

DÉCIMA EDICIÓN

Sistemas digitales

Principios y aplicaciones

Ronald J. Tocci

Monroe Community College

Neal S. Widmer

Purdue University

Gregory L. Moss

Purdue University

Traducción

Alfonso Vidal Romero Elizondo

Ingeniero en Electrónica y Comunicación
Instituto Tecnológico y de Estudios Superiores de
Monterrey - Campus Monterrey

Revisión técnica

Reynaldo Félix Acuña

Profesor investigador
Departamento de Ingeniería Eléctrica
y Electrónica
Instituto Tecnológico y de Estudios
Superiores de Monterrey
Campus Estado de México

Marcos de Alba

Profesor investigador
Departamento de Ingeniería Eléctrica
y Electrónica
Instituto Tecnológico y de Estudios
Superiores de Monterrey
Campus Estado de México

PEARSON
Educación®

México • Argentina • Brasil • Colombia • Costa Rica • Chile • Ecuador
España • Guatemala • Panamá • Perú • Puerto Rico • Uruguay • Venezuela

DISPOSITIVOS DE MEMORIA

■ CONTENIDO

- 12-1 Terminología de memoria
- 12-2 Operación general de la memoria
- 12-3 Conexiones entre CPU y memoria
- 12-4 Memorias de sólo lectura
- 12-5 Arquitectura de la ROM
- 12-6 Sincronización de la ROM
- 12-7 Tipos de ROMs
- 12-8 Memoria Flash
- 12-9 Aplicaciones de la ROM
- 12-10 RAM semiconductora
- 12-11 Arquitectura de la RAM
- 12-12 RAM estática (SRAM)
- 12-13 RAM dinámica (DRAM)
- 12-14 Estructura y operación de la RAM dinámica
- 12-15 Ciclos de lectura/escritura de la DRAM
- 12-16 Refresco de la DRAM
- 12-17 Tecnología de la DRAM
- 12-18 Expansión de tamaño de palabra y capacidad
- 12-19 Funciones especiales de la memoria
- 12-20 Diagnóstico de fallas en sistemas de RAM
- 12-21 Prueba de la ROM

■ OBJETIVOS

Al terminar este capítulo, usted podrá:

- Comprender y utilizar en forma correcta la terminología asociada con los sistemas de memoria.
- Describir la diferencia entre memoria de lectura/escritura y memoria de sólo lectura.
- Hablar sobre la diferencia entre memoria volátil y no volátil.
- Determinar la capacidad de un dispositivo de memoria con base en sus entradas y salidas.
- Describir los pasos que ocurren cuando la CP lee o escribe en memoria.
- Distinguir entre los diversos tipos de ROMs y citar algunas aplicaciones comunes.
- Comprender y describir la organización y operación de las RAMs estáticas y dinámicas.
- Comparar las ventajas y desventajas relativas de la EPROM, EEPROM y la memoria Flash.
- Combinar CIs de memoria para formar módulos de memoria con un tamaño de palabra más grande y/o mayor capacidad.
- Utilizar los resultados de las pruebas en un sistema de RAM o ROM para determinar las posibles fallas en el sistema de memoria.

■ INTRODUCCIÓN

Una de las principales ventajas de los sistemas digitales sobre los analógicos es su habilidad para almacenar con facilidad grandes cantidades de información y datos digitales, durante periodos cortos o largos. Esta capacidad de memoria es lo que hace a los sistemas digitales tan versátiles y adaptables a muchas situaciones. Por ejemplo, en una computadora digital la memoria principal interna almacena instrucciones que indican a la computadora lo que debe hacer bajo *todas* las posibles circunstancias, de manera que la computadora pueda realizar su trabajo con una mínima cantidad de intervención humana.

En este capítulo estudiaremos de los tipos de dispositivos y sistemas de memoria que se utilizan con más frecuencia. Ya nos hemos familiarizado bastante con el flip-flop, que es un dispositivo de memoria electrónico. También hemos visto cómo pueden utilizarse grupos de FFs, a los cuales se les llama *registros*, para almacenar información y cómo esta información puede transferirse hacia otras ubicaciones. Los registros de FF son elementos de memoria de alta velocidad, los cuales se utilizan mucho en las operaciones internas de una computadora digital, en donde la información digital se desplaza continuamente de una ubicación a otra. Los avances en las tecnologías LSI y VLSI han hecho posible colocar grandes

cantidades de FFs en un solo chip, ordenados en diversos formatos de arreglos de memoria. Estas memorias semiconductoras bipolares y MOS son los dispositivos de memoria más veloces disponibles, y su costo se ha ido reduciendo en forma continua, a medida que mejora la tecnología LSI.

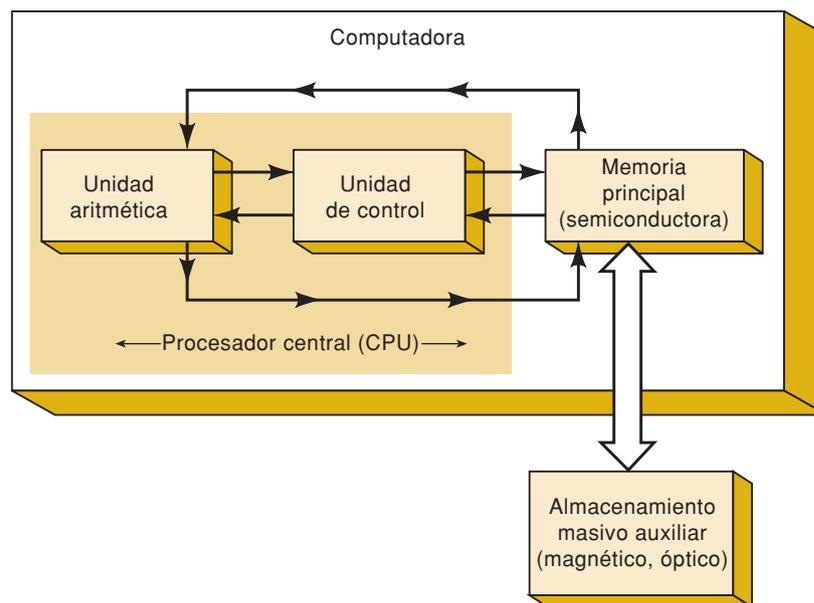
Los datos digitales también pueden almacenarse como cargas en capacitores; y un tipo muy importante de memoria semiconductor utiliza este principio para obtener un almacenamiento de alta densidad con niveles bajos de requerimiento de energía.

Las memorias semiconductoras se utilizan como **memoria principal** de una computadora (figura 12-1), en donde la operación rápida es importante. La memoria principal de una computadora (también conocida como su *memoria de trabajo*) está en comunicación constante con la unidad central de procesamiento (CPU) a medida que se ejecuta un programa de instrucciones. El programa y cualquier información que éste utilice residen en la memoria principal mientras la computadora trabaja con ese programa. La RAM y la ROM (que definiremos en breve) conforman la memoria principal.

La **memoria auxiliar** es otra forma de almacenamiento en la computadora (figura 12-1); esta memoria auxiliar (también conocida como almacenamiento masivo) está separada de la memoria de trabajo principal y tiene la capacidad de almacenar cantidades masivas de datos, sin necesidad de energía eléctrica. La memoria auxiliar opera a una velocidad mucho más lenta que la memoria principal; almacena programas y datos que la CPU no utiliza en ese momento. Esta información se transfiere a la memoria principal cuando la computadora la necesita. Los dispositivos de memoria auxiliar comunes son el disco magnético y el disco compacto (CD).

Veremos con detalle las características de los dispositivos de memoria más comunes que se utilizan como la memoria interna de una computadora. Primero definiremos algunos de los términos comunes que se utilizan en los sistemas de memoria.

FIGURA 12-1 Un sistema computacional utiliza, por lo general, memoria principal de alta velocidad y memoria auxiliar externa más lenta.



12-1 TERMINOLOGÍA DE MEMORIA

El estudio de los dispositivos y sistemas de memoria está lleno de terminología que algunas veces puede ser abrumadora para un estudiante. Antes de ver una descripción detallada sobre las memorias, sería conveniente que entendiera el significado

de algunos de los términos más elementales. El resto de los términos los definiremos a medida que aparezcan en el capítulo.

- **Celda de memoria.** Un dispositivo o un circuito eléctrico utilizado para almacenar un bit (0 o 1). Algunos ejemplos de celdas de memoria son: un flip-flop, un capacitor cargado y un punto individual en cinta o disco magnético.
- **Palabra de memoria.** Un grupo de bits (celdas) en una memoria, el cual representa instrucciones o datos de cierto tipo. Por ejemplo, un registro que consiste de ocho FFs puede considerarse como una memoria que almacena una palabra de ocho bits. Los tamaños de las palabras en las computadoras modernas varían, por lo general, de 8 a 64 bits, dependiendo del tamaño de la computadora.
- **Byte.** Un término especial que se utiliza para un grupo de ocho bits. Un byte siempre consiste de ocho bits. Los tamaños de las palabras pueden expresarse en bytes o en bits. Por ejemplo, una palabra de un tamaño de ocho bits puede expresarse también como una palabra de un byte; una palabra de 16 bits equivale a 2 bytes, y así sucesivamente.
- **Capacidad.** Una manera de especificar cuántos bits pueden almacenarse en un dispositivo de memoria específico o en un sistema completo de memoria. Por ejemplo, suponga que tenemos una memoria que puede almacenar 4096 palabras de 20 bits. Esto representa una capacidad total de 81,920 bits. También podríamos expresar la capacidad de esta memoria como 4096×20 . Cuando se expresa de esta manera, el primer número (4096) es el de palabras y el segundo número (20) es el de bits por palabra (tamaño de palabra). El número de palabras en una memoria es por lo común un múltiplo de 1024. Es común utilizar la designación “1K” para representar $1024 = 2^{10}$ cuando se hace referencia a la capacidad de la memoria. Por lo tanto, una memoria que tiene una capacidad de almacenamiento de $4K \times 20$ es en realidad una memoria de 4096×20 . El desarrollo de memorias más grandes trajo consigo la designación “1M” o “1 meg” para representar $2^{20} = 1,048,576$. Por lo tanto, una memoria con una capacidad de $2M \times 8$ tiene en realidad una capacidad de $2,097,152 \times 8$. La designación “giga” se refiere a $2^{30} = 1,073,741,824$.

EJEMPLO 12-1A

Cierto chip de memoria semiconductor se especifica como de $2K \times 8$. ¿Cuántas palabras pueden almacenarse en este chip? ¿Cuál es el tamaño de palabra? ¿Cuántos bits en total puede almacenar este chip?

Solución

$$2K = 2 \times 1024 = 2048 \text{ palabras}$$

Cada palabra es de ocho bits (un byte). Por lo tanto, el número total de bits es

$$2048 \times 8 = 16,384 \text{ bits}$$

EJEMPLO 12-1B

¿Qué memoria almacena más bits: una de $5M \times 8$ o una que almacena 1M palabras con un tamaño de palabra de 16 bits?

Solución

$$5M \times 8 = 5 \times 1,048,576 \times 8 = 41,943,010 \text{ bits}$$

$$1M \times 16 = 1,048,576 \times 16 = 16,777,216 \text{ bits}$$

La memoria de $5M \times 8$ almacena más bits.

- **Densidad.** Otro término para *capacidad*. Cuando decimos que un dispositivo de memoria tiene una mayor densidad que otra, queremos decir que puede almacenar más bits en la misma área de silicio; es más densa.
- **Dirección.** Un número que identifica la ubicación de una palabra en memoria. Cada palabra que se almacena en un dispositivo o sistema de memoria tiene una dirección única. Las direcciones siempre existen en un sistema digital en forma de un número binario, aunque a menudo se utilizan números octales, hexadecimales y decimales para representar la dirección. La figura 12-2 ilustra una pequeña memoria que consiste de ocho palabras. Cada una de estas ocho palabras tiene una dirección específica, la cual se representa como un número de tres bits que varía de 000 a 111. Cada vez que hacemos referencia a la ubicación de una palabra específica en memoria, utilizamos su código de dirección para identificarla.
- **Operación de lectura.** La operación mediante la cual la palabra binaria almacenada en una ubicación específica de memoria (dirección) se detecta y después se transfiere hacia otro dispositivo. Por ejemplo, si queremos utilizar la palabra 4 de la memoria de la figura 12-2 para cierto propósito, debemos realizar una operación de lectura en la dirección 100. A la operación de lectura, por lo general, se le conoce como operación *buscar (fetch)*, ya que se está buscando una palabra en memoria. Utilizaremos ambos términos de manera indistinta.
- **Operación de escritura.** La operación mediante la cual se coloca una nueva palabra en una ubicación de memoria específica. También se le conoce como operación *almacenar*. Cada vez que se escribe una nueva palabra en una ubicación de memoria, ésta sustituye a la palabra que estaba ahí almacenada.
- **Tiempo de acceso.** Una medida de la velocidad de operación de un dispositivo de memoria. Es la cantidad de tiempo que se requiere para realizar una operación de lectura. Dicho de manera más específica, es el tiempo desde que la memoria recibe una nueva dirección de entrada y hasta que los datos están disponibles en la salida de la memoria. Se utiliza el símbolo t_{ACC} para el tiempo de acceso.
- **Memoria volátil.** Cualquier tipo de memoria que requiere la aplicación de energía eléctrica para poder almacenar información. Si se quita la energía eléctrica, se perderá toda la información almacenada en la memoria. Muchas memorias semiconductoras son volátiles, mientras que todas las memorias magnéticas son *no volátiles*, lo cual significa que pueden almacenar información sin energía eléctrica.
- **Memoria de acceso aleatorio (RAM).** Memoria en la cual la ubicación física de una palabra de memoria no tiene efecto sobre el tiempo que tardan las opera-

FIGURA 12-2 La ubicación de cada palabra tiene una dirección binaria específica.

Direcciones	
000	Palabra 0
001	Palabra 1
010	Palabra 2
011	Palabra 3
100	Palabra 4
101	Palabra 5
110	Palabra 6
111	Palabra 7

ciones de lectura o escritura, en una ubicación determinada. En otras palabras, el tiempo de acceso es igual para cualquier dirección en memoria. La mayoría de las memorias semiconductoras son del tipo RAM.

- **Memoria de acceso secuencial (SAM).** Un tipo de memoria en la cual el tiempo de acceso no es constante, sino que varía dependiendo de la ubicación de la dirección. Una palabra específica que esté almacenada se encuentra mediante la búsqueda secuencial a través de todas las ubicaciones de las direcciones, hasta que se llega a la dirección deseada. Esto produce tiempos de acceso mucho más largos que los de las memorias de acceso aleatorio. Un ejemplo de un dispositivo de memoria de acceso secuencial es el respaldo en cinta magnética. Para ilustrar la diferencia entre SAM y RAM, considere la situación en la que se graban 60 minutos de canciones en un cassette de audio. Cuando deseamos escuchar una canción específica, tenemos que rebobinar o adelantar la cinta hasta encontrarla. El proceso es relativamente lento y la cantidad de tiempo requerido depende del lugar en la cinta en el que esté grabada la canción deseada. Esto es igual que la SAM, ya que hay que avanzar en secuencia a través de toda la información hasta encontrar lo que estamos buscando. La contraparte de la RAM para esto sería un CD de audio, en donde se puede seleccionar con rapidez cualquier canción con sólo introducir el código apropiado, y se requiere aproximadamente el mismo tiempo sin importar cuál canción seleccione. Las memorias de acceso secuencial se utilizan cuando los datos a los que se va a acceder siempre vienen en una larga secuencia de palabras sucesivas. Por ejemplo, la memoria de video debe enviar su contenido en el mismo orden una y otra vez, para regenerar la imagen en la pantalla de CRT.
 - **Memoria de lectura/escritura (RWM).** Cualquier memoria que se puede leer, o en la que se puede escribir, con la misma facilidad.
 - **Memoria de sólo lectura (ROM).** Una amplia variedad de memorias semiconductoras, diseñadas para aplicaciones en las que la proporción de operaciones de lectura en comparación con las de escritura es muy alta. Técnicamente, se puede escribir (programar) en una ROM sólo una vez, y esta operación, por lo general, se realiza en la fábrica. Después de eso, la información sólo se puede leer de la memoria. Otros tipos de ROM son en realidad memorias de casi siempre lectura (RMM), en las cuales se puede escribir más de una vez; pero la operación de escritura es más complicada que la operación de lectura, por lo cual no se realiza con mucha frecuencia. Más adelante hablaremos sobre los diversos tipos de ROM. *Toda la ROM es no volátil* y almacenará los datos aunque se retire la energía eléctrica.
 - **Dispositivos de memoria estática.** Los dispositivos de memoria semiconductoras en la cual los datos se almacenarán de manera permanente, siempre y cuando se aplique energía, sin necesidad de reescribir los datos en la memoria en forma periódica.
 - **Dispositivos de memoria dinámica.** Dispositivos de memoria semiconductoras en los cuales los datos *no* se almacenarán en forma permanente, aún y cuando se le aplique energía, a menos que se reescriban en forma periódica en la memoria. A esta última operación se le conoce como operación de *regeneración*.
 - **Memoria principal.** También se le conoce como *memoria de trabajo* de la computadora. Almacena instrucciones y datos con los que la CPU está trabajando en un momento dado. Es la de mayor velocidad en la computadora y siempre es semiconductoras.
 - **Memoria auxiliar.** También se le conoce como *almacenamiento masivo*, ya que guarda cantidades masivas de información externa a la memoria principal. Es más lenta en velocidad que la memoria principal y siempre es no volátil. Los discos magnéticos y los CDs son dispositivos comunes de memoria auxiliar.
-

PREGUNTAS DE REPASO

1. Defina los siguientes términos.
 - (a) *Celda de memoria.*
 - (b) *Palabra de memoria.*
 - (c) *Dirección.*
 - (d) *Byte.*
 - (e) *Tiempo de acceso.*
2. Cierta memoria tiene una capacidad de $8K \times 16$. ¿Cuántos bits hay en cada palabra? ¿Cuántas palabras se almacenan? ¿Cuántas celdas de memoria contiene esta memoria?
3. Explique la diferencia entre las operaciones leer (buscar) y escribir (almacenar).
4. *Verdadero o falso:* una memoria volátil pierde sus datos almacenados cuando se interrumpe la energía eléctrica.
5. Explique la diferencia entre SAM y RAM.
6. Explique la diferencia entre RWM y ROM.
7. *Verdadero o falso:* una memoria dinámica retendrá sus datos mientras se le aplique energía eléctrica.

12.2 OPERACIÓN GENERAL DE LA MEMORIA

Aunque cada tipo de memoria es distinto en su operación interna, ciertos principios de operación básicos son iguales para todos los sistemas de memoria. Comprender estas ideas básicas le ayudará en el estudio de los dispositivos de memoria individuales.

Todo sistema de memoria requiere varios tipos distintos de líneas de entrada y de salida para realizar las siguientes funciones:

1. Seleccionar la dirección en memoria a la que se va a acceder para una operación de lectura o de escritura.
2. Seleccionar una operación de lectura o de escritura a realizar.
3. Suministrar los datos de entrada que se van a almacenar en memoria durante una operación de escritura.
4. Retener los datos de entrada que provienen de la memoria durante una operación de lectura.
5. Habilitar (o deshabilitar) la memoria, de manera que responda (o no) a las entradas de dirección y a la línea de selección de lectura/escritura.

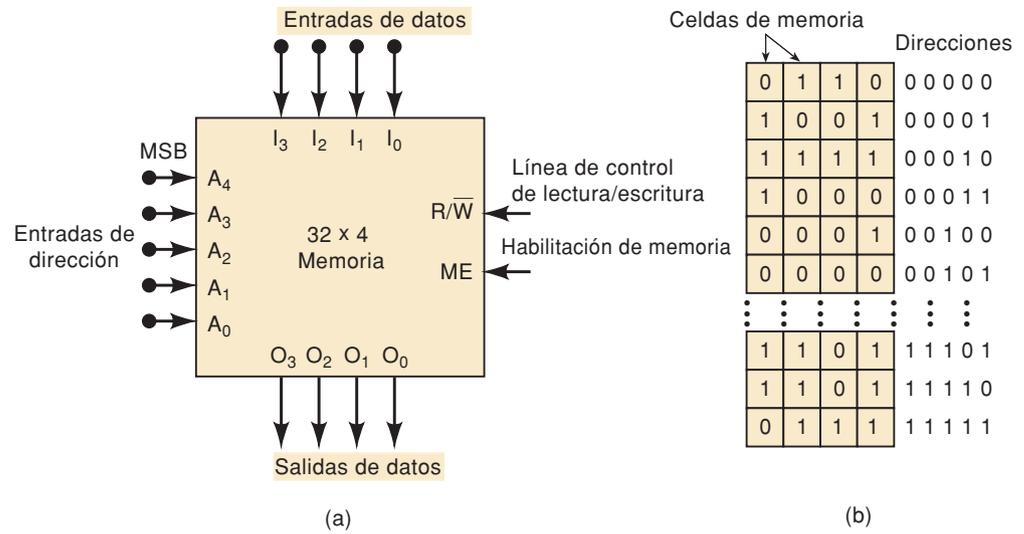
La figura 12-3(a) ilustra estas funciones básicas en un diagrama simplificado de una memoria de 32×4 , la cual almacena 32 palabras de cuatro bits. Como el tamaño de palabra es de cuatro bits, hay cuatro líneas de entrada de datos (I_0 a I_3) y cuatro líneas de salida de datos (O_0 a O_3). Durante una operación de escritura, los datos que se van a almacenar en memoria deben aplicarse a las líneas de entrada de datos. Durante una operación de lectura, la palabra que se va a leer de memoria aparece en las líneas de salida de datos.

Entradas de dirección

Como esta memoria almacena 32 palabras, tiene 32 distintas ubicaciones de almacenamiento y, por lo tanto, 32 distintas direcciones binarias, las cuales varían de 00000 a 11111 (de 0 a 31 en decimal). Por ende, hay cinco entradas de dirección (A_0 a A_4).

FIGURA 12-3

(a) Diagrama de una memoria de 32×4 ; (b) arreglo virtual de las celdas de memoria en 32 palabras de cuatro bits.



Para acceder a una de las ubicaciones de memoria para una operación de lectura o de escritura se aplica el código de dirección de cinco bits para esa ubicación específica a las entradas de dirección. En general, se requieren N entradas de dirección para una memoria con una capacidad de 2^N palabras.

Podemos visualizar la memoria de la figura 12-3(a) como un arreglo de 32 registros, en donde cada registro almacena una palabra de cuatro bits, como se muestra en la figura 12-3(b). La ubicación de cada dirección que se muestra contiene cuatro celdas de memoria que retienen 1s y 0s, los cuales conforman la palabra de datos que se almacena en esa ubicación. Por ejemplo, la palabra de datos 0110 se almacena en la dirección 00000, la palabra de datos 1001 se almacena en la dirección 00001, y así en lo sucesivo.

La entrada R/\overline{W}

Esta entrada controla qué operación de memoria se va a realizar: lectura (R) o escritura (W). La entrada se identifica como R/\overline{W} ; no hay barra sobre la R , lo cual indica que la operación de lectura ocurre cuando $R/\overline{W} = 1$. La barra sobre la W indica que la operación de escritura se realiza cuando $R/\overline{W} = 0$. A menudo se utilizan otras etiquetas para identificar esta entrada. Dos de las más comunes son \overline{W} (escritura) y \overline{WE} (habilita escritura). De nuevo, la barra indica que la operación de escritura se lleva a cabo cuando la entrada está en BAJO. En estos últimos casos se sobreentiende que la operación de lectura ocurre cuando la entrada está en ALTO.

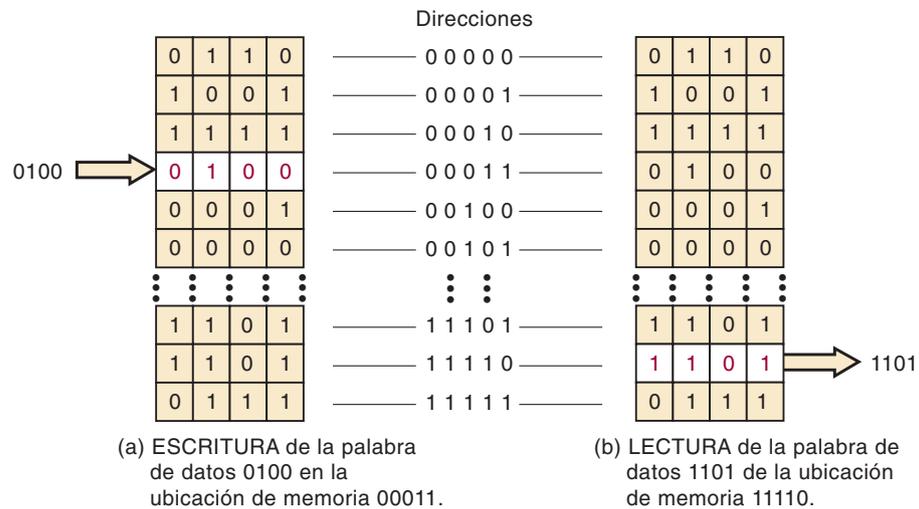
La figura 12-4 muestra una ilustración simplificada de las operaciones de lectura y de escritura. La figura 12-4(a) muestra cómo se escribe la palabra de datos 0100 en el registro de memoria que se encuentra en la ubicación de dirección 00011. Esta palabra de datos se hubiera aplicado a las líneas de entrada de datos de la memoria, y sustituye los datos que estaban almacenados antes en la dirección 00011. La figura 12-4(b) muestra cómo se lee la palabra de datos 1101 de la dirección 11110. Esta palabra de datos aparecería en las líneas de salida de datos de la memoria. Después de la operación de lectura, la palabra de datos 1101 sigue almacenada en la dirección 11110. En otras palabras, la operación de lectura no modifica los datos almacenados.

Habilitación de memoria

Muchos sistemas de memoria tienen ciertos medios para deshabilitar por completo toda o parte de la memoria, de manera que no responda a las demás entradas. Esto se representa en la figura 12-3 como la entrada HABILITACIÓN DE MEMORIA (MEMORY ENABLE), aunque puede tener distintos nombres en los diversos sistemas de memoria, como habilitación de chip (CE , acrónimo de CHIP ENABLE)

FIGURA 12-4

Ilustración simplificada de las operaciones de lectura y de escritura en la memoria de 32×4 ; (a) escritura de la palabra de datos 0100 en la ubicación de memoria 00011; (b) lectura de la palabra de datos 1101 de la ubicación de memoria 11110.



o selección de chip (*CS*, acrónimo de *CHIP SELECT*). Aquí se muestra como una entrada activa en ALTO, la cual habilita a la memoria para que opere en forma normal cuando se mantiene en ALTO. Un nivel BAJO en esta entrada deshabilita la memoria, de manera que no pueda responder a las entradas de dirección y R/\overline{W} .

Este tipo de entrada es útil cuando se combinan varios módulos de memoria para formar una memoria más grande. Más adelante examinaremos esta idea.

EJEMPLO 12-2

Describa las condiciones en cada entrada y salida cuando se lee el contenido de la ubicación de dirección 00100.

Solución

Entradas de dirección: 00100
Entradas de datos: xxxx (no se utilizan)
 R/\overline{W} : ALTO
HABILITACIÓN DE MEMORIA: ALTO
Salidas de datos: 0001

EJEMPLO 12-3

Describa las condiciones en cada entrada y salida cuando se escribe la palabra de datos 1110 en la ubicación de dirección 01101.

Solución

Entradas de dirección: 01101
Entradas de datos: 1110
 R/\overline{W} : BAJO
HABILITACIÓN DE MEMORIA: ALTO
Salidas de datos: xxxx (no se utilizan; por lo general, están en Hi-Z)

EJEMPLO 12-4

Cierta memoria tiene una capacidad de $4K \times 8$.

- ¿Cuántas líneas de entrada y salida de datos tiene?
- ¿Cuántas líneas de dirección tiene?
- ¿Cuál es su capacidad en bytes?

Solución

- (a) Ocho de cada una, ya que el tamaño de palabra es de ocho.
- (b) La memoria almacena $4K = 4 \times 1024 = 4096$ palabras. Por lo tanto, hay 4096 direcciones de memoria. Como $4096 = 2^{12}$, se requiere un código de dirección de 12 bits para especificar una de 4096 direcciones.
- (c) Un byte es de ocho bits. Esta memoria tiene una capacidad de 4096 bytes.

La memoria de ejemplo de la figura 12-3 ilustra las funciones de entrada y salida que son comunes para la mayoría de los sistemas de memoria. Desde luego que cada tipo de memoria puede tener otras líneas de entrada y de salida que sean peculiares para esa memoria. Describiremos estas líneas adicionales de entrada y salida cuando veamos los tipos de memoria individuales.

PREGUNTAS DE REPASO

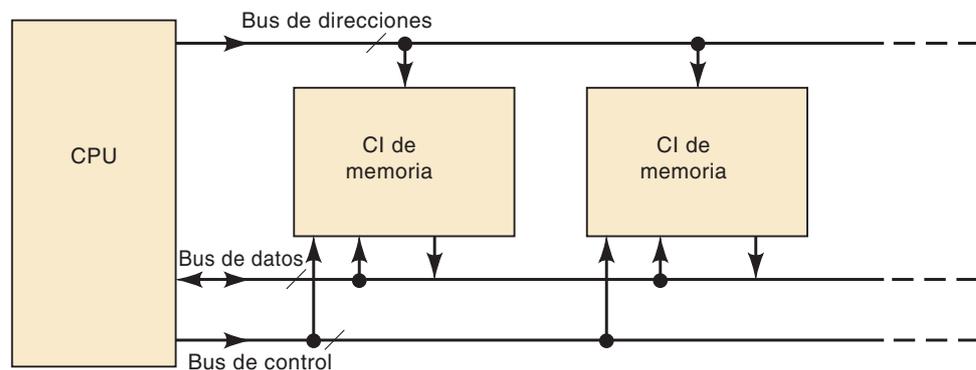
1. ¿Cuántas entradas de dirección, entradas de datos y salidas de datos se requieren para una memoria de $16K \times 12$?
2. ¿Cuál es la función de la entrada R/\overline{W} ?
3. ¿Cuál es la función de la entrada HABILITACIÓN DE MEMORIA?

12-3 CONEXIONES ENTRE CPU Y MEMORIA

Una buena parte de este capítulo se dedica al análisis de la memoria semiconductora, que como vimos antes, conforma la memoria principal de la mayoría de las computadoras modernas. Recuerde que esta memoria principal se encuentra en comunicación constante con la unidad central de procesamiento (CPU). No es necesario estar familiarizado con la operación detallada de una CPU en este punto, por lo que el siguiente tratamiento simplificado de la interfaz entre CPU y memoria le proveerá la perspectiva que necesita para que nuestro estudio de los dispositivos de memoria sea más enriquecedor.

La memoria principal de una computadora está compuesta por CIs tipo RAM y ROM, los cuales se conectan a la CPU a través de tres grupos de líneas de señal o buses, los cuales se representan en la figura 12-5 como las líneas de dirección o bus de dirección, las líneas de datos o el bus de datos, y las líneas de control o el bus de control. Cada uno de estos buses consiste de varias líneas (observe que se representan mediante una sola línea con una barra diagonal), y el número de líneas en cada bus varía de una computadora a otra. Los tres buses juegan un papel necesario para permitir que la CPU escriba datos en la memoria y lea datos de ella.

FIGURA 12-5 Tres grupos de líneas (buses) conectan a los CIs de la memoria principal con el CPU.



Cuando una computadora ejecuta un programa de instrucciones, la CPU busca en forma continua (lee) información de esas ubicaciones en memoria que contienen (1) los códigos del programa que representan las operaciones a realizar y (2) los datos sobre los cuáles se va a trabajar. La CPU también almacenará (escribirá) datos en ubicaciones de memoria, según lo dicten las instrucciones del programa. Cada vez que la CPU desea escribir datos a una ubicación de memoria específica, deben realizarse los siguientes pasos:

Operación de escritura

1. La CPU suministra la dirección binaria de la ubicación de memoria en la que se van a almacenar los datos. Coloca esta dirección en las líneas del bus de dirección.
2. La CPU coloca los datos que se van a almacenar en las líneas del bus de datos.
3. La CPU activa las líneas de señal de control apropiadas para la operación de escritura en la memoria.
4. Los CIs de la memoria decodifican la dirección binaria para determinar qué ubicación se está seleccionando para la operación de almacenamiento.
5. Los datos en el bus de datos se transfieren hacia la ubicación de memoria seleccionada.

Cada vez que la CPU desea leer datos de una ubicación específica en memoria, deben realizarse los siguientes pasos:

Operación de lectura

1. La CPU suministra la dirección binaria de la ubicación de memoria de la cual se van a recuperar los datos. Coloca esta dirección en las líneas del bus de dirección.
2. La CPU activa las líneas de señal de control apropiadas para la operación de lectura de la memoria.
3. Los CIs de la memoria decodifican la dirección binaria para determinar cuál ubicación se está seleccionando para la operación de lectura.
4. Los CIs de memoria colocan los datos de la ubicación de memoria seleccionada en el bus de datos, desde donde se van a transferir hacia la CPU.

Los pasos anteriores deben dejar en claro la función de cada uno de los buses del sistema:

- **Bus de dirección.** Este bus *unidireccional* transporta las salidas de la dirección binaria del CPU hacia los CIs de memoria, para seleccionar una ubicación de memoria.
- **Bus de datos.** Este bus *bidireccional* transporta los datos entre el CPU y los CIs de memoria.
- **Bus de control.** Este bus transporta las señales de control (tales como la señal R/\bar{W}) desde el CPU hacia los CIs de memoria.

A medida que profundicemos sobre los CIs de memoria reales, examinaremos la actividad de las señales que aparecen en estos buses para las operaciones de lectura y de escritura.

PREGUNTAS DE REPASO

1. Nombre los tres grupos de líneas que conectan a la CPU y con la memoria interna.
2. Describa los pasos que se llevan a cabo cuando la CPU lee de memoria.
3. Describa los pasos que se llevan a cabo cuando la CPU escribe en memoria.

12-4 MEMORIAS DE SÓLO LECTURA

La memoria de sólo lectura es un tipo de memoria semiconductora, diseñada para retener datos que sean permanentes o que no cambien con frecuencia. Durante la operación normal, no pueden escribirse datos en una ROM, pero sí se pueden leer. Para algunas ROMs, los datos almacenados deben integrarse durante el proceso de manufactura; para otras ROMs los datos pueden introducirse mediante electricidad. Al proceso de introducir datos se le llama **programación** o *quemado* de la ROM. Algunas ROMs no pueden cambiar sus datos una vez programadas; otras pueden *borrarse* y reprogramarse tantas veces como se requiera. Más adelante veremos con detalle estos diversos tipos de ROMs. Por ahora supondremos que las ROMs se han programado y que contienen datos.

Las ROMs se utilizan para almacenar datos e información que no va a cambiar durante la operación normal de un sistema. Un uso principal para las ROMs es el almacenamiento de programas en las microcomputadoras. Como todas las ROMs son *no volátiles*, estos programas no se pierden cuando se desconecta la energía eléctrica. Cuando la microcomputadora se enciende, puede comenzar de inmediato a ejecutar el programa almacenado en la ROM. Estas memorias también se utilizan para almacenar programas y datos en equipos controlados por microprocesador, tales como: cajas registradoras, electrodomésticos y sistemas de seguridad.

Diagrama de bloques de la ROM

La figura 12-6(a) muestra un diagrama de bloques típico para una ROM. Tiene tres conjuntos de señales: entradas de dirección, entrada(s) de control y salidas de datos. De nuestros análisis anteriores podemos determinar que esta ROM almacena 16 palabras, ya que tiene $2^4 = 16$ direcciones posibles, y cada palabra contiene ocho bits ya que hay ocho salidas de datos. Es decir, ésta es una ROM de 16×8 . Otra manera de describir la capacidad de esta ROM es decir que almacena 16 bytes de datos.

Las salidas de datos de la mayoría de los CIs de ROM son triestado, para permitir la conexión de muchos chips de ROM al mismo bus de datos para expandir la memoria. Los números más comunes de salidas de datos para las ROMs son cuatro, ocho y 16 bits, siendo más comunes las palabras de ocho bits.

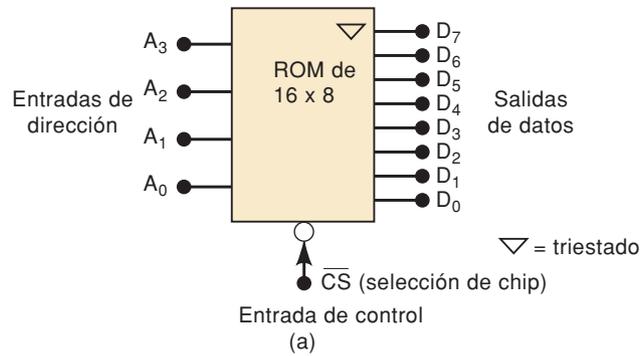
La entrada de control \overline{CS} significa **selección de chip**. En esencia es una entrada de habilitación, para habilitar o deshabilitar las salidas de la ROM. Algunos fabricantes utilizan distintas etiquetas para la entrada de control, como *CE* (habilitación de chip) u *OE* (habilitación de salida). Muchas ROMs tienen dos o más entradas de control que deben estar activas para poder habilitar las salidas de datos, de manera que éstos pueden leerse de la dirección seleccionada. En algunos CIs de ROM, una de las entradas de control (por lo general *CE*) se utiliza para colocar la ROM en un modo de espera de baja energía cuando no está en uso. Esto reduce el consumo de corriente de la fuente de energía del sistema.

La entrada \overline{CS} que se muestra en la figura 12-6(a) es activa en BAJO; por lo tanto, debe encontrarse en el estado BAJO para permitir que aparezcan los datos de la ROM en las salidas de datos. Observe que no hay entrada R/\overline{W} (lectura/escritura) ya que no se puede escribir en la ROM durante su operación normal.

La operación de lectura

Supongamos que la ROM se programó con los datos que se muestran en la tabla de la figura 12-6(b). Se almacenan dieciséis palabras de datos distintas en las 16 ubicaciones de dirección. Por ejemplo, la palabra de datos almacenada en la ubicación 0011 es 10101111. Desde luego que los datos se almacenan en binario dentro de la ROM, pero muy a menudo utilizamos notación hexadecimal para mostrar los datos programados con eficiencia. Esto se hace en la figura 12-6(c).

