Máquinas de estado algorítmico (ASM)

8

8-1 INTRODUCCION

La información binaria almacenada en un sistema digital puede clasificarse ya sea como datos o control de información. Los datos son elementos discretos de información que se manipulan para realizar tareas de aritmética, lógica, corrimiento y otras tareas similares de procesamiento de datos. Estas operaciones se implantan con componentes digitales como sumadores, decodificadores, multiplexores, contadores y registros de corrimiento. La información de control proporciona señales de mando que supervisan las diversas operaciones de la sección de datos con objeto de llevar a cabo las tareas deseadas de procesamiento de datos. El diseño lógico de un sistema digital puede dividirse en dos partes distintas. Una parte se ocupa del diseño de los circuitos digitales que llevan a cabo las operaciones de procesamiento de datos. La otra parte se ocupa del diseño del circuito de control que supervisa las operaciones y sus secuencias.

Las relaciones entre el control lógico y el procesador de datos en un sistema digital se muestran en la Fig. 8-1. El subsistema procesador de datos manipula los datos en los registros de acuerdo con los requisitos del sistema. El control lógico inicia los mandos en secuencia apropiada al procesador de datos. El control lógico usa las

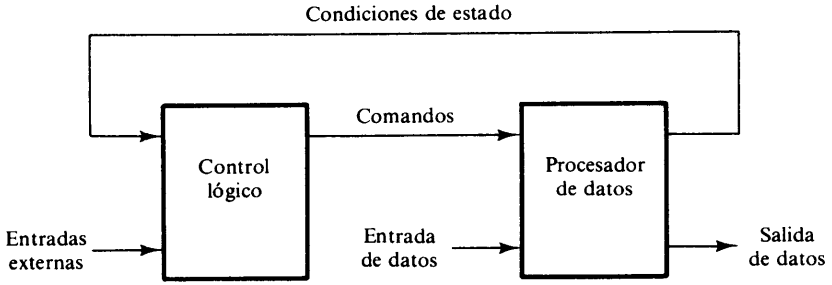


Figura 8-1 Interacción del control y el procesador de datos.

condiciones de estado del procesador de datos para servir como variables de decisión para determinar la secuencia de las señales de control.

El control lógico que genera las señales para dar la secuencia de las operaciones en el procesador de datos es un circuito secuencial cuyos estados internos dictan los mandos de control para el sistema. En cualquier momento el estado del control secuencial inicia un conjunto prescrito de mandos. Dependiendo de las condiciones de estado y otras entradas externas, el circuito secuencial pasa al estado siguiente para iniciar otras operaciones. Los circuitos digitales que actúan como el control lógico proporcionan una secuencia de tiempo de señales para iniciar las operaciones en el procesador de datos y determinar el siguiente estado del mismo subsistema de control.

La secuencia de control y las tareas de procesamiento de datos de un sistema digital se especifican mediante un algoritmo en el hardware. Un algoritmo consta de un número finito de pasos de procedimiento que especifican como obtener una solución a un problema. Un algoritmo de hardware es un procedimiento para implantar el problema con una pieza dada de equipo. La parte del diseño digital más creativa y más desafiante es la formulación de algoritmos de hardware para lograr los objetivos requeridos.

Una forma conveniente de especificar la secuencia de los pasos de proceso y las trayectorias de decisión para un algoritmo es un diagrama de flujo. El diagrama de flujo para un algoritmo de hardware traduce la estipulación en palabras a un diagrama de información que enumera la secuencia de las operaciones junto con las condiciones necesarias para su ejecución. Un diagrama especial de flujo que ha sido desarrollado específicamente para definir algoritmos de hardware digitales se denomina diagrama de máquina de estado algorítmico (ASM). Una máquina de estado es otro término para un circuito secuencial, el cual es la estructura básica de un sistema digital.

El diagrama ASM se asemeja a un diagrama convencional de flujo pero se interpreta en forma algo diferente. Un diagrama convencional de flujo describe la secuencia de los pasos de procedimiento y las trayectorias de decisión para un algoritmo sin ocuparse de sus relaciones en el tiempo. El diagrama ASM describe la secuencia de eventos lo mismo que las relaciones de temporizado entre los estados de un controlador secuencial y los eventos que ocurren cuando pasa de un estado al siguiente. En forma específica está adaptado para especificar con precisión la secuencia de control y las operaciones de procesamiento de datos en un sistema digital, tomando en consideración las restricciones del hardware digital.

Este capítulo presenta un método de diseño digital lógico que usa el diagrama ASM. Los diversos bloques que componen el diagrama se definen primero. Las relaciones de temporizado entre los bloques se explican entonces mediante ejemplos. Las diversas formas de implantar el control lógico se exponen junto con ejemplos de diagramas ASM y los sistemas digitales correspondientes que representan.

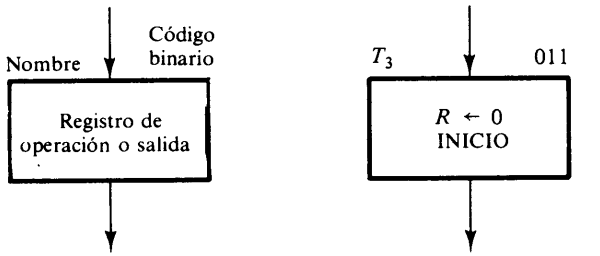
8-2 DIAGRAMA ASM

El diagrama ASM es un tipo especial de diagrama de flujo adecuado para describir las operaciones secuenciales en un sistema digital. El diagrama está compuesto de tres

elementos básicos: la casilla de estado, la casilla de decisión y la casilla condicional. Un estado en la secuencia de control se indica con una casilla de estado, como se muestra en la Fig. 8-2. La forma de la casilla de estado es rectangular dentro de la cual se escriben operaciones de registro o nombres de señal de salida que el control genera mientras se encuentra en este estado. El estado recibe un nombre simbólico, el cual se coloca en la esquina superior izquierda de la casilla. El código binario asignado al estado se coloca en la esquina superior de la derecha. En la Fig. 8-2(b) se muestra un ejemplo específico de una casilla de estado. El estado tiene el nombre simbólico T_3, y el código binario asignado a él es 011. Dentro de la casilla se escribe la operación del registro $R \leftarrow 0$, la cual indica que el registro R se despeja a 0 cuando el sistema está en el estado T_3. Por ejemplo, el nombre START dentro de la casilla puede indicar una señal de salida que inicia cierta operación.

La casilla de decisión escribe el efecto de una entrada en el subsistema de control. Es una casilla con forma de rombo con dos o más trayectorias de salida, como se muestra en la Fig. 8-3. La condición de entrada que va a probarse está escrita dentro de la casilla. Una trayectoria de salida se toma si la condición es cierta y la otra cuando la condición es falsa. Cuando una condición de entrada está asignada a un valor binario, las dos trayectorias se indican por 1 y 0.

Las casillas de estado y decisión se conocen por su uso en los diagramas convencionales de flujo. El tercer elemento, la casilla condicional, es de uso exclusivo en el diagrama ASM. La forma ovalada de la casilla condicional se muestra en la Fig. 8-4. Los lados redondeados la diferencían de la casilla de estado. La



(a) Descripción general (b) Ejemplo específico

Figura 8-2 Caja de estado.

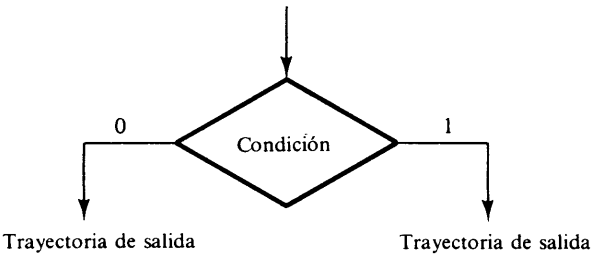


Figura 8-3 Caja de decisión.

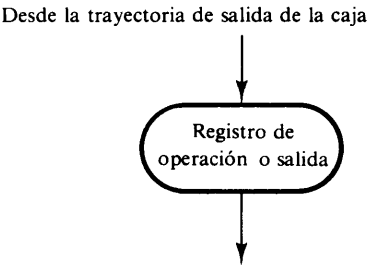


Figura 8-4 Caja condicional.

trayectoria de entrada a la casilla condicional debe llegar desde una de las trayectorias de salida de una casilla de decisión. Las operaciones de registro o salidas listadas dentro de la casilla condicional se generan durante un estado dado siempre que se satisfaga la condición de entrada. En la Fig. 8-5 se muestra un ejemplo con una casilla condicional. El control genera una señal de salida de START cuando se encuentra en el estado T_1. Mientras se encuentra en el estado T_1, el control verifica el estado de la entrada E. Si $E = 1$, entonces R se despeja a 0; en otra forma, R permanece sin cambio. En cualquier caso, el estado siguiente es T_2.

Bloque ASM

Un bloque ASM es una estructura que consta de una casilla de estado y todas las casillas de decisión y condicionales conectadas a sus trayectorias de salida. Un bloque ASM tiene una entrada y cualquier número de trayectorias de salida representadas por la estructura de las casillas de decisión. Un diagrama ASM consta de uno o más bloques interconectados. En la Fig. 8-6 se muestra un ejemplo de un bloque ASM.

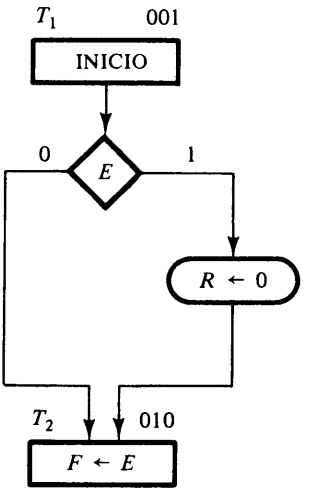


Figura 8-5 Ejemplo con caja condicional.

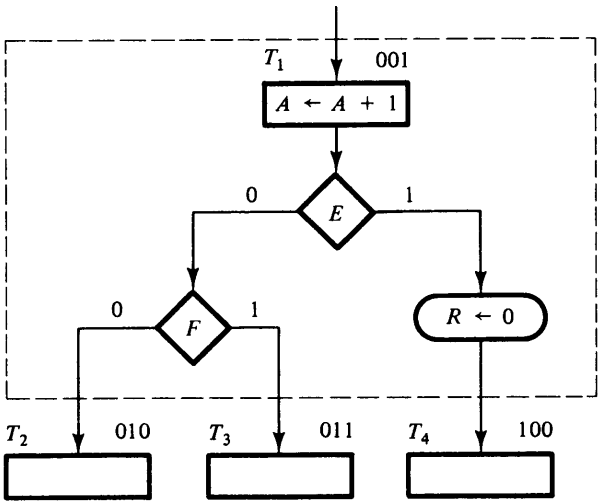


Figura 8-6 Bloque ASM.

Asociadas con el estado T_1 se encuentran dos casillas de decisión y una casilla condicional. El diagrama distingue el bloque con líneas punteadas alrededor de la estructura entera, pero esto por lo común no se hace, ya que el diagrama ASM define en forma única cada bloque mediante su estructura. Una casilla de estado sin casillas de decisión o condicionales constituye un bloque simple.

Cada bloque en el diagrama ASM describe el estado del sistema durante el intervalo de un pulso de reloj. Las operaciones dentro de las casillas de estado y condicionales en la Fig. 8-6 se ejecutan con un pulso común de reloj mientras el sistema se encuentra en el estado T_1. El mismo pulso de reloj también transfiere el sistema controlador a uno de los estados siguientes, T_2, T_3 o T_4, como dictan los valores binarios de E y F.

El diagrama ASM es muy similar a un diagrama de estado. Cada bloque de estado es equivalente a un estado en un circuito secuencial. La casilla de decisión es equivalente a la información binaria escrita a lo largo de las líneas dirigidas que conectan dos estados en un diagrama de estado. Como consecuencia, algunas veces es conveniente convertir el diagrama en un diagrama de estado y entonces usar los procedimientos de circuito secuencial para diseñar el control lógico. Como una ilustración, el diagrama ASM en la Fig. 8-6 se dibuja como un diagrama de estado en la Fig. 8-7. Los tres estados están simbolizados por círculos con su valor binario escrito dentro de cada círculo. Las líneas dirigidas indican las condiciones que determinan el Estado siguiente. Las operaciones incondicionales y condicionales que deben llevarse a cabo no se indican en el diagrama de estado.

