

SISTEMAS COMBINATORIOS

3-1 INTRODUCCION

Los circuitos lógicos de sistemas digitales pueden ser combinatorios o secuenciales. Un circuito combinatorio consta de compuertas lógicas cuyas salidas en cualquier momento se determinan directamente a partir de los valores de las entradas presentes. Un circuito combinatorio realiza una operación de procesamiento de información determinada que se puede especificar lógicamente por medio de un conjunto de expresiones booleanas. Los circuitos secuenciales emplean elementos de almacenamiento llamados multivibradores biestables (o *flip-flops*) además de las compuertas lógicas. El estado de los elementos de almacenamiento, a su vez, es función de entradas previas. Como consecuencia, las salidas de un circuito secuencial dependen no sólo de los valores presentes de las entradas, sino también de entradas pasadas, y el comportamiento del circuito debe especificarse por una secuencia de tiempo de entradas y estados internos. Los circuitos secuenciales se presentan en el capítulo siguiente.

Un circuito combinatorio consta de variables de entrada, compuertas lógicas o de lógica, y variables de salida. Las compuertas lógicas aceptan señales de las entradas y generan señales en las salidas. Este proceso transforma información binaria de datos de entrada dados en datos de salida requeridos. En la figura 3-1 se presenta un

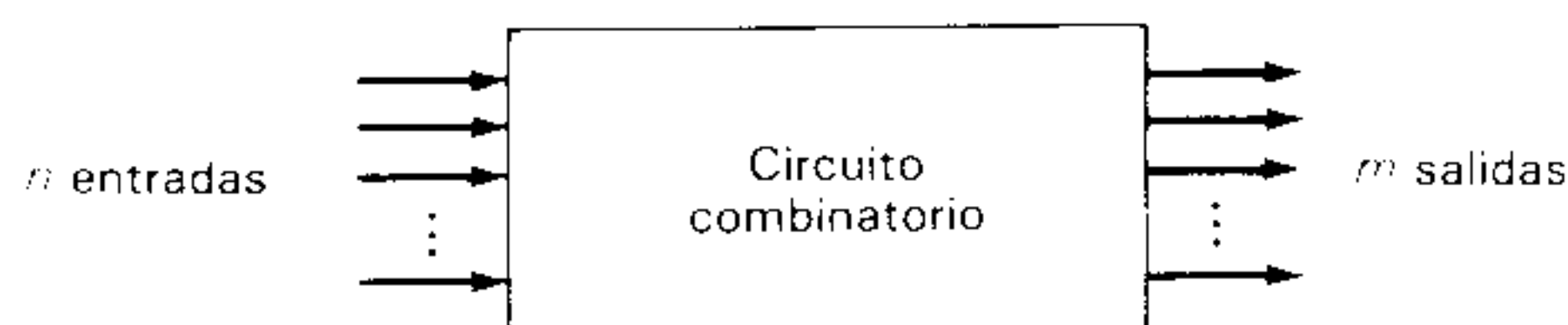


FIGURA 3—1

Diagrama de bloque de un circuito combinatorio

diagrama de bloque de un circuito combinatorio. Las n variables de entrada provienen de una fuente externa y las m variables de salida se dirigen a un destino externo. Cada variable de entrada y salida existe físicamente como una señal binaria que representa el equivalente del 1 o 0 lógicos.

Para n variables de entrada, existen 2^n combinaciones binarias posibles. Para cada combinación binaria de las variables de entrada, existe un valor binario de salida posible. Por lo tanto, un circuito combinatorio se puede especificar a través de una tabla de verdad que presente los valores de salida de cada combinación de las variables de entrada. Un circuito combinatorio además se puede describir mediante m funciones booleanas, una por cada variable de salida. Cada función booleana de salida se expresa como una función de las n variables de entrada.

En el capítulo 1 se aprendió a reconocer números y códigos binarios que representan cantidades de información discretas. En el capítulo 2 se presentaron las diversas compuertas lógicas y se aprendió cómo simplificar funciones booleanas a fin de lograr diseños económicos de compuertas. El objetivo de este capítulo es aplicar los conocimientos adquiridos en los capítulos anteriores y formular diversos procedimientos sistemáticos de análisis y diseño de circuitos combinatorios. Los diversos ejemplos que se presentan en este capítulo ofrecen cierta práctica para manejar tales circuitos.

Existen varios circuitos combinatorios que se emplean frecuentemente en el diseño y la construcción de computadoras digitales. Estas componentes están disponibles en circuitos integrados y se clasifican como circuitos MSI (de integración a mediana escala). Algunos de estos componentes se presentan en este capítulo junto con una explicación de sus propiedades lógicas. Las funciones MSI que se presentan en este capítulo ofrecerán un catálogo de componentes digitales elementales que se utilizan ampliamente como bloques básicos de construcción en el diseño de computadoras y sistemas digitales.

3.2 PROCEDIMIENTO DE ANALISIS

El análisis de un circuito combinatorio consiste en determinar la función que ejecuta el circuito. Se inicia con un diagrama del circuito lógico dado y culmina con un conjunto de funciones booleanas o una tabla de verdad junto con una posible explicación de la operación del circuito. Si el diagrama de lógica que se analizará se acompaña de un nombre de función o bien de un planteamiento de lo que se supone se realizará, entonces el problema de análisis se reduce a una verificación de la función expresada.

El primer paso en el análisis consiste en asegurarse que el circuito dado sea combinatorio y no secuencial. El diagrama de un circuito combinatorio tiene compuertas lógicas (o de lógica) sin elementos de retroalimentación o almacenamiento. Una trayectoria de retroalimentación existe si hay una conexión de la salida de una compuerta a la entrada de una segunda compuerta que forma parte de la entrada de la primera compuerta. Las trayectorias de retroalimentación o elementos de almacenamiento de un circuito digital pueden dar origen a un circuito secuencial y deben analizarse de acuerdo con procedimientos que se describen en el capítulo 4.

Deriva

Cuando se verifica que un diagrama de lógica es de un circuito combinatorio, se puede proceder a obtener las funciones booleanas de salida o la tabla de verdad. Si la función del circuito está sometida a investigación, entonces es necesario interpretar la operación del circuito a partir de las funciones booleanas o la tabla de verdad derivados. El resultado de esta investigación se ve reforzado si se tiene experiencia y conocimiento previos de una amplia variedad de circuitos digitales. La posibilidad de correlacionar una tabla de verdad con una tarea de procesamiento de información es un arte que se adquiere por la experiencia.

Derivación u obtención de funciones booleanas

Para obtener las funciones booleanas de salida a partir de un diagrama de lógica se procede de la manera siguiente:

1. Rotúlense todas las compuertas de salida que sean función de variables de entrada con símbolos arbitrarios. Determinénse las funciones booleanas para cada compuerta.
2. Rotúlense las compuertas que sean función de variables de entrada, y las compuertas antes rotuladas, con diferentes símbolos arbitrarios. Determinénse las funciones booleanas para estas compuertas.
3. Repítase el proceso que se describió en el paso 2 hasta que se obtengan las salidas del circuito en términos de las variables de entrada.

El análisis del circuito combinatorio de la figura 3-2 ilustrará este procedimiento. Observamos que el circuito tiene cuatro variables de entrada binarias, A , B , C y D , y dos variables de salida binarias, F_1 y F_2 . Las salidas de las diversas compuertas se rotulan con símbolos intermedios. Las salidas de las compuertas que son función únicamente de variables de entrada son T_1 y T_2 . Las funciones booleanas que corresponden a estas dos salidas son

$$T_1 = \overline{BC}$$

$$T_2 = \overline{AB}$$

Después se consideran las salidas de las compuertas que son función de símbolos ya definidos.

$$T_3 = A + T_1 = A + \overline{BC}$$

$$T_4 = T_2 \oplus D = (\overline{AB}) \oplus D = \overline{AB}\overline{D} + AD + \overline{B}D$$

$$T_5 = T_2 + D = \overline{AB} + D$$

Las funciones booleanas para las salidas son

$$F_2 = T_5 = \overline{AB} + D$$

$$\begin{aligned} F_1 = T_3 + T_4 &= A + \overline{BC} + \overline{AB}\overline{D} + AD + \overline{B}D \\ &= A + \overline{BC} + \overline{B}\overline{D} + \overline{B}D \end{aligned}$$

La última simplificación de F_1 se puede verificar por manipulación algebraica o por medio de un mapa.

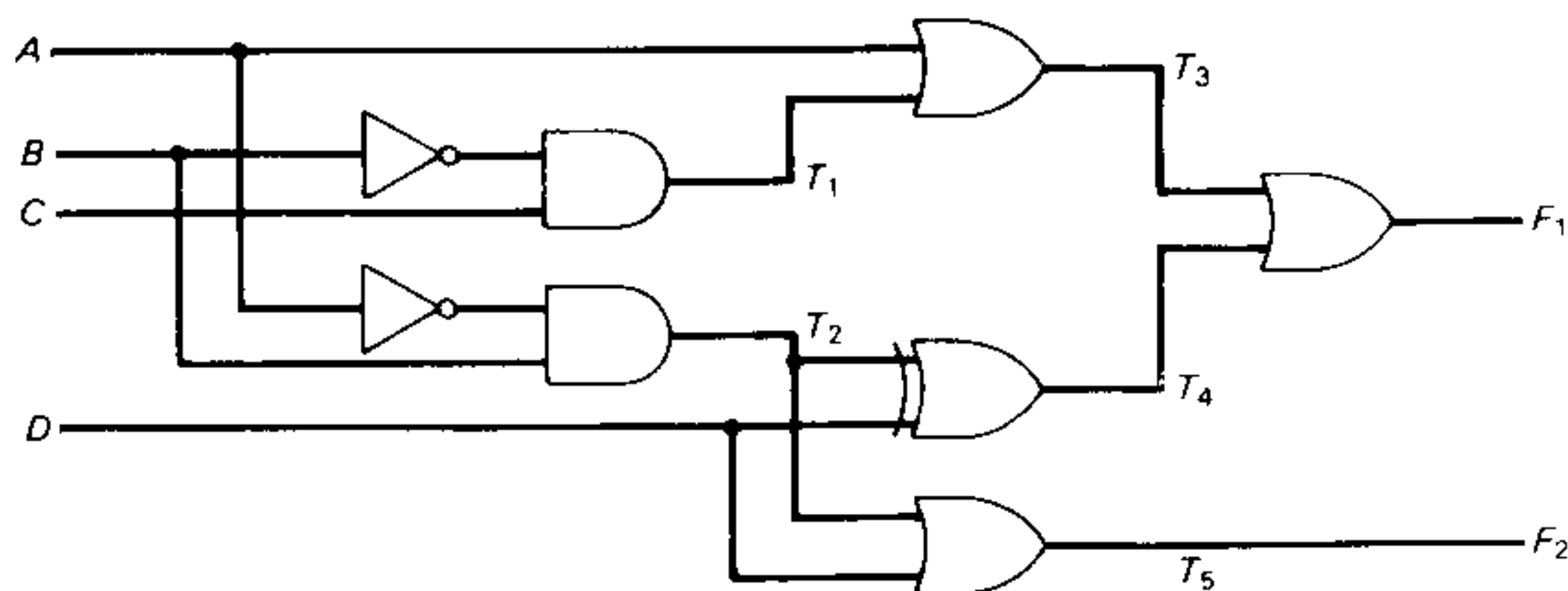


FIGURA 3—2

Diagrama de lógica del ejemplo de análisis

Derivación u obtención de la tabla de verdad

La derivación de la tabla de verdad de un circuito combinatorio es un proceso directo una vez que se conocen las funciones booleanas de salida. Para obtener la tabla de verdad a partir del diagrama de lógica sin pasar por la derivación de las funciones booleanas, procédase en la forma siguiente:

1. Determínese el número de variables de entrada en el circuito. Para n entradas, escríbanse los números binarios del 0 al $2^n - 1$ en una tabla.
2. Rotúlense las salidas de compuertas seleccionadas con símbolos arbitrarios.
3. Obténgase la tabla de verdad para las salidas de las compuertas que únicamente sean función de las variables de entrada.
4. Procédase a obtener la tabla de verdad de las salidas de las compuertas que sean función de valores definidos previamente, hasta que se determinen las columnas de todas las salidas.

El proceso se ilustra por medio del circuito combinatorio de un sumador binario en la figura 3-3. El problema aquí consiste en verificar que el circuito forme la suma aritmética de los tres bits en la entrada X , Y y Z . Las salidas F_1 y F_2 cambian en valor del 00 al 11 binario (3 decimal), según el número de unos (1) en las entradas; por ejemplo, cuando $XYZ = 101$, F_2F_1 debe ser igual al 10 binario para indicar que hay dos unos en las entradas.

La tabla 3-1 muestra el procedimiento de obtención de la tabla de verdad del circuito. Primero, se forman las ocho combinaciones binarias de las tres variables de entrada. La tabla de verdad de F_2 se determina a partir de los valores de las entradas X , Y y Z . F_2 es igual a 1 cuando $XY = 11$, o $XZ = 11$, o bien $YZ = 11$. En caso contrario es igual a 0. La tabla de verdad de $\overline{F_2}$ es el complemento de F_2 . La tabla de verdad de T_1 y T_2 son las funciones AND y OR de las variables de entrada, respectivamente. Los valores binarios de T_3 se determinan comparando (con AND) T_2 y $\overline{F_2}$. Por lo tanto T_3 es igual a 1 cuando T_2 y $\overline{F_2}$ son iguales a 1, y en caso contrario a 0. Por último, F_1 es igual a 1 para combinaciones en las que T_1 o T_3 , o ambas son iguales a 1.

