumerar los principales tipos de dispositivos de almacenamiento magnético.
2. ¿Cuál es actualmente la capacidad de almacenamiento de los discos flexibles?
3. Generalmente, ¿cómo está organizado un disco magnético?
4. ¿Cómo se escriben y se leen los datos en un disco magneto-óptico?
5. Enumerar los tipos de almacenamiento óptico.

## 10.9 LOCALIZACIÓN DE AVERÍAS

---

Ya que las memorias pueden contener una gran cantidad de celdas de almacenamiento, comprobar cada una de ellas puede ser un proceso muy largo y frustrante. Afortunadamente, las memorias usualmente se prueban mediante un procedimiento automático realizado con un equipo de pruebas programable, o con la ayuda de un software para comprobación interna del sistema. La mayoría de los sistemas basados en microprocesador proporcionan la comprobación automática de memoria como parte de su software de sistema.

Al finalizar esta sección, el lector deberá ser capaz de:

- Explicar el método de suma de comprobación utilizado para verificar memorias ROM.
- Explicar el método del patrón ajedrezado utilizado para probar memorias RAM.

### Comprobación de una ROM

Puesto que las ROM contienen datos conocidos, se puede comprobar la corrección de los datos almacenados leyendo cada palabra de datos de la memoria, y comparándola con la palabra de datos que se sabe que es correcta. En la Figura 10.65 se ilustra una forma de hacer esto. Este procedimiento requiere una ROM de referencia que contenga los mismos datos que la ROM que se va a comprobar. Un equipo de pruebas especial se programa para leer cada dirección de ambas memorias ROM simultáneamente y comparar los contenidos. El organigrama de la Figura 10.66 presenta la secuencia básica.

**Método de la suma de comprobación.** Aunque el método anterior comprueba cada dirección de la ROM para asegurar la corrección de los datos, tiene la desventaja de que se requiere una ROM de referencia para cada ROM diferente que se desee probar. También puede ocurrir que la ROM de referencia falle, dando lugar a una indicación falsa de error.

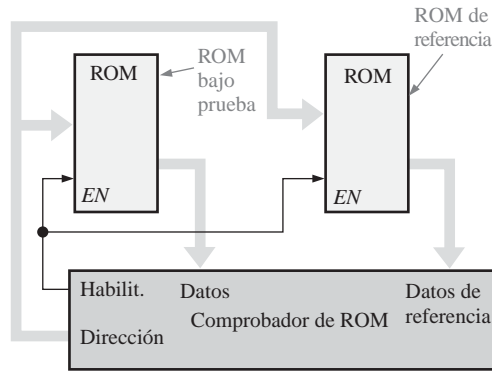
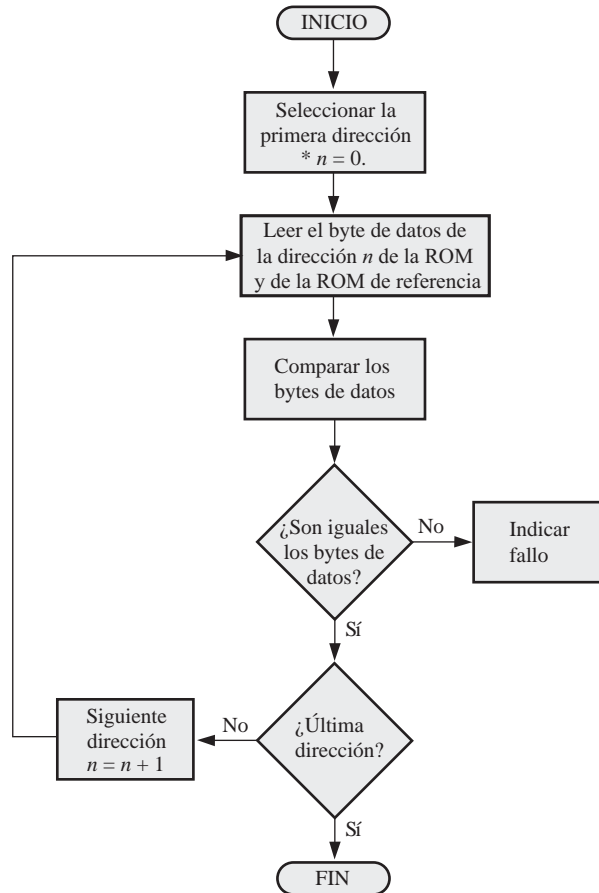


FIGURA 10.65 Diagrama de bloques para la comprobación de todo el contenido de una ROM.



\*  $n$  es el número de dirección.

FIGURA 10.66 Organigrama para la comprobación de todo el contenido de una ROM.

En el método de suma de comprobación, un número, que es la suma de todos los contenidos de todas las direcciones de la ROM, se almacena en una dirección específica de la memoria cuando ésta se programa. Para

probar la ROM, se suman los contenidos de todas las direcciones, excepto la propia suma de verificación, y el resultado se compara con la suma de verificación almacenada en la ROM. Si existe diferencia, es seguro que hay un fallo. Si las sumas de verificación son iguales, muy probablemente la ROM está bien. Sin embargo, existe la remota posibilidad de que una combinación de celdas de memoria erróneas haga que las sumas de verificación sean iguales.

En la Figura 10.67 se ilustra este procedimiento mediante un sencillo ejemplo. En este caso, la suma de comprobación se genera sumando cada columna de bits de datos y descartando los acarrees. Esto equivale a realizar la operación XOR en cada columna. El organigrama de la Figura 10.68 presenta el método de prueba básico mediante la suma de comprobación.

Este método de prueba puede implementarse con un equipo de pruebas especial, o puede incorporarse en una rutina de prueba dentro del software del sistema, en los sistemas basados en microprocesador. En este caso, la rutina de prueba de la ROM se ejecuta automáticamente al arrancar el sistema.

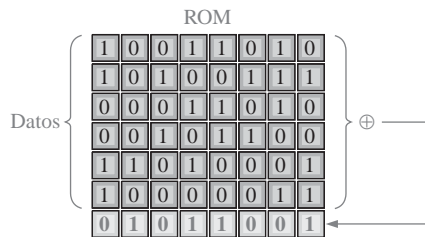


FIGURA 10.67 Ilustración simplificada de una ROM programada, con una suma de comprobación almacenada en una dirección específica.

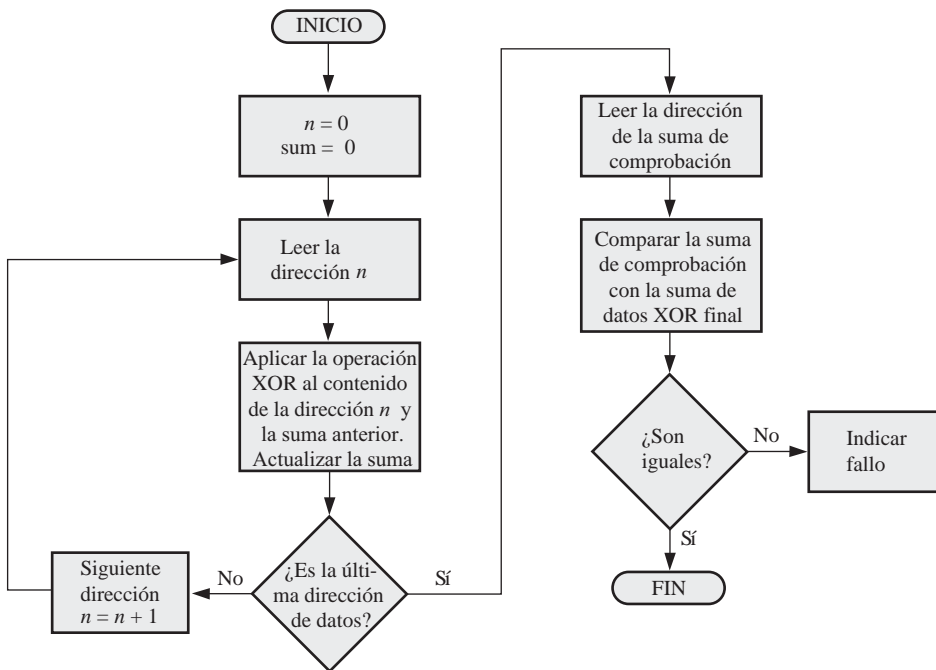


FIGURA 10.68 Organigrama para el método básico de prueba mediante la suma de comprobación.

## Comprobación de la RAM

Para probar la capacidad de las memorias RAM de almacenar 0s y 1s en sus celdas se escriben, en primer lugar 0s en todas las celdas de cada dirección, y luego se extraen y verifican. A continuación, se escriben 1s en todas las celdas de cada dirección, y después se leen y verifican. Este método de pruebas básico detectará una celda que se mantenga en el estado 1 o en el estado 0.

Algunos fallos de memoria no se pueden detectar escribiendo ceros en todas las direcciones y unos en todas las direcciones. Por ejemplo, si dos celdas de memoria adyacentes se cortocircuitan, siempre estarán en el mismo estado, siendo ambas 0, o ambas 1. También este método de prueba es inefectivo si existen problemas de ruido interno, consistentes en que los contenidos de una o más direcciones se alteran debido al cambio de los contenidos de otras direcciones.

**Método de prueba con patrón ajedrezado.** Una forma más completa para probar una RAM consiste en utilizar un patrón de 1s y 0s alternativos, como ilustra la Figura 10.69. Observe que todas las celdas adyacentes contienen bits opuestos. Este patrón comprueba que no haya un cortocircuito entre dos celdas adyacentes, ya que, si existe un cortocircuito, ambas células estarán en el mismo estado.