Operaciones de registro

Un sistema digital con bastante frecuencia se define por los registros que contiene y las operaciones que se realizan en los datos almacenados en ellos. Un registro en su sentido

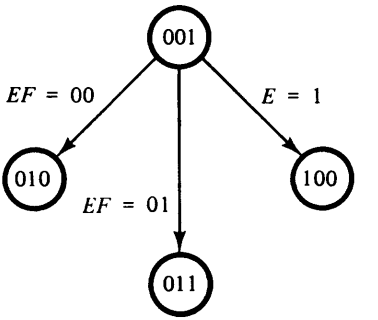


Figura 8-7 Diagrama de estado equivalente a la carta ASM en la Fig. 8-6.

más amplio incluye registros de almacenamiento, registros de corrimiento, contadores y flip-flops solos. Los ejemplos de operaciones de registros son corrimiento, incremento, adición, despeje y transferencia de datos. Algunas veces es conveniente adoptar una notación adecuada para describir las operaciones llevadas a cabo entre los registros.

En la Tabla 8-1 se dan ejemplos de notación simbólica para algunas operaciones de registro. Un registro se indica por una o más letras mayúsculas como A, B, o RA. Las celdas individuales o flip-flops dentro de un registro de n-bit se numeran en secuencia desde 1 a n o desde 0 a $n-1$. Un flip-flop sólo se considera como un registro de 1-bit. La transferencia de datos de un registro a otro se simboliza por una flecha dirigida que denota una transferencia de contenido de un registro fuente a un registro de destino. La operación de despejar registro se simboliza por una transferencia de 0 al registro. Un Flip-flop sólo puede establecerse en 1 o despejarse a 0. Para incrementar un registro en 1, es necesario que el registro sea capaz de contar hacia arriba como en un contador binario. La operación de decremento requiere un contador descendente. El contenido de dos registros puede agregarse mediante un circuito sumador. Algunas operaciones, como por ejemplo la operación de corrimiento, no tienen símbolo conocido. En tal caso se usan las palabras "corrimiento a la derecha R" para denotar un corrimiento a la derecha del registro R.

TABLA 8-1 Notación simbólica para las operaciones de registro.

Notación simbólica	Descripción
	Transferencia del contenido del registro
$A \leftarrow B$	B al registro A
$R \leftarrow 0$	Despejar el registro R
$F \leftarrow 1$	Establecer el flip-flop en 1
$A \leftarrow A + 1$	Incrementar el registro A en 1 (cuenta arriba)
$A \leftarrow A - 1$	Disminuir el registro A en 1 (cuenta abajo)
$A \leftarrow A + B$	Agregar el contenido del registro B al registro A

8-3 CONSIDERACIONES DE TEMPORIZADO

El temporizado de todos los registros y flip-flops en un sistema digital se controla por un reloj generador maestro. Los pulsos de reloj se aplican no sólo a los registros de la subsección del procesador de datos, sino también a todos los flip-flops en el circuito lógico. Las entradas también están sincronizadas con los pulsos de reloj porque normalmente se generan como salidas de otro circuito que usa las mismas señales del reloj. Si la señal de entrada cambia en un tiempo arbitrario independientemente del reloj, se le llama una entrada asíncrona. Las entradas asíncronas pueden causar una variedad de problemas, como se expone en el Capítulo 9. Para simplificar el diseño, se supone que todas las entradas están sincronizadas con el reloj y que cambian de estado como respuesta a una transición de borde del pulso de reloj. En forma similar, cualquier salida que es una función del estado presente y una entrada síncrona también estarán sincronizadas.

La mayor diferencia entre un diagrama de flujo convencional y un diagrama ASM está en la interpretación de las relaciones de tiempo entre las diversas operaciones. Por ejemplo, si la Fig. 8-6 fuera un diagrama de flujo convencional, entonces se consideraría que las operaciones listadas siguen una después de otra en secuencia de tiempo: el registro A se incrementa primero y sólo entonces E se evalúa. Si $E = 1$, entonces el registro R se despeja y el control pasa al estado T_4. En otra forma, si $E = 0$, el paso siguiente es evaluar F y pasar al estado T_2 o T_3. En contraste, un diagrama ASM considera el bloque entero como una unidad. Todas las operaciones que están especificadas dentro del bloque deben ocurrir en sincronismo durante la transición en borde del mismo pulso de reloj, mientras el sistema cambia desde T_1 al estado siguiente. Esto se presenta en forma gráfica en la Fig. 8-8. Se supone disparo de borde positivo para todos los flip-flops. La primera transición positiva del reloj transfiere el circuito de control al estado T_1. Mientras se encuentra en el estado T_1, los circuitos de control verifican las entradas E y F para generar las señales apropiadas en acuerdo. Las operaciones siguientes ocurren en forma simultánea durante la siguiente transición positiva del pulso de reloj:

1. El registro A se incrementa.
2. Si $E = 1$, el registro R se despeja.
3. Dependiendo de los valores de E y F, el control se transfiere al estado siguiente, T_2 o T_3 o T_4.

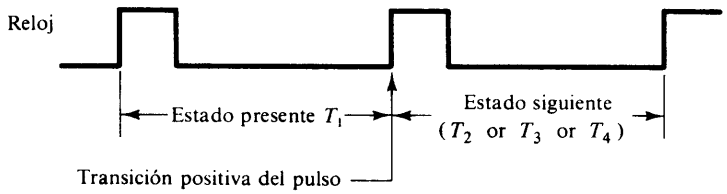


Figura 8-8 Transición entre estados.

Obsérvese que las operaciones en la subsección del procesador de datos y el cambio de estado en el control lógico ocurren al mismo tiempo.

Ahora se demostrará la relación de tiempo entre los componentes de un diagrama ASM al pasar a un ejemplo específico de diseño. El ejemplo no tiene ninguna aplicación conocida y sólo se formula para mostrar la utilidad del diagrama ASM. Se principia desde las especificaciones iniciales y se procede con el desarrollo de un diagrama ASM apropiado mediante el cual puede derivarse el hardware digital.

Ejemplo de diseño

Se desea diseñar un sistema digital con dos flip-flops E y F y un contador A binario de 4-bit. Los flip-flops individuales en A se denotan por A_4, A_3, A_2 y A_1, con A_4 que mantiene el bit más significativo de la cuenta. Una señal de inicio S pone en operación el sistema despejando el contador A y el flip-flop E. Entonces el contador se aumenta en 1 principiando desde el siguiente pulso de reloj y continúa incrementando hasta que la operación se detiene. Los contadores de bits A_3 y A_4 determinan la secuencia de operaciones:

Si $A_3 = 0$, E se despeja a 0 y continúa el conteo.

Si $A_3 = 1$, E se ajusta en 1; entonces si $A_4 = 0$, el conteo continúa, pero si $A_4 = 1$, F se ajusta en 1 en el siguiente pulso de reloj y el sistema detiene el conteo.

Diagrama ASM

El diagrama ASM se muestra en la Fig. 8-9. Cuando no se realizan operaciones, el sistema está en el estado inicial T_0, esperando la señal de inicio S. Cuando la entrada S es igual a 1, el contador A y el flip-flop F se despejan a 0 y el controlador pasa al estado T_1. Obsérvese la casilla condicional que sigue a la casilla de decisión para S. Esto significa que el contador y el flip-flop se despejarán durante T_0 si $S = 1$; y al mismo tiempo, el control se transfiere al estado T_1. El bloque asociado con el estado T_1 tiene dos casillas de decisión y dos casillas condicionales. El contador se incrementa con cada pulso de reloj. Al mismo tiempo, una de tres operaciones posibles ocurre durante la transición del mismo pulso de reloj:

Ya sea que E se despeje y el control permanezca en el estado T_1 ($A_3 = 0$);

o E se ajusta y el control permanece en el estado T_1($A_3A_4 = 10$);

o E se ajusta y el control pasa al estado T_2($A_3A_4 = 11$).

Cuando el control está en el estado T_2, el flip-flop F se ajusta en 1 y el circuito retorna a su estado inicial, T_0.

Los diagramas ASM constan de tres estados y tres bloques. El bloque asociado con T_0 consta de la casilla de estado, una casilla de decisión y una casilla condicional. El bloque asociado con T_2 consta sólo de la casilla de estado. El control lógico tiene una salida externa, S y dos entradas de estado, A_3 y A_4.

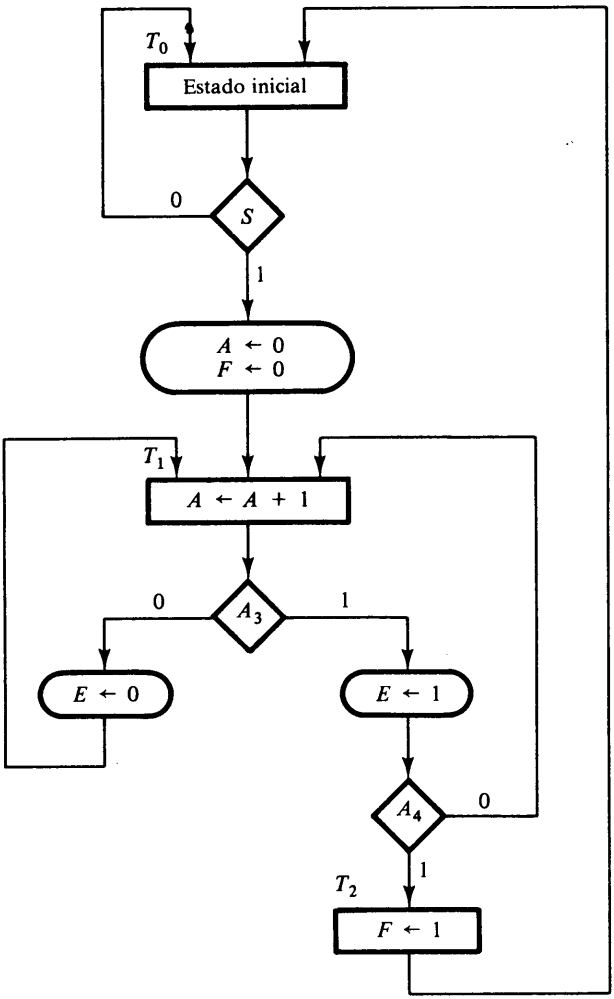


Figura 8-9 Carta ASM para ejemplo de diseño.

Secuencia de temporizado

Cada bloque en un diagrama ASM especifica las operaciones que van a realizarse durante un pulso común de reloj. Las operaciones especificadas dentro de las casillas de estado y condicional en el bloque se llevan a cabo en la subsección del procesador de datos. El cambio de un estado al siguiente se lleva a cabo en el control lógico. Con objeto de apreciar la relación de tiempo implicada, se lista la secuencia de paso por paso de las operaciones después de cada pulso de reloj desde el tiempo en que ocurre la señal de inicio hasta que el sistema regresa a su estado inicial.

En la Tabla 8-2 se muestran los valores binarios del contador y los dos flip-flops después de cada pulso de reloj. En la tabla también se muestran por separado los

estados de A_3 y A_4, al igual que el estado presente en el controlador. Se principia con el estado T_1 precisamente después de que la señal de entrada S ha causado que el contador y el flip-flops F se despejen. El valor de E se supone que es 1, porque E es igual a 1 al T_0 (como se muestra al final de la tabla) y porque E no cambia durante la transición desde T_0 hasta T_1. El sistema permanece en el estado T_1 durante los siguientes trece pulsos de reloj. Cada pulso incrementa el contador y despeja o bien ajusta E. Obsérvese la relación entre el tiempo al cual E_3 llega a ser un 1 y el tiempo al cual E se ajusta en 1. Cuando $A = 0011$, el siguiente pulso de reloj incrementa el contador a 0100, pero el mismo pulso de reloj ve el valor de A_3 como 0, de modo que E se despeja. El siguiente pulso cambia el contador desde 0100 a 0101 y, ahora, A_3 es inicialmente igual a 1, de modo que E se ajusta en 1. En forma similar, E se despeja a 0 no cuando la cuenta pasa desde 0011 a 1000, sino cuando pasa desde 1000 a 1001, lo cual es cuando A_3 es 0 en el valor presente del contador.

Cuando la cuenta alcanza 1100, tanto A_3 como A_4 son iguales a 1. El siguiente pulso de reloj incrementa A en 1, ajusta E en 1, y transfiere el control al estado T_2. El control permanece en T_2 sólo por un periodo del reloj. La transición de pulso asociada con T_2 ajusta el flip-flop F en 1 y transfiere el control al estado T_0. El sistema permanece en el estado inicial T_0 en tanto que S sea igual a 0.