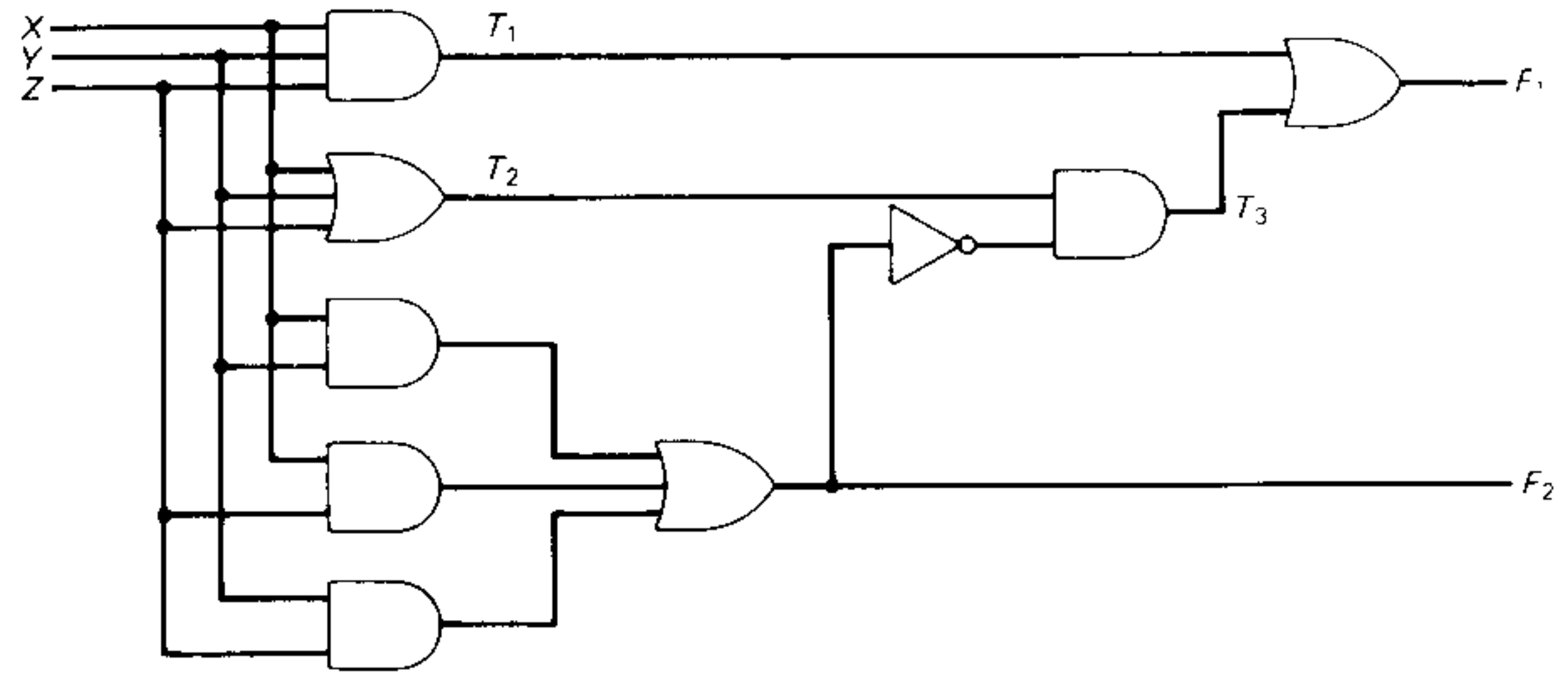


FIGURA 3—3

Diagrama de lógica de un sumador binario

TABLA 3—1

Tabla de verdad del sumador binario

X	Y	Z	F_2	\bar{F}_2	T_1	T_2	T_3	F_1
0	0	0	0	1	0	0	0	0
0	0	1	0	1	0	1	1	1
0	1	0	0	1	0	1	1	1
0	1	1	1	0	0	1	0	0
1	0	0	0	1	0	1	1	1
1	0	1	1	0	0	1	0	0
1	1	0	1	0	0	1	0	0
1	1	1	1	0	1	1	0	1

La inspección de la tabla de verdad revela que $F_2F_1 = 00, 01, 10$ u 11 , cuando el número total de unos en las tres entradas XYZ es cero, uno, dos o tres, respectivamente. Esto verifica la operación del circuito como un sumador binario. El diseño de un sumador binario con una estructura de compuertas diferente se presenta en la sección 3-4.

3.3 PROCEDIMIENTO DE DISEÑO

El diseño de circuitos combinatorios comienza desde la especificación del problema y culmina en un diagrama de circuito lógico o en un conjunto de funciones booleanas del cual se puede obtener el diagrama de lógica. En el procedimiento se aplican los pasos siguientes:

1. De las especificaciones del circuito, determínese el número requerido de entradas y salidas y asígnese un símbolo alfabético (o letra) a cada una.
2. Derívese la tabla de verdad que define la relación requerida entre entradas y salidas.

3. Obténganse las funciones booleanas simplificadas para cada salida como función de las variables de entrada.
4. Trácese el diagrama de lógica.

Una tabla de verdad de un circuito combinatorio consta de columnas de entrada y columnas de salida. Las columnas de entrada se obtienen a partir de los 2^n números binarios de las n variables de entrada. Los valores binarios de las salidas se determinan a partir de las especificaciones citadas. Las funciones de salida especificadas en la tabla de verdad dan la definición exacta del circuito combinatorio. Es importante que se interpreten con corrección las especificaciones verbales de la tabla de verdad. Con mucha frecuencia las especificaciones verbales son incompletas y cualquier interpretación errónea puede degenerar en una tabla de verdad incorrecta.

Las funciones binarias de salida de la tabla de verdad se simplifican por cualquier método disponible como la manipulación algebraica, el método del mapa o con programas de simplificación generados por computadora. Por lo general existen diversas expresiones simplificadas entre las cuales elegir. En una aplicación en particular, ciertos criterios servirán como guía en el proceso de elegir una construcción. Un diseño práctico debe considerar restricciones tales como el número mínimo de compuertas, el número mínimo de entradas a una compuerta, el tiempo mínimo de propagación de la señal a través de las compuertas, el número mínimo de interconexiones, limitaciones de conducción de cada compuerta y otros criterios diversos que deben tomarse en cuenta cuando se diseñe con circuitos integrados. Como la importancia de cada restricción está regida por la aplicación en particular, es difícil hacer un planteamiento general acerca de lo que constituye un diseño aceptable. En la mayoría de los casos la simplificación empieza satisfaciendo un objetivo elemental, como la producción de las funciones booleanas simplificadas en forma estándar, y después cumple con otros criterios de desempeño. Los ejemplos que siguen ilustran el procedimiento delineado.

Ejemplo 3-1

Diséñese un circuito combinatorio con tres entradas y una salida. La salida debe ser un 1 lógico cuando el valor binario de las entradas sea menor que tres y 0 lógico en caso contrario. Utilícense sólo compuertas NAND.

El diseño del circuito se aborda en la figura 3-4. Designamos a las entradas con los símbolos alfabéticos X , Y , Z y a la salida con F . La tabla de verdad se presenta en la parte (a) de la figura. $F = 1$ cuando las entradas binarias son 0, 1, o 2; en caso contrario $F = 0$. La función booleana simplificada de salida se deriva del mapa de la parte (b). El diagrama de lógica de NAND está representado en la parte (c) con notación simbólica mixta (véase la sección 2-6).

Convertidor de código

Cuando un circuito combinatorio tiene dos o más salidas, cada salida debe expresarse en forma independiente como función de todas las variables de entrada. Un ejemplo de un circuito de salidas múltiples es un convertidor de código, esto es, un

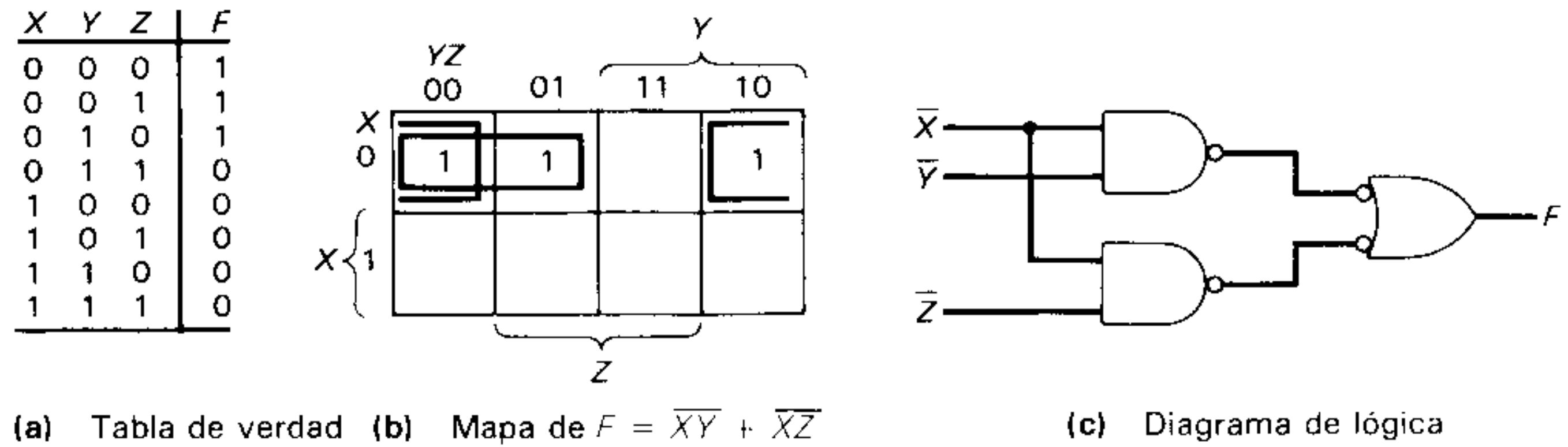


FIGURA 3—4

Solución al ejemplo 3—1

circuito que traduce información de un código binario a otro. Las entradas del circuito proporcionan la combinación de bits de los elementos según especifica el primer código, y las salidas generan la combinación de bits correspondiente al segundo código. El circuito combinatorio realiza la transformación de un código al otro. El diseño de un convertidor de código se ilustrará por medio de un ejemplo que convierte el decimal codificado binario (BCD) en el código de exceso 3 de los dígitos decimales.

Como ambos códigos utilizan cuatro bits para representar cada dígito decimal, debe haber cuatro variables de entrada y cuatro de salida. Designense las entradas por A , B , C y D , y las salidas por W , X , Y , Z . La tabla de verdad que relaciona las variables de entrada y salida se muestra en la tabla 3-2. Las combinaciones de bits de las entradas y sus salidas correspondientes se obtienen directamente de la tabla 1-5. Nótese que cuatro variables binarias pueden tener 16 combinaciones de bits pero en la tabla de verdad sólo se presentan diez. Las seis combinaciones de bits de entrada, de la 1010 a la 1111, no se presentan en las columnas de entrada. Estas combinaciones no tienen significado en el código BCD y podemos suponer que nunca ocurrirán. Por lo tanto, no importa qué valores binarios asignemos a las salidas y podemos tratarlas como condiciones no importa.

TABLA 3—2
Tabla de verdad del ejemplo del convertidor de código

Dígito decimal	Entrada BCD				Salida exceso-3			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

Los mapas de la figura 3-5 se trazan para obtener las funciones booleanas simplificadas de las salidas. Cada uno de los cuatro mapas representa una de las salidas del circuito como función de las cuatro entradas. Los unos del mapa se obtienen directamente de la tabla de verdad pasando por las columnas de salida, una a la vez. Por ejemplo, la columna debajo de la salida W tiene unos para los minterminos 5, 6, 7, 8 y 9. Por lo tanto, el mapa de W debe tener cinco unos en los cuadrados que correspondan a estos minterminos. Los seis minterminos no importa, del 10 al 15, están marcados cada uno con una X en todos los mapas. Las funciones simplificadas en suma de productos se colocan debajo del mapa de cada variable.