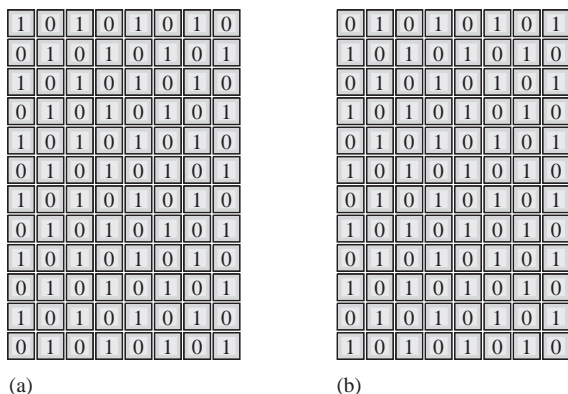


FIGURA 10.69 Patrón de prueba ajedrezado de una RAM.

Después de comprobar la RAM con el patrón de la Figura 10.69(a), éste se invierte, como se muestra en la parte (b) de la figura. Esta inversión verifica la capacidad de todas las celdas de almacenar tanto un 1 como un 0.

Una comprobación más consiste en alternar el patrón en una dirección de memoria cada vez y comprobar que el patrón original no se haya modificado en las restantes direcciones. Esta prueba detectará los problemas consistentes en que los contenidos de una dirección se alteren dinámicamente cuando los contenidos de otra dirección cambien.

En el organigrama de la Figura 10.70 se presenta un procedimiento básico para la prueba del patrón ajedrezado. El procedimiento se puede implementar con el software del sistema en los sistemas basados en microprocesador, de modo que las comprobaciones se realizan automáticamente cuando se enciende el sistema o se pueden iniciar desde el teclado.

### REVISIÓN DE LA SECCIÓN 10.9

1. Describir el método de suma de comprobación para probar una ROM.
2. ¿Por qué no se puede aplicar el método de suma de comprobación para probar una RAM?
3. Enumerar los tres fallos básicos que se pueden detectar en una RAM con la prueba del patrón ajedrezado.

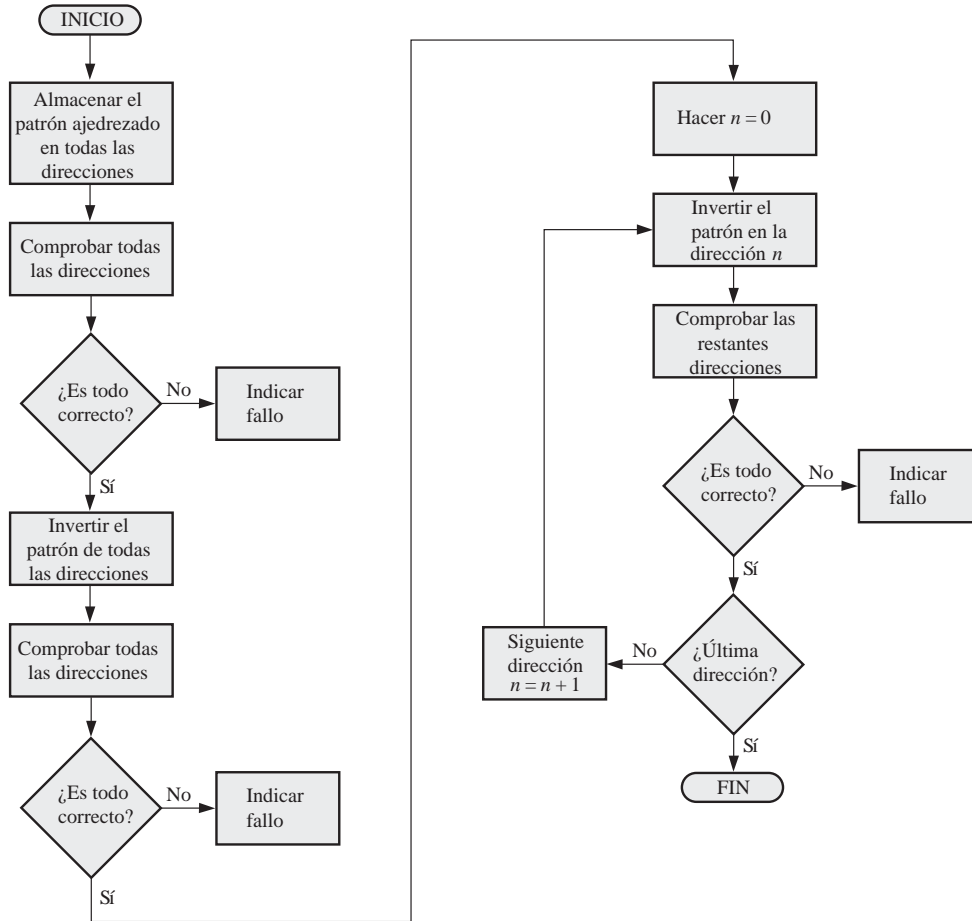


FIGURA 10.70 Organigrama para el método básico de prueba del patrón ajedrezado.



## APLICACIÓN A LOS SISTEMAS DIGITALES

En esta aplicación se desarrolla la lógica de memoria para el sistema de seguridad presentado en el Capítulo 9. En el

capítulo anterior se ha completado la lógica para el código de seguridad, la cual se combinará con la lógica de memoria para formar el sistema completo.

### Funcionamiento general

En la Figura 10.71 se presenta un diagrama de bloques básico del sistema completo de seguridad de acceso.

La lógica de memoria almacena un código de acceso de 4 dígitos en formato BCD. En el modo de desactivación (*Desarm*), se introducen en la memoria cuatro dígitos a través del teclado. Una vez almacenados en la memoria, los cuatro dígitos BCD pasan a ser el código de seguridad permanente que hay que introducir. Si fuera necesario cambiar el código de seguridad, la memoria se reprogramaría con uno nuevo.

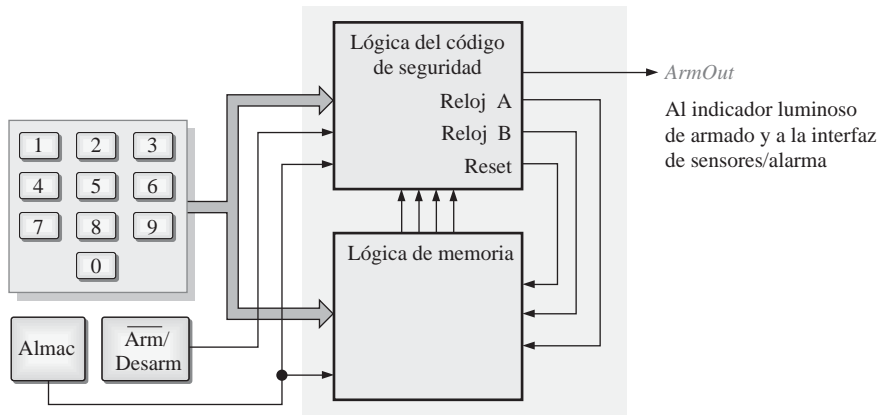


FIGURA 10.71 Diagrama de bloques básico del sistema de seguridad de acceso.

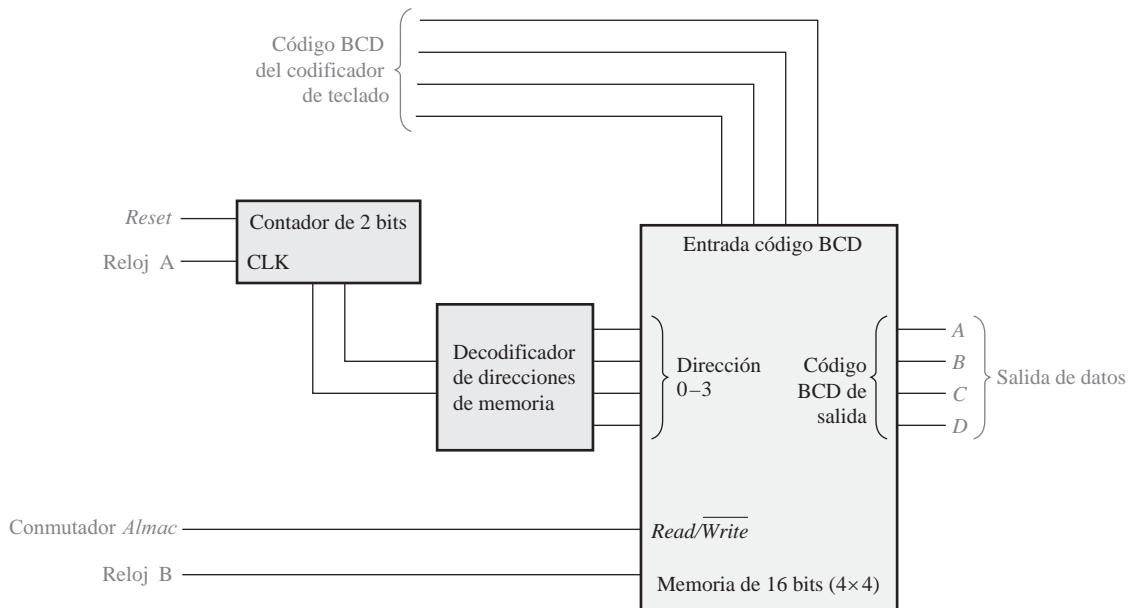


FIGURA 10.72 Diagrama de bloques de la lógica de memoria.

