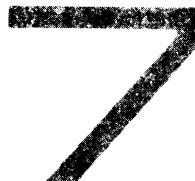


# Registros, contadores y unidad de memoria



## 7-1 INTRODUCCION

Un circuito secuencial temporizado consta de un grupo de flip-flops y compuertas combinacionales conectadas para formar una trayectoria de retroalimentación. Los flip-flops son esenciales porque, cuando están ausentes, el circuito se reduce a un circuito combinacional puro (siempre que no haya trayectoria de retroalimentación). Un circuito solo con flip-flops se considera un circuito secuencial incluso cuando están ausentes las compuertas combinacionales.

Un circuito MSI que contiene celdas de almacenamiento en su interior es, por definición, un circuito secuencial. Los circuitos MSI que incluyen flip-flops u otras celdas de almacenamiento por lo común se clasifican según la función que realizan más que por el nombre "circuito secuencial". Estos circuitos MSI se clasifican en una de tres categorías: registros, contadores o memoria de acceso aleatorio. En este capítulo se presentan varios registros y contadores disponibles en la forma IC y se explica su operación. También se presenta la organización de la memoria de acceso aleatorio.

Un *registro* es un grupo de celdas de almacenamiento binario adecuadas para mantener información binaria. Un grupo de flip-flops constituye un registro, ya que cada flip-flop es una celda binaria capaz de almacenar un bit de información. Un registro de  $n$ -bit tiene un grupo de  $n$  flip-flops y es capaz de almacenar cualquier información binaria que contenga  $n$  bits. Además de los flip-flops, un registro puede tener compuertas combinacionales que realicen ciertas tareas de procesamiento de datos. En su definición más amplia, un registro consta de un grupo de flip-flops y compuertas que efectúan su transición. Los flip-flops mantienen información binaria y las compuertas controlan cuando y cómo se transfiere información nueva al registro.

Los contadores están en la Sección 6-8. Un contador es un registro que pasa a través de una secuencia determinada de estados bajo la aplicación de pulsos de entrada. Las compuertas en un contador están conectadas de tal forma que producen una secuencia prescrita de estados binarios en el registro. Aunque los contadores son un tipo especial de registro, es costumbre diferenciarlos dándoles un nombre especial.

Una unidad de memoria es una colección de celdas de almacenamiento junto con circuitos asociados necesarios para transferir la información de entrada y salida

del almacenamiento. Una memoria de acceso aleatorio (RAM) difiere de una memoria de sólo lectura (ROM), en que una RAM puede transferir la información almacenada a una salida (lectura) y también es capaz de recibir información nueva de entrada para almacenamiento (escritura). Un nombre más apropiado para dicha memoria sería *memoria de lectura-escritura*.

Los registros, contadores y memorias tienen amplia aplicación en el diseño de sistemas digitales en general y en computadoras digitales en particular. Los registros también pueden usarse para facilitar el diseño de circuitos secuenciales. Los contadores son útiles para generar variables temporizadoras para las operaciones de secuencia y control en un sistema digital. Las memorias son esenciales para el almacenamiento de programas e información en una computadora digital. El conocimiento de la operación de estos componentes es indispensable para comprender la organización y diseño de los sistemas digitales.

## 7-2 REGISTROS

Están disponibles varios tipos de registros en la forma de circuitos MSI. El registro más simple posible es el que consta de flip-flops solos sin ninguna compuerta externa. En la Fig. 7-1 se muestra uno de dichos registros construido con cuatro flip-flops tipo *D* y una entrada común de pulsos de reloj. La entrada de pulsos de reloj, *CP*, capacita todos los flip-flops, de modo que la información presente disponible en las cuatro entradas puede transferirse al registro de 4-bit. Las cuatro salidas pueden muestrearse para obtener la información almacenada en ese momento en el registro.

La forma en que los flip-flops en un registro se disparan es de importancia primaria. Si los flip-flops están construidos con seguros de compuertas tipo *D* como en la Fig. 6-5, entonces la información presentada en una entrada de información (*D*) se transfiere a la salida *Q* cuando el pulso de habilitación (*CP*) es 1, y la salida *Q* sigue los datos de entrada en tanto que la señal *CP* permanezca 1. Cuando *CP* pasa a 0, la información que estaba presente en la entrada de información precisamente antes de la transición se retiene en la salida *Q*. En otras palabras, los flip-flops son sensativos a la duración del pulso y el registro se habilita mientras *CP* = 1. Un registro que responde a la duración del pulso en forma común se denomina *compuerta con seguro* y la entrada

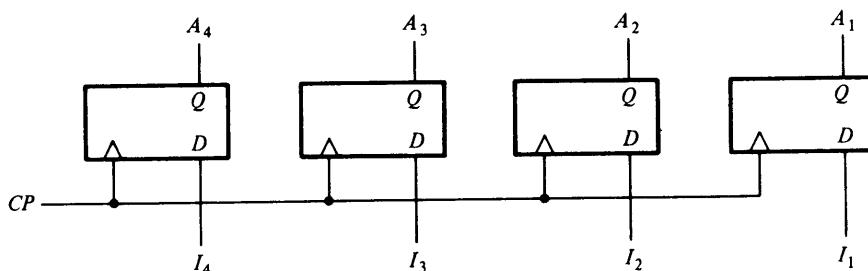


Figura 7-1 Registro de 4-bit.

$CP$  con frecuencia se etiqueta con la variable  $G$  (en lugar de  $CP$ ). Los seguros son adecuados para usarse como almacenamiento temporal de información binaria que va a transferirse a un destino externo. No deben utilizarse en el diseño de circuitos secuenciales que tienen conexiones de retroalimentación.

Como se explicó en la Sección 6-3, un flip-flop puede usarse en el diseño de circuitos secuenciales temporizados siempre que sea sensitivo a la transición de pulso más que a la duración del pulso. Esto significa que los flip-flops en el registro deben ser del tipo de disparo de borde o maestro-esclavo. En forma normal, no es posible distinguir mediante un diagrama lógico cuando un flip-flop es de seguro con compuerta, disparado por borde, o maestro-esclavo, debido a que los símbolos gráficos para todos los tres son iguales. La distinción debe hacerse en el nombre dado a la unidad. Un grupo de flip-flops sensitivos a la duración del pulso por lo común se denomina como *seguro*, mientras que un grupo de flip-flops sensitivo a una transición de pulso se conoce como *registro*.\* Un registro siempre puede reemplazar a un seguro, pero la situación inversa debe tomarse con precaución para tener la seguridad de que las salidas de un seguro nunca van a otras entradas flip-flop que se disparan con el mismo pulso de reloj común. En las exposiciones que siguen, se supone que cualquier grupo de flip-flops que se dibuje constituye un *registro* y que todos los flip-flops son de los tipos de disparo por borde o maestro-esclavo. Si el registro es sensitivo a la duración del pulso, se referirá como un *seguro*.

### Registro con carga paralela

La transferencia de información nueva a un registro se conoce como *cargar* el registro. Si todos los bits del registro se cargan en forma simultánea con un solo pulso de reloj, se dice que la carga se hace en paralelo. Un pulso aplicado a la entrada  $CP$  del registro en la Fig. 7-1 cargará las cuatro entradas en paralelo. En esta configuración, el pulso de reloj debe inhibirse desde la terminal  $CP$  si el contenido del registro debe dejarse sin cambio. En otras palabras, la entrada  $CP$  actúa como una señal de habilitación que controla la carga de nueva información en el registro. Cuando  $CP$  pasa a 1, la información de entrada se carga en el registro. Si  $CP$  permanece en 0, el contenido del registro no se cambia. Obsérvese que el cambio de estado en las salidas ocurre en el borde positivo del pulso. Si un flip-flop cambia de estado al borde negativo, habrá un pequeño círculo bajo el símbolo de triángulo en la entrada  $CP$  del flip-flop.

La mayoría de los sistemas digitales tienen un reloj generador maestro que suministra un tren continuo de pulsos de reloj. Todos los pulsos de reloj se aplican a los flip-flops y registros en el sistema. El reloj generador maestro actúa como una bomba que suministra un ritmo constante a todas las partes del sistema. Una señal separada de control decide entonces qué pulsos específicos de reloj tendrán un efecto en un registro particular. En dicho sistema, los pulsos de reloj deben combinarse con una compuerta AND con la señal de control, y la salida de la compuerta AND se aplica entonces a la terminal  $CP$  del registro que se muestra en la Fig. 7-1. Cuando la señal de control es 0, la salida de la compuerta AND es 0 y la información almacenada en el

\* Por ejemplo, el IC tipo 7475 es un seguro de 4-bit, en tanto el tipo 74175 es un registro de 4-bit.

registro permanece sin cambio. Sólo cuando la señal de control es un 1 el pulso de reloj pasa a través de la compuerta AND y a la terminal *CP* para cargar información nueva dentro del registro. Tal variable de control se denomina control de entrada de *carga*.

La inserción de una compuerta AND en la trayectoria de pulsos de reloj significa que la lógica se lleva a cabo con pulsos de reloj. La inserción de compuertas lógicas produce retardos de propagación entre el reloj generador maestro y las entradas de reloj de los flip-flops. Para sincronizar el sistema en forma completa, debe tenerse la seguridad de que todos los pulsos de reloj llegan al mismo tiempo a todas las entradas de todos los flip-flops, de modo que todos puedan cambiar en forma simultánea. La realización de la lógica con pulsos de reloj inserta retardos variables y puede lanzar al sistema fuera de sincronismo. Por esta razón, es aconsejable (pero no necesario, en tanto los retardos se toman en consideración) aplicar los pulsos de reloj directamente a todos los flip-flops y controlar la operación del registro con otras entradas, como las entradas *R* y *S* de un flip-flop *RS*.

Un registro de 4-bit con una entrada de control de carga usando flip-flops *RS* se muestra en la Fig. 7-2. La entrada *CP* del registro recibe pulsos sincronizados continuos que se aplican a todos los flip-flops. El inversor en la trayectoria *CP* provoca que todos los flip-flops se disparen por el borde negativo de los pulsos que llegan. El objetivo del inversor es reducir la carga del reloj generador maestro. Esto se debe a que la entrada *CP* está conectada sólo a una compuerta (el inversor) en lugar de las cuatro entradas de compuertas que se requerirían si las conexiones se hicieran directamente en las entradas de reloj flip-flop (marcadas con triángulos pequeños).

La entrada de *despeje* va a una terminal especial en cada flip-flop a través de una compuerta buffer no inversora. Cuando esta terminal pasa a 0, el flip-flop se despeja en forma asíncrona. La entrada de despeje es útil para dejar el registro de todos los 0 antes de su operación temporizada. La entrada de despeje debe mantenerse en 1 durante las operaciones temporizadas normales (véase la Fig. 6-14).

La entrada de *carga* pasa a través de una compuerta buffer (para reducir el cargado) y a través de una serie de compuertas AND a las entradas *R* y *S* de cada flip-flop. Aunque están presentes en forma continua pulsos de reloj, la entrada de carga es la que controla la operación del registro. Las dos compuertas AND y el inversor asociado con cada entrada IC determinan los valores de *R* y *S*. Si la carga de entrada es 0, tanto *R* como *S* son 0, y no ocurre cambio de estado con ningún pulso de reloj. Por consiguiente, la carga de entrada es una variable de control que puede evitar cualquier cambio de información en el registro en tanto que su entrada sea 0. Cuando el control de carga pasa a 1, las entradas  $I_1$  a la  $I_4$  especifican qué información binaria se carga en el registro en el siguiente pulso de reloj. Para cada *I* que es igual a 1, las entradas flip-flop correspondientes son *S* = 1, *R* = 0. Para cada *I* que es igual a 0, las entradas flip-flop correspondientes son *S* = 0, *R* = 1. Por eso, el valor de entrada se transfiere al registro siempre que la carga de entrada es 1. La entrada de despeje es 1 y un pulso de reloj pasa de 1 a 0. Este tipo de transferencia se denomina transferencia *carga-paralela*, ya que todos los bits del registro se cargan en forma simultánea. Si la compuerta buffer asociada con la entrada de carga se cambia a una compuerta inversora, entonces el registro se carga cuando la entrada de carga es 0 y se inhibe cuando la entrada de carga es 1.

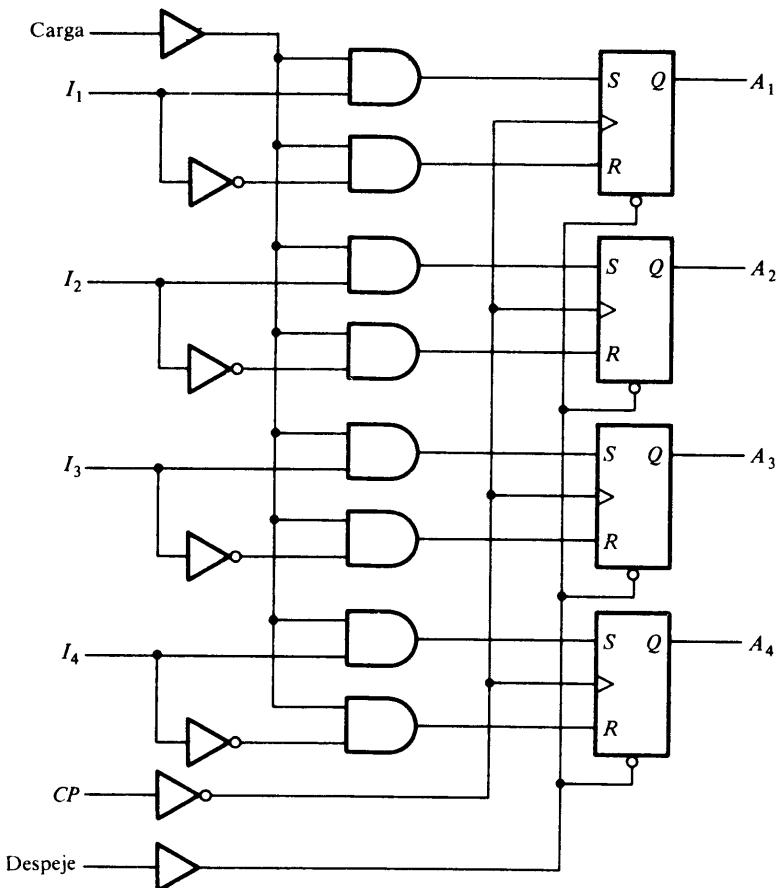


Figura 7-2 Registro de 4-bit con carga paralela.

Un registro con carga paralela puede construirse con flip-flops *D* como se muestra en la Fig. 7-3. Las entradas de reloj y despeje son las mismas que antes. Cuando la entrada de carga es 1, las entradas *I* se transfieren al registro en el siguiente pulso de reloj. Cuando la entrada de carga es 0, las entradas del circuito están inhibidas y los flip-flops *D* están recargados con el estado presente, manteniendo por tanto el contenido del registro. La conexión de retroalimentación en cada flip-flop es necesaria cuando se usa el tipo *D* ya que un flip-flop *D* tiene una condición “sin cambio” de entrada. Con cada pulso de reloj, la entrada *D* determina el estado siguiente de la salida. Para dejar la salida sin cambio, es necesario hacer la entrada *D* igual a la salida *Q* presente en cada flip-flop.

### Implementación de lógica secuencial

En el capítulo 6 se vio que un circuito secuencial temporizado consta de un grupo de flip-flops y compuertas combinacionales. Ya que los registros están disponibles como

circuitos MSI, a veces se vuelve conveniente emplear un registro como parte del circuito secuencial. En la Fig. 7-4 se muestra un diagrama de bloques de un circuito secuencial que usa un registro. El estado presente del registro y las entradas externas determinan el estado siguiente del registro y los valores de las salidas externas. Parte del circuito combinacional determina el estado siguiente y la otra parte genera las salidas. El valor del estado siguiente del circuito combinacional se carga en el registro con un pulso de reloj. Si el registro tiene una entrada de carga, debe establecerse en 1. En otra forma, si el registro no tiene entrada de carga (como en la Fig. 7-1), el valor del estado siguiente se transferirá en forma automática cada pulso de reloj.

La parte del circuito combinacional de un circuito secuencial puede implementarse por cualquiera de los métodos expuestos en el Capítulo 5. Puede construirse con

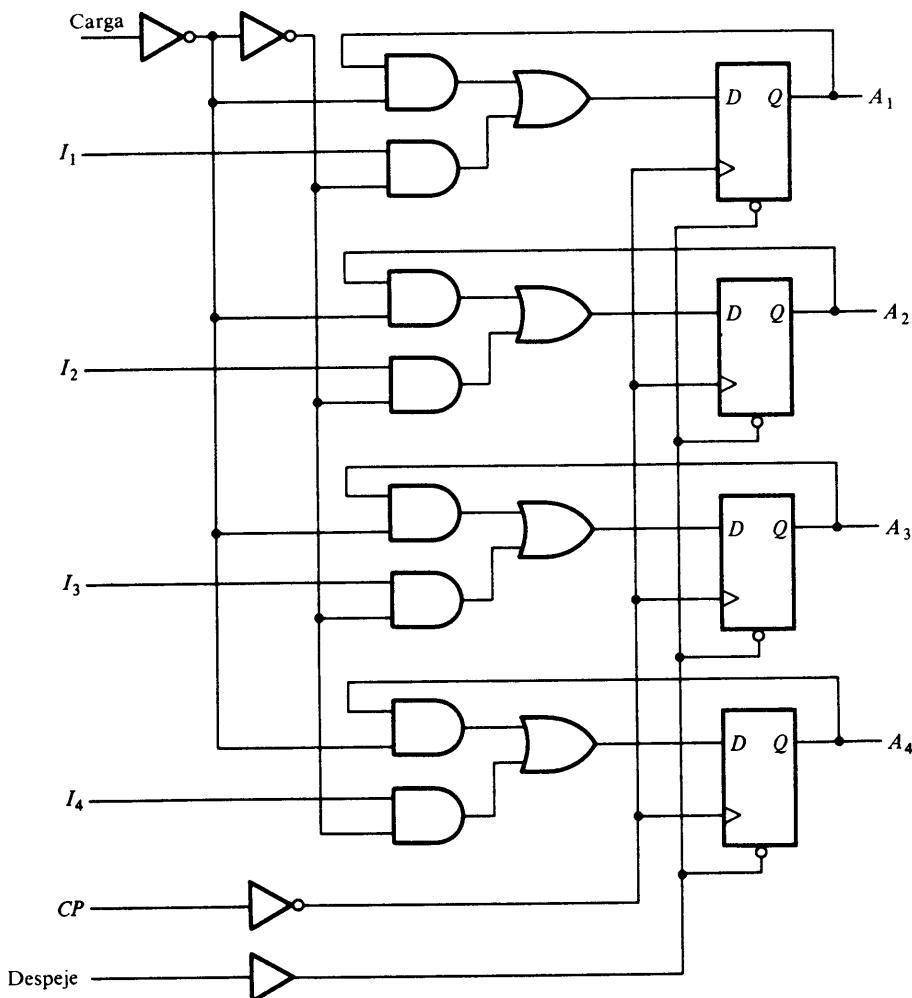


Figura 7-3 Registro con carga paralela que usa flip-flops D.