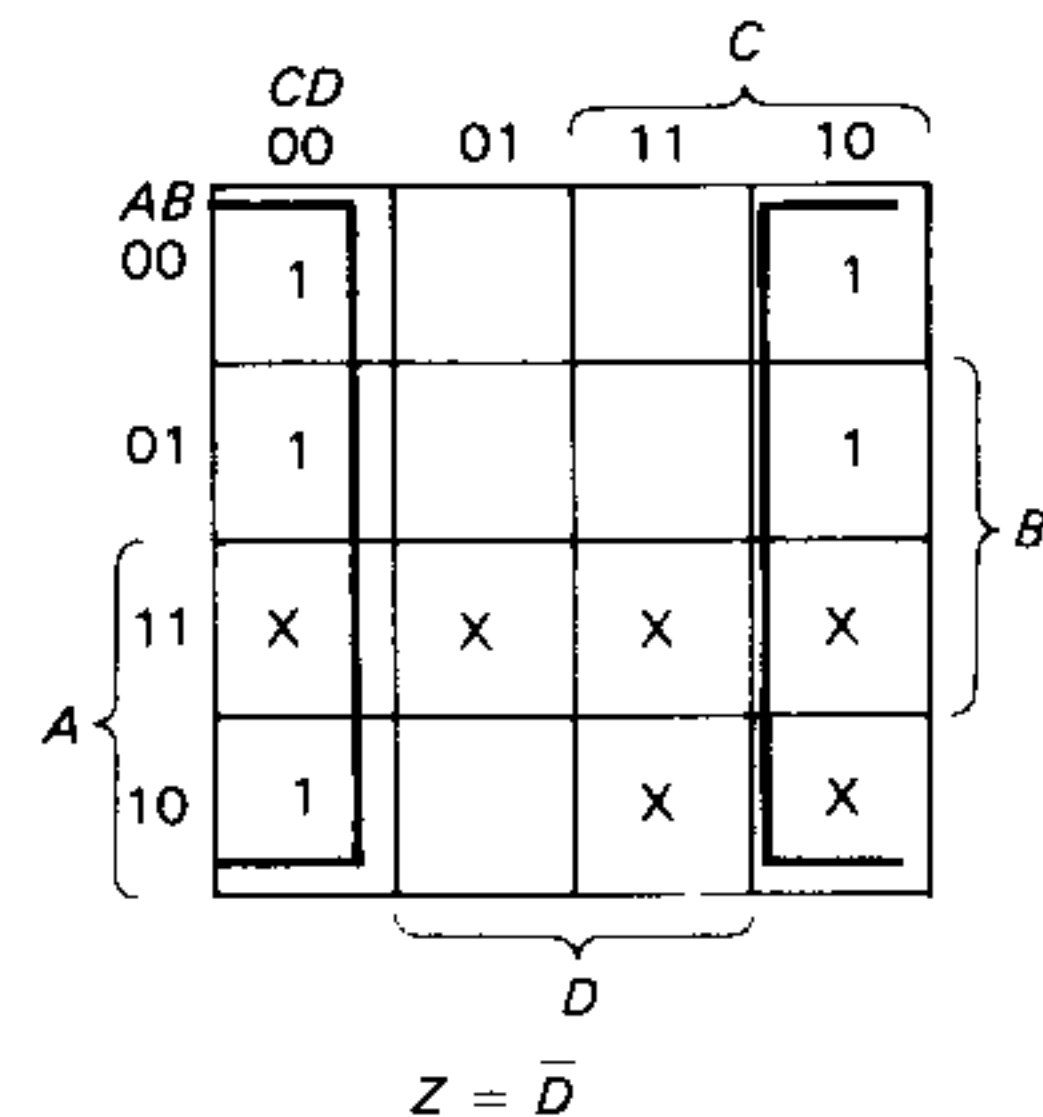
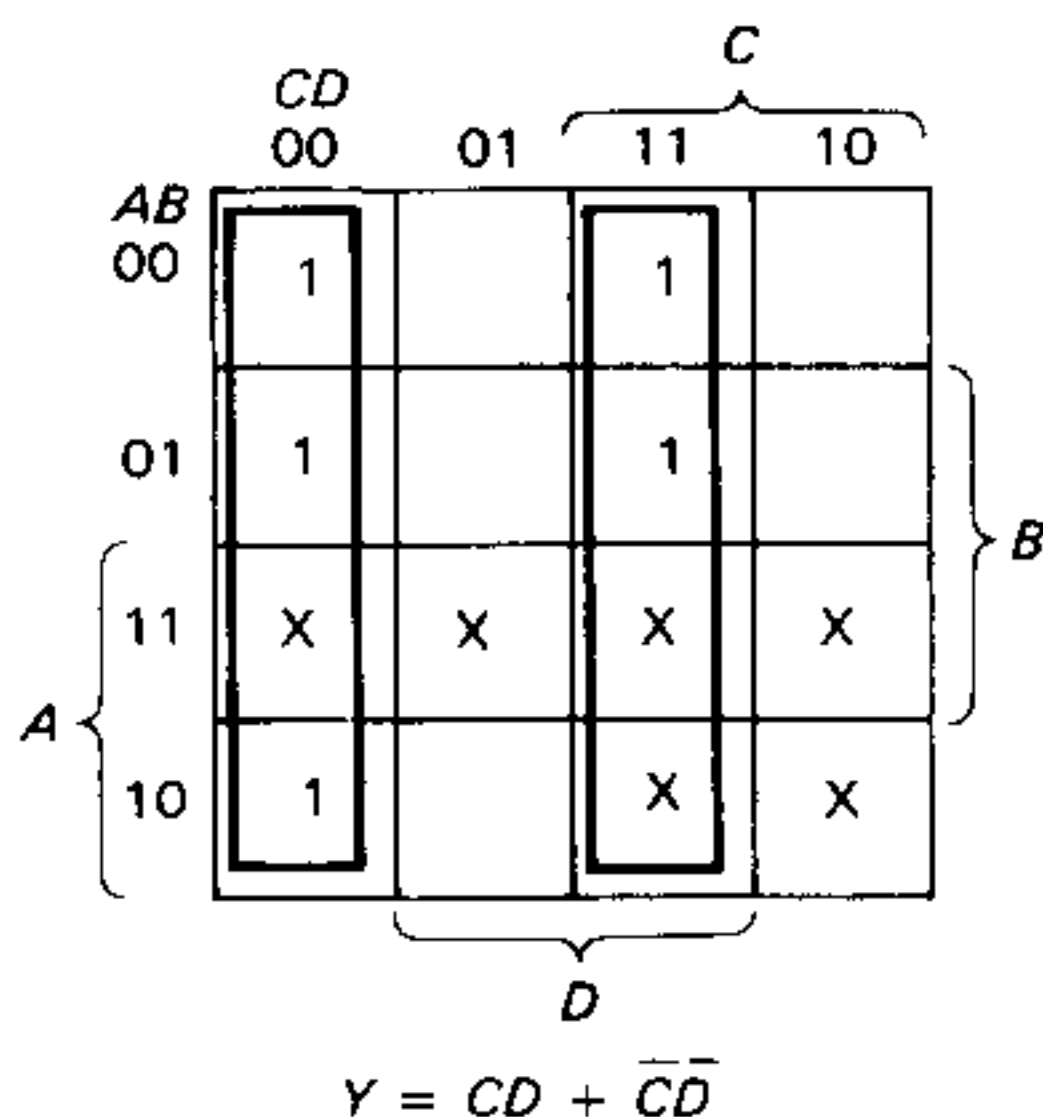
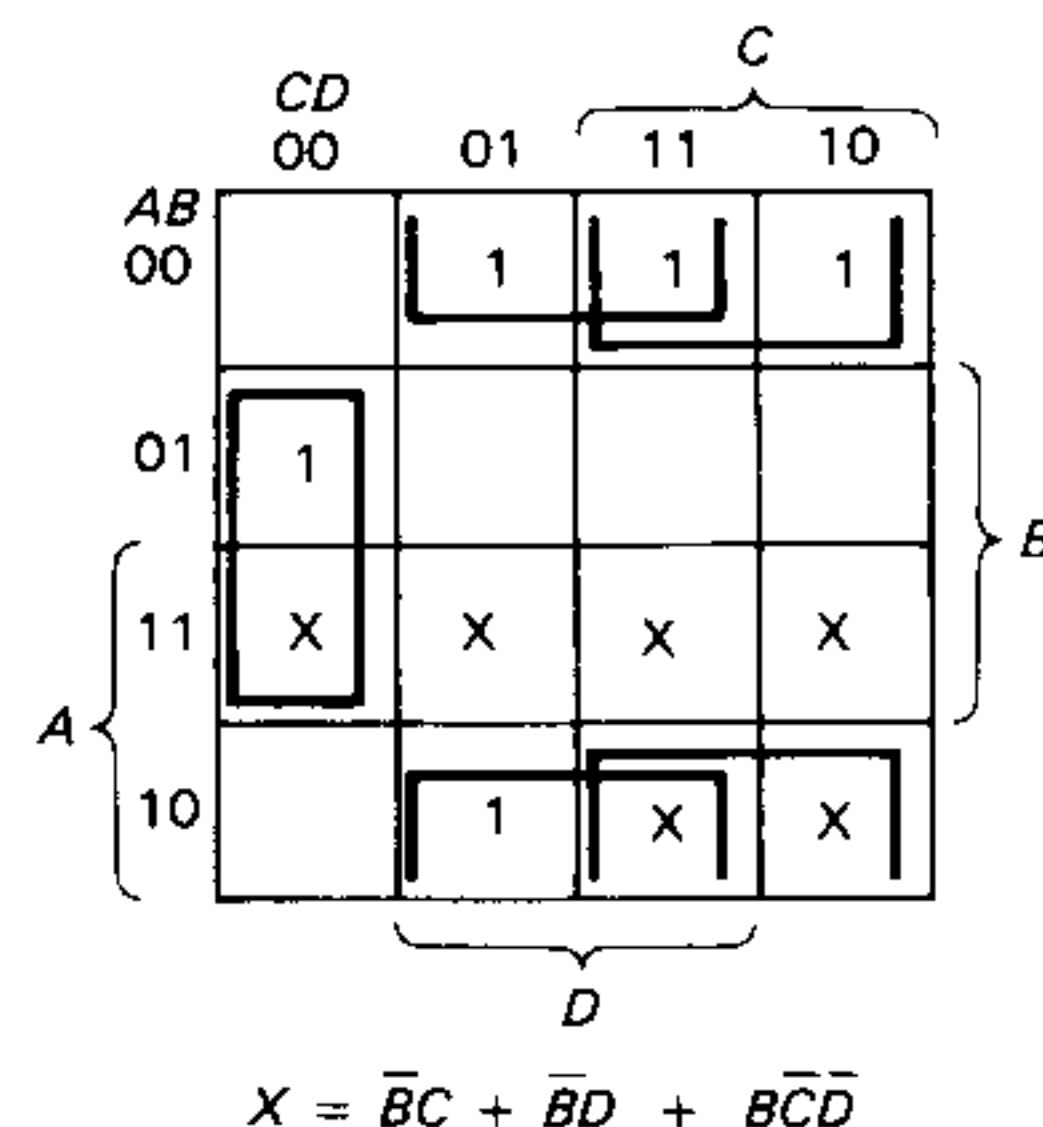
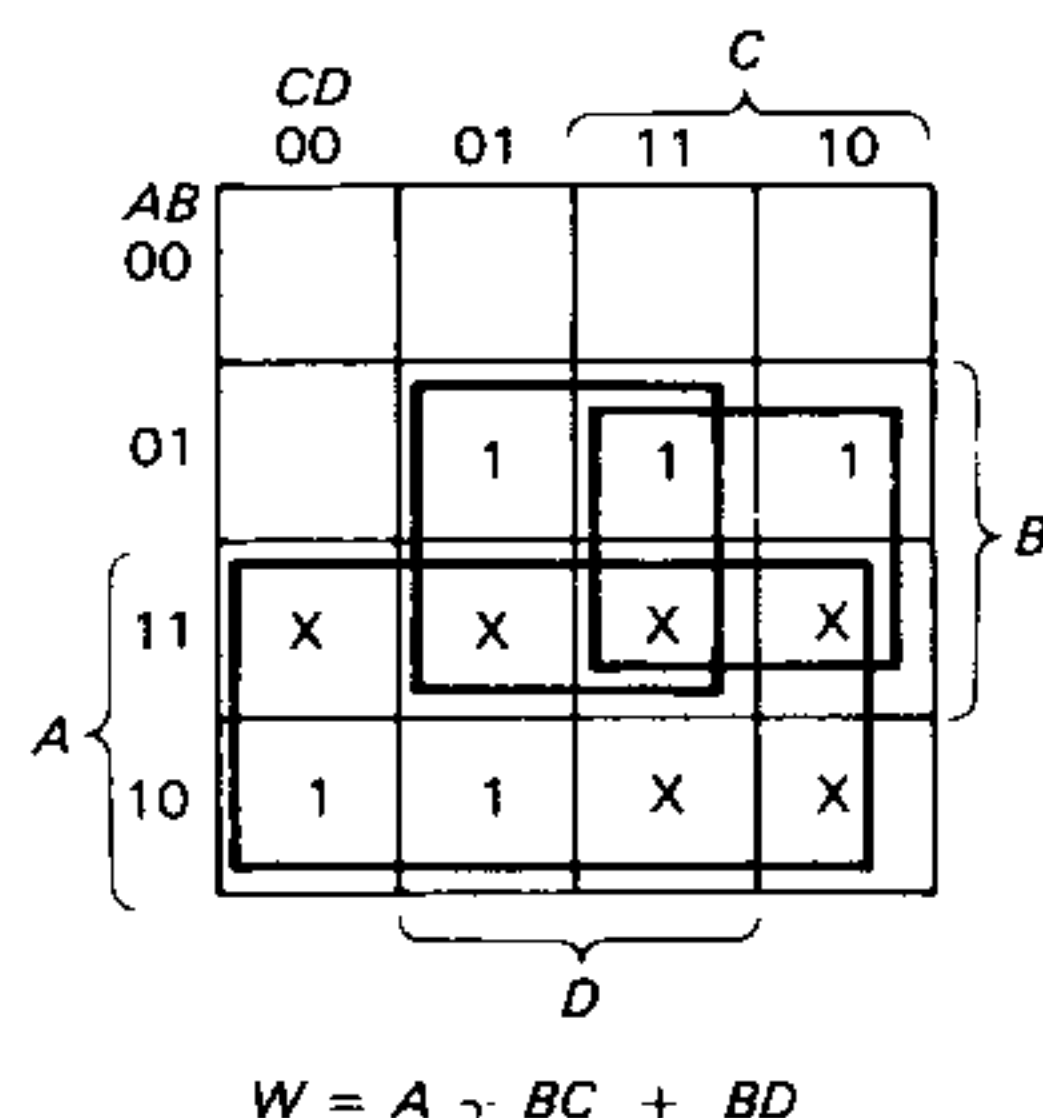


FIGURA 3—5

Mapas del convertidor de código de BCD a exceso 3

El diagrama de lógica de AND-OR de dos niveles que corresponde al circuito, se puede obtener directamente de las expresiones booleanas derivadas de los mapas. Sin embargo, existen otras posibilidades para un diagrama de lógica que aplique el circuito. Por ejemplo, las expresiones se pueden manipular algebraicamente con el fin de utilizar compuertas comunes. La manipulación que se presenta a continuación

ilustra la flexibilidad que se obtiene con sistemas de salidas múltiples cuando se construyen con tres niveles de compuertas.

$$\begin{aligned} W &= A + BC + BD = A + B(C + D) \\ X &= \overline{B}C + \overline{B}D + B\overline{C}\overline{D} = \overline{B}(C + D) + B\overline{C}\overline{D} \\ Y &= CD + \overline{C}\overline{D} = \overline{C} \oplus \overline{D} \\ Z &= \overline{D} \end{aligned}$$

El diagrama de lógica que aplica las expresiones anteriores se muestra en la figura 3-6.

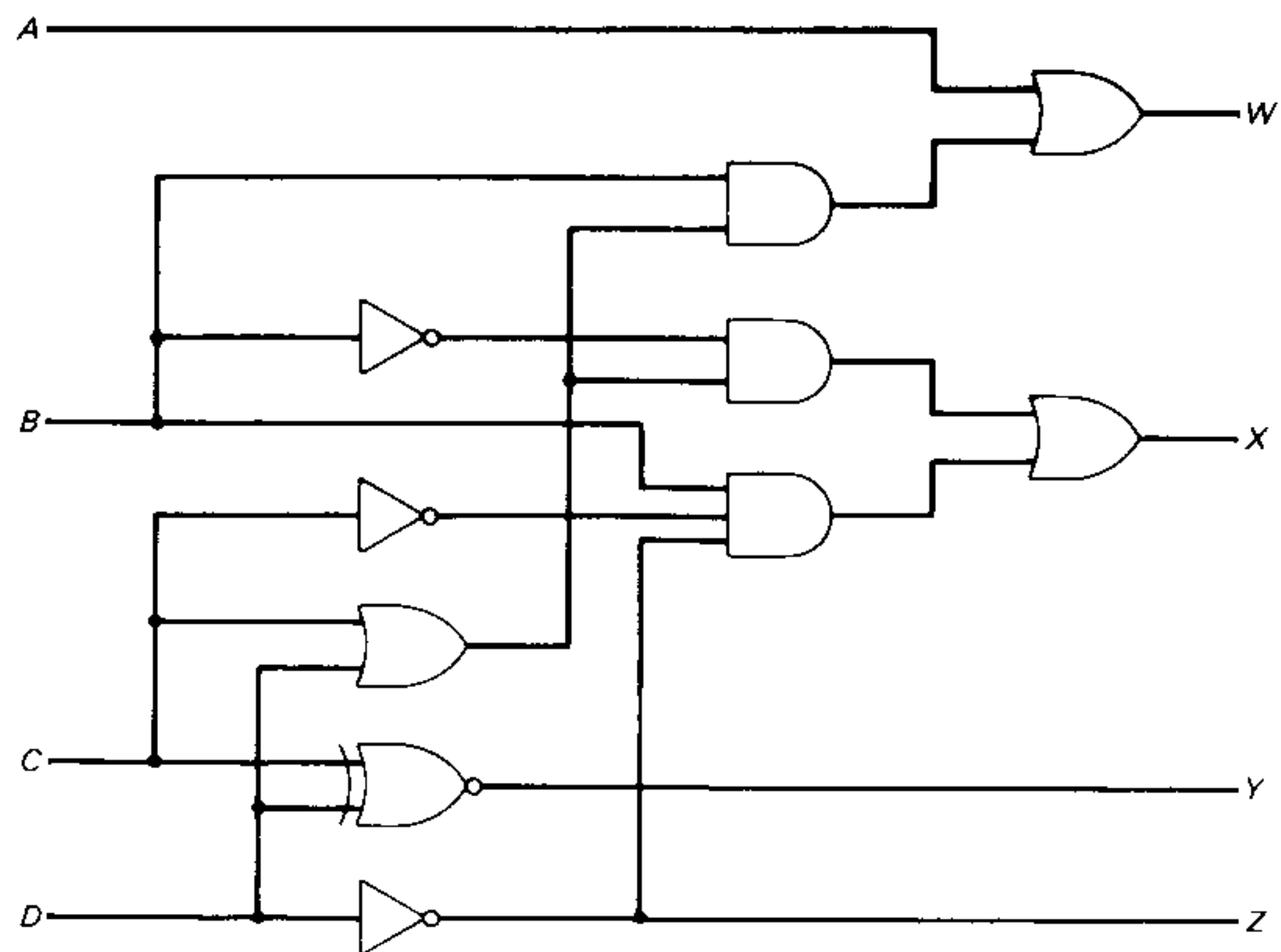


FIGURA 3—6
Diagrama de lógica del convertidor de código BCD a exceso 3

Decodificador de BCD de siete segmentos

La pantalla digital que se encuentra en calculadoras electrónicas y relojes digitales se vale de dispositivos de despliegue como diodos emisores de luz (LED) o despliegue de cristal líquido (LCD). Cada dígito del despliegue está formado por siete segmentos, cada uno de los cuales consta de un LED o cristal líquido que se puede iluminar por medio de señales digitales. Un decodificador de BCD de siete segmentos es un circuito combinatorio que acepta un dígito decimal en BCD y genera las salidas apropiadas para la selección de segmentos que despliegan el dígito decimal. Las siete salidas del decodificador (*a*, *b*, *c*, *d*, *e*, *f*, *g*) seleccionan los segmentos correspondientes en el despliegue, como se muestra en la figura 3-7(a). La designación numérica escogida para representar los dígitos decimales se presenta en la figura 3-7(b).

completo es el componente aritmético básico a partir del cual se construyen todos los demás circuitos aritméticos.