La memoria se programa poniendo en primer lugar el sistema en modo de desactivación (*desarm*) y utilizando el conmutador *Almacenar* y el teclado para introducir el código de cuatro dígitos deseado. Esto es una operación de *escritura* en memoria. Una vez que se ha programado la memoria con el código de seguridad, el conmutador *Arm/Desarm* se cambia al modo *arm* (activación), que prepara a la memoria para realizar operaciones de *lectura*. En la Figura 10.72 se muestra un diagrama de bloques de la lógica de memoria.

## La celda de memoria

La memoria requiere 16 celdas para almacenar los cuatro dígitos BCD del código de seguridad. En la Figura 10.73 se muestra un posible diseño de una celda de memoria. Se utiliza un flip-flop J-K como dispositivo elemento básico de almacenamiento; dicho flip-flop puede operar en dos modos (*lectura* y *escritura*). En el modo de *escritura*, *SelDir* (selección de dirección) está a nivel ALTO y la entrada  $R/\bar{W}$  (*read/write*, *lectura/escritura*) está a nivel BAJO. Las puertas  $G1$  y  $G2$  están habilitadas, el bit de entrada se aplica a la entrada  $J$  y su complemento se apli-



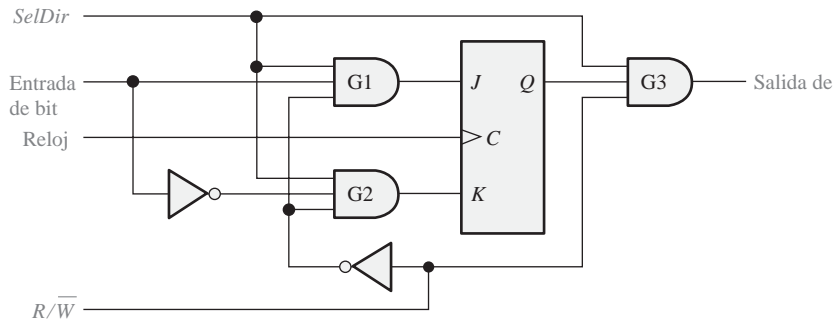


FIGURA 10.73 Lógica de la celda de memoria.

ca a la entrada  $K$ . El bit de entrada se almacena entonces con el flanco positivo del impulso de reloj. En el modo de lectura,  $SelDir$  está a nivel ALTO y  $R/\bar{W}$  está también a nivel ALTO, habilitando  $G3$ . El bit almacenado en la salida  $Q$  del flip-flop aparece en la salida de  $G3$  (Salida de bit).

## El decodificador de direcciones de memoria

En la Figura 10.74 se presenta la lógica del decodificador de direcciones de memoria. Se aplica una secuencia binaria de 2 bits a las entradas de selección ( $S_0, S_1$ ) para seleccionar cada una de las cuatro direcciones de memoria utilizando las líneas  $SelDir$ .

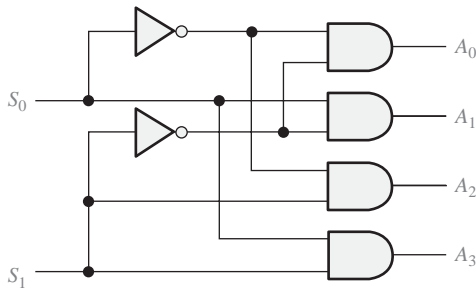


FIGURA 10.74 Decodificador de direcciones de memoria.

## La matriz de memoria

La memoria tiene dieciseis celdas como se muestra en la Figura 10.75. Cuando el decodificador de direcciones selecciona una de las filas de memoria y la entrada  $read/write$  está a nivel BAJO, el código de entrada BCD de 4 bits se almacena a impulsos del reloj en las cuatro celdas seleccionadas. Las entradas al decodificador de direc-

ciones pasan secuencialmente a través de cada uno de los cuatro estados (00, 01, 10 y 11) para seleccionar sucesivamente cada fila de la memoria.

La Figura 10.76 ilustra la programación de la memoria a medida que se introduce secuencialmente el código de seguridad 4739.

## Lógica de memoria completa

Es necesario un codificador de teclado para convertir una pulsación de tecla en un código BCD, y se utiliza un contador de 2 bits para producir la secuencia de selección de las direcciones de memoria. Esto se muestra en la Figura 10.77. Al comienzo de la programación, se reinicializa el contador al estado 0 mediante una entrada de reinicialización procedente de la lógica de introducción del código, y el contador avanza a través de la secuencia de introducción de cada dígito.

## Sistema de seguridad completo

Ahora que hemos completado la lógica de memoria, podemos combinarla con la lógica del código de seguridad del Capítulo 9 como muestra el diagrama de bloques de la Figura 10.78, para formar el sistema de seguridad completo mostrado como diagrama de bloques en la Figura 10.79.

## Práctica de sistemas

- **Actividad 1.** Explicar qué función tiene el codificador de teclado en la lógica de memoria.
- **Actividad 2.** Explicar qué función tiene el contador de 2 bits en la lógica de memoria.
- **Actividad opcional.** Construir el sistema de seguridad de acceso completo usando dispositivos estándar 74XX y otros componentes que sean necesarios. Probar el sistema.

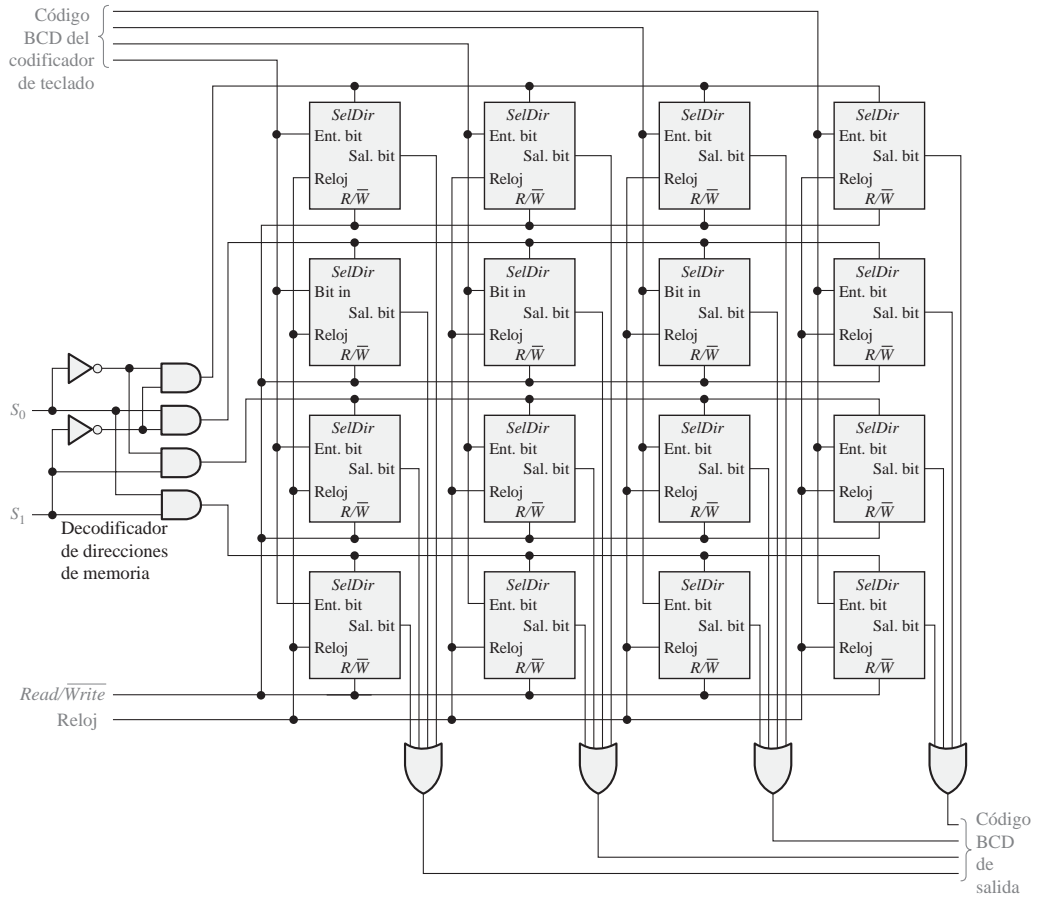
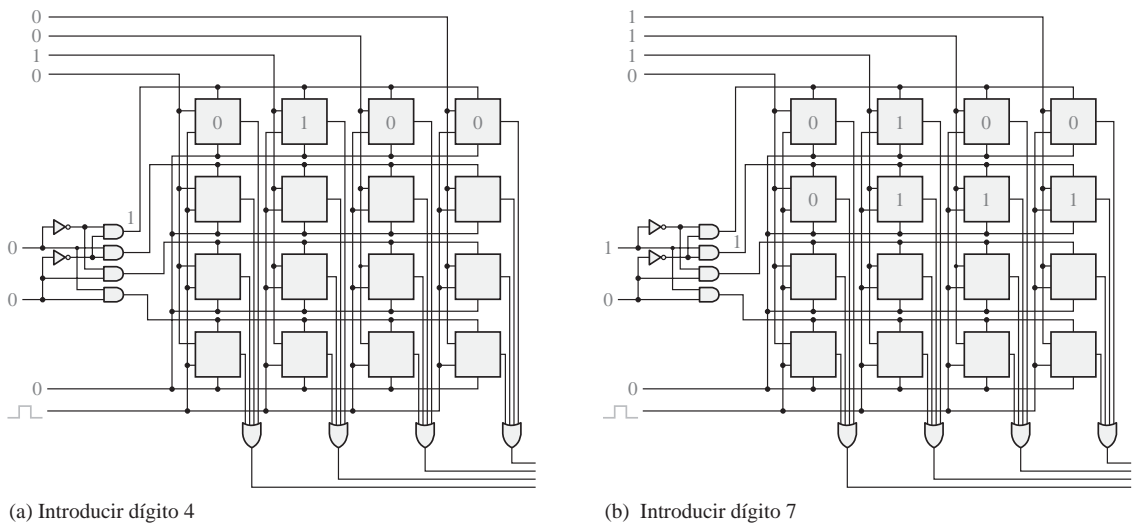


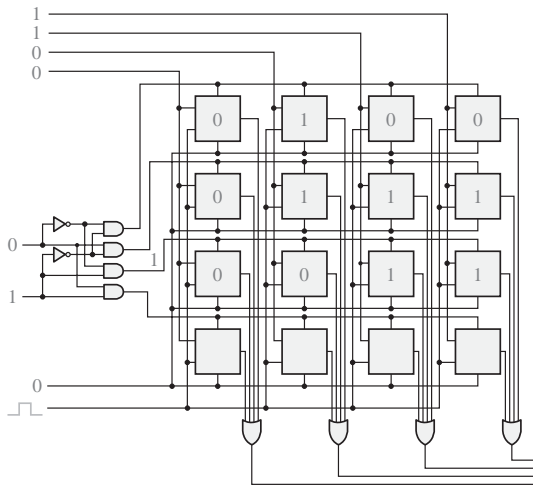
FIGURA 10.75 Matriz de memoria y decodificador de direcciones.