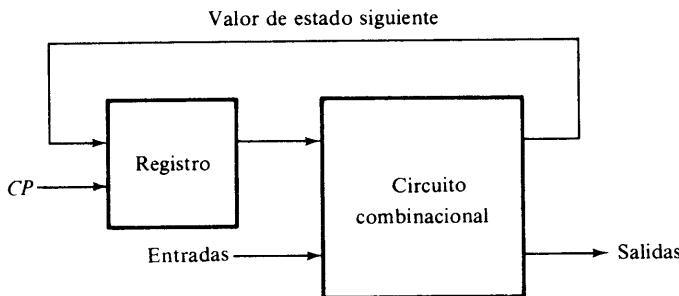


Figura 7-4 Diagrama de bloques de un circuito secuencial.

compuertas SSI, con ROM, o con un arreglo lógico programable (PLA). Mediante el uso de un registro, es posible reducir el diseño de un circuito secuencial al de un circuito combinacional conectado a un registro.

**EJEMPLO 7-1:** Diseñe el circuito secuencial cuya tabla de estado se lista en la Fig. 7-5(a).

La tabla de estado especifica dos flip-flops  $A_1$  y  $A_2$ , una entrada  $x$  y una salida  $y$ . El siguiente estado y la información de salida se obtienen de manera directa mediante la tabla:

$$A_1(t+1) = \Sigma(4, 6)$$

$$A_2(t+1) = \Sigma(1, 2, 5, 6)$$

$$y(A_1, A_2, x) = \Sigma(3, 7)$$

Los valores mintérminos son para las variables  $A_1$ ,  $A_2$  y  $x$ , las cuales son el estado presente y las variables de entrada. Las funciones para el estado siguiente y salida pueden simplificarse mediante mapas para dar:

$$A_1(t+1) = A_1x'$$

$$A_2(t+1) = A_2 \oplus x$$

$$y = A_2x$$

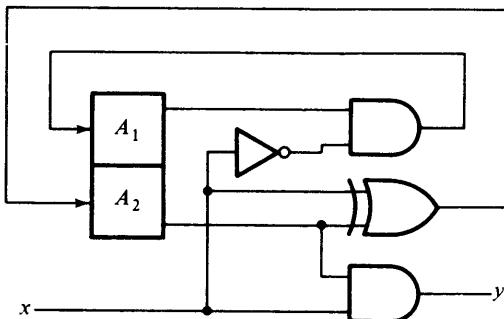
El diagrama lógico se muestra en la Fig. 7-5(b).

**EJEMPLO 7-2:** Repita el Ejemplo 7-1, pero ahora use una ROM y un registro.

La ROM puede utilizarse para implementar el circuito combinacional y el registro proporcionará los flip-flops. El número de entradas a la ROM es igual al número de flip-flops más el número de las entradas externas. El número de salidas de la ROM es igual al número de flip-flops más el número de las entradas externas. En este caso se tienen tres entradas y tres salidas de la ROM, de modo que su tamaño debe ser  $8 \times 3$ . La implementación se muestra en la Fig. 7-6. La tabla de verdad

Estado presente Entrada			Estado siguiente		Salida
$A_1$	$A_2$	$x$	$A_1$	$A_2$	$y$
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	1	1	0
1	1	1	0	0	1

(a) Tabla de estados



(b) Diagrama lógico

Figura 7-5 Ejemplo de la implementación de un circuito secuencial.

ROM es idéntica a la tabla de estado con “estado presente” y “entradas” especificando la dirección de la ROM y “estado siguiente” y “salidas” especificando las salidas ROM. Los valores de estado siguiente deben conectarse desde las salidas ROM a las entradas de registro.

### 7-3 REGISTROS CON CORRIMIENTO

Un registro capaz de correr su información binaria ya sea a la derecha o a la izquierda se denomina *registro con corrimiento*. La configuración lógica de un registro con corrimiento consta de una cadena de flip-flops conectados en cascada, con la salida de un flip-flop conectada a la entrada del siguiente flip-flop. Todos los flip-flops reciben un pulso común de reloj que causa el corrimiento de una etapa a la siguiente.

El registro con corrimiento más simple posible es uno que usa sólo flip-flops, como se muestra en la Fig. 7-7. La salida  $Q$  de un flip-flop dado se conecta a la entrada  $D$  del flip-flop a su derecha. Cada pulso de reloj corre el contenido del registrador una posición de bit a la derecha. La *entrada serial* determina qué pasa al flip-flop de la extrema izquierda durante el corrimiento. La *salida serial* se toma de la salida del

Tabla de verdad de una ROM

Dirección			Salidas		
1	2	3	1	2	3
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	1	1	0
1	1	1	0	0	1

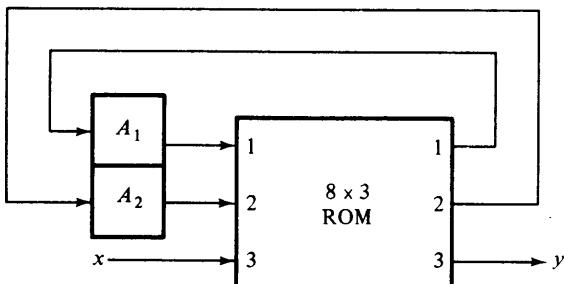


Figura 7-6 Circuito secuencial que usa un registro y una ROM.

flip-flop de la extrema derecha antes de la aplicación de un pulso. Aunque este registro corre su contenido a la derecha, si se gira la página de arriba hacia abajo, se encuentra que el registro corre su contenido a la izquierda. En consecuencia, un registro con corrimiento unidireccional puede funcionar ya sea como un registro de corrimiento a la derecha o como un registro de corrimiento a la izquierda.

El registro en la Fig. 7-7 corre su contenido con cada pulso de reloj durante el borde negativo de la transición del pulso. (Esto se indica por el círculo pequeño asociado con la entrada de reloj en todos los flip-flops.) Si se desea controlar el corrimiento de modo que ocurra sólo con ciertos pulsos pero no con otros, se debe controlar la entrada  $CP$  del registro. Se mostrará después que las operaciones de corrimiento pueden controlarse a través de las entradas  $D$  de los flip-flops más bien que a través de la entrada  $CP$ . Sin embargo, si se usa el registro con corrimiento mostrado en la Fig. 7-7, el corrimiento puede controlarse con facilidad mediante una compuerta externa AND como se muestra a continuación.

### Transferencia serial

Se dice que un sistema digital opera en modo serial cuando la información se transfiere y se manipula un bit a la vez. El contenido de un registro se transfiere a otro corriendo los bits de un registro a otro. La información se transfiere un bit a la vez corriendo los bits fuera del registro fuente al registro de destino.

La transferencia serial de información desde el registro  $A$  al registro  $B$  se hace con registros con corrimiento, como se muestra en el diagrama de bloque en la Fig. 7-8(a). La salida serial ( $SO$ ) del registro  $A$  pasa a la entrada serial ( $SI$ ) del registro  $B$ . Para evitar la pérdida de información almacenada en el registro fuente, el registro  $A$  se hace que circule su información por la conexión de salida serial a su terminal de entrada serial. El contenido inicial del registro  $B$  se corre saliendo a través de su salida serial y se pierde a menos que se transfiera a un tercer registro con corrimiento. La entrada de control de corrimiento determina cuándo y por cuántas veces se corren los registros. Esto se hace por la compuerta AND que permite que los pulsos de reloj pasen a las terminales  $CP$  sólo cuando el control de corrimiento es 1.

Se supone que los registros con corrimiento tienen cuatro bits cada uno. La unidad de control que supervisa la transferencia debe diseñarse de tal forma que capacite los registros con corrimiento, a través de la señal de control de corrimiento, por un tiempo de duración fija igual a cuatro pulsos de reloj. Esto se muestra en el diagrama de temporizado en la Fig. 7-8(b). La señal de control de corrimiento se

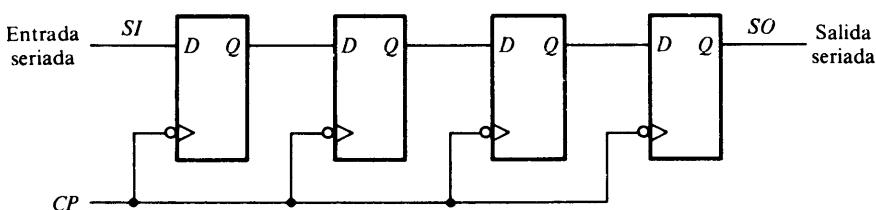
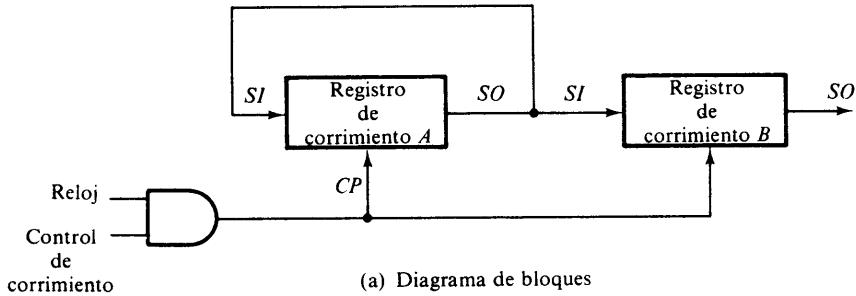
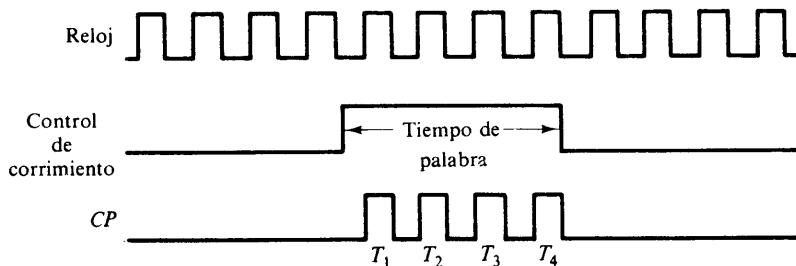


Figura 7-7 Registro de corrimiento.



(a) Diagrama de bloques



(b) Diagrama de temporizado

Figura 7-8 Transferencia seriada del registro A al registro B.

sincroniza con el reloj y cambia de valor precisamente después del borde negativo de un pulso de reloj. Los siguientes cuatro pulsos de reloj encuentran la señal de control de corrimiento en el estado 1, de modo que la salida de compuerta AND conectada a las terminales  $CP$  produce los cuatro pulsos  $T_1, T_2, T_3$  y  $T_4$ . El cuarto pulso cambia el control de corrimiento a 0 y se inhabilitan los cuatro registros con corrimiento.

Se supone que el contenido binario de  $A$  antes del corrimiento es 1011 y el de  $B$  0010. La transferencia serial desde  $A$  a  $B$  ocurrirá en cuatro pasos como se muestra en la Tabla 7-1. Después del primer pulso  $T_1$ , el bit de la extrema derecha de  $A$  se corre al bit de la extrema izquierda de  $B$  y, al mismo tiempo, este bit se circula a la posición de la extrema izquierda de  $A$ . Los otros bits de  $A$  y  $B$  se corren un lugar a la derecha. La

TABLA 7-1 Ejemplo de transferencia en serie

Pulso temporizador	Registro A de corrimiento	Registro B de corrimiento	Salida en serie de B
Valor inicial	1 0 1 1	0 0 1 0	0
Después de $T_1$	1 1 0 0	1 0 0 1	1
Después de $T_2$	1 1 1 0	0 1 1 0	0
Después de $T_3$	0 1 1 1	1 0 1 0	0
Después de $T_4$	1 0 1 1	1 1 0 1	1

salida serial previa desde  $B$  se pierde y su valor cambia de 0 a 1. Los siguientes tres pulsos realizan operaciones idénticas, corriendo los bits de  $A$  a  $B$  uno a la vez. Después del cuarto corrimiento, el control de corrimiento pasa a 0 y tanto el registro  $A$  como el  $B$  tienen el valor 1011. De este modo, el contenido de  $A$  se transfiere a  $B$  en tanto que el contenido a  $A$  permanece sin cambio.

La diferencia entre los modos de operación serial y paralelo se hace aparente mediante este ejemplo. En el modo paralelo, está disponible la información de todos los bits de un registro y todos los bits pueden transferirse en forma simultánea durante un pulso de reloj. En el modo serial, los registros tienen una sola entrada serial y una sola salida serial. La información se transfiere un bit a la vez, mientras los registros están corriendo en la misma dirección.

Las computadoras pueden operar en un modo serial, un modo paralelo o en una combinación de ambos. Las operaciones seriales son más lentas debido al tiempo que toma transferir la información de entrada y salida de los registros con corrimiento. Sin embargo, las computadoras seriales requieren menos hardware para realizar operaciones, porque un circuito común puede usarse una y otra vez para manipular los bits que salen de los registros con corrimiento en una forma secuencial. El intervalo de tiempo entre los pulsos de reloj se denomina *tiempo de bit*, y el tiempo requerido para correr el contenido completo de un registro con corrimiento se conoce como *tiempo de palabra*. Estas secuencias de tiempo se generan por las secciones de control del sistema. En una computadora en paralelo, las señales de control se habilitan durante un intervalo de pulso de reloj. Las transferencias en los registros son en paralelo y ocurren bajo la aplicación de un solo pulso de reloj. En una computadora serial, las señales de control deben mantenerse por un periodo igual a un tiempo de palabra. El pulso aplicado cada tiempo de bit transfiere el resultado de la operación, uno a la vez, dentro de un registro con corrimiento. La mayoría de las computadoras operan en un modo paralelo ya que es un modo más rápido de operación.

### Registro con corrimiento bidireccional con carga paralela

Los registros con corrimiento pueden usarse para convertir datos seriales en datos en paralelo y viceversa. Si se tiene acceso a todas las salidas flip-flop de un registro con corrimiento, entonces la información que se introduce de manera serial por corrimiento puede tomarse en salida en paralelo mediante las salidas de los flip-flops. Si se agrega capacidad de carga en paralelo a un registro con corrimiento, entonces la información que se introduce en paralelo puede tomarse en salida en forma serial corriendo la información almacenada en el registro.

Algunos registros con corrimiento proporcionan las terminales necesarias de entrada y salida para la transferencia en paralelo. También pueden tener capacidades tanto de corrimiento a la derecha como de corrimiento a la izquierda. El registro con corrimiento más general tiene todas las capacidades que se listan más adelante. Otros pueden tener algunas de esas funciones, cuando menos con una operación de corrimiento.

1. Un control de *despeje* para despejar el registro a 0.
2. Una entrada *CP* para los pulsos de reloj que sincronizan todas las operaciones.
3. Un control de corrimiento a la derecha para habilitar la operación de *corrimiento a la derecha* y las líneas de entrada serial en líneas de salida asociadas con el corrimiento a la derecha.
4. Un control de *corrimiento a la izquierda* para habilitar la operación de corrimiento a la izquierda y las líneas de *entrada serial* y las líneas de *salida* asociadas con el corrimiento a la izquierda.
5. Un control de *carga en paralelo* para habilitar una transferencia en paralelo de las  $n$  líneas de entrada asociadas con la transferencia en paralelo.
6.  $n$  líneas de salida en paralelo.
7. Un estado de control que deja la información sin cambio en el registro aun cuando se apliquen en forma continua pulsos de reloj.

Un registro capaz de correr tanto a la izquierda como a la derecha se denomina *registro de corrimiento bidireccional*. Uno que puede correr en una sola dirección se conoce como *registro de corrimiento unidireccional*. Si el registro tiene tanto capacidades de corrimiento como de carga en paralelo, se llama *registro con corrimiento y carga paralela*.

El diagrama de un registro con corrimiento que tiene todas las capacidades que se listaron con anterioridad se muestra en la Fig. 7-9.\* Consta de cuatro flip-flops *D*, aunque pueden usarse flip-flops *RS* siempre que se inserte un inversor entre las terminales *S* y *R*. Los cuatro multiplexores (MUX) son parte del registro y aquí se dibujan en forma de diagrama de bloques. (Véase la Fig. 5-16 para el diagrama lógico del multiplexor.) Los cuatro multiplexores tienen dos variables de selección comunes,  $s_1$  y  $s_0$ . La entrada 0 en cada MUX se selecciona cuando  $s_1s_0 = 00$ , la entrada 1 se selecciona cuando  $s_1s_0 = 1$  y en forma similar para las otras dos entradas de los multiplexores.

Las entradas  $s_1$  y  $s_0$  controlan el modo de operación del registro como se especifica en las funciones listadas en la Tabla 7-2. Cuando  $s_1s_0 = 00$ , el valor presente del registro se aplica a las entradas *D* de los flip-flops. Esta condición forma una trayectoria desde la salida de cada flip-flop hasta la entrada del mismo flip-flop. El siguiente pulso de reloj transfiere en cada flip-flop el valor binario que tenía previamente y no ocurre cambio de estado. Cuando  $s_1s_0 = 01$ , las terminales 1 de las entradas del multiplexor tienen una trayectoria a las entradas *D* de los flip-flops. Esto causa una operación de corrimiento a la derecha, con la entrada serial transferida al flip-flop  $A_4$ . Cuando  $s_1s_0 = 10$ , resulta una operación de corrimiento a la izquierda, con la otra entrada serial pasando al flip-flop  $A_1$ . Por último, cuando  $s_1s_0 = 11$ , la información binaria en las líneas de entrada en paralelo se transfiere al registro en forma simultánea durante el siguiente pulso de reloj.

\* Este registro es similar al IC tipo 74194.