TABLA 3—4
Tabla de verdad del semisumador

Entradas		Salidas	
X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Semisumador (o sumador medio)

Un semisumador es un circuito aritmético que genera la suma de dos dígitos binarios. El circuito tiene dos entradas y dos salidas. Las variables de entrada son los bits sumando y consumando que se sumarán y las variables de salida producen la suma y el acarreo. Asignamos los símbolos X y Y a las dos entradas y S (suma) y C (acarreo) a las salidas. La tabla de verdad del semisumador se presenta en la tabla 3-4. La salida C es 1 sólo cuando ambas entradas son 1. La salida S representa el bit menos significativo de la suma. Las funciones booleanas de las dos salidas se pueden obtener fácilmente de la tabla de verdad.

$$S = \bar{X}Y + X\bar{Y} = X \oplus Y$$

$$C = XY$$

El semisumador se puede construir con una compuerta OR excluyente y una AND como se muestra en la figura 3-8.

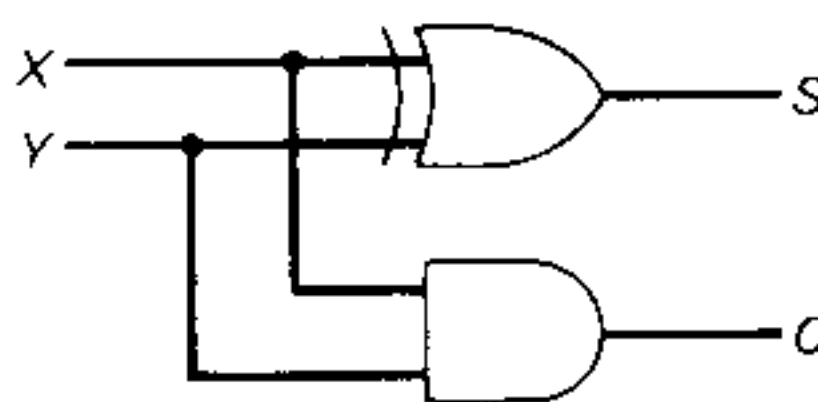


FIGURA 3—8
Diagrama de lógica del semisumador

Sumador completo

Un sumador completo es un circuito combinatorio que forma la suma aritmética de tres bits de entrada. Consta de tres entradas y dos salidas. Dos de las variables de entrada, denotadas por X y Y , representan los dos bits significativos que se sumarán. La tercera entrada Z , representa el acarreo de la posición significativa inferior anterior. Se necesitan dos salidas puesto que la suma aritmética de tres bits cambia de valor de cero a tres, y el dos y el tres binarios necesitan dos dígitos. De nuevo, las dos salidas están designadas por los símbolos S de suma y C de acarreo; la variable

binaria S da el valor del bit menos significativo de la suma, y la variable binaria C , el acarreo de salida.

La tabla de verdad del sumador completo se muestra en la tabla 3-5. Los valores de las salidas se determinan a partir de la suma aritmética de los tres bits de entrada. Cuando todos los bits de entrada son 0, las salidas valen 0. La salida S es igual a 1 sólo cuando una entrada es igual a 1 o cuando las tres entradas son iguales a 1. La salida C tiene un acarreo de 1 si dos o tres entradas son iguales a 1.

TABLA 3—5
Tabla de verdad del sumador completo

Entradas			Salidas	
X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Los mapas de las dos salidas del sumador completo se presentan en la figura 3-9. La suma simplificada de funciones de productos de las dos salidas son

$$S = \overline{X}\overline{Y}Z + \overline{X}Y\overline{Z} + X\overline{Y}\overline{Z} + XYZ$$
$$C = XY + XZ + YZ$$

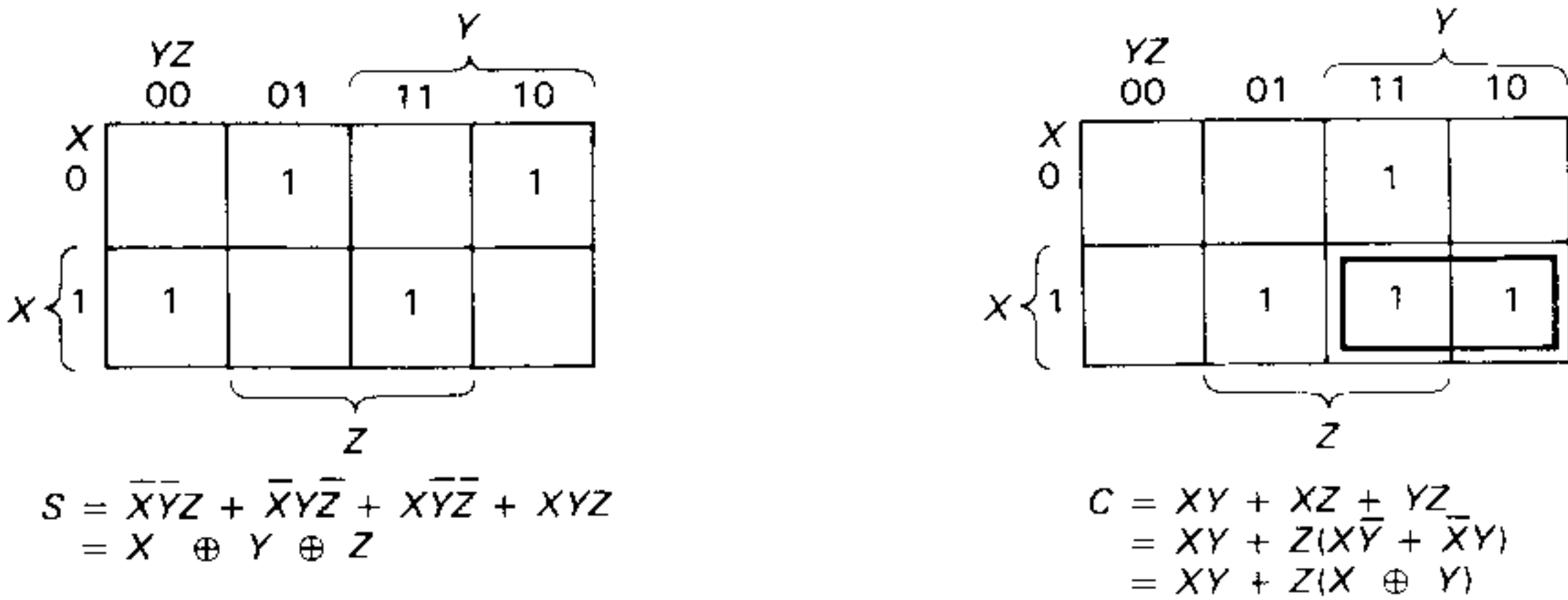


FIGURA 3—9
Mapas del sumador completo

Esta ejecución requiere siete compuertas AND y dos OR. Sin embargo, el mapa de la salida S se reconoce como una función OR excluyente, según se vio en la sección 2-7. Además, la función de salida C se puede manipular como se muestra en la figura 3-9 para incluir a la OR excluyente de X y Y . Las funciones booleanas del sumador completo en términos de operaciones OR excluyente se pueden expresar de la siguiente manera:

$$S = X \oplus Y \oplus Z$$

$$C = XY + Z(X \oplus Y)$$

El diagrama de lógica de esta ejecución se muestra en la figura 3-10. Este consta de dos semisumadores y una compuerta OR.

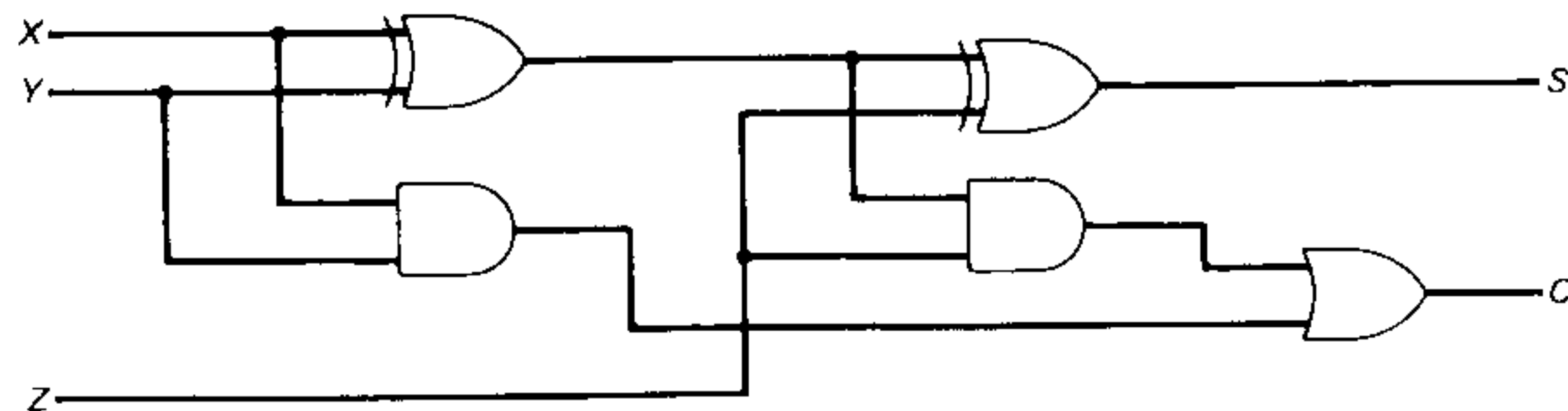


FIGURA 3—10

Diagrama de lógica del sumador completo

Sumador paralelo binario

La suma de dos números binarios de n bits se puede generar en serie o en paralelo. El método de adición en serie se vale sólo de un sumador completo y un dispositivo de almacenamiento para alojar el acarreo de salida. Cada par de bits se transfiere uno a la vez a través del sumador completo para producir una cadena de bits de salida para la suma. El acarreo de salida almacenado de un par de bits se usa como acarreo de entrada para el siguiente par de bits. El método paralelo emplea n sumadores completos y todos los bits se aplican en forma simultánea a fin de producir la suma.

Un sumador paralelo binario es un circuito digital que produce la suma aritmética de dos números binarios en paralelo. Consta de circuitos sumadores completos conectados en cascada, con el acarreo de salida de un sumador completo conectado al acarreo de entrada del sumador completo siguiente. La figura 3-11 ilustra la interconexión de cuatro sumadores completos (FA) para producir un sumador paralelo de cuatro bits. Los bits del consumando A y los del sumando B se designan por números de subíndice de derecha a izquierda, donde el subíndice 0 denota el bit de orden inferior o más bajo. Los acarreos se conectan en cadena a través de los sumadores completos. El acarreo de entrada al sumador en paralelo es C_0 y el de salida es C_4 . Un sumador paralelo de n bits requiere n sumadores completos con cada acarreo

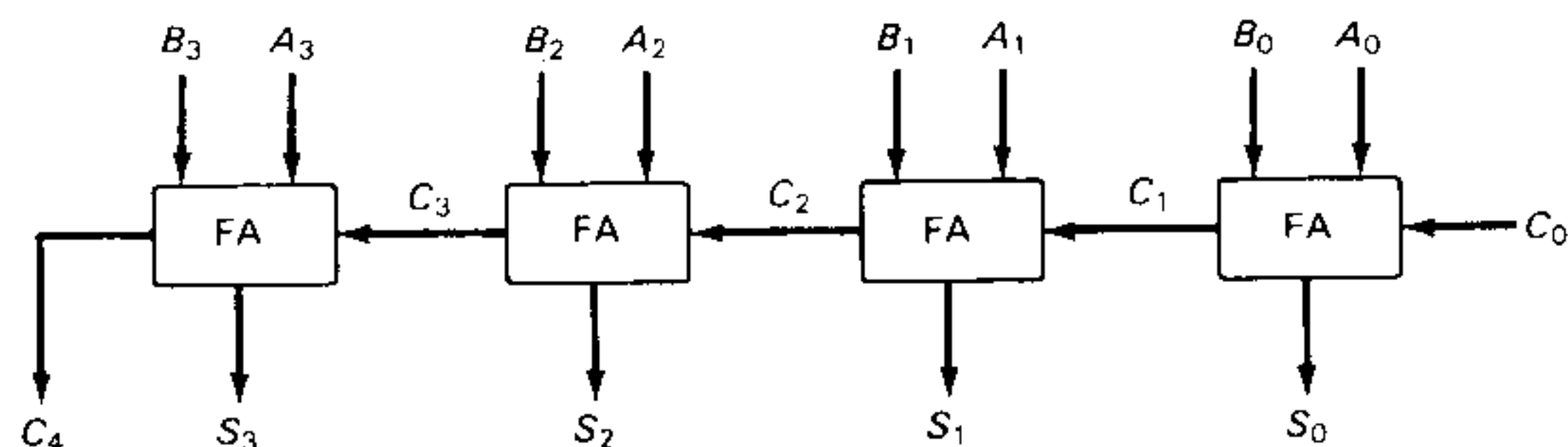


FIGURA 3—11

Sumador paralelo de 4 bits

Sumad

de salida conectado al acarreo de entrada del sumador completo de siguiente orden superior.

Considérense los números binarios $A = 1011$ y $B = 0011$. Su suma $S = 1110$ se forma con un sumador paralelo de cuatro bits como sigue:

Acarreo de entrada	0 1 1 0
Consumando A	1 0 1 1
Sumando B	0 0 1 1
Suma S	1 1 1 0
Acarreo de salida	0 0 1 1

El acarreo de entrada en la posición menos significativa es 0. Cada sumador completo recibe los bits correspondientes de A y B y el acarreo de entrada, y genera el bit de la suma de S y el acarreo de salida. El acarreo de salida en cada posición se transfiere al acarreo de entrada de la posición de orden superior siguiente.