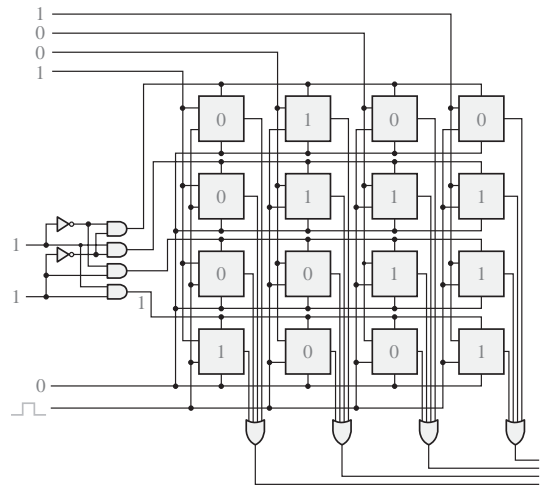
(a) Introducir dígito 4

(b) Introducir dígito 7

FIGURA 10.76 Ilustración de cómo se introduce en memoria un código de seguridad (4739). (Continúa)

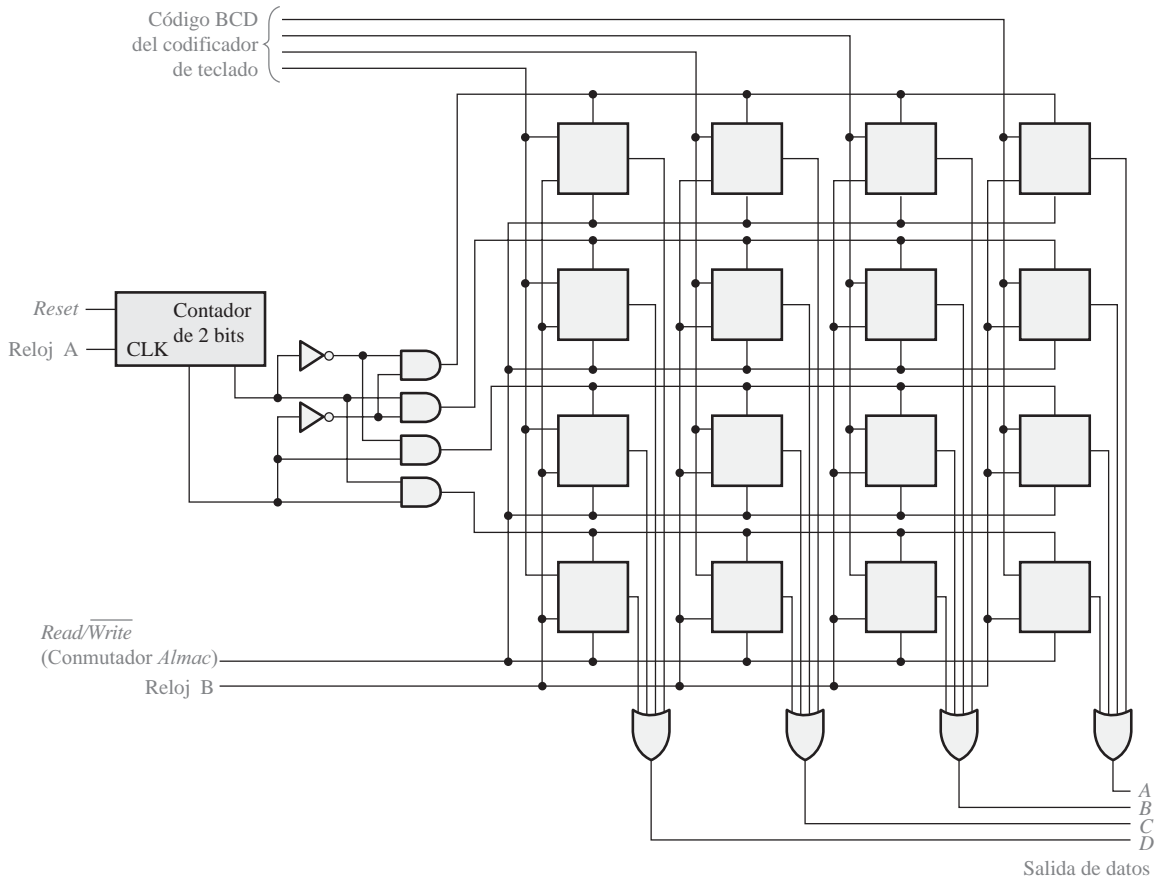


(c) Introducir dígito 3



(d) Introducir dígito 9

**FIGURA 10.76** Ilustración de cómo se introduce en memoria un código de seguridad (4739). (Continuación)



**FIGURA 10.77** Lógica de memoria completa.

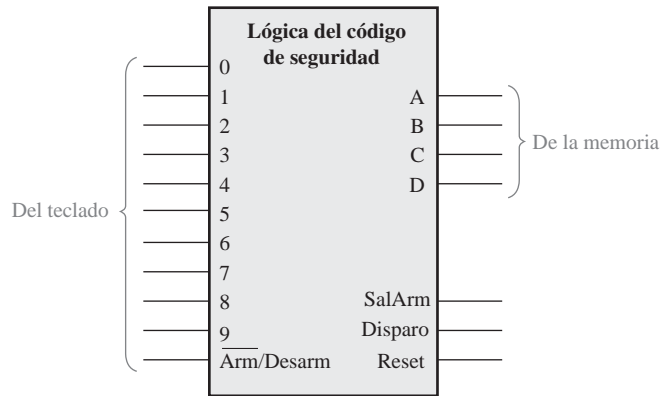


FIGURA 10.78 Lógica del código de seguridad (del Capítulo 9).