Para poder leer una palabra de datos de la ROM, necesitamos hacer dos cosas: (1) aplicar las entradas de dirección apropiadas y, después, (2) activar las entradas de control. Por ejemplo, si deseamos leer los datos almacenados en la ubicación



Palabra	Dirección				Datos							
	A ₃	A ₂	A ₁	A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	0	0	0	0	1	1	0	1	1	1	1	0
1	0	0	0	1	0	0	1	1	1	0	1	0
2	0	0	1	0	1	0	0	0	0	1	0	1
3	0	0	1	1	1	0	1	0	1	1	1	1
4	0	1	0	0	0	0	0	1	1	0	0	1
5	0	1	0	1	0	1	1	1	1	0	1	1
6	0	1	1	0	0	0	0	0	0	0	0	0
7	0	1	1	1	1	1	1	0	1	1	0	1
8	1	0	0	0	0	0	1	1	1	1	0	0
9	1	0	0	1	1	1	1	1	1	1	1	1
10	1	0	1	0	1	0	1	1	1	0	0	0
11	1	0	1	1	1	1	0	0	0	1	1	1
12	1	1	0	0	0	0	1	0	0	1	1	1
13	1	1	0	1	0	1	1	0	1	0	1	0
14	1	1	1	0	1	1	0	1	0	0	1	0
15	1	1	1	1	0	1	0	1	1	0	1	1

(b)

Palabra	Dirección				Datos							
	A ₃	A ₂	A ₁	A ₀	D ₇ -D ₀							
0	0				DE							
1	1				3A							
2	2				85							
3	3				AF							
4	4				19							
5	5				7B							
6	6				00							
7	7				ED							
8	8				3C							
9	9				FF							
10	A				B8							
11	B				C7							
12	C				27							
13	D				6A							
14	E				D2							
15	F				5B							

(c)

FIGURA 12-6 (a) Símbolo de bloque típico de la ROM; (b) tabla que muestra los datos binarios en cada una de las ubicaciones de memoria; (c) la misma tabla en hexadecimal.

0111 de la ROM en la figura 12-6, debemos aplicar $A_3A_2A_1A_0 = 0111$ a las entradas de dirección y después aplicar un nivel BAJO a \overline{CS} . Las entradas de dirección se decodificarán dentro de la ROM para seleccionar la palabra de datos correcta, 11101101, que debe aparecer en las salidas D_7 a D_0 . Si \overline{CS} se mantiene en ALTO, las salidas de la ROM se deshabilitarán y quedarán en el estado Hi-Z.

PREGUNTAS DE REPASO

1. Verdadero o falso: todas las ROMs son no volátiles.
2. Describa el procedimiento para leer de ROM.
3. ¿Qué es programación o quemado en una ROM?

12-5 ARQUITECTURA DE LA ROM

La arquitectura (estructura) interna de un CI de ROM es muy compleja, por lo que no intentaremos familiarizarnos con todos sus detalles. No obstante, por cuestión didáctica sí podemos ver un diagrama simplificado de la arquitectura interna, como el que se muestra en la figura 12-7 para la ROM de 16×8 . Hay cuatro partes básicas: arreglo de registros, decodificador de fila, decodificador de columna y búferes de salida.

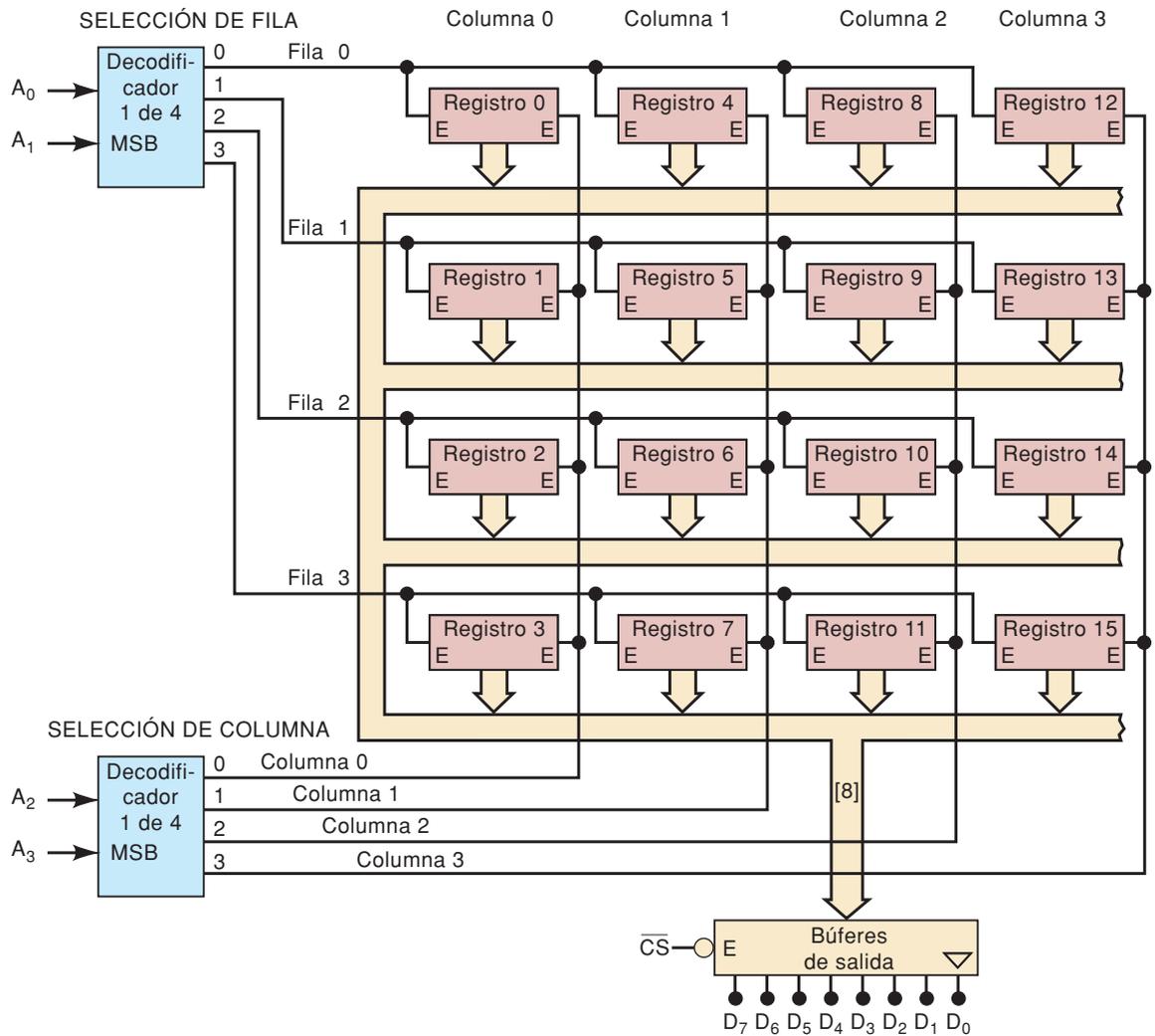


FIGURA 12-7 Arquitectura de una ROM de 16×8 . Cada registro almacena una palabra de ocho bits

Arreglo de registros

El arreglo de registros almacena los datos que se programan en la ROM. Cada registro contiene varias celdas de memoria que equivalen al tamaño de palabra. En este caso, cada registro almacena una palabra de ocho bits. Los registros se ordenan en un arreglo de matriz cuadrada, común para muchos chips de memoria semiconductora. Podemos especificar que la posición de cada registro está en una fila y una columna específicas. Por ejemplo, el registro 0 está en la fila 0, columna 0, y el registro 9 está en la fila 1, columna 2.

Decodificadores de dirección

El código de dirección $A_3A_2A_1A_0$ que se aplica es el que determina qué registro en el arreglo se habilitará para colocar su palabra de datos de ocho bits en el bus. Los bits de dirección A_1A_0 se alimentan a un decodificador 1 de 4, el cual activa una línea de selección de fila, y los bits de dirección A_3A_2 se alimentan a un segundo decodificador 1 de 4, el cual activa una línea de selección de columna. Sólo habrá un registro

en la fila y la columna seleccionadas por las entradas de dirección, por lo que éste es el que se habilitará.

EJEMPLO 12-5

¿Cuál registro se habilitará a través de la dirección de entrada 1101?

Solución

$A_3A_2 = 11$ hará que el decodificador de columna active la línea de selección de la columna 3, y $A_1A_0 = 01$ hará que el decodificador de fila active la línea de selección de la fila 1. Esto hará que ambas entradas de habilitación del registro 13 cambien a ALTO, para provocar que se coloquen sus salidas de datos en el bus. Observe que los otros registros de la columna 3 tendrán sólo una entrada de habilitación activada; lo mismo se aplica para los registros de la otra fila 1.

EJEMPLO 12-6

¿Qué dirección de entrada habilitará el registro 7?

Solución

Las entradas de habilitación de este registro están conectadas a las líneas de selección de la fila 3 y la columna 1, respectivamente. Para seleccionar la fila 3, las entradas A_1A_0 deben estar en 11 y para seleccionar la columna 1, las entradas A_3A_2 deben estar en 01. Por lo tanto, la dirección requerida será $A_3A_2A_1A_0 = 0111$.

Búferes de salida

El registro que se habilite mediante las entradas de dirección colocará sus datos en el bus de datos. Estos datos se alimentan hacia los búferes de salida, los cuales pasarán esos datos hacia las salidas de datos externas, siempre y cuando \overline{CS} permanezca en BAJO. Si \overline{CS} cambia a ALTO, los búferes de salida se colocarán en el estado Hi-Z y las salidas D_7 a D_0 estarán flotando.

La arquitectura que se muestra en la figura 12-7 es similar a la de muchas ROMs de CI. Dependiendo del número de palabras de datos almacenadas, los registros en algunas ROMs no se ordenarán en un arreglo cuadrado. Por ejemplo, el chip Intel 27C64 es una ROM tipo CMOS que almacena 8192 palabras de ocho bits. Sus 8192 registros están ordenados en un arreglo de 256 filas \times 32 registros. Las capacidades de la ROM varían desde 256×4 hasta $8M \times 8$.

EJEMPLO 12-7

Describe la arquitectura interna de una ROM que almacena 4K bytes y utiliza un arreglo de registros cuadrado.

Solución

4K son en realidad $4 \times 1024 = 4096$, por lo que esta ROM almacena 4096 palabras de ocho bits. Se puede considerar que cada palabra se almacena en un registro de 8 bits y que hay 4096 registros conectados a un bus de datos común interno del chip. Como $4096 = 64^2$, los registros están ordenados en un arreglo de 64×64 ; es decir, hay 64 filas y 64 columnas. Esto requiere un decodificador 1 de 64 para decodificar seis entradas de dirección para la selección de fila, y un segundo decodificador 1 de 64 para decodificar otras seis entradas de dirección para la selección de columna. Por lo tanto, se requiere un total de 12 entradas de dirección. Esto tiene sentido, ya que $2^{12} = 4096$ y hay 4096 direcciones distintas.

PREGUNTAS DE REPASO

1. ¿Qué código de dirección de entrada se requiere si queremos leer los datos del registro 9 en la figura 12-7?
2. Describa la función del decodificador de selección de fila, del decodificador de selección de columna y de los búferes de salida en la arquitectura de la ROM.

12-6 SINCRONIZACIÓN DE LA ROM

Durante una operación de lectura se produce un tiempo de propagación entre la aplicación de las entradas de una ROM y la aparición de las salidas de datos. Este tiempo de propagación, conocido como tiempo de acceso (t_{ACC}), es una medida de la velocidad de operación de la ROM. En la figura 12-8 se describe el tiempo en forma gráfica, mediante las formas de onda.

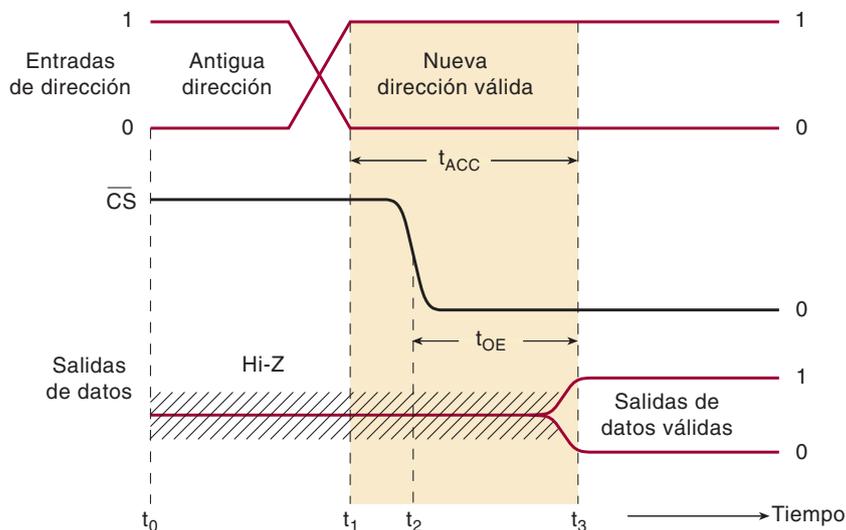
La forma de onda superior representa las entradas de dirección; la forma de onda de en medio es la señal de selección de chip \overline{CS} activa en bajo; y la forma de onda inferior representa las salidas de datos. En el tiempo t_0 , todas las entradas de dirección se encuentran en un nivel específico, algunas en ALTO y otras en BAJO. \overline{CS} está en ALTO, por lo que las salidas de datos de la ROM se encuentran en su estado Hi-Z (el cual se representa mediante la línea sombreada).

Justo antes de t_1 , las entradas de dirección cambian a una nueva dirección para una nueva operación de lectura. En t_1 , la nueva dirección es válida; es decir, cada entrada de dirección se encuentra en un nivel lógico válido. En este punto, el circuito interno de la ROM comienza a decodificar las nuevas entradas de dirección para seleccionar el registro que debe enviar sus datos a los búferes de salida. En t_2 se activa la entrada \overline{CS} para habilitar los búferes de salida. Por último, en t_3 las salidas cambian del estado Hi-Z a los datos válidos que representan los datos almacenados en la dirección especificada.

El retraso de tiempo entre t_1 , cuando la nueva dirección se hace válida, y t_3 , cuando las salidas de datos se hacen válidas, es el tiempo de acceso t_{ACC} . Las ROMs bipolares ordinarias tendrán tiempos de acceso que varían en el intervalo de 30 a 90 ns; los tiempos de acceso de los dispositivos NMOS variarán de 35 a 500 ns. Las mejoras en la tecnología CMOS han hecho posibles tiempos de acceso en el intervalo de 20 a 60 ns. En consecuencia, los dispositivos bipolares y NMOS se producen raras veces en ROMs más recientes (grandes).

Otro parámetro importante de sincronización es el *tiempo de habilitación de salida* (t_{OE}), que es el retraso entre la entrada \overline{CS} y la salida de datos válida. Los valores

FIGURA 12-8
Sincronización típica para una operación de lectura de la ROM.



comunes para t_{OE} son de 10 a 20 ns para las ROMs bipolares, de 25 a 100 ns para las ROMs NMOS y de 12 a 50 para las ROMs CMOS. Este parámetro de sincronización es importante en situaciones en las que las entradas de dirección ya se encuentran establecidas en sus nuevos valores, pero las salidas de la ROM todavía no se han habilitado. Cuando CS cambie a BAJO para habilitar las salidas, el retraso será t_{OE} .

12-7 TIPOS DE ROMS

Ahora que tenemos una comprensión general sobre la arquitectura interna y la operación externa de los dispositivos de ROM, analizaremos los diversos tipos de ROMs para ver cómo difieren en la manera en que se programan, se borran y se reprograman.

ROM programada por máscara

La ROM programada por máscara (MROM) almacena su información al momento en que se fabrica el circuito integrado. Como puede ver en la figura 12-9, las ROMs están compuestas de un arreglo rectangular de transistores. La información se almacena ya sea mediante la conexión o la desconexión de la fuente de un transistor con

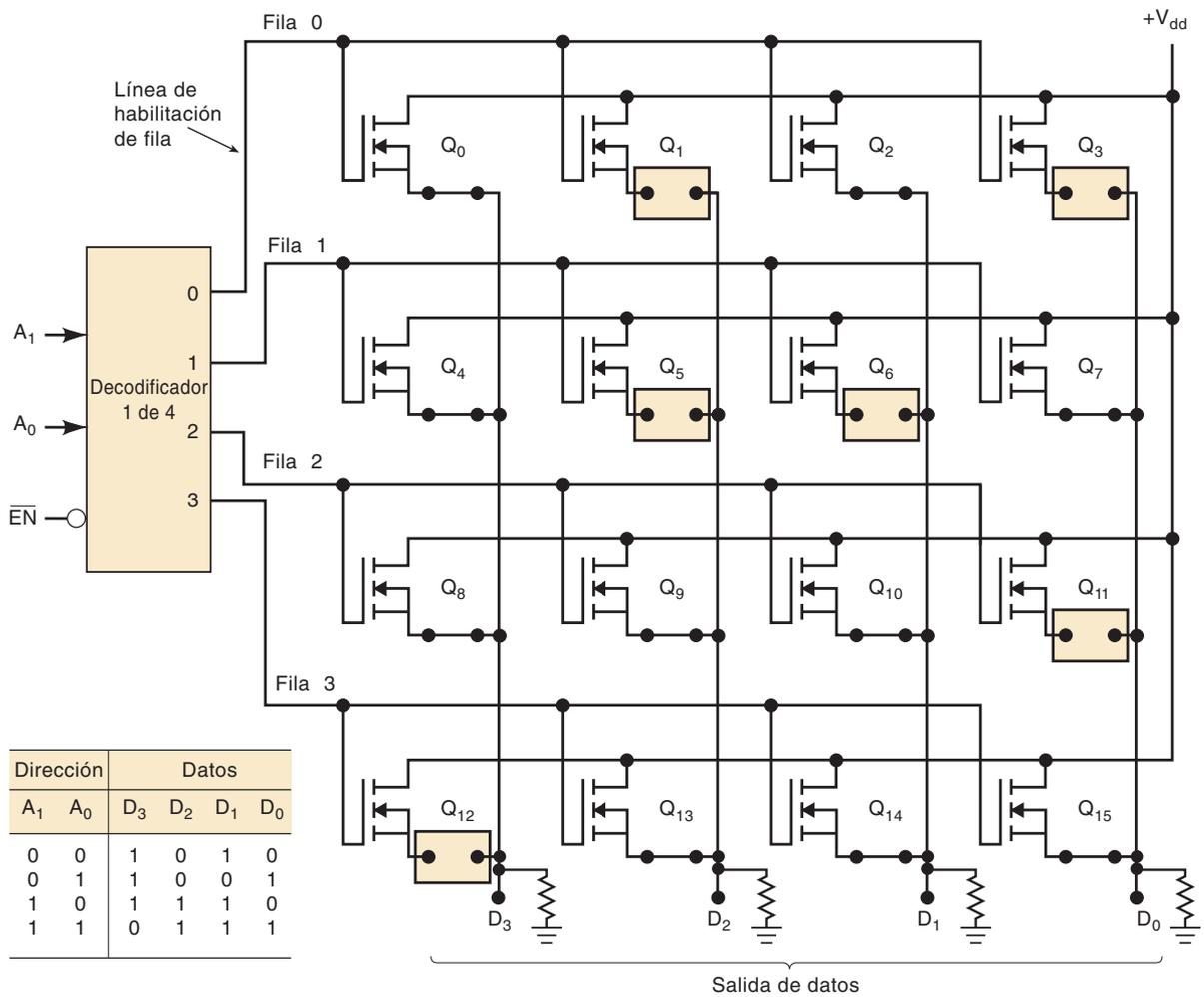


FIGURA 12-9 La estructura de una MROM tipo MOS muestra que se utiliza un MOSFET para cada celda de memoria. Una conexión de fuente abierta almacena un “0”; una conexión de fuente cerrada almacena un “1”.

la columna de salida. El último paso en el proceso de fabricación es formar todas estas rutas conductoras o conexiones. El proceso utiliza una “máscara” para depositar metales en el silicio, los cuales determinan en dónde se forman las conexiones de una manera similar al uso de patrones y pintura en aerosol, pero en una escala mucho más pequeña. La máscara es muy precisa y costosa, y debe fabricarse de manera específica para el cliente, con la información binaria correcta. En consecuencia, este tipo de ROM es económica sólo cuando se van a fabricar muchas ROMs con la misma información exacta.

Por lo general, a las ROMs programadas por máscara se les conoce sólo como ROMs, lo cual puede ser confuso ya que el término ROM en realidad representa a la amplia categoría de dispositivos que, durante la operación normal, sólo se pueden leer. Utilizaremos la abreviación MROM cada vez que hagamos referencia a las ROMs programadas por máscara.

La figura 12-9 muestra la estructura de una pequeña MROM tipo MOS de 16 celdas de memoria ordenadas en cuatro filas o cuatro celdas. Cada celda es un transistor MOSFET de canal N, conectado en la configuración de colector común (entrada en la compuerta, salida en la fuente). La fila superior de celdas (FILA 0) constituye un registro de cuatro bits. Observe que algunos de los transistores en esta fila (Q_0 y Q_2) tienen su fuente conectada a la línea de la columna de salida, mientras que otros (Q_1 y Q_3) no. Lo mismo se aplica a las celdas en cada una de las otras filas. La presencia o ausencia de estas conexiones de fuente determina si una celda almacena un 1 o un 0, respectivamente. La condición de cada conexión de fuente se controla durante la producción mediante la máscara fotográfica, con base en los datos suministrados por el cliente.

Observe que las salidas de datos están conectadas a las líneas de columna. Por ejemplo, al hacer referencia a la salida D_3 , cualquier transistor que tenga una conexión de la fuente (como Q_0 , Q_4 y Q_8) a la columna de salida puede cambiar a V_{dd} hacia la columna, para que tenga un nivel ALTO lógico. Si V_{dd} no se conecta a la línea de columna, la salida se mantendrá en un nivel lógico BAJO mediante la resistencia descendente. En un momento determinado, se encenderá un máximo de un transistor en una columna debido al decodificador de fila.

El decodificador de 1 a 4 se utiliza para decodificar las entradas de dirección A_1A_0 para seleccionar de cuál fila (registro) se van a leer los datos. Las salidas activas en ALTO del decodificador proporcionan las líneas de habilitación de FILA que son las entradas de compuerta para las diversas filas de celdas. Si la entrada de habilitación \overline{EN} del decodificador se mantiene en ALTO, todas las salidas del decodificador estarán en su estado BAJO inactivo, y todos los transistores en el arreglo estarán apagados debido a la ausencia de voltaje en compuerta. Debido a esta situación, todas las salidas de datos estarán en el estado BAJO.

Cuando \overline{EN} se encuentra en su estado activo en BAJO, las condiciones en las entradas de dirección determinan qué fila (registro) se habilitará para que puedan leerse sus datos en las salidas de datos. Por ejemplo, para leer la FILA 0, las entradas A_1A_0 se establecen en 00. Esto coloca un nivel ALTO en la línea de la FILA 0; todas las demás líneas de fila están en 0 V. Este nivel ALTO en FILA 0 enciende los transistores Q_0 , Q_1 , Q_2 y Q_3 . Cuando todos los transistores en la fila estén conduciendo, V_{dd} se cambiará a cada una de las terminales de la fuente del transistor. Las salidas D_3 y D_1 cambiarán a ALTO debido a que Q_0 y Q_2 están conectados a sus respectivas columnas. D_2 y D_0 permanecerán en BAJO ya que no hay ruta de las terminales de fuente de Q_1 y Q_3 hacia sus columnas. De manera similar, la aplicación de los otros códigos de dirección producirá salidas de datos del registro correspondiente. La tabla en la figura 12-9 muestra los datos para cada dirección. Usted deberá verificar cómo se correlaciona esto con las conexiones de fuente a las diversas celdas.

EJEMPLO 12-8

Las MROMs pueden usarse para almacenar tablas de funciones matemáticas. Demuestre cómo puede usarse la MROM de la figura 12-9 para almacenar la función $y = x^2 + 3$, en donde la dirección de entrada suministra el valor para x y el valor de los datos de salida es y .

TABLA 12-1

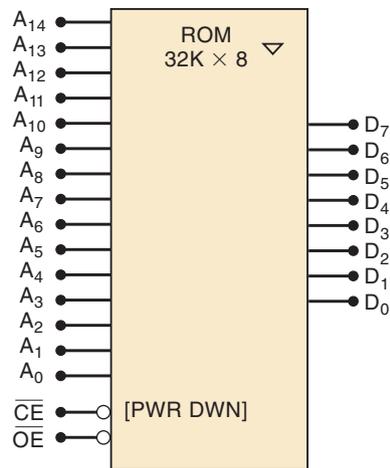
x		y = x ² = 3			
A ₁	A ₀	D ₃	D ₂	D ₁	D ₀
0	0	0	0	1	1
0	1	0	1	0	0
1	0	0	1	1	1
1	1	1	1	0	0

Solución

El primer paso es utilizar una tabla que muestre la salida deseada para cada conjunto de entradas. El número binario de entrada x se representa mediante la dirección A₁A₀. El número binario de salida es el valor deseado de y. Por ejemplo, cuando x = A₁A₀ = 10₂ = 2₁₀, la salida deberá ser 2² + 3 = 7₁₀ = 0111₂. La tabla completa se muestra en la tabla 12-1. Esta tabla se proporciona al fabricante de MROMs para que desarrolle la máscara que realizará las conexiones apropiadas dentro de las celdas de memoria, durante el proceso de fabricación. Por ejemplo, la primera fila en la tabla indica que las conexiones a la fuente de Q₀ y Q₁ se dejarán desconectadas, mientras que las conexiones a Q₂ y Q₃ sí se realizarán.

Como vimos en el capítulo 9, por lo general, las MROMs tienen salidas triestado que les permiten utilizarse en un sistema de bus. En consecuencia, debe haber una entrada de control para habilitar y deshabilitar las salidas triestado. Esta entrada de control se etiqueta por lo común como OE (habilitación de salida). Para poder distinguir esta entrada de habilitación triestado de la entrada de habilitación del decodificador de dirección, a esta última, por lo general, se le hace referencia como CE (habilitación de chip). La señal de habilitación de chip realiza algo más que sólo habilitar el decodificador de dirección. Cuando se deshabilita CE, todas las funciones del chip están deshabilitadas, incluyendo las salidas triestado, y todo el circuito se coloca en un modo de **bajo consumo de energía**, en el cual se consume una cantidad mucho menor de corriente de la fuente de energía. La figura 12-10 muestra una MROM de 32K × 8. Las 15 líneas de dirección (A0-A14) pueden identificar 2¹⁵ ubicaciones de memoria (32, 767 o 32K). Cada ubicación de memoria guarda un valor de datos de ocho bits, que puede colocarse en las líneas de datos D7-D0 cuando el chip está habilitado y las salidas están habilitadas.

FIGURA 12-10 Símbolo lógico para una MROM de 32K × 8.

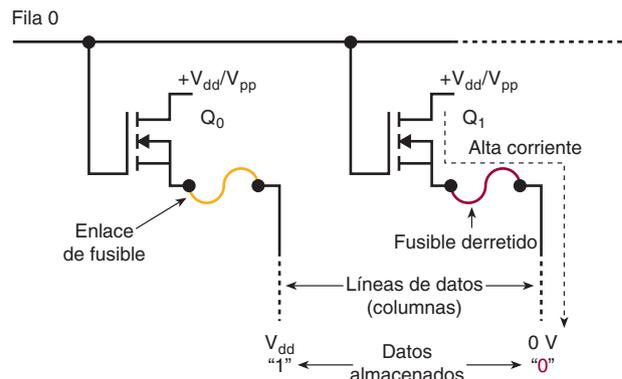


ROMs programables (PROMs)

Una ROM programable por máscara es muy costosa y se utiliza sólo en aplicaciones donde se debe fabricar una alta cantidad de memorias idénticas, y así repartir el costo entre muchas unidades. Para las aplicaciones de menor volumen, los fabricantes han desarrollado PROMs con **enlaces de fusible**, las cuales son programables por el usuario; esto es, no se programan durante el proceso de fabricación sino que el usuario las programa según sus necesidades. Sin embargo, una vez programada, una PROM es como una MROM ya que no puede borrarse y reprogramarse. Por lo tanto, si el programa en la PROM tiene fallas o debe modificarse, hay que desechar esa PROM. Por esta razón, a estos dispositivos se les conoce por lo general como ROMs “programables una sola vez” (OTP).

La estructura de una PROM con enlace de fusible es muy similar a la estructura de una MROM, ya que ciertas conexiones se dejan intactas o se abren para poder programar una celda de memoria como 1 o 0, respectivamente. Una PROM viene del fabricante con una conexión delgada de enlace de fusible en la terminal de la fuente de cada transistor. En esta condición, cada transistor almacena un 1. El usuario entonces puede “volar” el fusible de cualquier transistor que necesite almacenar un 0. Por lo común, la programación o “quemado” de datos en una PROM se realiza mediante la selección de una fila al aplicar la dirección deseada en las entradas de dirección, después se colocan los datos deseados en las terminales de datos y luego se aplica un pulso a una terminal especial de programación en el CI. La figura 12-11 muestra el funcionamiento interno de este proceso.

FIGURA 12-11 Las PROMs utilizan enlaces de fusible; el usuario puede abrir estos enlaces en forma selectiva para programar un 0 lógico en una celda.



Todos los transistores en la fila seleccionada (fila 0) se encienden y se aplica V_{pp} en sus terminales de colector. Las columnas (líneas de datos) que tengan un 0 lógico en ellas (por ejemplo, Q_1) proveerán una ruta de alta corriente a través del enlace de fusible para quemarlo y dejarlo abierto, con lo cual se almacenará un 0 lógico en forma permanente. Las columnas que tengan un 1 lógico (por ejemplo, Q_0) tendrán V_{pp} en un lado del fusible y V_{dd} en el otro lado, con lo cual consumirán mucha menos corriente y dejarán el fusible intacto. Una vez que se hayan programado todas las ubicaciones de dirección de esta forma, los datos quedarán almacenados en forma permanente en la PROM y podrán leerse una y otra vez, mediante el acceso a la dirección apropiada. Los datos no cambiarán cuando se quite la energía del chip PROM, ya que no hay nada que pueda hacer que un enlace de fusible abierto se vuelva a cerrar.

Una PROM se programa mediante el mismo equipo y proceso que describimos en el capítulo 4 para programar un PLD. La TMS27PC256 es una PROM tipo CMOS muy popular, con una capacidad de $32\text{K} \times 8$ y una disipación de energía en modo suspendido de sólo 1.4 mW. Está disponible con tiempos de acceso que varían entre 100 y 250 ns.

ROM programable y borrable (EPROM)

Una EPROM puede programarse por el usuario, y también puede *borrarse* y reprogramarse casi todas las veces que se requiera. Una vez programada, la EPROM es

una memoria *no volátil* que almacenará sus datos en forma indefinida. El proceso para programar una EPROM es el mismo que para una PROM.

El elemento de almacenamiento de una EPROM es un transistor MOS con una compuerta de silicio que no tiene conexión eléctrica (es decir, una compuerta flotante) pero está muy cerca de un electrodo. En su estado normal no hay carga almacenada en la compuerta flotante, por lo que el transistor producirá un 1 lógico cada vez que el decodificador de dirección lo seleccione. Para programar un 0 se utiliza un pulso de alto voltaje para dejar una carga neta en la compuerta flotante. Esta carga hace que el transistor produzca como salida un 0 lógico cuando se selecciona. Como la carga está atrapada en la compuerta flotante y no tiene ruta de descarga, el 0 se almacenará hasta que se borre. Para borrar los datos se restauran todas las celdas a un 1 lógico. Para ello se neutraliza la carga en el electrodo flotante al exponer el silicio a una luz ultravioleta (UV) de alta intensidad durante varios minutos.

La 27C64 es un ejemplo de un CI de memoria pequeño de $8K \times 8K$, disponible como PROM “programable sólo una vez” (OTP) o como EPROM borrable con luz UV. La diferencia obvia en los dos CIs es la “ventana” transparente de cuarzo de la EPROM, que se muestra en la figura 12-12(b). Esta ventana permite que la luz UV se refleje sobre el silicio. Ambas versiones operan a partir de una sola fuente de energía de +5 V durante su operación normal.

La figura 12-12(a) es el símbolo lógico para la 27C64. Observe que muestra 13 entradas de dirección (debido a que $2^{13} = 8192$) y ocho salidas de datos. Tiene cuatro entradas de control. \overline{CE} es la entrada de habilitación de chip que se utiliza para colocar el dispositivo en un modo de suspensión, en donde se reduce su consumo de energía. \overline{OE} es la entrada de habilitación de las salidas y se utiliza para controlar los búferes triestado de salida de datos del dispositivo, de manera que éste pueda conectarse al bus de datos de un microprocesador sin contención de bus. V_{PP} es el voltaje de programación especial requerido durante el proceso de programación. \overline{PGM} es la entrada de habilitación de programa que se activa para almacenar datos en la dirección seleccionada.

La 27C64 tiene varios modos de operación que se controlan mediante las terminales \overline{CE} , \overline{OE} , V_{PP} y PGM, como se presentan en la figura 12-12(c). El modo pro-

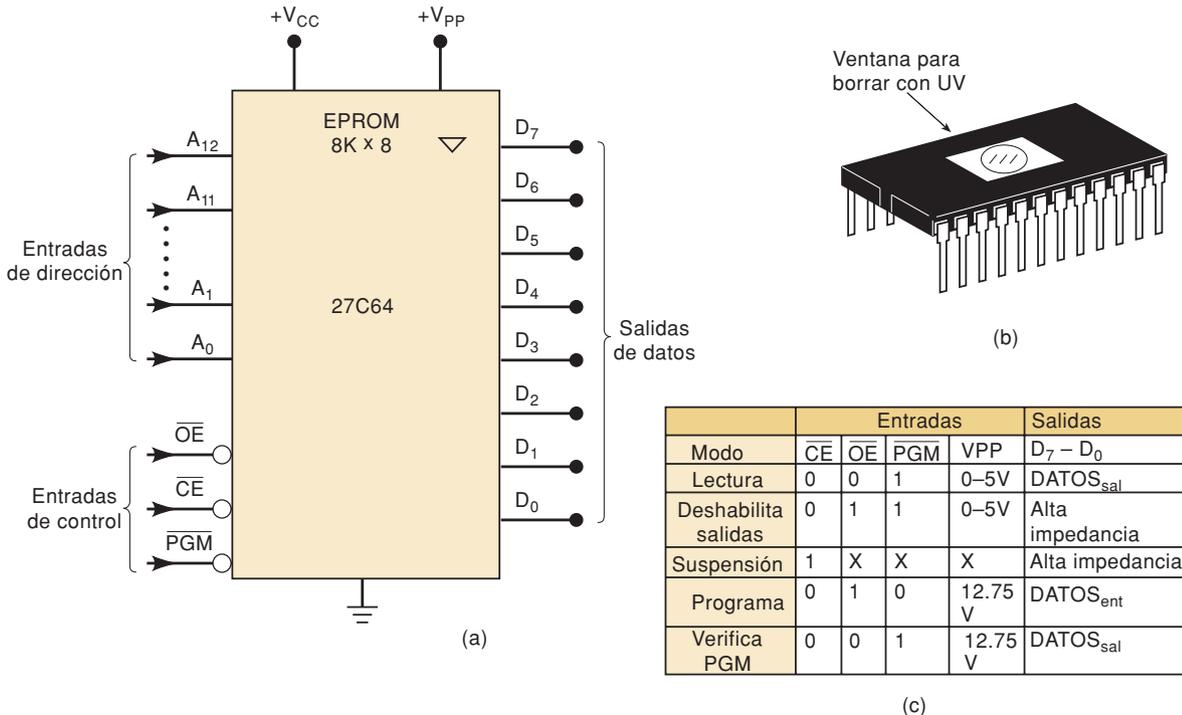


FIGURA 12-12 (a) Símbolo lógico para la EPROM 27C64; (b) encapsulado típico de una EPROM en el que se muestra una ventana ultravioleta; (c) modos de operación del 27C64.

grama se utiliza para escribir nuevos datos en las celdas de la EPROM. Esto se hace con más frecuencia en una EPROM “limpia”, una que haya sido borrada antes con luz UV para que todas las celdas sean 1. El proceso de programación escribe una palabra de ocho bits en una ubicación de memoria en un momento dado, como se indica a continuación: (1) la dirección se aplica a las terminales de dirección; (2) los datos deseados se colocan en las terminales de datos, que funcionan como entradas durante el proceso de programación; (3) se aplica un voltaje de programación más alto de 12.75 V en V_{PP} ; (4) \overline{CE} se mantiene en BAJO; (5) se aplica un pulso a \overline{PGM} para que cambie a BAJO durante 100 μ s y los datos se leen de vuelta. Si los datos no se almacenaron con éxito, se aplica otro pulso a \overline{PGM} . Esto se repite en la misma dirección hasta que los datos se almacenan con éxito.

Una EPROM limpia puede programarse en menos de un minuto, una vez que se hayan introducido, transferido o descargado los datos deseados en el programador de EPROMs. La 27C512 es una EPROM común de 64K \times 8 que opera en forma muy parecida a la 27C64, pero ofrece más capacidad de almacenamiento.

Las principales desventajas de las UV-EPROMs son que deben removerse del circuito para programarse y borrarse, la operación borra todo el chip y puede durar hasta 20 minutos.

PROM programable y borrrable eléctricamente (EEPROM)

Las desventajas de la EPROM se solucionaron con el desarrollo de la **PROM programable y borrrable eléctricamente (EEPROM)** como una mejora sobre la EPROM. La EEPROM retiene la misma estructura de compuerta flotante que la EPROM, pero agrega una región de óxido muy delgada por encima del colector de la celda de memoria de MOSFET. Esta modificación produce la característica principal de la EEPROM: su capacidad de borrarse mediante electricidad. Al aplicar un alto voltaje (21 V) entre la compuerta y el colector del MOSFET, se puede inducir una carga en la compuerta flotante, en donde permanecerá aunque se corte la energía; si se invierte el mismo voltaje se eliminarán las cargas atrapadas de la compuerta flotante y se borrará la celda. Como este mecanismo de transporte de cargas requiere corrientes muy bajas, el borrado y la programación de una EEPROM pueden realizarse *en el circuito* (es decir, sin una fuente de luz UV ni dispositivo especial para programar PROMs).

Otra ventaja de la EEPROM en comparación con la EPROM es la habilidad de borrar y reescribir bytes *individuales* (palabras de ocho bits) en el arreglo de memoria mediante electricidad. Durante una operación de escritura, el circuito interno borra de manera automática todas las celdas en una ubicación de dirección antes de escribir los nuevos datos. Esta capacidad de borrar bytes facilita en forma considerable la realización de modificaciones en los datos que se almacenan en una EEPROM.

Las primeras EEPROMs, como la 2816 de Intel, requerían circuitos externos de soporte apropiados para los chips de memoria. Estos circuitos de soporte incluyen el voltaje de programación de 21 V (V_{PP}), que por lo común se genera a partir de una fuente de +5 V a través de un convertidor de corriente directa a corriente directa, e incluye circuitos para controlar la sincronización y la secuencia de las operaciones de borrar y programar. Los dispositivos más recientes, como el 2864 de Intel, integran estos circuitos de soporte en el mismo chip junto con el arreglo de memoria, por lo que sólo se requiere una sola terminal de energía de +5 V. Esto hace que la EEPROM sea tan fácil de usar como la memoria de lectura/escritura que veremos en breve.