Por la observación de la Tabla 8-2 puede parecer que las operaciones realizadas en E se retardan por un pulso de reloj. Esta es la diferencia entre un diagrama ASM y un diagrama de flujo convencional. Si la Fig. 8-9 fuera un diagrama de flujo convencional, podría suponerse que A se incrementa primero y que el valor incrementado podría haberse utilizado para verificar el estado de A_3. Las operaciones que se realizan

TABLA 8-2 Secuencia de las operaciones en el diseño de ejemplo

Contador				Flip-flops			
A_4	A_3	A_2	A_1	E	F	Condiciones	Estado
0	0	0	0	1	0	$A_3 = 0, A_4 = 0$	T_1
0	0	0	1	0	0		
0	0	1	0	0	0		
0	0	1	1	0	0		
0	1	0	0	0	0	$A_3 = 1, A_4 = 0$	
0	1	0	1	1	0		
0	1	1	0	1	0		
0	1	1	1	1	0		
1	0	0	0	1	0	$A_3 = 0, A_4 = 1$	
1	0	0	1	0	0		
1	0	1	0	0	0		
1	0	1	1	0	0		
1	1	0	0	0	0	$A_3 = 1, A_4 = 1$	
1	1	0	1	1	0		T_2
1	1	0	1	1	1		T_0

en el hardware digital como se especifica por un bloque en el diagrama ASM ocurren durante el mismo periodo de reloj y no en una secuencia de operaciones que siguen una a otra en el tiempo, como se interpreta por lo común en un diagrama de flujo convencional. Por lo tanto, el valor de A_3 que se considera en la casilla de decisión se toma del valor del contador en el estado presente y antes de que se incremente. Esto es porque la casilla de decisión para E pertenece al mismo bloque como en el estado T_1. Los circuitos digitales en el control generan las señales para todas las operaciones especificadas en el bloque presente antes de la llegada del siguiente pulso de reloj. La transición siguiente del reloj ejecuta todas las operaciones en los registros y flip-flops, incluyendo los flip-flops en el controlador que determina el estado siguiente.

Procesador de datos

El diagrama ASM da toda la información necesaria para diseñar el sistema digital. Los requisitos para el diseño del subsistema procesador de datos se especifican dentro de las casillas de estado y condicionales. El control lógico se determina mediante las casillas de decisión y las transiciones de estado requeridas. Un diagrama que muestra el hardware para el ejemplo de diseño se muestra en la Fig. 8-10. El subsistema de control se ilustra sólo con sus entradas y salidas. El diseño detallado de control se considera en la siguiente sección. El procesador de datos consta de un contador binario de 4-bit, dos flip-flops y un número de compuertas. El contador es similar al que se muestra en la Fig. 7-17 excepto que se requieren compuertas adicionales para la operación síncrona de despeje. El contador se incrementa con cada pulso de reloj cuando el control está en el estado T_1. Se despeja sólo cuando el control se encuentra en el estado T_0 y S es igual a 1. Esta operación condicional requiere una compuerta AND para garantizar que ambas condiciones estén presentes. Las otras dos operaciones condicionales usan otras dos compuertas AND para ajustar despejar el flip-flop E. El flip-flop F se ajusta incondicionalmente durante el estado T_2. Obsérvese que todos los flip-flops y registros, incluyendo los flip-flops en el control, utilizan una fuente común de pulso de reloj.

Este ejemplo demuestra un método de diseño digital empleando el diagrama ASM. El diseño del subsistema procesador de datos requiere una interpretación de las operaciones de registros y su implementación mediante los componentes que se expusieron en los Capítulos 5 y 7, como por ejemplo registros, contadores, multiplexores y adicionadores.El diseño del subsistema de control requiere la aplicación de procedimientos de diseño basados en la teoría de la lógica secuencial. Las siguientes tres secciones presentan algunas de las alternativas que están disponibles para diseñar el control lógico.

8-4 IMPLANTACION DEL CONTROL

La sección de control de un sistema digital es en esencia un circuito secuencial que puede diseñarse por el procedimiento delineado en el Capítulo 6. Sin embargo, en la mayor parte de los casos este método no es práctico debido al gran número de estados

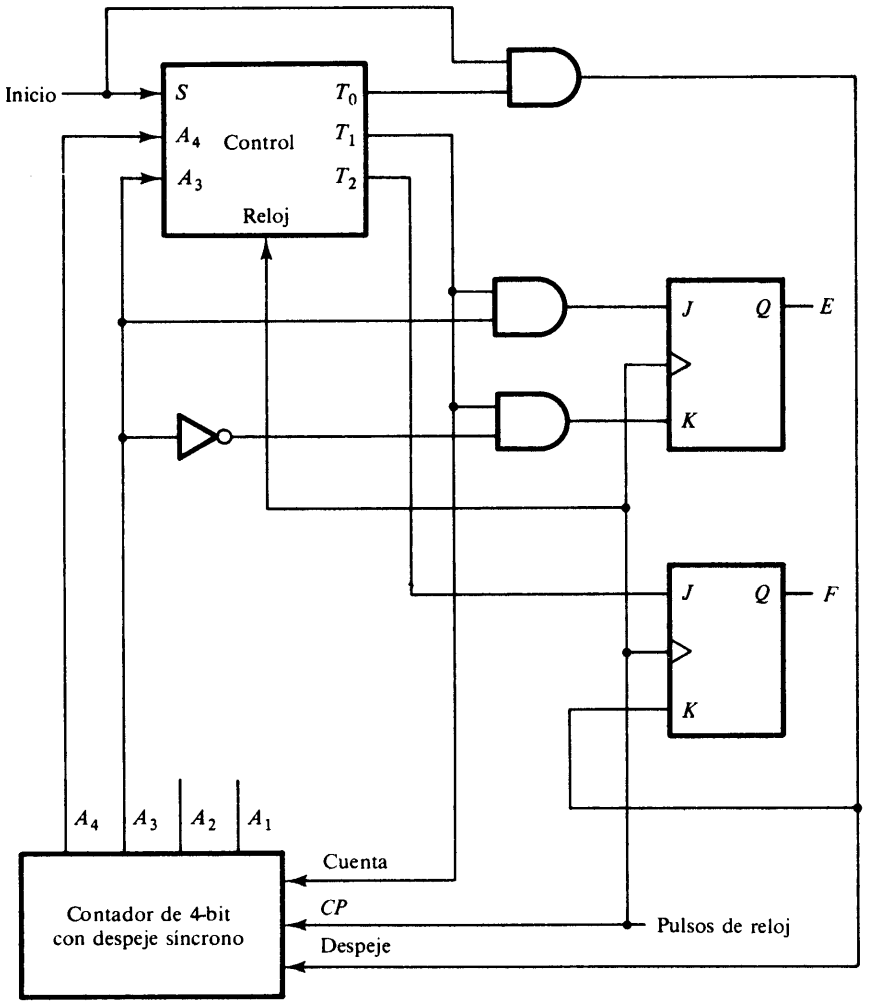


Figura 8-10 Procesador de datos para el ejemplo de diseño.

y salidas que puede tener un circuito de control típico. Excepto por controladores muy simples, el método de diseño que usa tablas de estado y exitación es engorroso y difícil de manejar. Los diseñadores experimentados de sistemas digitales usan métodos especializados para el diseño del control lógico que pueden considerarse como una extensión del método secuencial clásico combinado con otras suposiciones de simplificación. Dos de estos métodos especializados se presentan en esta sección y un tercer método se explica en la Sección 8-5. Otra alternativa es utilizar una ROM o un PLA para diseñar el control lógico. Esto se cubre en la Sección 8-6.

Tabla de estado

Como se mencionó con anterioridad, el diagrama ASM se asemeja a un diagrama de estado con cada casilla de estado representando un estado. El diagrama de estado puede convertirse en una tabla de estado mediante la cual puede diseñarse el circuito secuencial del controlador. Primero deben asignarse valores binarios a cada estado en el diagrama ASM. Para n flip-flops en el circuito secuencial de control, el diagrama ASM puede acomodar hasta 2^n estados. Un diagrama con tres o cuatro estados requiere un circuito secuencial con dos flip-flops. Con cinco a ocho estados, se necesitan tres flip-flops. Cada combinación de valores flip-flop representa un número binario para uno de los estados.

Una tabla de estados para un controlador es una lista de los estados y entrada spresentes y sus correspondientes estados siguientes y salidas. En la mayoría de los casos hay muchas condiciones no importa de entrada que deben incluirse, de modo que es aconsejable arreglar la tabla de estado para tomar esto en consideración. Con objeto de aclarar el procedimiento, se ilustrará por la obtención de la tabla de estado del controlador definido en el ejemplo de la sección anterior.

El diagrama ASM del diseño de ejemplo se muestra en la Fig. 8-9. Se asignan los siguientes valores binarios a los tres estados: $T_0 = 00$, $T_1 = 01$, $T_2 = 11$. El estado binario 10 no se utiliza y se tratará como una condición no importa. La tabla de estado correspondiente al diagrama ASM se muestra en la Tabla 8-3. Son necesarios dos flip-flops, y se etiquetan G_1 y G_2. Hay tres entradas y tres salidas. Las entradas se toman mediante las condiciones en las casillas de decisión. Las salidas son equivalentes al estado presente de control. Obsérvese que hay un renglón en la tabla para cada transición posible entre estados. El estado inicial 00 pasa al estado 01 o permanece en 00 dependiendo del valor de la entrada S. Las otras dos entradas se marcan con X sin preocupación, ya que no determinan en este caso el estado siguiente. En tanto el sistema está en el estado binario 00, el control proporciona una salida etiquetada T_0 para iniciar las operaciones requeridas de registro. La transición del estado binario 01 depende de las entradas A_3 y A_4. El sistema pasa al estado binario 11 sólo si $A_3A_4 = 11$; en otra forma, permanece en el estado binario 01. Al final, el estado binario 11 pasa a 00 en forma independiente de las variables de entrada.

TABLA 8-3 Tabla de estados para control en la Fig. 8-10

Símbolo del estado presente	Estado presente		Entradas			Estado siguiente		Salidas		
	G_1	G_2	S	A_3	A_4	G_1	G_2	T_0	T_1	T_2
T_0	0	0	0	X	X	0	0	1	0	0
T_0	0	0	1	X	X	0	1	1	0	0
T_1	0	1	X	0	X	0	1	0	1	0
T_1	0	1	X	1	0	0	1	0	1	0
T_1	0	1	X	1	1	1	1	0	1	0
T_2	1	1	X	X	X	0	0	0	0	1

Este ejemplo demuestra una tabla de estado para un controlador secuencial. Obsérvese otra vez el gran número de condiciones no importa bajo las entradas. El número de renglones en la tabla de estado es igual al número de trayectorias distintas entre los estados en el diagrama ASM.

Diagrama lógico con flip-flops *JK*

El procedimiento para diseñar un circuito secuencial principiando desde una tabla de estado se presenta en la Sección 6-7. Este procedimiento requiere que se obtenga la tabla de excitación de las entradas flip-flop y entonces se simplifica la parte del circuito combinacional del circuito secuencial. Si se aplica este procedimiento a la Tabla 8-3, se necesita utilizar mapas de cinco variables (véase la Fig. 3-11) para simplificar las funciones de entrada. Esto es porque hay cinco variables listadas bajo las columnas de "estado presente" y "entrada". Ya que este procedimiento se explicó en el Capítulo 6, no se mostrará aquí el trabajo de detalle. Las funciones de entrada flip-flop obtenidas por este método, si se suponen flip-flops *JK*, son:

$$JG_1 = G_2A_3A_4 \qquad JG_2 = S$$
$$KG_1 = 1 \qquad KG_2 = G_1$$

Para derivar las tres funciones de salida, se utiliza el hecho de que el estado binario 10 no se usa y se obtienen las siguientes funciones simplificadas:

$$T_0 = G_2'$$
$$T_1 = G_1'G_2$$
$$T_2 = G_1$$

El diagrama lógico del control se muestra en la Fig. 8-11. Este circuito reemplaza el bloque control en la Fig. 8-10.