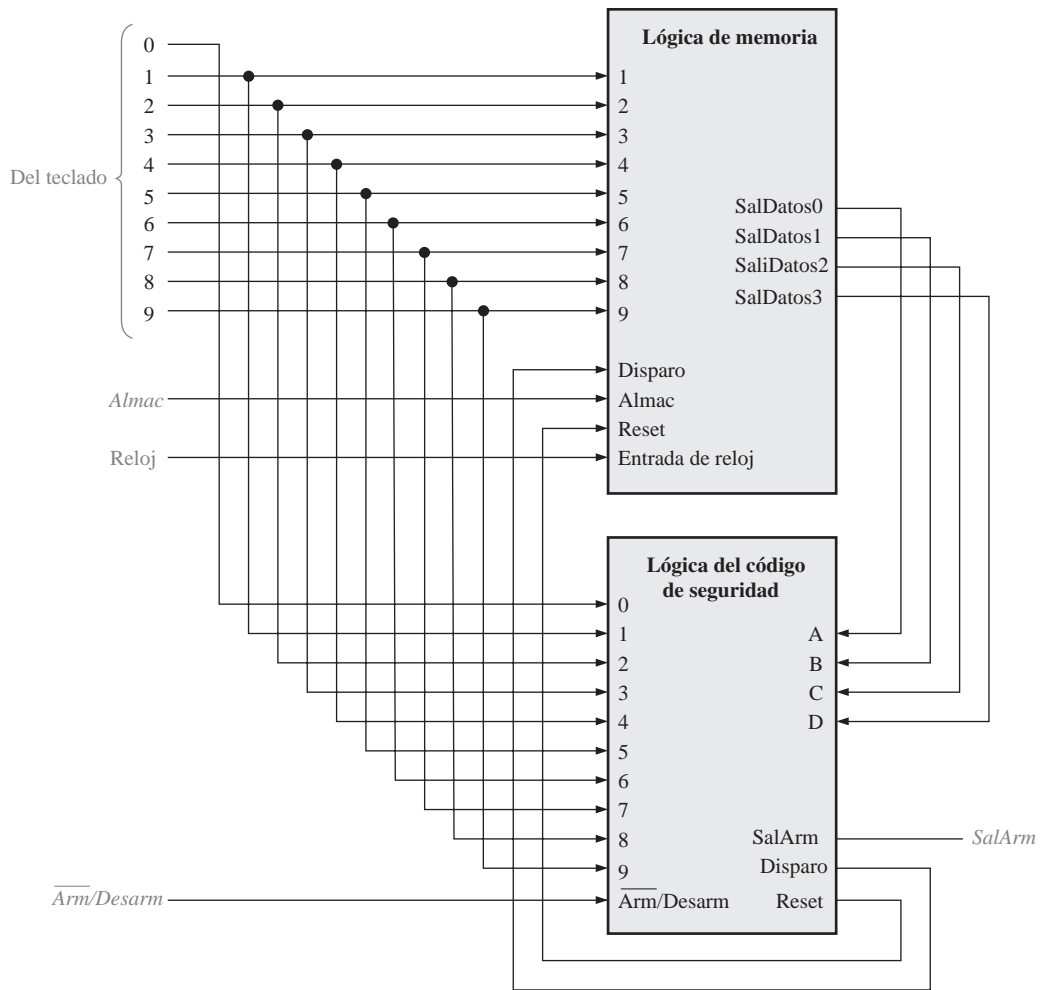
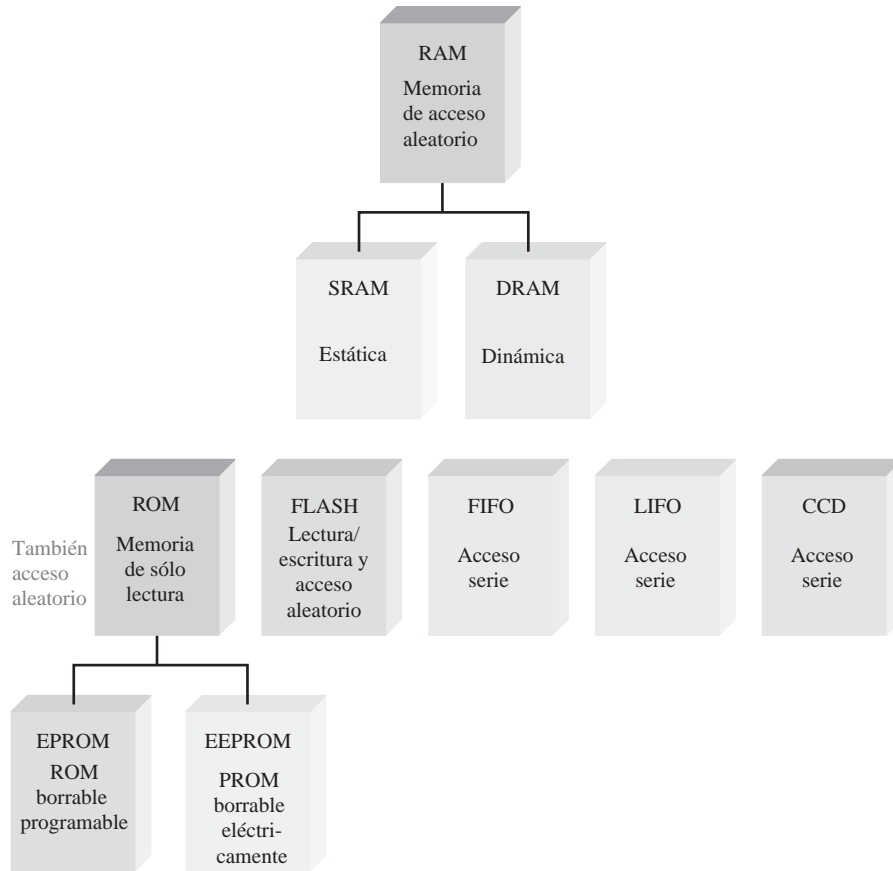


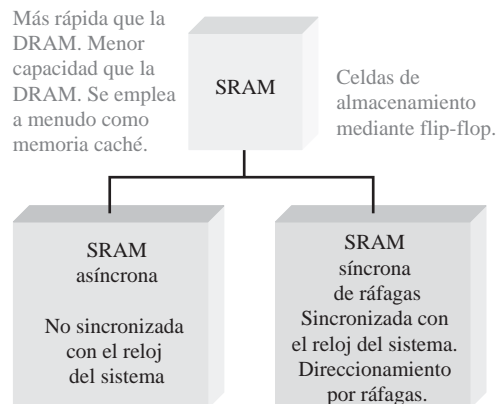
FIGURA 10.79 El sistema de seguridad completo.

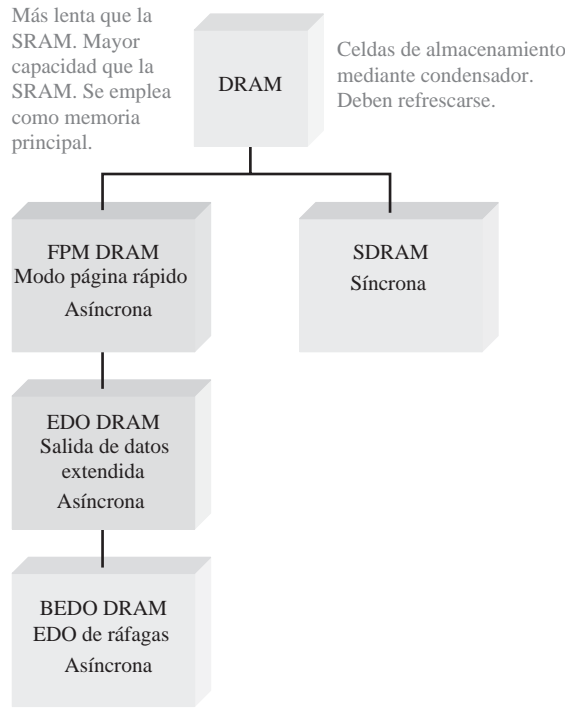
## RESUMEN

## ■ Tipos de memorias semiconductoras:

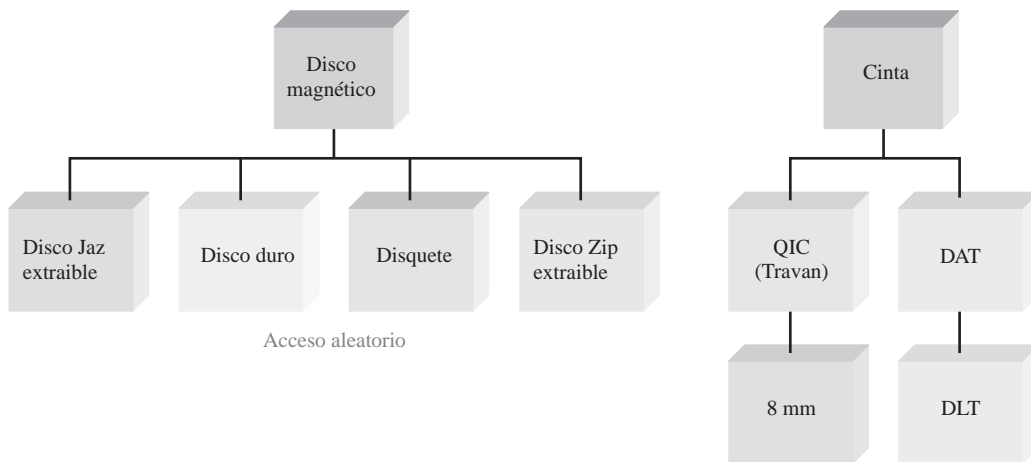


## ■ Tipos de memorias SRAM (RAM estática) y DRAM (RAM dinámica):

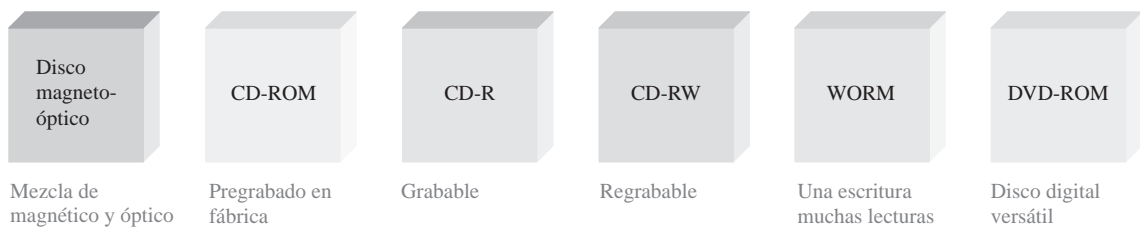




■ Tipos de dispositivos de almacenamiento magnético



■ Tipos de dispositivos de almacenamiento óptico (láser):



**PALABRAS  
CLAVE**

*Las palabras clave y otros términos que se han resaltado en negrita se encuentran en el glosario final del libro.*

**Bus** Conjunto de interconexiones que establece la interfaz entre uno o más dispositivos basándose en una especificación estandarizada.

**Byte** Grupo de ocho bits.

**Capacidad** Número total de unidades de datos (bits, nibbles, bytes, palabras) que puede almacenar una memoria.

**Celda** Un elemento de almacenamiento en una memoria.

**Dirección** Posición de una determinada celda de almacenamiento o grupo de celdas en memoria.

**Disco duro** Dispositivo de almacenamiento magnético; normalmente, una pila de dos o más discos rígidos encerrados en un compartimento sellado.

**DRAM** *Dynamic Random-Access Memory*, memoria dinámica de acceso aleatorio. Un tipo de memoria semiconductora que utiliza condensadores como elemento de almacenamiento y es una memoria de lectura/escritura volátil.

**EPROM** *Electrically Erasable Programmable Read-Only Memory*, un tipo de memoria semiconductora que normalmente utiliza luz ultravioleta para borrar los datos.

**Escritura** El proceso de almacenar datos en memoria.

**FIFO** *First In-First Out*, primero en entrar-primero en salir.

**Lectura** El proceso de recuperar datos de una memoria.

**LIFO** *Last In-First Out*, memoria de tipo primero en entrar-último en salir. Pila de memoria.

**Memoria flash** Memoria semiconductora de acceso aleatorio de lectura/escritura no volátil, en la que los datos se almacenan como carga en la puerta flotante de un determinado tipo de FET.