La capacidad de borrar bytes de la EEPROM y su alto nivel de integración incurren en dos faltas: densidad y costo. La complejidad de la celda de memoria y los circuitos de soporte en el chip ponen a las EEPROM en desventaja considerable con las EPROM, en cuanto a la capacidad de bits por milímetro cuadrado de silicio; una EEPROM de 1 Mbit requiere casi el doble de silicio que una EPROM de 1 Mbit. Por lo tanto, y a pesar de su superioridad operacional, las desventajas de la EEPROM en cuanto a densidad y efectividad en costo han evitado que sustituya a la EPROM en aplicaciones en donde la densidad y el costo son factores imperativos.

El símbolo lógico para el Intel 2864 se muestra en la figura 12-13(a). Se organiza como un arreglo de $8K \times 8$, con 13 direcciones de entrada ($2^{13} = 8192$) y ocho terminales de E/S de datos. Tres entradas de control determinan el modo de operación, de acuerdo con la tabla que se proporciona en la figura 12-13(b). Cuando $\overline{CE} = \text{ALTO}$, el chip está en su modo de suspensión de bajo consumo de energía, en el cual no se

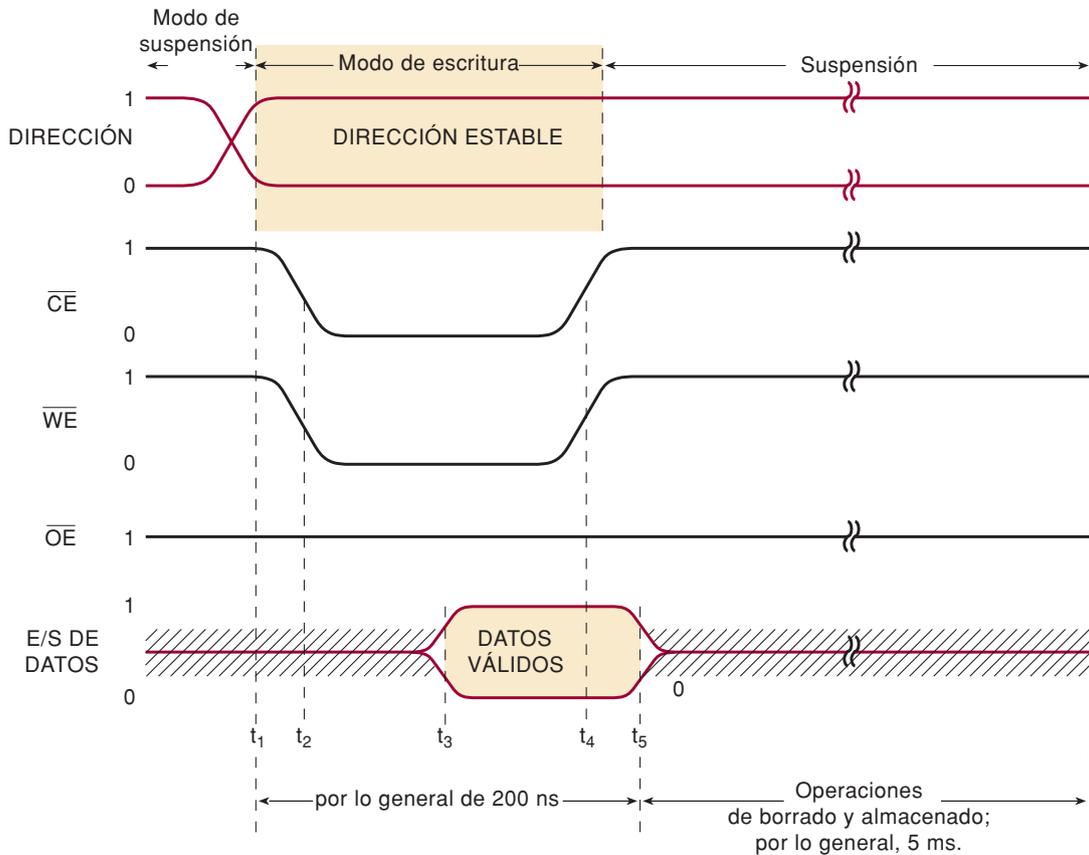
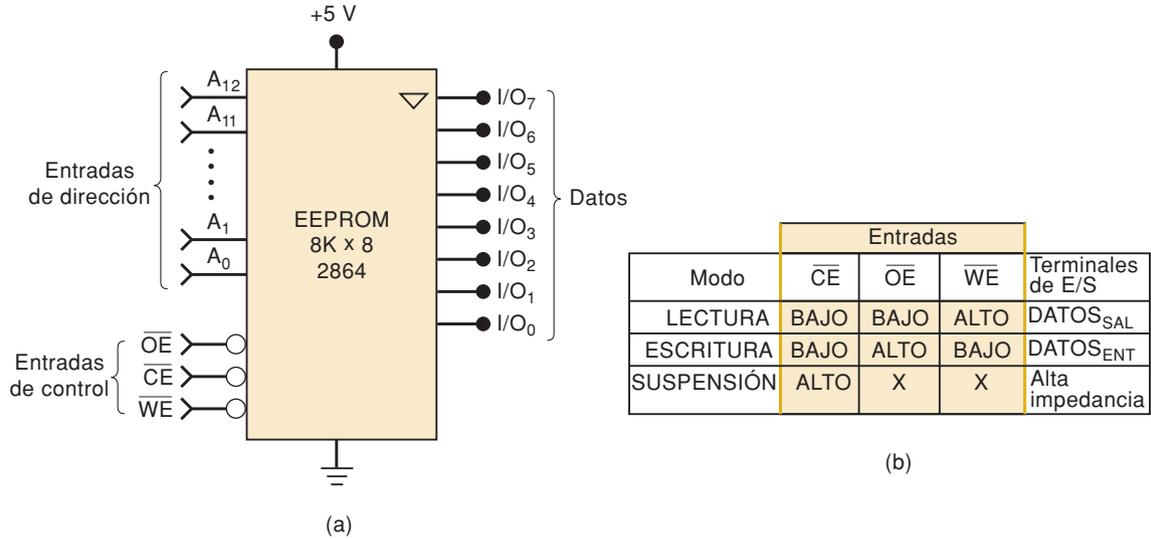


FIGURA 12-13 (a) Símbolo para la EEPROM 2864; (b) modos de operación; (c) sincronización para la operación de escritura.

realizan operaciones en ninguna ubicación de memoria y las terminales de datos se encuentran en el estado Hi-Z.

Para leer el contenido de una ubicación de memoria determinada, se aplica la dirección deseada en las terminales de dirección; \overline{CE} se lleva a BAJO; y la terminal de habilitación de salida \overline{OE} se lleva a BAJO para habilitar los búferes de datos de salida del chip. La terminal de habilitación de salida \overline{WE} se mantiene en ALTO durante una operación de lectura.

Para escribir en (es decir, programar) una ubicación de memoria se deshabilitan los búferes de salida, de manera que los datos que se van a escribir puedan aplicarse como entradas para las terminales de E/S. La figura 12-13(c) muestra un diagrama de la sincronización para la operación de escritura. Antes de t_1 , las entradas \overline{CE} y \overline{WE} se llevan a BAJO para comenzar la operación de escritura; \overline{OE} está en ALTO, por lo que las terminales de datos permanecerán en el estado Hi-Z. En t_3 se aplican los datos en las terminales de E/S y en t_4 se escriben en la ubicación de dirección en el flanco positivo de \overline{WE} . En t_5 se extraen los datos. En realidad, los datos se fijan primero (en el flanco positivo de \overline{WE}) en una memoria de búfer de FF que forma parte de los circuitos del 2864. Los datos se retienen ahí mientras otro circuito en el chip realiza una operación de borrado en la ubicación de dirección seleccionada en el arreglo de la EEPROM, después de lo cual el byte de datos se transfiere del búfer al arreglo de la EEPROM y se almacena en esa ubicación. Esta operación de borrado y almacenado dura aproximadamente 5 ms. Cuando \overline{CE} regresa al nivel ALTO en t_4 , el chip está nuevamente en el modo de espera mientras se completan las operaciones internas de borrado y almacenado.

La 2864 tiene un modo de escritura mejorado, en el cual se permite al usuario escribir hasta 16 bytes de datos en la memoria del búfer de FF, en donde se retienen mientras el circuito de la EEPROM borra las ubicaciones de dirección seleccionadas. Después, los 16 bytes de datos se transfieren al arreglo de la EEPROM para almacenarlos en esas ubicaciones. Este proceso dura también alrededor de 5 ms.

Debido a que el proceso interno de almacenar un valor de datos en una EEPROM es bastante lento, la velocidad de la operación de transferencia de datos también puede ser más lenta. En consecuencia, muchos fabricantes ofrecen dispositivos EEPROM en encapsulados de ocho terminales que se integran a un bus *serial* de dos o tres alambres. Esto ahorra espacio físico en el tablero del sistema, lo contrario a utilizar una 2864 en un encapsulado DIP amplio de 28 terminales. También simplifica la interfase de hardware entre la CPU y la EEPROM.

CD-ROM

El disco compacto (CD) es un tipo muy prominente de almacenamiento de sólo lectura que se utiliza en la actualidad en los sistemas computacionales. La tecnología de disco y el hardware necesario para recuperar la información son los mismos que se utilizan en los sistemas de audio. Sólo el formato de los datos es distinto. Los discos se fabrican con una superficie altamente reflectiva. Para almacenar datos en los discos, un rayo láser muy intenso se enfoca en un punto *muy* pequeño del disco. Este rayo quema un pozo difractor de luz en ese punto de la superficie del disco. Los datos digitales (1s y 0s) se almacenan en el disco un bit a la vez, mediante el proceso de quemar o no quemar un pozo en el recubrimiento reflectivo. La información digital se ordena en el disco como una espiral continua de puntos de datos. La precisión del rayo láser permite almacenar muy grandes cantidades de datos (más de 550 Mbytes) en un pequeño disco de 120 mm.

Para poder leer los datos se enfoca un rayo láser mucho menos poderoso, que el utilizado para almacenar, en la superficie del disco. En cualquier punto, la luz reflejada se detecta ya sea como un 1 o como un 0. Este sistema óptico está montado en un transporte mecánico que se mueve hacia delante y hacia atrás a lo largo del radio del disco, siguiendo la espiral de datos a medida que el disco gira. Los datos que se recuperan del sistema óptico llegan un bit a la vez en un flujo de datos en serie. La rotación angular del disco se controla para mantener una velocidad constante de puntos de datos entrantes. Si el disco se va a utilizar para grabación de audio, este flujo de datos se convierte en una forma de onda analógica. Si el disco se va a utilizar como ROM, los datos se decodifican en bytes en paralelo que la computadora

pueda utilizar. La tecnología del reproductor de CD, aunque muy sofisticada, tiene un bajo costo relativo y se está convirtiendo en la manera estándar de cargar grandes cantidades de datos en una computadora personal. Las principales mejoras que están ocurriendo ahora en la tecnología del CD-ROM incluyen un tiempo de acceso más rápido al recuperar los datos.

PREGUNTAS DE REPASO

1. *Verdadero o falso:* una MROM puede programarse por el usuario.
2. ¿Qué diferencia tiene una PROM en comparación con una MROM? ¿Puede borrarse y reprogramarse?
3. *Verdadero o falso:* una PROM almacena un 1 lógico cuando su enlace de fusible está intacto.
4. ¿Cómo se borra una EPROM?
5. *Verdadero o falso:* no hay forma de borrar sólo una porción de una memoria EPROM.
6. ¿Qué función realizan los programadores de PROM y de EPROM?
7. ¿Qué desventajas de las EPROMs solucionan las EEPROMs?
8. ¿Cuáles son las principales desventajas de la EEPROM?
9. ¿Qué tipo de ROM puede borrar un byte a la vez?
10. ¿Cuántos bits se leen de un disco CD-ROM en cualquier punto en el tiempo?

12-8 MEMORIA FLASH

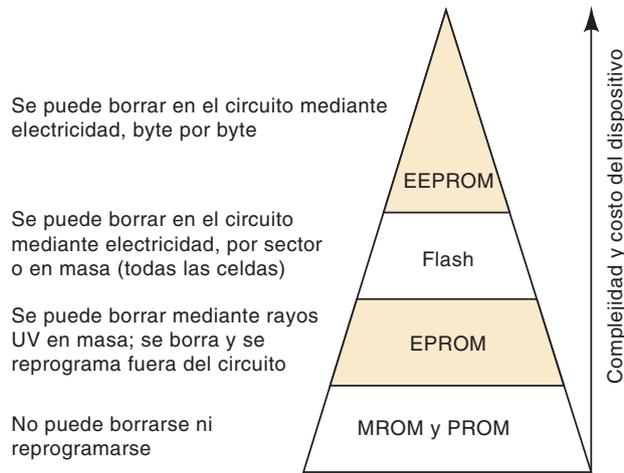
Las EPROMs son no volátiles, ofrecen tiempos de acceso de lectura rápidos (por lo general de 120 ns) y tienen una alta densidad y un bajo costo por bit. No obstante, hay que extraerlas de su circuito/sistema para borrarlas y reprogramarlas. Las EEPROMs son no volátiles, ofrecen un acceso de lectura rápido y permiten borrar y reprogramar bytes individuales con rapidez y dentro del circuito. Sufren de una menor densidad y un costo mucho más alto que las EPROMs.

El reto para los ingenieros de semiconductores era fabricar una memoria no volátil con la capacidad de la EEPROM de borrarse mediante electricidad y dentro del circuito, pero con densidades y costos mucho más cercanos a los de las EPROMs, al tiempo que se mantuviera el acceso de lectura de alta velocidad de ambos tipos de memorias. La respuesta a este reto fue la **memoria Flash**.

En su estructura, la celda de una memoria Flash es parecida a la celda simple de un solo transistor de la EPROM (y distinta de la celda más compleja de dos transistores de la EEPROM), sólo que es un poco más grande. Tiene una capa de óxido más delgada en la compuerta, la cual le permite la capacidad de borrarse mediante electricidad, pero puede construirse con densidades mucho más altas que las EEPROMs. El costo de la memoria Flash es mucho menor que el de la EEPROM. La figura 12-14 ilustra las ventajas y desventajas de las diversas memorias semiconductoras no volátiles. A medida que aumenta la flexibilidad en el borrado/la programación (de la base al vértice del triángulo), también aumentan la complejidad y el costo. Las memorias MROM y PROM son los dispositivos más económicos y simples, pero no pueden borrarse y reprogramarse. La EEPROM es la más compleja y costosa debido a que puede borrarse y reprogramarse en un circuito, byte por byte.

Las memorias Flash se llaman así debido a sus tiempos rápidos de borrado y escritura. La mayoría de los chips Flash utilizan una operación de *borrado en masa*, en el cual todas las celdas en el chip se borran en forma simultánea; este proceso de borrado en masa requiere, por lo general, de cientos de milisegundos, en comparación con los 20 minutos para las EPROMs de UV. Algunas memorias Flash más recientes ofrecen un modo de *borrado por sector*, en donde pueden borrarse sectores específicos del arreglo de memoria (por ejemplo, 512 bytes) uno a la vez. Esto evita tener que borrar y reprogramar todas las celdas cuando sólo se necesita actualizar

FIGURA 12-14 Las ventajas y desventajas de las memorias semiconductoras no volátiles muestran que la complejidad y el costo aumentan a medida que se incrementa la flexibilidad en el borrado y la programación.



una parte de la memoria. Una memoria Flash común tiene un tiempo de escritura de 10 μ s por byte, en comparación con los 100 μ s para la EPROM más avanzada y 5 ms para la EEPROM (que incluye el tiempo de borrado de bytes automático).

El CI de memoria Flash CMOS 28F256A

La figura 12-15(a) muestra el símbolo lógico para el circuito de memoria Flash CMOS 28F256A de Intel Corporation, el cual tiene una capacidad de 32K \times 8. El diagrama muestra 15 entradas de dirección ($A_0 - A_{14}$) necesarias para seleccionar las distintas direcciones de memoria; es decir, $2^{15} = 32K = 32,768$. Las ocho terminales de entrada/salida de datos ($DQ_0 - DQ_7$) se utilizan como entradas durante las operaciones de escritura de memoria y como salidas durante las operaciones de lectura de memoria. Estas terminales de datos flotan en el estado Hi-Z cuando el chip se deselecciona ($\overline{CE} = \text{ALTO}$) o cuando las salidas están deshabilitadas ($\overline{OE} = \text{ALTO}$). La entrada de habilitación de escritura (\overline{WE}) se utiliza para controlar las operaciones de escritura en memoria. Observe que el chip requiere dos voltajes de fuente de energía: V_{CC} es el voltaje estándar de +5 V que se utiliza para los circuitos lógicos; V_{PP} es el voltaje de la fuente de energía para borrar/programar, cuyo valor

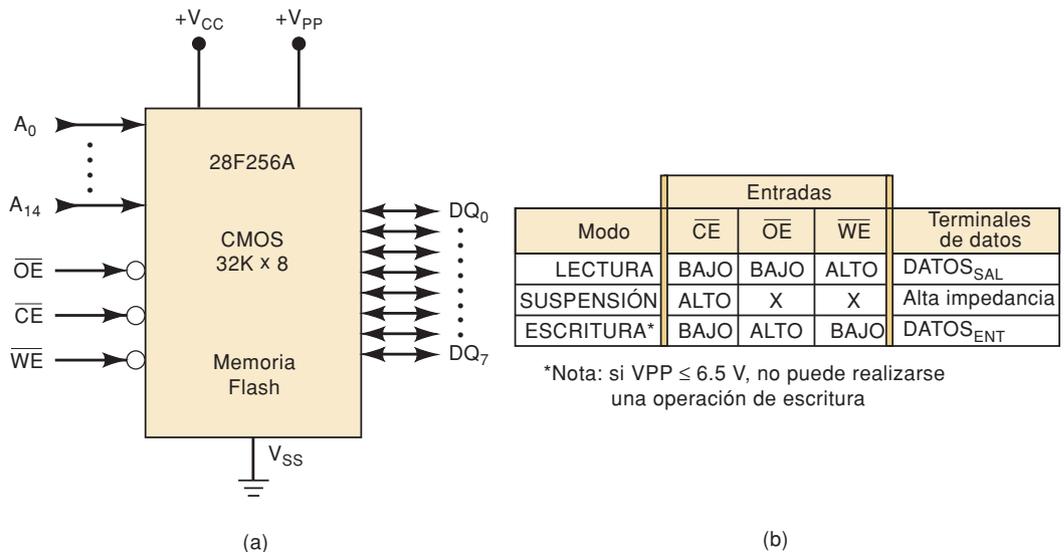


FIGURA 12-15 (a) Símbolo lógico para el chip de memoria Flash 28F256A; (b) las entradas de control \overline{CE} , \overline{WE} y \overline{OE} .

nominal es de +12V, el cual se necesita para las operaciones de borrado y programación (escritura). Los chips Flash más nuevos generan el V_{PP} de manera interna y sólo requieren una fuente de energía. Los dispositivos de bajo voltaje más recientes operan sólo con 1.8 V.

Las entradas de control (\overline{CE} , \overline{OE} y \overline{WE}) controlan lo que ocurre en las terminales de datos, en forma muy parecida a la de la EEPROM 2864, como muestra la tabla de la figura 12-15(b). Por lo general estas terminales de datos se conectan a un bus de datos. Durante una operación de escritura, los datos se transfieren a través del bus (por lo general desde el microprocesador) y hacia el chip. Durante una operación de lectura, los datos en el interior del chip se transfieren a través del bus de datos (por lo general hacia el microprocesador).

Podremos comprender mejor la operación de este chip de memoria Flash si analizamos su estructura interna. La figura 12-16 es un diagrama del 28F256A en el que se muestran sus principales bloques funcionales. Deberá remitirse a este diagrama según lo necesite durante el siguiente análisis. La característica única de esta estructura es el *registro de instrucciones*, el cual se utiliza para administrar todas las funciones del chip. Los códigos de instrucción se escriben en este registro para controlar qué operaciones se deben llevar a cabo dentro del chip (por ejemplo, borrar, borrar-verificar, programar, programar-verificar). Por lo general, estos códigos de instrucción llegan a través del bus de datos desde el microprocesador. La lógica de control de estado examina el contenido del registro de instrucciones y genera señales de lógica y control para que el resto de los circuitos del chip lleve a cabo los pasos en la operación. Algunos ejemplos de los tipos de instrucciones que pueden enviarse a la memoria Flash se muestran aquí para que se dé una idea de por qué son necesarios. Cada instrucción se almacena en el registro mediante el uso del mismo ciclo de escritura que se describió para la EEPROM en la figura 12-13(c).

Instrucción leer. Al escribir un código de 00 hexadecimal en el registro de instrucciones se prepara el CI de memoria para la operación de lectura. Después de esto puede usarse un ciclo normal de lectura para acceder a los datos almacenados en cualquier dirección.

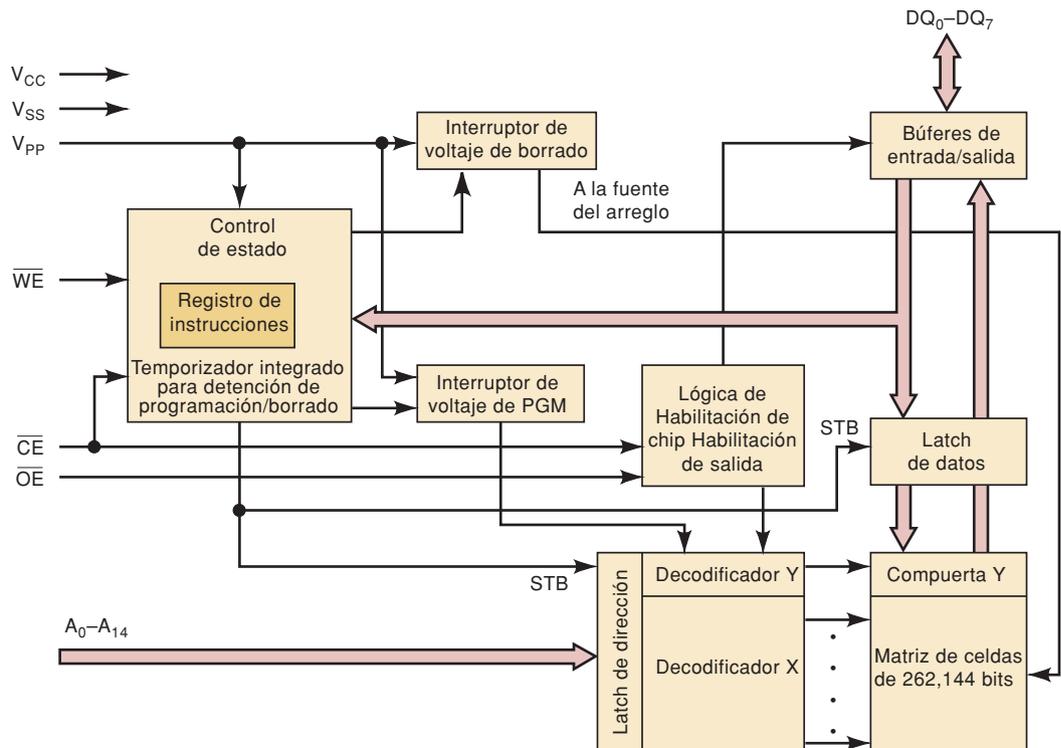


FIGURA 12-16 Diagrama funcional del chip de memoria Flash 28F256A. (Cortesía de Intel Corporation.)

Instrucción establecer-borrar/borrar. Debe escribirse el código de 20 hexadecimal en el registro de instrucciones dos veces en una fila para comenzar la secuencia interna de borrado.

Instrucción verificar borrado. Esta instrucción (FF hexadecimal) hace que el CI de memoria compruebe todas sus ubicaciones de memoria para verificar que todos los bits estén en ALTO.

Instrucción establecer programar/programar. Esta instrucción (40 hexadecimal) coloca el CI de memoria en un modo que permite que los ciclos de escritura subsiguientes almacenen datos en una dirección especificada, un byte a la vez.

Instrucción programar-verificar. Esta instrucción (C0 hexadecimal) se utiliza para verificar que se hayan almacenado los datos correctos en la ROM Flash. Después de escribir este código en el registro de instrucciones, la siguiente operación de lectura producirá el contenido de la última ubicación en la que se escribió, y estos datos pueden compararse con el valor deseado.

Memoria Flash mejorada

La arquitectura básica de la memoria Flash en la actualidad y el conjunto básico de códigos de instrucción son muy similares a los de los dispositivos de primera generación. Los dispositivos Flash más recientes tienen nuevas características y nuevos códigos de instrucción para controlarlas, además de las comunes para los dispositivos anteriores. Desde luego que los dispositivos Flash más recientes tienen mucha más capacidad, operan con mucho menos energía (y a menores voltajes), vienen en encapsulados más pequeños y tienen un costo mucho menor por bit que sus antecesores. También ofrecen características tales como la habilidad de leer/escribir datos mientras que se borra un bloque de memoria. El voltaje de programación V_{PP} se genera en forma interna, lo cual permite el uso de una sola fuente de energía. La velocidad de su operación puede mejorarse mediante el uso de un modo ráfaga. Esto tan sólo significa que se puede acceder a varias direcciones en una fila con mucha rapidez, con lo cual se proporciona una ráfaga de transferencia de datos. Se proporciona una entrada de reloj síncrona para controlar la operación de ráfaga. Una dirección base se fija en la memoria y después el contenido de esta ubicación se transfiere durante el flanco del reloj; también se incrementa la dirección a la siguiente ubicación. De esta forma se puede acceder a varias ubicaciones de memoria secuenciales a una velocidad tan rápida como la velocidad con la que puede oscilar el reloj del sistema, sin la sobrecarga de generar cada dirección. Todas estas características han hecho de la memoria Flash la tecnología predominante de memoria no volátil de estado sólido en uso hoy en día.

PREGUNTAS DE REPASO

1. ¿Cuál es la principal ventaja de la memoria Flash en comparación con las EPROMs?
2. ¿Cuál es la principal ventaja de la memoria Flash en comparación con las EEPROMs?
3. ¿De dónde proviene la palabra *flash*?
4. ¿Para qué se necesita el voltaje V_{PP} ?
5. ¿Cuál es la función del registro de instrucciones del 28F256A?
6. ¿Cuál es el propósito de la instrucción borrar-verificar?
7. ¿Cuál es el propósito de la instrucción programar-verificar?

12-9 APLICACIONES DE LA ROM

Con la excepción de MROM y PROM, la mayoría de los dispositivos ROM pueden reprogramarse, por lo que técnicamente no son memorias de *sólo lectura*. Sin embargo,

todavía puede usarse el término *ROM* para incluir EPROMs, EEPROMs y memoria Flash debido a que, durante la operación normal, el contenido almacenado de estos dispositivos no cambia con tanta frecuencia como se lee. Por lo tanto, se considera que las ROMs incluyen a todos los dispositivos de memoria semiconductora no volátil, y que se utilizan en aplicaciones en donde se requiere el almacenamiento no volátil de información, datos o códigos de programa, y en donde los datos almacenados cambian pocas veces o incluso nunca. He aquí algunas de las áreas de aplicación más comunes.

Memoria de programa de microcontrolador embebido

Los microcontroladores prevalecen en la mayoría de los productos electrónicos para el consumidor que hay en el mercado en la actualidad. El sistema de frenos automáticos de su automóvil y el controlador del motor, su teléfono celular, su cámara digital, su horno de microondas y muchos otros productos tienen un microcontrolador como cerebro. Estas pequeñas computadoras tienen sus instrucciones de programa almacenadas en memoria no volátil (en otras palabras, en una ROM). La mayoría de los microcontroladores embebidos en la actualidad tienen una ROM Flash integrada en el mismo CI que la CPU. Muchos también tienen un área de EEPROM que ofrece las características de borrado de bytes y el almacenamiento no volátil.

Transferencia de datos y portabilidad

La necesidad de almacenar y transferir grandes conjuntos de información binaria es un requerimiento de muchos sistemas de bajo consumo operados por baterías hoy en día. Los teléfonos celulares almacenan fotografías y clips de video. Las cámaras digitales almacenan muchas imágenes en medios de memoria removibles. Las unidades Flash se conectan en el puerto USB de una computadora y almacenan gigabytes de información. Su reproductor de MP3 está cargado con música y opera todo el día con baterías. Un PDA (asistente digital personal) almacena información sobre citas, e-mail, direcciones e incluso hasta libros completos. Todos estos aparatos electrónicos personales comunes requieren el almacenamiento de baja energía, bajo costo y alta densidad con una capacidad de escritura en el circuito que está disponible en la memoria Flash.

Memoria de arranque

Muchas microcomputadoras y la mayoría de las computadoras más grandes no tienen sus programas de sistema operativo almacenados en ROM, sino que estos programas se almacenan en la memoria en masa externa, por lo general en disco magnético. ¿Cómo entonces saben estas computadoras lo que deben hacer cuando se encienden? Un programa relativamente pequeño, conocido como **programa de arranque**, se almacena en ROM. Cuando la computadora se enciende, ejecuta las instrucciones que están en este programa de arranque. Por lo general, estas instrucciones provocan que la CPU inicialice el hardware del sistema. Después este programa de arranque carga los programas del sistema operativo desde el dispositivo de almacenamiento masivo (disco) hacia su memoria principal interna. En ese punto la computadora comienza a ejecutar el programa del sistema operativo y está lista para responder a las instrucciones del usuario. A este proceso inicial se le conoce como “arrancar el sistema”.

Muchos de los chips de procesamiento de señales digitales cargan su memoria de programa interna mediante una ROM de arranque externa cuando se les aplica energía. Algunos de los PLDs más avanzados también cargan la información de programación que configura sus circuitos lógicos desde una ROM externa, hacia un área de RAM dentro del PLD. Así, para reprogramar el PLD se cambia la ROM de arranque en vez de cambiar el propio chip PLD.

Tablas de datos

A menudo las ROMs se utilizan para almacenar tablas de datos que no cambian. Algunos ejemplos son las tablas trigonométricas (por ejemplo: seno, coseno, etcéte-

ra) y las tablas de conversión de código. El sistema digital puede utilizar estas tablas de datos para “buscar” el valor correcto. Por ejemplo, una ROM puede usarse para almacenar la función seno para ángulos desde 0° hasta 90° . Podría organizarse como de 128×8 con siete entradas de dirección y ocho salidas de datos. Las entradas de dirección representan el ángulo en incrementos de 0.7° aproximadamente. Por ejemplo, la dirección 0000000 es 0° , la dirección 0000001 es 0.7° , la dirección 0000010 es 1.41° , y así en lo sucesivo hasta la dirección 1111111, que es 89.3° . Cuando se aplica una dirección a la ROM, las salidas de datos representan el seno aproximado del ángulo. Por ejemplo, con la dirección de entrada 1000000 (que representa un valor aproximado a 45°) las salidas de datos serán 10110101. Como el seno es menor o igual que 1, estos datos se interpretan como una fracción, es decir, 0.10110101, que cuando se convierte en decimal es igual a 0.707 (el seno de 45°). Es imprescindible que el usuario de esta ROM comprenda el formato en el que se almacenan los datos.

Las ROMs estándar de tabla de búsqueda para funciones tales como estas estuvieron alguna vez disponibles como chips TTL. Sólo hay unas cuantas todavía en producción. Hoy en día, la mayoría de los sistemas que necesitan buscar valores equivalentes involucran a un microprocesador y los datos de la tabla de “búsqueda” se almacenan en la misma ROM que guarda las instrucciones del programa.

Convertidor de datos

El circuito convertidor de datos recibe datos expresados en un tipo de código y produce una salida expresada en otro tipo. Por ejemplo, la conversión de código se necesita cuando una computadora produce como salida datos en código binario directo y queremos convertirlo en BCD para poder visualizarlo en pantallas de LEDs de 7 segmentos.

Uno de los métodos más sencillos de conversión de código utiliza una ROM programada de tal forma que la aplicación de una dirección específica (el código anterior) produzca una salida de datos que represente el equivalente en el nuevo código. La 74185 es una ROM que almacena la conversión de código binario a BCD para una entrada binaria de seis bits. Para ilustrar esto, una entrada de dirección binaria de 100110 (38 decimal) producirá una salida de datos de 00111000, que es el código BCD para el 38 decimal.

Generador de funciones

El generador de funciones es un circuito que produce formas de onda tales como senoidales, dientes de sierra, triangulares y cuadradas. La figura 12-17 muestra cómo se utilizan una tabla de búsqueda de ROM y un DAC para generar una señal de salida de onda senoidal.

La ROM almacena 256 valores distintos de ocho bits, cada uno de los cuales corresponde a un valor distinto de la forma de onda (es decir, un punto de voltaje distinto en la onda senoidal). Al contador de ocho bits se le aplica un pulso continuo mediante una señal de reloj, para proporcionar entradas de dirección secuenciales a la ROM. A medida que el contador avanza en ciclo a través de las 256 direcciones distintas, la ROM produce como salida los 256 puntos de datos y los envía al DAC. La salida del DAC será una forma de onda que pase a través de los 256 valores de voltaje analógicos distintos, correspondientes a los puntos de datos. El filtro pasa-bajas suaviza los intervalos en la salida del DAC para producir una forma de onda uniforme.

FIGURA 12-17
Generador de funciones que utiliza una ROM y un DAC.

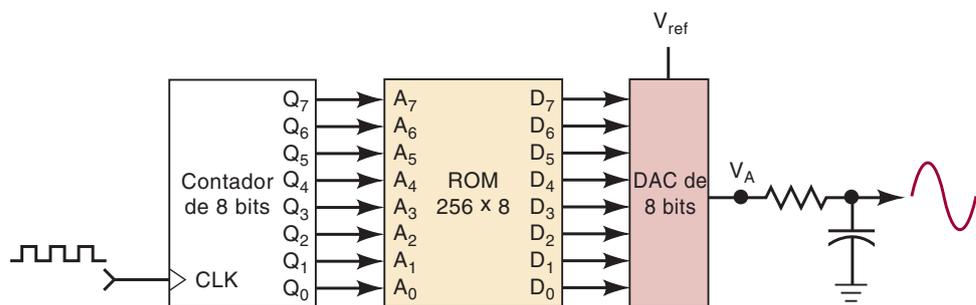
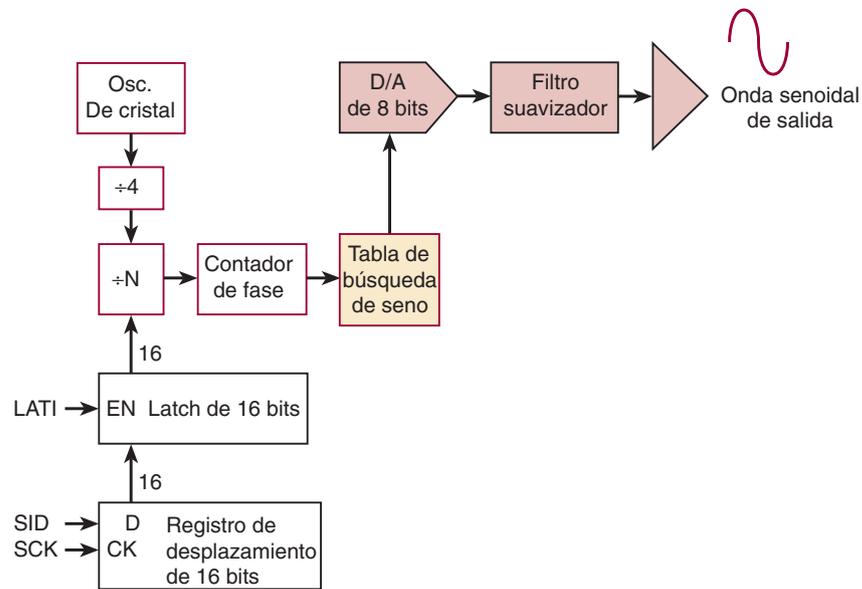


FIGURA 12-18 El generador de ondas senoidales programable ML2035. (Cortesía de MicroLinear Corp.)



Los circuitos como éste se utilizan en ciertos generadores de funciones comerciales. La misma idea se aplica en algunos sintetizadores de voz, en donde los valores de la forma de onda de voz digitalizada se almacenan en la ROM. El ML2035, que se muestra en la figura 12-18, es un chip generador de ondas senoidales programable, el cual incorpora esta estrategia básica para generar una onda senoidal de amplitud fija y una frecuencia que puede seleccionarse desde corriente directa hasta 50 kHz. El número que se desplaza en el registro de desplazamiento de 16 bits se utiliza para determinar la frecuencia de reloj para el contador que controla las entradas de dirección en la tabla de búsqueda de ROM. El ML2035 está diseñado para aplicaciones de telecomunicaciones que requieren la generación de tonos precisos de varias frecuencias.

Almacenamiento auxiliar

Debido a su no volatilidad, alta velocidad, bajos requerimientos de energía y ausencia de piezas móviles, los módulos de memoria Flash se han convertido en alternativas factibles para el almacenamiento en disco magnético. Esto se aplica en especial para las capacidades bajas (5 Mbytes o menos), en donde la memoria Flash puede competir en costo contra el disco magnético. El bajo consumo de energía de la memoria Flash la hace en especial atractiva para las computadoras portátiles que utilizan energía de una batería.

PREGUNTAS DE REPASO

1. Describa cómo utiliza una computadora un programa de arranque.
2. ¿Qué es un convertidor de código?
3. ¿Cuáles son los elementos principales de un generador de funciones?
4. ¿Por qué los módulos de memoria Flash son una alternativa factible para el almacenamiento auxiliar en disco?

12-10 RAM SEMICONDUCTORA

Recuerde que el término *RAM* significa *memoria de acceso aleatorio*, lo cual indica que se puede acceder con la misma facilidad a cualquier ubicación de dirección de memoria. Muchos tipos de memoria pueden clasificarse como de acceso aleatorio, pero

cuando se utiliza el término *RAM* con las memorias semiconductoras, por lo general, se considera como que se refiere a la memoria de lectura/escritura (RWM), a diferencia de la ROM. Como es una práctica común el usar RAM para identificar a la RWM semiconductor, mantendremos esta práctica a lo largo de las siguientes discusiones.

La RAM se utiliza en las computadoras para el almacenamiento *temporal* de programas y datos. Cuando la computadora ejecute un programa, se realizarán operaciones de lectura y de escritura sobre muchas ubicaciones de dirección de la RAM. Para ello se requieren tiempos de ciclo de lectura y de escritura rápidos para la RAM, de manera que no disminuya la velocidad de operación de la computadora.

La principal desventaja de la RAM es que es volátil y perderá toda la información almacenada si se interrumpe o se desconecta la energía. Sin embargo, algunas RAMs tipo CMOS utilizan cantidades tan pequeñas de energía en el modo de suspensión (sin que se realicen operaciones de lectura o de escritura) que pueden operar mediante baterías cuando se interrumpe la energía principal. Desde luego que la principal ventaja de la RAM es que se puede escribir en ella y leer de ella con la misma rapidez y facilidad.