El sumador de cuatro bits es un ejemplo típico de un componente digital que está disponible en paquetes con integración a mediana escala (MIS). Este se puede utilizar en muchas aplicaciones en que se efectúan operaciones aritméticas. Obsérvese que el diseño de este circuito por el método habitual requeriría una tabla de verdad con 512 entradas o registros porque hay nueve entradas al circuito. Aplicando un método iterativo de colocación en cascada de un circuito sumador completo ya conocido, es posible obtener una ejecución sencilla y directa.

Sumador-restador binario

La resta o sustracción de números binarios se puede hacer en forma más conveniente por medio de complementos, según se vio en la sección 1-4. Recuérdese que la resta $A - B$ puede efectuarse calculando el complemento a 2's de B y sumándolo a A . El complemento a 2's se puede obtener determinando el complemento a 1's y sumando uno al bit menos significativo. El complemento a 1's se puede efectuar fácilmente con circuitos inversores y podemos sumar uno a la suma haciendo que el acarreo de entrada del sumador paralelo sea igual a 1.

El circuito para hacer la resta $A - B$ consta de un sumador en paralelo, como el de la figura 3-11, con inversores entre cada terminal B y la entrada del sumador completo correspondiente. El acarreo de entrada C_0 debe ser igual a 1. La operación que se realiza se convierte en A más el complemento a 1's de B más 1. Esto es igual a A más el complemento a 2's de B . En el caso de números sin signo, esto produce $A - B$ si $A \leq B$, o el complemento a 2's de $(B - A)$ si $A < B$. Para números con signo, el resultado es $A - B$ siempre que no haya desbordamiento (véase la sección 1-5).

Las operaciones de suma y resta se pueden combinar en un circuito con un sumador binario común. Esto se hace incluyendo una compuerta OR excluyente con cada sumador completo. En la figura 3-12 se presenta un circuito sumador-restador de cuatro bits. La entrada S controla la operación. Cuando $S = 0$, el circuito es un sumador, y cuando $S = 1$ se convierte en restador. Cada compuerta OR excluyente recibe la entrada S y una de las entradas de B . Cuando $S = 0$ se tiene $B \oplus 0 = B$. Los sumadores completos reciben el valor de B , el acarreo de entrada es 0 y el circuito hace la suma A más B . Cuando $S = 1$, se tiene $B \oplus 1 = \bar{B}$ y $C_0 = 1$. El circuito realiza la operación A más el complemento a 2's de B .

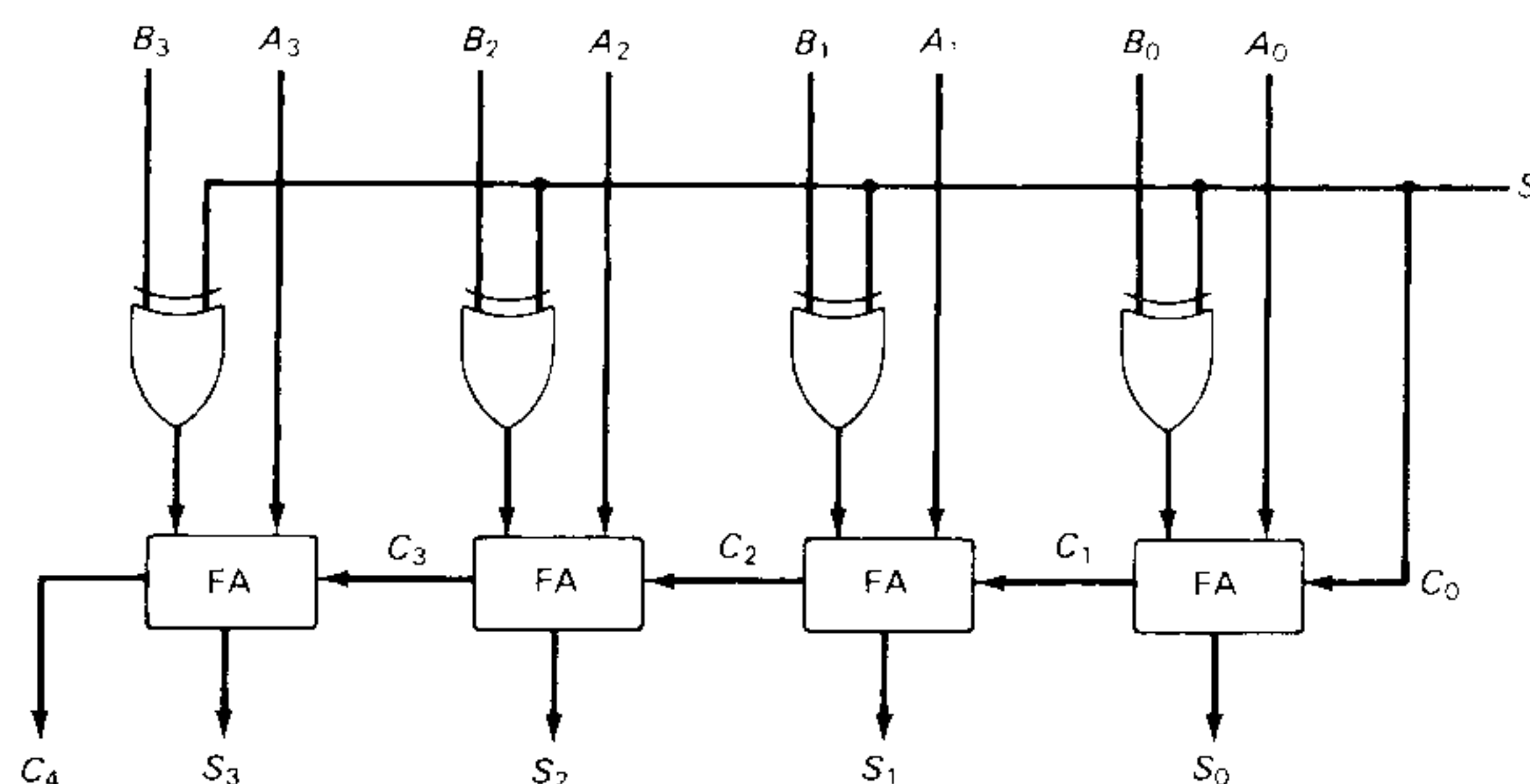


FIGURA 3—12

Circuito sumador-restador

Sumador BCD

Las computadoras o calculadoras que efectúan operaciones aritméticas directamente en el sistema numérico decimal, representan los números decimales en forma codificada en binario. Un sumador para un sistema de este tipo debe emplear circuitos aritméticos que acepten números decimales codificados y presenten resultados en el mismo código. En el caso de la suma o adición binaria, basta considerar un par de bits significativos junto con un acarreo previo. Un sumador decimal requiere un mínimo de nueve entradas y cinco salidas porque se necesitan cuatro bits para codificar cada dígito decimal y el circuito debe tener un acarreo de entrada y uno de salida. Hay una amplia variedad de circuitos sumadores decimales posibles, dependiendo del código que se utilice para representar los dígitos decimales. Aquí consideraremos un sumador decimal para el código BCD.

Las reglas de la adición o suma BCD se establecieron en la sección 1-6. Primero, se suman los dígitos BCD como si fueran dos números binarios de cuatro bits. Cuando la suma binaria es menor que o igual a 1001 (9 decimal), la suma de dígitos BCD correspondiente es correcta. Sin embargo, cuando la suma binaria es mayor que 1001, se obtiene un resultado BCD no válido. La suma del 0110 binario (6 decimal) a la suma binaria la convierte en la representación BCD correcta y produce también un acarreo de salida, según se requiera.

Considérese la adición de dos dígitos decimales en BCD con un acarreo de entrada. Como cada dígito decimal no es mayor que 9, la suma no puede ser mayor que $9 + 9 + 1 = 19$. El circuito lógico que verifica la corrección BCD necesaria se puede obtener determinando la presencia de los números binarios del 1010 al 10011 (10 al 19 decimales). Es evidente que se necesita hacer una corrección cuando la suma binaria tiene un acarreo de salida. Esta condición ocurre cuando la suma es mayor de o igual a 16. Las otras seis combinaciones de 1010 a 1111 que necesitan corrección tienen un 1 en la posición más significativa y un 1 en la segunda o tercera posición significativa.

Un sumador BCD que suma dos dígitos BCD y produce un dígito de suma en BCD se ilustra en la figura 3-13. Este tiene dos sumadores binarios de cuatro bits y un circuito de corrección. Los dos dígitos decimales junto con un acarreo de entrada se suman en el primer sumador binario de cuatro bits para producir la suma binaria. La condición para que haya corrección se puede expresar mediante la función booleana siguiente.

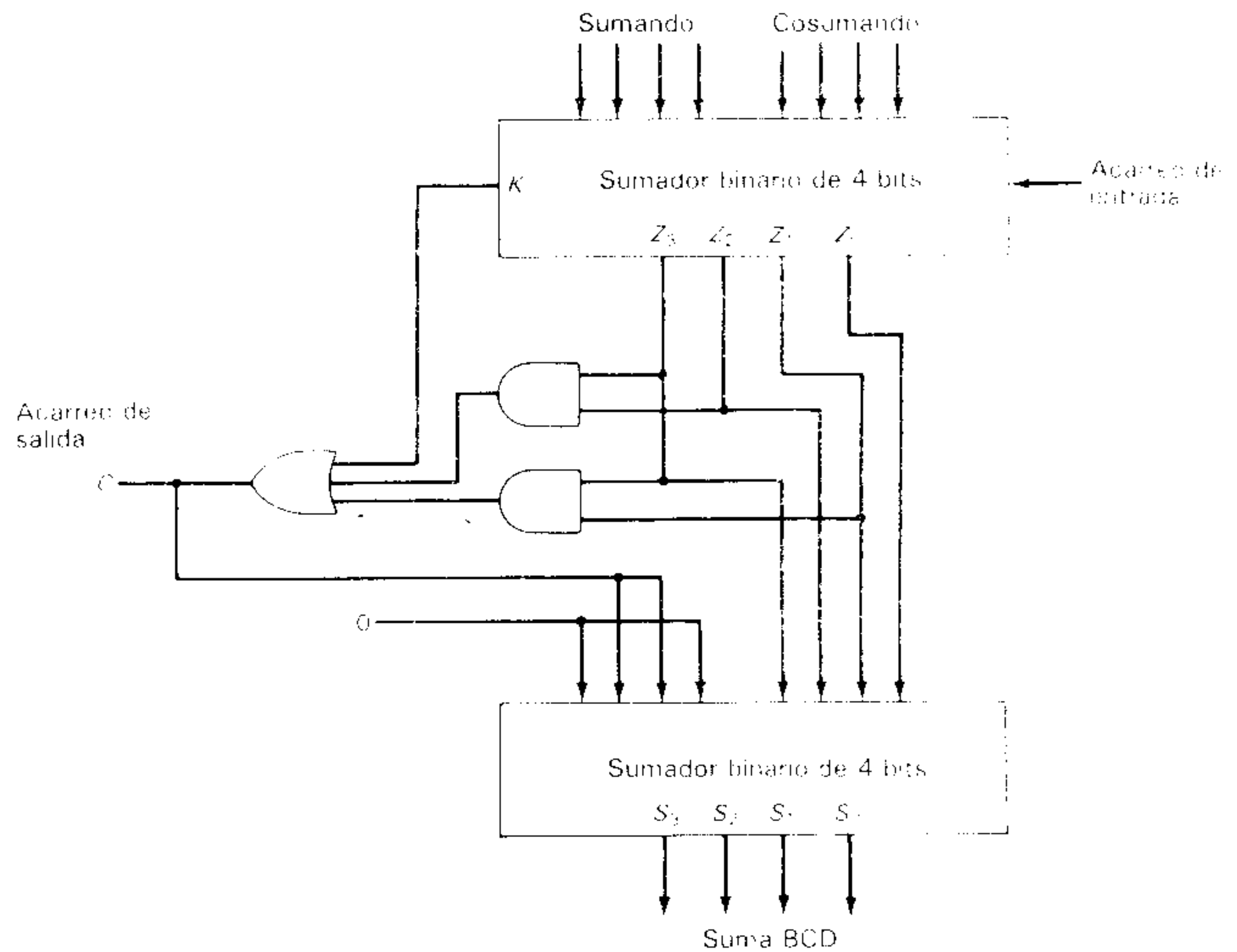


FIGURA 3—13

Multiplicador de bloque del sumador BCD

$$C = K + Z_1Z_3 + Z_2Z_3$$

C es el acarreo de salida del sumador BCD y K es el acarreo de salida del primer sumador binario. Los dos términos con las variables Z detectan las salidas binarias de 1010 a 1111.

Cuando el acarreo BCD es igual a 0, no se suma nada a la suma binaria. Esta condición se presenta si la suma de los dos dígitos más el acarreo de entrada es menor que o igual a 1001 binario. Cuando el acarreo de salida es igual a 1, el 0110 binario se suma a la suma binaria a través del segundo sumador de cuatro bits. Esta condición ocurre cuando la suma es mayor que o igual a 1010. Cualquier acarreo de salida del segundo sumador binario se puede depreciar.

Un sumador paralelo decimal que suma dos dígitos decimales n necesita n sumadores BCD. El acarreo de salida de cada sumador BCD debe conectarse al acarreo de entrada del sumador en la siguiente posición superior.

Multiplicador binario

La multiplicación de números binarios se realiza en la misma forma que con números decimales. El multiplicando se multiplica por cada bit del multiplicador, empezando desde el bit menos significativo. Cada multiplicación de esta manera forma un producto parcial. Los productos parciales sucesivos se corren una posición a la izquierda. El producto final se obtiene de la suma de los productos parciales.

Para ver cómo se puede construir un multiplicador binario con un circuito combinatorio, considérese la multiplicación de dos números de dos bits como se muestra en la figura 3-14. Los bits del multiplicando son B_1 y B_0 , los del multiplicador son A_1 y A_0 , y el producto es $C_3C_2C_1C_0$. El primer producto parcial se forma multiplicando B_1B_0 por A_0 . La multiplicación de dos bits como A_0 y B_0 produce un 1 si ambos bits son 1; de lo contrario produce un 0. Esta es idéntica a una operación AND. Por lo tanto, el producto parcial se puede ejecutar con compuertas AND según se muestra en el diagrama. El segundo producto parcial se forma multiplicando B_1B_0 por A_1 y se corre una posición a la izquierda. Los dos productos parciales se suman con dos circuitos semisumadores (HA). Por lo general hay más bits en los productos parciales y será necesario utilizar sumadores completos para producir la suma de los productos parciales. Nótese que el bit menos significativo del producto no tiene que pasar por un sumador porque está formado por la salida de la primera compuerta AND.