Flip-flops *D* y decodificador

Cuando el número de flip-flops más entradas en una tabla de estado es mayor de cinco, es necesario usar grandes mapas para simplificar las funciones de entrada. Esto es engorroso y difícil de lograr, como se explicó en el Capítulo 3. En consecuencia es necesario encontrar vías alternas para diseñar controladores excepto cuando son muy simples. Una posibilidad es usar flip-flips tipo *D* y obtener las funciones de entrada directamente mediante la tabla de estado sin que sea necesaria una tabla de excitación. Esto se debe a que el estado siguiente es el mismo que los requisitos de entrada para los flip-flops *D* (véase la Sección 6-9). Para diseñar el circuito secuencial con flip-flops *D*, es necesario pasar a la siguiente columna de estado en la tabla de estado y derivar todas las condiciones que deben establecer en 1 a cada flip-flop. Mediante la Tabla 8-3 se observa que la siguiente columna de estado de G_1 tiene un solo 1 en el quinto renglón. La entrada T de flip-flop G_1 debe ser igual a 1 durante el estado presente $T_1 = G_1'G_2$

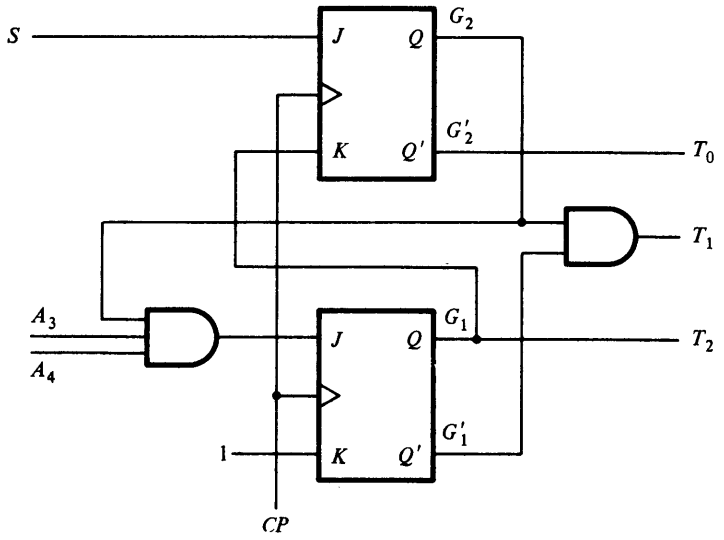


Figura 8-11 Diagrama lógico de control con uso de flip-flops *JK*.

cuando tanto la entrada A_3 como la A_4 son iguales a 1. Esto se expresa con la función de entrada del flip-flop *D*:

$$DG_1 = G'_1G_2A_3A_4$$

En forma similar, la siguiente columna de estado de G_2 tiene cuatro número 1 y la condición para ajustar este flip-flop es:

$$DG_2 = G'_1G'_2S + G'_1G_2$$

Puede avanzarse un paso más e insertar un decodificador a la salida de los flip-flops para obtener las tres salidas necesarias, T_0, T_1 y T_2. Entonces, en lugar de usar las salidas de flip-flop como la condición de estado presente, también pueden usarse las salidas del decodificador para suministrar esta información. Las funciones de entrada a los flip-flops *D* pueden expresarse ahora como sigue:

$$DG_1 = A_3A_4T_1$$
$$DG_2 = ST_0 + T_1$$

El diagrama lógico alterno se muestra en la Fig. 8-12. El decodificador proporciona las tres salidas de control, y esas salidas también se usan para determinar el estado siguiente de cada flip-flop. El segundo circuito de control requiere más componentes que el primero, pero tiene la ventaja de que puede derivarse por inspección mediante la tabla de estado.

Un flip-flop por estado

Otro método posible de diseño del control lógico es usar un flip-flop por estado en el circuito secuencial. Sólo un flip-flop se establece en cualquier momento particular; todos los otros se despejan a 0. El único bit se hace para propagarse de un flip-flop al otro bajo el control de lógica de decisión. En dicho arreglo, cada flip-flop representa un estado que es activado sólo cuando el bit de control se transfiere a él.

Es obvio que en este método no se usa un número mínimo de flip-flops para circuito secuencial. De hecho, se utiliza un número máximo de flip-flops. Por ejemplo, un circuito secuencial con 12 estados requiere un mínimo de cuatro flip-flops. Sin embargo, por este método el circuito necesita 12 flip-flops, uno para cada estado.

Una organización de control que utiliza un flip-flop por estado tiene la característica conveniente de que el circuito puede derivarse de manera directa mediante el diagrama de estado sin necesidad de tablas de estado por excitación. Considérese, por ejemplo, el diagrama de estado de la Fig. 8-13. Este diagrama es equivalente al diagrama ASM del ejemplo de diseño en la Fig. 8-9 en lo que respecta a las transiciones de control de estado. Ya que el diagrama tiene tres estados, se asignan tres flip-flops al circuito y se etiquetan T_0, T_1 y T_2. El controlador puede diseñarse por inspección del diagrama de estado si se usan flip-flops tipo D. La función booleana para ajustar el flip-flop se determina mediante el estado presente y las condiciones de entrada a lo largo de las líneas dirigidas. Por ejemplo, el flip-flop T_0 se ajusta con el siguiente pulso de reloj si el estado presente $T_2 = 1$ o si el estado presente $T_0 = 1$ y la entrada $S = 0$. Esta condición se define por la función de entrada flip-flop:

$$DT_0 = T_2 + S'T_0$$

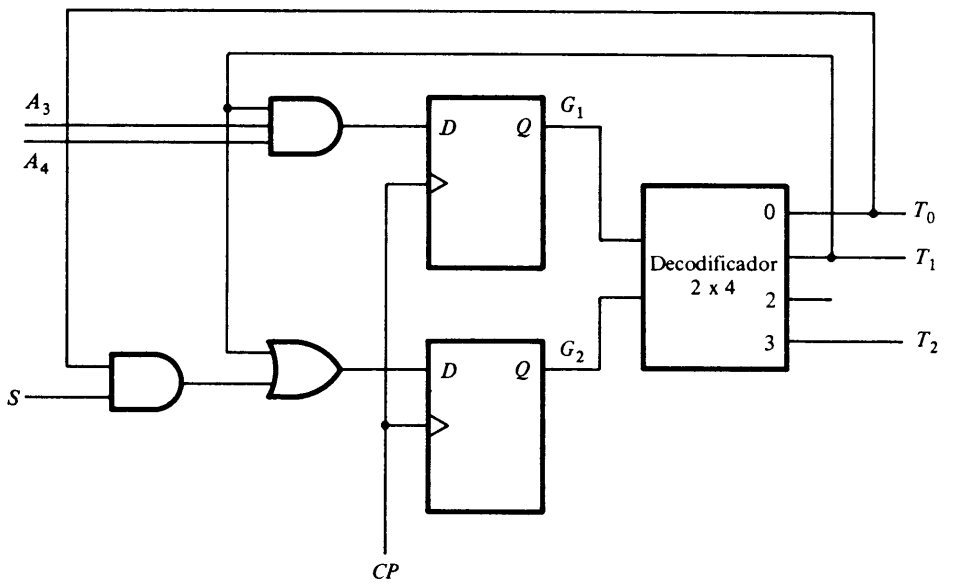


Figura 8-12 Diagrama lógico alterno de control con uso de flip-flops D y un decodificador.

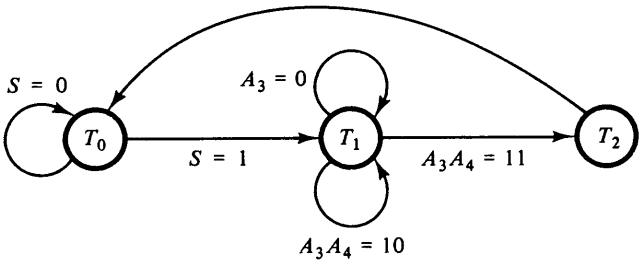


Figura 8-13 Diagrama de estado de un controlador.

en donde DT_0 denota la entrada D del flip-flop T_0. De hecho, la condición para ajustar un flip-flop en 1 se obtiene en forma directa del diagrama de estado por la condición especificada en las líneas dirigidas, que van al estado correspondiente flip-flop combinado en lógica AND con el estado previo flip-flop. Si hay más de una línea dirigida que va a un estado, todas las condiciones deben combinarse en lógica OR. Por el uso de este procedimiento para los otros dos flip-flops, se obtienen las funciones de entrada:

$$DT_1 = ST_0 + A_3'T_1 + A_3A_4'T_1 = ST_0 + (A_3A_4)'T_1$$
$$DT_2 = A_3A_4T_1$$

El diagrama lógico se muestra en la Fig. 8-14. Consta de tres flip-flops tipo D: T_0, T_1 y T_2, y las compuertas asociadas especificadas por las funciones de entrada listadas arriba.

Al inicio, el flip-flop T_0 debe ajustarse en 1 y todos los otros flip-flops se despejan a 0 de modo que el flip-flop que representa el estado inicial es igual a 1 y todos los otros estados son iguales a 0. Una vez iniciado, el flip-flop controlador uno por estado se propagará por sí mismo de estado a estado en la forma apropiada. Para un registro con una entrada común asíncrona de despeje, como se muestra en la Fig. 8-14, todos los flip-flops incluyendo la salida Q de T_0 se despejan a 0. Tomando la salida de T_0 de la salida complemento Q' proporciona la señal inicial requerida de 1 para T_0. Con objeto de mantener Q' como la salida de T_0, es necesario que la función de entrada de la entrada D se complemente. Esto se hace por el inversor adicional que se coloca a la entrada D del flip-flop T_0.

8-5 DISEÑO CON MULTIPLEXORES

Un importante objetivo del diseño de control lógico es el desarrollo de un circuito que implemente la secuencia de control deseada en una forma lógica y directa. El intento de minimizar el número de compuertas tiende a producir un circuito irregular, lo que hace difícil para cualquiera, excepto para el diseñador, identificar la secuencia de eventos que emprende el control. Como consecuencia, es difícil alterar, dar servicio o mantener el equipo después del diseño inicial. La secuencia de estados en el control debe ser evidente en forma clara de la configuración del circuito aun si esto requiere componentes adicionales y resulta en un circuito con número no mínimo de componentes. Una implantación tal es el método de diseño con multiplexores.

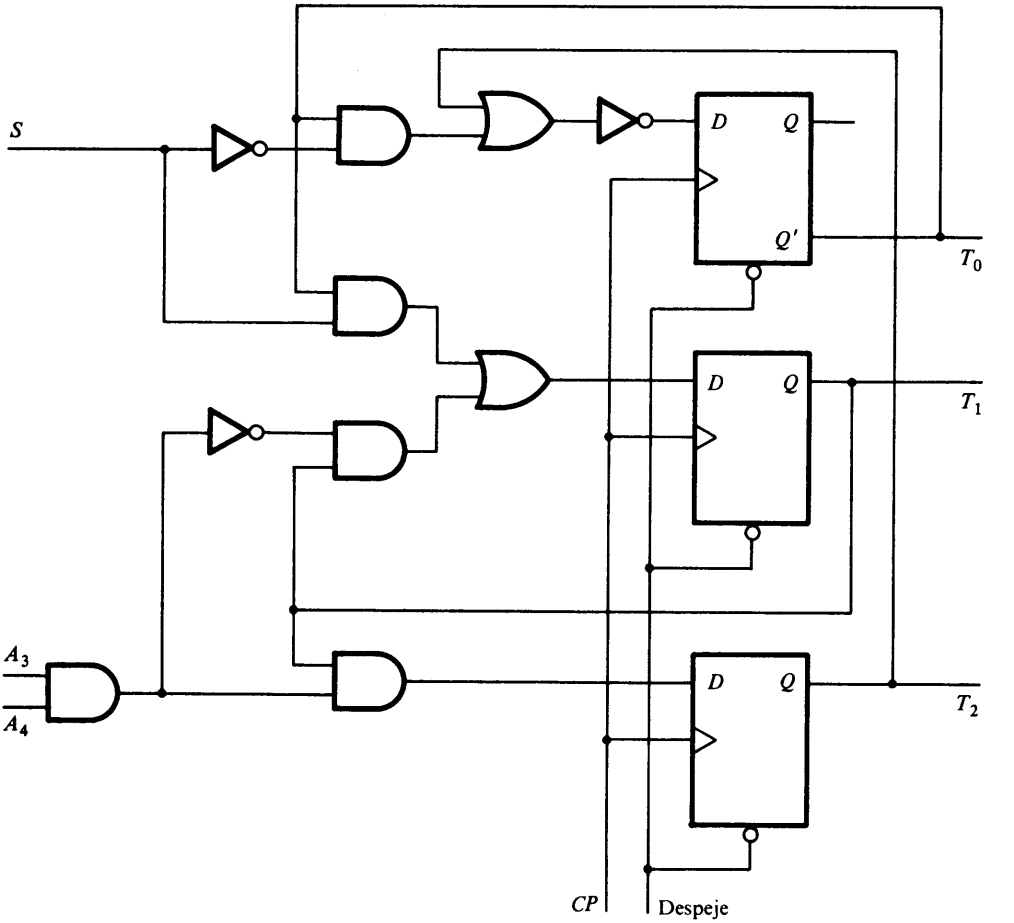


Figura 8-14 Diagrama de una tercera alternativa de control con uso de un flip-flop por estado.