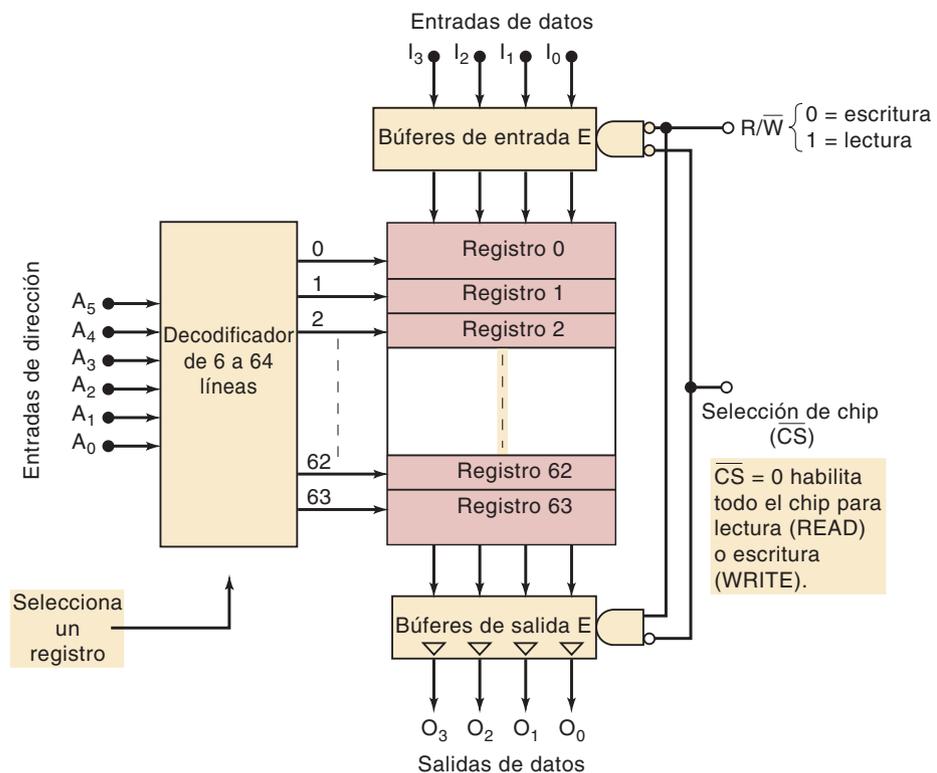
El siguiente análisis sobre la RAM se tratará parte del material que cubrimos en nuestro estudio sobre la ROM, ya que muchos de los conceptos básicos son comunes para ambos tipos de memorias.

12-11 ARQUITECTURA DE LA RAM

Al igual que con la ROM, es útil pensar en la RAM como si consistiera de un número de registros, cada uno de los cuales almacena una sola palabra de datos y tiene una dirección única. Por lo general las RAMs tienen capacidades de palabras de 1K, 4K, 8K, 16K, 64K, 128K, 256K y 1024K, con tamaños de palabra de uno, cuatro u ocho bits. Como veremos más adelante, la capacidad de palabras y el tamaño de palabra pueden expandirse mediante la combinación de chips de memoria.

La figura 12-19 muestra la arquitectura simplificada de una RAM que almacena 64 palabras de cuatro bits cada una (es decir, una memoria de 64×4). Estas pala-

FIGURA 12-19
Organización interna de
una RAM de 64×4 .



bras tienen direcciones que varían de 0 a 63_{10} . Para poder seleccionar una de las 64 ubicaciones de dirección para leer o escribir, se aplica un código de dirección binario a un circuito decodificador. Como $64 = 2^6$, el decodificador requiere un código de entrada de seis bits. Cada código de dirección activa una salida específica del decodificador, que a su vez habilita su registro correspondiente. Por ejemplo, suponga que el código de dirección que se aplica es:

$$A_5A_4A_3A_2A_1A_0 = 011010$$

Como $011010_2 = 26_{10}$, la salida 26 del decodificador cambiará a nivel ALTO y seleccionará el registro 26 para una operación de lectura o de escritura.

Operación de lectura

El código de dirección selecciona un registro en el chip de memoria para leer o escribir. Para poder *leer* el contenido del registro seleccionado, la entrada LEER/ESCRIBIR (R/\overline{W})* debe ser un 1. Además, la entrada SELECCIÓN DE CHIP (\overline{CS}) debe activarse (un 0 en este caso). La combinación de $R/\overline{W} = 1$ y $\overline{CS} = 0$ habilita los búferes de salida, de manera que aparecerá el contenido del registro seleccionado en las cuatro salidas de datos. $R/\overline{W} = 1$ también *deshabilita* los búferes de entrada, de manera que las entradas de datos no afecten a la memoria durante una operación de lectura.

Operación de escritura

Para escribir una nueva palabra de cuatro bits en el registro seleccionado se requiere que $R/\overline{W} = 0$ y $\overline{CS} = 0$. Esta combinación *habilita* los búferes de entrada, de manera que la palabra de cuatro bits que se aplica a las entradas de datos se cargue en el registro seleccionado. La condición $R/\overline{W} = 0$ también *deshabilita* los búferes de salida, los cuales son triestado y, por lo tanto, las salidas de datos se encontrarán en su estado Hi-Z durante una operación de escritura. Desde luego que la operación de escritura destruye la palabra que estaba almacenada antes en esa dirección.

Selección de chip

La mayoría de los chips de memoria tienen una o más entradas \overline{CS} , las cuales se utilizan para habilitar todo el chip o deshabilitarlo por completo. En el modo deshabilitado, todas las entradas y salidas de datos están deshabilitadas (Hi-Z), de manera que no puedan llevarse a cabo operaciones de lectura ni de escritura. En este modo, el contenido de la memoria no se ve afectado. La razón de tener entradas \overline{CS} le será clara cuando combinemos chips de memoria para obtener memorias más grandes. Observe que muchos fabricantes llaman a estas entradas HABILITACIÓN DE CHIP (\overline{CE}). Cuando las entradas \overline{CS} o \overline{CE} se encuentran en su estado activo, se dice que el chip de memoria está *seleccionado*; en caso contrario, se dice que está *deseleccionado*. Muchos CIs de memoria están diseñados para consumir mucho menos energía cuando están deseleccionados. En los sistemas de memoria extensos, para una operación de memoria dada se seleccionarán uno o más chips de memoria mientras todos los demás se deseleccionan. Más adelante veremos más detalles sobre este tema.

Terminales comunes de entrada/salida

Para poder conservar terminales en un paquete de CI, por lo general, los fabricantes combinan las funciones de entrada y salida de datos mediante el uso de terminales comunes de entrada/salida. La entrada R/\overline{W} controla la función de estas terminales de E/S. Durante una operación de lectura, las terminales de E/S actúan como salidas de datos que reproducen el contenido de la ubicación de dirección seleccionada.

* Algunos fabricantes utilizan el símbolo \overline{WE} (habilitación de escritura) o \overline{W} en vez de R/\overline{W} . En cualquier caso, la operación es la misma.

Durante una operación de escritura, las terminales de E/S actúan como entradas de datos en las que se aplicarán los datos que se van a escribir.

Para ver por qué ocurre esto, considere el chip de la figura 12-19. Con terminales de entrada y salida separadas, se requiere un total de 18 terminales (incluyendo tierra y fuente de energía). Con cuatro terminales comunes de E/S, sólo se requieren 14 terminales. El ahorro de terminales se vuelve más considerable para chips con un tamaño de palabra más grande.

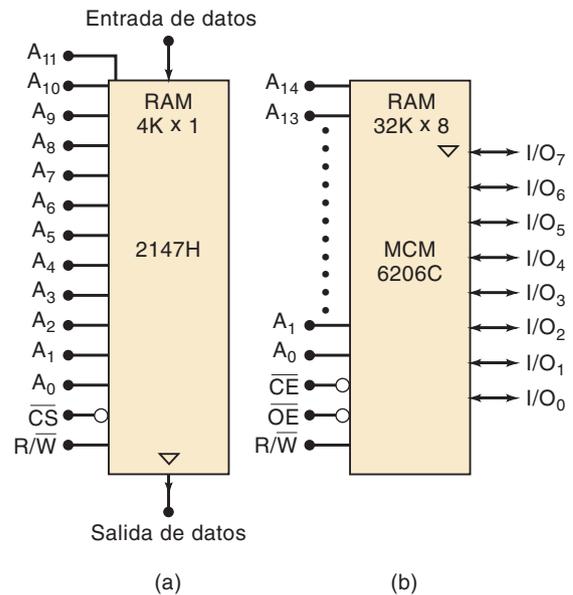
EJEMPLO 12-9

La 2147H es una RAM tipo NMOS, la cual se organiza como de $4K \times 1$, con entradas separadas de entrada y salida de datos, y una entrada de selección de chip activa en BAJO. Dibuje el símbolo lógico para este chip, en donde se muestren las funciones de todas las terminales.

Solución

El símbolo lógico se muestra en la figura 12-20(a).

FIGURA 12-20 Símbolos lógicos para (a) el chip de RAM 2147H; (b) la RAM MCM6206C.

**EJEMPLO 12-10**

El chip MCM6206C es una RAM tipo CMOS con capacidad de $32K \times 8$, terminales comunes de E/S, una señal de habilitación de chip activa en BAJO y una señal de habilitación de salida activa en BAJO. Dibuje el símbolo lógico.

Solución

El símbolo lógico se muestra en la figura 12-20(b).

En la mayoría de las aplicaciones se utilizan dispositivos de memoria con un bus de datos bidireccional, como el que vimos en el capítulo 9. Para este tipo de sistema, aún si el chip de memoria tuviera terminales separadas de entrada y salida, se conectarían entre sí en el mismo bus de datos. Una RAM que tiene terminales separadas de entrada y salida se denomina como RAM de puerto dual. Este tipo de memorias se utiliza en aplicaciones en las que la velocidad es muy importante y los datos entrantes provienen de un dispositivo distinto al que va a recibir los datos. Un buen ejemplo es la RAM de video en su PC. La tarjeta de video debe leer la RAM en forma repetida para regenerar la pantalla y llenarla en forma constante con la nueva información actualizada del bus del sistema.

PREGUNTAS DE REPASO

1. Describa las condiciones de entrada necesarias para leer una palabra desde una ubicación de dirección de RAM específica.
2. ¿Por qué algunos chips de RAM tienen terminales comunes de entrada/salida?
3. ¿Cuántas terminales se requieren para la RAM MCM6208C de $64K \times 4$, con una entrada CS y E/S común?

12-12 RAM ESTÁTICA (SRAM)

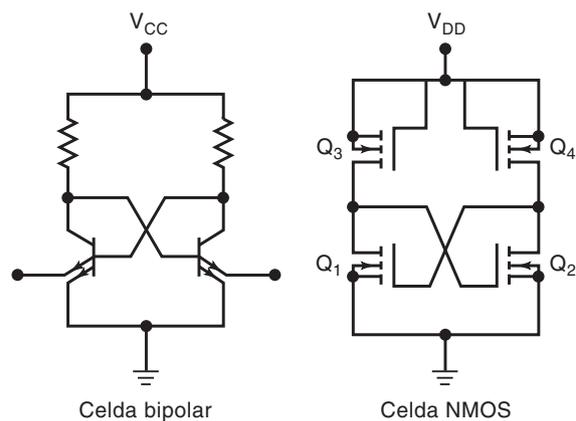
La operación de la RAM que hemos visto hasta este punto se aplica a una **RAM estática** (que puede almacenar datos mientras se aplique energía al chip). En esencia, las celdas de memoria de la RAM estática son flip-flops que permanecerán en un estado dado (almacenan un bit) de manera indefinida, siempre y cuando no se interrumpa la energía del circuito. En la sección 12-13 describiremos la **RAM dinámica**, que almacena datos en forma de cargas en los capacitores. Con las RAMs dinámicas, los datos almacenados desaparecerán en forma gradual debido a la descarga del capacitor, por lo que es necesario **regenerar** los datos en forma periódica (es decir, recargar los capacitores).

Las RAMs estáticas (SRAMs) están disponibles en las tecnologías bipolar, MOS y BiCMOS; la mayoría de las aplicaciones utilizan RAMs tipo NMOS o CMOS. Como dijimos antes, los dispositivos bipolares tienen la ventaja en velocidad (aunque CMOS está cada vez más cerca) y los dispositivos MOS tienen mucho mayor capacidad y un bajo consumo de energía. La figura 12-21 muestra para fines de comparación una celda de memoria estática bipolar común y una celda de memoria estática NMOS común. La celda bipolar contiene dos transistores bipolares y dos resistencias, mientras que la celda NMOS contiene cuatro MOSFETs de canal N. La celda bipolar requiere más área en el chip que la celda MOS, ya que un transistor bipolar es más complejo que un MOSFET y porque la celda bipolar requiere resistencias separadas, mientras que la celda MOS utiliza MOSFETs como resistencias (Q_3 y Q_4). Una celda de memoria CMOS sería similar a la celda NMOS, sólo que utilizaría MOSFETs de canal P en vez de Q_3 y Q_4 . Esto produce el consumo más bajo de energía pero incrementa la complejidad del chip.

Sincronización de la RAM estática

El uso más común de los CIs de RAM es como memoria interna de una computadora. La CPU (unidad central de proceso) realiza en forma continua operaciones de lectura y de escritura en esta memoria, a una velocidad bastante rápida, la cual se determina con base en las limitaciones de la CPU. Los chips de memoria que se integran

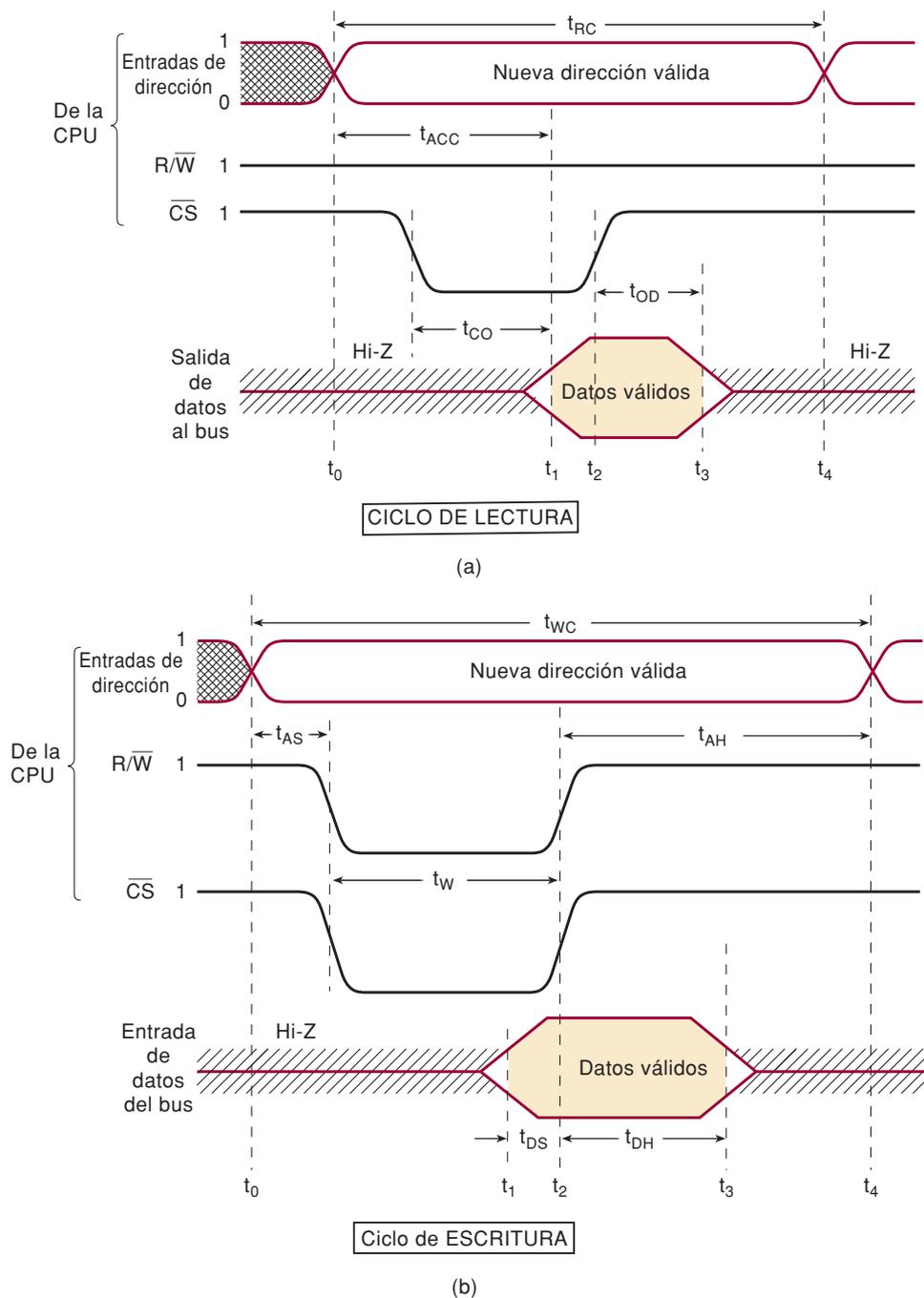
FIGURA 12-21 Celdas bipolares y NMOS de RAM estática típicas.



a la CPU deben ser lo bastante rápidos como para responder a las instrucciones de lectura y escritura de la CPU; un diseñador de computadoras debe tomar en cuenta las diversas características de sincronización de la RAM.

No todas las RAMs tienen las mismas características de sincronización, pero la mayoría de ellas son similares y, por lo tanto, utilizaremos un conjunto típico de características para fines ilustrativos. La nomenclatura para los distintos parámetros de sincronización variará de un fabricante a otro, pero casi siempre es fácil determinar el significado de cada parámetro con base en los diagramas de sincronización de la memoria en las hojas de datos de RAM. La figura 12-22 muestra los diagramas de sincronización para un ciclo completo de lectura y un ciclo completo de escritura en un chip de RAM ordinario.

FIGURA 12-22
Sincronización típica para la RAM estática:
(a) ciclo de lectura;
(b) ciclo de escritura.



Ciclo de lectura

Los diagramas de tiempo de la figura 12-22(a) muestran cómo las entradas de dirección, R/\overline{W} y selección de chip se comportan durante un ciclo de lectura de memoria. Como se indica, la CPU suministra estas señales de entrada a la RAM cuando desea leer datos de una ubicación de dirección de RAM específica. Aunque una RAM puede tener muchas entradas de dirección que provengan del bus de direcciones de la CPU, por cuestión de claridad en el diagrama sólo se muestran dos. También se muestra la salida de datos de la RAM; vamos a suponer que esta RAM en particular tiene una salida de datos. Recuerde que la salida de datos de la RAM se conecta al bus de datos de la CPU (figura 12-5).

El ciclo de lectura comienza en el tiempo t_0 . Antes de ese tiempo, las entradas de dirección pueden tener cualquier valor de dirección que esté en el bus de dirección debido a la operación anterior. Como la señal de selección de chip de la RAM no está activa, no responderá a su dirección “antigua”. Observe que la línea R/\overline{W} está en ALTO antes de t_0 y permanece en ALTO durante todo el ciclo de lectura. Por lo general, en la mayoría de los sistemas de memoria la línea R/\overline{W} se mantiene en el estado ALTO, sólo cuando se lleva a BAJO durante un ciclo de escritura. La salida de datos de la RAM se encuentra en su estado Hi-Z, debido a que $\overline{CS} = 1$.

En t_0 , la CPU aplica una nueva dirección a las entradas de la RAM; ésta es la dirección de la ubicación que se va a leer. Después de dejar pasar un tiempo para que las señales de dirección se establezcan, se activa la línea \overline{CS} . Como respuesta, la RAM coloca los datos de la ubicación direccionada en la línea de salida de datos en t_1 . El tiempo entre t_0 y t_1 es el tiempo de acceso de la RAM, t_{ACC} , y representa el tiempo que transcurre entre la aplicación de la nueva dirección y la aparición de los datos de salida válidos. El parámetro de sincronización t_{CO} es el tiempo que tarda la salida de la RAM en cambiar de Hi-Z a un nivel de datos válido, una vez que se activa \overline{CS} .

En el tiempo t_2 la señal \overline{CS} regresa al nivel ALTO y la salida de la RAM regresa a su estado Hi-Z después de un intervalo de tiempo, t_{OD} . Así, los datos de la RAM se encontrarán en el bus de datos entre t_1 y t_3 . La CPU puede recibir la información que provienen del bus de datos en cualquier punto durante este intervalo. En la mayoría de las computadoras, la CPU utilizará la PGT e la señal \overline{CS} en t_2 para fijar estos datos en uno de sus registros internos.

El tiempo completo del ciclo de lectura (t_{RC}) se extiende desde t_0 a t_4 , cuando la CPU cambia las entradas de dirección por una dirección distinta para el siguiente ciclo de lectura o de escritura.

Ciclo de escritura

La figura 12-22(b) muestra la actividad de las señales para un ciclo de escritura que comienza cuando la CPU suministra una nueva dirección a la RAM, en el tiempo t_0 . La CPU lleva las líneas R/\overline{W} y \overline{CS} a BAJO después de esperar durante un intervalo de tiempo t_{AS} , al cual se le conoce como *tiempo de establecimiento de dirección*. Esto da tiempo a los decodificadores de dirección de la RAM de responder a la nueva dirección. Las señales R/\overline{W} y \overline{CS} se mantienen en BAJO durante un intervalo de tiempo t_W , conocido como el intervalo de tiempo de escritura.

Durante este intervalo de escritura, en el tiempo t_1 la CPU aplica datos válidos al bus de datos para escribirlos en la RAM. Estos datos deben retenerse en la entrada de la RAM durante por lo menos un intervalo de tiempo t_{DS} antes de, y por lo menos un intervalo de tiempo t_{DF} después de la desactivación de R/\overline{W} y \overline{CS} en t_2 . Al intervalo t_{DS} se le conoce como *tiempo de establecimiento de datos* y a t_{DH} se le conoce como *tiempo de retención de datos*. De manera similar, las entradas de dirección deben permanecer estables durante el intervalo de tiempo de retención de la dirección t_{AH} , después de t_2 . Si no se cumple con cualquiera de estos requerimientos de tiempo de establecimiento o de tiempo de retención, la operación de escritura no se realizará en forma confiable.

El tiempo del ciclo de escritura completo (t_{WC}) se extiende desde t_0 hasta t_4 , cuando la CPU cambia las líneas de dirección a una nueva dirección para el siguiente ciclo de escritura o lectura.

El tiempo del ciclo de lectura (t_{RC}) y el tiempo del ciclo de escritura (t_{WC}) son los que en esencia determinan qué tan rápido puede operar un chip de memoria. Por ejemplo, en una aplicación real la CPU estará leyendo con frecuencia palabras de datos sucesivas de memoria, una después de la otra. Si la memoria tiene un t_{RC} de 50 ns, la CPU puede leer una palabra cada 50 ns, o 20 millones de palabras por segundo; si $t_{RC} = 10$ ns, la CPU puede leer 100 millones de palabras por segundo. La tabla 12-2 muestra los tiempos mínimos para el ciclo de lectura y el ciclo de escritura de ciertos chips de RAM estática representativos.

TABLA 12-2

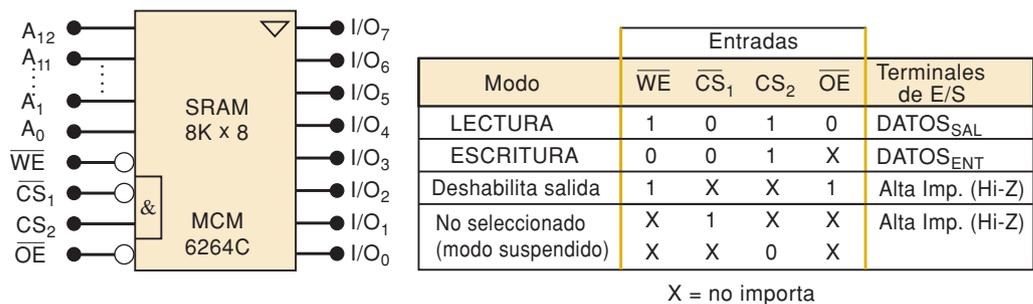
Dispositivo	t_{RC} (mín) (ns)	t_{WC} (mín) (ns)
CMOS MCM6206C, 32K × 8	15	15
NMOS 2147H, 4K × 1	35	35
BiCMOS MCM6708A, 64K × 4	8	8

Chip de SRAM real

La RAM MCM6264C CMOS de 8K × 8 es un ejemplo de un CI de SRAM real, con tiempos de ciclo de lectura y ciclo de escritura de 12 ns y un consumo de energía en suspensión de sólo 100 mW. En la figura 12-23 se muestra el símbolo lógico para este CI. Observe que tiene 13 entradas de dirección, ya que $2^{13} = 8192 = 8K$ y ocho líneas de E/S de datos. Las cuatro entradas de control determinan el modo de operación del dispositivo, de acuerdo con la tabla de modos que lo acompaña.

La entrada \overline{WE} es la misma que la entrada R/\overline{W} que hemos estado utilizando. Un nivel BAJO en \overline{WE} hará que se escriban datos en la RAM, siempre y cuando esté seleccionado el dispositivo (que ambas entradas de selección de chip estén activas). Observe que se utiliza el símbolo “&” para denotar que *ambas* deben estar activas. Un nivel ALTO en \overline{WE} producirá la operación de lectura, siempre y cuando esté seleccionado el dispositivo y los búferes de salida se habiliten mediante $\overline{OE} =$ BAJO. Cuando está deseleccionado, el dispositivo se encuentra en su modo de bajo consumo de energía y ninguna de las otras entradas tiene efecto.

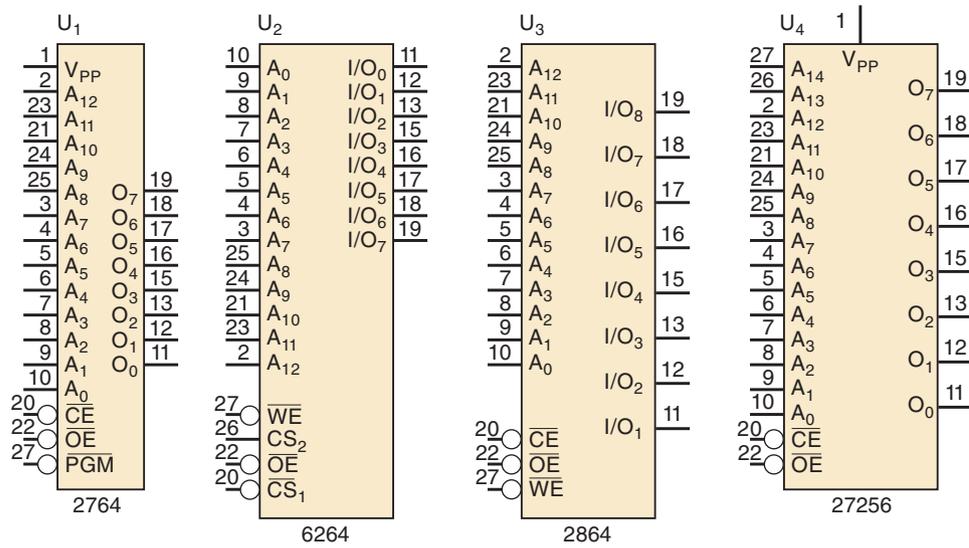
FIGURA 12-23 Símbolo y tabla de modos para el MCM6264C tipo CMOS.



La mayoría de los dispositivos que hemos visto en este capítulo están disponibles de parte de varios fabricantes distintos. Cada fabricante puede ofrecer dispositivos distintos de la misma dimensión (por ejemplo, 32K × 8) pero con distintas especificaciones o características. También hay varios tipos de encapsulados disponibles, como DIP, PLCC y varias formas de ala de gaviota y montaje en superficie.

A medida que analice los diversos dispositivos de memoria que hemos descrito en este capítulo, observará ciertas similitudes. Por ejemplo, analice los chips en la figura 12-24 y tome nota de las asignaciones de las terminales. El hecho de que se asigne la misma función a las mismas terminales en todos estos diversos dispositivos, fabricados

FIGURA 12-24
Encapsulados de memoria estándar de JEDEC.



por distintas compañías, no es coincidencia. Los estándares industriales creados por el Consejo Común de Ingeniería de Dispositivos Electrónicos (**JEDEC**) han originado la creación de dispositivos de memoria compatibles e intercambiables.

EJEMPLO 12-11

Un sistema se configura para un chip ROM de $8K \times 8$ (2764) y dos chips SRAM de $8K \times 8$ (6264). Todo el espacio de la ROM de $8K$ se utiliza para almacenar las instrucciones del microprocesador. Usted desea actualizar el sistema para que tenga cierta capacidad de almacenamiento de lectura/escritura no volátil. ¿Puede modificarse el circuito existente para dar cabida a las nuevas revisiones?

Solución

Sólo basta con sustituir un chip EEPROM 2864 en uno de los zócalos de RAM. La única diferencia funcional está en los requerimientos de un tiempo de ciclo de escritura mucho más largo de la EEPROM. Por lo general, esto se resuelve al modificar el programa de la microcomputadora que utiliza el dispositivo de memoria. Como no hay espacio libre en la ROM para estos cambios, necesitamos una ROM más grande. Una ROM de $32K \times 8$ (27C256) tiene básicamente la misma distribución de terminales que una 2764. Tan sólo tenemos que conectar dos líneas de dirección más (A_{13} y A_{14}) al zócalo de la ROM y sustituir el chip anterior con un chip 27C256.

Muchos sistemas de memoria aprovechan la versatilidad que proporcionan los estándares de JEDEC. Las terminales que son comunes para todos los dispositivos se cablean de forma fija a los buses del sistema. Las pocas terminales que son distintas entre los diversos dispositivos se conectan a circuitos que pueden modificarse con facilidad para configurar el sistema para el tamaño y tipo adecuado de dispositivo de memoria. Esto permite al usuario reconfigurar el hardware sin necesidad de cortar o soldar en el tablero. El circuito de configuración puede ser tan simple como unos puentes removibles o interruptores DIP configurables por el usuario, o tan complicados como un dispositivo lógico programable en el circuito que la computadora pueda configurar o modificar para cumplir con los requerimientos del sistema.

PREGUNTAS DE REPASO

1. ¿Qué diferencia hay entre la celda de una RAM estática y la celda de una RAM dinámica?
2. ¿Cuál tecnología de memoria utiliza, por lo general, la menor cantidad de energía?
3. ¿Qué dispositivo coloca datos en el bus de datos durante un ciclo de lectura?
4. ¿Qué dispositivo coloca datos en el bus de datos durante un ciclo de escritura?
5. ¿Qué parámetros de sincronización de la RAM determinan su velocidad de operación?
6. *Verdadero o falso:* un nivel BAJO en \overline{OE} habilitará los búferes de salida de un MCM6264C, siempre y cuando ambas entradas de selección de chip estén activas.
7. ¿Qué se debe hacer con la terminal 26 y la terminal 27 si se sustituye un 27256 por un 2764?

12-13 RAM DINÁMICA (DRAM)

Las RAMs dinámicas se fabrican mediante el uso de tecnología MOS y se distinguen por su alta capacidad, bajo requerimiento de energía y velocidad moderada de operación. Como dijimos antes, a diferencia de las RAMs estáticas que almacenan información en FFs, las RAMs dinámicas almacenan 1s y 0s en forma de cargas en un pequeño capacitor MOS (por lo general, de unos cuantos pico-Farads). Debido a la tendencia de estas cargas de fugarse después de un periodo de tiempo, las RAMs dinámicas requieren una recarga periódica de las celdas de memoria; a esto se le conoce como regenerar la RAM dinámica. En los chips DRAM modernos, cada celda de memoria debe regenerarse cada 2, 4 u 8 ms, o se perderán sus datos.

La necesidad de regenerarse es una desventaja de la RAM dinámica en comparación con la RAM estática, debido a que puede requerir de circuitos de soporte externos. Algunos chips de DRAM tienen circuitos de control de regeneración integrados, los cuales no requieren hardware externo adicional pero sí una sincronización especial de las señales de control de entrada del chip. Además y como veremos más adelante, las entradas de dirección para una DRAM deben manejarse de una manera menos directa que con la SRAM. En general, es más complejo diseñar circuitos con DRAM y utilizar este tipo de memoria en un sistema que la SRAM. No obstante, sus capacidades mucho mayores y su consumo de energía mucho menor hacen de las DRAMs la mejor opción para memoria en sistemas en donde las consideraciones de diseño más importantes son mantener un tamaño reducido, un bajo costo y bajo consumo de energía.

Para aplicaciones en las que la velocidad y la reducción en complejidad son más importantes que las consideraciones de costo, espacio y energía, las RAMs estáticas siguen siendo la mejor opción. Por lo general, son más veloces que las RAMs dinámicas y no requieren operación de regeneración. Es más fácil diseñar circuitos con RAMs estáticas, pero no pueden competir con la mayor capacidad y el bajo consumo de energía de las RAMs dinámicas.

Debido a su estructura de celdas simples, las DRAMs tienen cuatro veces la densidad de las SRAMs. Este aumento en densidad permite colocar cuatro veces más capacidad de memoria en una sola tarjeta. El costo por bit del almacenamiento en RAM dinámica es comúnmente de una quinta a una cuarta parte de las RAMs estáticas. Se obtiene un ahorro adicional en costo ya que los requerimientos de bajo consumo de energía de la RAM dinámica, que, por lo general, son de una sexta parte a la mitad de los de una RAM estática, permiten el uso de fuentes de energía más pequeñas y menos costosas.

Las principales aplicaciones de las SRAMs son en áreas en las que sólo se necesitan pequeñas cantidades de memoria o en donde se requiere una alta velocidad. Muchos instrumentos y aparatos controlados por microprocesadores tienen muy pocos requerimientos en cuanto a la capacidad de memoria. Algunos instrumentos

tales como los osciloscopios de almacenamiento digital y los analizadores lógicos, requieren de una memoria de muy alta velocidad. Para aplicaciones tales como estas se utiliza, por lo general, la SRAM.

La memoria interna principal de la mayoría de las microcomputadoras personales (por ejemplo, PCs basadas en Windows o Macs) utiliza DRAM debido a su alta capacidad y bajo consumo de energía. No obstante, estas computadoras utilizan algunas veces pequeñas cantidades de SRAM para las funciones que requieren de una máxima velocidad, como los gráficos de video, las tablas de búsqueda y la memoria caché.

PREGUNTAS DE REPASO

1. ¿Cuáles son las principales desventajas de la RAM dinámica, en comparación con la estática?
2. Liste las ventajas de la RAM dinámica en comparación con la RAM estática.
3. ¿Qué tipo de RAM esperarías encontrar en los módulos de memoria principal de su PC?

12-14 ESTRUCTURA Y OPERACIÓN DE LA RAM DINÁMICA

La arquitectura interna de la RAM dinámica puede visualizarse como un arreglo de celdas de un solo bit, como se muestra en la figura 12-25. Aquí se ordenan 16,384 celdas en un arreglo de 128×128 . Cada celda ocupa una posición única de fila y columna dentro del arreglo. Se necesitan catorce entradas de dirección para seleccionar una de las celdas ($2^{14} = 16,384$); los bits de dirección inferiores (A_0 - A_6) seleccionan la columna y los bits de mayor orden (A_7 - A_{13}) seleccionan la fila. Cada dirección de 14 bits selecciona una celda única en la que se va a escribir, o de la que se va a leer información. La estructura en la figura 12-25 es un chip DRAM de $16K \times 1$. Hoy en día los chips DRAM están disponibles en diversas configuraciones. Las DRAMs con un tamaño de palabra de cuatro bits (o mayor) tienen una distribución de celdas similar a la de la figura 12-25, sólo que cada posición en el arreglo contiene cuatro celdas y cada dirección que se aplica selecciona un grupo de cuatro celdas para una operación de lectura o de escritura. Como veremos más adelante, también pueden obtenerse tamaños de palabra más grandes si se combinan varios chips en el orden apropiado.

FIGURA 12-25
Distribución de celdas en una RAM dinámica de $16K \times 1$.

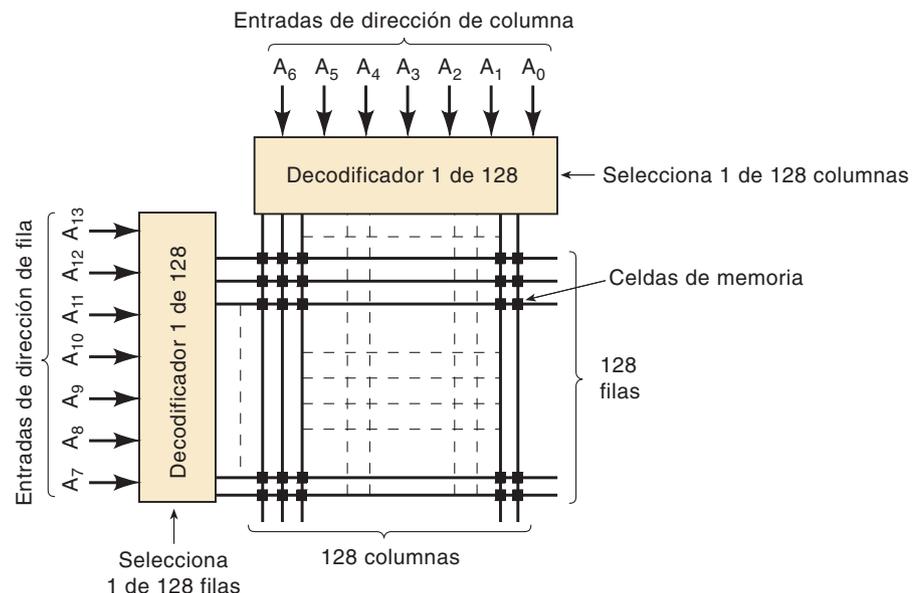
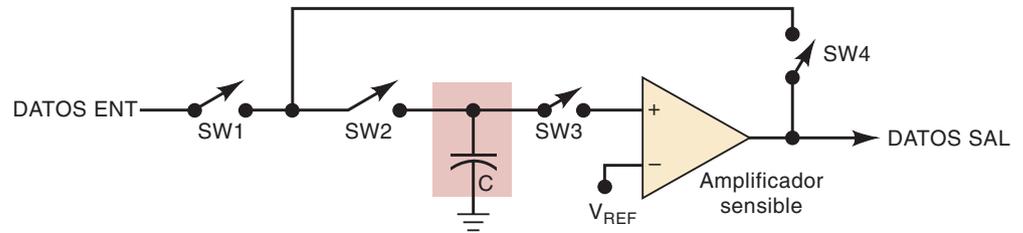


FIGURA 12-26

Representación simbólica de una celda de memoria dinámica. Durante una operación de ESCRITURA se cierran los interruptores semiconductores SW1 y SW2. Durante una operación de lectura se cierran todos los interruptores, excepto SW1.