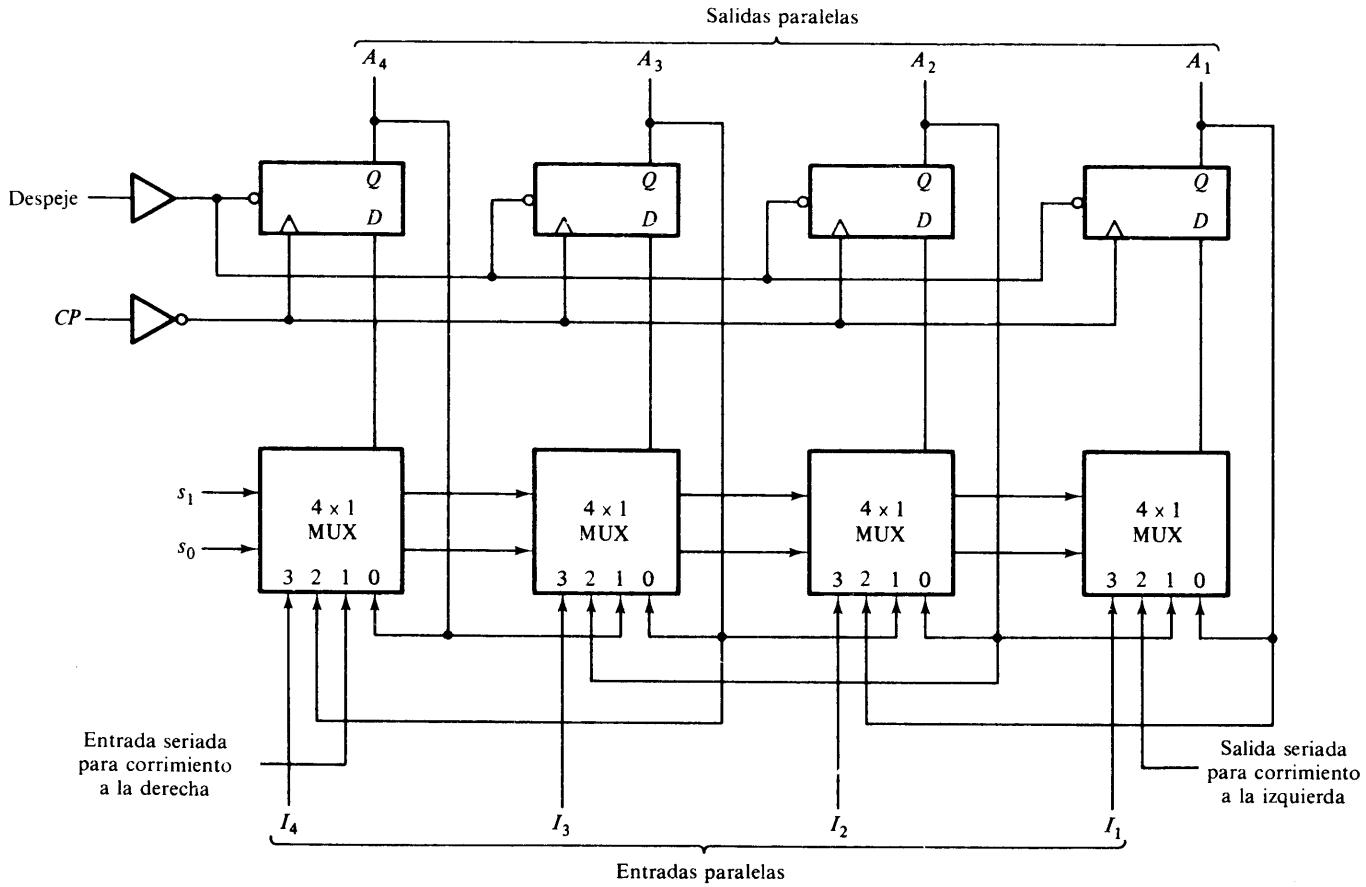


Figura 7-9 Registrador de corrimiento bidireccional de 4-bit con carga paralela.

Un registro con corrimiento bidireccional con carga paralela es un registro de propósito general capaz de realizar tres operaciones: corrimiento a la izquierda, corrimiento a la derecha y carga paralela. No todos los registros con corrimiento disponibles en los circuitos MSI tienen todas estas capacidades. La aplicación particular determina la elección de un registro de corrimiento MSI con preferencia a otro.

### Adición serial

Las operaciones en las computadoras digitales se hacen principalmente en paralelo debido a que es el modo de operación más rápido. Las operaciones seriales son lentas pero requieren menos equipo. Para demostrar el modo serial de operación, se presenta aquí el diseño de un adiconador serial. La contraparte en paralelo se expuso en la Sección 5-2.

Los dos números binarios que van a sumarse en forma serial se almacenan en dos registros de corrimiento. Se añaden bits en un par a la vez, en forma secuencial, a través de un circuito sumador completo (*FA*), como se muestra en la Fig. 7-10. El acarreo de salida del sumador completo se transfiere a un flip-flop *D*. La salida de este flip-flop se usa entonces como un acarreo de entrada para el siguiente par de bits significativos. Los dos registros de corrimiento se corren a la derecha por el periodo de un tiempo de palabra. Los bits suma de la salida *S* del sumador completo pueden transferirse a un tercer registro con corrimiento. Por el corrimiento de la suma a *A* mientras los bits de *A* se corren saliendo, es posible utilizar un registro para almacenar tanto los bits de adendo como los de sumando. La entrada serial (SI) del registro *B* es capaz de recibir un nuevo número binario mientras los bits sumando se corren hacia afuera durante la adición.

El sumador serial opera como sigue. En forma inicial, el registro *A* retiene el sumando, el registro *B* retiene el adendo y el flip-flop de acarreo, se despeja a 0. Las salidas seriales (*SO*) de *A* y *B* proporcionan un par de bits significativos para el sumador completo *x* y *y*. La salida *Q* del flip-flop de la entrada de acarreo a *z*. El control de corrimiento a la derecha habilita tanto a los registros como al flip-flop de acarreo de modo que al siguiente pulso de reloj, ambos registros se corren un lugar a la derecha, el bit de suma de *S* entra al flip-flop de la extrema izquierda de *A*. Y el acarreo de salida se transfiere al flip-flop *Q*. El control de corrimiento a la derecha habilita los registros por un número de pulsos de reloj igual al número de bits en los registros. Para

TABLA 7-2 Tabla de función para el registro en la Fig. 7-9

Modo de control		Operación del registro
<i>s</i> <sub>1</sub>	<i>s</i> <sub>0</sub>	
0	0	Sin cambio
0	1	Corrimiento a la derecha
1	0	Corrimiento a la izquierda
1	1	Carga paralela

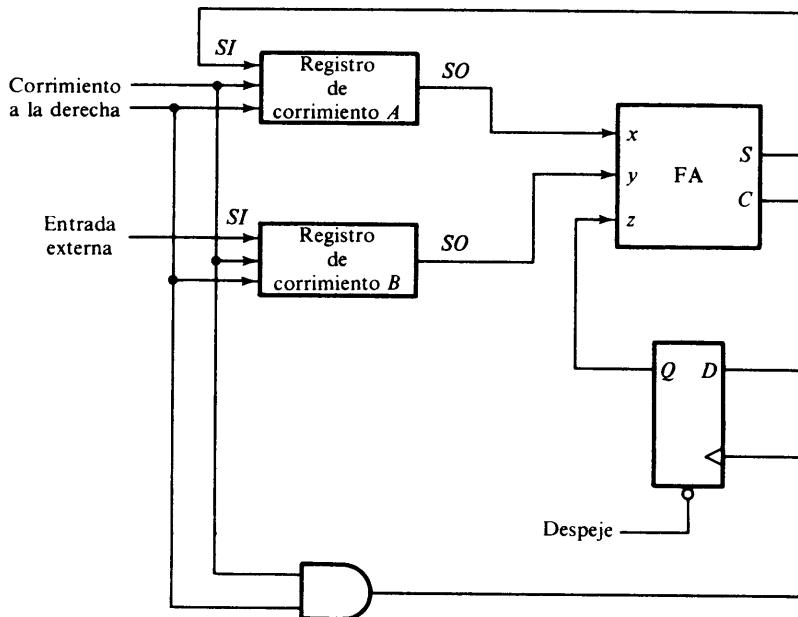


Figura 7-10 Sumador serial.

cada pulso de reloj subsecuente, se transfiere un nuevo bit de suma a  $A$ , un acarreo se transfiere a  $Q$  y ambos registros se corren un lugar a la derecha. Este proceso continúa hasta que el control de corrimiento a la derecha se habilita. Por tanto, la adición se lleva a cabo pasando cada par de bits junto con la cuenta previa que se lleva a través de un solo circuito sumador completo y se transfiere la suma, un bit a la vez, al registro  $A$ .

Si tiene que sumarse un nuevo número a los contenidos del registro  $A$ , este número debe transferirse serialmente primero al registro  $B$ . La repetición del proceso una vez más agregará el segundo número al número previo en  $A$ .

Al comparar el sumador serial con el sumador paralelo que se describe en la Sección 5-2, se observan las siguientes diferencias. El sumador paralelo debe usar registros con capacidad de carga en paralelo, en tanto que el sumador serial utiliza registros de corrimiento. El número de circuitos sumadores completos en el sumador paralelo es igual al número de bits en los números binarios, en tanto que el sumador serial requiere sólo un circuito sumador completo y un flip-flop de acarreo. Excluyendo los registros, el sumador paralelo es un circuito combinacional puro, en tanto que el sumador serial es un circuito secuencial. El circuito secuencial en el sumador serial consta de un circuito sumador completo y un flip-flop que almacena la salida de acarreo. Esto es típico en las operaciones seriales ya que el resultado de una operación de un bit a la vez puede depender no sólo de las entradas presentes, sino también de las entradas previas.

Para mostrar que las operaciones de un bit a la vez en las computadoras seriales pueden requerir un circuito secuencial, se rediseñará el sumador serial considerándolo como un circuito secuencial.

**EJEMPLO 7-3:** Diseñe un sumador serial usando un procedimiento de lógica secuencial.

Primero, debe estipularse que los dos registros de corrimiento están disponibles para almacenar los números binarios que van a sumarse en forma serial. Las salidas seriales de los registros se designan por las variables  $x$  y  $y$ . El circuito secuencial que va a diseñarse no incluirá los registros de corrimiento; se insertarán después para mostrar la unidad completa. El propio circuito secuencial tiene dos entradas,  $x$  y  $y$ , que proporcionan un par de bits significativos, y una salida  $S$  que genera el bit de suma, y el flip-flop  $Q$  para almacenar el acarreo. El estado presente de  $Q$  proporciona el valor presente del acarreo. El pulso de reloj que corre los registros habilita el flip-flop  $Q$  para cargar el siguiente acarreo. Este acarreo se utiliza entonces con el siguiente par de bits en  $x$  y  $y$ . La tabla de estado que especifica el circuito secuencial se da en la Tabla 7-3.

El estado presente de  $Q$  es el valor presente del acarreo. El acarreo presente  $Q$  se suma junto con las entradas  $x$  y  $y$  para producir el bit de suma en la salida  $S$ . El siguiente estado de  $Q$  es equivalente a la salida del acarreo. Recuérdese que los listados de la tabla de estado son idénticos a los listados en la tabla de verdad del sumador completo, excepto que la entrada del acarreo, ahora es el estado presente de  $Q$  y el acarreo de salida, ahora es el estado siguiente de  $Q$ .

Si se usa un flip-flop  $D$  para  $Q$ , se obtiene el mismo circuito que en la Fig. 7-10 ya que los requisitos de entrada de la entrada  $D$  son los mismos que para los valores de estado siguiente. Si se utiliza un flip-flop  $JK$  para  $Q$ , se obtienen los requisitos de entrada de excitación que se listan en la Tabla 7-3. Las tres funciones booleanas de interés son las funciones de entrada flip-flop para  $JK$  y  $KQ$  y salida  $S$ . Estas funciones

TABLA 7-3 Tabla de excitación para un adiconador serial

Estado presente $Q$	Entradas		Estado siguiente $Q$	Salida $S$	Entradas flip-flop	
	$x$	$y$			$JQ$	$KQ$
0	0	0	0	0	0	$X$
0	0	1	0	1	0	$X$
0	1	0	0	1	0	$X$
0	1	1	1	0	1	$X$
1	0	0	0	1	$X$	1
1	0	1	1	0	$X$	0
1	1	0	1	0	$X$	0
1	1	1	1	1	$X$	0

se especifican en la tabla de excitación y puede simplificarse mediante mapas:

$$JQ = xy$$

$$KQ = x'y' = (x + y)'$$

$$S = x \oplus y \oplus Q$$

Como se muestra en la Fig. 7-11, el circuito consta de tres compuertas y un flip-flop *JK*. Los dos registros de corrimiento también se incluyen en el diagrama para mostrar el sumador completo. Observe que la salida *S* es una función no sólo de *x* y *y*, sino también del estado presente de *Q*. El estado siguiente de *Q* es una función de los valores presentes de *x* y *y* que provienen de las salidas seriales de los registros de corrimiento.

## 7-4 CONTADORES DE ONDULACION O PULSACION

Los contadores MSI se dividen en dos categorías: contadores de ondulación o pulsación y contadores síncronos. En un contador de ondulación, la transición de salida del flip-flop sirve como una fuente para disparar otro flip-flop. En otras palabras, las entradas *CP* de todos los flip-flops (excepto el primero) se disparan no por los pulsos que llegan, sino más bien por la transición que ocurre en otros flip-flops. En un contador síncrono, los pulsos de entrada se aplican a todas las entradas *CP* de todos los flip-flops. El cambio de estado de un flip-flop particular depende del estado presente en otros flip-flops. Los contadores MSI síncronos se exponen en la sección siguiente. Aquí se presentan algunos contadores de ondulación MSI comunes y se explica su operación.

### Contador binario de ondulación

Un contador binario de ondulación consta de una conexión en serie de flip-flops complementarios (tipo *C* o *JK*), con la salida de cada flip-flop conectada a la entrada *CP* del siguiente flip-flop de orden más alto. El flip-flop que retiene el bit menos significativo recibe los pulsos de acarreo que llegan. El diagrama de un contador de ondulación binario de 4-bit se muestra en la Fig. 7-12. Todas las entradas *J* y *K* son iguales a 1. El círculo pequeño en la entrada *CP* indica que el flip-flop se complementa durante una transición que pasa a negativo o cuando la salida a la cual está conectado pasa de 1 a 0. Para entender la operación del contador binario, hay que referirse a su secuencia de conteo dada en la Tabla 7-4. Es obvio que el bit *A*<sub>1</sub> del orden más bajo debe complementarse con cada pulso de conteo. Cada vez que *A*<sub>1</sub> pasa de 1 a 0, se complementa a *A*<sub>2</sub>. Cada vez que *A*<sub>2</sub> pasa de 1 a 0, se complementa a *A*<sub>3</sub> y así sucesivamente. Por ejemplo, tómese la transición de la cuenta 0111 a 1000. Las flechas en la tabla indican las transiciones en este caso. *A*<sub>1</sub> se complementa con el pulso de conteo. Ya que *A*<sub>1</sub> pasa de 1 a 0, dispara *A*<sub>2</sub> y lo complementa. Como resultado, *A*<sub>2</sub> pasa de 1 a 0, lo cual a su vez complementa a *A*<sub>3</sub>. *A*<sub>3</sub> ahora pasa de 1 a 0, lo cual

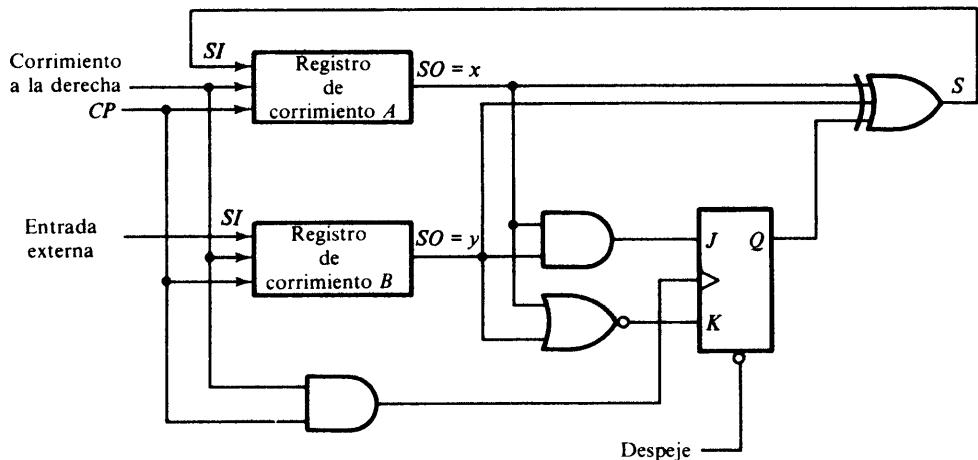


Figura 7-11 Segunda forma de un sumador serial.

complementa a  $A_4$ . La transición de salida de  $A_4$ , si se conecta a una siguiente etapa, no disparará el flip-flop siguiente ya que pasa de 0 a 1. El flip-flop tiene un cambio a la vez en una sucesión rápida, y la señal se propaga a través del contador en una forma de *ondulación*. Los conductores de ondulación algunas veces se denominan *contadores asincrónicos*.

Un contador binario con un conteo en reversa se conoce como contador binario de *decremento*. En un contador de decremento, el conteo binario se disminuye en 1 con cada pulso de entrada de conteo. El conteo de un contador de decremento de 4-bit principia del binario 15 y continúa al conteo binario 14, 13, 12, ..., 0 y entonces regresa a 15. El circuito en la Fig. 7-12 funcionará como un contador de decremento binario si las salidas se toman de las terminales complemento  $Q'$  de todos los flip-flops. Si sólo están disponibles las salidas normales de los flip-flops, el circuito debe modificarse ligeramente como se describe a continuación.

Una lista de la secuencia de conteo de un contador binario de decremento muestra que el bit de orden más bajo debe complementarse con cada pulso de conteo. Cualquier otro bit en la secuencia se complementa si su bit previo de orden más bajo pasa de 0 a 1. En consecuencia, el diagrama de un contador binario de decremento es el mismo que en la Fig. 7-12, siempre que todos los flip-flops disparen en el borde

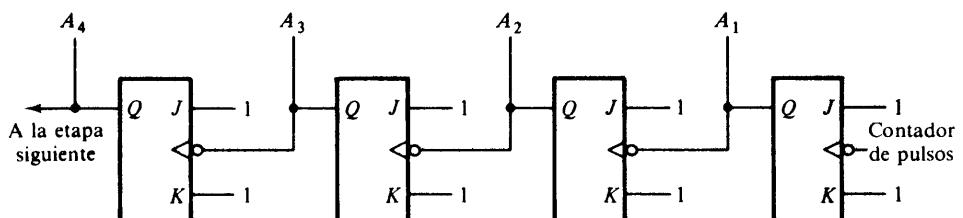


Figura 7-12 Contador binario de ondulación de 4-bit.

TABLA 7-4 Secuencia de cuenta para un contador binario de ondulación

Secuencia de cuenta				Condiciones para complementar flip-flops	
$A_4$	$A_3$	$A_2$	$A_1$		
0	0	0	0	Complemento $A_1$	
0	0	0	1	Complemento $A_1$	$A_1$ irá de 1 a 0 y complemento $A_2$
0	0	1	0	Complemento $A_1$	$A_1$ irá de 1 a 0 y complemento $A_2$ ;
0	0	1	1	Complemento $A_1$	$A_2$ irá de 1 a 0 y complemento $A_3$ ;
0	1	0	0	Complemento $A_1$	$A_1$ irá de 1 a 0 y complemento $A_2$
0	1	0	1	Complemento $A_1$	$A_1$ irá de 1 a 0 y complemento $A_2$
0	1	1	0	Complemento $A_1$	$A_1$ irá de 1 a 0 y complemento $A_3$ ;
0	1	1	1	Complemento $A_1$	$A_2$ irá de 1 a 0 y complemento $A_3$ ;
1	0	0	0	y así sucesivamente...	$A_3$ irá de 1 a 0 y complemento $A_4$

positivo del pulso. (Los círculos pequeños en las entradas  $CP$  deben estar ausentes.) Si se usan flip-flops con disparo en borde negativo, entonces la entrada  $CP$  de cada flip-flop debe conectarse a la salida  $Q'$  del flip-flop previo. Entonces, cuando  $Q$  pasa de 0 a 1,  $Q'$  pasará de 1 a 0 y complementa el siguiente flip-flop como se requiere.