El circuito de control que se muestra en la Fig. 8-12 consta de tres componentes: los flip-flops que mantienen el valor binario de estado, el decoficador que genera las salidas de control, y las compuertas que determinan el estado siguiente. Ahora se reemplazan las compuertas con multiplexores y se usa un registro para los flip-flops individuales. Este método de diseño da por resultado un patrón regular de tres niveles de componentes. El primer nivel consta de multiplexores que determinan el estado siguiente del registro. El segundo nivel contiene un registro que mantiene el estado binario presente. El tercer nivel tiene el decodificador que proporciona una salida separada para cada estado de control.

Por ejemplo, considérese el diagrama ASM en la Fig. 8-15. Consta de cuatro estados y cuatro entradas de control. Las casillas de estado se dejan vacías en este caso porque se tiene interés sólo en la secuencia de control, la cual es independiente de las operaciones del registro. La asignación binaria para cada estado se indica en la

esquina superior derecha de las casillas de estado. Las casillas de decisión especifican las transiciones de estado como una función de las cuatro entradas de control w, x, y y z. La implementación del control de tres niveles se muestra en la Fig. 8-16. Consta de dos multiplexores MUX1 y MUX2, un registro con dos flip-flops G_1 y G_2 y un decodificador con cuatro salidas. Las salidas del registro se aplican a las entradas del decodificador y también a las entradas seleccionadas de los multiplexores. En esta forma, el estado presente del registro se usa para seleccionar una de las entradas para cada multiplexor. Las salidas de los multiplexores se aplican entonces a las entradas D de G_1 y G_2. El propósito de cada multiplexor es producir una entrada a su flip-flop correspondiente igual al valor binario del estado siguiente.

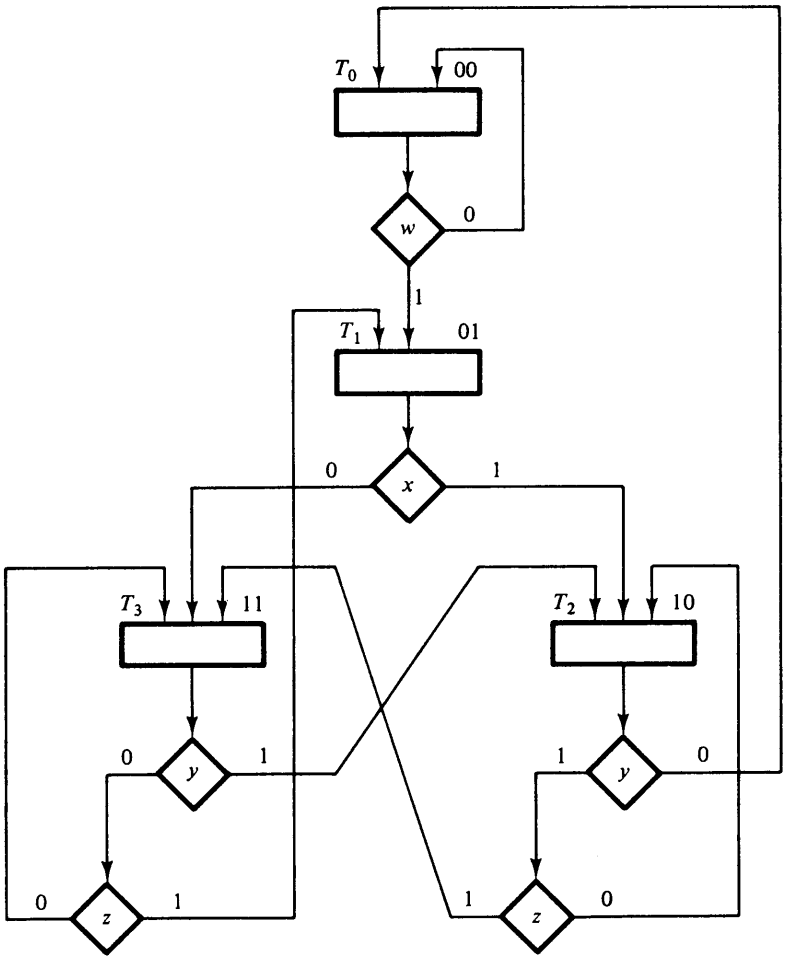


Figura 8-15 Ejemplo de una carta ASM con cuatro entradas de control.

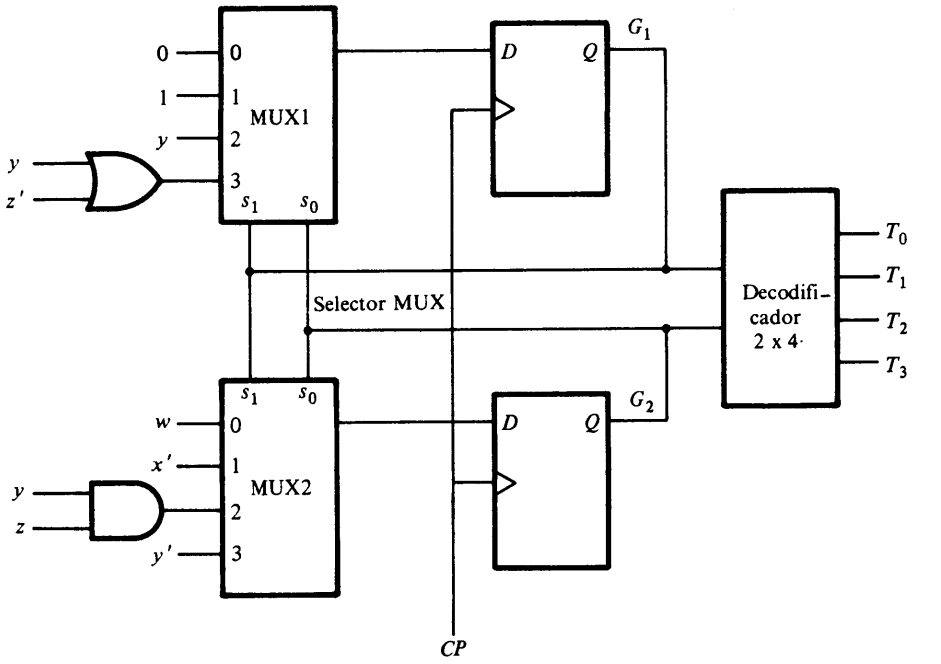


Figura 8-16 Implementación de un control con multiplexores.

Las entradas de los multiplexores se determinan mediante las casillas de decisión y las transiciones de estado dadas en el diagrama ASM. Por ejemplo, el estado 00 permanece en 00 o pasa a 01 dependiendo del valor de la entrada w. Ya que el estado siguiente de G_1 es 0 en cualquier caso, se coloca una señal equivalente a la lógica 0 en la entrada 0 del MUX1. El estado siguiente de G_2 es 0 si $w = 0$ y 1 si $w = 1$. Ya que el estado siguiente de G_2 es igual a w, se aplica el control de entrada w a la entrada 0 del MUX2. Esto significa que cuando las entradas seleccionadas de los multiplexores son iguales al estado presente 00, las salidas de los multiplexores proporcionan el valor binario que se transfiere al registro durante el siguiente pulso de reloj.

Para facilitar la evaluación de las entradas de multiplexor, se prepara una tabla mostrando las condiciones de entrada para cada transición posible en el diagrama ASM. En la Tabla 8-4 se da esta información para el diagrama ASM en la Fig. 8-15. Hay dos transiciones desde el estado presente 00 o 01 y tres transiciones desde el estado presente 10 o 11. Estas transiciones están separadas por líneas horizontales a través de la tabla. Las condiciones de entrada que se listan en la tabla se obtienen mediante las casillas de decisión en el diagrama ASM. Por ejemplo, en la Fig. 8-15 se observa que el estado presente 01 pasará al estado siguiente 10 si $x = 1$, o al estado siguiente 11 si $x = 0$. En la tabla se marcan estas condiciones de entrada como x y x', respectivamente. Las dos columnas bajo "entradas multiplexoras" en la tabla especifican los valores de entrada que deben aplicarse a MUX1 y MUX2. La entrada de multiplexor para cada estado presente se determina mediante las condiciones de entrada cuando el estado

siguiente del flip-flop es igual a 1. Por tanto, después del estado presente 01, el estado siguiente de G_1 siempre es igual a 1 y el estado siguiente de G_2 es igual al valor complemento de x. Así que, la entrada de MUX1 se hace igual a 1 y la de MUX2 a x' cuando el estado presente del registro es 01. Como otro ejemplo, después del estado presente 10, el estado siguiente de G_1 debe ser igual a 1 si las condiciones de entrada son yz' o yz. Cuando estos dos términos booleanos se combinan juntos en lógica OR y se simplifican entonces, se obtiene la sola variable binaria y, como se indica en la tabla. El estado siguiente de G_2 es igual a 1 si las condiciones de entrada son $yz = 11$. Si el estado siguiente de G_1 permanece en 0 después de un estado presente dado, se coloca un 0 en la entrada del multiplexor como se muestra en el estado presente 00 para el MUX1. Si el estado siguiente de G_1 siempre es 1, se coloca un 1 en la entrada del multiplexor como se muestra en el estado presente 01 para el MUX1. Las otras anotaciones para MUX1 y MUX2 se derivan de manera semejante. Las entradas a los multiplexores mediante la tabla se usan entonces en la implementación del control en la Fig. 8-16. Obsérvese que si el estado siguiente de un flip-flop es una función de dos o más variables de control, el multiplexor puede requerir una o más compuertas en su entrada. De otra forma, la entrada del multiplexor es igual a la variable de control, o el complemento de la variable de control, o 0 o 1.

Ejemplo de diseño

Se demostrará la implementación de un multiplexor de control mediante un segundo ejemplo de diseño. En el ejemplo también se demostrará la formulación del diagrama ASM y la implementación del subsistema procesador de datos.

El sistema digital que se diseñará consta de dos registros $R1$ y $R2$ y un flip-flop E. El sistema cuenta el número de los 1 en el número cargado dentro del registro $R1$ y

TABLA 8-4 Condiciones de entrada al multiplexor

Estado presente		Estado siguiente		Condiciones de entrada	Entradas al multiplexor	
G_1	G_2	G_1	G_2		MUX1	MUX2
0	0	0	0	w'		
0	0	0	1	w	0	w
0	1	1	0	x		
0	1	1	1	x'	1	x'
1	0	0	0	y'		
1	0	1	0	yz'		
1	0	1	1	yz	$yz' + yz = y$	yz
1	1	0	1	$y'z$		
1	1	1	0	y		
1	1	1	1	$y'z'$	$y + y'z' = y + z'$	$y'z + y'z' = y'$

establece el registro $R2$ en ese número. Por ejemplo, si el número binario que se carga dentro de $R1$ es 10111001, el circuito cuenta los cinco 1 en $R1$ y establece el registro $R2$ en la cuenta binaria 101. Esto se hace corriendo cada bit desde el registro $R1$ uno a la vez dentro del flip-flop E. El valor en E se verifica por el control, y cada vez que es igual a 1, el registro $R2$ se incrementa en 1.

El subsistema de control usa una entrada S externa para iniciar la operación y dos entradas de estados E y Z desde el procesador de datos. E es la salida del flip-flop. Z es la salida de un circuito que verifica el contenido del registro $R1$ para todos los 0. El circuito produce una salida $Z = 1$ cuando $R1$ es igual a 0.