**Palabra** Unidad completa de datos binarios.

**PROM** *Programmable Read-Only Memory* (memoria de sólo lectura programable); un tipo de memoria semiconductora.

**RAM** *Random-Access Memory*, memoria de acceso aleatorio. Memorias semiconductoras volátiles de lectura/escritura.

**ROM** *Read-Only Memory*, memoria semiconductora no volátil de acceso aleatorio.

**SRAM** *Static Random Access Memory*, memoria estática de acceso aleatorio; un tipo de memoria volátil semiconductora de lectura/escritura.

**AUTOTEST**

*Las respuestas se encuentran al final del capítulo.*

- La capacidad de bits de una memoria que tiene 1024 direcciones y que puede almacenar 8 bits en cada dirección es:  
(a) 1024 (b) 8192 (c) 8 (d) 4096
- Una palabra de datos de 32 bits está formada por:  
(a) 2 bytes (b) 4 nibbles (c) 4 bytes (d) 3 bytes y 1 *nibble*
- Los datos en una memoria de acceso aleatorio (RAM) se almacenan durante:  
(a) la operación de lectura.  
(b) la operación de habilitación.  
(c) la operación de escritura.  
(d) la operación de direccionamiento.
- Los datos que se almacenan en una determinada dirección de una memoria de acceso aleatorio (RAM) se pierden cuando:  
(a) se apaga la alimentación.  
(b) se leen los datos de dicha dirección.

- (c) se escriben nuevos datos en dicha dirección.
  - (d) las respuestas (a) y (c).
5. Una ROM es:
- (a) una memoria no volátil.
  - (b) una memoria volátil.
  - (c) una memoria de lectura/escritura.
  - (d) una memoria organizada en bytes.
6. Una memoria con 256 direcciones tiene:
- (a) 256 líneas de dirección.
  - (b) 6 líneas de dirección.
  - (c) 1 línea de dirección.
  - (d) 8 líneas de dirección.
7. Una memoria organizada en bytes tiene:
- (a) 1 línea de salida de datos.
  - (b) 4 líneas de salida de datos.
  - (c) 8 líneas de salida de datos.
  - (d) 16 líneas de salida de datos.
8. La celda de almacenamiento en una SRAM es:
- (a) un flip-flop      (b) un condensador
  - (c) un fusible      (d) un punto magnético
9. Una DRAM debe ser:
- (a) reemplazada periódicamente.
  - (b) refrescada periódicamente.
  - (c) habilitada siempre.
  - (d) programada antes de cada uso.
10. Una memoria flash es:
- (a) volátil
  - (b) una memoria de sólo lectura.
  - (c) una memoria de lectura/escritura.
  - (d) no volátil.
  - (e) las respuestas (a) y (c).
  - (f) las respuestas (c) y (d).
11. Disco duro, disquete, disco Zip y disco Jaz son todos ellos:
- (a) dispositivos de almacenamiento magneto-óptico.
  - (b) dispositivos de almacenamiento semiconductores.
  - (c) dispositivos de almacenamiento magnéticos.
  - (d) dispositivos de almacenamiento ópticos.
12. Los dispositivos de almacenamiento óptico emplean:
- (a) luz ultravioleta.
  - (b) campos electromagnéticos.
  - (c) acopladores ópticos.
  - (d) láseres.

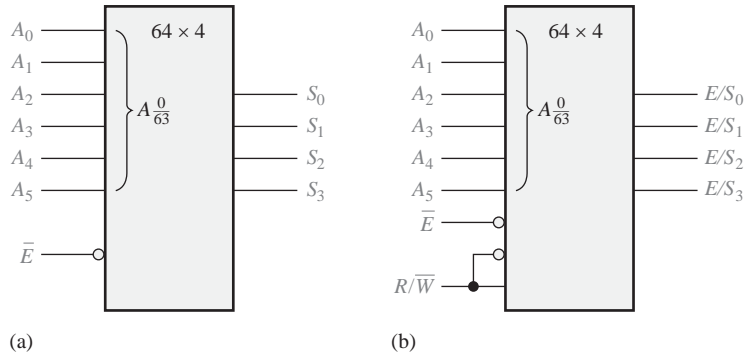


**PROBLEMAS**

Las respuestas a los problemas impares se encuentran al final del libro.

**SECCIÓN 10.1 Principios de las memorias semiconductoras**

1. Identificar la ROM y la RAM de la Figura 10.80.



**FIGURA 10.80**

2. Explicar por qué las ROM y las RAM son memorias de acceso aleatorio.
3. Explicar los propósitos del bus de direcciones y del bus de datos.
4. Cuál es la dirección de memoria (de 0 hasta 256) que representa cada uno de los siguientes números hexadecimales:  
**(a)**  $0A_{16}$     **(b)**  $3F_{16}$     **(c)**  $CD_{16}$

**SECCIÓN 10.2 Memorias de acceso aleatorio (RAM)**

5. En una matriz de memoria estática con cuatro filas similar a la de la Figura 10.9, se almacenan inicialmente todos ceros. ¿Cuál es el contenido después de las siguientes condiciones? Suponer que un 1 selecciona una fila.  
 Fila 0 = 1, Entrada de datos (bit 0) = 1,  
 Fila 1 = 0, Entrada de datos (bit 1) = 1,  
 Fila 2 = 1, Entrada de datos (bit 2) = 1,  
 Fila 3 = 0, Entrada de datos (bit 3) = 0
6. Dibujar un diagrama lógico básico para una RAM estática de  $512 \times 8$  bits, indicando todas las entradas y salidas.
7. Suponiendo que una SRAM de  $64k \times 8$  tiene una estructura similar a la SRAM de la Figura 10.11, determinar el número de filas y de columnas de 8 bits en la matriz de celdas de memoria.
8. Dibujar de nuevo el diagrama de bloques de la Figura 10.11 para una memoria de  $64k \times 8$ .
9. Explicar la diferencia entre una SRAM y una DRAM.
10. ¿Cuál es la capacidad de una DRAM con doce líneas de dirección?

**SECCIÓN 10.3 Memorias de sólo lectura (ROM)**

11. Para la matriz ROM de la Figura 10.81, determinar las salidas para todas las posibles combinaciones de entrada, y resumirlas en forma de tabla (celda en gris claro es 1, celda en gris oscuro es 0).
12. Determinar la tabla de verdad de la ROM de la Figura 10.82.

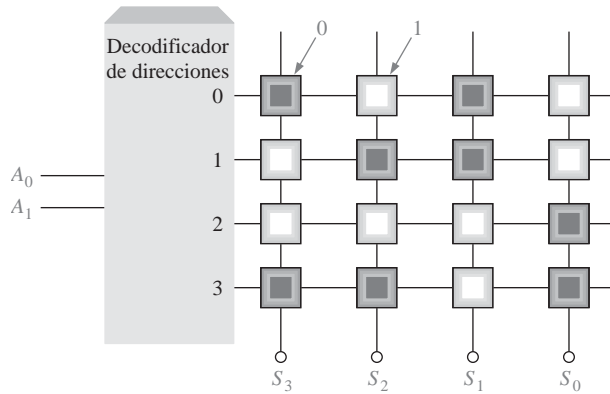


FIGURA 10.81

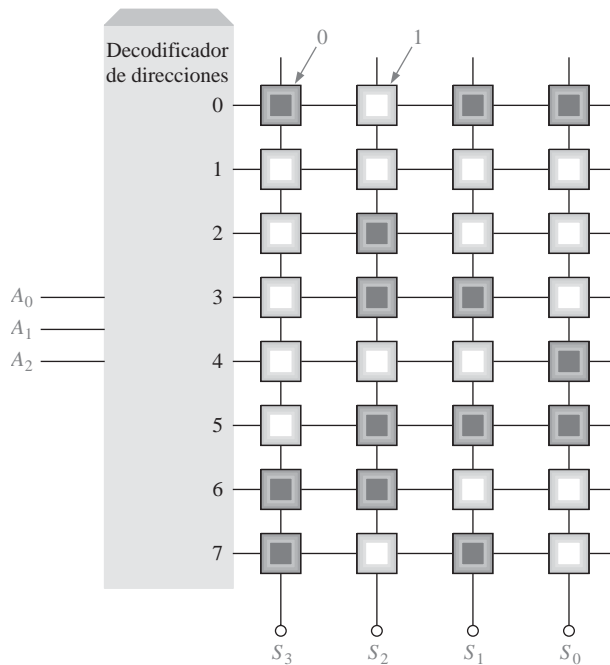


FIGURA 10.82

13. Utilizando un procedimiento similar al del Ejemplo 10.1, diseñar una ROM para convertir un único dígito BCD en código de exceso 3.
14. ¿Cuál es la capacidad total de bits de una ROM que tiene 14 líneas de dirección y 8 salidas de datos?

**SECCIÓN 10.4 Memorias ROM programables (PROM y EPROM)**

15. Suponer que la matriz PROM de la Figura 10.83 se programa fundiendo un hilo fusible para crear un 0. Indicar los hilos que hay que fundir para programar una tabla índice para la operación  $X^3$ , donde  $X$  es un número de 0 a 7.