### Contador de ondulación BCD

Un contador decimal sigue una secuencia de diez estados y regresa a 0 después del conteo de 9. Dicho contador debe tener cuando menos cuatro flip-flops para representar cada dígito decimal, ya que un dígito decimal se representa por un código binario cuando menos con cuatro bits. La secuencia de estados en un contador decimal está determinada por el código binario que se utiliza para representar un bit decimal. Si se usa el BCD, la secuencia de estados es como se muestra en el diagrama de estados en la Fig. 7-13. Este es similar a un contador binario, excepto que el estado después de 1001 (código para el dígito decimal 9) es 0000 (código para el dígito decimal 0).

El diseño de un contador de ondulación decimal o de cualquier contador de ondulación que no siga la secuencia binaria no es un procedimiento directo. Las herramientas formales del diseño lógico pueden servir sólo como una guía. Un producto final satisfactorio requiere el ingenio y la imaginación del diseñador.

El diagrama lógico de un contador de ondulación BCD se muestra en la Fig. 7-14.\* Las cuatro salidas se denotan con el símbolo  $Q$  con un subíndice numérico igual al peso binario del bit correspondiente en el código BCD. Los flip-flops disparan en el borde negativo, esto es, cuando la señal  $CP$  pasa de 1 a 0. Obsérvese que la salida de  $Q_1$  se aplica a las entradas  $CP$  tanto de  $Q_2$  como de  $Q_8$  y la salida de  $Q_2$  se aplica a la

\* Este circuito es similar al IC tipo 7490.

entrada  $CP$  de  $Q_4$ . Las entradas  $J$  y  $K$  se conectan ya sea a una señal 1 permanente o a las salidas de los flip-flops como se muestra en el diagrama.

Un contador de ondulación es un circuito secuencial asíncrono y no puede describirse por ecuaciones booleanas desarrolladas para describir los circuitos secuenciales temporizados. Las señales que afectan la transición del flip-flop dependen del orden en el cual cambian de 1 a 0. La operación del contador puede explicarse con una lista de condiciones para transiciones de flip-flop. Estas condiciones se derivan mediante el diagrama lógico y mediante el conocimiento de cómo opera un flip-flop  $JK$ . Recuérdese que cuando la entrada  $CP$  pasa de 1 a 0, el flip-flop se ajusta si  $J=1$ , se despeja si  $K=1$ , se complementa si  $J=K=1$  y se deja sin cambio si  $J=K=0$ . Se requieren las siguientes condiciones para cada transición de estado del flip-flop:

1.  $Q_1$  se complementa en el borde negativo de cada pulso de conteo.
2.  $Q_2$  se complementa si  $Q_8 = 0$  y  $Q_1$  pasa de 1 a 0.  $Q_2$  se despeja si  $Q_8 = 1$  y  $Q_1$  pasa de 1 a 0.
3.  $Q_4$  se complementa cuando  $Q_2$  pasa de 1 a 0.
4.  $Q_8$  se complementa cuando  $Q_4 Q_2 = 11$  y  $Q_1$  pasa de 1 a 0.  $Q_8$  se despeja ya sea si  $Q_4$  o  $Q_2$  es 0 y  $Q_1$  pasa de 1 a 0.

Para verificar que estas condiciones resultan en la secuencia requerida por un contador de ondulación BCD, es necesario verificar que todas las transiciones de los flip-flops por supuesto siguen una secuencia de estado, como se especifica por el diagrama de estado en la Fig. 7-13. Otra forma de verificar la operación del contador es derivar el diagrama de tiempos para cada flip-flop para las condiciones listadas con anterioridad. Este diagrama se muestra en la Fig. 7-15 con los estados binarios listados después de cada pulso de reloj.  $Q_1$  cambia de estado después de cada pulso de reloj.  $Q_2$  se complementa cada vez que  $Q_1$  pasa de 1 a 0 en tanto que  $Q_8 = 0$ . Cuando  $Q_8$  llega a ser 1,  $Q_2$  permanece despejado a 0.  $Q_4$  se complementa cada vez que  $Q_2$  pasa de 1 a 0.  $Q_8$  permanece despejado mientras que  $Q_2$  o  $Q_4$  esté en 0. Cuando tanto  $Q_2$  como  $Q_4$  llegan a ser 1,  $Q_8$  se complementa cuando  $Q_1$  pasa de 1 a 0.  $Q_8$  se despeja en la siguiente transición de  $Q_1$ .

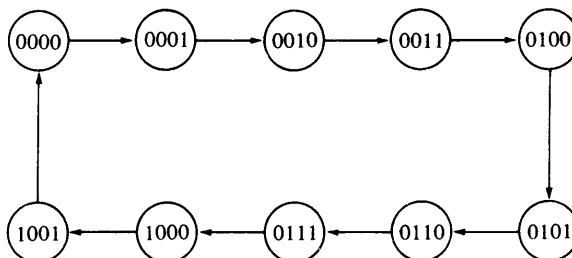


Figura 7-13 Diagrama de estado de un contador BCD decimal.

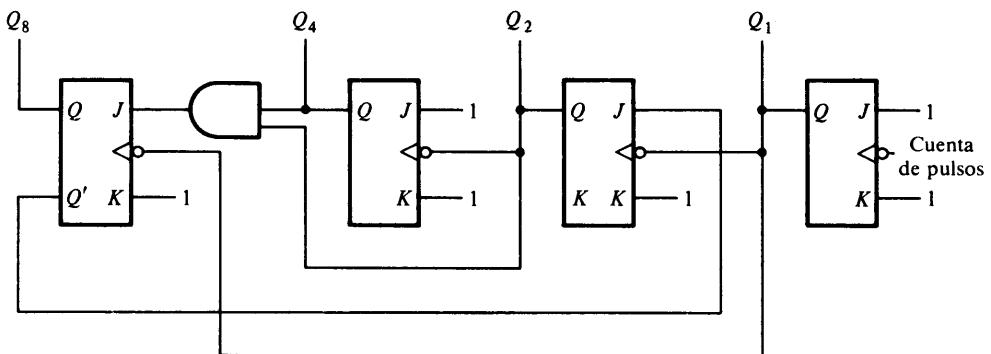


Figura 7-14 Diagrama lógico de un contador BCD de ondulación.

El contador BCD de la Fig. 7-14 es un contador de *década*, ya que cuenta desde 0 a 9. Para contar en decimal desde 0 a 99, se necesita un contador de dos décadas. Para contar desde 0 a 999, se requiere un contador de tres décadas. Los contadores de décadas múltiples pueden construirse conectando contadores BCD en cascada, uno para cada década. Un contador de tres décadas se muestra en la Fig. 7-16. Las entradas a la segunda y tercera década vienen de  $Q_8$  de la década previa. Cuando  $Q_8$  en una década pasa de 1 a 0, dispara la cuenta para la década siguiente de orden más alto, mientras su propia década pasa de 9 a 0. Por ejemplo, el conteo después de 399 será 400.

## 7-5 CONTADORES SÍNCRONOS

Los contadores síncronos se distinguen de los contadores de pulsación en que los pulsos de reloj se aplican a las entradas *CP* de *todos* los flip-flops. El pulso común dispara en forma simultánea todos los flip-flops, en lugar de uno a la vez en sucesión como en un contador de pulsación. La decisión de cuándo un flip-flop va a complementarse o no se determina por los valores de las entradas *J* y *K* al tiempo del pulso. Si  $J = K = 0$ , el flip-flop permanece sin cambio. Si  $J = K = 1$ , el flip-flop se complementa.

Un procedimiento de diseño para cualquier tipo de contador síncrono se presentó en la Sección 6-8. El diseño de un contador binario de 3-bit se llevó a cabo en detalle y se ilustró en la Fig. 6-30. En esta sección, se presentan algunos contadores síncronos MSI típicos y se explica su operación. Debe tomarse en cuenta que no es necesario diseñar un contador si ya está disponible en el comercio en la forma IC.

### Contador binario

El diseño de contadores binarios síncronos es tan simple que no es necesario pasar a través de un proceso de diseño riguroso de lógica secuencial. En un contador binario síncrono, el flip-flop en la posición del más bajo orden se complementa con cada pulso. Esto significa que sus entradas *J* y *K* deben mantenerse en la lógica 1. Un

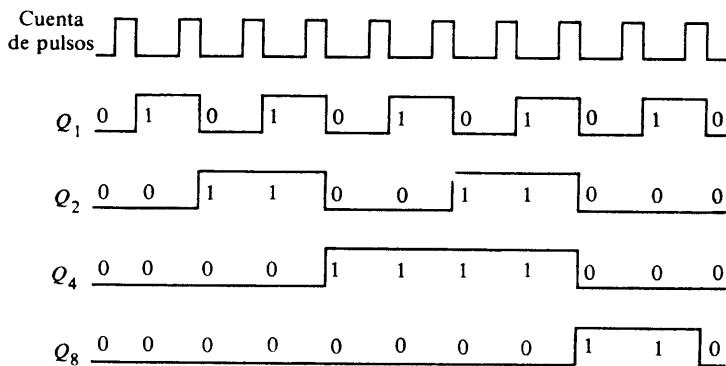


Figura 7-15 Diagrama de temporizado para el contador decimal de la Fig. 7-14.

flip-flop en cualquier otra posición se complementa con un pulso si todos los bits en las posiciones de orden más bajo son iguales a 1, porque los bits de más bajo orden (cuando todos son 1) cambiarán a 0 en el siguiente pulso de conteo. El conteo binario dicta que el siguiente bit de orden más alto se complemente. Por ejemplo, si el estado presente de un contador de 4-bit es  $A_4A_3A_2A_1 = 0011$ , el siguiente conteo será 0100.  $A_1$  se complementa siempre.  $A_2$  se complementa ya que el estado presente de  $A_1 = 1$ .  $A_3$  se complementa porque el estado presente de  $A_2A_1 = 11$ . Pero  $A_4$  no se complementa ya que el estado presente de  $A_3A_2A_1 = 011$ , lo cual no da la condición de todos 1.

Los contadores síncronos binarios tienen un patrón regular y pueden construirse con facilidad con flip-flops que se complementan y compuertas. El patrón regular puede verse con claridad mediante el contador de 4-bit que se muestra en la Fig. 7-17. Las terminales  $CP$  de todos los flip-flops se conectan a una fuente común de pulsos de reloj. La primera etapa  $A_1$  tiene sus entradas  $J$  y  $K$  iguales a 1 si el contador está habilitado. Las otras entradas  $J$  y  $K$  son iguales a 1 si todos los bits previos de orden bajo son iguales a 1 y el conteo está habilitado. La cadena de compuertas AND genera la lógica requerida para las entradas  $J$  y  $K$  en cada etapa. El contador puede extenderse a cualquier número de etapas, y cada etapa tiene un flip-flop adicional y una compuerta AND que da una salida de 1 si todas las salidas previas de los flip-flops son 1.

Obsérvese que los flip-flops disparan en el borde negativo del pulso. Esto no es esencial aquí como era en el caso del contador de pulsación. El contador también puede dispararse en el borde positivo del pulso.

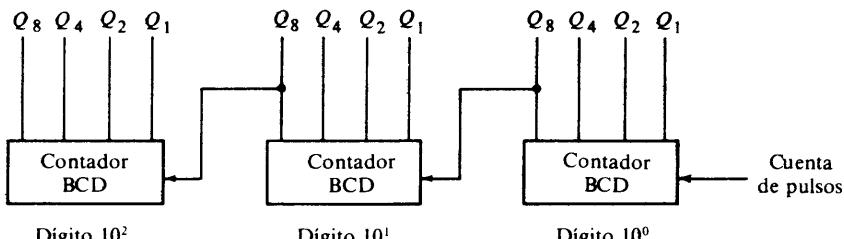


Figura 7-16 Diagrama de bloques de un contador decimal BCD de 3-décadas.

## Contador binario de incremento-decremento

En un contador binario síncrono de decremento, el flip-flop en la posición de orden más bajo se complementa con cada pulso. Un flip-flop en cualquier otra posición se complementa con un pulso siempre que todos los bits de orden más bajo sean iguales a 0. Por ejemplo, si el estado presente de un contador binario de decremento de 4-bit es  $A_4A_3A_2A_1 = 1100$ , el siguiente conteo será 1011.  $A_1$  está siempre complementado.  $A_2$  está complementado ya que el estado presente de  $A_1 = 0$ .  $A_3$  está complementado porque el estado presente de  $A_2A_1 = 00$ . Pero  $A_4$  no está complementado ya que el estado presente de  $A_3A_2A_1 = 100$ , lo cual no es una condición de todos 0.

Un contador binario de decremento puede construirse como se muestra en la Fig. 7-17, excepto que las entradas a las compuertas AND deben llegar de las salidas complementadas  $Q'$  y no de las salidas normales  $Q$  de los flip-flops previos. Las dos operaciones pueden combinarse en un circuito. Un contador binario capaz de contar ya sea hacia arriba o hacia abajo se muestra en la Fig. 7-18. Los flip-flops  $T$  que se emplean en este circuito pueden considerarse como flip-flops  $JK$  con las terminales  $J$  y  $K$  ligadas. Cuando el control de entrada de incremento es 1, el circuito cuenta hacia arriba, ya que las entradas  $T$  están determinadas por los valores previos de las salidas normales en  $Q$ . Cuando el control de entrada de decremento es 1, el circuito cuenta hacia *abajo*, ya que las salidas complementarias  $Q'$  determinan los estados de las entradas  $T$ . Cuando tanto las señales hacia *arriba* como las hacia *abajo* son 0, el registro no cambia de estado y permanece en la misma cuenta.

## Contador BCD

Un contador BCD cuenta en código decimal binario desde 0000 a 1001 y de vuelta a 0000. Debido al retorno a 0 después de una cuenta de 9, un contador BCD no tiene un patrón regular como en una cuenta binaria directa. Para derivar el circuito de un contador síncrono BCD, es necesario pasar a través de un procedimiento de diseño como se expuso en la Sección 6-8.

La secuencia de conteo de un contador BCD se da en la Tabla 7-5. La excitación para los flip-flops  $T$  se obtiene mediante la secuencia de conteo. En la tabla también se muestra una salida  $y$ . Esta salida es igual a 1 cuando el estado presente en el contador es 1001. En esta forma  $y$  puede habilitar el conteo de la siguiente década de orden más alto, mientras el mismo pulso cambia la década presente desde 1001 hasta 0000.

Las funciones de entrada flip-flop de la tabla de excitación pueden simplificarse mediante mapas. Los estados sin uso para los mintérminos 10 al 15 se toman como términos no importa. Las funciones simplificadas se listan a continuación:

$$TQ_1 = 1$$

$$TQ_2 = Q'_8Q_1$$

$$TQ_4 = Q_2Q_1$$

$$TQ_8 = Q_8Q_1 + Q_4Q_2Q_1$$

$$y = Q_8Q_1$$

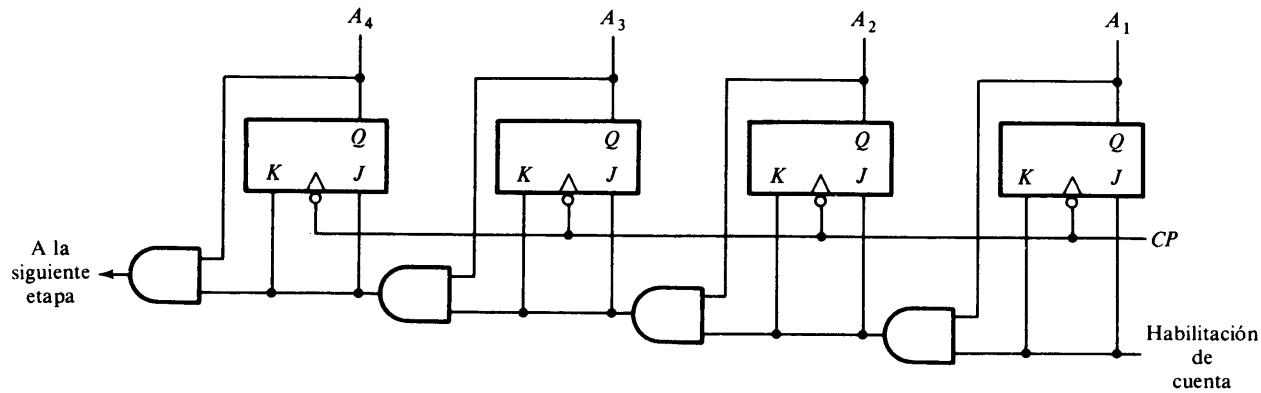


Figura 7-17 Contador síncrono binario de 4-bit.

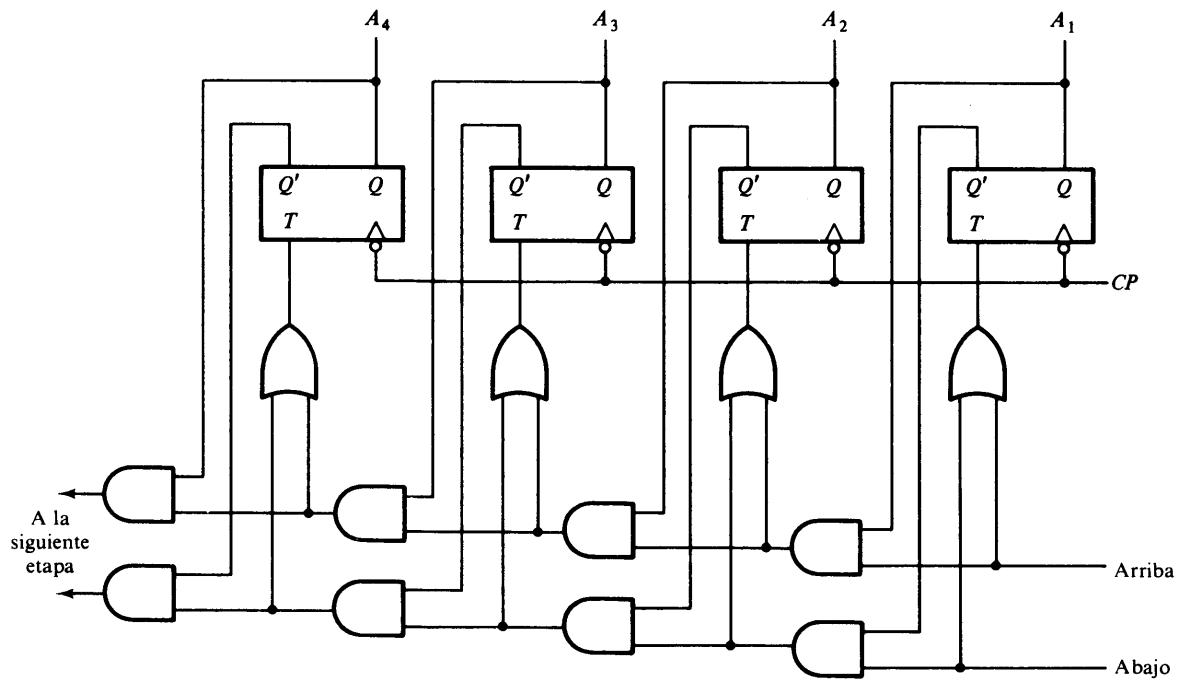


Figura 7-18 Contador binario de incremento decremto de 4-bit.