El diagrama ASM para el ejemplo de diseño se muestra en la Fig. 8-17. El número binario se carga dentro de $R1$ y el registro $R2$ se ajusta en un valor por

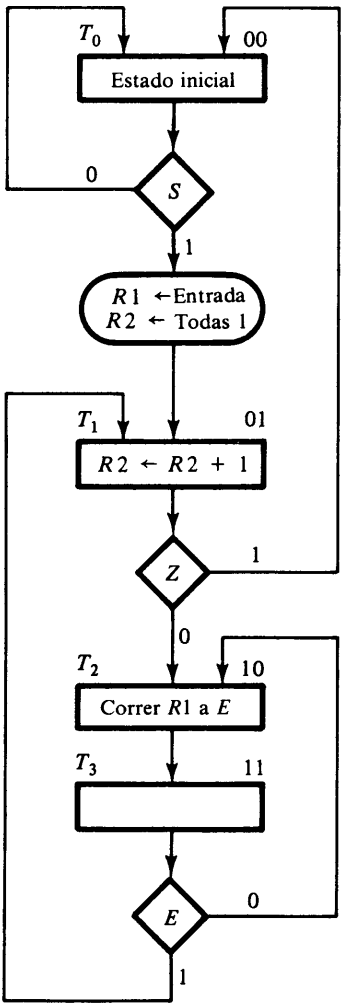


Figura 8-17 Carta ASM para el ejemplo de diseño.

completo en número 1. Obsérvese que un número con todos 1 en un registro cuando se incrementa produce un número por completo en 0. En el estado T_1, el registro $R2$ se incrementa si el contenido de $R1$ se examina. Si el contenido es 0, entonces $Z = 1$, y esto significa que no hay 1 almacenados en el registro; de modo que la operación termina con $R2$ igual a 0. Si el contenido de $R1$ no es cero, entonces $Z = 0$ y esto indica que hay algunos 1 almacenados en el registro. El número en $R1$ se corre y su bit de la extrema izquierda se transfiere dentro de E. Esto se hace cuantas veces sea necesario hasta que se transfiere un 1 dentro de E. Por cada 1 detectado en E, el registro $R2$ se incrementa y el registro $R1$ se verifica otra vez para buscar más números 1. El lazo mayor se repite hasta que todos los 1 en $R1$ están contados. Obsérvese que la casilla de estado de T_3 no tiene operaciones de registro, pero el bloque asociado con ella contiene la casilla de decisión para E. También obsérvese que la entrada serial al registro de corrimiento $R1$ debe ser igual a 0 porque no se desea correr 1 externos dentro de $R1$.

El subsistema procesador de datos se muestra en la Fig. 8-18. El control tiene tres entradas y cuatro salidas. Sólo se usan tres salidas por el procesador de datos. El registro $R1$ es un registro de corrimiento similar al que se ilustra en la Fig. 7-9. El registro $R2$ es un contador con carga paralela parecido al que se ilustra en la Fig. 7-19. Con objeto de no complicar el diagrama, no se muestran los pulsos de reloj, pero deben aplicarse a los dos registros, al flip-flop E y a los flip-flops en el control. El circuito que verifica para 0 es una compuerta NOR. Por ejemplo, si $R1$ es un registro de cuatro bits con salidas R_1, R_2, R_3, R_4, entonces Z se genera con la función booleana

$$Z = R_1' R_2' R_3' R_4' = (R_1 + R_2 + R_3 + R_4)'$$

la cual comprende las funciones NOR de todos los bits en el registro.

Las condiciones de entrada del multiplexor para el control se determinan mediante la Tabla 8-5. Las condiciones de entrada se obtienen mediante el diagrama ASM para cada posible transición de estado binario. La asignación binaria a cada estado está escrita en la esquina superior derecha de las casillas de estado. La transición del estado presente 00 depende de S, del estado presente 01 depende de Z y del estado presente de E. El estado presente 10 pasa al estado siguiente 11 de manera incondicional. Los valores bajo MUX1 y MUX2 en la tabla se determinan mediante las condiciones booleanas de entrada para el estado siguiente de G_1 y G_2, respectivamente.

La implementación de control del ejemplo de diseño se muestra en la Fig. 8-19. Esta es una implementación de tres niveles con los multiplexores en el primer nivel. Las entradas a los multiplexores se obtienen mediante la Tabla 8-5.

8-6 CONTROL PLA

Mediante los ejemplos que se presentaron en este capítulo se ha visto que el diseño de un circuito de control es en forma esencial un problema de lógica secuencial. En la Sección 7-2 se mostró que un circuito secuencial puede construirse mediante un registro conectado a un circuito combinacional. En la Sección 5-8 se investigó el arreglo lógico programable (PLA) y se mostró que puede utilizarse para implementar

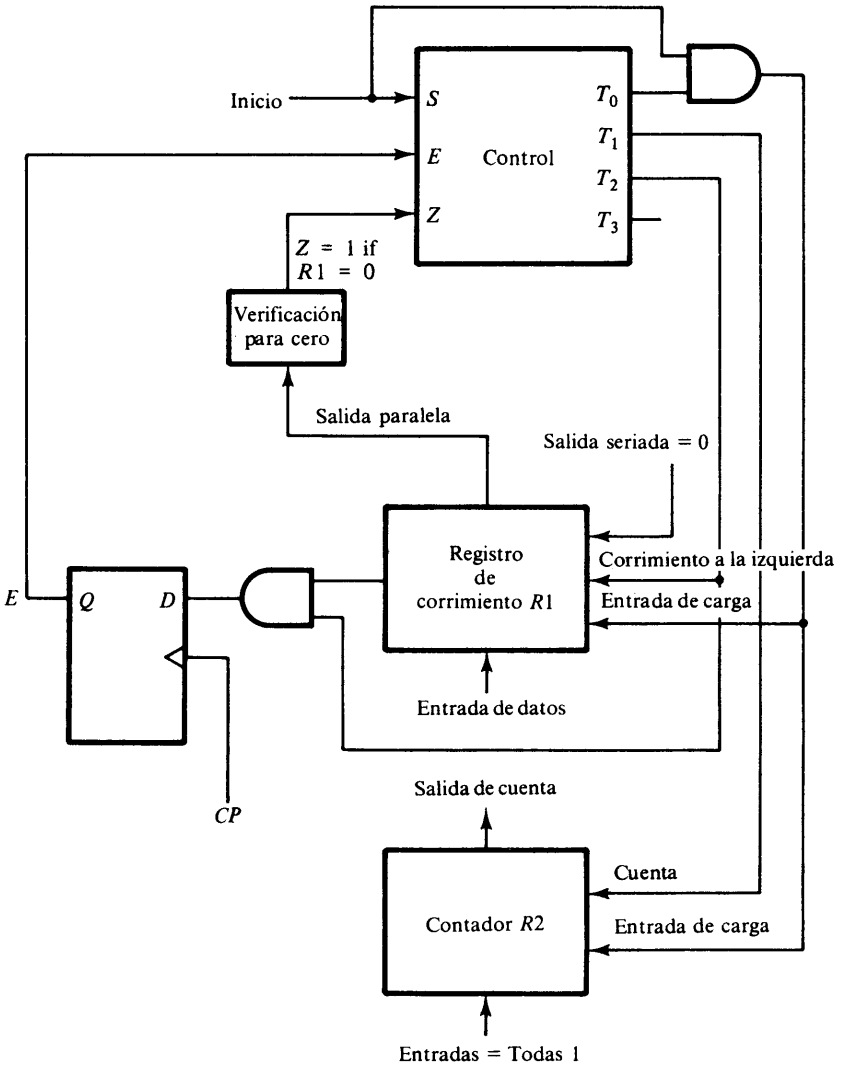


Figura 8-18 Subsistema procesador de datos para el ejemplo de diseño.

cualquier circuito combinacional. Ya que el control lógico es un circuito secuencial, entonces es posible diseñar el circuito de control con un registro conectado a un PLA. El diseño de un control PLA requiere que se obtenga la tabla de estado del circuito. El método PLA debe usarse si la tabla de estado contiene muchas anotaciones no importa en otra forma, puede ser ventajoso utilizar una ROM en lugar de un PLA.

El control PLA se desmotrará mediante un tercer ejemplo de diseño. Este ejemplo es un circuito que modifica dos números binarios sin signo y produce su producto binario.

TABLA 8-5 Condiciones de entrada al multiplexor del diseño de ejemplo

Estado presente		Estado siguiente		Condiciones de entrada	Entradas al multiplexor	
G_1	G_2	G_1	G_2		MUX1	MUX2
0	0	0	0	S'		
0	0	0	1	S	0	S
0	1	0	0	Z		
0	1	1	0	Z'	Z'	0
1	0	1	1	Ninguna	1	1
1	1	1	0	E'		
1	1	0	1	E	E'	E

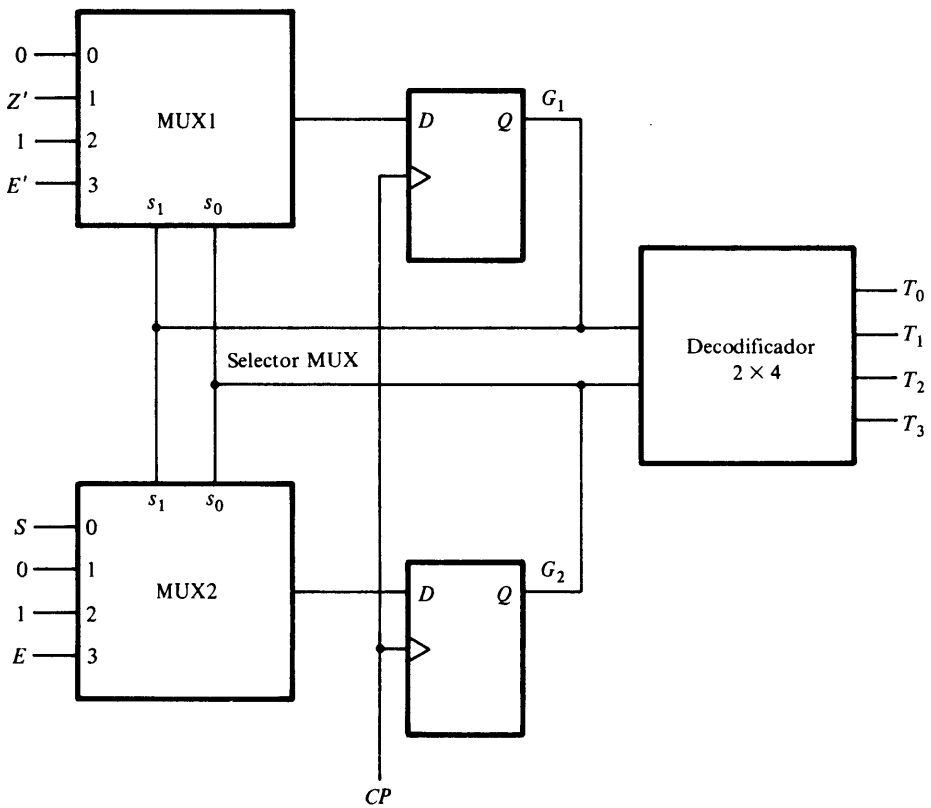


Figura 8-19 Implementación de control para ejemplo de diseño.

Multiplicador binario

La multiplicación de dos números binarios se hace con papel y lápiz por adiciones sucesivas y corrimientos. Este proceso se ilustra de mejor manera con un ejemplo numérico. Permítase multiplicar los dos números binario 10111 y 10011.

23	10111	multiplicando
19	10011	multiplicador
	10111	
	10111	
	00000	
	00000	
	10111	
437	110110101	producto

El proceso consiste de observar los bits sucesivos del multiplicador, primero el bit menos significativo. Si el bit de multiplicador es un 1, el multiplicando se copia abajo. En otra forma, los 0 se copian abajo. Los números que se copian abajo en líneas sucesivas se corren una posición a la izquierda del número previo. Por último, los números se adicionan y sus sumas forman el producto. Obsérvese que el producto obtenido mediante la multiplicación de dos números binarios de n bits cada uno puede ser hasta 2^n de largo.

Cuando se implementa el proceso anterior con hardware digital, es conveniente cambiar ligeramente el proceso. Primero, en lugar de proporcionar circuitos digitales para almacenar y sumar en forma simultánea tantos números binarios como haya números 1 en el multiplicador, es conveniente proporcionar circuitos para la suma sólo de dos números binarios y acumular en dorma sucesiva los productos parciales en un registro. Segundo, en lugar de correr el multiplicando a la izquierda, el producto parcial se corre a la derecha, lo cual resulta en dejar el producto parcial y el multiplicando en las posiciones relativas requeridas. Tercero, cuando el bit correspondiente del multiplicador es un 0, no hay necesidad de sumar todos los 0 al producto parcial, ya que esto no altera su valor.