La figura 12-26 es una representación simbólica de una celda de memoria dinámica y sus circuitos asociados. Muchos de los detalles del circuito no se muestran, pero podemos utilizar este diagrama simplificado para describir las ideas esenciales relacionadas con las operaciones de escritura y lectura en una DRAM. Los interruptores del SW1 al SW4 son en realidad MOSFETs que se controlan mediante varias salidas de un decodificador de direcciones y la señal R/\bar{W} . Desde luego que el capacitor es propiamente la celda de almacenamiento. Un amplificador sensible puede dar servicio a una columna completa de celdas de memoria, pero sólo opera en el bit de la fila seleccionada.

Para escribir datos en la celda, las señales de la decodificación de direcciones y la lógica de lectura/escritura cerrarán los interruptores SW1 y SW2, y mantendrán a SW3 y SW4 abiertos. Esto conecta a los datos de entrada con el capacitor C. Un 1 lógico en la entrada de datos carga a C y un 0 lógico lo descarga. Después los interruptores se abren para que C se desconecte del resto del circuito. En teoría, C retendría su carga por un tiempo indefinido, pero siempre hay una ruta de fuga a través de los interruptores apagados, de manera que C perderá su carga en forma gradual.

Para leer los datos de la celda se cierran los interruptores SW2, SW3 y SW4, y SW1 se mantiene abierto. Esto conecta el voltaje almacenado en el capacitor con el *amplificador sensible*. Este amplificador compara el voltaje con cierto valor de referencia para determinar si es un 0 o un 1 lógico y produce un valor sólido de 0 V o 5 V para la salida de datos. Esta salida de datos se conecta también a C (SW2 y SW4 están cerrados) y regenera el voltaje del capacitor al recargarlo o descargarlo. En otras palabras, el bit de datos en una celda de memoria se regenera cada vez que se lee.

Multiplexaje de direcciones

El arreglo de la DRAM de $16K \times 1$ que se muestra en la figura 12-25 es obsoleto y casi no está disponible en el mercado. Tiene 14 entradas de dirección; un arreglo de DRAM de $64K \times 1$ tendría 16 entradas de dirección. Una DRAM de $1M \times 4$ necesita 20 entradas de dirección; una de $4M \times 1$ necesita 22 entradas de dirección. Los chips de memoria de alta capacidad tales como éstos requerirían muchas terminales si cada entrada de dirección requiriera una terminal separada. Para poder reducir el número de terminales en sus chips DRAM de alta capacidad, los fabricantes utilizan el **multiplexaje de direcciones**, en el cual cada terminal de entrada de dirección puede alojar dos bits de dirección distintos. El ahorro en el número de terminales se traduce en una considerable disminución en el tamaño de los encapsulados de CI. Esto es muy importante en los tableros de memoria de gran capacidad, en donde se desea maximizar la cantidad de memoria que puede acomodarse en un tablero.

En las siguientes discusiones vamos a describir el orden en el cual se lleva a cabo el multiplexaje de direcciones en los chips DRAM. Hay que recalcar que en las DRAMs antiguas de pequeña capacidad la convención era presentar la dirección de menor orden especificando primero la fila, seguida de la dirección de mayor orden que especificaba la columna. Las DRAMs más recientes y los controladores que realizan el multiplexaje utilizan la convención opuesta de aplicar los bits de mayor orden como la dirección de fila y después los bits de menor orden como la dirección de columna. Nosotros describiremos la convención más reciente, pero usted deberá tener en cuenta este cambio cuando estudie los sistemas antiguos.

Utilizaremos la DRAM TMS44100 de $4M \times 1$ de Texas Instruments para ilustrar la operación de los chips DRAM en la actualidad. El diagrama de bloques funcional

DIAGRAMA DE BLOQUES FUNCIONAL

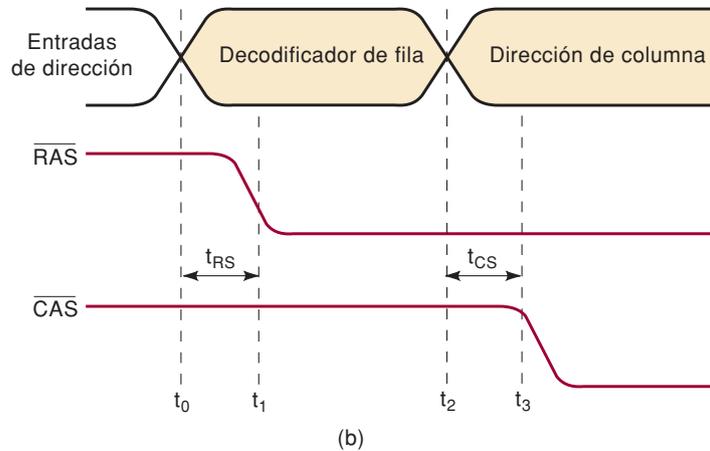
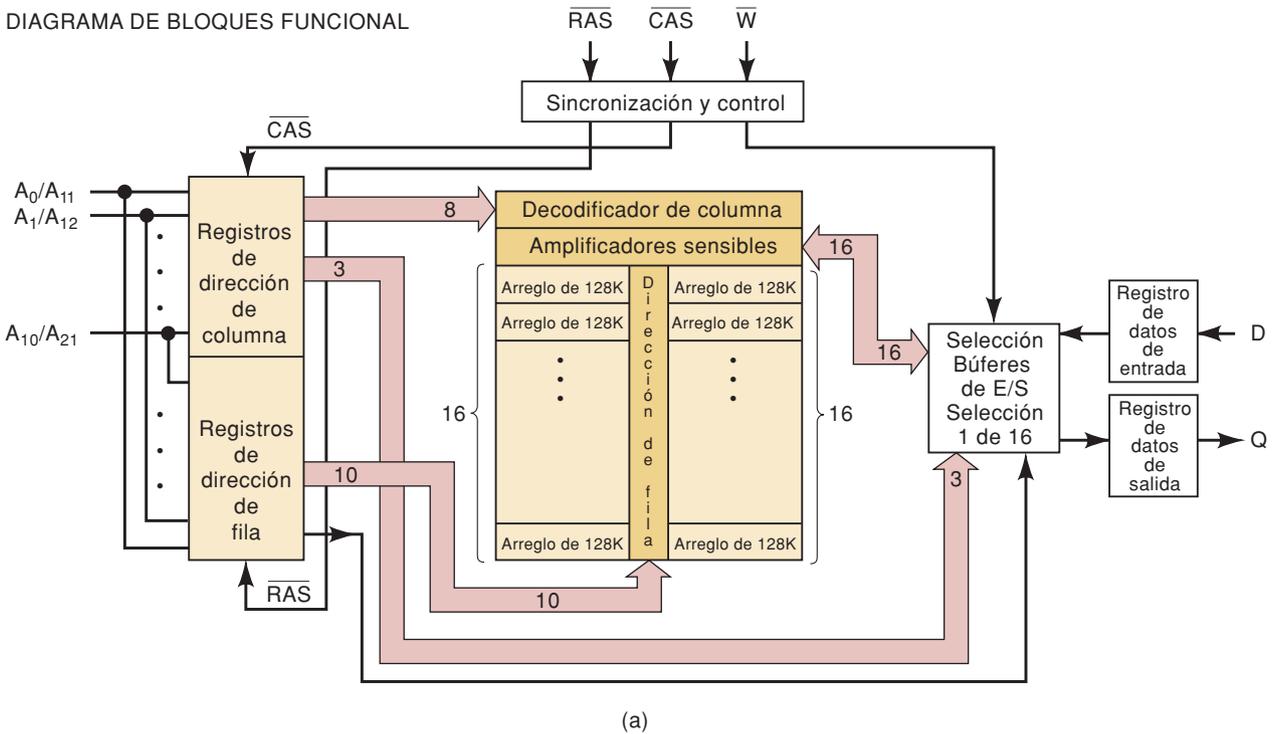


FIGURA 12-27 (a) Arquitectura simplificada de la DRAM TMS44100 de $4M \times 1$; (b) sincronización de $\overline{RAS}/\overline{CAS}$. (Reimpreso con permiso de Texas Instruments.)

de la arquitectura interna de este chip (que se muestra en la figura 12-27) es común en los diagramas que encontrará en los libros de datos. La distribución del arreglo de memoria en este diagrama puede parecer complicada a primera instancia, pero se puede considerar tan sólo como una versión más grande de la DRAM de $16K \times 1$ de la figura 12-25. Visto de manera funcional, es un arreglo de celdas distribuidas en forma de 2048 filas por 2048 columnas. El circuito decodificador de dirección selecciona una sola fila; este circuito puede considerarse como un decodificador 1 de 2048. De igual forma, se selecciona una sola columna mediante lo que se conoce como decodificador 1 de 2048. Como las líneas de dirección están multiplexadas, no se puede presentar en forma simultánea toda la dirección completa de 22 bits. Observe que sólo hay 11 líneas de dirección y que van tanto hacia el registro de

dirección de fila como el de columna. Cada uno de los dos registros de dirección almacena la mitad de la dirección de 22 bits. El registro de fila almacena la mitad superior y el registro de columna almacena la mitad inferior. Dos entradas de **estrobo** muy importantes controlan cuando se debe fijar la información de la dirección. La señal **estrobo de dirección de fila (RAS)** aplica pulsos de reloj al registro de dirección de fila de 11 bits. La señal **estrobo de dirección de columna (CAS)** aplica pulsos de reloj al registro de dirección de columna de 11 bits.

Una dirección de 22 bits se aplica a esta DRAM en dos pasos, mediante el uso de \overline{RAS} y \overline{CAS} . En la figura 12-27(b) se muestra la sincronización. Al principio, tanto \overline{RAS} como \overline{CAS} están en ALTO. En el tiempo t_0 se aplica la dirección de fila de 11 bits ($A_{11}-A_{21}$) a las entradas de dirección. Después de permitir un tiempo para el requerimiento del tiempo de establecimiento (t_{RS}) del registro de dirección de fila, la entrada \overline{RAS} se lleva al nivel BAJO en t_1 . Esta NGT carga la dirección de fila en el registro de dirección de fila, por lo que ahora los bits $A_{11}-A_{21}$ aparecen en las entradas del decodificador de fila. El nivel BAJO en \overline{RAS} también habilita este decodificador, para que pueda decodificar la dirección de fila y seleccionar una fila del arreglo.

En el tiempo t_2 se aplica la dirección de columna de 11 bits (A_0-A_{10}) a las entradas de dirección. En t_3 la entrada \overline{CAS} se lleva al nivel BAJO para cargar la dirección de columna en el registro de dirección de columna. \overline{CAS} también habilita el decodificador de columna para que pueda decodificar la dirección de columna y seleccionar una columna del arreglo.

En este punto las dos partes de la dirección están en sus respectivos registros, los decodificadores las han decodificado para seleccionar la celda que corresponde a la dirección de fila y columna, y puede realizarse una operación de lectura o de escritura en esa celda, justo igual que en una RAM estática.

Tal vez haya observado que esta DRAM no tiene una entrada de selección de chip (CS). Las señales \overline{RAS} y \overline{CAS} realizan la función de selección de chip, ya que ambas deben estar en BAJO para que los decodificadores puedan seleccionar una celda para leer o escribir.

Como puede ver, hay varias operaciones que deben realizarse antes de que los datos almacenados en la DRAM puedan aparecer en las salidas. El término **latencia** se utiliza a menudo para describir el tiempo requerido para realizar estas operaciones. Cada operación requiere cierta cantidad de tiempo, la cual determina la velocidad máxima a la que podemos acceder a los datos en la memoria.

EJEMPLO 12-12

¿Cuántas terminales se ahorran si utilizamos el multiplexaje de direcciones para una DRAM de $16M \times 1$?

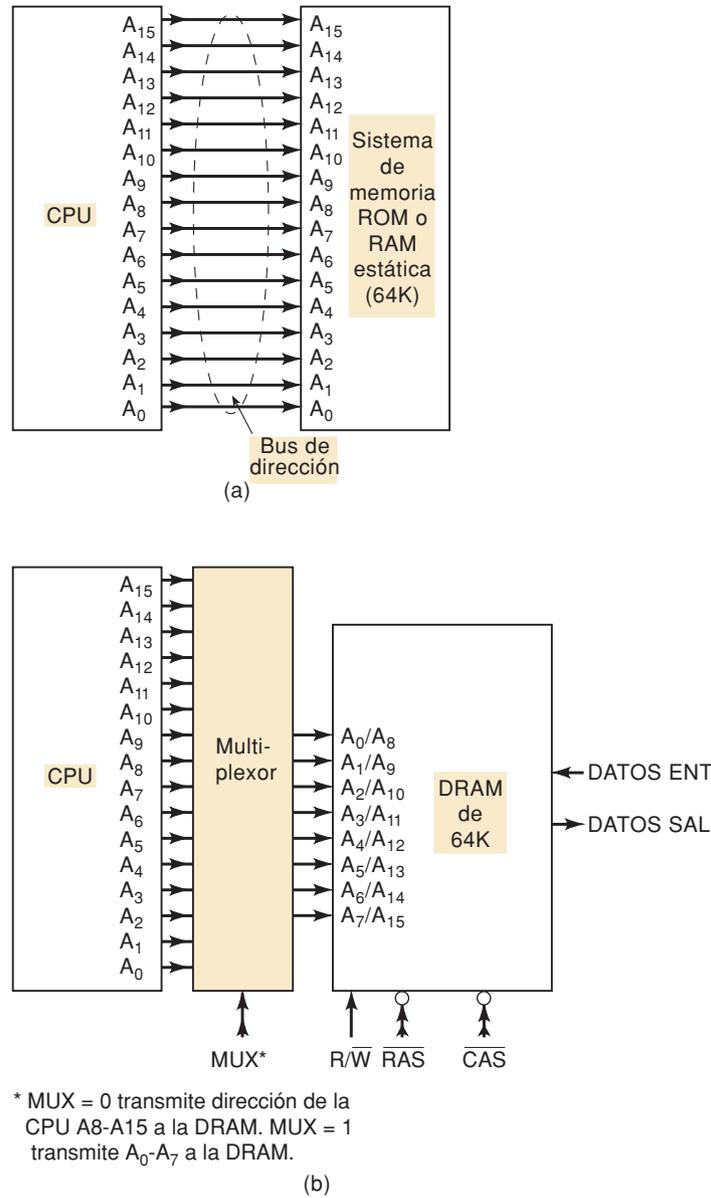
Solución

Se utilizan doce entradas de dirección en vez de 24; se agregan \overline{RAS} y \overline{CAS} ; no se requiere CS. Por lo tanto, hay un ahorro neto de once terminales.

En un sistema computacional simple, las entradas de dirección para el sistema de memoria provienen de la unidad central de procesamiento (CPU). Cuando la CPU desea acceder a una ubicación de memoria específica, genera la dirección completa y la coloca en las líneas de dirección que conforman un bus de direcciones. La figura 12-28(a) muestra esto para una pequeña memoria de computadora que tiene una capacidad de 64K palabras y por ello requiere que un bus de direcciones con 16 líneas vaya directamente de la CPU a la memoria.

Este arreglo funciona para la ROM o la RAM estática, pero debe modificarse para la DRAM que utiliza direccionamiento con multiplexaje. Si todos los 64K de la memoria son de DRAM, entonces sólo tendrá ocho entradas de dirección. Esto significa que las 16 líneas de dirección que provienen del bus de direcciones de la CPU deben alimentarse a un circuito multiplexor que transmita ocho bits de dirección a

FIGURA 12-28 (a) El bus de direcciones de la CPU que controla la memoria ROM o RAM estática; (b) Direcciones de la CPU que controlan un multiplexor, el cual se utiliza para multiplexar las líneas de dirección de la CPU en la DRAM.

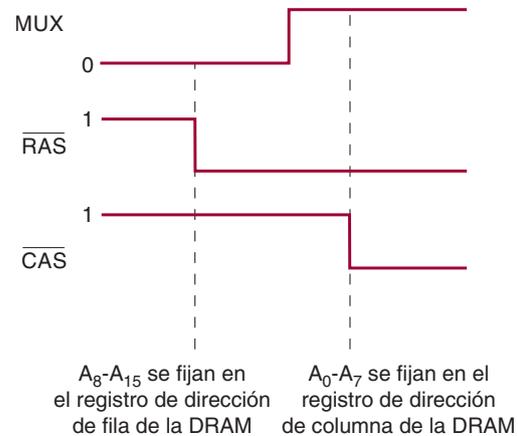


la vez, a las entradas de dirección de la memoria. Esto se muestra en forma simbólica en la figura 12-28(b). La entrada de selección del multiplexor (*MUX*) controla si se van a presentar las líneas de dirección A₀-A₇ de la CPU, o las líneas A₈-A₁₅ en las entradas de dirección de la DRAM.

La señal *MUX* debe sincronizarse con las señales *RAS* y *CAS* que aplican los pulsos de reloj para enviar las direcciones a la DRAM. Esto se muestra en la figura 12-29. *MUX* debe estar en BAJO cuando se aplica un pulso a *RAS* para que cambie a BAJO, de manera que las líneas de dirección A₈-A₁₅ de la CPU lleguen a las entradas de dirección de la DRAM para cargarse en la NGT de *RAS*. De igual forma, *MUX* debe estar en ALTO cuando se aplica un pulso a *CAS* para que cambie a BAJO, de manera que se presenten las líneas de dirección A₀-A₇ de la CPU en las entradas de la DRAM, para cargarse en la NGT de *CAS*.

No vamos a mostrar aquí el circuito real de multiplexaje y sincronización, sino que lo dejaremos para los problemas de final de capítulo (problemas 12-26 y 12-27).

FIGURA 12-29 Sincronización requerida para el multiplexaje de direcciones.



PREGUNTAS DE REPASO

1. Describa la estructura del arreglo de una DRAM de $64K \times 1$.
2. ¿Cuál es el beneficio del multiplexaje de direcciones?
3. ¿Cuántas entradas de dirección habría en un chip DRAM de $1M \times 1$?
4. ¿Cuáles son las funciones de las señales \overline{RAS} y \overline{CAS} ?
5. ¿Cuál es la función de la señal MUX ?

12-15 CICLOS DE LECTURA/ESCRITURA DE LA DRAM

La sincronización de las operaciones de lectura y escritura de una DRAM es mucho más compleja que para una RAM estática; además hay muchos requerimientos de sincronización críticos que el diseñador de la memoria DRAM debe tener en cuenta. En este punto es probable que una discusión detallada sobre estos requerimientos provoque más confusión que entendimiento, por lo cual vamos a concentrarnos en la secuencia de sincronización básica para las operaciones de lectura y escritura en un pequeño sistema de DRAM, como el de la figura 12-28(b).

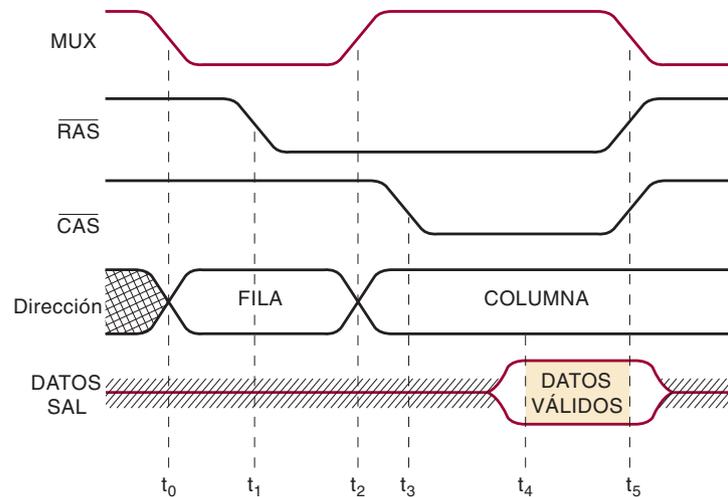
Ciclo de lectura de la DRAM

La figura 12-30 muestra la actividad común de las señales durante la operación de lectura. Se supone que R/W se encuentra en su estado ALTO a lo largo de toda la operación. La siguiente es una descripción paso a paso de los eventos que ocurren en los tiempos que se indican en el diagrama.

- t_0 : MUX se lleva al nivel BAJO para aplicar los bits de dirección de fila (A_8-A_{15}) en las entradas de dirección de la DRAM.
- t_1 : \overline{RAS} se lleva a BAJO para cargar la dirección de fila en la DRAM.
- t_2 : MUX cambia a ALTO para colocar la dirección de columna (A_0-A_7) en las entradas de dirección de la DRAM.
- t_3 : \overline{CAS} cambia a BAJO para cargar la dirección de columna en la DRAM.
- t_4 : la DRAM responde colocando datos válidos provenientes de la celda de memoria seleccionada, en la línea DATOS SAL.
- t_5 : MUX , \overline{RAS} , \overline{CAS} y DATOS SAL regresan a sus estados iniciales.

FIGURA 12-30

Actividad de las señales durante una operación de lectura en una RAM dinámica. La entrada R/\bar{W} (no se muestra) se supone en ALTO.



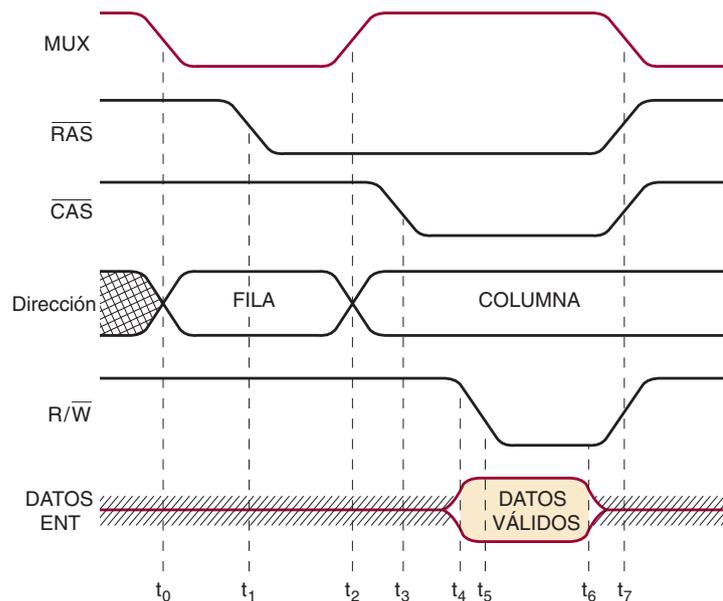
Ciclo de escritura de la DRAM

La figura 12-31 muestra la actividad común de las señales durante una operación de escritura en la DRAM. He aquí una descripción de la secuencia de eventos.

- t_0 : el nivel BAJO en MUX coloca las direcciones de fila en las entradas de la DRAM.
- t_1 : la NGT en \bar{RAS} carga la dirección de fila en la DRAM.
- t_2 : MUX cambia a ALTO para colocar la dirección de columna en las entradas de la DRAM.
- t_3 : la NGT en \bar{CAS} carga la dirección de columna en la DRAM.
- t_4 : los datos que se van a escribir se colocan en la línea DATOS ENT.
- t_5 : se aplica un pulso a R/\bar{W} para que cambie a BAJO y se escriban los datos en la celda seleccionada.
- t_6 : los datos de entrada se extraen de DATOS ENT.
- t_7 : MUX , \bar{RAS} , \bar{CAS} y R/\bar{W} se regresan a sus estados iniciales.

FIGURA 12-31

Actividad de las señales para una operación de escritura en una RAM dinámica.



PREGUNTAS DE REPASO

1. Verdadero o falso:
 - (a) Durante un ciclo de lectura, la señal \overline{RAS} se activa antes de la señal \overline{CAS} .
 - (b) Durante una operación de escritura, \overline{CAS} se activa antes de \overline{RAS} .
 - (c) R/\overline{W} se mantiene en BAJO durante toda la operación de escritura.
 - (d) Las entradas de dirección para una DRAM cambiarán dos veces durante una operación de lectura o de escritura.
2. ¿Qué señal de la figura 12-28(b) asegura que aparezca la parte correcta de la dirección completa en las entradas de la DRAM?

12-16 REFRESCO DE LA DRAM

La celda de una DRAM se regenera cada vez que se realiza una operación de lectura sobre esa celda. Cada celda de memoria debe regenerarse por lo general, cada 4 a 16 ms (dependiendo del dispositivo) o se perderán sus datos. Este requerimiento parece en extremo difícil, si no es que imposible, sobre todo para cumplirlo con DRAMs de gran capacidad. Por ejemplo, una DRAM de $1M \times 1$ tiene $10^{20} = 1,048,576$ celdas. Para asegurar que cada celda se refresque dentro de un lapso no mayor de 4 ms, debe realizar las operaciones de lectura en direcciones sucesivas a la velocidad de una por cada 4 ns ($4 \text{ ms}/1,048,756 \approx 4 \text{ ns}$). Esto es demasiado rápido para cualquier chip DRAM. Por fortuna, los fabricantes han diseñado chips DRAM de manera que

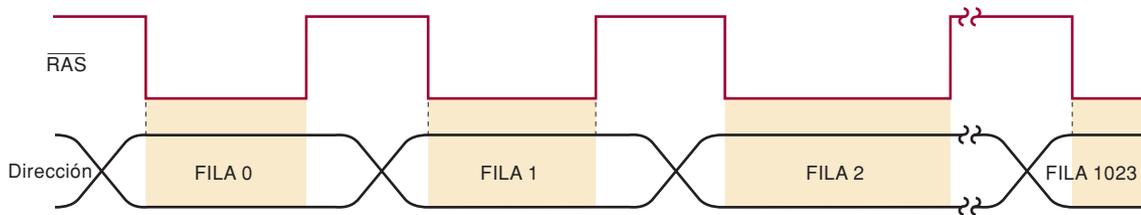
cada vez que se realice una operación de lectura en una celda, se regenerarán todas las celdas en esa fila.

Por lo tanto, es necesario realizar una operación de lectura solo en cada *fila* de un arreglo de DRAM una vez cada 4 ms para garantizar que se refresque cada una de las *celdas* del arreglo. Al referirnos a la DRAM de $4M \times 1$ de la figura 12-27(a), si se introduce cualquier dirección en el registro de dirección de fila, se regenerarán de manera automática todas (las 2048) celdas en esa fila.

Es evidente que esta característica facilita la labor de mantener regeneradas todas las celdas de la DRAM. No obstante, durante la operación normal del sistema en el cual funciona una DRAM, es muy poco probable que se realice una operación de lectura en cada fila de la DRAM dentro del límite de tiempo de regeneración requerido. Por lo tanto, se necesita cierto tipo de lógica de control de regeneración, ya sea externa para el chip DRAM o como parte de sus circuitos internos. En cualquier caso hay dos modos de regeneración: en *ráfaga* y *distribuida*.

En el modo de regeneración en ráfaga se suspende la operación normal de la memoria y cada fila de la DRAM se regenera en forma sucesiva hasta que se hayan regenerado todas las filas. En el modo de regeneración distribuida, la regeneración de las filas se entremezcla con las operaciones normales de la memoria.

El método más universal para regenerar una DRAM es la **regeneración sólo de RAS**. Para ello se aplica mediante estrobo una dirección de fila con *RAS*, mientras que *CAS* y *R/W* permanecen en ALTO. La figura 12-32 ilustra cómo se utiliza la regeneración sólo de *RAS* para una regeneración en ráfaga del TMS44100. Parte de la complejidad del arreglo de memoria en este chip está ahí para facilitar las operaciones de regeneración. Como hay dos bancos alineados en la misma fila, ambos pueden regenerarse al mismo tiempo, que en efecto es lo mismo que si sólo hubiera 1024 filas. Un **contador de regeneración** se utiliza para suministrar direcciones de fila de 10 bits a las entradas de dirección de la DRAM, empezando desde 0000000000 (fila 0). Se aplica un pulso a *RAS* para que cambie a BAJO y se cargue esta dirección en la DRAM, con lo cual se regenera la fila 0 en ambos bancos. El contador se incrementa y el proceso se repite hasta la dirección 1111111111 (fila 1023). Para el TMS44100 puede completarse una regeneración en ráfaga en casi 113 μs y debe repetirse por lo menos cada 16 ms.



* Las líneas $\overline{R/W}$ y \overline{CAS} se mantienen en ALTO

FIGURA 12-32 El método de regeneración sólo de \overline{RAS} utiliza únicamente la señal \overline{RAS} para cargar la dirección de fila en la DRAM y regenerar todas las celdas en esa fila. La regeneración sólo de \overline{RAS} puede utilizarse para realizar una regeneración en ráfaga, como se muestra. Un contador de regeneración proporciona las direcciones de fila en secuencia, desde la fila 0 hasta la fila 1023 (para una DRAM de $4M \times 1$).

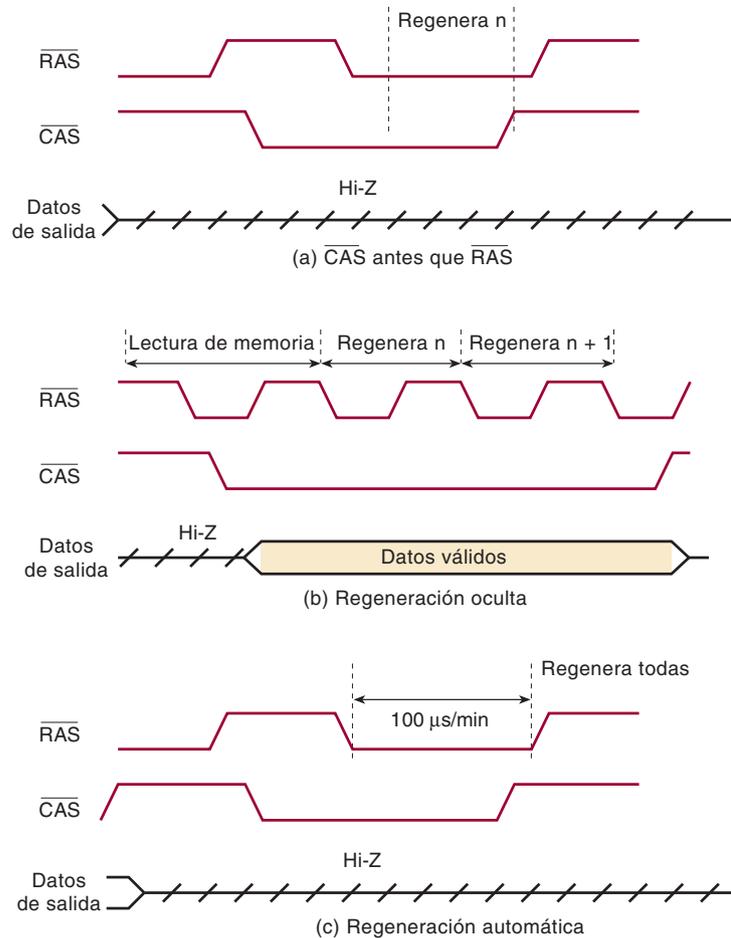
Aunque la idea del contador de regeneración parece bastante sencilla, debemos tener en cuenta que las direcciones de fila del contador de regeneración no pueden interferir con las direcciones que vienen de la CPU durante las operaciones normales de lectura/escritura. Por esta razón, las direcciones del contador de regeneración deben multiplexarse con las direcciones de la CPU, de manera que se active la fuente apropiada de direcciones de la DRAM en los tiempos apropiados.

Para poder liberar a la CPU de la computadora de algunas de estas cargas, comúnmente se utiliza un chip especial conocido como **controlador de RAM dinámica (DRAM)**. Como mínimo, este chip realizará el multiplexaje de direcciones y la generación de secuencias de conteo de regeneración, dejando la generación de la sincronización para las señales \overline{RAS} , \overline{CAS} y MUX para otros circuitos lógicos y la persona que programa la computadora. Otros controladores de DRAM son completamente automáticos. Sus entradas son muy parecidas a las de una RAM estática o una ROM. Generan en forma automática la secuencia de regeneración con la frecuencia suficiente como para mantener la memoria, multiplexar el bus de direcciones, generar las señales \overline{RAS} y \overline{CAS} , y supervisar el control de la DRAM entre los ciclos de lectura/escritura de la CPU y las operaciones de regeneración locales. En las computadoras personales actuales, el controlador de DRAM y otros circuitos controladores de alto nivel se integran en un conjunto de circuitos VLSI, a los cuales se les conoce como “conjunto de chips”. A medida que se desarrollan nuevas tecnologías de DRAM, se diseñan nuevos conjuntos de chips para aprovechar los últimos avances. En muchos casos, el número de conjuntos de chips existentes (o anticipados) que soportan cierta tecnología en el mercado es lo que determina en qué tecnología de DRAM invertirán los fabricantes.

La mayoría de los chips DRAM en producción hoy en día tienen capacidad de regeneración en el chip, con lo cual se elimina la necesidad de suministrar direcciones de regeneración externas. Uno de estos métodos, que se muestra en la figura 12-33(a), se llama *regeneración de \overline{CAS}* antes que \overline{RAS} . En este método, la señal \overline{CAS} se lleva al nivel BAJO primero y se mantiene hasta después de que \overline{RAS} cambia a BAJO. Esta secuencia regenerará una fila del arreglo de memoria e incrementará un contador interno que genera las direcciones de fila. Para realizar una regeneración en ráfaga mediante esta característica, \overline{CAS} puede mantenerse en BAJO mientras que se aplica un pulso a \overline{RAS} para cada fila, hasta que se regeneren todas. Durante este ciclo de regeneración se ignoran todas las direcciones externas. El TMS44100 también ofrece una “regeneración oculta”, la cual permite regenerar una fila mientras se retienen datos en la salida. Para ello hay que mantener a \overline{CAS} en BAJO después de un ciclo de lectura y posteriormente aplica un pulso a \overline{RAS} , como en la figura 12-33(b).

El modo de regeneración automática de la figura 12-33(c) automatiza el proceso por completo. Al forzar a \overline{CAS} para que cambie a BAJO antes que \overline{RAS} , y después mantener ambas señales en BAJO durante por lo menos $100 \mu s$, un oscilador interno aplica pulsos de reloj al contador de dirección de fila hasta que se regeneran todas las celdas. El modo que seleccione un diseñador de sistemas dependerá de qué tan ocupada esté la CPU de la computadora. Si puede ceder $100 \mu s$ sin acceder a su

FIGURA 12-33 Modos de regeneración del TMS44100.



memoria, y si puede hacer esto cada 16 ms, lo más conveniente es la regeneración automática.

No obstante, si esto reduce demasiado la velocidad de ejecución del programa, tal vez se requiera cierto tipo de regeneración distribuida, mediante el uso de CAS antes que RAS o ciclos de regeneración oculta. En cualquier caso, todas las celdas deben regenerarse dentro del tiempo permitido o se perderán los datos.

PREGUNTAS DE REPASO

1. Verdadero o falso:

- (a) En la mayoría de las DRAMs, es necesario leer sólo de una celda en cada fila para poder regenerar todas las celdas en esa fila.
- (b) En el modo de regeneración en ráfaga, todo el arreglo se regenera mediante un pulso en \overline{RAS} .

2. ¿Cuál es la función de un contador de regeneración?

3. ¿Qué funciones realiza un controlador de DRAM?

4. Verdadero o falso:

- (a) En el método de regeneración sólo de \overline{RAS} , la señal \overline{CAS} se mantiene en BAJO.
- (b) La regeneración de \overline{CAS} antes que \overline{RAS} sólo la pueden utilizar DRAMs con circuitos de control de generación en el chip.

12-17 TECNOLOGÍA DE LA DRAM*

Al seleccionar un tipo específico de dispositivo de RAM para un sistema, un diseñador se topa con ciertas decisiones difíciles. La capacidad (tan grande como sea posible), la velocidad (tan rápida como sea posible), la energía necesaria (lo menos que sea posible), el costo (lo más bajo que sea posible), y la conveniencia (lo más fácil de cambiar que sea posible) deben mantenerse en un balance razonable, ya que ningún tipo de RAM puede maximizar todas estas características deseadas. El mercado de la RAM semiconductora intenta constantemente producir la mezcla ideal de estas características en sus productos para diversas aplicaciones. En esta sección se explican algunos de los términos actuales que se utilizan en relación con la tecnología de la RAM. Éste es un tema muy dinámico y tal vez algunos de estos términos serán historia en el momento que lea este libro, pero he aquí lo último en la tecnología.

Módulos de memoria

Gracias a que muchas compañías fabrican tarjetas madre para sistemas de computadora personal, se han adoptado conectores de interfase de memoria estándar. Estos conectores reciben una pequeña tarjeta de circuito impreso con puntos de contacto en ambos lados del borde de la tarjeta. Estas tarjetas modulares permiten instalar o sustituir con facilidad los componentes de memoria en una computadora. El módulo de memoria simple en línea (SIMM) es una tarjeta de circuito con 73 contactos equivalentes en funcionalidad, en ambos lados de la tarjeta. Un punto de contacto redundante en cada lado de la tarjeta ofrece cierta seguridad de que se realiza un contacto confiable. Estos módulos utilizan chips DRAM sólo de 5 V, los cuales varían en capacidad desde 1 hasta 16 Mbits en encapsulados tipo ala de gaviota con montaje superficial, o de terminal en J. Los módulos de memoria varían en capacidad, de 1 a 32 Mbytes.

El módulo de memoria dual en línea (DIMM) más reciente tiene 84 contactos únicos en funcionalidad, en cada lado de la tarjeta. Las terminales adicionales son necesarias debido a que los DIMMS se conectan a buses de datos de 64 bits, como los que se utilizan en las PCs modernas. Existen versiones de 3.3 V y de 5 V. También hay versiones con búfer y sin búfer. La capacidad del módulo depende de los chips DRAM que estén montados en él; y a medida que aumente la capacidad de la DRAM, también aumentará la capacidad de los DIMMs. El conjunto de chips y el diseño de la tarjeta madre que se utilicen en un sistema dado es lo que determina el tipo de DIMM que puede utilizarse. Para aplicaciones compactas, como las de computadoras portátiles, hay un módulo de memoria dual en línea de diseño pequeño (SODIMM).

El principal problema en la industria de las computadoras personales es proporcionar un sistema de memoria que sea lo bastante rápido como para mantenerse a la par con las velocidades de reloj del microprocesador que siempre van en aumento, y mantener al mismo tiempo el costo en un nivel aceptable. Se están agregando características especiales a los dispositivos básicos de DRAM para mejorar su ancho de banda total. Acaba de introducirse al mercado un nuevo tipo de encapsulado, conocido como RIMM. RIMM significa Módulo de memoria Rambus en línea. Rambus es una compañía que ha inventado varios nuevos enfoques revolucionarios en relación con la tecnología de la memoria. El RIMM es su encapsulado propietario que almacena sus chips de memoria propietarios, llamados chips de DRAM Rambus directa (DRDRAM). Aunque estos métodos para mejorar el rendimiento cambian en forma constante, en los textos relacionados con la memoria se hace una extensa referencia a las tecnologías que describiremos en las siguientes secciones.

DRAM FPM

El modo de paginación rápida (FPM) permite un acceso más rápido a las ubicaciones de memoria aleatorias dentro de la “página” actual. En esencia, una página es un intervalo de direcciones de memoria que tiene valores idénticos en los bits de

* Puede omitir este tema sin que se vea afectada la continuidad del resto del libro.

dirección superiores. Para poder acceder a los datos en la página actual sólo hay que modificar las líneas de dirección inferiores.