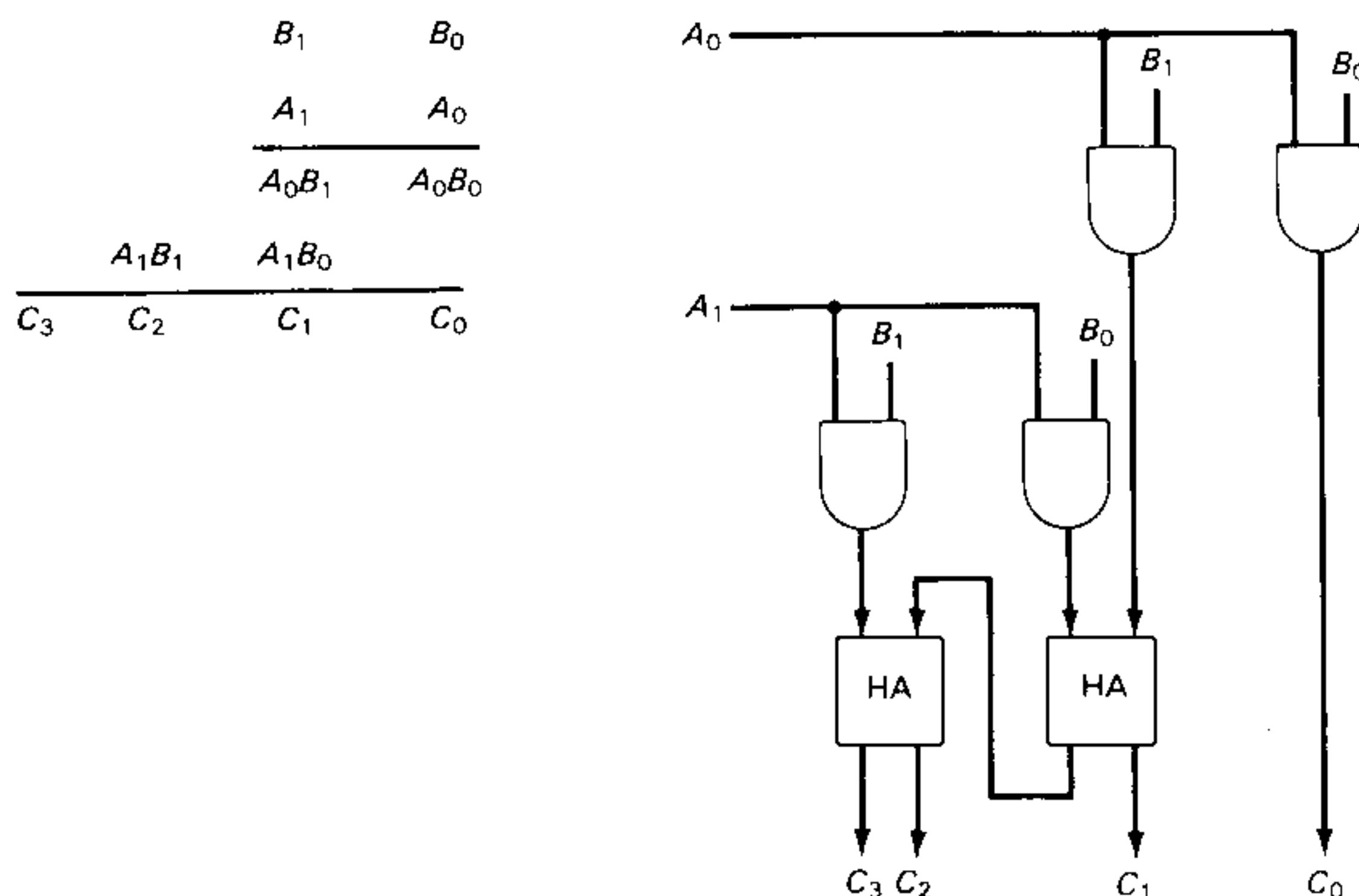


FIGURA 3—14

Multiplicador binario de 2 por 2 bits

Se puede construir en forma semejante un multiplicador binario de un circuito combinatorio con más bits. Un bit del multiplicador se compara (con AND) con cada bit del multiplicando en tantos niveles como bits haya en el multiplicador. La salida binaria en cada nivel de compuertas AND se suma en paralelo con el producto parcial del nivel previo a fin de formar un nuevo producto parcial. El último nivel genera el producto. En el caso de J bits del multiplicador y K bits del multiplicando

se necesitan $J \times K$ compuertas AND y $(J - 1)$ sumadores de K bits para formar un producto de $J + K$ bits.

Para poner un segundo ejemplo, considérese un circuito multiplicador que multiplica un número binario de cuatro bits por un número de tres bits. Sea que el multiplicando esté representado por $B_3B_2B_1B_0$ y el multiplicador por $A_2A_1A_0$. Como $K = 4$ y $J = 3$, necesitamos 12 compuertas AND y dos sumadores de cuatro bits para generar un producto de siete bits. El diagrama de lógica del circuito multiplicador se presenta en la figura 3-15.

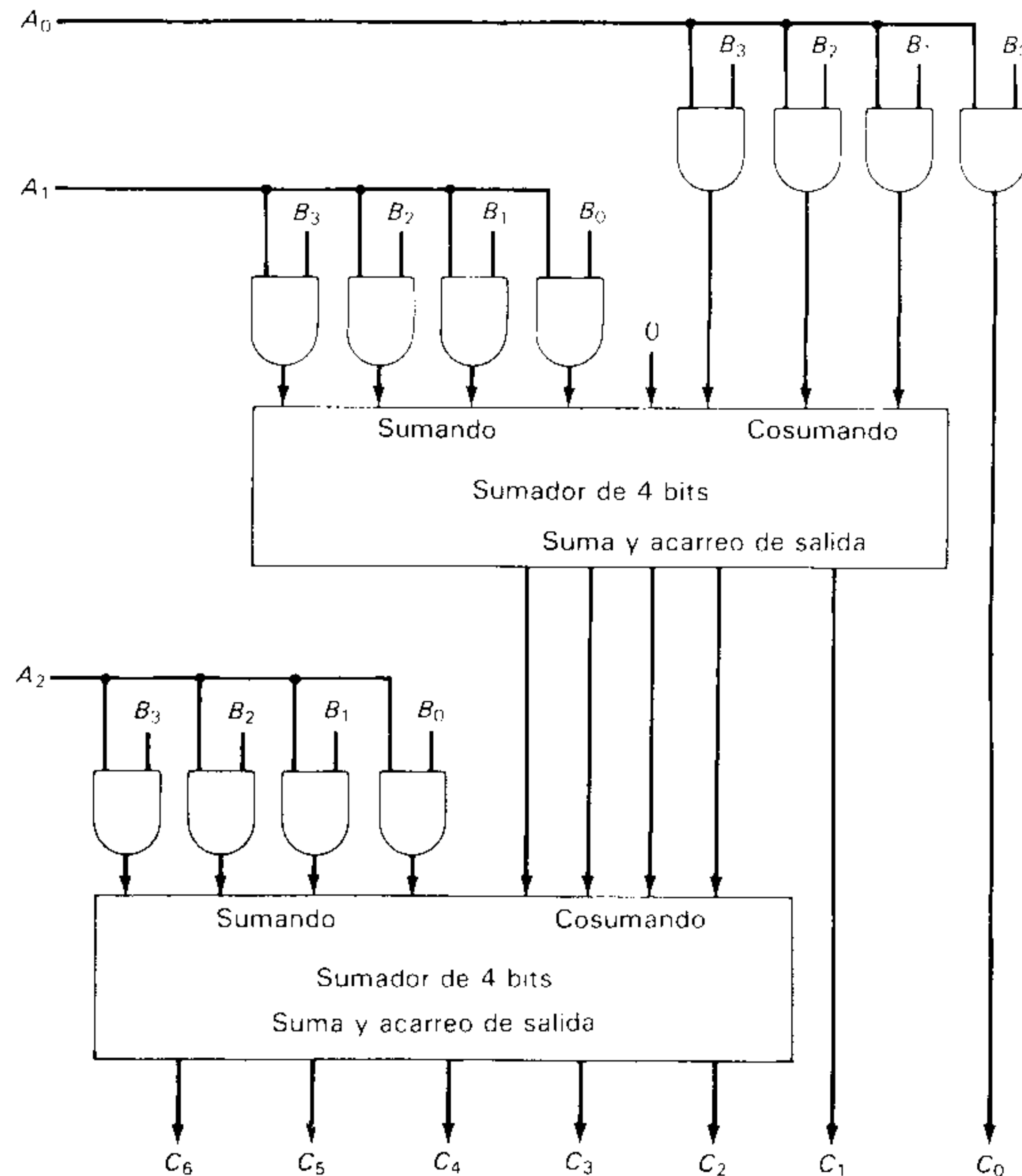


FIGURA 3—15

Multiplicador binario de 4 por 3 bits

3-5 DECODIFICADORES

Las cantidades discretas de información se representan en computadoras digitales con códigos binarios. Un código binario de n bits puede representar hasta 2^n elementos distintos de la información codificada. Un decodificador es un circuito combinatorio que convierte información binaria de las n entradas codificadas a un máximo

de 2^n salidas únicas. Si la información codificada de n bits tiene combinaciones no usadas de bits, el codificador puede tener menos de 2^n salidas.

Los decodificadores que se presentan en esta sección reciben el nombre de decodificadores de n en m líneas, donde $m \leq 2^n$. Su objetivo consiste en generar los 2^n (o menos) minitérminos de n variables de entrada. Un decodificador tiene n entradas y m salidas y se conoce como decodificador $n \times m$.

El diagrama de lógica de un decodificador de 3 líneas en 8 líneas se muestra en la figura 3-16. Las tres entradas se decodifican o descifran en ocho salidas, donde cada salida representa uno de los minitérminos de las tres variables de entrada. Los tres inversores producen el complemento de las entradas y cada una de las ocho compuertas AND genera uno de los términos mínimos. Una aplicación particular de este decodificador es una conversión de binario en octal. Las variables de entrada representan un número binario y las salidas denotan los ocho dígitos del sistema numérico octal. Pese a ello, se puede utilizar un decodificador de 3 líneas en 8 líneas para descifrar cualquier código de tres bits con el fin de producir ocho salidas, una para cada elemento del código.

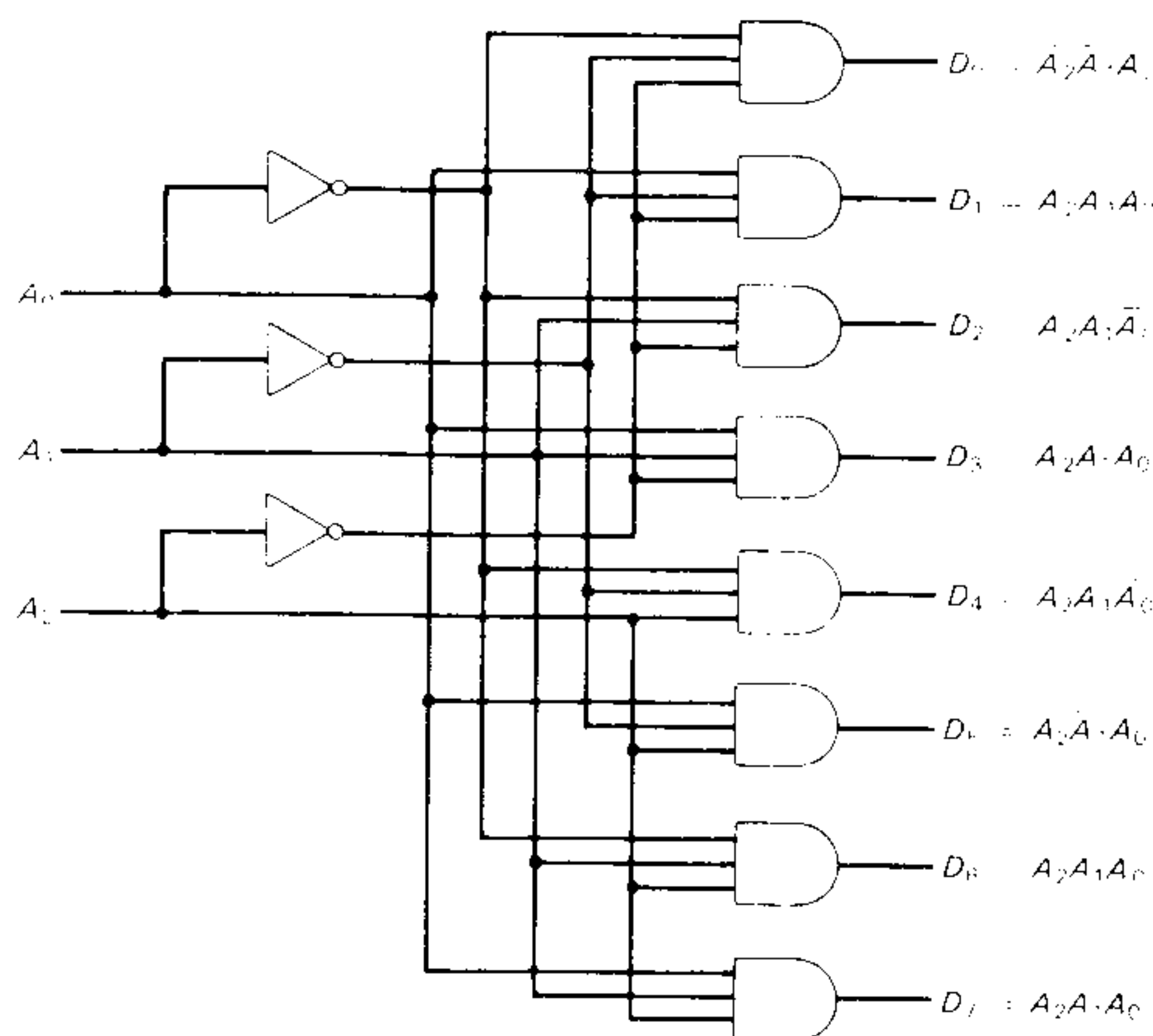


FIGURA 3—16

Decodificador de 3 líneas en 8 líneas

La operación del decodificador se puede explicar en forma más precisa a partir de la tabla de verdad de la tabla 3-6. Para cada combinación de entrada posible, hay siete salidas que son iguales a 0 y sólo una igual a 1. La variable de salida que es igual a 1 representa el término mínimo equivalente del número binario disponible en las líneas de entrada.