El subsistema procesador de datos para el multiplicador binario se muestra en la Fig. 8-20. El multiplicando se almacena en el registro B. El multiplicador se almacena en el registro Q, y el producto parcial se forma en el registro A. Un adicionador paralelo similar al circuito mostrado en la Fig. 5-1 se utiliza para añadir los contenidos del registro B al registro A. El flip-flop A almacena la cuenta que se lleva después de la adición. El contador P inicialmente se establece para mantener un número binario igual al número de bits en el multiplicador. Este contador se decrementa después de la formación de cada producto parcial. Cuando el contenido del contador alcanza cero, el producto se forma en el registro doble A y Q y se detiene el proceso.

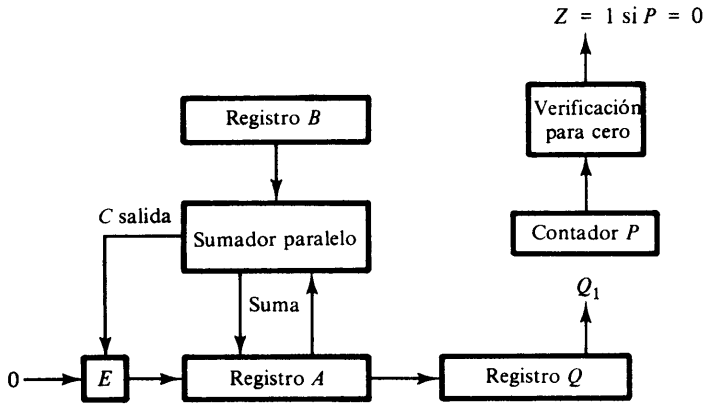


Figura 8-20 Procesador de datos para multiplicador binario.

El control lógico permanece en un estado inicial hasta que la señal de inicio S llega a ser un 1. El sistema realiza entonces la multiplicación. La suma de A y B forma un producto parcial, el cual se transfiere a A. La salida del acarreo mediante la adición ya sea 0 o 1 se transfiere a E. Tanto el producto parcial en A como el multiplicador en Q se corren a la derecha. El bit menos significativo de A se corre a la posición más significativa de Q; el acarreo de E se corre a la posición más significativa de A; y 0 se corre dentro de E. Después de la operación de corrimiento a la derecha, un bit del producto parcial se transfiere dentro de Q en tanto los bits del multiplicador en Q se corren una posición a la derecha. En esta forma, el bit de la extrema derecha del registro Q, designado por Q_1 siempre mantiene el bit del multiplicador que debe inspeccionarse a continuación.

Diagrama ASM

El diagrama ASM para el multiplicador binario se muestra en la Fig. 8-21. Al inicio, el multiplicando está en B y el multiplicador en Q. El proceso de multiplicación se inicia cuando $S = 1$. El registro A y el flip-flop E se despejan y el contador de secuencia P se ajusta en un número binario n, el cual es igual al número de bits en el multiplicador.

A continuación se entra en un lazo que mantiene formando los productos parciales. El bit multiplicador en Q_1 se verifica, y si es igual a 1, el multiplicando en B se agrega al producto parcial en A. La cuenta que se lleva de la adición se transfiere a E. El producto parcial en A se deja sin cambio si $Q_1 = 0$. El contador P se decrementa en 1 sin importar el valor de Q_1. Los registros E, A y Q se combinan en un registro compuesto EAQ, el cual se corre entonces un lugar a la derecha para obtener un nuevo producto parcial.

El valor en el contador P se verifica después de la formación de cada producto parcial. Si el contenido de P no es cero, la entrada de control Z es igual a 0 y el proceso se repite para formar un nuevo producto parcial. El proceso se detiene cuando el contador P llega a ser 0 y la entrada de control Z es igual a 1. Obsérvese que el producto parcial formado en A se corre dentro de Q un bit a la vez y, a la larga, repone el

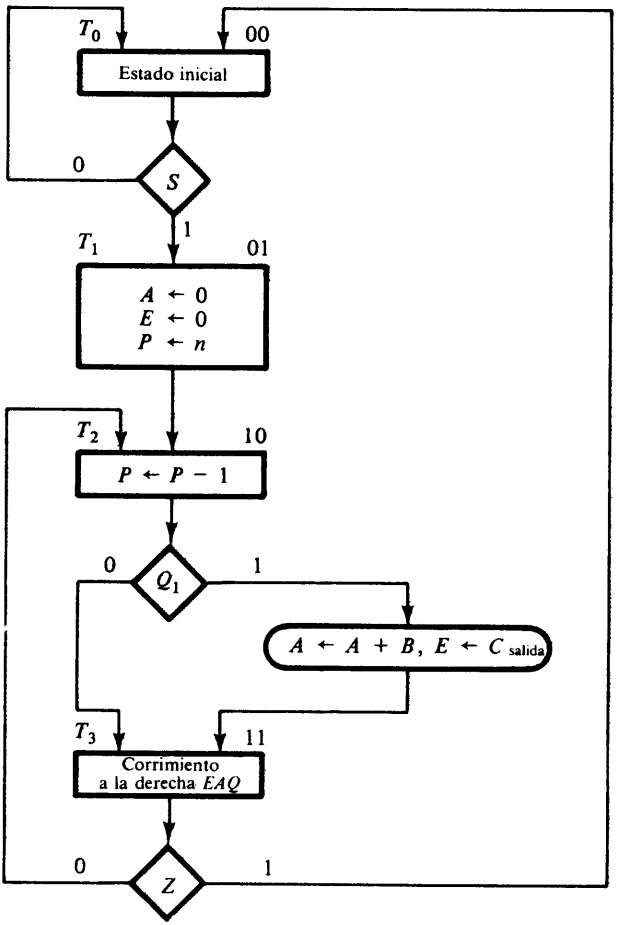


Figura 8-21 Carta ASM para multiplicador binario.

multiplicador. El producto final está disponible en A y Q, con A manteniendo los bits más significativos y Q los bits menos significativos.

El ejemplo numérico anterior se repite en la Fig. 8-22 para aclarar el proceso de multiplicación. El procedimiento sigue los pasos delineados en el diagrama ASM.

Control PLA

El control para el multiplicador binario tiene cuatro estados y tres entradas. La asignación del estado binario se muestra en el diagrama ASM sobre cada casilla de estado. Las tres entradas de control son S, Q_1 y Z. El diseño de una unidad de control con un PLA es similar al diseño que usa flip-flops D y un decodificador. La única diferencia estriba en la forma en que el circuito combinacional parte del control de implanta. El PLA en forma esencial reemplaza al decodificador y todas las compuertas en las entradas de los flip-flops.

Multiplicando $B = 10111$

	E	A	Q	P
Multiplicador en Q	0	00000	10011	101
$Q_1 = 1$; añadir B		10111		
Primer producto parcial	0	10111		100
correr a la derecha EAQ	0	01011	11001	
$Q_1 =$: añadir B		10111		
Segundo producto parcial	1	00010		011
Correr a la derecha EAQ	0	10001	01100	
$Q_1 = 0$; correr a la derecha EAQ	0	01000	10110	010
$Q_1 = 0$; correr a la derecha EAQ	0	00100	01011	001
$Q_1 = 1$; añadir B		10111		
Quinto producto parcial	0	11011		000
Correr a la derecha EAQ	0	01101	10101	
Producto final en $AQ = 0110110101$				

Figura 8-22 Ejemplo de multiplicación binaria.

El diagrama de bloques del control PLA se muestra en la Fig. 8-23. El PLA se conecta a un registro con dos flip-flops, G_1 y G_2. Las entradas al PLA son los valores del estado presente en el registro y las tres entradas de control. Las salidas del PLA proporcionan los valores para el estado siguiente en el registro y las variables de salida del control. Hay una salida para cada estado presente y una salida adicional para la operación condicional $D = Q_1T_2$. Ya que el PLA implanta el circuito combinacional de control, bien pueden incluirse dentro de él las compuertas para todas las operaciones condicionales. En el multiplicador binario hay una operación condicional para añadir B a A durante el estado T_2 siempre que $Q_1 = 1$. La salida D del PLA activará esta operación en el procesador de datos

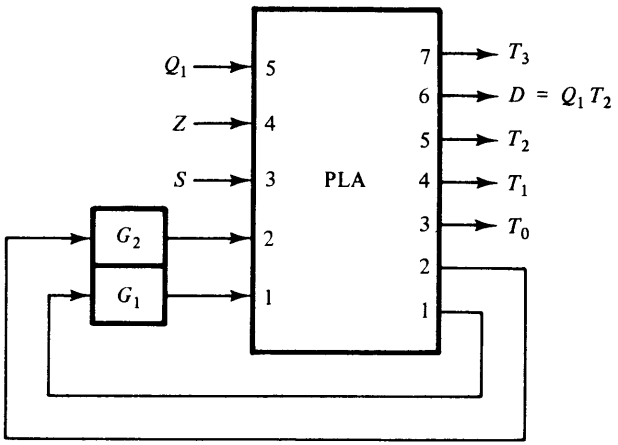


Figura 8-23 Diagrama de bloques del control PLA.

En cualquier momento dado, el estado presente del registro junto con las condiciones de entrada determinan los valores de salida y el estado siguiente para el registro. El siguiente pulso de reloj inicia las operaciones en el registro especificadas por las salidas PLA y transfiere el siguiente valor de estado dentro del registro. Esto proporciona un nuevo estado de control y posibles valores diferentes de entrada. Por eso, el PLA actúa como la parte del circuito combinacional del circuito secuencial para generar las salidas de control y los valores del estado siguiente para el registro.

Tabla del programa PLA

La organización interna del PLA se presentó en la Sección 5-8. También se mostró cómo obtener la tabla del programa PLA. Se aconseja al lector que repase esta sección para tener la seguridad de que se ha comprendido el significado de una tabla de programa PLA. Las trayectorias internas del PLA están construidas de acuerdo con las especificaciones que se dan en la tabla del programa. El diseño de un control PLA requiere que se obtenga la tabla de estado para el circuito. La tabla de estado da esencialmente toda la información necesaria para obtener la tabla del programa PLA.

La tabla de programa PLA puede obtenerse en forma directa mediante la tabla de estado sin necesidad de procedimientos de simplificación. La tabla del programa PLA que se lista en la Tabla 8-7 especifica siete términos productos, uno para cada puede ser una función de una de las variables de entrada de control o puede ser independiente de cualquier entrada. Si una variable de entrada no influencia el estado siguiente, se marca con una condición no importa, X. Si hay dos transiciones diferentes desde el mismo estado presente, el estado presente se repite en la tabla pero los estados siguientes tienen asignados diferentes valores binarios. En la tabla también se listan todas las salidas de control como una función del estado presente. Obsérvese que la entrada Q_1 no afecta el estado siguiente sino sólo determina el valor de la salida D durante el estado T_2.

La tabla de programa PLA puede obtenerse en forma directa mediante la tabla de estado sin necesidad de procedimientos de simplificación. La tabla del programa PLA que se lista en la Tabla 8-7 especifica siete términos productos, uno para cada

TABLA 8-6 Tabla de estado para el circuito de control

Estado presente		Entradas			Estado siguiente		Salidas				
G_1	G_2	S	Z	Q_1	G_1	G_2	T_0	T_1	T_2	D	T_3
0	0	0	X	X	0	0	1	0	0	0	0
0	0	1	X	X	0	1	1	0	0	0	0
0	1	X	X	X	1	0	0	1	0	0	0
1	0	X	X	0	1	1	0	0	1	0	0
1	0	X	X	1	1	1	0	0	1	1	0
1	1	X	0	X	1	0	0	0	0	0	1
1	1	X	1	X	0	0	0	0	0	0	1

TABLA 8-7 Tabla del programa para el PLA

Término producto	Entradas 1	2	3	4	5	Salidas 1	2	3	4	5	6	7	Comentarios
1	0	0	0	—	—	—	—	1	—	—	—	—	$T_0 = 1$, $S = 0$
2	0	0	1	—	—	—	1	1	—	—	—	—	$T_0 = 1$, $S = 1$
3	0	1	—	—	—	1	—	—	1	—	—	—	$T_1 = 1$
4	1	0	—	—	0	1	1	—	—	1	—	—	$T_2 = 1$, $Q_1 = 0$
5	1	0	—	—	1	1	1	—	—	1	1	—	$T_2 = 1$, $D = 1$,
6	1	1	—	0	—	1	—	—	—	—	—	1	$T_3 = 1$, $Z = 0$
7	1	1	—	1	—	—	—	—	—	—	—	1	$T_3 = 1$, $Z = 1$

renglón en la tabla de estado. Las terminales de entrada y salida están marcadas con número, y las variables aplicadas a esas terminales numeradas se indican en el diagrama de bloques en la Fig. 8-23. Los comentarios no son parte de la tabla, pero se incluyen para mayor claridad.