TABLA 7-5 Tabla de excitación para un contador BCD

Secuencia de cuenta				Entradas flip-flop				Salida de acarreo
$Q_8$	$Q_4$	$Q_2$	$Q_1$	$TQ_8$	$TQ_4$	$TQ_2$	$TQ_1$	$y$
0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	1	1	0
0	0	1	0	0	0	0	1	0
0	0	1	1	0	1	1	1	0
0	1	0	0	0	0	0	1	0
0	1	0	1	0	0	1	1	0
0	1	1	0	0	0	0	1	0
0	1	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1	0
1	0	0	1	1	0	0	1	1

El circuito puede dibujarse fácilmente con cuatro flip-flops  $T$ , cinco compuertas AND y una compuerta OR.

Los contadores síncronos BCD pueden configurarse en cascada para formar un contador para números decimales de cualquier longitud. La configuración en cascada se hace como en la Fig. 7-16, excepto que la salida  $y$  debe conectarse a la entrada de cuenta de la década siguiente de orden más alto.

### Contador binario con carga en paralelo

Los contadores empleados en los sistemas digitales con frecuencia requieren una capacidad de carga en paralelo para transferir un número inicial binario antes de la operación de conteo. En la Fig. 7-19 se muestra en diagrama lógico de un registro que tiene una capacidad de carga en paralelo y también puede operar como un contador.\* El control de la carga de entrada, cuando es igual a 1, inhabilita la secuencia de conteo y provoca una transferencia de datos de las entradas  $I_1$  hasta  $I_4$  a los flip-flops  $A_1$  hasta  $A_4$ , respectivamente. Si la carga de entrada es 0 y el control de cuenta de entrada es 1, el circuito opera como un contador. Los pulsos de reloj entonces deben provocar que el estado de los flip-flops cambie de acuerdo con la secuencia de conteo binario. Si ambas entradas de control son 0, los pulsos de reloj no cambian el estado del registro.

La terminal de salida de la cuenta que se lleva a ser 1 si todos los flip-flops son iguales a 1 mientras se habilita la entrada de conteo. Esta es la condición para complementar el flip-flop que retiene el bit siguiente de orden más alto. Esta salida es útil para aumentar el contador a más de cuatro bits. La velocidad del contador se aumenta si este acarreo se genera directamente de las salidas de todos los cuatro flip-flops en lugar de pasar a través de una cadena de compuertas AND. En forma similar, cada flip-flop está asociado con una compuerta AND que recibe todas las

\* Este es similar pero no idéntico al IC tipo 74161.

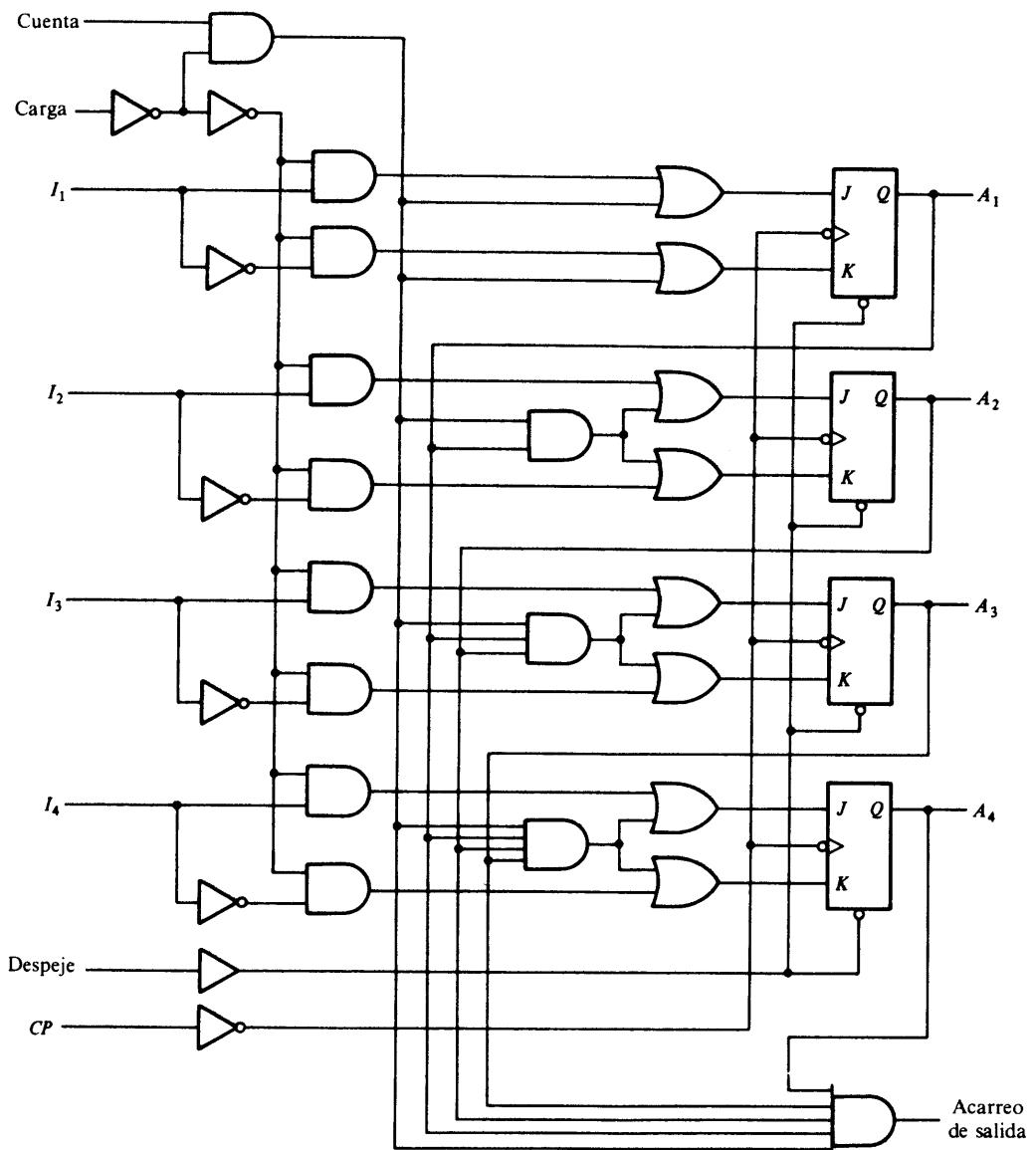


Figura 7-19 Contador binario de 4-bit con carga paralela.

salidas flip-flop previas directamente para determinar cuándo debe complementarse el flip-flop.

La operación del contador se resume en la Tabla 7-6. Las cuatro entradas de control: despeje,  $CP$ , carga y conteo determinan el siguiente estado de salida. El despeje de entrada es asíncrono y, cuando es igual a 0, causa que el contador se despeje a todos 0, sin importar la presencia de pulsos de reloj u otras entradas. Esto se indica en la tabla con las  $X$  anotadas, que simbolizan condiciones no importa para las otras entradas, de modo que su valor puede ser 0 o bien 1. La entrada despejada debe pasar al estado 1 para las operaciones temporizadas que se listan en las siguientes tres anotaciones en la tabla. Con las entradas tanto de carga como de cuenta en 0, las salidas no cambian, ya sea que se aplique o no un pulso a la terminal  $CP$ . Una carga en la entrada de 1 causa una transferencia de las entradas  $I_1$ – $I_4$  al registro durante el borde positivo de un pulso de entrada. La información de entrada se carga en el registro sin importar el valor de la entrada de cuenta, ya que la entrada de cuenta está dividida cuando la entrada de carga es 1. Si la carga de entrada se mantiene en 0, la entrada de conteo controla la operación del contador. Las salidas cambian al siguiente conteo binario en la transición de borde positivo de cada pulso de reloj, pero no ocurre cambio de estado si la entrada de cuenta es 0.

El contador de 4-bit que se muestra en la Fig. 7-19 puede encerrarse en un paquete IC. Son necesarios dos IC para la construcción de un contador de 8-bit; cuatro IC para un contador de 16-bit; y así sucesivamente. El acarreo de salida de un IC debe conectarse a la entrada de cuenta del IC que retiene los bits siguientes de orden más alto del contador.

Los contadores con capacidad de carga paralela que tienen un número especificado de bits son muy útiles en el diseño de sistemas digitales. Se hará referencia después a estos como registros con capacidades de carga y de incremento. La función de *incremento* es una operación que añade 1 al contenido presente de un registro. Al capacitar el control de cuenta durante el periodo de un pulso de reloj, el contenido del registro puede incrementarse en 1.

Un contador con carga paralela puede utilizarse para generar cualquier número deseado de secuencias de conteo. Un contador módulo- $N$  (abreviado mod  $N$ ) es un contador que pasa a través de una secuencia repetida de  $N$  cuentas. Por ejemplo, un contador binario de 4-bit es un contador mod-16. Un contador BCD es un contador mod-10. En algunas aplicaciones, puede ser innecesario preocuparse de los  $N$  estados particulares que usa un contador mod- $N$ . Si este es el caso, entonces puede emplearse

TABLA 7-6 Tabla de función para el contador en la Fig. 7-19

Despeje	$CP$	Carga	Cuenta	Función
0	$X$	$X$	$X$	Despeje a 0
1	$X$	0	0	Sin cambio
1	↑	1	$X$	Entradas de carga
1	↑	0	1	Cuenta en el siguiente estado binario

un contador con carga paralela para construir cualquier contador mod- $N$ , con  $N$  correspondiendo a cualquier valor deseado. Esto se muestra en el siguiente ejemplo.

**EJEMPLO 7-4:** Construya un contador mod-6 usando el circuito MSI especificado en la Fig. 7-19.

En la Fig. 7-20 se muestran cuatro formas en las cuales puede usarse un contador con carga paralela para generar una secuencia de seis conteos. En cada caso el control de cuenta se establece en 1 para habilitar la cuenta a través de pulsos en la entrada  $CP$ . También se utilizan los hechos de que el control de carga inhibe el conteo y que la operación de despeje es independiente de otras entradas de control.

La compuerta AND en la Fig. 7-20(a) detecta la ocurrencia del estado 0101 en la salida. Cuando el contador se encuentra en ese estado, se habilita la entrada de carga y una entrada de todos a 0 se carga en el registro. Por tanto, el contador pasa a través de los estados binarios 0, 1, 2, 3, 4 y 5 y, entonces, regresa a 0. Este procedimiento produce una secuencia de seis conteos.

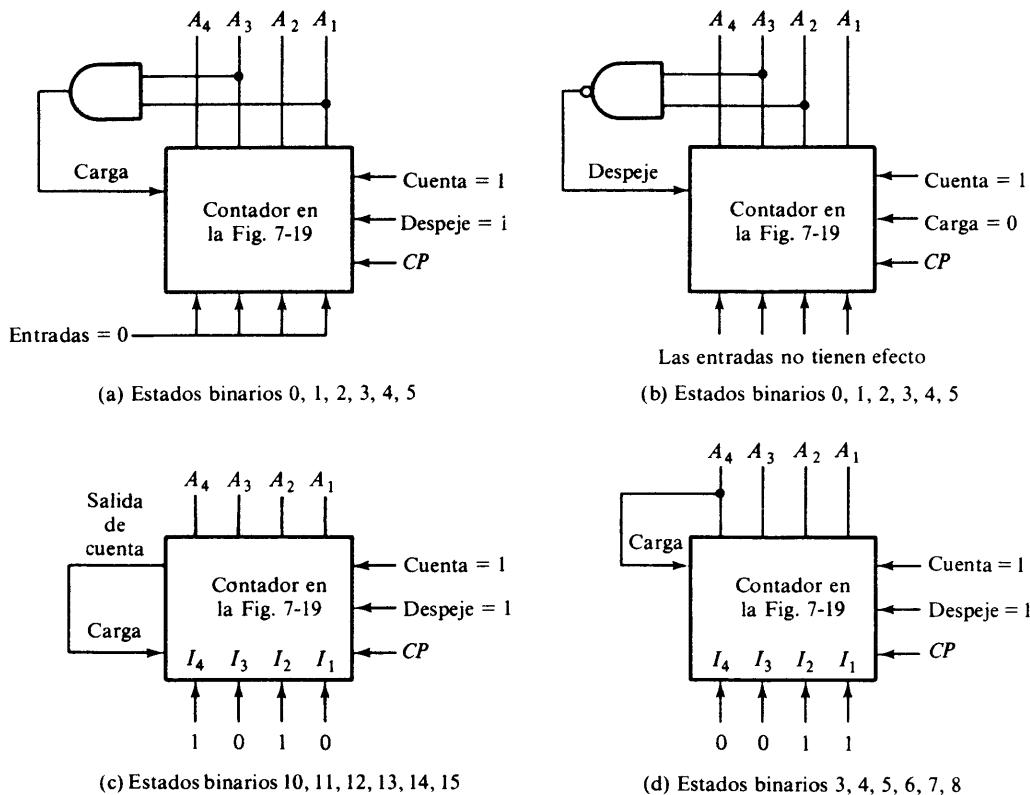
La entrada de despeje del registro es asíncrona, esto es, no depende del reloj. En la Fig. 7-20(b), la compuerta NAND detecta la cuenta de 0110, pero tan pronto como ocurre esta cuenta, el registro se despeja. La cuenta 0110 no tiene oportunidad de permanecer por algún tiempo apreciable ya que el registro pasa inmediatamente a 0. Ocurre un pico momentáneo en la salida  $A_2$  conforme la cuenta pasa desde 0101 hasta 0110 e inmediatamente a 0000. Este pico montáneo puede ser indeseable y por esta razón no se recomienda esta configuración. Si el contador tiene una entrada asíncrona de despeje, es posible despejar el contador con el reloj después de una ocurrencia de la cuenta 0101.

En lugar de usar los primeros seis conteos, puede desearse elegir los últimos seis conteos desde 10 a 15. En este caso es posible tomar ventaja del acarreo de salida, para cargar un número en el registro. En la Fig. 7-20(c), el contador principia con la cuenta 1010 y continúa hasta 1111. El acarreo de salida generado durante el último estado habilita el control de carga, el cual, carga la entrada que se establece en 1010.

También es posible escoger cualquier cuenta intermedia de seis estados. El contador mod-6 en la Fig. 7-20(d) pasa a través de la secuencia de conteo 3, 4, 5, 6, 7 y 8. Ya obtenido el último conteo 1000 la salida  $A_4$  pasa a 1 y se capacita el control de carga. Esto carga en el registro el valor 0011 y la cuenta binaria continúa desde este estado.

## 7-6 SECUENCIAS DE TEMPORIZADO

La secuencia de las operaciones en un sistema digital se especifica por una unidad de control. La unidad de control que supervisa las operaciones en un sistema digital en forma normal consta de señales de temporizado que determinan la secuencia del



**Figura 7-20** Cuatro formas de realizar un contador mod-6 usando un contador con carga paralela.

tiempo en el cual se ejecutan las operaciones. Las secuencias de tiempo en la unidad de control pueden generarse con facilidad mediante contadores o registros de corrimiento. Esta sección demuestra el uso de esas funciones MSI en la generación de señales de temporizado para una unidad de control.

### Generación de tiempo de palabra

Primero, se demuestra un circuito que genera la señal de temporizado requerida para el modo serial de operación. La transferencia serial de información se expuso en la Sección 7-3, con un ejemplo mostrado en la Fig. 7-8. La unidad de control en una computadora serial debe generar una señal de *tiempo de palabra* que permanezca activa por un número de pulsos igual al número de bits en los registros de corrimiento. La señal de tiempo de palabra puede generarse mediante un contador que cuente el número requerido de pulsos.

Se supone que la señal de tiempo de palabra que va a generarse debe permanecer activa por un periodo de ocho pulsos de reloj. En la Fig. 7-21(a) se muestra un circuito contador que lleva a cabo esta tarea. Inicialmente, el contador de 3-bit se despeja a 0.

Una señal de arranque ajustará el flip-flop  $Q$ . La salida de este flip-flop suministra el control de tiempo de palabra y también habilita el contador. Después de la cuenta de ocho pulsos, el flip-flop se restaura y  $Q$  pasa a 0. El diagrama en tiempo de la Fig. 7-21(b) demuestra la operación del circuito. La señal de arranque se sincroniza con el reloj y permanece activa por un periodo de pulso de reloj. Después  $Q$  se establece en 1, el contador principia a contar los pulsos de reloj. Cuando el contador alcanza la cuenta de 7 (binario 111), envía una señal de paro a la entrada restaurar del flip-flop. La señal de paro llega a ser un 1 después de la transición de borde negativo del pulso 7. El siguiente pulso de reloj cambia el contador al estado 000 y también dejea  $Q$ . Ahora el contador está inhabilitado y la señal de tiempo de palabra permanece en 0. Obsérvese que el contador de tiempo de palabra permanece activo por un periodo de ocho pulsos. Obsérvese también que la señal de paro en este circuito puede usarse para arrancar otro control de cuenta de palabra en otro circuito, precisamente como la señal de arranque se usa en este circuito.

### Señales de temporizado

En un modo paralelo de operación, un sólo pulso de reloj puede especificar el tiempo al cual debe ejecutarse una operación. La unidad de control en un sistema digital que opera en el modo paralelo debe generar señales de tiempo que permanecen activas sólo

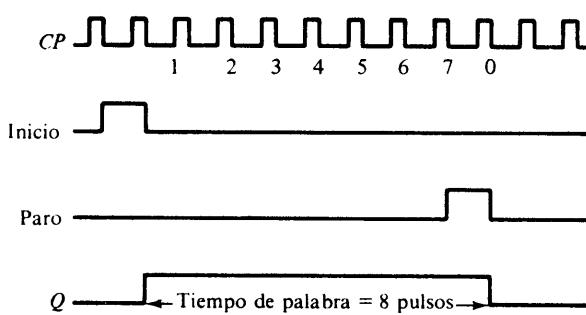
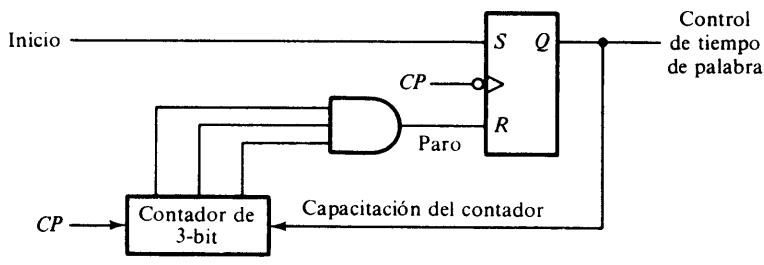


Figura 7-21 Generación de un control de tiempo de palabra para operaciones seriadas.

por un periodo de pulso de reloj, pero estas señales de tiempo deben diferenciarse unas de otras.