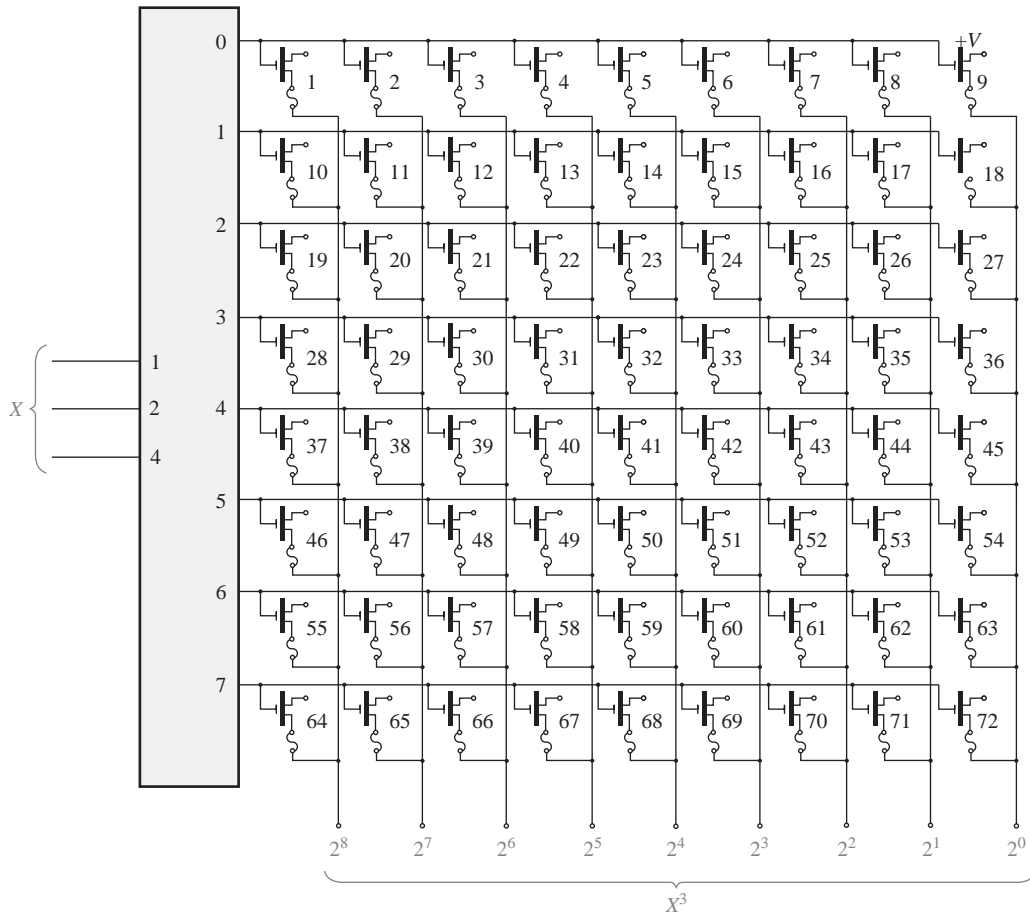


FIGURA 10.83

16. Determinar las direcciones que se programan y los contenidos de cada dirección después de aplicar la secuencia de programación de la Figura 10.84 a una EPROM como la mostrada en la Figura 10.31.

**SECCIÓN 10.6 Expansión de memoria**

- 17. Utilizar memorias DRAM de  $16k \times 4$  para formar una RAM de  $64k \times 8$ . Dibujar el diagrama lógico.
- 18. Utilizando un diagrama de bloques, demostrar cómo se pueden expandir memorias RAM dinámicas de  $64k \times 1$  para formar una RAM de  $256k \times 4$ .
- 19. ¿Cuál es la longitud de palabra y la capacidad de palabra de la memoria del Problema 17? ¿Y del Problema 18?

**SECCIÓN 10.7 Tipos especiales de memorias**

- 20. Completar el diagrama de tiempos de la Figura 10.85, mostrando las formas de onda de salida, que inicialmente están a nivel BAJO, para una memoria serie FIFO como la mostrada en la Figura 10.49.
- 21. Considerar una RAM de  $4096 \times 8$  en la que las 64 últimas direcciones se usan como pila LIFO. Si la primera dirección de la RAM es  $000_{16}$ , indicar las 64 direcciones utilizadas para la pila.

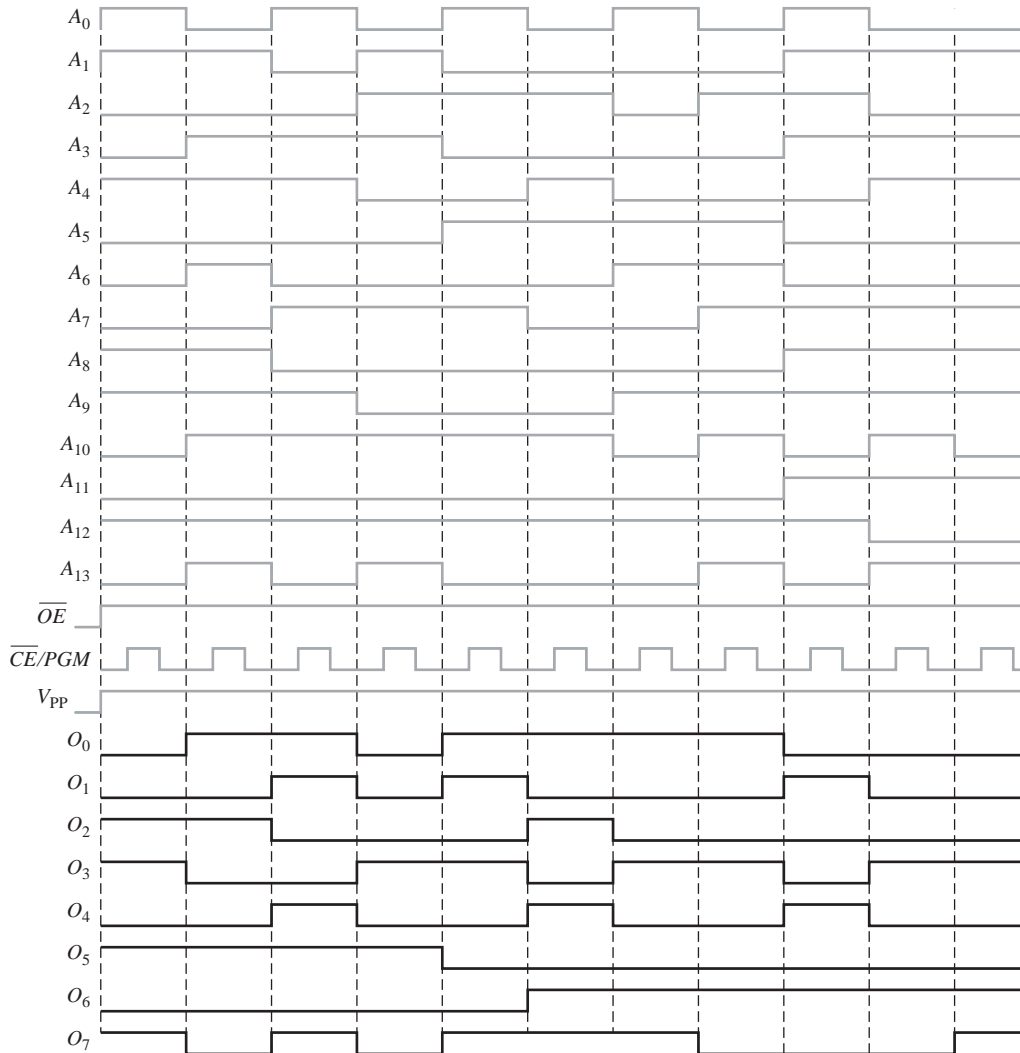


FIGURA 10.84

22. En la memoria del Problema 21, se introducen 16 bytes en la pila. ¿En qué dirección se encuentra el primer byte? ¿Y el último byte?

### SECCIÓN 10.8 Dispositivos de almacenamiento magnético y óptico

23. Describir el formato general de un disco duro.  
 24. Explicar qué es el tiempo de búsqueda y el período de latencia en una unidad de disco duro.  
 25. ¿Por qué la cinta magnética requiere un tiempo de acceso mucho mayor que un disco?  
 26. Explicar las diferencias entre un disco magneto-óptico, un CD-ROM y un WORM.

### SECCIÓN 10.9 Localización de averías

27. Determinar si los contenidos de la ROM de la Figura 10.86 son correctos.  
 28. Una ROM de  $128k \times 8$  se implementa como se indica en la Figura 10.87. El decodificador decodifica los dos bits de dirección más significativos para habilitar una de las ROM cada vez, dependiendo de la dirección seleccionada.

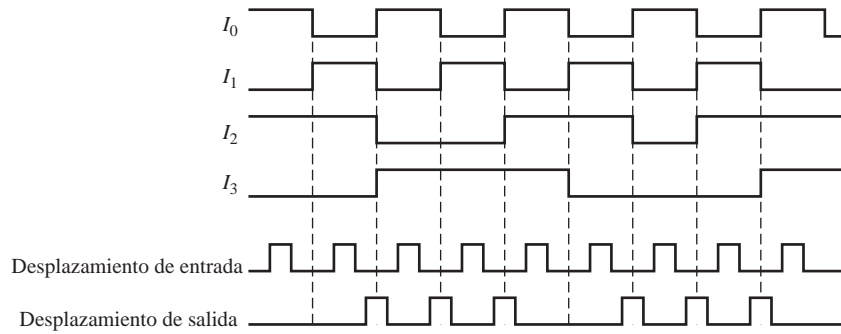


FIGURA 10.85

| ROM                  |           |
|----------------------|-----------|
| 1                    | 0         |
| 1                    | 1         |
| 1                    | 1         |
| 1                    | 1         |
| 1                    | 0         |
| 1                    | 1         |
| 1                    | 0         |
| 1                    | 1         |
| 1                    | 1         |
| 1                    | 0         |
| 0                    | 0         |
| 0                    | 0         |
| 0                    | 0         |
| 0                    | 0         |
| 0                    | 1         |
| Suma de comprobación | 0 1 1 0 0 |

FIGURA 10.86

- (a) Expresar la dirección más baja y la más alta de cada ROM con números hexadecimales.
- (b) Suponer que se usa una única suma de comprobación para la memoria completa y se almacena en la dirección más alta. Desarrollar el organigrama para probar el sistema de memoria completo.
- (c) Suponer que cada ROM tiene una suma de comprobación almacenada en su dirección más alta. Modificar el organigrama desarrollado en el apartado (b) para reflejar este cambio.
- (d) ¿Cuál es la desventaja de utilizar una única suma de comprobación para la memoria completa en lugar de una suma de comprobación para cada ROM individual?