De acuerdo con las reglas establecidas en la Sección 5-8, una trayectoria sin conexión para un PLA se indica por un guión (-) en la tabla. Las *X* en la tabla de estado denotan condiciones no importa que implican que no hay conexión para el PLA. Los 0 en las columnas de salida también indican que no hay conexiones en las compuertas OR dentro del PLA. La traducción de la tabla de estado a una tabla de programa PLA es muy simple. Las *X* en las columnas de "entrada" y los 0 en las columnas de "estado siguiente" y "salidas" se cambian a guiones, y todas las demás anotaciones permanecen igual. Las entradas al PLA son las mismas que en el estado presente y las entradas en la tabla de estado. Las salidas del PLA son iguales que en el estado siguiente y las salidas en la tabla de estado.

En el ejemplo anterior se demuestra el procedimiento para diseñar el control lógico con un PLA. Mediante las especificaciones del sistema, se obtiene primero una tabla de estado para el controlador. El número de estados determina el número de flip-flops para el registro. Entonces se conecta el PLA al registro y a las variables de entrada y salida. La tabla de programa PLA se obtiene de manera directa mediante la tabla de estado.

Los ejemplos que se presentaron en este capítulo demuestran cinco métodos de diseño de control lógico. No deben considerarse estos métodos como los únicos posibles. Un diseñador con recursos puede ser capaz de formular una configuración de control para adecuarla a una aplicación particular. Esta configuración puede constar de una configuración de métodos o puede constituir una organización de control diferente de las que se presentan aquí.

BIBLIOGRAFIA

1. Clare, C. R., *Designing Logic Systems Using State Machines.* New York: McGraw-Hill Book Co., 1973.

2. Winkel, D., y F. Prosser, *The Art of Digital Design.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1980.

3. Peatman, J. B., *Digital Hardware Design.* New York: McGraw-Hill Book Co., 1980

4. Wiatrowski, C. A., C. H. House, *Logic Circuits and Microcomputer Systems.* New York: McGraw-Hill Book Co., 1980.

5. Fletcher, W. I., *An Engineering Approach to Digital Design.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1980.

6. Rhyne, V. T., *Fundamentals of Digital Systems Design.* Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1973.

PROBLEMAS

8-1. Dibuje la porción de un diagrama ASM que especifica una operación condicional para incrementar el registro R durante el estado T_1 y transferir el estado T_2 si las entradas de control z y y son iguales a 1 y 0, respectivamente.

8-2. Muestre ocho trayectorias de salida en un bloque ASM que emanan desde las casillas de decisión que verifican los ocho valores binarios posibles de tres variables de control x, y y z.

8-3. Obtenga el diagrama ASM para las siguientes transiciones de estado:
(a) Si $x = 0$, el control pasa desde el estado T_1 al estado T_2; si $x = 1$, genere una operación condicional y pase desde T_1 a T_2.
(b) Si $x = 1$, el control pasa desde T_1 a T_2 y entonces a T_3; si $x = 0$, el control pasa desde T_1 a T_3.
(c) Principie desde el estado T_1; entonces: si $xy = 00$, pasa a T_2; si $xy = 01$, pasa a T_3; si $xy = 10$, pasa a T_1; en otra forma, pasa a T_3.

8-4. Construya un diagrama ASM para un sistema digital que cuenta el número de personas en un cuarto. Las personas entran a la habitación por una puerta con una fotocelda que cambia una señal x desde 1 a 0 cuando se interrumpe la luz. Salen del cuarto por una segunda puerta con una fotocelda similar con una señal y. Tanto x como y están sincronizadas con el reloj, pero pueden permanecer encendidas o apagadas por más de un periodo de pulso de reloj. El subsistema procesador de datos consta de un contador hacia abajo con un exhibidor de su contenido.

8-5. Explique cómo difiere el diagrama ASM de un diagrama de flujo convencional. Utilice la Fig. 8-5 como una ilustración, muestre la diferencia en interpretación.

8-6. Diseñe el contador de 4-bit con despeje síncrono especificado en la Fig. 8-10.

8-7. Use mapas de cinco variables, derive las funciones booleanas de entrada a los flip-flops JK en la Fig. 8-11.

8-8. Diseñe el control cuya tabla de estado está dada en la Tabla 8-3 utilizando dos multiplexores, un registro y un decodificador.

8-9. Diseñe el control del ejemplo de diseño en la Sección 8-5 por el método de un flip-flop por estado.

8-10. El diagrama de estado de una unidad de control se muestra en la Fig. P8-10. Tiene cuatro estados y dos entradas, x y y.
(a) Dibuje el diagrama ASM equivalente, dejando vacías las casillas de estado.
(b) Diseñe el control con multiplexores.

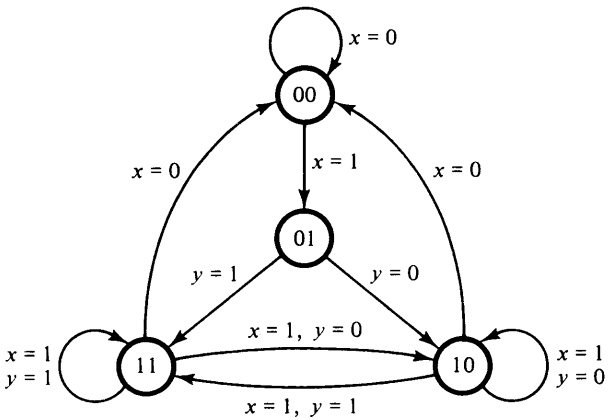


Figura P8-10 Diagrama de estados de control para el problema 8-10.

8-11. Suponga que $R1$ en la Fig. 8-18 es el registro de corrimiento de 4-bit que se muestra en la Fig. 7-9. Muestre cómo las entradas de corrimiento y carga en la Fig. 8-18 están conectadas a las entradas s_1 y s_0 del registro de corrimiento.

8-12. Diseñe un sistema digital con tres registros de 4-bit, A, B y C, para realizar las siguientes operaciones:
1. Transferencia de dos números binarios A y B cuando se habilita una señal de inicio.
2. Si $A < B$, se corren a la izquierda los contenidos de A y se transfiere el resultado al registro B.
3. Si $A > B$, se corren a la derecha los contenidos de B y se transfiere el resultado al registro C.
4. Si $A = B$, se transfiere el número al registro C sin cambio.

8-13. Diseñe un sistema digital que multiplique dos números binarios por el método de adición repetida. Por ejemplo, para multiplicar 5 X 4, el sistema digital evalúa el producto por la adición del multiplicando cuatro veces: $5 + 5 + 5 + 5 = 20$. Permita que el multiplicando esté en el registro BR, el multiplicador en el registro AR y el producto en el registro PR. Un circuito sumador añade los contenidos de BR a PR. Un circuito para detección de cero Z verifica cuando AR llega a ser 0 después de cada vez que se decrementa.

8-14. Demuestre que la multiplicación de dos números de n-bit da un producto de longitud menor que o igual a $2n$ bits.

8-15. En la Fig. 8-20, el registro Q mantiene el multiplicador y el registro B mantiene el multiplicando. Suponga que cada número consta de 15 bits.
(a) ¿Cuántos bits pueden esperarse en el producto y dónde está disponible?
(b) ¿Cuántos bits hay en el contador P y cuál es el número binario cargado dentro de él al inicio?
(c) Diseñe el circuito que verifique para cero en el contador P.

8-16. Liste los contenidos de los registros E, A, Q y P similar a la Fig. 8-22 durante el proceso de multiplicar los dos números 11111 (multiplicando) y 10101 (multiplicador).

8-17. Determine el tiempo que toma procesar la operación de multiplicación en el multiplicador binario que se describe en la Sección 8-6. Suponga que el registro Q tiene n bits y el periodo de reloj es T nanosegundos.

8-18. Diseñe el circuito de control del multiplicador binario especificado por el diagrama ASM en la Fig. 8-21 utilizando cada uno de los siguientes métodos:
(a) Flip-Flops *JK* y compuertas.
(b) Flip-flops *D* y un decodificador.
(c) Multiplexores de entrada y un registro.
(d) Un flip-flop por estado.

8-19. Diseñe el control cuya tabla de estado se muestra en la Tabla 8-3 usando el método PLA.

8-20. Considere el diagrama ASM de la Fig. P8-20. Las operaciones de registro no se especifican, porque sólo se tiene interés en diseñar el control lógico.
(a) Dibuje el diagrama de estado equivalente.
(b) Diseñe el control con un flip-flop por estado.

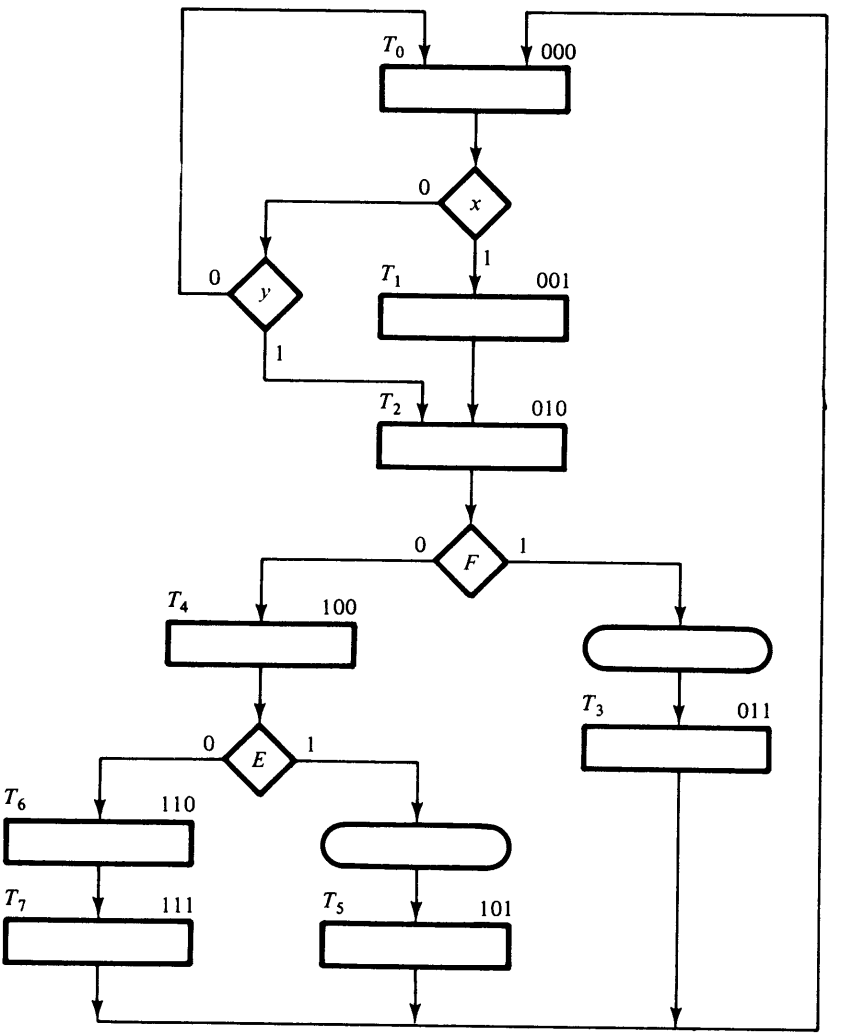


Figura P8-20 Diagrama ASM para los problemas 8-20 al 8-22 inclusive.

8-21. (a) Derive la tabla de estado para el diagrama ASM en la fig. P8-20.
(b) Diseñe el control con tres flip-flops *D*, un decodificador y compuertas.
(c) Diseñe el control con un registro y un PLA. Liste la tabla de programa PLA.

8-22. (a) Derive una tabla mostrando las condiciones de entrada del multiplexor para el control especificado en el diagrama ASM en la Fig. P8-20.
(b) Diseñe el control con tres multiplexores, un registro con tres flip-flops, y un decodificador 3×8.