Las señales de temporizado que controlan la secuencia de operaciones en un sistema digital pueden generarse con un registro de corrimiento o un contador con un decodificador. Un *contador de anillo* es un registro de corrimiento circular con sólo un flip-flop que se ajusta en cualquier tiempo particular; todos los otros están despejados. El único bit se corre de un flip-flop a otro para producir la secuencia de señales de temporizado. En la Fig. 7-22(a) se muestra un registro de corrimiento de 4-bit conectado como un contador en anillo. El valor inicial del registro es 1000, lo cual produce la variable  $T_0$ . El bit único se corre a la derecha con cada pulso de reloj y circula regresando desde  $T_3$  a  $T_0$ . Cada flip-flop está en el estado 1 una vez cada cuatro pulsos de reloj y produce una de las cuatro señales de temporizado que se muestran en la Fig. 7-22(c). Cada salida se vuelve un 1 después de la transición de borde negativo de un pulso de reloj y permanece en 1 durante el siguiente pulso de reloj.

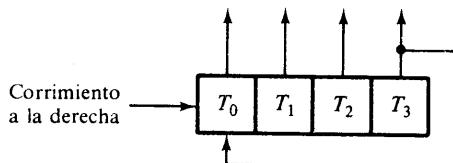
Las señales de temporizado pueden generarse también habilitando en forma continua un contador de 2-bit que pasa a través de cuatro estados distintos. El decodificador que se muestra en la Fig. 7-22(b) decodifica los cuatro estados del contador y genera la secuencia requerida de señales de temporizado.

Las señales de temporizado, cuando se habilita por los pulsos de reloj, proporcionarán pulsos de reloj de fases múltiples. Por ejemplo, si  $T_0$  se opera AND con  $CP$ , la salida de la compuerta AND generará pulsos de reloj a un cuarto de la frecuencia de los pulsos del reloj maestro. Los pulsos de reloj de fase múltiple pueden usarse para controlar registros diferentes con escalas de tiempo distintas.

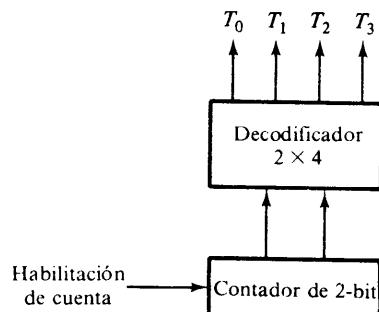
Para generar  $2^n$  señales de temporizado, se necesita ya sea un registro de corrimiento con  $2^n$  flip-flops o un contador de  $n$ -bit junto con un decodificador de  $n$ -a- $2^n$  líneas. Por ejemplo, pueden generarse 16 señales temporizadoras con un registro de corrimiento de 16-bit conectado con un contador de anillo o con un contador de 4-bit y un decodificador 4-a-16 líneas. En el primer caso, se necesitan 16 flip-flops. En el segundo caso, se necesitan 4 flip-flops y 16 compuertas AND de cuatro entradas para el decodificador. También es posible generar las señales de temporizado con una combinación de un registro de corrimiento y un decodificador. En esta forma, el número de flip-flops es menos que en un contador de anillo, y el decodificador requiere sólo compuertas de dos entradas. Esta combinación algunas veces se denomina *contador Johnson*.

### Contador Johnson

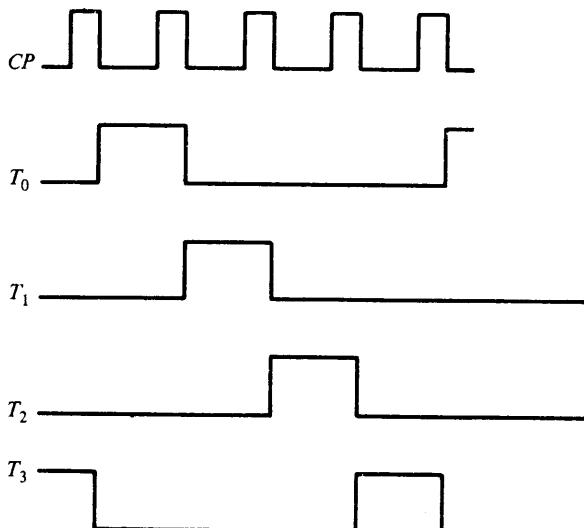
Un contador de anillo de  $k$ -bit circula un bit único entre los flip-flops para proporcionar  $k$  estados distintos. El número de estados puede duplicarse si el registro de corrimiento se conecta como un contador de anillo con *cambio en cola*. Un contador de anillo con cambio en cola es un registro de corrimiento circular con la salida complementaria del último flip-flop conectada a la entrada del primer flip-flop. En la Fig. 7-23(a) se muestra dicho registro de corrimiento. La conexión circular se hace desde la salida complementaria del flip-flop de la extrema derecha a la entrada del flip-flop de la extrema izquierda. El registro corre su contenido una posición a la derecha con cada



(a) Contador en anillo (valor inicial = 1000)



(b) Contador y decodificador



(c) Secuencia de cuatro señales de temporizado

**Figura 7-22** Generador de señales de temporizado.

pulso de reloj, y al mismo tiempo, el valor complementario del flip-flop *E* se transfiere al flip-flop *A*. Principiando desde un estado despejado, el contador de anillo de cambio en cola pasa a través de una secuencia de ocho estados como se lista en la Fig. 7-23(b). En general un contador de anillo de cambio en cola de *k*-bit pasará a través de una secuencia de  $2k$  estados. Principiando desde todos 0, cada operación de corrimiento inserta 1 desde la izquierda hasta que el registro está lleno con todos los 1. En las siguientes secuencias, se insertan los 0 desde la izquierda hasta que el registro se llena otra vez con todos los 0.

Un contador Johnson es un contador de anillo que cambia en cola de *k*-bit con  $2k$  compuertas decodificadoras para proporcionar salidas para las  $2k$  señales de temporizado. Las compuertas decodificadoras se muestran en la Fig. 7-23, pero están especificadas en la última columna de la tabla. Las ocho compuertas AND listadas en la tabla, cuando se conectan al circuito, complementarán la construcción del contador Johnson. Ya que cada compuerta se habilita durante una secuencia de estado particular, las salidas de las compuertas generarán ocho secuencias de temporizado en sucesión.

La decodificación de un contador de anillo de cambio en cola de *k*-bit para obtener  $2k$  secuencias de temporizado sigue un patrón regular. El estado de todos 0 se decodifica tomando el complemento de las salidas de los dos flip-flops extremos. El estado de todos 1 se decodifica tomando las salidas normales de los dos flip-flops extremos. Todos los demás estados se decodifican de un patrón adyacente 1, 0 o 0, 1 en la secuencia. Por ejemplo, la secuencia 7 tiene un patrón adyacente 0, 1 en los flip-flops *B* y *C*. La salida decodificada se obtiene entonces tomando el complemento de *B* y salida normal de *C*, o  $B'C$ .

Una desventaja del circuito que se muestra en la Fig. 7-23(a) es que, si se encuentra en un estado sin uso por sí mismo, persistirá en moverse de un estado inválido a otro y nunca encontrará su camino a un estado válido. Esta dificultad puede corregirse modificando el circuito para evitar esta condición indeseable. Un procedimiento de corrección es desconectar la salida del flip-flop *B* que va a la entrada *D* del flip-flop *C* y habilitar entonces la entrada del flip-flop *C* por la función:\*

$$DC = (A + C)B$$

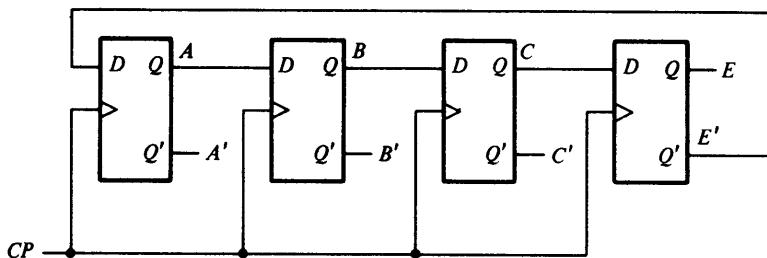
en donde *DC* es la función de entrada del flip-flop para la entrada *D* del flip-flop *C*.

Los contadores Johnson pueden construirse para cualquier número de secuencias de temporizado. El número de flip-flops necesarios es la mitad del número de las señales de temporizado. El número de compuertas decodificadoras es igual al número de señales de temporizado y sólo se emplean compuertas de dos entradas.

## 7-7 UNIDAD DE MEMORIA

Los registros en una computadora digital pueden clasificarse ya sea en el tipo operacional o de almacenamiento. Un registro *operacional* es capaz de almacenar información binaria en sus flip-flops y, además, tiene compuertas combinacionales capaces de

\* Esta es la forma en que está hecho el IC tipo 4022.



(a) Contador en anillo de cola-cambio de 4 etapas

Número de secuencia	Salidas flip-flop				Compuerta AND requerida para la salida
	<i>A</i>	<i>B</i>	<i>C</i>	<i>E</i>	
1	0	0	0	0	$A'E'$
2	1	0	0	0	$AB'$
3	1	1	0	0	$BC'$
4	1	1	1	0	$CE'$
5	1	1	1	1	$AE$
6	0	1	1	1	$A'B$
7	0	0	1	1	$B'C$
8	0	0	0	1	$C'E$

(b) Secuencia de cuenta y decodificación requerida

Figura 7-23 Construcción de un contador Johnson.

realizar las tareas de procesamiento de datos. Un registro de *almacenamiento* se utiliza sólo para almacenamiento temporal de la información binaria. Esta información no puede alterarse cuando se transfiere dentro y fuera del registro. Una *unidad de memoria* es una colección de registros de almacenamiento junto con los circuitos asociados necesarios para transferir la información dentro y fuera de los registros. Los registros de almacenamiento en una unidad de memoria se denominan *registros de memoria*.

La mayor parte de los registros en una computadora digital son registros de memoria, a los cuales se transfiere para almacenamiento la información y de los cuales está disponible la información cuando se necesita para procesarla. En la unidad de proceso, en forma comparativa, se encuentran pocos registros operacionales. Cuando tiene lugar el procesamiento de datos, la información de registros seleccionados en la unidad de memoria se transfiere primero a los registros operacionales en la unidad de procesamiento. Los resultados intermedios y final obtenidos en los registros operacionales se transfieren devolviéndolos a registros de memoria seleccionados. En forma similar, la información binaria recibida de los dispositivos de entrada se almacena primero en los registros de memoria; la información transferida a los dispositivos de salida se toma de los registros en la unidad de memoria.

El componente que forma las celdas binarias de los registros en una unidad de memoria debe tener ciertas propiedades básicas; las más importantes de estas son: (1) Debe tener una propiedad confiable de dos estados para la representación binaria. (2) Debe tener tamaño pequeño. (3) El costo por bit de almacenamiento debe ser tan

bajo como sea posible. (4) El tiempo de acceso a un registro de memoria debe ser razonablemente rápido. Los ejemplos de componentes de unidades de memoria son núcleos magnéticos, IC semiconductores, y superficies magnéticas en cintas, tambores o discos.

Una unidad de memoria almacena información binaria en grupos llamados *palabras* y cada palabra se almacena en un registro de memoria. Una palabra en memoria es una entidad de  $n$  bits que se mueven hacia adentro y hacia afuera del almacenamiento como una unidad. Una palabra de memoria puede representar un operando, una instrucción, un grupo de caracteres alfanuméricos, o cualquier información codificada en binario. La comunicación entre una unidad de memoria y su medio ambiente se logra a través de dos señales de control y dos registros externos. Las señales de control especifican la dirección de la transferencia requerida, esto es, ya sea que una palabra se almacene en un registro de memoria o que una palabra previamente almacenada se transfiera fuera del registro de memoria. Un registro externo especifica el registro particular de memoria escogido entre los miles disponibles; el otro especifica la configuración particular de bits de la palabra en cuestión. Las señales de control y los registros se muestran en el diagrama de bloques en la Fig. 7-24.

El *registro de dirección* de memoria especifica la memoria de palabras seleccionada. Cada palabra en una memoria está asignada a un número de especificación que principia desde 0 hasta el número máximo de palabras disponibles. Para comunicarse con una palabra de memoria específica, su número de localización, o *dirección*, se transfiere al registro de dirección. Los circuitos internos de la unidad de memoria aceptan esta dirección del registro y abren las trayectorias necesarias para seleccionar

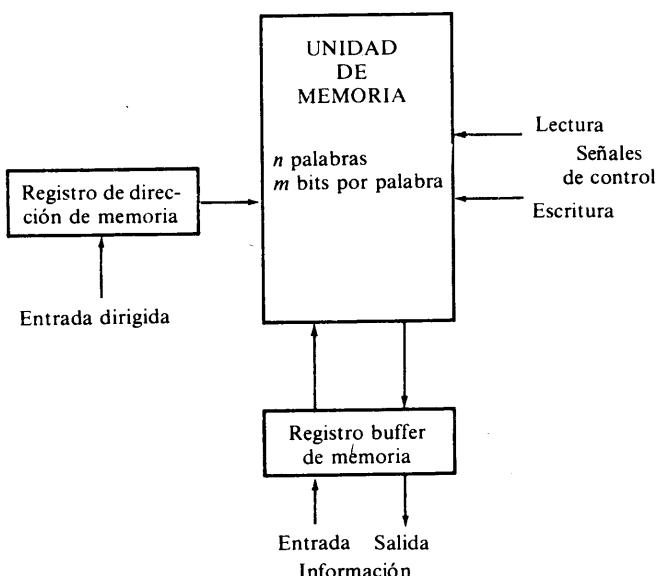


Figura 7-24 Diagrama de bloques de una memoria mostrando la comunicación con el medio ambiente.

la palabra llamada. Un registro de dirección con  $n$  bits puede especificar hasta  $2^n$  palabras de memoria. Las unidades de memoria de computadora pueden variar desde 1 024 palabras, que requieren un registro de dirección de 10 bits, a  $1\,048\,576 = 2^{20}$  palabras, que requiere un registro de dirección de 20-bit.

Las dos señales de control aplicadas a la unidad de memoria se denominan *lectura* y *escritura*. Una señal de escritura especifica una función de transferencia interna; una señal de lectura especifica una función de transferencia externa. Cada una se referencia desde la unidad de memoria. Al aceptar una de las señales de control, los circuitos internos de la unidad de memoria proporcionan la función deseada. Ciertos tipos de unidades de almacenamiento, debido a las características de su componente, destruyen la información almacenada en una celda cuando el bit en esa celda se lee al exterior. Dicha unidad se dice que es una memoria de lectura destructiva, en contraposición a una memoria no destructiva donde la información permanece en la celda después de que se lee al exterior. En cualquier caso, la información anterior siempre se destruye cuando se escribe información nueva. La secuencia del control interno en una memoria de lectura destructiva debe proporcionar señales de control que provoquen que la palabra se restablezca en sus celdas binarias si la aplicación exige una función no destructiva.

La transferencia de información hacia y desde los registros en memoria y el medio externo se comunica a través de un registro común llamado *registro buffer* de memoria (otros nombres son *registros de información* y *registro de almacén*). Cuando la unidad de memoria recibe una señal de control para *escritura*, el control interno interpreta el contenido de registro buffer como la configuración de bits de la palabra que va a almacenarse en un registro de memoria. Con una señal de control de *lectura*, el control interno envía la palabra de un registro de memoria al registro buffer. En cada caso el contenido del registro de dirección especifica el registro particular de memoria referenciado para escritura o lectura.

Ahora se resumen las características de transferencia de información de una unidad de memoria mediante un ejemplo. Considérese una unidad de memoria de 1 024 palabras con ocho bits por palabra. Para especificar 1 024 palabras, se necesita una dirección de diez bits, ya que  $2^{10} = 1\,024$ . Por consiguiente, el registro de dirección debe contener diez flip-flops. El registro buffer debe tener ocho flip-flops para almacenar el contenido de palabras transferidas hacia adentro y hacia afuera de la memoria. La unidad de memoria tiene 1 024 registros, con números asignados de dirección desde 0 hasta 1 023.

En la Fig. 7-25 se muestra el contenido inicial de tres registros: registro de dirección de memoria (MAR), registro buffer de memoria (MBR), y registro de memoria dirigida por MAR. Ya que el número binario equivalente en MAR es decimal 42, el registro de memoria dirigida por MAR es el que tiene el número de dirección 42.

La secuencia de las operaciones necesarias para comunicarse con la unidad de memoria con el propósito de transferir una palabra a la MBR es:

1. Transferencia de los bits de dirección de la palabra seleccionada al MAR.
2. Activación del control de entrada de *lectura*.

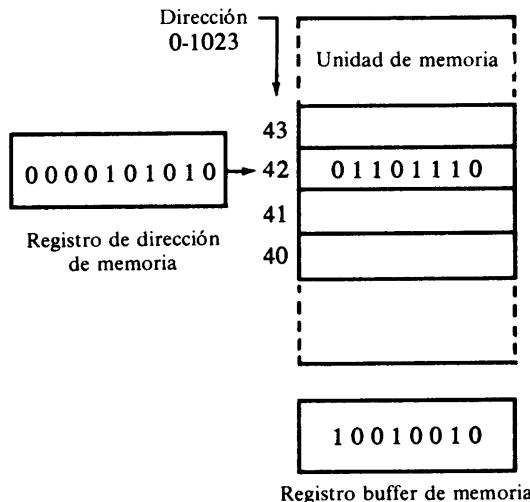


Figura 7-25 Valores iniciales de registros.

El resultado de la operación de lectura se indica en la Fig. 7-26(a). La información binaria al presente almacenada en el registro de memoria 42 se transfiere al registro MBR.

La secuencia de las operaciones necesarias para almacenar una palabra nueva en la memoria es:

1. Transferencia de los bits de dirección de la palabra seleccionada al registro MAR.
2. Transferencia de los bits de información de la palabra al registro MBR.
3. Activación de la entrada de control de *escritura*.

El resultado de la operación de escritura se indica en la Fig. 7-26(b). Los bits de información que provienen del MBR se almacenan en el registro de memoria 42.

En el ejemplo anterior, se supone una unidad de memoria con la propiedad de lectura al exterior no destructiva. Tales memorias pueden construirse con IC semiconductores. Retienen la información en el registro de memoria cuando el registro se muestrea durante el proceso de lectura de modo que no ocurre pérdida de información. Otro componente de uso común en las unidades de memoria es el núcleo magnético. En forma característica un núcleo magnético tiene lectura destructiva, esto es, pierde la información binaria almacenada durante el proceso de lectura. En la Sección 7-8 se presentan ejemplos de memoria de semiconductor y de núcleo magnético.