DRAM EDO

Las DRAMs con salida de datos extendida (EDO) ofrecen una mejora en comparación con las DRAMs FPM. Para los accesos en una página dada, se detecta el valor de datos en la ubicación de memoria actual y se fija en las terminales de salida. En las DRAMs FPM, el amplificador sensible controla la salida sin un latch, para lo cual requiere que *CAS* permanezca en el nivel bajo hasta que los valores de datos se vuelvan válidos. Con EDO, mientras estos datos están presentes en las salidas, *CAS* puede completar su ciclo, se puede decodificar una nueva dirección en la página actual y los circuitos de la ruta de datos pueden restablecerse para el siguiente acceso. Esto permite al controlador de memoria enviar como salida la siguiente dirección al mismo tiempo que se lee la palabra actual.

SDRAM

La DRAM síncrona está diseñada para transferir datos en *ráfagas* de disparo rápido de varias ubicaciones de memoria secuenciales. La primera ubicación a la que se accede es la más lenta, debido a la sobrecarga (latencia) del proceso de fijar la dirección de fila y de columna. Después el reloj del sistema de bus aplica pulsos de reloj a los valores de datos (en vez de la línea de control *CAS*) en ráfagas de ubicaciones de memoria dentro de la misma página. En su interior, las SDRAMs se organizan en dos bancos. Esto permite leer datos a una velocidad muy rápida, ya que se accede en forma alternativa a cada uno de los dos bancos. Para poder proveer todas las características y la flexibilidad necesaria para que este tipo de DRAM funcione con una amplia variedad de requerimientos de sistema, los circuitos dentro de la SDRAM se han vuelto más complicados. Se necesita una secuencia de instrucciones para indicar a la SDRAM qué opciones son necesarias, como la longitud de la ráfaga, los datos secuenciales o interpaginados y los modos de *CAS* antes que *RAS* o de regeneración automática. El modo de regeneración automática permite al dispositivo de memoria realizar todas las funciones necesarias para mantener sus celdas actualizadas.

DDRSDRAM

La SDRAM de doble velocidad de transferencia de datos ofrece una mejora en comparación con la SDRAM. Para poder agilizar la operación de la SDRAM mientras se opera con base en un reloj de sistema síncrono, esta tecnología transfiere datos en los flancos positivo y negativo del reloj del sistema, duplicando con efectividad la velocidad potencial de la transferencia de datos.

SLDRAM

La DRAM de enlace síncrono es una mejora evolutiva en comparación con la DDR-SDRAM. Puede operar a velocidades de bus de hasta 200 MHz y aplica pulsos de reloj a los datos en forma síncrona, en los flancos positivo y negativo del reloj del sistema. Un consorcio de varios fabricantes de DRAM lo está desarrollando como un estándar abierto. Si se desarrollan conjuntos de chips que puedan aprovechar estos dispositivos de memoria y suficientes diseñadores de sistemas adoptan esta tecnología, es muy probable que se convierta en una forma muy utilizada de DRAM.

DRDRAM

La DRAM Rambus directa es un dispositivo propietario, fabricado y comercializado por Rambus, Inc. Utiliza un nuevo enfoque revolucionario de la arquitectura del sistema de DRAM, con un control más intenso, integrado en el dispositivo de memoria. Esta tecnología aún se encuentra luchando con los demás estándares para encontrar su nicho en el mercado.

PREGUNTAS DE REPASO

1. ¿Pueden intercambiarse los SIMMs y los DIMMs?
2. ¿Qué es una “página” de memoria?
3. ¿Por qué el “modo de paginación” es más rápido?
4. ¿Qué significa *EDO*?
5. ¿Qué término se utiliza para acceder a varias ubicaciones de memoria consecutivas?
6. ¿Con qué se sincroniza una DRAM?

12-18 EXPANSIÓN DE TAMAÑO DE PALABRA Y CAPACIDAD

En muchas aplicaciones de memoria no se puede satisfacer la capacidad o el tamaño de palabra requeridos para una memoria RAM o ROM con un solo chip de memoria. Hay que combinar varios chips de memoria para proporcionar la capacidad y/o el tamaño de palabra. En esta sección veremos cómo se hace esto a través de varios ejemplos en los que se ilustran las ideas importantes que se utilizan cuando se integran chips de memoria con un microprocesador. Los siguientes ejemplos están diseñados para ser instructivos, y los tamaños de los chips de memoria utilizados se eligieron de manera que se conservara espacio. Las técnicas que vamos a presentar pueden extenderse a chips de memoria más grandes.

Expansión del tamaño de palabra

Suponga que necesitamos una memoria que pueda almacenar 16 palabras de ocho bits y todo lo que tenemos son chips de RAM ordenados como memorias de 16×4 , con líneas comunes de E/S. Podemos combinar dos de estos chips de 16×4 para producir la memoria deseada. La configuración para hacerlo se muestra en la figura 12-34. Examine este diagrama con cuidado y vea lo que puede averiguar de él antes de seguir leyendo.

Como cada chip puede almacenar 16 palabras de cuatro bits y queremos almacenar 16 palabras de 8 bits, vamos a utilizar cada chip para que almacene la *mitad* de cada palabra. Es decir, la RAM-0 almacena los cuatro bits de *mayor* orden de cada una de las 16 palabras y la RAM-1 almacena los cuatro bits de *menor* orden de cada una de las 16 palabras. De esta manera hay una palabra de ocho bits completa en las salidas de la RAM que se conectan al bus de datos.

Para seleccionar cualquiera de las 16 palabras, se aplica el código de dirección apropiado al *bus de direcciones* de cuatro líneas (A_3, A_2, A_1, A_0). Por lo general, las líneas de dirección se originan en la CPU. Observe que cada línea del bus de direcciones está conectada a la entrada de dirección correspondiente de cada chip. Esto significa que, una vez que se coloca un código de dirección en el bus de direcciones, este mismo código de dirección se aplica a los dos chips, de manera que pueda accederse a la misma ubicación en cada chip al mismo tiempo.

Una vez seleccionada la dirección, podemos leer o escribir en ella bajo el control de la línea común R/\bar{W} y \bar{CS} . Para leer, R/\bar{W} debe estar en alto y \bar{CS} debe estar en bajo. Esto hace que las líneas de E/S de la RAM actúen como *salidas*. La RAM-0 coloca su palabra de cuatro bits seleccionada en las cuatro líneas superiores del bus de datos y la RAM-1 coloca su palabra de cuatro bits seleccionada en las cuatro líneas inferiores del bus de datos. Así, el bus de datos contiene la palabra de ocho bits completa que se seleccionó, la cual puede ahora transmitirse hacia algún otro dispositivo (por lo general, hacia un registro en la CPU).

Para escribir, la condición $R/\bar{W} = 0$ y $\bar{CS} = 0$ hace que las líneas de E/S de la RAM actúen como *entradas*. La palabra de ocho bits que se va a escribir se coloca en el bus de datos (por lo general, la CPU se encarga de ello). Los cuatro bits superiores se escribirán en la ubicación seleccionada de la RAM-0, y los cuatro bits inferiores se escribirán en la RAM-1.

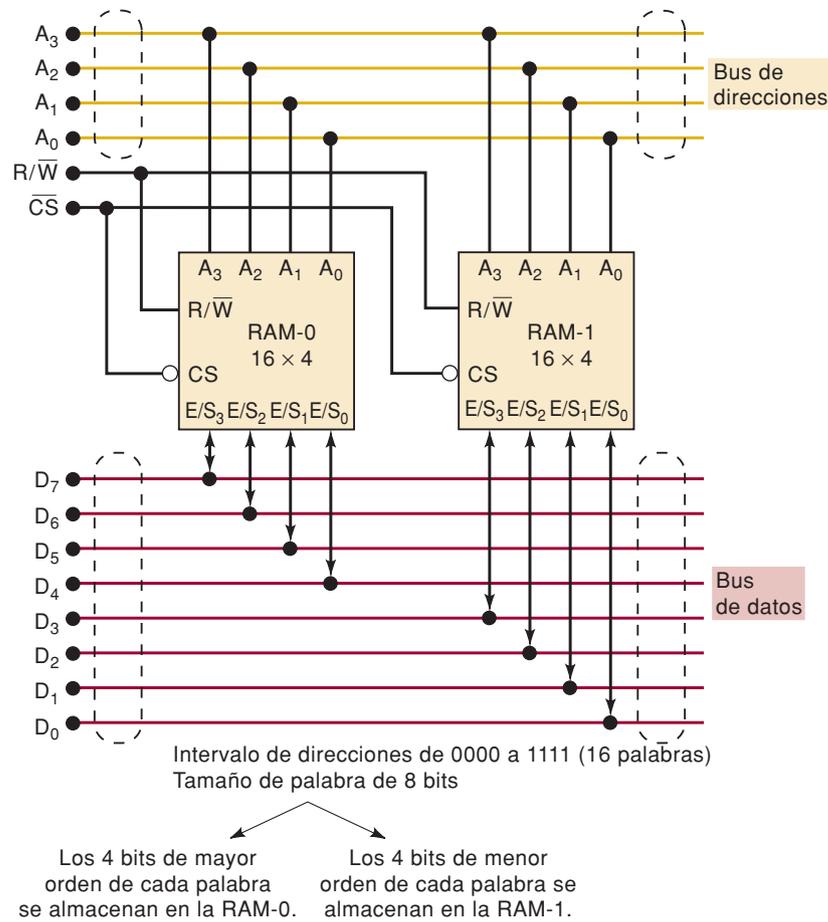


FIGURA 12-34 Combinación de dos RAMs de 16×4 para obtener un módulo de 16×8 .

En esencia, la combinación de los dos chips de RAM actúa como un chip de memoria individual de 16×8 . A esta combinación se le denomina *módulo de memoria* de 16×8 .

La misma idea básica para expandir el tamaño de palabra funcionará en distintas situaciones. Lea el siguiente ejemplo y dibuje un diagrama sencillo que ilustre la apariencia del sistema antes de ver la solución.

EJEMPLO 12-13

El 2125A es un CI de RAM estática que tiene una capacidad de $1K \times 1$, una entrada de selección de chip activa en BAJO y líneas separadas de entrada y salida de datos. Muestre cómo se pueden combinar varios CIs 2125A para formar un módulo de $1K \times 8$.

Solución

El arreglo se muestra en la figura 12-35, en donde se utilizan ocho chips 2125A para un módulo de $1K \times 8$. Cada chip almacena uno de los bits de cada una de las 1024 palabras de ocho bits. Observe que todas las entradas R/\bar{W} y \overline{CS} están conectadas entre sí, y que el bus de direcciones de 10 líneas está conectado a las entradas de dirección de cada chip. Observe además que como el 2125A tiene terminales separadas de entrada y salida de datos, ambas terminales de cada chip están conectadas a la misma línea del bus de datos.

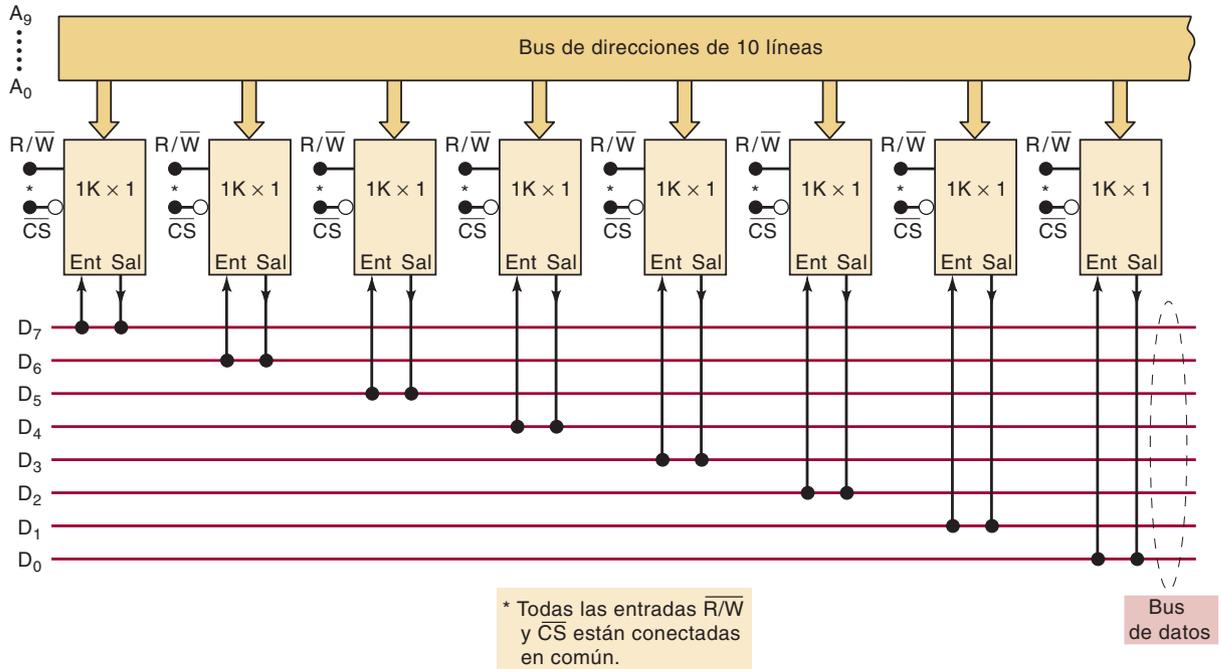


FIGURA 12-35 Ocho chips 2125A de 1K × 1 que forman un arreglo de memoria de 1K × 8.

Expansión de la capacidad

Suponga que necesitamos una memoria que pueda almacenar 32 palabras de cuatro bits y que todo lo que tenemos son los chips de 16 × 4. Si combinamos dos chips de 16 × 4 como se muestra en la figura 12-36, podremos producir la memoria deseada. Una vez más, examine este diagrama y vea qué puede determinar de él antes de seguir leyendo.

Cada RAM se utiliza para almacenar 16 palabras de cuatro bits. Las cuatro terminales de E/S de datos de cada RAM se conectan a un bus de datos común de cuatro líneas. Sólo uno de los chips de RAM puede seleccionarse (habilitarse) en un momento dado, de manera que no haya problemas de colisión de bus. Para asegurar esto se controlan las respectivas entradas \overline{CS} mediante distintas señales lógicas.

La capacidad total de este módulo de memoria es de 32 × 4, por lo que debe haber 32 direcciones distintas. Para esto se requieren cinco líneas en el bus de direcciones. La línea de dirección superior A_4 se utiliza para seleccionar una u otra RAM (a través de las entradas \overline{CS}), para leer o escribir información en ella. Las otras cuatro líneas de dirección (A_0 - A_3) se utilizan para seleccionar una de las 16 ubicaciones de memoria del chip de RAM seleccionado.

Para ilustrar esto, cuando $A_4 = 0$ la señal \overline{CS} de la RAM-0 habilita este chip para lectura o escritura. Después se puede acceder a cualquier ubicación en la RAM-0 a través de las líneas A_3 - A_0 . Las últimas cuatro líneas de dirección pueden variar de 0000 a 1111 para seleccionar la ubicación deseada. Así, el intervalo de direcciones que representan ubicaciones en la RAM-0 es

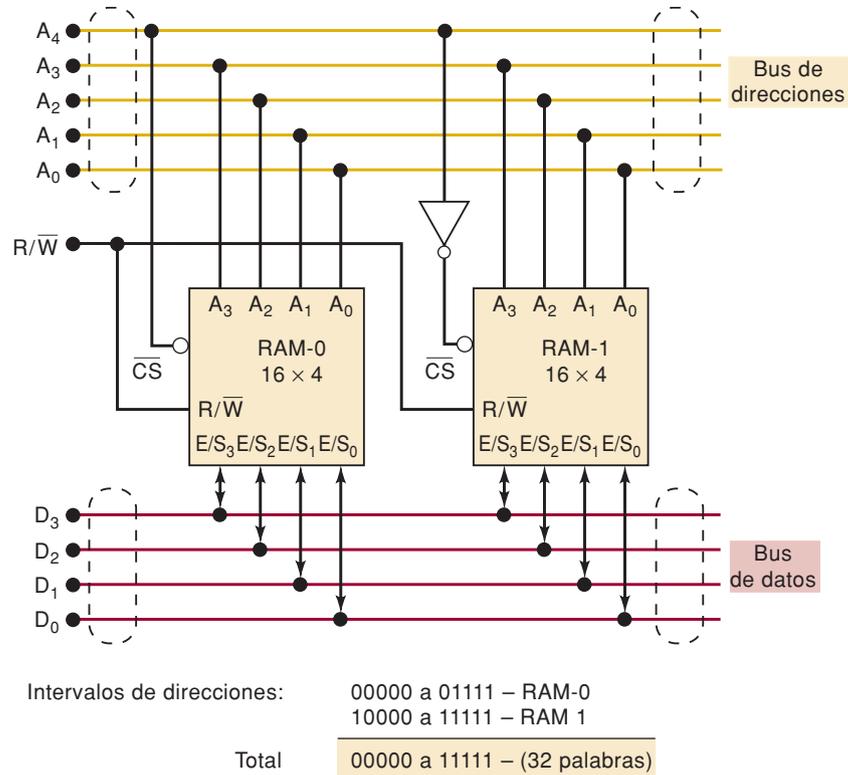
$$A_4A_3A_2A_1A_0 = 00000 \text{ a } 01111$$

Observe que cuando $A_4 = 0$ la señal \overline{CS} de la RAM-1 está en alto, por lo cual sus líneas de E/S están deshabilitadas (Hi-Z) y no pueden comunicarse con (dar o recibir datos) el bus de datos.

Hay que dejar en claro que cuando $A_4 = 1$ se invierten los roles de la RAM-0 y la RAM-1. Ahora la RAM-1 está habilitada y las líneas A_3 - A_0 seleccionan una de sus ubicaciones. Así, el intervalo de direcciones ubicadas en la RAM-1 es

$$A_4A_3A_2A_1A_0 = 10000 \text{ a } 11111$$

FIGURA 12-36
Combinación de dos chips de 16×4 para formar una memoria de 32×4 .



EJEMPLO 12-14

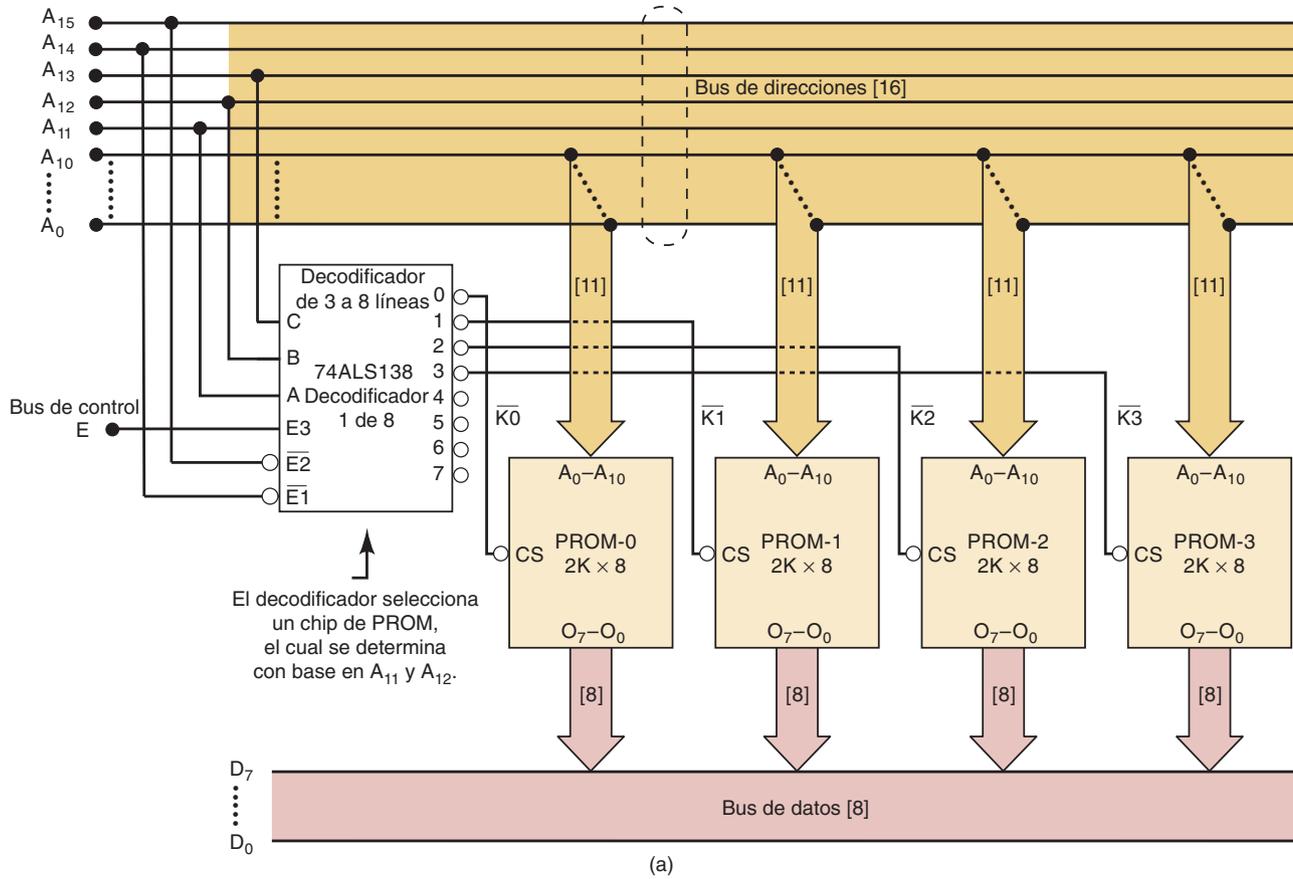
Queremos combinar varias PROMs de $2K \times 8$ para producir una capacidad total de $8K \times 8$. ¿Cuántos chips de PROM se necesitan? ¿Cuántas líneas se requieren para el bus de direcciones?

Solución

Se requieren cuatro chips de PROM, cada uno con capacidad de almacenar $2K$ de las $8K$ palabras. Como $8K = 8 \times 1024 = 8192 = 2^{13}$, se requieren *trece* líneas de dirección.

La configuración para la memoria del ejemplo 12-14 es similar a la memoria de 32×4 de la figura 12-36. No obstante, es un poco más compleja debido a que requiere un circuito decodificador para generar las señales de entrada \overline{CS} . En la figura 12-37(a) se muestra el diagrama completo para esta memoria de 8192×8 .

La capacidad total del bloque de memoria es de 8192 bytes. Este sistema que contiene el bloque de memoria tiene un bus de direcciones de 16 bits, algo común en un sistema pequeño basado en microcontrolador. El decodificador en este sistema sólo puede habilitarse cuando A_{15} y A_{14} estén en BAJO, y que E esté en ALTO. Esto significa que sólo puede decodificar direcciones menores del valor 4000H (4000H). Es más fácil comprender esto si analizamos el mapa de memoria de la figura 12-37(b). Ahí podemos ver que los dos MSBs superiores (A_{15} y A_{14}) están siempre en BAJO para las direcciones menores que 4000H. Las líneas de dirección A_{13} - A_{11} se conectan a las entradas C-A del decodificador, respectivamente. Estos tres bits se decodifican y se utilizan para seleccionar uno de los CIs de memoria. Observe en el mapa de bits de la figura 12-37(b) que todas las direcciones dentro de la PROM-0 tienen $A_{13}, A_{12}, A_{11} = 0, 0, 0$; la PROM-1 se selecciona cuando estos bits tienen un valor de $0, 0, 1$; la PROM-2 cuando son $0, 1, 0$; y la PROM-3 cuando son $0, 1, 1$. Cuando se selecciona cualquier PROM, las líneas de dirección A_{10} - A_0 pueden variar desde sólo 0s hasta sólo 1s. Para sintetizar el esquema de direcciones de este sistema, se



A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Dirección	Mapa del sistema	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	PROM-0	2K
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	07FF		
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0800	PROM-1	2K
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFF		
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000	PROM-2	2K
0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	17FF		
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1800	PROM-3	2K
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF		
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000	O_4	8K
		1	0	1													O_5 Decodificada	
		1	1	0													O_6 Expansión	
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	3FFF	O_7	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000	Disponible	48K
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFF		

FIGURA 12-37 (a) Cuatro PROMs de $2K \times 8$ en un arreglo para formar una capacidad total de $8K \times 8$. (b) Mapa de memoria del sistema completo.

utilizan los dos bits superiores para seleccionar este decodificador, se utilizan los siguientes tres bits (A_{13} - A_{11}) para seleccionar uno de los cuatro chips de PROM y se utilizan las 11 líneas de dirección inferiores para seleccionar una de las 2048 ubicaciones de memoria del tamaño de un byte en la PROM habilitada.

Cuando hay una dirección del sistema de 4000H o mayor en el bus de direcciones, ninguna de las PROMs se habilitará. No obstante, pueden utilizarse las salidas 4-7 del decodificador para habilitar más chips de memoria si deseamos expandir la capacidad del sistema de memoria. El mapa de memoria a la derecha de la figura 12-37(b) muestra un área de 48K del espacio del sistema que no está ocupada por este bloque de memoria. Para poder expandirse a esta área del mapa de memoria, se requeriría más lógica de decodificación.

EJEMPLO 12-15

¿Qué se necesitaría para expandir la memoria de la figura 12-37 a $32K \times 8$? Describa qué líneas de dirección se utilizan.

Solución

Una capacidad de 32K requiere 16 de los chips PROM de 2K. Ya se muestran cuatro de ellos y pueden conectarse cuatro más a las salidas O_4 - O_7 del decodificador. Esto cubre la mitad del sistema. Para seleccionar los otros ocho chips PROM hay que agregar otro decodificador 74ALS138 y habilitarlo sólo cuando $A_{15} = 0$ y $A_{14} = 1$. Para lograrlo se conecta un inversor entre A_{14} y \bar{E}_1 mientras se conecta A_{15} en forma directa con E_2 . Las otras conexiones son iguales que en el decodificador existente.

Decodificación incompleta de direcciones

En muchos casos es necesario utilizar varios dispositivos de memoria en el mismo sistema de memoria. Por ejemplo, considere los requerimientos de un sistema de tablero de controles digital en un automóvil. Éste se implementa, por lo general, mediante el uso de un microprocesador. En consecuencia, necesitamos cierta ROM no volátil para almacenar las instrucciones del programa. Necesitamos cierta memoria de lectura/escritura para almacenar los dígitos que representan la velocidad, RPM, galones de combustible, etcétera. Hay que almacenar otros valores digitalizados para representar la presión del aceite, la temperatura del motor, voltaje de la batería, etcétera. También necesitamos cierto almacenamiento de lectura/escritura no volátil (EEPROM) para la lectura del odómetro, ya que no sería conveniente que este número se restableciera a 0 o que asumiera un valor aleatorio cada vez que se desconectara la batería del automóvil.

La figura 12-38 muestra un sistema de memoria que podría utilizarse en un sistema de microcomputadora. Observe que la parte correspondiente a la ROM está compuesta de dos dispositivos de $8K \times 8$ (PROM-0 y PROM-1). La sección de RAM requiere un solo dispositivo de $8K \times 8$. La EEPROM disponible es sólo un dispositivo de $2K \times 8$. El sistema de memoria requiere un decodificador para seleccionar sólo un dispositivo a la vez. Este decodificador divide todo el espacio de memoria (suponiendo 16 bits de dirección) en bloques de direcciones de 8K. En otras palabras, la salida de cada decodificador se activa mediante 8192 (8K) direcciones distintas. Observe que las tres líneas de dirección superiores controlan el decodificador. Las 13 líneas de dirección de menor orden se conectan en forma directa a las entradas de dirección en los chips de memoria. La única excepción a esto es la EEPROM, la cual tiene sólo 11 líneas de dirección para su capacidad de 2 Kbytes. Si el intervalo de la dirección (en hexadecimal) de esta EEPROM debe estar entre 6000 y 67FF, responderá a estas direcciones de la manera esperada. No obstante, las dos líneas de dirección A_{11} y A_{12} no están involucradas en el esquema de decodificación para este chip. La salida del decodificador (K_3) está activa para direcciones de 8K, pero el chip al que está conectado contiene sólo 2K ubicaciones. Como resultado, la EEPROM también responderá a los otros 6K de direcciones en este bloque decodificado de

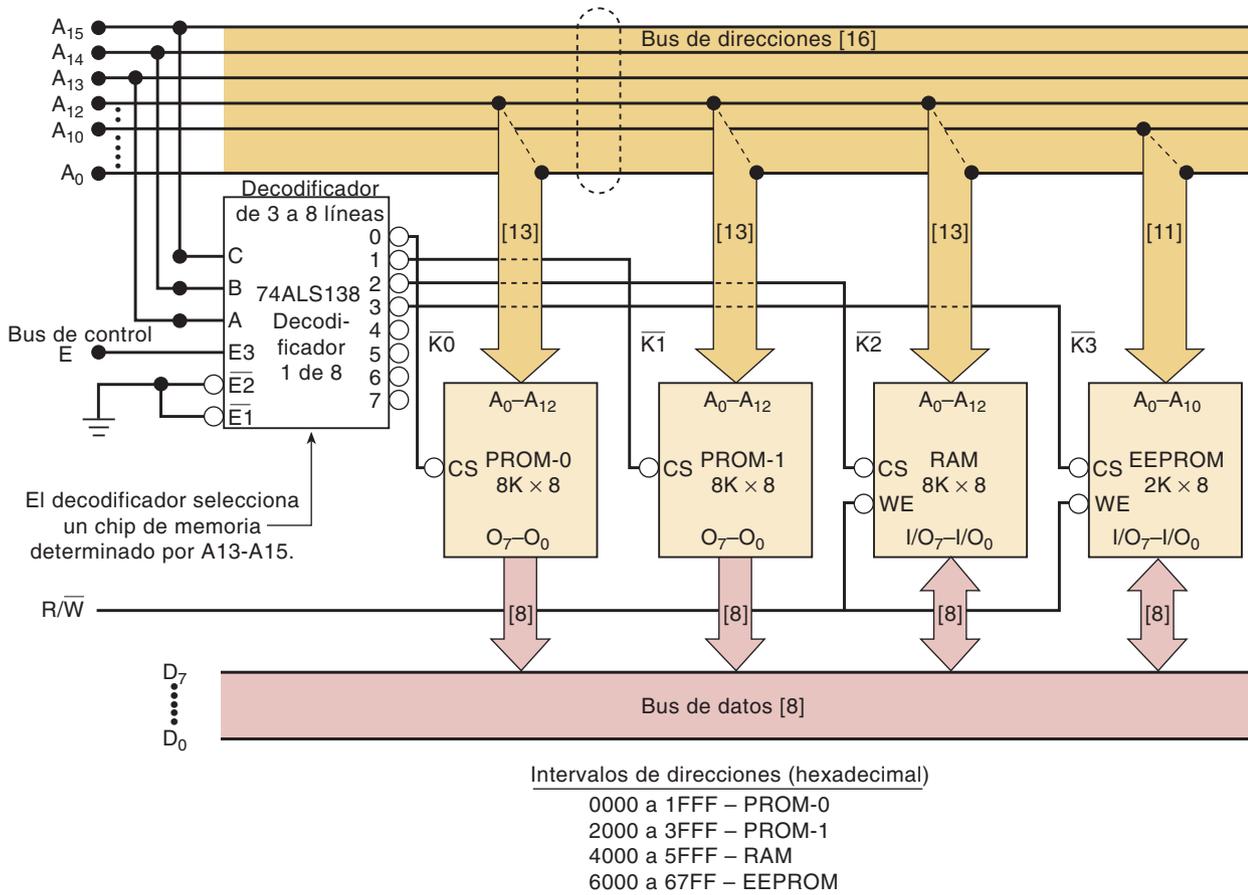
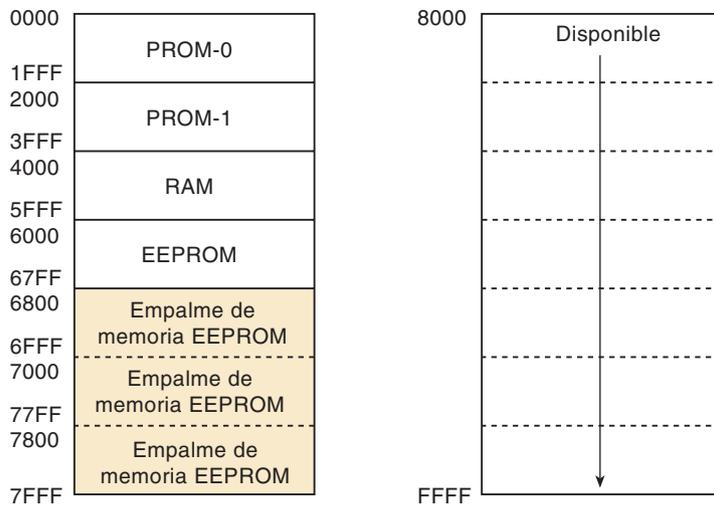


FIGURA 12-38 Un sistema con decodificación incompleta de direcciones.

memoria. El mismo contenido de la EEPROM aparecerá también en las direcciones 6800-6FFF, 7000-77FF y 7800-7FFF. Estas áreas de memoria que se ocupan de manera redundante por un dispositivo debido a la decodificación incompleta de direcciones se conocen como áreas de **empalme de memoria**. Esto ocurre con frecuencia en sistemas en donde hay una abundancia de espacio de direcciones y existe la necesidad de minimizar la lógica de decodificación. Un **mapa de memoria** de este sistema

FIGURA 12-39 Un mapa de memoria de un sistema de tablero de controles digital.



(vea la figura 12-39) muestra con claridad las direcciones a las que está asignado cada dispositivo, así como el espacio de memoria disponible para expansión.

Combinación de chips de DRAM

Por lo general, los CIs de DRAM tienen tamaños de palabra de uno o cuatro bits, por lo que es necesario combinar varios de ellos para formar módulos con tamaños de palabra más grandes. La figura 12-40 muestra cómo combinar ocho chips de DRAM TSM44100 para formar un módulo de $4M \times 8$. Cada chip tiene una capacidad de $4M \times 1$.

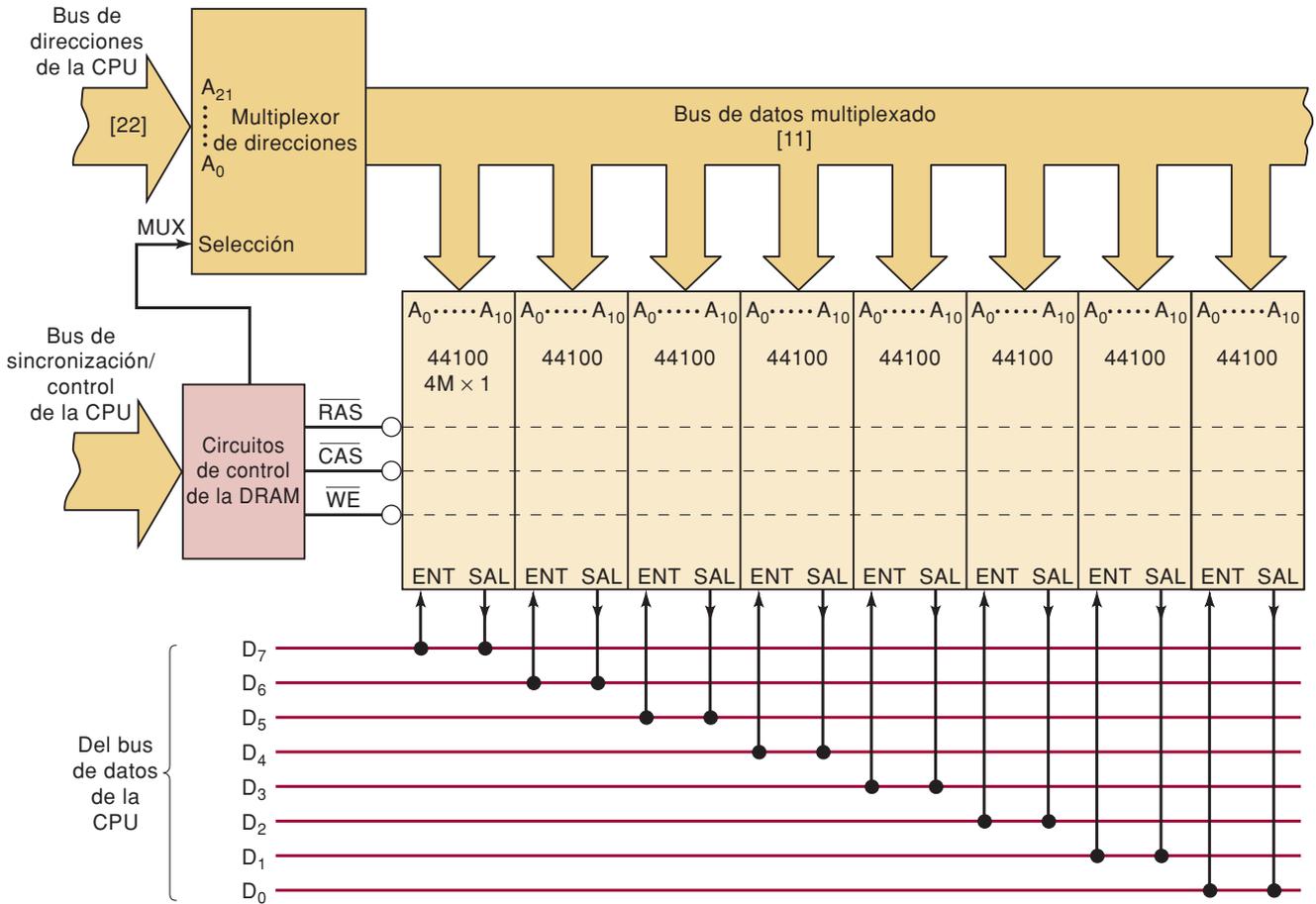


FIGURA 12-40 Ocho chips de DRAM de $4M \times 1$ combinados para formar un módulo de memoria de $4M \times 8$.

Hay varios puntos importantes que observar. En primer lugar, como $4M = 2^{22}$ el chip TSM44100 tiene *once* entradas de dirección; recuerde que las DRAMs utilizan entradas de dirección multiplexadas. El multiplexor de direcciones recibe el bus de direcciones de 22 líneas de la CPU y lo cambia por un bus de direcciones de 11 líneas para los chips de DRAM. En segundo lugar, las entradas RAS, CAS y WE de los ocho chips se conectan entre sí para que todos los chips se activen de manera simultánea para cada operación de memoria. Por último, recuerde que el TSM44100 cuenta con circuitos de control de regeneración integrados en el chip, por lo que no hay necesidad de un contador de regeneración externo.