combinaciones no

l nombre de deco-
en generar los 2^n (o
tiene n entradas y

as se muestra en la
alidas, donde cada
entrada. Los tres
de las ocho com-
n particular de este
s de entrada repre-
sistema numérico
8 líneas para des-
das, una para cada

$$A_2 \bar{A}_1 \bar{A}_0$$

$$A_2 \bar{A}_1 A_0$$

$$\bar{A}_2 A_1 \bar{A}_0$$

$$\bar{A}_2 A_1 A_0$$

$$A_2 \bar{A}_1 \bar{A}_0$$

$$A_2 \bar{A}_1 A_0$$

$$\bar{A}_2 A_1 \bar{A}_0$$

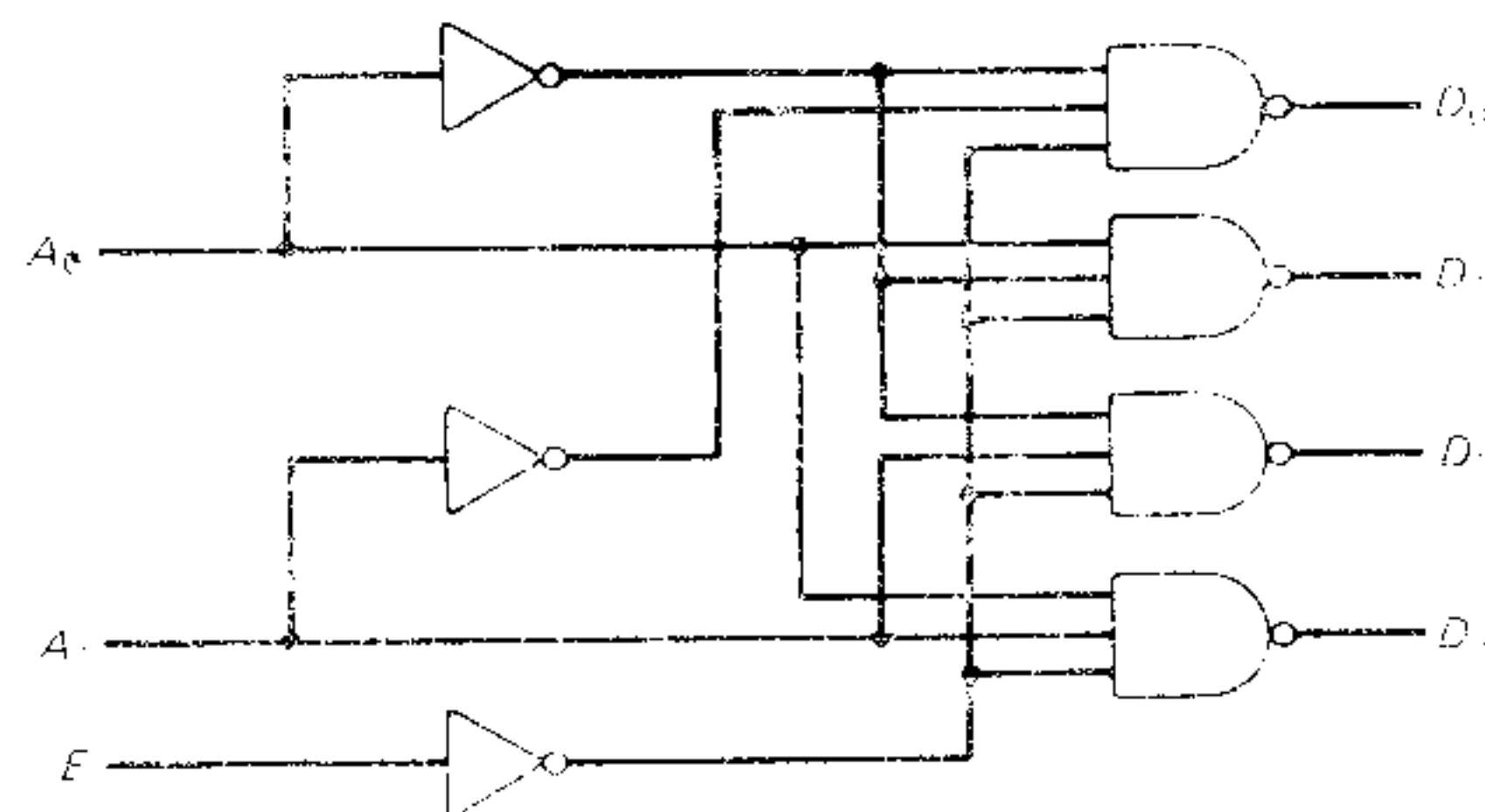
$$\bar{A}_2 A_1 A_0$$

precisa a partir de
rada posible, hay
salida que es igual
disponible en las

TABLA 3—6
Tabla de verdad del decodificador de 3 líneas en 8 líneas

Entradas			Salidas							
A_2	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Algunos decodificadores se construyen con compuertas NAND en vez de AND. Debido a que una compuerta NAND produce la operación AND con una salida invertida, se vuelve más económico generar los minitérminos del decodificador en su forma de complemento. Además, la mayoría de los decodificadores comerciales incluyen una o más entradas *habilitadoras* para controlar la operación del circuito. Un decodificador de 2 líneas en 4 líneas con una entrada habilitadora construido con compuertas NAND se muestra en la figura 3-17. El circuito opera con salidas complementadas y una entrada habilitadora complementada \bar{E} . El decodificador se habilita cuando E es igual a 0. Como lo indica la tabla de verdad, sólo una salida puede ser igual a 0 en cualquier momento dado y todas las otras salidas son igual a 1. La salida con un valor de 0 representa el minitérmino seleccionado por las entradas A_1 y A_0 . El circuito se inhabilita o desactiva cuando E es igual a 1, independientemente de los valores de las otras dos entradas. Cuando se desactiva el circuito, ninguna de las salidas es igual a 0 y ninguno de los términos mínimos es seleccionado. En general, un decodificador puede operar con salidas complementadas o no complementadas. La entrada habilitadora se puede activar con una señal 0 o 1. Algunos decodificadores tienen dos o más entradas habilitadoras que deben satisfacer una condición lógica dada a fin de habilitar o activar el circuito.



(a) Diagrama de lógica

E	A_1	A_0	D_7	D_6	D_5	D_4
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	X	X	1	1	1	1

(b) Tabla de verdad

FIGURA 3—17
Decodificador de 2 líneas en 4 líneas con entrada habilitadora

Expansión del decodificador

Hay ocasiones en que se necesita un decodificador de cierto tamaño pero sólo se dispone de tamaños menores. Cuando ocurre esto, es posible combinar dos o más de estos dispositivos con entradas habilitadoras para formar un decodificador de mayor capacidad. Por lo tanto, si se necesita un decodificador de 6 líneas en 64 líneas, es posible construirlo con cuatro decodificadores de 4 líneas en 16 líneas.

La figura 3-18 presenta cómo se pueden conectar decodificadores con entradas habilitadoras para formar un decodificador de mayor tamaño. Se combinan dos decodificadores de 2 líneas en 4 líneas para obtener uno de 3 líneas en 8 líneas. Los dos bits menos significativos de la entrada se conectan a ambos decodificadores. El bit más significativo se conecta a la entrada habilitadora de un decodificador y a través de un inversor a la entrada habilitadora del otro decodificador. Cuando $A_2 = 0$, se habilita el decodificador superior y el otro se desactiva. Las salidas del dispositivo inferior quedan inactivas con todas las salidas en 0. El decodificador superior genera los minitérminos del D_0 al D_3 utilizando los valores de A_1 y A_0 . Cuando $A_2 = 1$, se invierten las condiciones de habilitación y se generan los términos mínimos del D_4 al D_7 . Nótese que estos decodificadores se habilitan con una señal 1 lógico.

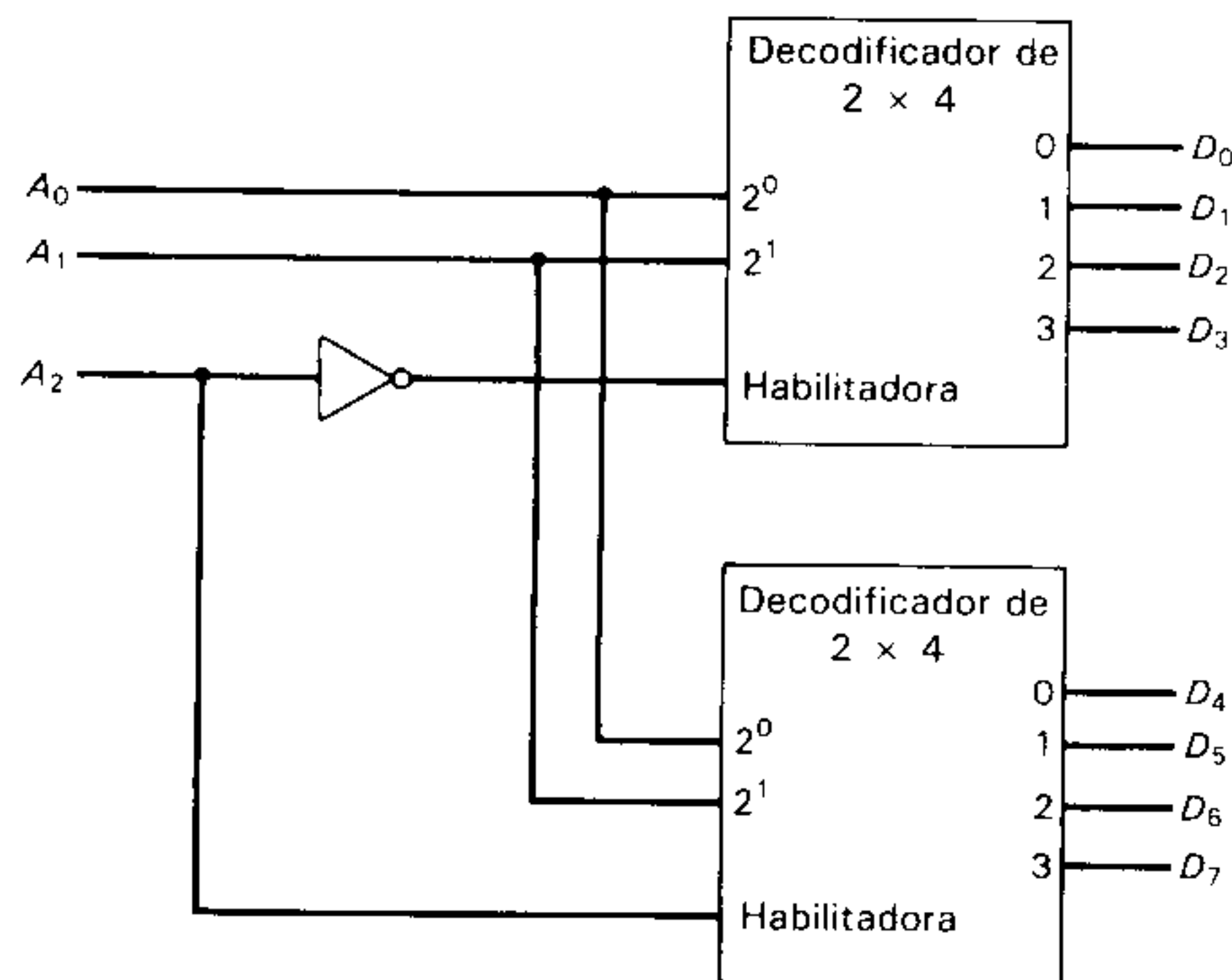


FIGURA 3—18

Decodificador de 3 × 8 construido con dos decodificadores de 2 × 4

El ejemplo demuestra la utilidad de la entrada habilitadora en decodificadores o cualquier otro componente lógico combinatorio. Las entradas habilitadoras son una característica adecuada para interconectar dos o más circuitos integrados con el fin de ampliar la función digital en una función análoga con más entradas y salidas.

Diseño de circuitos combinatorios

Un decodificador genera los 2^n minterminos de n variables de entrada. Como cualquier función booleana se puede expresar como una suma de minterminos, se puede emplear un decodificador para generar los términos mínimos y una compuerta OR externa para formar su suma lógica. En esta forma, cualquier circuito combinatorio con n entradas y m salidas se puede construir con un decodificador de n a 2^n líneas y m compuertas OR.

El procedimiento para diseñar un circuito combinatorio por medio de un decodificador y compuertas OR requiere que las funciones booleanas del circuito se expresen como una suma de minterminos. Esta forma se puede obtener de la tabla de verdad o graficando cada función en un mapa. Después se escoge un decodificador que genere todos los términos mínimos de las variables de entrada. Las entradas a cada compuerta OR se seleccionan de las salidas del decodificador de acuerdo con la lista de minterminos de cada función.

Ejemplo 3-2

Diséñese un circuito sumador completo con un decodificador y compuertas OR.

De la tabla de verdad del sumador completo (tabla 3-5) se obtienen las funciones del circuito combinatorio en suma de términos mínimos.

$$S(X, Y, Z) = \sum m(1, 2, 4, 7)$$

$$C(X, Y, Z) = \sum m(3, 5, 6, 7)$$

Como hay tres entradas y un total de ocho minterminos, necesitamos un decodificador de 3 líneas en 8 líneas. El diseño se presenta en la figura 3-19. El decodificador genera los ocho minterminos para X, Y, Z . La compuerta OR de la salida S forma la suma lógica de los minterminos 1, 2, 4 y 7. La compuerta OR de la salida C forma la suma lógica de los términos mínimos 3, 5, 6 y 7.

Una función con una larga lista de minterminos requiere una compuerta OR con un gran número de entradas. Una función que tiene una lista de K minterminos se puede expresar en su forma de complemento con $2^n - k$ términos mínimos. Si el número de minterminos en una función F es mayor que $2^n/2$ entonces su complemento

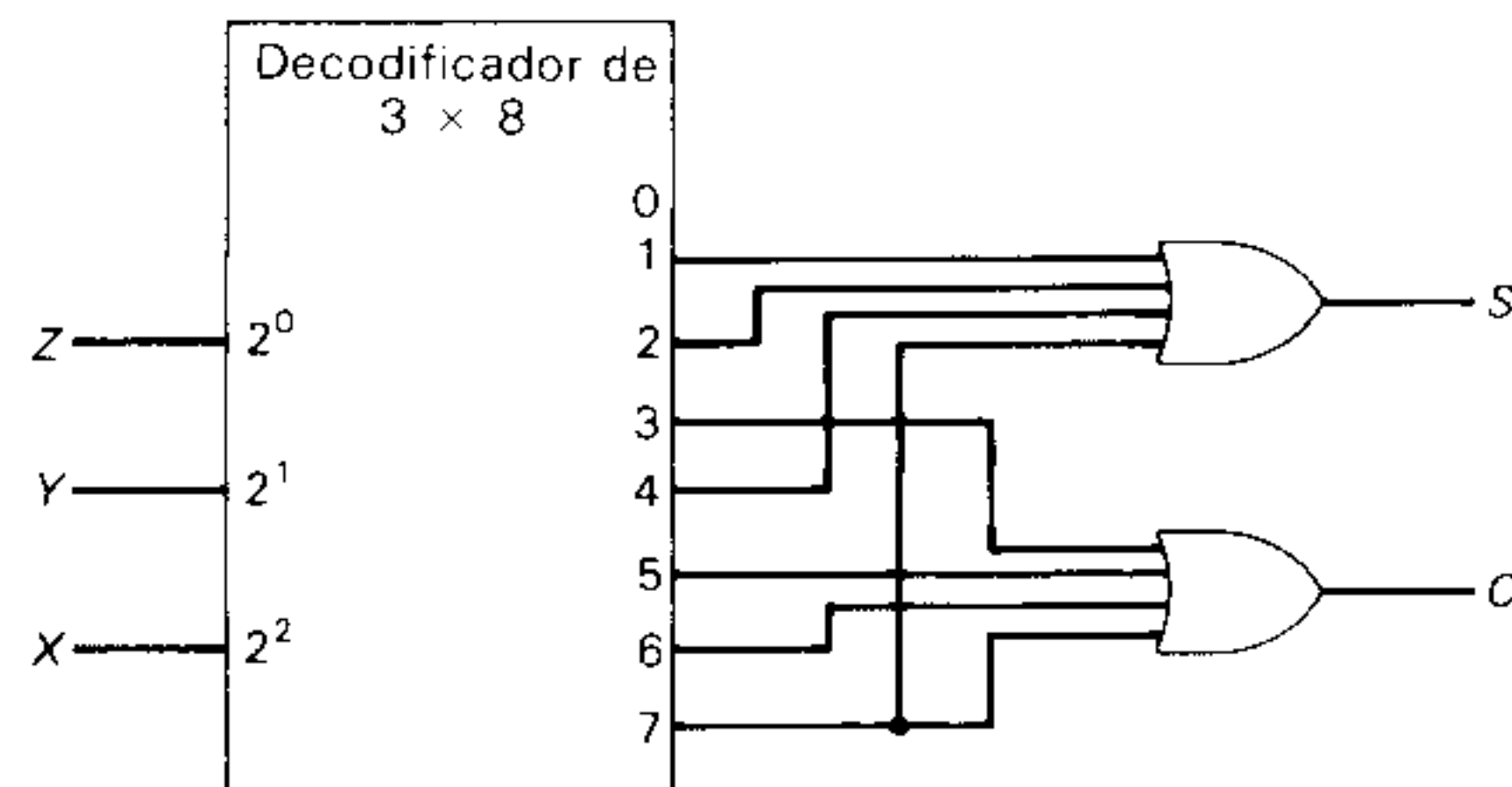


FIGURA 3—19

Construcción de un sumador completo usando un decodificador

F se puede expresar con menos términos mínimos. En tal caso, resulta ventajoso utilizar una compuerta NOR en vez de una OR. La parte OR de la compuerta produce la suma lógica de los minitérminos de \bar{F} . La salida de la compuerta NOR complementa esta suma y genera la salida normal F .