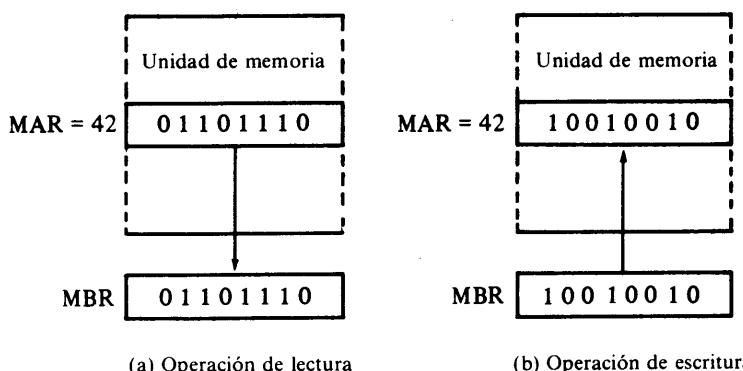
Debido a su propiedad de lectura destructiva, una memoria de núcleo magnético debe proporcionar una función adicional de control para restablecer la palabra en el registro de memoria. Una señal de control de lectura aplicada a una memoria de núcleo magnético transfiere el contenido de la palabra dirigida a un registro externo y,

al mismo tiempo, el registro de memoria se despeja en forma automática. La secuencia del control interno en una memoria de núcleo magnético entonces proporciona señales apropiadas que provocan el restablecimiento de la palabra en el registro de memoria. La transferencia de información en una memoria de núcleo magnético durante una operación de lectura se indica en la Fig. 7-27. Una operación de lectura destructiva transfiere la palabra seleccionada a un MBR pero deja el registro de memoria con todos los 0. La operación normal de memoria requiere que el contenido de la palabra seleccionada permanezca en la memoria después de una operación de lectura. Por tanto, es necesario pasar a través de una operación de *restablecimiento* que escribe el valor en el MBR al registro seleccionado de memoria. Durante la operación de restablecimiento, el contenido del MAR y el MBR debe permanecer sin cambio.

Una entrada de control de escritura aplicada a una memoria de núcleo magnético provoca una transferencia de información como se indica en la Fig. 7-28. Para transferir información nueva a un registro seleccionado, la información anterior debe borrarse primero despejando todos los bits de la palabra a 0. Despues de que se ha hecho esto, el contenido del MBR puede transferirse a la palabra seleccionada. El MAR no debe cambiar durante la operación para asegurar que la misma palabra seleccionada que se despeja es la que recibe la información nueva.

Una memoria de núcleo magnético requiere dos medios ciclos, ya sea para lectura o escritura. El tiempo que necesita la memoria para pasar a través de ambos medios ciclos se conoce como *tiempo de ciclo de memoria*.

El modo de acceso de un sistema de memoria se determina por el tipo de componente usado. En una memoria de *acceso aleatorio*, debe considerarse que los registros están separados en el espacio, con cada registro ocupando una localización especial particular como en una memoria de núcleo magnético. En una memoria de *acceso secuencial*, la información almacenada en cierto medio no es accesible de inmediato, pero está disponible sólo a ciertos intervalos de tiempo. Una unidad de cinta magnética es de este tipo. Cada localización de memoria pasa las cabezas de lectura en turno, pero la información se lee hacia afuera sólo cuando se ha alcanzado la palabra requerida. El *tiempo de acceso* de una memoria es el tiempo requerido para



**Figura 7-26** Transferencia de información durante las operaciones de lectura y escritura.

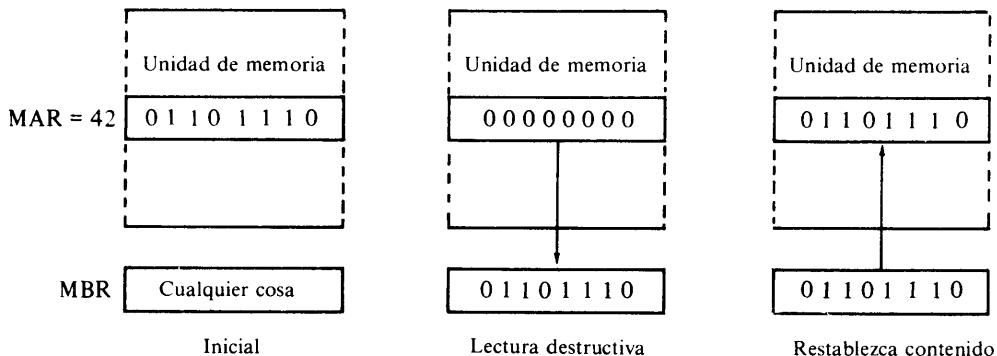


Figura 7-27 Transferencia de información en una memoria de núcleo magnético durante una operación de lectura.

seleccionar una palabra y leerla o bien escribirla. En una memoria de acceso aleatorio, el tiempo de acceso es siempre el mismo sin importar la localización particular de la palabra en el espacio. En una memoria secuencial, el tiempo de acceso depende de la posición de la palabra al tiempo del requerimiento. Si la palabra está surgiendo justamente del almacenamiento al tiempo que se le requiere, el tiempo de acceso es precisamente el tiempo necesario para leerla o escribirla. Pero si sucede que la palabra está en la última posición, el tiempo de acceso también incluye el tiempo requerido para que las otras palabras se muevan pasando las terminales. En consecuencia, el tiempo de acceso en una memoria secuencial es variable.

Las unidades de memoria cuyos componentes pierden la información almacenada con el tiempo o cuando se desconecta la potencia se dice que son *volátiles*. Una unidad semiconductor de memoria es de esta categoría ya que sus celdas binarias necesitan potencia externa para mantener las señales necesarias. En contraste, una unidad de memoria no volátil, como un núcleo magnético o discos magnéticos, retiene su información almacenada después de la remoción de la potencia. Esto se debe a que la información almacenada en los componentes magnéticos se manifiesta por la dirección de magnetización, la cual se retiene cuando se desconecta la potencia. En las computadoras digitales es deseable una propiedad no volátil porque muchos progra-

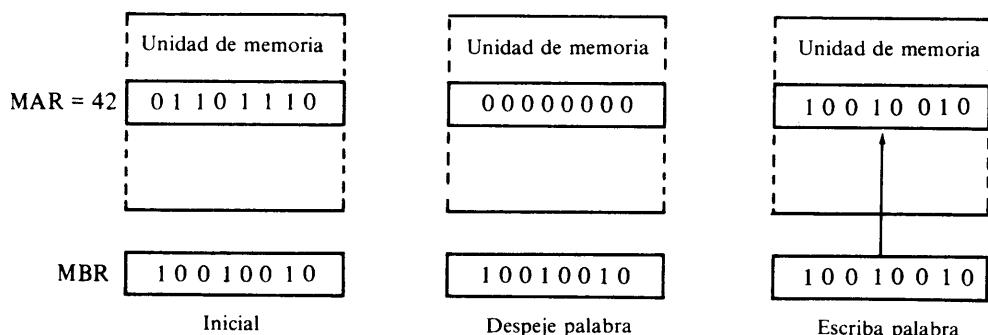


Figura 7-28 Transferencia de información en una memoria de núcleo magnético durante una operación de escritura.

mas útiles se dejan en forma permanente en la unidad de memoria. Cuando la potencia se desconecta y se vuelve a conectar otra vez, los programas previamente almacenados y otra información no se pierde, sino que continúan su residencia en la memoria.

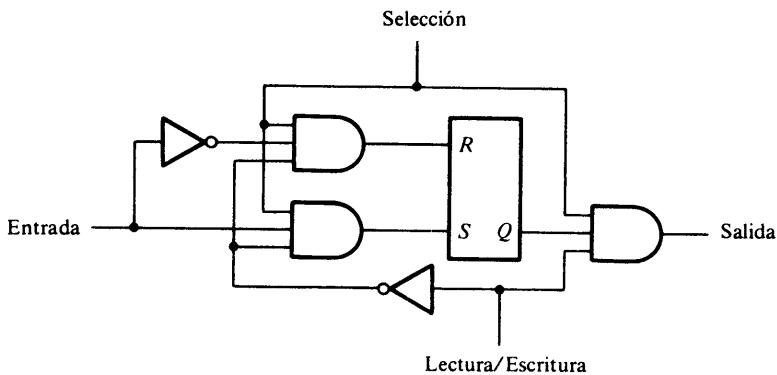
## 7-8 EJEMPLOS DE MEMORIAS DE ACCESO ALEATORIO

La construcción interna de dos tipos diferentes de memorias de acceso aleatorio se presenta en forma de diagrama en esta sección. El primer tipo está construido con flip-flops y compuertas y el segundo con núcleos magnéticos. Para facilitar que se incluya completa la unidad de memoria en un diagrama, debe usarse una capacidad de almacenamiento indicada. Por esta razón, las unidades de memoria que se presentan aquí tienen una capacidad pequeña de 12 bits arreglados en cuatro palabras de tres bits cada una. Las memorias de acceso aleatorio comerciales pueden tener una capacidad de miles de palabras y cada palabra puede variar entre 8 y 64 bits. La construcción lógica de las unidades de memoria de gran capacidad sería una extensión directa de la configuración que aquí se muestra.

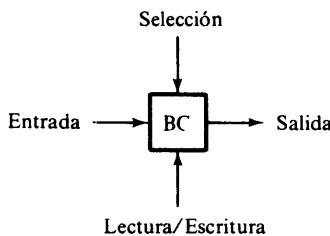
### Circuito integrado de memoria

La construcción interna de una memoria de acceso aleatorio de  $m$  palabras con  $n$  bits por palabra consta de  $m \times n$  celdas de almacenamiento binario y la lógica asociada para seleccionar palabras individuales. La celda de almacenamiento binario es el bloque básico de construcción de una unidad de memoria. El circuito equivalente de una celda binaria que almacena un bit de información se muestra en la Fig. 7-29. Aunque la celda se muestra para incluir compuertas y un flip-flop, está construida en forma interna con dos transistores que tienen entradas múltiples. Una celda de almacenamiento binario debe ser muy pequeña con objeto de poder agrupar tantas celdas como sea posible en la pequeña área disponible en la pastilla de circuito integrado. La celda binaria tiene tres entradas y una salida. La entrada de selección capacita la celda para lectura o escritura. La entrada de lectura/escritura determina la operación de la celda cuando se selecciona. Un 1 en la entrada de lectura/escritura forma una trayectoria desde el flip-flop a la terminal de salida. La información en la terminal de entrada se transfiere al flip-flop cuando el control de lectura/escritura es 0. Obsérvese que el flip-flop opera sin pulsos de reloj y que su propósito es almacenar el bit de información en la celda binaria.

Las memorias en el circuito integrado algunas veces tienen una línea única para el control de lectura y escritura. Un estado binario en la línea única especifica una operación de lectura y el otro estado especifica una operación de escritura. Además, se incluye una o más líneas de habitación para proporcionar los medios para seleccionar el IC y para la expansión de varios paquetes en una unidad de memoria con un número más grande de palabras. En la Fig. 7-30 se muestra la construcción lógica de un IC RAM. Consta de 4 palabras de 3 bits cada una, para un total de 12 celdas binarias. Las casillas pequeñas etiquetadas BC representan celdas binarias, y las tres entradas y la salida única en cada BC son como se especificó en el diagrama de la Fig. 7-29.



(a) Diagrama lógico



(b) Diagrama de bloque

**Figura 7-29** Celda de memoria.

Las dos líneas de entrada de dirección van a través de un decodificador interno de 2-a-4 línea. El decodificador se habilita con la entrada de habilitación de memoria. Cuando la habilitación de memoria es 0, todas las salidas de decodificador son 0 y ninguna de las palabras de memoria se selecciona. Cuando la habilitación de la memoria es 1, se selecciona una de cuatro palabras, dependiendo del valor de las dos líneas de dirección. Ahora, con el control de lectura/escritura en 1, los bits de la palabra seleccionada pasan a través de las tres compuertas OR a las terminales de salida. Las celdas binarias no seleccionadas producen 0 en las entradas de las compuertas OR y no tienen efecto en las salidas. Con el control de lectura/escritura en 0, la información disponible en las líneas de entrada se transfiere a las celdas binarias de la palabra seleccionada. Las celdas binarias no seleccionadas en las otras palabras están inhabilitadas por sus entradas de selección y sus valores previos permanecen sin cambio. Con el control de habilitación de memoria en 0, el contenido de todas las celdas en la memoria permanece sin cambio, sin importar el valor del control de lectura/escritura.

Los IC RAM tienen construcción interna con celdas que tienen una capacidad OR alambrada. Esto elimina la necesidad de las compuertas OR en el diagrama. Las

líneas de las salidas externas también pueden formar lógica alambrada para facilitar la conexión de dos o más paquetes IC para formar una unidad de memoria con un número mayor de palabras.

### Memoria de núcleo magnético

Una memoria de núcleo magnético usa núcleos magnéticos para almacenar la información binaria. Un núcleo magnético tiene forma toroidal y está hecho de material magnético. En contraste con un flip-flop semiconductor que necesita sólo una cantidad física, como por ejemplo voltaje, para su operación un núcleo magnético emplea

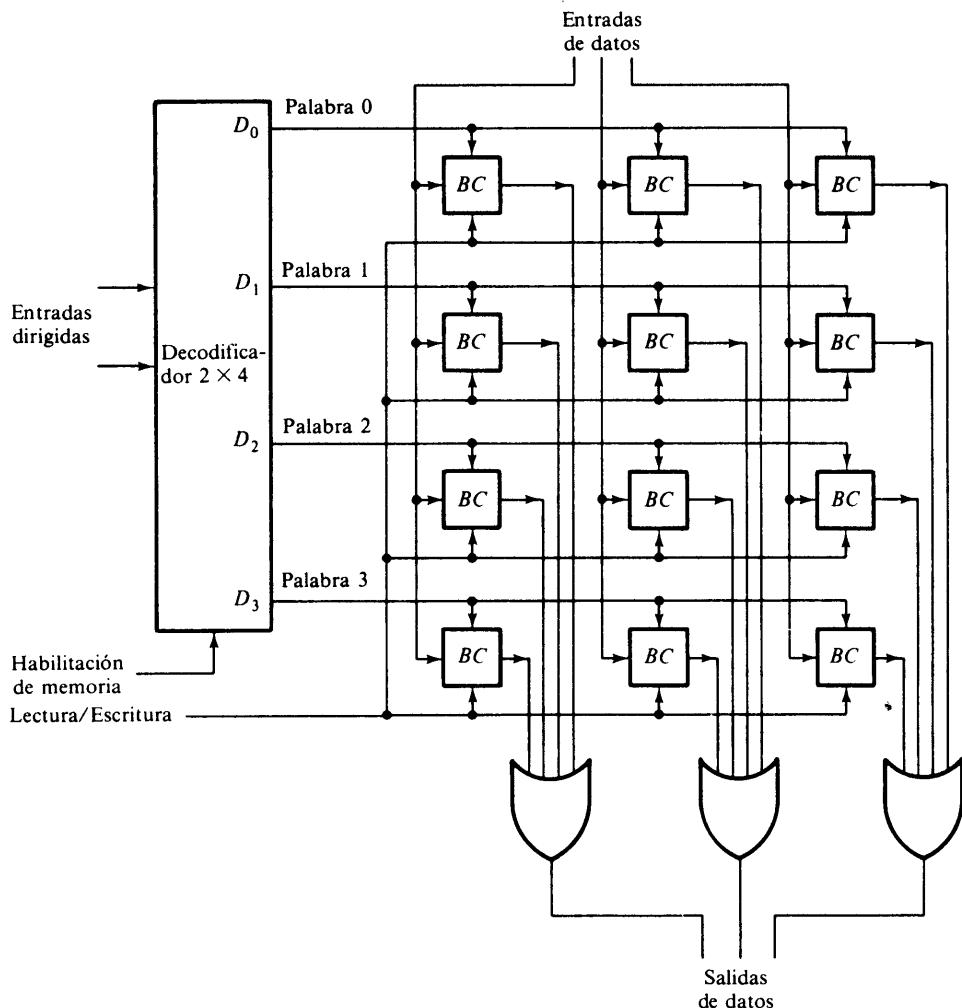


Figura 7-30 Memoria en circuito integrado.

tres cantidades físicas: corriente, flujo magnético y voltaje. La señal que excita al núcleo es un pulso de *corriente* en un alambre que pasa a través del núcleo. La información binaria almacenada se representa por la dirección del *flujo magnético* dentro del núcleo. La información binaria de salida se extrae mediante un alambre que se eslabona al núcleo en la forma de un pulso de *voltaje*.

La propiedad física que hace que un núcleo magnético sea adecuado para el almacenamiento binario es su circuito de histéresis, que se muestra en la Fig. 7-31(c). Este circuito es una gráfica de corriente comparada con el flujo magnético y tiene la forma de un circuito cuadrado. Con corriente cero, un flujo que es positivo (dirección en sentido contrario a las manecillas del reloj) o bien negativo (dirección en el sentido de las manecillas del reloj) permanece en el núcleo magnetizado. Una dirección, por ejemplo magnetización contraria al sentido de las manecillas del reloj, se usa para representar un 1 y la otra para representar un 0.

Un pulso de corriente aplicado al devanado a través del núcleo puede cambiar la dirección de magnetización. Como se muestra en la Fig. 7-31(a), la corriente en la dirección hacia abajo produce flujo en la dirección de las manecillas del reloj, provocando que el núcleo pase al estado 0. En la Fig. 7-31(b) se muestra la corriente y la dirección del flujo para almacenar un 1. La trayectoria que toma el flujo cuando se aplica el pulso de corriente se indica con flechas en el circuito de histéresis.

La toma de la lectura de la información binaria almacenada en el núcleo se complica por el hecho de que el flujo no puede detectarse cuando no tiene cambio. Sin embargo, si el flujo cambia con respecto al tiempo, induce un voltaje en un alambre que se enlaza con el núcleo. Por tanto, la toma de lectura puede realizarse con la aplicación de una corriente en la dirección negativa como se muestra en la Fig. 7-32. Si el núcleo está en el estado 1, la corriente invierte la dirección de magnetización, y el cambio resultante del flujo produce un pulso de voltaje en el alambre sensor. Si el núcleo ya está en el estado 0, la corriente negativa deja magnetizado al núcleo en la misma dirección, causando una perturbación muy ligera del flujo magnético, lo cual resulta en un voltaje de salida muy pequeño en el alambre sensor. Obsérvese que ésta es una toma de lectura destructiva, ya que la corriente leída siempre devuelve el núcleo al estado 0. Se pierde el valor almacenado previamente.

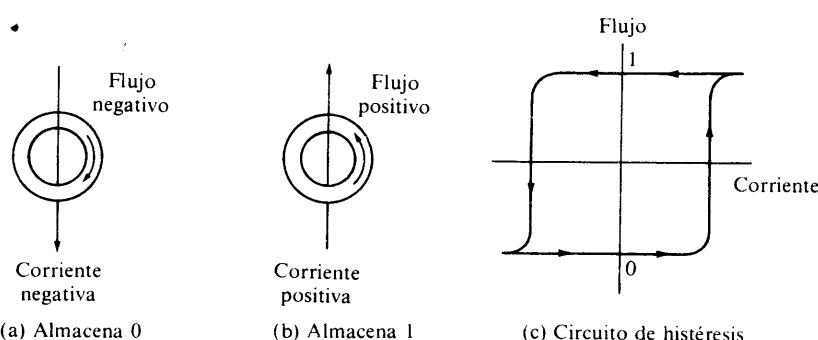


Figura 7-31 Almacén de un bit en un núcleo magnético.