PREGUNTAS DE REPASO

1. El MCM6209C es un chip de RAM estática de $64K \times 4$. ¿Cuántos de estos chips se necesitan para formar un módulo de $1M \times 4$?
2. ¿Cuántos se necesitan para un módulo de $64K \times 16$?
3. *Verdadero o falso:* cuando se combinan chips de memoria para formar un módulo con un tamaño de palabra o capacidad más grande, las entradas CS de cada chip se conectan siempre juntas.
4. *Verdadero o falso:* cuando se combinan chips de memoria para obtener una capacidad mayor, cada chip se conecta a las mismas líneas del bus de datos.

12-19 FUNCIONES ESPECIALES DE LA MEMORIA

Hemos visto que los dispositivos de RAM y ROM se utilizan como una memoria de computadora interna de alta velocidad, la cual se comunica en forma directa con la CPU (por ejemplo, el microprocesador). En esta sección describiremos con brevedad algunas de las funciones especiales que realizan los dispositivos de memoria semiconductor en las computadoras y en otros equipos y sistemas digitales. Nuestra intención no es proporcionar detalles acerca de cómo se implementan estas funciones, sino introducir las ideas básicas.

Almacenamiento cuando falla la energía

En muchas aplicaciones, la volatilidad de la RAM semiconductor puede indicar la pérdida de datos importantes cuando se desconecta la energía del sistema, ya sea a propósito o como resultado de una interrupción no planeada de energía. A continuación se muestran dos de los muchos ejemplos:

1. Los parámetros críticos de operación para las terminales de gráficos, terminales inteligentes e impresoras. Estos parámetros modificables determinan los modos de operación y los atributos que entrarán en efecto al encender el sistema.
2. Los sistemas de control de procesos industriales que nunca deben “perder su lugar” a mitad de una tarea cuando falla la energía en forma inesperada.

Hay varios métodos para proporcionar el almacenamiento de datos críticos en situaciones de fallo de energía. En uno de estos métodos, todos los datos críticos durante la operación normal del sistema se almacenan en RAM que puede operar a través de baterías de respaldo cuando se interrumpe la energía. Algunos chips de RAM tipo CMOS tienen requerimientos muy bajos de energía en modo suspendido (hasta de 0.5 mW) y se adaptan muy bien para este tipo de tarea. Algunas SRAMs tipo CMOS incluyen una pequeña batería de litio en el chip. Desde luego que aunque tengan un bajo consumo de energía, estas RAMs CMOS drenarán las baterías en un momento dado si se desconecta la energía durante periodos prolongados, y se perderán los datos.

En otro de los métodos se almacenan todos los datos críticos del sistema en memoria Flash no volátil. Este método tiene la ventaja de que no requiere energía mediante batería de respaldo, por lo cual no presenta riesgo de pérdida de datos incluso para fallas de energía de larga duración. Sin embargo, la memoria Flash no puede modificar sus datos con la misma facilidad que la RAM estática. Recuerde que con un chip Flash no podemos borrar y escribir en uno o dos bytes, se debe borrar todo un sector a la vez. Para ello la CPU tiene que reescribir un bloque grande de datos, aún y cuando sólo haya que modificar unos cuantos bytes.

En un tercer método, la CPU almacena todos sus datos en RAM volátil de alta velocidad durante la operación normal del sistema. Cuando se corta la energía, la CPU ejecuta un programa corto para los cortes de energía (desde la ROM), el cual transfiere los datos críticos de la RAM del sistema hacia RAM CMOS con batería de

respaldo o hacia memoria Flash no volátil. Para ello se requiere un circuito especial que detecta la llegada de una interrupción de energía y envía una señal a la CPU para indicarle que comience a ejecutar la secuencia de apagado.

En cualquier caso, cuando regresa la energía la CPU ejecuta un programa de encendido (desde la ROM), el cual transfiere los datos críticos de la memoria de almacenamiento de respaldo hacia la RAM del sistema, de manera que éste puede continuar su operación desde donde se había quedado cuando se interrumpió la energía.

Memoria caché

Las computadoras y demás sistemas digitales pueden tener miles o millones de bytes de memoria interna (RAM y ROM) para almacenar programas y datos que necesita la CPU durante su operación normal. Por lo general, se requiere que toda la memoria interna tenga una velocidad de operación comparable con la de la CPU, para poder lograr la máxima operación del sistema. En muchos sistemas no es económico utilizar dispositivos de memoria de alta velocidad para toda la memoria interna. En vez de ello, los diseñadores utilizan un bloque de memoria caché de alta velocidad. Este bloque de memoria caché es el único bloque que se comunica en forma directa con la CPU en alta velocidad; las instrucciones del programa y los datos se transfieren desde la memoria interna que es más lenta y económica, hacia la memoria caché cuando lo requiere la CPU. El éxito de la memoria caché depende de muchos factores complejos, por lo que algunos sistemas no se benefician del uso de la memoria caché.

Las CPUs de las PCs modernas tienen una pequeña caché de memoria interna (8-64 Kbytes) a la cual se le conoce como caché de nivel 1, o L1. El conjunto de chips de la mayoría de los sistemas computacionales también controla un banco externo de RAM estática (SRAM), el cual implementa una caché de nivel 2, o L2 (de 64 Kbytes a 2 Mbytes). La memoria caché se llena con una secuencia de palabras de instrucciones de la memoria del sistema. La CPU (muchas operan a velocidades de reloj de más de 2 GHz) puede acceder al contenido de la caché a una velocidad muy alta. No obstante, cuando la CPU necesita una pieza de información que no se encuentra en ese momento en la caché L1 o L2 (es decir, una omisión en la caché), debe pasar a la DRAM de sistema más lenta para obtenerla. Esta transferencia debe realizarse a la *velocidad de reloj de bus* mucho más lenta, la cual puede ser de 66 MHz a 800 MHz, dependiendo del sistema. Además de la velocidad de reloj más lenta, el tiempo de acceso de la DRAM (latencia) es mucho mayor.

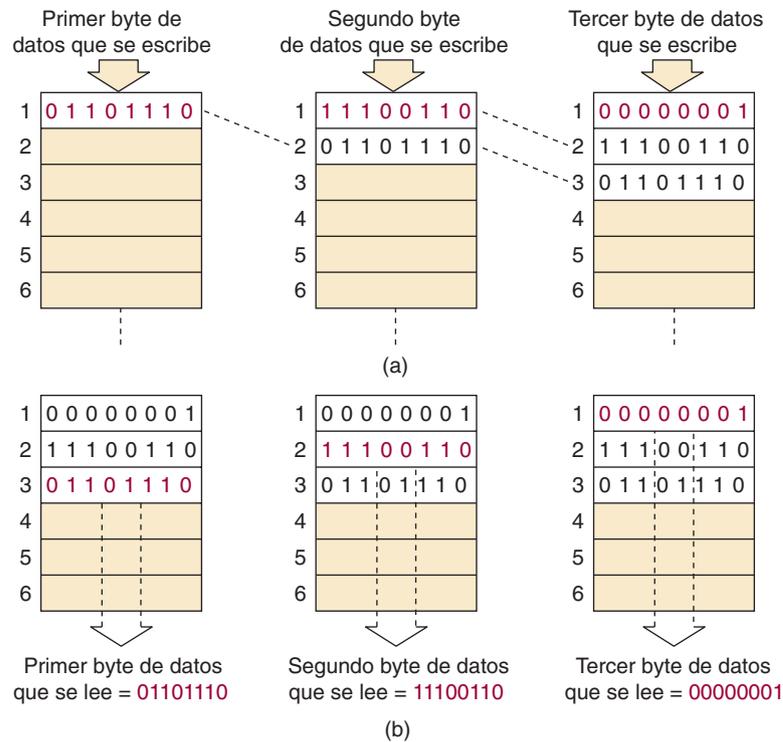
La especificación de 7-2-2-2 o 5-1-1-1 para un sistema de memoria se refiere al número de ciclos de reloj de *bus* necesarios para transferir una ráfaga de cuatro palabras de 64 bits de la DRAM hacia la caché L2. El primer acceso es el que tarda más tiempo debido a la latencia asociada con los ciclos de *RAS/CAS*. Los datos subsiguientes se envían mediante pulsos de reloj en una ráfaga que tarda mucho menos tiempo. Por ejemplo, el sistema 7-2-2-2 requeriría 7 ciclos de reloj para obtener la primera palabra de 64 bits, y cada una de las siguientes tres palabras de 64 bits requerirían 2 ciclos cada una. En total se requieren 13 ciclos de reloj para obtener las cuatro palabras de la memoria.

Memoria tipo Primero en entrar, primero en salir (FIFO)

En los sistemas de memoria FIFO, los datos que se escriben en el área de almacenamiento de la RAM se leen en el mismo orden en el que se escribieron. En otras palabras, la primera palabra que se escribe en el bloque de memoria es la primera palabra que se lee del bloque de memoria: de aquí que se utilice el nombre FIFO. Esta idea se ilustra en la figura 12-41.

La figura 12-41(a) muestra la secuencia al escribir tres bytes de datos en el bloque de memoria. Observe que a medida que se escribe cada nuevo byte en la ubicación 1, los otros bytes avanzan hacia la siguiente ubicación. La figura 12-41(b) muestra la secuencia al leer los datos del bloque FIFO. El primer byte que se lee es el mismo byte que se escribió primero, y así en lo sucesivo. La operación FIFO se controla mediante *registros apuntadores de direcciones* especiales, los cuales llevan el registro de la ubicación en la que se van a escribir los datos y desde dónde se van a leer.

FIGURA 12-41 En FIFO, los valores de los datos se leen de la memoria (b) en el mismo orden en el que se escribieron en la memoria (a).



Una FIFO es útil como **búfer de velocidad (transferencia) de datos** entre sistemas que transfieren datos a muchas velocidades distintas. Un ejemplo es la transferencia de datos de una computadora a una impresora. La computadora envía datos tipo carácter a la impresora, a una velocidad muy alta, por decir un byte cada 10 μ s. Estos datos llenan una memoria FIFO en la impresora, la cual después lee los datos de la FIFO a una velocidad mucho menor, por decir un byte cada 5 ms, e imprime los caracteres correspondientes en el mismo orden que los envió la computadora.

Una FIFO también puede utilizarse como búfer de transferencia de datos entre un dispositivo lento, como un teclado, y una computadora de alta velocidad. Aquí la FIFO acepta datos del teclado a una velocidad lenta y asíncrona de los dedos humanos y los almacena. Después la computadora puede leer con mucha rapidez todas las pulsaciones de tecla recientes que estén almacenadas, en un punto conveniente en su programa. De esta manera, la computadora puede realizar otras tareas mientras la FIFO se llena lentamente de datos.

Búferes circulares

Con frecuencia, a los búferes de transferencia de datos (FIFOs) se les denomina **búferes lineales**. En el momento en el que se llenan todas las ubicaciones en el búfer, no se introducen más datos sino hasta que se vacía el búfer. De esta manera no se pierde la información “antigua”. Hay un sistema de memoria similar que se conoce como **búfer circular**. Estos sistemas de memoria se utilizan para almacenar los últimos n valores introducidos, en donde n es el número de ubicaciones de memoria en el búfer. Cada vez que se escribe un nuevo valor en un búfer circular, sobrescribe (sustituye) el valor más antiguo. Los búferes circulares se direccionan mediante un contador de direcciones MOD- n . En consecuencia, cuando se llegue a la dirección más alta el contador “dará la vuelta” y la siguiente ubicación será la dirección más baja. Como podrá recordar del capítulo 11, el filtrado digital y otras operaciones de DSP realizan cálculos mediante el uso de un grupo de muestras recientes. El hardware especial que se incluye en un DSP permite implementar con facilidad los búferes circulares en la memoria.

PREGUNTAS DE REPASO

1. ¿Cuáles son las diversas formas de manejar la posible pérdida de datos críticos cuando se interrumpe la energía?
2. ¿Cuál es la razón principal de utilizar una memoria caché?
3. ¿Qué significa *FIFO*?
4. ¿Qué es un búfer de transferencia de datos?
5. ¿Qué diferencia hay entre un búfer circular y un búfer lineal?

12-20 DIAGNÓSTICO DE FALLAS EN SISTEMAS DE RAM

Todas las computadoras utilizan RAM. Muchas computadoras de propósito general y la mayoría de las computadoras de propósito especial (como los controladores basados en microprocesador y las computadoras de control de procesos) utilizan también alguna forma de ROM. Cada CI de RAM y de ROM que forma parte de la memoria interna de una computadora, por lo general, contiene miles de celdas de memoria. Una sola celda de memoria defectuosa puede ocasionar una falla completa del sistema (lo que se conoce comúnmente como “caída del sistema”) o cuando menos que el sistema no opere en forma confiable. Los procedimientos de prueba y diagnóstico de fallas en los sistemas de memoria implican el uso de técnicas que no se utilizan con frecuencia en otras partes del sistema digital. Como la memoria consiste de miles de circuitos idénticos que actúan como ubicaciones de almacenamiento, cualquier prueba en cuanto a su operación debe involucrar el proceso de comprobar cuáles ubicaciones exactas están funcionando y cuáles no. Después, si analizamos el patrón de ubicaciones buenas y malas junto con la organización del circuito de memoria, podemos determinar las posibles causas de la falla de la memoria. Por lo general, el problema puede rastrearse hasta llegar a un CI de memoria defectuoso; un CI decodificador defectuoso, compuerta lógica o búfer de señal; o un problema en las conexiones del circuito (es decir, cortos o conexiones abiertas).

Debido a que se debe escribir y leer en la RAM, las pruebas son en general más complejas que las de la ROM. En esta sección veremos algunos procedimientos comunes para realizar pruebas con la porción de RAM de la memoria e interpretar los resultados de la prueba. En la siguiente sección examinaremos las pruebas para la ROM.

Conozca la operación

En nuestros ejemplos utilizaremos el sistema de memoria RAM que muestra la figura 12-42. Como enfatizamos en discusiones anteriores, el diagnóstico de fallas exitoso de un circuito o sistema relativamente complejo comienza con un profundo conocimiento de su operación. Antes de poder hablar sobre cómo probar este sistema de RAM, debemos primero analizarlo con cuidado para poder comprender su operación por completo.

La capacidad total es de $4K \times 8$ y está compuesta de cuatro módulos RAM de $1K \times 8$. Un módulo puede ser un solo CI o puede consistir de varios CIs (por ejemplo, dos chips de $1K \times 4$). Cada módulo se conecta a la CPU mediante los buses de direcciones y de datos, y a través de la línea de control R/\bar{W} . Los módulos tienen líneas de datos de E/S comunes. Durante una operación de lectura, estas líneas se convierten en líneas de salida de datos a través de las cuales el módulo seleccionado coloca sus datos en el bus para que la CPU los lea. Durante una operación de escritura, estas líneas actúan como líneas de entrada para que la memoria acepte los datos generados por la CPU a través del bus de datos, para escribirlos en la ubicación seleccionada.

El decodificador 74ALS138 y la compuerta OR de cuatro entradas se combinan para decodificar las seis líneas de dirección de mayor orden para generar las señales de selección de chip $\bar{K}0$, $\bar{K}1$, $\bar{K}2$ y $\bar{K}3$. Estas señales habilitan un módulo específico de RAM para una operación de lectura o de escritura. El INVERSOR se utiliza para invertir la señal de Habilitación (E) generada por la CPU, de manera que el

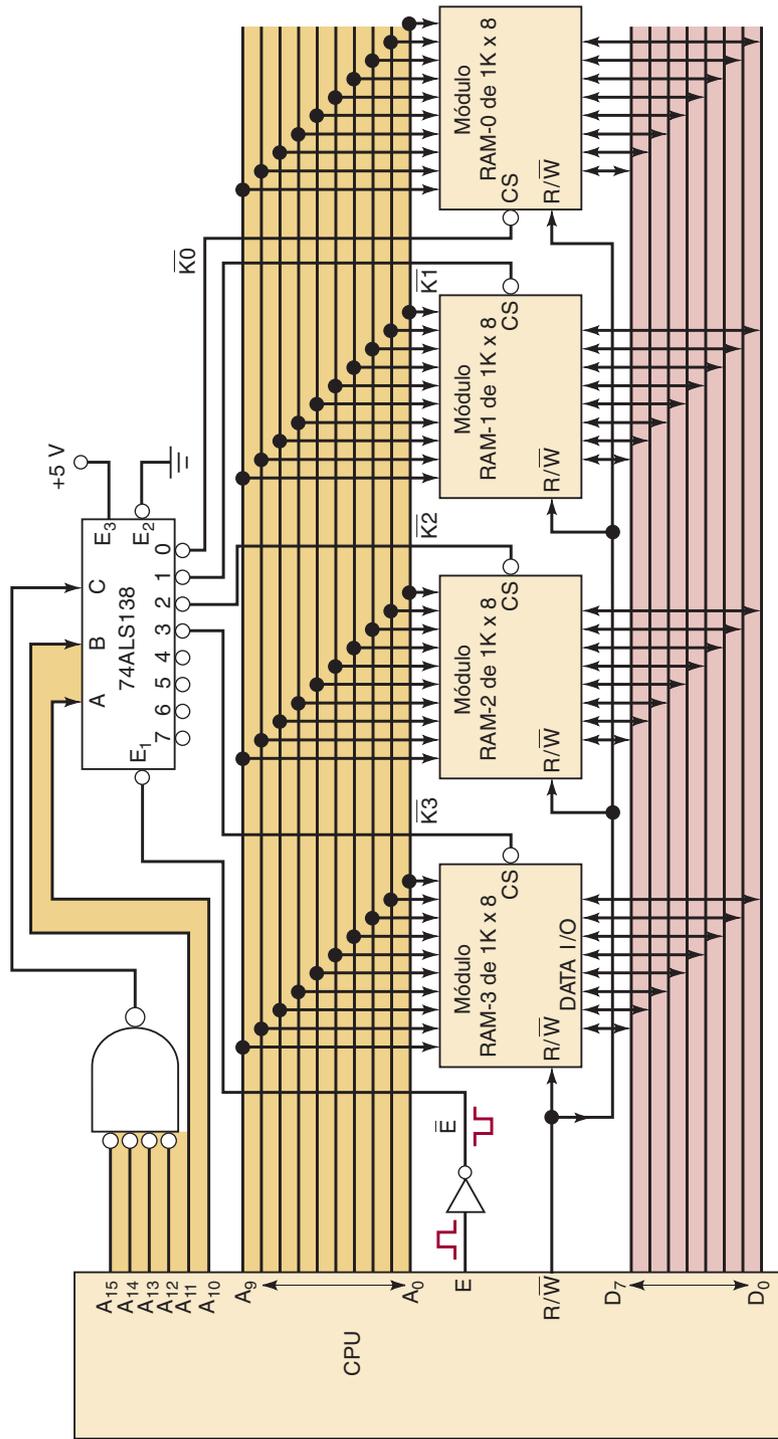
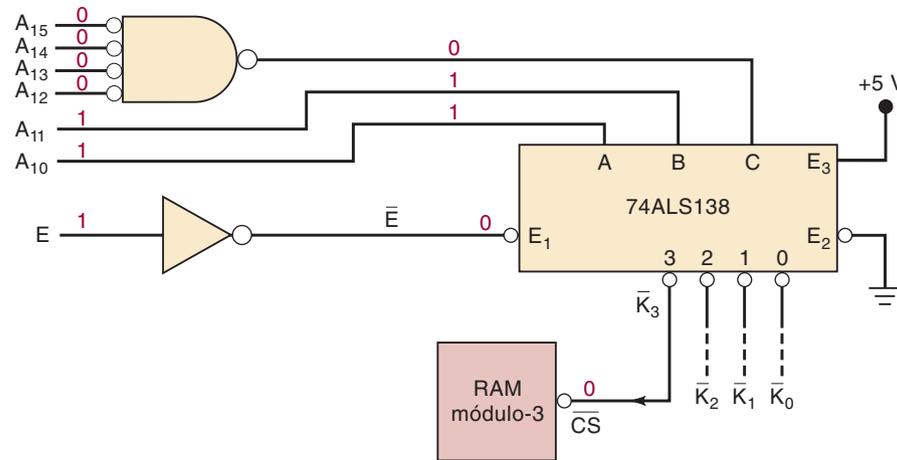


FIGURA 12-42 Memoria RAM de 4K × 8, conectada a una CPU.

FIGURA 12-43

Ejemplo 12-18, en el que se muestran las condiciones del bus de direcciones necesarias para seleccionar el módulo de RAM-3.



Las “x” bajo las entradas A_9 - A_0 indican la condición “no importa”, ya que el decodificador no utiliza estas líneas de dirección para seleccionar el módulo-3. A_0 - A_9 pueden tener cualquier combinación desde 0000000000 hasta 1111111111, dependiendo de cuál sea la palabra en el módulo-3 a la que se esté accediendo. Así, el intervalo completo de direcciones para el módulo-3 se determina mediante el uso de 0s en todas las entradas, y después 1s para todas las “x”.

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	$\rightarrow 0C00_{16}$
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	$\rightarrow 0FFF_{16}$

Por último, esto nos da el intervalo de direcciones hexadecimales almacenadas en el módulo-3 de 0C00H a 0FFFH. Cuando la CPU coloque en el bus de datos cualquier dirección en este intervalo, sólo el módulo-3 se habilitará para una operación ya sea de lectura o de escritura, dependiendo del estado de R/W .

Puede utilizarse un análisis similar para determinar los intervalos de direcciones para cada uno de los otros módulos de RAM. Los resultados son los siguientes:

- Módulo-0: 0000H-03FFH
- Módulo-1: 0400H-07FFH
- Módulo-2: 0800H-0BFFH
- Módulo-3: 0C00H-0FFFH

Observe que los cuatro módulos se combinan para un intervalo total de direcciones de 0000H 0FFFH.

Prueba de la lógica de decodificación

En algunas situaciones, la porción del circuito de RAM correspondiente a la lógica de decodificación (figura 12-43) puede probarse mediante el uso de las diversas técnicas que hemos aplicado a los circuitos combinatoriales. Puede probarse mediante la aplicación de señales a las seis líneas de dirección más significativas y a E , para después monitorear las salidas del decodificador. Para ello, debe ser posible desconectar la CPU con facilidad de estas líneas de señal. Si la CPU es un chip microprocesador en un zócalo, basta con quitarlo de su zócalo.

Una vez que se desconecta la CPU, usted puede suministrar las señales A_{10} - A_{15} y E desde un circuito de prueba externo para realizar una prueba estática, mediante el uso de interruptores operados en forma manual para cada señal, o una prueba dinámica mediante el uso de algún tipo de contador para recorrer en ciclo los diversos códigos de dirección. Al aplicar estas señales de prueba, se pueden comprobar las líneas de salida del decodificador para ver si se obtiene la respuesta apropiada.

Pueden utilizarse técnicas estándar de rastreo de señales para aislar cualquier falla en la lógica de decodificación.

Si usted no tiene acceso a las líneas de dirección del sistema, o si no tiene una manera conveniente de generar las señales lógicas estáticas, a menudo es posible forzar el sistema para que genere una secuencia de direcciones. La mayoría de los sistemas computacionales que se utilizan para el desarrollo tienen un programa almacenado en una ROM, el cual permite al usuario visualizar y modificar el contenido de cualquier ubicación de memoria. Cada vez que la computadora accede a una ubicación de memoria debe colocarse la dirección apropiada en el bus, la cual debe hacer que la salida del decodificador cambie al nivel bajo, incluso aunque sea por un tiempo corto. Escriba la siguiente instrucción para la computadora:

```
Display from 0400H to 07FFH
```

Después coloque la sonda lógica en la salida K1. La sonda lógica deberá mostrar pulsos durante el tiempo en el que se visualicen los valores de datos.

EJEMPLO 12-19

En la lógica de decodificación de la figura 12-43 se realiza una prueba dinámica; para ello se mantiene $E = 1$ y se conectan las salidas de un contador de seis bits a las entradas de dirección A_{10} - A_{15} . Las salidas del decodificador se monitorean a medida que el contador recorre en forma cíclica y repetitiva todos los códigos de seis bits. Una comprobación con sonda lógica en las salidas del decodificador muestra pulsos en $\overline{K1}$ y $\overline{K3}$, pero muestra que $\overline{K0}$ y $\overline{K2}$ permanecen en ALTO. ¿Cuáles son las fallas más probables?

Solución

Es posible, pero muy poco probable, que $\overline{K0}$ y $\overline{K2}$ pudieran estar atascadas en el nivel ALTO debido a un corto interno o externo con V_{CC} . Una falla más probable sería un circuito abierto entre A_{10} y la entrada A del decodificador, ya que esto actuaría como un nivel lógico ALTO y evitaría que se activara cualquier salida del decodificador con numeración par. También es posible que la entrada A del decodificador se ponga en corto con V_{CC} , pero esto también es muy poco probable ya que de seguro este corto habría afectado la operación del contador que suministra las entradas de dirección.

Prueba del sistema completo de RAM

Los procedimientos de prueba y diagnóstico de fallas en la lógica de decodificación no revelarán problemas con los chips de memoria y sus conexiones con los buses de la CPU. Los métodos más comunes para probar la operación del sistema *completo* de RAM implican la escritura de patrones conocidos de 1s y 0s en cada ubicación de memoria, para luego leerlos y verificar que la ubicación haya almacenado los patrones en forma apropiada. Aunque pueden utilizarse muchos patrones distintos, uno de los más populares es el “patrón de tablero de damas”. En este patrón se alternan 1s y 0s, como en 01010101. Una vez que se han probado todas las ubicaciones mediante el uso de este patrón, se invierte (es decir, 10101010) y se prueba cada ubicación otra vez. Hay que recalcar que esta secuencia de pruebas comprobará que cada celda tenga la habilidad de almacenar y leer tanto un 1 como un 0. Debido a que alterna 1s y 0s, el patrón de tablero de damas también detectará cualquier interacción o corto entre celdas adyacentes. Pueden utilizarse muchos otros patrones para detectar varios modos de falla dentro de los chips de RAM.

Ninguna prueba de memoria puede detectar todas las posibles fallas de RAM con un 100 por ciento de precisión, aún y cuando pueda demostrar que cada celda puede almacenar y leer un 0 o un 1. Algunas RAMs defectuosas pueden ser sensibles a los patrones. Por ejemplo, tal vez una RAM pueda almacenar y leer 01010101 y 10101010, pero podría fallar al almacenar 11100011. Incluso para un sistema de

RAM pequeño, se requeriría de un tiempo exageradamente largo para tratar de almacenar y leer todos los patrones posibles en cada ubicación. Por esta razón, si un sistema de RAM pasa la prueba del tablero de damas, puede concluir que *probablemente* está bien; si falla la prueba, entonces *definitivamente* contiene una falla.

El proceso de prueba manual de miles de ubicaciones de RAM mediante el almacenamiento y la lectura de patrones de tablero de damas requeriría cientos de horas, por lo que es obvio que no es factible. La prueba de patrones de RAM se realiza, por lo general, en forma automática, ya sea haciendo que la CPU ejecute un programa de prueba de memoria o conectando un instrumento de prueba especial a los buses del sistema de RAM en lugar de la CPU. De hecho, en muchas computadoras y equipos basados en microprocesador, la CPU ejecuta de manera automática un programa de prueba de memoria cada vez que se enciende; a esto se le conoce como **autoprueba al encendido**. La rutina de autoprueba (la llamaremos AUTOPRUEBA) se almacena en la ROM y se ejecuta cada vez que se enciende el sistema, o cuando el operador la solicita desde el teclado. Cuando el CPU ejecute la AUTOPRUEBA, escribirá los patrones de prueba y leerá esos patrones en/desde cada ubicación de la RAM, y mostrará algún tipo de mensaje al usuario. Puede ser algo tan simple como un LED para indicar una falla en la memoria, o puede ser un mensaje descriptivo impreso en la pantalla o en la impresora. Algunos mensajes típicos podrían ser:

```
RAM module 3 test OK
ALL RAM working properly
Location 027F faulty in bit positions 6 and 7
```

Con mensajes como éstos y un conocimiento de la operación del sistema de RAM, el técnico de diagnóstico de fallas puede determinar qué acción adicional es necesaria para aislar la falla.

PREGUNTAS DE REPASO

1. ¿Cuál es la función de E en el circuito de RAM de la figura 12-42?
2. ¿Cuál es la prueba del tablero de damas? ¿Por qué se utiliza?
3. ¿Qué es una autoprueba al encendido?

12-21 PRUEBA DE LA ROM

El circuito de la ROM en una computadora es muy similar al circuito de la RAM (compare las figuras 12-37 y 12-42). La lógica de decodificación de la ROM puede probarse de la misma forma que describimos en la sección anterior para el sistema de RAM. No obstante, los chips de ROM deben probarse en forma distinta a los chips de RAM, ya que no podemos escribir patrones en la ROM y leerlos de vuelta, como en la RAM. Para comprobar el contenido de un CI de ROM se utilizan varios métodos.

En uno de esos métodos, la ROM se coloca en un zócalo de un instrumento especial de prueba que, por lo general, se controla mediante un microprocesador. El instrumento especial de prueba puede programarse para que lea cada una de las ubicaciones en la ROM de prueba e imprima un listado del contenido de cada ubicación. Después el listado puede compararse con lo que se supone debe contener la ROM. Excepto en los chips de ROM de baja capacidad, esta prueba puede requerir de mucho tiempo.

En un método más eficiente, el instrumento de prueba tiene los datos correctos almacenados en su propio chip de ROM de *referencia*. Después, el instrumento de prueba se programa para que lea el contenido de cada ubicación en la ROM de prueba y lo compare con el contenido de la ROM de referencia. Desde luego que este método requiere la disponibilidad de una ROM de referencia preprogramada.

Un tercer método utiliza una **suma de comprobación**, que es un código especial que se coloca en la última o en las dos últimas ubicaciones del chip de ROM al

FIGURA 12-44
Método de suma de comprobación para una ROM de 8×8 : (a) ROM con los datos correctos; (b) ROM con error en sus datos.

Dirección	Datos	Dirección	Datos
000	00000110	000	00000110
001	10010111	001	10010111
010	00110001	010	00110001
011	11111111	011	1111111 0 ← Error
100	00000000	100	00000000
101	10000001	101	10000001
110	01000110	110	01000110
111	10010100	111	10010100

Suma de comprobación

(a) (b)

momento en que se va a programar. Este código se deriva de la suma de todas las palabras de datos que se van a almacenar en todas las ubicaciones de la ROM (excluyendo las que contienen la suma de comprobación). Cuando el instrumento de prueba lee los datos de cada ubicación de la ROM de prueba, los sumará y desarrollará su propia suma de comprobación. Después comparará la suma de comprobación que calculó con la que está almacenada en las últimas ubicaciones de la ROM, y las dos deberán concordar. De ser así, hay una alta probabilidad de que la ROM esté bien (hay una muy pequeña probabilidad de que una combinación de errores en los datos de la ROM de prueba siga produciendo el mismo valor de la suma de comprobación). Si no concuerdan, hay un problema definitivo en la ROM de prueba.

La idea de la suma de comprobación se ilustra en la figura 12-44(a) para una ROM muy pequeña. La palabra de datos almacenada en la última dirección es la suma de ocho bits de las otras siete palabras de datos (ignorando los acarrees del MSB). Cuando esta ROM se programa, la suma de comprobación se coloca en la última ubicación. La figura 12-44(b) muestra los datos que podrían leerse de una ROM defectuosa que se programó originalmente con los datos de la figura 12-44(a). Observe el error en la palabra de la dirección 011. Cuando el instrumento de prueba lee los datos de cada ubicación de la ROM defectuosa, calcula su propia suma de comprobación con base en esos datos. Debido al error, la suma de comprobación calculada será de 10010011. Cuando el instrumento de prueba compara esta suma con el valor de suma de comprobación almacenado en la ubicación 111 de la ROM, verá que estas dos sumas no concuerdan y se indicará un error en la ROM. Desde luego que no puede determinarse la ubicación exacta del error.

El método de la suma de comprobación también lo puede utilizar una computadora o equipo basado en microprocesador durante una autoprueba automática al encendido, para comprobar el contenido de las ROMs del sistema. De nuevo, como en la autoprueba utilizada para la RAM, la CPU ejecutaría un programa al encenderse, el cual realizaría una prueba de suma de comprobación en cada chip de ROM e imprimiría algún tipo de mensaje de estado. El programa de autoprueba en sí estará ubicado en una ROM, por lo que cualquier error en esa ROM evitaría que se ejecutaran con éxito las pruebas de suma de comparación.

PREGUNTAS DE REPASO

1. ¿Qué es una suma de comprobación? ¿Cuál es su fin?

RESUMEN

1. Todos los dispositivos de memoria almacenan niveles lógicos binarios (1s y 0s) en una estructura tipo arreglo. El tamaño de cada palabra binaria (número de bits) que se almacena varía, dependiendo del dispositivo de memoria. A estos valores binarios se les conoce como *datos*.

2. El lugar (ubicación) en el dispositivo de memoria en donde se almacena cualquier valor de datos se identifica mediante otro número binario, al cual se le conoce como *dirección*. Cada ubicación de memoria tiene una dirección única.
3. Todos los dispositivos de memoria operan de la misma forma general. Para escribir datos en la memoria, la dirección a la cual se va a acceder se coloca en la entrada de dirección, el valor de datos que se va a almacenar se aplica a las entradas de datos y las señales de control se manipulan para almacenar los datos. Para leer datos de la memoria se aplica la dirección, después se manipulan las señales de control y el valor de datos aparece en las terminales de salida.
4. A menudo los dispositivos de memoria se utilizan junto con una CPU de microprocesador que genere las señales de dirección y de control, y que provea los datos que se van a almacenar o que utilice los datos de la memoria. Las operaciones de lectura y de escritura *siempre* se realizan desde la perspectiva de la CPU. Al escribir se colocan datos en la memoria y al leer se obtienen datos de la memoria.
5. La mayoría de las memorias de sólo lectura (ROMs) reciben datos una vez, y de ahí en adelante su contenido no cambia. A este proceso de almacenamiento se le conoce como *programación*. No pierden sus datos cuando se corta la energía del dispositivo. Las MROMs se programan durante el proceso de fabricación. Las PROMs se programan una vez, por medio del usuario. Las EPROMs son justo iguales que las PROMs, sólo que pueden borrarse mediante el uso de luz UV. Las EEPROMs y los dispositivos de memoria Flash pueden borrarse mediante electricidad y alterar su contenido después de la programación. Los CD-ROMs se utilizan para el almacenamiento masivo de información que no necesita cambiar.
6. La memoria de acceso aleatorio (RAM) es un término genérico que se da a los dispositivos que pueden almacenar y recuperar datos con facilidad. Los datos se retienen en un dispositivo de RAM sólo mientras se esté aplicando la energía.
7. La RAM estática (SRAM) utiliza elementos de almacenamiento que son, en esencia, circuitos tipo latch. Una vez que se almacenen los datos, permanecerán sin cambios hasta que se aplique energía al chip. La RAM estática es más fácil de usar, pero más costosa por bit y consume más energía que la RAM dinámica.
8. La RAM dinámica (DRAM) utiliza capacitores para almacenar datos; para ello los carga o los descarga. La simpleza de la celda de almacenamiento permite a las DRAMs almacenar una gran cantidad de datos. Como la carga en los capacitores debe regenerarse con frecuencia, las DRAMs son más complicadas de usar que las SRAMs. A menudo se agregan circuitos extras a los sistemas de DRAM para controlar los ciclos de lectura, de escritura y de regeneración. En muchos de los nuevos dispositivos estas características se están integrando en el mismo chip de DRAM. El objetivo de la tecnología de DRAM es colocar más bits en una pieza de silicio más pequeña, para que consuma menos energía y responda con mayor rapidez.
9. Los sistemas de memoria requieren una amplia variedad de configuraciones distintas. Los chips de memoria pueden combinarse para que implementen cualquier configuración deseada, ya sea que su sistema necesite más bits por ubicación o más capacidad total de palabras. Todos los diversos tipos de ROM y de RAM pueden combinarse dentro del mismo sistema de memoria.

TÉRMINOS IMPORTANTES

apagado
 autopruueba al encendido
 búfer circular
 búfer de transferencia de
 datos
 búfer lineal

bus de control
 bus de datos
 bus de direcciones
 byte
 caché
 capacidad

celda de memoria
 contador de regeneración
 controlador de DRAM
 densidad
 dirección
 empalme de memoria

enlace de fusible	memoria de lectura/ escritura (RWM)	programa de arranque
estrobo de dirección de columna (CAS)	memoria de sólo lectura (ROM)	programación
estrobo de dirección de fila (RAS)	memoria Flash	PROM programable y borrable eléctricamente (EEPROM)
FIFO	memoria principal	RAM dinámica (DRAM)
JEDEC	memoria volátil	RAM estática (SRAM)
latencia	multiplexaje de direcciones	regeneración
mapa de memoria	operación de escritura	regeneración sólo de RAS
memoria auxiliar	operación de lectura	selección de chip
memoria de acceso aleatorio (RAM)	palabra de memoria	suma de comprobación
memoria de acceso secuencial (SAM)		tiempo de acceso
		uso de estrobos

PROBLEMAS

SECCIONES 12-1 A 12-3

- B** 12-1.* Cierta memoria tiene una capacidad de $16K \times 32$. ¿Cuántas palabras almacena? ¿Cuál es el número de bits por palabra? ¿Cuántas celdas de memoria contiene?
- B** 12-2. ¿Cuántas direcciones distintas requiere la memoria del problema 12-1?
- B** 12-3.* ¿Cuál es la capacidad de una memoria que tiene 16 entradas de dirección, cuatro entradas de datos y cuatro salidas de datos?
- B** 12-4. Cierta memoria almacena 8K palabras de 16 bits. ¿Cuántas líneas de entrada y de salida de datos tiene? ¿Cuántas líneas de dirección tiene? ¿Cuál es su capacidad en bytes?

PREGUNTAS DE PRÁCTICA

- 12-5. Defina cada uno de los siguientes términos.
- B**
- RAM.
 - RWM.
 - ROM.
 - Memoria interna.
 - Memoria auxiliar.
 - Capacidad.
 - Volátil.
 - Densidad.
 - Lectura.
 - Escritura.
- 12-6. (a) ¿Cuáles son los tres buses en un sistema de memoria de computadora?
- B**
- ¿Qué bus utiliza la CPU para seleccionar la ubicación de memoria?
 - ¿Cuál bus se utiliza para llevar datos de la memoria a la CPU, durante una operación de lectura?
 - ¿Cuál es el origen de los datos en el bus de datos, durante una operación de escritura?

* Encontrará las respuestas a los problemas marcados con un asterisco al final del libro.

SECCIONES 12-4 Y 12-5

12-7.* Consulte la figura 12-6. Determine las salidas de datos para cada una de las siguientes condiciones de entrada:

- B** (a) $[A] = 1011$; $CS = 1$
 (b) $[A] = 0111$; $CS = 0$

B 12-8. Consulte la figura 12-7.

- (a) ¿Cuál registro se habilita mediante la dirección de entrada 1011?
 (b) ¿Qué código de dirección selecciona el registro 4?

B 12-9.* Cierta ROM tiene una capacidad de $16K \times 4$ y una estructura interna como la que se muestra en la figura 12-7.