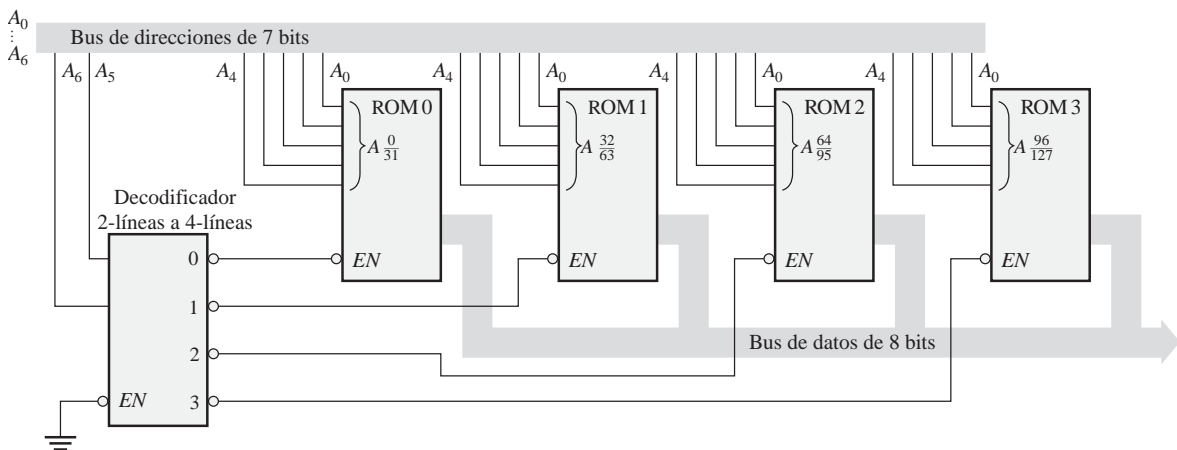


FIGURA 10.87

29. Suponer que se ejecuta una prueba de suma de comprobación en la memoria de la Figura 10.87 y que cada ROM tiene su suma de comprobación en la dirección más alta. ¿Cuál o cuáles circuitos integrados reemplazaría para cada uno de los siguientes mensajes de error que aparecen en el monitor de vídeo del sistema?
- (a) DIRECCIÓN 40 - 5F FALLO
  - (b) DIRECCIÓN 20 - 3F FALLO
  - (c) DIRECCIÓN 00 - 7F FALLO



### Aplicación a los sistemas digitales

30. Desarrollar un cronograma de la lógica de memoria básica de la Figura 10.72, para ilustrar la introducción de los dígitos 4321 en la RAM. Incluir todas las entradas y salidas de los dispositivos.
31. En el modo de programación, ¿cuál es el estado del contador de la Figura 10.77 después de haber introducido dos dígitos del código?
32. ¿Cuál es el propósito de la lógica de memoria?
33. Comentar las ventajas y desventajas de utilizar una PROM externa al CPLD en lugar de la memoria del chip CPLD en la lógica de memoria.



### Problemas especiales de diseño

34. Modificar el diseño de la lógica de memoria del sistema de seguridad de acceso, para acomodar un código de acceso de 5 dígitos.
37. Realizar las modificaciones apropiadas en la lógica de introducción del código en el sistema de seguridad de acceso, para un código de acceso de 5 dígitos. Consulte la sección "Aplicación a los sistemas digitales" del Capítulo 9.

## RESPUESTAS

### SECCIÓN 10 Principios de las memorias semiconductoras

1. El bit es la unidad de datos más pequeña.
2. 2048 bits son 256 bytes
3. Una operación de escritura almacena datos en la memoria.
4. Una operación de lectura realiza una copia de datos de la memoria.
5. Una unidad de datos se localiza mediante una dirección
6. Una RAM es volátil y tiene capacidad de lectura/escritura. Una ROM es no volátil y sólo tiene capacidad de lectura.

### SECCIÓN 10.2 Memorias de acceso aleatorio (RAM)

1. Ráfaga síncrona y asíncrona.
2. Una pequeña memoria rápida entre la UCP y la memoria principal.
3. Las SRAM tienen celdas de almacenamiento *latch* que pueden mantener indefinidamente los datos. Las DRAM tienen celdas de almacenamiento capacitivo que se deben refrescar periódicamente.
4. La operación de refresco evita que los datos se pierdan debido a la descarga del condensador. Cada bit almacenado se restaura periódicamente recargando el condensador a su nivel nominal.

5. FPM, EDO, BEDO, síncrona

### SECCIÓN 10.3 Memorias de sólo lectura (ROM)

1.  $512 \times 8$  es igual a 4096 bits.
2. ROM de máscara, PROM, EPROM, UV EPROM, EEPROM.
3. Se requieren ocho bits de dirección para 256 localizaciones de bytes ( $2^8 = 256$ ).

### SECCIÓN 10.4 Memorias ROM programables (PROM y EPROM)

1. Las PROM son programables eléctricamente, las ROM no.
2. Los bits quedan a 1 después de borrar la EPROM.
3. La lectura es el modo normal de operación de una PROM.

### SECCIÓN 10.5 Memorias flash

1. Flash, ROM, EPROM y EEPROM son no volátiles.
2. La memoria flash es no volátil. Las memorias SRAM y DRAM son volátiles.
3. Programación, lectura, borrado.

### SECCIÓN 10.6 Expansión de memoria

1. Ocho memorias RAM.
2. Ocho memorias RAM.
3. SIMM: módulo de memoria de una única fila de pines.
4. DIMM: módulo de memoria con doble fila de pines.
5. RIMM: *rambus in-line memory module*.

### SECCIÓN 10.7 Tipos especiales de memoria

1. En una memoria FIFO el *primer* bit (o palabra) *en entrar* es el *primer* bit (o palabra) *en salir*.
2. En una memoria LIFO el *último* bit (o palabra) *en entrar* es el *primer* bit (o palabra) *en salir*. Una pila es una LIFO.
3. La operación o instrucción que añade datos a la pila de memoria.
4. La operación o instrucción que elimina datos de la pila de memoria.
5. CCD significa dispositivo de acoplamiento de carga.

### SECCIÓN 10.8 Dispositivos de almacenamiento magnético y óptico

1. Almacenamiento magnético: disquetes, disco duro, cinta y disco magneto-óptico.
2. Capacidad de almacenamiento de los discos flexibles: 1,44 MB.
3. Un disco magnético está organizado en pistas y sectores.
4. Un disco magneto-óptico utiliza un rayo láser y un electroimán.
5. Almacenamiento óptico: CD-ROM, CD-R, CD-RW, DVD-ROM, WORM.

### SECCIÓN 10.9 Localización de averías

1. Los contenidos de la ROM se suman y comparan con una suma de comprobación prealmacenada.
2. La suma de comprobación no se puede utilizar porque los contenidos de una RAM no son fijos.
3. (1) un cortocircuito entre celdas adyacentes. (2) la incapacidad de algunas celdas para almacenar tanto 1s como 0s. (3) la alteración dinámica de los contenidos de una dirección cuando los contenidos de otra varían.

## PROBLEMAS RELACIONADOS

10.1  $G_3G_2G_1G_0 = 1110$

10.2 Deben conectarse ocho memorias ROM de  $64k \times 1$  en paralelo para formar una ROM de  $64k \times 8$ .

10.3 Dieciséis memorias ROM de  $64k \times 1$ .

10.4 Véase la Figura 10.88.

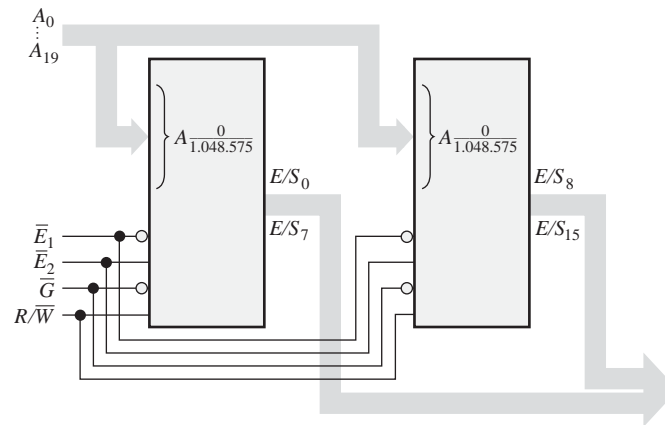


FIGURA 10.88

10.5 ROM 1: 0 a 524.287; ROM 2: 524.288 a 1.048.575

## AUTOTEST

1. (b) 2. (c) 3. (c) 4. (d) 5. (a) 6. (d) 7. (c)  
8. (a) 9. (b) 10. (f) 11. (c) 12. (d)