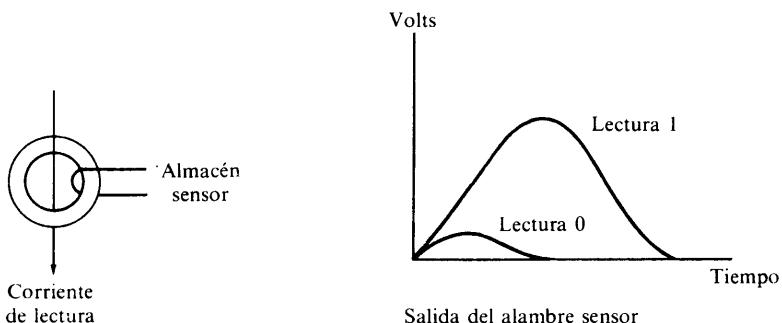


Figura 7-32 Lectura de un bit a partir de un núcleo magnético.

En la Fig. 7-33 se muestra la organización de una memoria de núcleo magnético que contiene cuatro palabras con tres bits cada una. Al compararla con la unidad de memoria IC que se muestra en la Fig. 7-30, se observa que la celda binaria ahora es un núcleo magnético con los alambres eslabonados. La excitación del núcleo se lleva a cabo mediante un pulso de corriente generado en un impulsor (DR). La información de salida pasa a través de un amplificador sensor (SA) cuyas salidas establecen los flip-flops correspondientes en el registro buffer. Se eslabonan tres alambres a cada núcleo. El alambre de palabras se excita por un impulsor de palabra que pasa a través de los tres núcleos de una palabra. Un alambre de bit se excita con un impulsor de bit y pasa a través de cuatro núcleos en la misma posición de bit. Los alambres sensores eslabonan los mismos núcleos como el alambre de bit y se aplica a un amplificador sensor que forma el pulso de voltaje cuando se lee un 1 y rechaza la pequeña perturbación cuando se lee un 0.

Durante una operación de lectura, un pulso de corriente del impulsor de palabra se aplica a los núcleos de la palabra seleccionada por el decodificador. La corriente de lectura está en la dirección negativa (Fig. 7-32) y causa que todos los núcleos de la palabra seleccionada pasen al estado 0, sin importar su estado previo. Los núcleos que previamente contienen un 1 cambian su flujo e inducen un voltaje en su alambre sensor. Los flujos de los núcleos que ya contienen un 0 no cambian. El pulso de voltaje en un alambre sensor de núcleos con un 1 previo se amplifica en el amplificador sensor y establece el flip-flop correspondiente en el registro buffer.

Durante una operación de escritura, el registro buffer mantiene la información por almacenarse en la palabra especificada por el registro de dirección. Se supone que todos los núcleos en la palabra seleccionada están despejados inicialmente, esto es, todos están en el estado 0, de modo que los núcleos que requieren un 1 necesitan pasar por un cambio de estado. Se genera en forma simultánea un pulso de corriente en el impulsor de palabra seleccionado por el decodificador y en el impulsor de bit, cuyo flip-flop correspondiente de registro buffer contiene un 1. Ambas corrientes están en la dirección positiva, pero su magnitud es sólo la mitad de la necesaria para cambiar el flujo al estado 1. Esta media corriente es demasiado pequeña por sí misma para cambiar la dirección de magnetización. Pero la suma de dos medias corrientes es suficiente para cambiar la dirección de la magnetización al estado 1. Un núcleo cambia

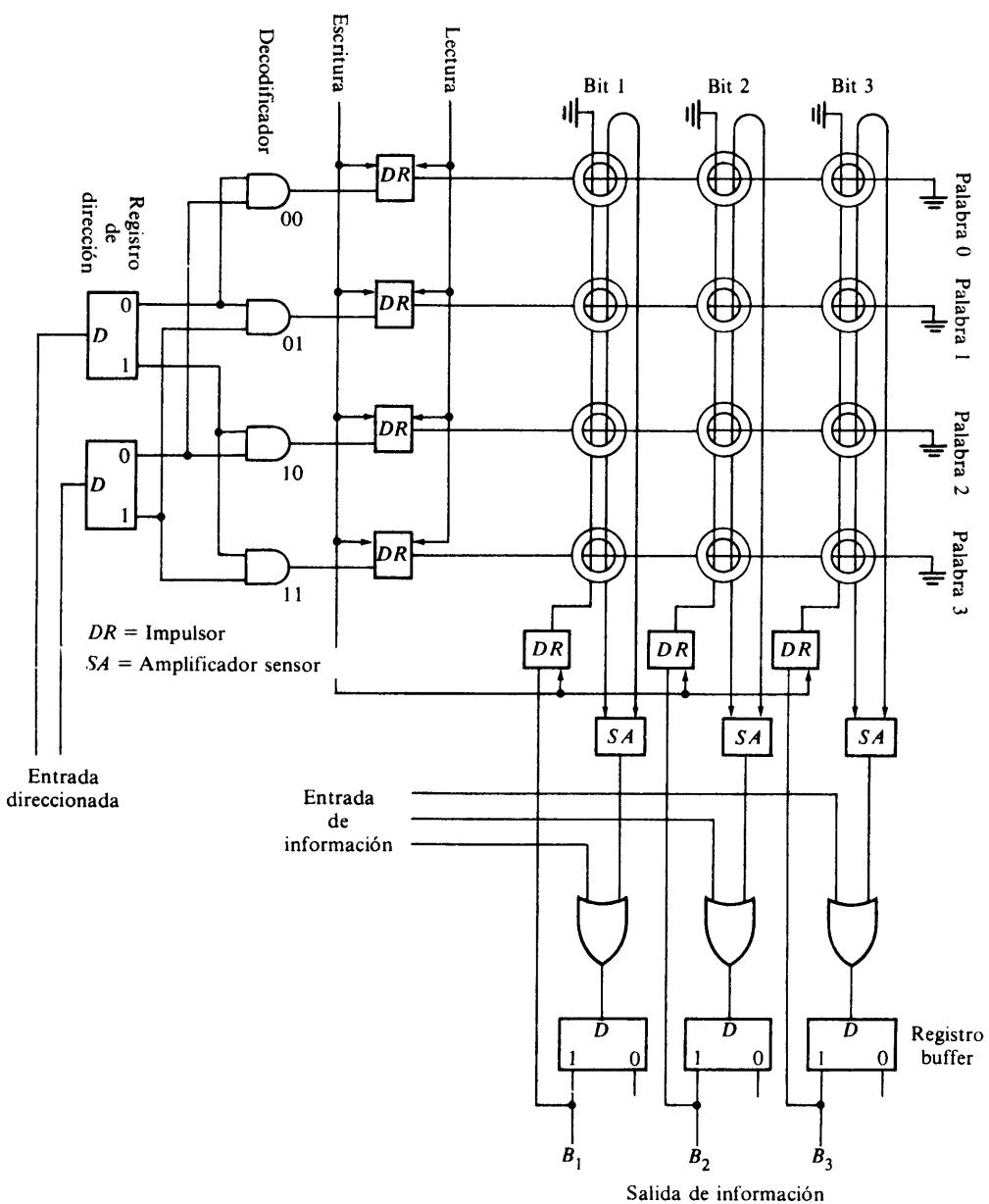


Figura 7-33 Unidad de memoria en núcleo magnético.

al estado 1 sólo si hay una coincidencia de dos medias corrientes que provienen de un impulsor de palabra y un impulsor de bit. La dirección de magnetización de un núcleo no cambia si recibe sólo media corriente de uno de los impulsores. El resultado es que la magnetización de los núcleos se cambia al estado 1 sólo si se intersecan los alambres de palabra y bit esto es, sólo en la palabra seleccionada y sólo en la posición de bit en la cual el registro buffer está en 1.

Las operaciones de lectura y de escritura que se describieron con anterioridad son incompletas, porque la información almacenada en la palabra que se selecciona se destruye por el proceso de lectura y la operación de escritura trabaja en forma apropiada sólo si los núcleos están despejados inicialmente. Como se mencionó en la Sección 7-7, una operación de lectura debe ir seguida por otro ciclo que establece valores previamente almacenados en los núcleos. Una operación de escritura está precedida por un ciclo que despeja los núcleos de la palabra seleccionada.

El restablecimiento de la operación durante un ciclo de lectura es equivalente a una operación de escritura, la cual, en efecto, escribe la información leída previamente desde el registro buffer devolviéndola a la palabra seleccionada. La operación de despeje durante un ciclo de escritura es equivalente a una operación de lectura que destruye la información almacenada pero evita que la información de lectura alcance el registro buffer al inhibir el amplificador sensor. Los ciclos de restablecimiento y despeje normalmente se inician por el control interno de memoria, de modo que la unidad de memoria tiene la apariencia exterior de poseer una propiedad de extracción de lectura no destructiva.

## BIBLIOGRAFIA

1. *The TTL Data Book for Design Engineers*. Dallas: Texas Instruments, Inc., 1976.
2. Blakeslee, T. R., *Digital Design with Standard MSI and LSI*. New York: John Wiley & Sons, 1975.
3. Barna A., y D. I. Porat, *Integrated Circuits in Digital Electronics*. New York: John Wiley & Sons, 1973.
4. Taub, H., y D. Schilling, *Digital Integrated Electronics*. New York: McGraw-Hill Book Co., 1977.
5. Grinich, V. H., y H. G. Jackson, *Introduction to Integrated Electronics*. New York: McGraw-Hill Book Co., 1975.
6. Kostopoulos, G. K., *Digital Engineering*. New York: McGraw-Hill Book Co., 1975.
7. Scott, N. R., *Electronic Computer Technology*. New York: McGraw-Hill Book Co., 1970, cap. 10.
8. Kline, R. M., *Digital Computer Design*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1977, cap. 9.

## PROBLEMAS

- 7-1. El registro en la Fig. 7-1 transfiere la información de entrada a los flip-flops cuando la entrada  $CP$  pasa a través de una transición de borde positivo. Modifique el circuito de modo que la información de entrada se transfiera al registro cuando un pulso de reloj pasa a través de una transición de borde negativo, siempre que una entrada de carga de control es igual al binario 1.
- 7-2. El registro en la Fig. 7-3 carga las entradas durante una transición negativa de un pulso de reloj. ¿Qué cambios internos son necesarios para que las entradas se carguen durante el borde negativo del pulso?
- 7-3. Verifique el circuito en la Fig. 7-5 usando mapas para simplificar las ecuaciones del estado siguiente.
- 7-4. Diseñe el circuito secuencial cuya tabla de estado se da a continuación, usando un registro de 2-bit y compuertas combinacionales.

Estado presente		Entrada	Estado siguiente	
A	B		A	B
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	0	1

- 7-5. Diseñe un circuito secuencial cuyo diagrama de estado está dado en la Fig. 6-27 usando un registro de 3-bit y una ROM 16 × 4.
- 7-6. El contenido de un registro de corrimiento de 4-bit en la forma inicial es 1101. El registro se corre seis veces a la derecha, con la entrada serial de 101101. ¿Cuál es el contenido del registro después de cada corrimiento?
- 7-7. ¿Cuál es la diferencia entre las transferencias serial y paralela? ¿Qué tipo de registro se usa en cada caso?
- 7-8. El registro de corrimiento bidireccional de 4-bit que se muestra en la Fig. 7-9 está encerrado dentro de un paquete IC.
- Dibuje un diagrama de bloques del IC mostrando todas las entradas y salidas.
  - Dibuje un diagrama de bloques usando tres IC para producir un registro de corrimiento bidireccional de 12-bit.
- 7-9. El adiconador serial que se muestra en la Fig. 7-10 usa registros de corrimiento de 4-bit. El registro  $A$  mantiene el número binario 0101 y el registro  $B$  mantiene 0111. El flip-flop  $Q$  de la cuenta que se lleva está despejado al inicio. Haga la lista de los valores binarios en el registro  $A$  y el flip-flop  $Q$  después de cada corrimiento.

- 7-10. ¿Qué cambio se necesita en el circuito que se muestra en la Fig. 7-11 para convertirlo en un circuito que reste el contenido de  $B$  del contenido de  $A$ ?
- 7-11. Diseñe un contador serial; en otras palabras, determine el circuito que debe incluirse en forma externa con un registro de corrimiento con objeto de obtener un contador que opere en una forma serial.
- 7-12. Dibuje el diagrama de un contador binario de ondulación de 4-bit usando flip-flops que disparen en el borde positivo.
- 7-13. Un flip-flop tiene un retardo de 20-ns desde el tiempo en que su entrada  $CP$  pasa desde 1 a 0 al tiempo que se complementa la salida. ¿Cuál es el retardo máximo en un contador binario de ondulación de 10-bit que usa esos flip-flops? A qué frecuencia máxima puede operar el contador con confiabilidad?
- 7-14. ¿Cuántos flip-flops deben complementarse en un contador binario de ondulación de 10-bit para alcanzar la cuenta siguiente después de 011111111?
- 7-15. Dibuje el diagrama de un contador binario de ondulación de 4-bit hacia abajo usando flip-flops que disparen en la (a) transición de borde positivo y (b) transición de borde negativo.
- 7-16. Dibuje un diagrama de temporizado similar al que se muestra en la Fig. 7-15 para el contador binario de ondulación que se muestra en la Fig. 7-12.
- 7-17. Determine el estado siguiente para cada uno de los seis estados sin uso en el contador de ondulación BCD que se muestra en la Fig. 7-14. ¿El contador arranca por sí mismo?
- 7-18. El contador de ondulación que se muestra en la Fig. P7-18 usa flip-flop que dispara en la transición de borde negativo de la entrada  $CP$ . Determine la secuencia de conteo del contador. ¿El contador arranca por sí solo?

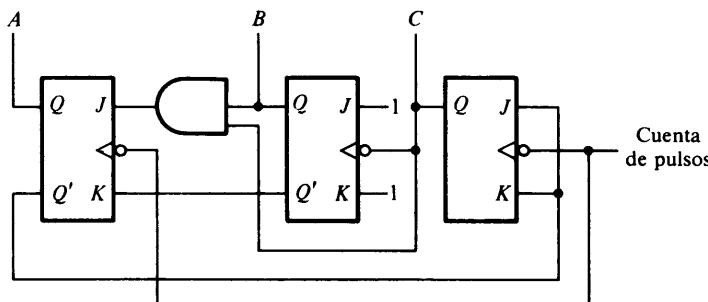


Figura P7-18 Contador de ondulación.

- 7-19. ¿Qué sucede en el contador en la Fig. 7-18 si tanto la entrada de incremento como la entrada de decremento son iguales a 1 al mismo tiempo? Modifique el circuito de modo de que cuente hacia arriba si ocurre esta condición.
- 7-20. Verifique las funciones de entrada del flip-flop del contador síncrono BCD especificado en la Tabla 7-5. Dibuje el diagrama lógico del contador BCD e incluya una entrada de control para habilitar el conteo.
- 7-21. Diseñe un contador síncrono BCD con flip-flops  $JK$ .

- 7-22. Muestre las conexiones externas para cuatro IC contadores binarios con carga paralela (Fig. 7-19) para producir un contador binario de 16-bit. Use un diagrama de bloques para cada IC.
- 7-23. Construya un contador BCD usando el circuito MSI que se muestra en la Fig. 7-19.
- 7-24. Construya un contador mod-12 usando el circuito MSI especificado en la Fig. 7-19. Dé cuatro alternativas.
- 7-25. Mediante el uso de dos circuitos MSI como se especifica en la Fig. 7-19, construya un contador binario que cuente desde 0 hasta el binario 64.
- 7-26. Utilizando la variable *paro* de la Fig. 7-21 como señal de arranque, construya un segundo control de tiempo de palabra que permanezca encendido por un periodo de 16 pulsos de reloj.
- 7-27. Muestre un contador binario de  $n$ -bit conectado a un decodificador de  $n$ -a- $2^n$  líneas es equivalente a un contador de anillo con  $2^n$  flip-flops. Dibuje los diagramas de bloque de ambos circuitos para  $n = 3$ . ¿Cuántas señales de temporizado se generan?
- 7-28. Incluya una entrada de habilitación a decodificador en la Fig. 7-22(b) y conéctelo a los pulsos de reloj. Dibuje las señales de temporizado que se generan ahora en las salidas del decodificador.
- 7-29. Complete el diseño del contador Johnson que se muestra en la Fig. 7-23, ilustrando las salidas de las ocho señales de temporizado.
- 7-30. (a) Haga la lista de los ocho estados sin uso en el contador de anillo con conmutación en cola en la Fig. 7-23. Determine el estado siguiente para cada estado sin uso y muestre que, si el circuito se encuentra por sí mismo en un estado inválido, no regresa a un estado inválido. (b) Modifique el circuito como se recomienda en el texto y muestre que (1) el circuito produce la misma secuencia de estados como se muestra en la lista que se incluye en la Fig. 7-23(b) y (2), el circuito alcanza un estado válido desde cualquiera de los estados sin uso.
- 7-31. Construya un contador Johnson con diez señales de temporizado.
- 7-32. (a) La unidad de memoria que se muestra en la Fig. 7-24 tiene una capacidad de 8 192 palabras de 32 bits por palabra. ¿Cuántos flip-flops se necesitan para el registro de dirección de memoria y el registro buffer de memoria? (b) ¿Cuántas palabras contendrá la unidad de memoria si el registro de dirección tiene 15 bits?
- 7-33. Cuando el número de palabras que van a seleccionarse en una memoria es demasiado grande, es conveniente usar una celda de almacenamiento binario con dos entradas seleccionadas: una *X*(horizontal) y una *Y*(vertical) entrada seleccionada. Tanto *X* como *Y* deben habilitarse para seleccionar la celda.  
(a) Dibuje una celda binaria similar a la que se muestra en la Fig. 7-29 con *X* y *Y* entradas seleccionadas.  
(b) Muestre cómo dos decodificadores  $4 \times 16$  pueden usarse para seleccionar una palabra en una memoria de 256-palabras.
- 7-34. (a) Dibuje un diagrama de bloques de la memoria  $4 \times 3$  que se muestra en la Fig. 7-30. Muestre todas las entradas y todas las salidas. (b) Construya una memoria  $8 \times 3$  usando dos de esas unidades. Utilice una construcción de diagrama de bloques.

- 7-35. Se requiere construir una memoria con 256 palabras, 16 bits por palabra, organizada como se muestra en la Fig. 7-33. Están disponibles núcleos en una matriz de 16 renglones y 16 columnas.
- (a) ¿Cuántas matrices se necesitan?
  - (b) ¿Cuántos flip-flops están en los registros de dirección y buffer?
  - (c) ¿Cuántos núcleos reciben corriente durante un ciclo de lectura?
  - (d) ¿Cuántos núcleos reciben cuando menos media corriente durante un ciclo de escritura?