- (a) ¿Cuántos registros hay en el arreglo?
 (b) ¿Cuántos bits hay por registro?
 (c) ¿Qué tamaño de decodificadores requiere?

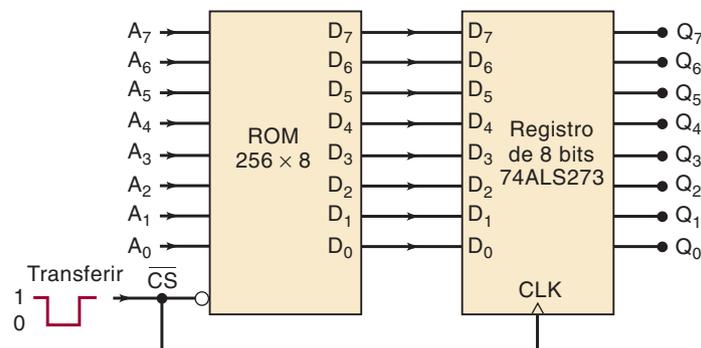
PREGUNTA DE PRÁCTICA

- B** 12-10. (a) *Verdadero o falso*: las ROMs no se pueden borrar.
 (b) ¿Qué significa *programar* o *quemar* una ROM?
 (c) Defina el tiempo de acceso de una ROM.
 (d) ¿Cuántas entradas de datos, salidas de datos y entradas de dirección se necesitan para una ROM de 1024×4 ?
 (e) ¿Cuál es la función de los decodificadores en un chip de ROM?

SECCIÓN 12-6

- A, D** 12-11.* La figura 12-45 muestra cómo pueden transferirse los datos de una ROM hacia un registro externo. La ROM tiene los siguientes parámetros de sincronización: $t_{ACC} = 250$ ns y $t_{OE} = 120$ ns. Suponga que se han aplicado las nuevas entradas de dirección a la ROM 500 ns antes de que ocurra el pulso en TRANSFERIR. Determine la duración mínima del pulso en TRANSFERIR para una transferencia de datos confiable.

FIGURA 12-45
 Problema 12-11.



- A, D** 12-12. Repita el problema 12-11 si las entradas de dirección se cambian 70 ns antes del pulso en TRANSFERIR.

SECCIONES 12-7 Y 12-8

B 12-13. PREGUNTA DE PRÁCTICA

Para cada elemento que se muestra a continuación, indique el tipo de memoria que se está describiendo: MROM, PROM, EPROM, EEPROM, Flash. Algunos elementos corresponderán a más de un tipo.

- (a) El usuario puede programarla pero no se puede borrar.
- (b) El fabricante la programa.
- (c) Es volátil.
- (d) Puede borrarse y reprogramarse una y otra vez.
- (e) Pueden borrarse y reescribirse palabras individuales.
- (f) Se borra con luz UV.
- (g) Se borra mediante electricidad.
- (h) Utiliza enlaces de fusible.
- (i) Puede borrarse en masa o en sectores de 512 bytes.
- (j) No tiene que quitarse del sistema para borrarse y reprogramarse.
- (k) Requiere un voltaje de suministro especial para reprogramarse.
- (l) El tiempo de borrado es aproximadamente de 15 a 20 min.

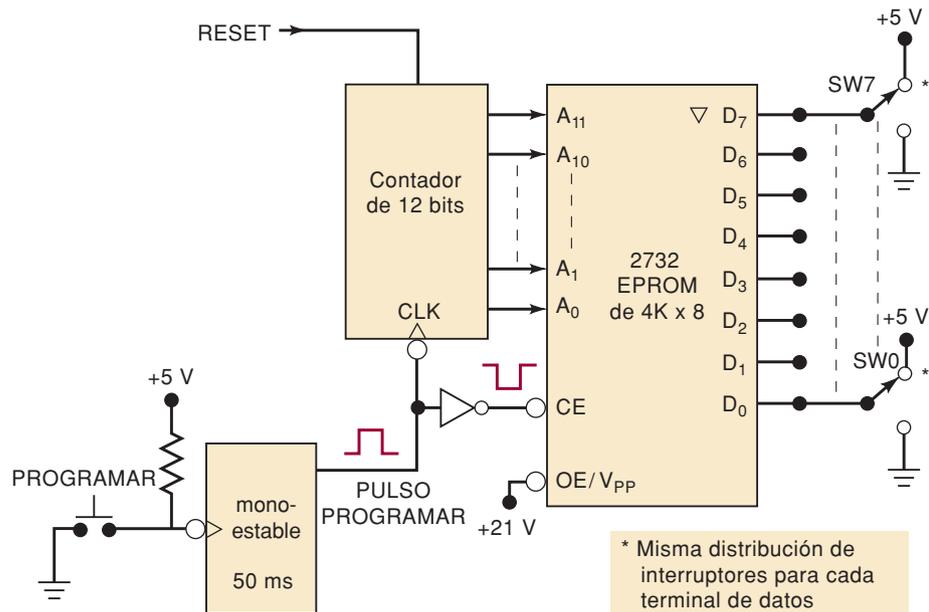
B 12-14. ¿Qué transistores en la figura 12-9 estarán conduciendo cuando $A_1 = A_0 = 1$ y $\overline{EN} = 0$?

12-15.* Cambie las conexiones de la MROM en la figura 12-9, de manera que la MROM almacene la función $y = 3x + 5$.

D 12-16. La figura 12-46 muestra un circuito simple para programar en forma manual una EPROM 2732. Cada una de las terminales de datos de la EPROM está conectada a un interruptor que puede establecerse en un nivel de 1 o de 0. Las entradas de dirección se controlan mediante un contador de 12 bits. El pulso de programación de 50 ms proviene de un monoestable cada vez que se oprime el botón PROGRAMAR.

- (a) Explique cómo puede utilizarse este circuito para programar las ubicaciones de memoria de la EPROM en forma secuencial con los datos deseados.

FIGURA 12-46
Problema 12-16.



- (b) Muestre cómo pueden utilizarse chips 74293 y un 74121 para implementar este circuito.
 - (c) ¿Debería el rebote de interruptores tener algún efecto sobre la operación del circuito?
- N** 12-17.* La figura 12-47 muestra un chip de memoria Flash 28F256A conectado a una CPU a través de un bus de datos y un bus de direcciones. Para escribir/en/leer del arreglo de memoria Flash, la CPU envía la dirección de memoria deseada y genera las señales de control apropiadas para el chip [figura 12-15(b)]. La CPU también puede escribir en el registro de instrucciones del chip (figura 12-16) mediante la generación de las señales de control apropiadas y enviando la instrucción deseada a través del bus de datos. Para esta última operación, la CPU no tiene que enviar una dirección de memoria específica al chip; en otras palabras, las líneas de dirección son “no importa”.
- (a) Considere la siguiente secuencia de operaciones de la CPU. Determine lo que habrá ocurrido a la memoria Flash cuando se complete la secuencia. Suponga que el registro de instrucciones está reteniendo el valor 00_{16} .
 1. La CPU coloca 20_{16} en el bus de datos y aplica pulsos a \overline{CE} y \overline{WE} para que cambien a BAJO mientras que \overline{OE} se mantiene en ALTO. El bus de direcciones está en 0000_{16} .
 2. La CPU repite el paso 1.
 - (b) Después de la secuencia anterior se ha ejecutado, la CPU ejecuta la siguiente secuencia. Determine qué es lo que esto provoca en el chip de memoria Flash.
 1. La CPU coloca 40_{16} en el bus de datos y aplica pulsos a \overline{CE} y \overline{WE} para que cambien a BAJO mientras que \overline{OE} se mantiene en ALTO. El bus de direcciones está en 0000_{16} .
 2. La CPU coloca $3C_{16}$ en el bus de datos y 2003_{16} en el bus de direcciones, y aplica pulsos a \overline{CE} y \overline{WE} para que cambien a BAJO mientras que \overline{OE} se mantiene en ALTO.

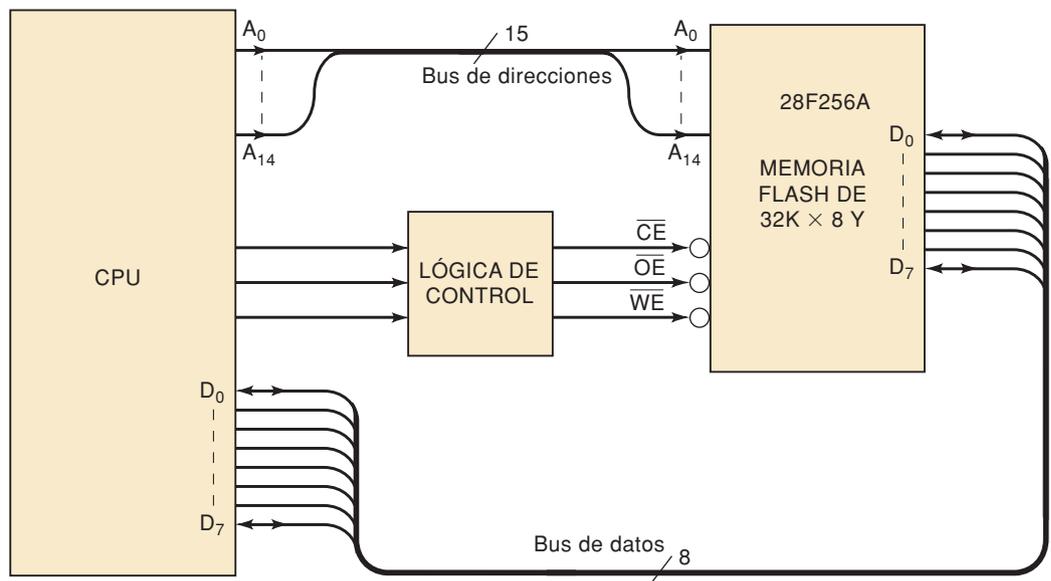
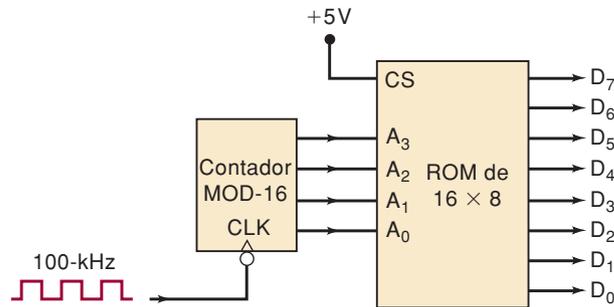


FIGURA 12-47 Problema 12-17.

SECCIÓN 12-9

- N** 12-18. La generación de señales de sincronización y control es otra de las aplicaciones de la ROM. La figura 12-48 muestra una ROM de 16×8 en la que

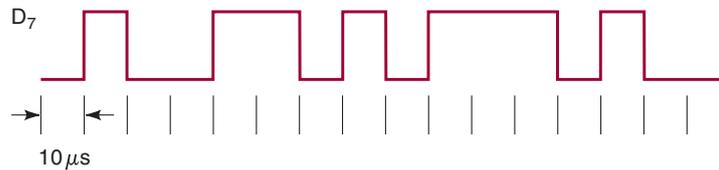
FIGURA 12-48
Problema 12-18



sus entradas de dirección se controlan mediante un contador MOD-16, de manera que se incrementen las direcciones de la ROM en cada pulso de entrada. Suponga que la ROM se programa como en la figura 12-6 y realice un bosquejo de las formas de onda en cada salida de la ROM a medida que se aplican pulsos. Ignore los tiempos de retraso de la ROM. Suponga que el contador inicia en 0000.

- D** 12-19.* Modifique el programa almacenado en la ROM del problema 12-18 para que genere la forma de onda de D_7 de la figura 12-49.

FIGURA 12-49
Problema 12-19.



- D** 12-20.* Consulte el generador de funciones de la figura 12-17.
- ¿Qué frecuencia de reloj producirá una onda senoidal de 100 Hz en la salida?
 - ¿Qué método podría utilizarse para variar la amplitud de pico a pico de la onda senoidal?
- A** 12-21. Para el ML2035 de la figura 12-18, suponga que un valor de 038E (hexadecimal) en latch producirá la frecuencia deseada. Dibuje el diagrama de sincronización para las entradas *LATI*, *SID* y *SCK* y suponga que primero se desplaza el LSB.
- N, A** 12-22.* El sistema que se muestra en la figura 12-50 es un generador de formas de funciones. Utiliza cuatro tablas de búsqueda de 256 puntos en una ROM de 1 Kbyte, para almacenar en cada una un solo ciclo de una onda senoidal (dirección 000-0FF), una rampa con pendiente positiva (dirección 100-1FF), una rampa con pendiente negativa (200-2FF) y una onda triangular (300-3FF). La relación de fase entre los tres canales de salida se controla mediante los valores que se cargan al principio en los tres contadores. Los parámetros críticos de sincronización son $t_{pd(ck-Q \text{ y } OE-Q \text{ máx})}$, contadores = 10 ns, latches = 5 ns y t_{ACC} de ROM = 20 ns. Estudie el diagrama hasta que comprenda cómo funciona y después responda a lo siguiente:
- Si al principio el contador A se carga con 0, ¿qué valores deben cargarse en los contadores B y C, de forma que A se retrase con respecto a B por 90° y que A se retrase con respecto a C por 180° ?
 - Si al principio el contador A se carga con 0, ¿qué valores deben cargarse en los contadores B y C para generar una onda senoidal de tres fases, con un desplazamiento de 120° entre cada salida?
 - ¿Cuál debe ser la frecuencia de pulsos en DAC_SAL para poder generar una salida de onda senoidal de 60 Hz?
 - ¿Cuál es la frecuencia máxima de la entrada CLK?
 - ¿Cuál es la frecuencia máxima de las formas de onda de salida?
 - ¿Cuál es el propósito del contador de selección de función?

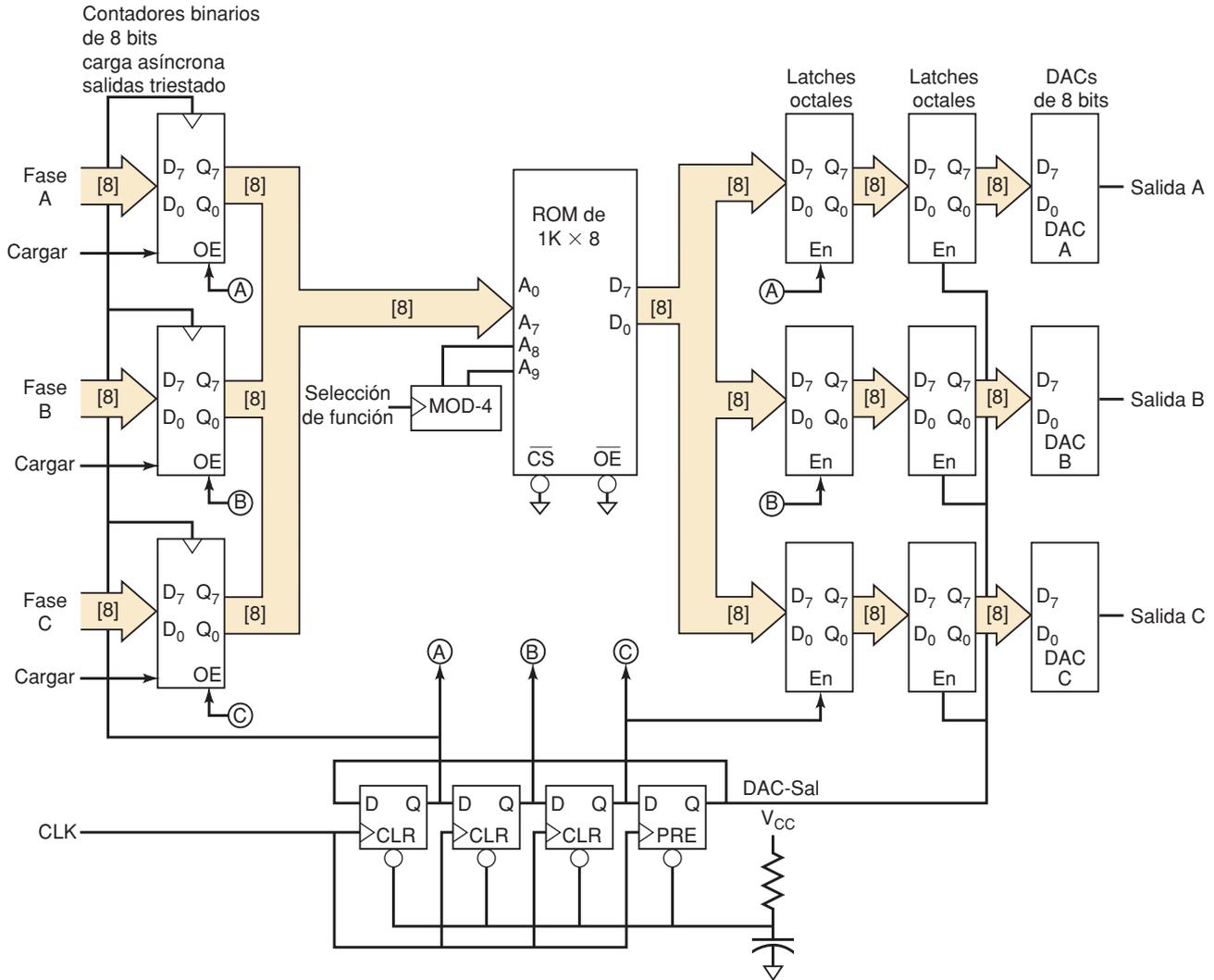


FIGURA 12-50 Problema 12-22.

SECCIÓN 12-11

- 12-23. (a) Dibuje el símbolo lógico para un MCM101514, una RAM estática tipo CMOS organizada como memoria de 256K × 4 con entradas y salidas de datos separadas, y una señal de habilitación de chip activa en BAJO.
- (b) Dibuje el símbolo lógico para un MCM6249, una RAM estática tipo CMOS organizada como memoria de 1M × 4 con E/S común, señal de habilitación de chip activa en BAJO y señal de habilitación de salida activa en BAJO.

SECCIÓN 12-12

12-24.* Cierta RAM estática tiene los siguientes parámetros de sincronización (en nanosegundos):

$t_{RC} = 100$	$t_{AS} = 20$
$t_{ACC} = 100$	$t_{AH} = \text{no se proporciona}$
$t_{CO} = 70$	$t_W = 40$
$t_{OD} = 30$	$t_{DS} = 10$
$t_{WC} = 100$	$t_{DH} = 20$

- (a) ¿Cuánto tiempo después de que se establecen las líneas de dirección aparecerán datos válidos en las salidas, durante un ciclo de lectura?
- (b) ¿Cuánto tiempo permanecerán válidos los datos después de que \overline{CS} regrese al nivel ALTO?
- (c) ¿Cuántas operaciones de lectura pueden realizarse por segundo?
- (d) ¿Cuánto tiempo deberán mantenerse R/\overline{W} y \overline{CS} en ALTO después de que se establezca la nueva dirección, durante un ciclo de escritura?
- (e) ¿Cuál es el mínimo de tiempo que deben permanecer válidos los datos de entrada para que ocurra una operación de escritura confiable?
- (f) ¿Cuánto tiempo deben permanecer estables las entradas de dirección después de que R/\overline{W} y CS regresen al nivel ALTO?
- (g) ¿Cuántas operaciones de escritura pueden realizarse por segundo?

SECCIONES 12-13 A 12-17

12-25. Dibuje el símbolo lógico para el TMS4256, un chip de DRAM de 256K × 1. ¿Cuántas terminales podemos ahorrar si utilizamos el multiplexaje de direcciones para esta DRAM?

- D** 12-26. La figura 12-51(a) muestra un circuito que genera las señales \overline{RAS} , \overline{CAS} y MUX necesarias para que el circuito de la figura 12-28(b) opere en forma apropiada. La señal de reloj maestro de 10 MHz proporciona la sincronización básica para la computadora. La CPU genera la señal de petición de memoria ($MEMR$) en sincronía con el reloj maestro, como se muestra en la parte (b) de la figura. Por lo general $MEMR$ se encuentra en BAJO y se lleva a ALTO cada vez que la CPU quiere acceder a la memoria para una operación de lectura o de escritura. Determine las formas de onda en Q_0 , \overline{Q}_1 y Q_2 , y compárelas con las formas de onda deseadas de la figura 12-29.

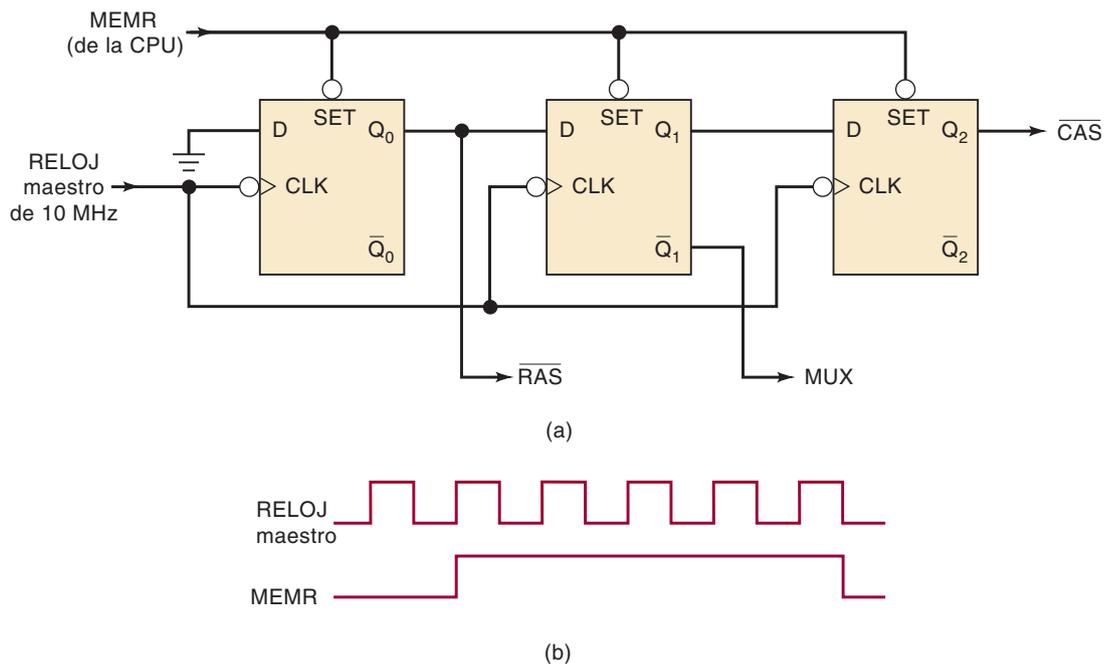


FIGURA 12-51 Problema 12-26.

- D** 12-27. Muestre cómo conectar dos multiplexores 74157 (sección 9-6) para proporcionar la función de multiplexaje requerida en la figura 12-28(b).

12-28. Consulte las señales en la figura 12-30. Describa lo que ocurre en cada uno de los puntos de tiempo etiquetados.

12-29. Repita el problema 12-28 para la figura 12-31.

- A** 12-30.* El chip 21256 es una DRAM de $256K \times 1$ que consiste de un arreglo de 512×512 celdas. Estas celdas deben regenerarse dentro de un intervalo no mayor de 4 ms para que puedan retenerse los datos. La figura 12-33(a) muestra las señales que se utilizan para ejecutar un ciclo de regeneración de \overline{CAS} antes que \overline{RAS} . Cada vez que ocurre un ciclo como éste, los circuitos de regeneración integrados en el chip regenerarán una fila del arreglo en la dirección de fila especificada por un contador de regeneración. El contador se incrementa después de cada regeneración. ¿Con qué frecuencia deben aplicarse los ciclos de \overline{CAS} antes que \overline{RAS} para poder retener todos los datos?
- 12-31.* Estudie el diagrama de bloques funcional de la DRAM TMS44100 en la figura 12-27.
- ¿Cuáles son las dimensiones reales del arreglo de celdas de DRAM?
 - Si el arreglo de celdas fuera cuadrado, ¿cuántas filas habría?
 - ¿Cómo afectaría esto al tiempo de regeneración?

SECCIÓN 12-18

- D** 12-32. Muestre cómo combinar dos chips de RAM 6206 (figura 12-20) para producir un módulo de $32K \times 16$.
- D** 12-33. Muestre cómo conectar dos de los chips de RAM 6264 que se simbolizan en la figura 12-23, para producir un módulo de RAM de $16K \times 8$. El circuito no debe requerir lógica adicional. Dibuje un mapa de memoria que muestre el intervalo de direcciones de cada chip de RAM.
- D** 12-34.* Describa cómo modificar el circuito de la figura 12-37 para que tenga una capacidad total de $16K \times 8$. Use el mismo tipo de chips de PROM.
- D** 12-35. Modifique el circuito decodificador de la figura 12-37 para que opere a partir de un bus de direcciones de 16 líneas (es decir, agregue A_{13} , A_{14} y A_{15}). Las cuatro PROMs deberán mantener los mismos intervalos de direcciones hexadecimales.
- A** 12-36. Para el sistema de memoria de la figura 12-38, suponga que la CPU va a almacenar un byte de datos en la dirección 4000H del sistema.
- ¿En cuál chip está almacenado el byte?
 - ¿Hay alguna otra dirección en este sistema que pueda acceder a este byte de datos?
 - Responda a las partes (a) y (b) suponiendo que la CPU almacenó un byte en la dirección 6007H. (*Sugerencia:* recuerde que la EEPROM no está decodificada por completo.)
 - Suponga que el programa va a almacenar una secuencia de bytes de datos en la EEPROM y que acaba de completar el byte 2048 en la dirección 67FFH. Si el programador le permite almacenar un byte más en la dirección 6800H, ¿cuál será el efecto sobre los primeros 2048 bytes?
- D** 12-37. Dibuje el diagrama completo para una memoria de $256K \times 8$ que utiliza chips de RAM con las siguientes especificaciones: capacidad de $64K \times 4$, línea común de entrada/salida y dos entradas de selección de chip activas en BAJO. [*Sugerencia:* el circuito puede diseñarse utilizando sólo dos inversores (más los chips de memoria).]

SECCIÓN 12-20

- 12-38.* Modifique el circuito de RAM de la figura 12-42 de la siguiente manera: cambie la compuerta OR por una compuerta AND y desconecte su salida de C;

conecte la salida de la compuerta AND a E_3 ; conecte C a tierra. Determine el intervalo de direcciones para cada módulo de RAM.

- A, D** 12-39. Muestre cómo expandir el sistema de la figura 12-42 a una memoria de $8K \times 8$, con un intervalo de direcciones de 0000H a 1FFFH. (*Sugerencia:* puede agregar los módulos de memoria necesarios y modificar la lógica de decodificación existente.)
- F** 12-40.* En la lógica de decodificación de la figura 12-42 se realiza una prueba dinámica; para ello se mantiene $E = 1$ y se conectan las salidas de un contador de seis bits a las entradas de dirección A_{10} - A_{15} . Las salidas del decodificador se monitorean con un osciloscopio (o un analizador lógico), a medida que se aplican pulsos al contador en forma continua mediante un reloj de 1 MHz. La figura 12-52(a) muestra las señales visualizadas. ¿Cuáles son las fallas más probables?

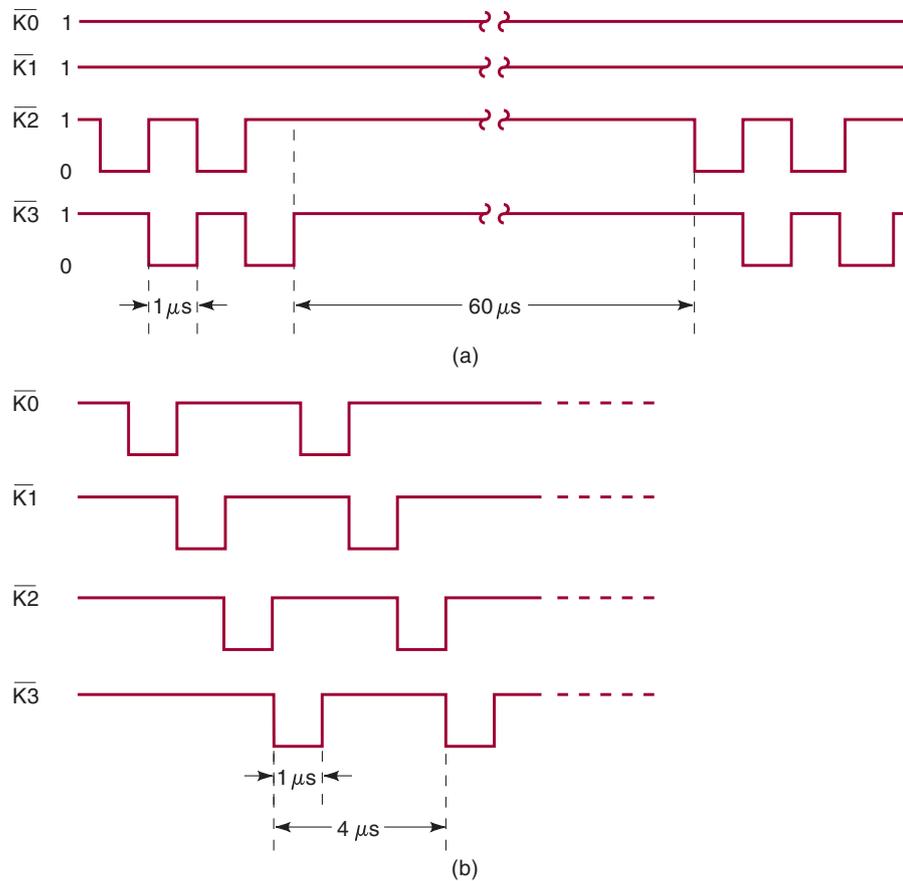


FIGURA 12-52 Problemas 12-40 y 12-41.

- A, F** 12-41. Repita el problema 12-40 para las salidas del decodificador que se muestran en la figura 12-52(b).
- A, D** 12-42.* Considere el sistema de RAM de la figura 12-42. La prueba del patrón de tablero de damas no podrá detectar ciertos tipos de fallas. Por ejemplo, suponga que hay una interrupción en la conexión que va a la entrada A del decodificador. Si se realiza una AUTOPRUEBA de patrón de tablero de damas en este circuito, los mensajes visualizados indicarán que la memoria está bien.
 - (a) Explique por qué no se detectó la falla en el circuito.
 - (b) ¿Cómo modificaría la AUTOPRUEBA, de manera que puedan detectarse fallas como ésta?

- F** 12-43.* Suponga que los módulos de $1K \times 8$ que se utilizan en la figura 12-42 están formados por dos chips de RAM de $1K \times 4$. Los siguientes mensajes se imprimen cuando se realiza la autopruueba al encendido en este sistema de RAM:

```

module-0 test OK
module-1 test OK
address 0800 faulty at bits 4-7
address 0801 faulty at bits 4-7
address 0802 faulty at bits 4-7
. . . . .
. . . . .
address 0BFE faulty at bits 4-7
address 0BFF faulty at bits 4-7
module-3 test OK

```

Examine estos mensajes y liste las posibles fallas.

- F** 12-44.* Los siguientes mensajes se imprimen cuando se realiza la autopruueba al encendido en el sistema de RAM de la figura 12-42.

```

module-0 test OK
module-1 test OK
module-2 test OK
address 0C00 faulty at bit 7
address 0C01 faulty at bit 7
address 0C02 faulty at bit 7
. . . . .
. . . . .
address 0FFE faulty at bit 7
address 0FFF faulty at bit 7

```

Examine estos mensajes y liste las posibles fallas.

- F** 12-45. ¿Qué mensajes se imprimirían si se realizara una autopruueba al encendido en el sistema de RAM de la figura 12-42, si hubiera un corto entre las salidas $\overline{K2}$ y $\overline{K3}$ del decodificador?

SECCIÓN 12-21

- F** 12-46.* Considere la ROM de 16×8 en la figura 12-6. Sustituya la palabra de datos almacenada en la ubicación de dirección 1111 con una suma de comprobación calculada con base en las otras 15 palabras de datos.

RESPUESTAS A LAS PREGUNTAS DE REPASO DE SECCIÓN

SECCIÓN 12-1

1. Vea el texto.
 2. 16 bits por palabra; 8192 palabras; 131,072 bits o celdas
 3. En una operación de lectura, se obtiene una palabra de una ubicación de memoria y se transfiere hacia otro dispositivo. En una operación de escritura, se coloca una nueva palabra en una ubicación de memoria y se sustituye la que estaba almacenada ahí.
 4. Verdadero
 5. SAM: el tiempo de acceso no es constante, sino que depende de la ubicación física de la palabra a la que se esté accediendo. RAM: el tiempo de acceso es el mismo para cualquier ubicación de dirección.
 6. RWM es memoria en la que se puede leer o escribir con igual facilidad. ROM es memoria que, por lo general, se lee y casi nunca se escribe.
 7. Falso; sus datos deben regenerarse en forma periódica.
-

SECCIÓN 12-2

1. 14, 12, 12
2. Indica a la memoria que debe realizar una operación de lectura o de escritura.
3. Cuando se encuentra en su estado activo, esta entrada habilita a la memoria para que realice la operación de lectura o de escritura seleccionada por la entrada R/\overline{W} . Cuando se encuentra en su estado inactivo, esta entrada deshabilita a la memoria para que no pueda realizar la función de lectura o la de escritura.

SECCIÓN 12-3

1. Líneas de dirección, líneas de datos, líneas de control
2. Vea el texto.
3. Vea el texto.

SECCIÓN 12-4

1. Verdadero
2. Aplicar las entradas de dirección deseadas; activar la(s) entrada(s) de control; los datos aparecen en las salidas de datos.
3. El proceso de introducir datos en la ROM.

SECCIÓN 12-5

1. $A_3A_2A_1A_0 = 1001$
2. El decodificador de selección de fila activa una de las entradas de habilitación de todos los registros en la fila seleccionada. El decodificador de selección de columna activa una de las entradas de habilitación de todos los registros en la columna seleccionada. Los búferes de salida pasan los datos del bus de datos interno hacia las terminales de salida de la ROM cuando la terminal CS está activada.

SECCIÓN 12-7

1. Falso; por el fabricante.
2. Una PROM puede programarse una vez por el usuario. No puede borrarse y reprogramarse.
3. Verdadero
4. Mediante la exposición a la luz UV.
5. Verdadero.
6. Programan los datos de manera automática en las celdas de memoria, una dirección a la vez.
7. Una EEPROM puede borrarse y reprogramarse mediante electricidad sin tener que quitarla de su circuito, y puede borrarse byte por byte.
8. Baja densidad; alto costo.
9. EEPROM.
10. Uno.

SECCIÓN 12-8

1. Se puede borrar y programar mediante electricidad sin quitarla del circuito.
2. Mayor densidad; menor costo.
3. Tiempos de borrado y programación cortos.
4. Para las operaciones de borrado y programación.
5. El contenido de este registro controla todas las funciones internas del chip.
6. Para confirmar que se ha borrado una dirección de memoria con éxito (es decir, datos = todos 1s).
7. Para confirmar que se ha programado una dirección de memoria con los datos correctos.

SECCIÓN 12-9

1. Al encendido, la computadora ejecuta un pequeño programa de arranque desde la ROM para inicializar el hardware del sistema y cargar el sistema operativo desde la unidad de almacenamiento masivo (disco).
2. Circuito que recibe datos representados en un tipo de código y los convierte en otro tipo de código.
3. Contador, ROM, DAC, filtro paso bajo.
4. Son no volátiles, rápidos, confiables, pequeños y consumen poca energía.

SECCIÓN 12-11

1. Se aplica la dirección deseada a las entradas de dirección; $R/\overline{W} = 1$; se activa CS o CE.
2. Para reducir el conteo de terminales.
3. 24, incluyendo V_{CC} y tierra

SECCIÓN 12-12

1. Las celdas de SRAM son flip-flops; las celdas de DRAM utilizan capacitores.
2. CMOS
3. Memoria.
4. CPU.
5. Los tiempos de los ciclos de lectura y escritura.
6. Falso; cuando \overline{WE} está en BAJO, las terminales de E/S actúan como entradas de datos, sin importar el estado de \overline{OE} (segunda entrada en la tabla de modos).
7. A_{13} puede permanecer conectada a la terminal 26. A_{14} debe quitarse y la terminal 27 debe conectarse a +5 V.

SECCIÓN 12-13

1. Una velocidad, por lo general, más lenta; necesidad de regenerarse. de energía; alta capacidad; menor costo por bit.
2. Bajo consumo
3. DRAM.

SECCIÓN 12-14

1. 256 filas \times 256 columnas. Necesita 10 entradas de dirección.
2. Ahorra terminales en el chip.
3. $1M = 1024K = 1024 \times 1024$. Por lo tanto, hay 1024 filas por 1024 columnas. Como $1024 = 2^{10}$, el chip
4. \overline{RAS} se utiliza para fijar la dirección de fila en el registro de dirección de fila de la DRAM. \overline{CAS} se utiliza para fijar la dirección de columna en el registro de dirección de columna.
5. MUX multiplexa la dirección completa en las direcciones de fila y de columna para introducirlas a la RAM.

SECCIÓN 12-15

1. (a) Verdadero (b) falso (c) falso (d) verdadero
2. MUX

SECCIÓN 12-16

1. (a) Verdadero (b) falso.
2. Proporciona las direcciones de fila a la DRAM durante los ciclos de regeneración.
3. Multiplexaje de direcciones y la operación de regeneración.
4. (a) Falso (b) verdadero.

SECCIÓN 12-17

1. No
2. Ubicaciones de memoria con la misma dirección superior (misma fila).
3. Sólo debe fijarse la dirección de columna.
4. Salida de datos extendida.
5. *Ráfaga*
6. El reloj del sistema.

SECCIÓN 12-18

1. Dieciséis
2. Cuatro
3. Falso; cuando se expande la capacidad de la memoria, cada chip se selecciona mediante una salida distinta del decodificador (vea la figura 12-43).
4. Verdadero

SECCIÓN 12-19

1. Respaldo de batería para la RAM tipo CMOS; memoria Flash.
2. La economía
3. Los datos se leen de la memoria en el mismo orden en el que se escribieron.
4. Una FIFO que se utiliza para transferir datos entre dispositivos con velocidades de operación muy distintas.
5. Los búferes circulares “dan la vuelta” de la dirección más alta hasta la dirección más baja, y los datos más recientes siempre sobrescriben a los datos más antiguos.

SECCIÓN 12-20

1. Evita las deformaciones en la decodificación al deshabilitar el decodificador mientras cambian las líneas de dirección.
2. Una manera de probar la RAM mediante la escritura de un patrón de tablero de damas (primero 01010101, después 10101010) en cada ubicación de memoria, para después leerlo. Se utiliza porque detectará cualquier corto o interacción entre las celdas adyacentes.
3. Una prueba automática de la RAM, la cual se lleva a cabo mediante la computadora al momento del encendido.

SECCIÓN 12-21

1. Un código que se coloca en la última o en las últimas dos ubicaciones de la ROM, el cual representa la suma de los datos esperados en la ROM para todas las demás ubicaciones. Se utiliza como un medio para probar si hay errores en una o más ubicaciones de la ROM.