Si se utilizan compuertas NAND para el decodificador, como en la figura 3-17, las compuertas externas deben ser NAND en vez de OR. Esto se debe a que el circuito de compuertas NAND de dos niveles efectúa una suma de minitérminos y es equivalente a un circuito AND-OR de dos niveles (refiérase a la sección 2-6).

El método del decodificador se puede aplicar para diseñar cualquier circuito combinatorio. No obstante, su construcción debe compararse con todas las demás construcciones posibles a fin de determinar la mejor solución. En algunos casos este método puede generar la solución más adecuada, en especial si el circuito combinatorio tiene muchas salidas y cada función de salida se expresa con un número pequeño de minitérminos.

3-6 CODIFICADORES

Un codificador es una función digital que realiza la operación inversa de un decodificador. Un codificador tiene 2^n (o menos) líneas de entrada y n líneas de salida. Las líneas de salida generan el código binario correspondiente al valor de entrada. Un ejemplo de codificador es el codificador de octal en binario cuya tabla de verdad está representada en la tabla 3-7. Este tiene ocho entradas, una para cada uno de los dígitos octales, y tres salidas que generan el número binario correspondiente. Se supone que sólo una entrada tiene un valor de 1 en cualquier momento dado; en caso contrario, el circuito no tiene significado.

TABLA 3—7
Tabla de verdad del codificador de octal en binario

D_7	D_6	D_5	Entradas					Salidas		
			D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

El codificador se puede construir con compuertas OR utilizando entradas determinadas directamente de la tabla de verdad. La salida $A_0 = 1$ si el dígito octal de entrada es 1, 3, 5 o 7. Se aplican condiciones semejantes a las otras dos salidas. Estas condiciones se pueden expresar por medio de las siguientes funciones booleanas de salida:

$$A_1 = D_1 + D_2 + D_3 + D_4$$

$$A_2 = D_2 + D_3 + D_4 + D_5$$

$$A_3 = D_3 + D_4 + D_5 + D_6$$

El codificador se puede construir con tres compuertas OR.

El codificador que se define en la tabla 3-7 tiene la limitación de que solo una entrada puede estar activa en un momento dado. Si dos entradas están activas en forma simultánea, la salida produce una combinación indefinida. Por ejemplo, si D_4 y D_6 son 1 al mismo tiempo, la salida del codificador será 111 porque las tres salidas son igual a 1. Esto no representa ni un 3 ni un 6 binarios. Para resolver esta ambigüedad, los circuitos codificadores deben establecer una prioridad de entrada para asegurar que sólo se codifique una entrada. Si establecemos una prioridad mas alta para entradas con números de subíndice mayores y si D_3 y D_6 son 1 al mismo tiempo, la salida será 110 porque D_6 tiene mayor prioridad que D_3 .

Otra ambigüedad en el codificador de octal en binario es que se genera una salida de puros ceros (0) cuando todas las entradas son 0, pero esta salida es la misma cuando D_0 es igual a 1. Esta discrepancia se puede solucionar generando una salida más con el objeto de señalar que cuando menos una entrada es igual a 1.

Codificador de prioridad

Un codificador de prioridad es un circuito combinatorio que ejecuta la función de prioridad. La operación de este codificador es tal que si dos o más entradas son iguales a 1 al mismo tiempo, tomará precedencia la entrada que tenga la más alta prioridad. La tabla de verdad de un codificador de prioridad de cuatro entradas se presenta en la tabla 3-8. Las X designan condiciones no importa. La entrada D_3 tiene la más alta prioridad; de modo que sin que importen los valores de las otras entradas, cuando esta entrada es 1, la salida de A_1, A_0 es 11 (3 binario). D_2 tiene el siguiente nivel de prioridad. La salida es 10 si $D_2 = 1$ siempre que $D_3 = 0$, sin importar los valores de las otras dos entradas de menor prioridad. La salida de D_1 se genera solo si las entradas de más alta prioridad son 0, y así sucesivamente bajando por los niveles de prioridad. La salida válida designada por V se inicializa a 1 solo cuando una o más de las entradas son iguales a 1. Si todas las entradas son 0, V es igual a 0 y las otras dos entradas del circuito no se utilizan.

TABLA 3—8
Tabla de verdad del codificador de prioridad

Entradas				Salidas		
D_3	D_2	D_1	D_0	A_1	A_0	V
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

Los mapas para simplificar las salidas A_1 y A_0 se ilustran en la figura 3-20. Los minitérminos de las dos funciones se determinan a partir de la tabla 3-8. Aunque la tabla sólo tiene cinco renglones o hileras, cuando cada condición no importa se sustituye primero por 0 y luego por 1, se obtienen las 16 combinaciones de entradas posibles. Por ejemplo, el tercer renglón de la tabla con 001X representa los términos mínimos 0010 y 0011 porque a X se le puede asignar 0 o 1. Las funciones simplificadas obtenidas se citan debajo de cada mapa. La condición para que ocurra la salida V es una función OR de todas las variables de entrada. El codificador de prioridad se aplica en la figura 3-21 de acuerdo con las siguientes funciones booleanas:

$$A_0 = D_3 + D_1 \bar{D}_2$$

$$A_1 = D_2 + D_3$$

$$V = D_0 + D_1 + D_2 + D_3$$

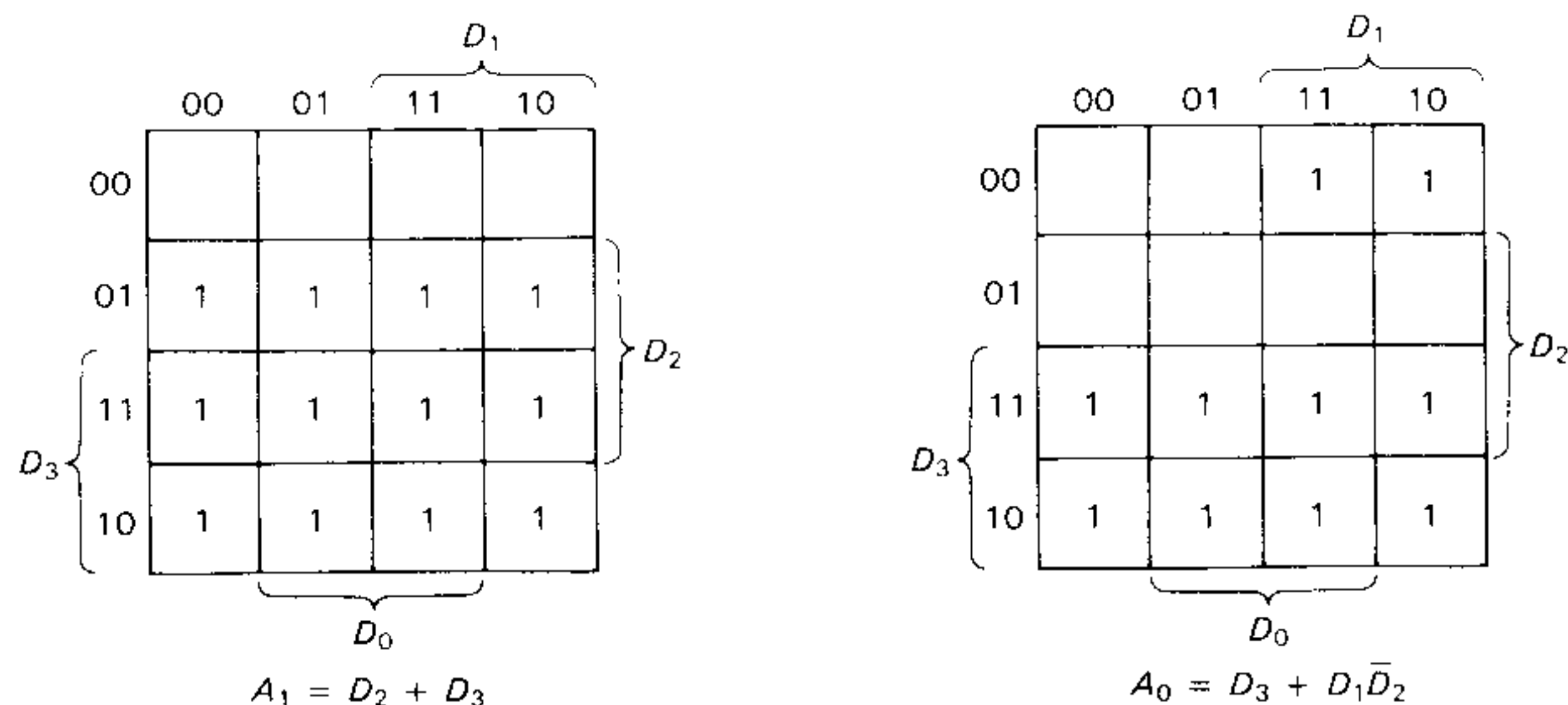


FIGURA 3—20

Mapas del codificador de prioridad

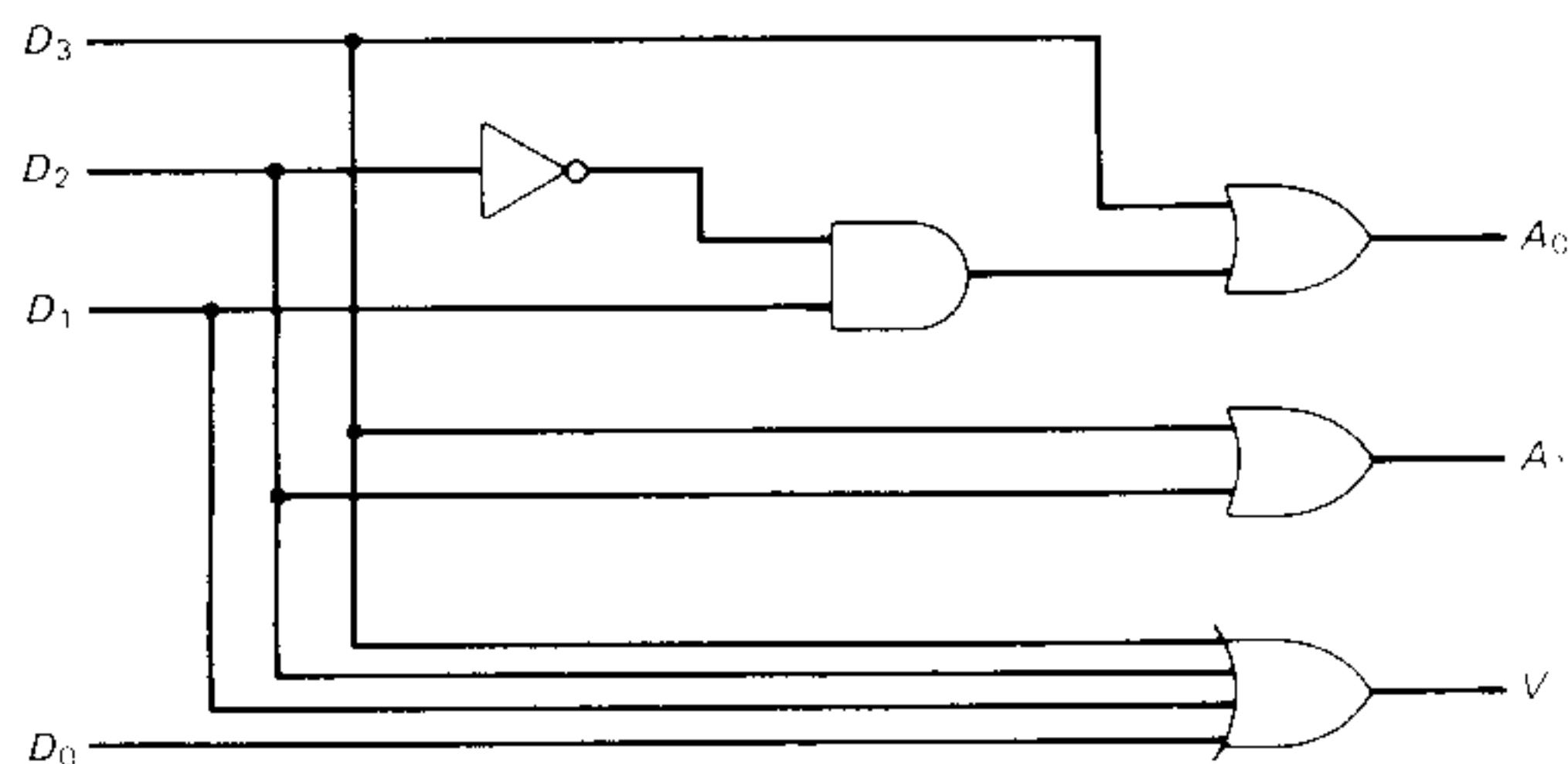


FIGURA 3—21

Diagrama de lógica de un codificador de prioridad de 4 entradas

3-7 MULTIPLEXORES

Un multiplexor (o multicanalizador, como se le conoce también) es un circuito combinatorio que selecciona información binaria de una de muchas líneas de entrada y la dirige a una sola línea de salida. La selección de una línea de entrada en particular es controlada por un conjunto de variables de selección. Normalmente hay 2^n líneas de entrada y n variables de selección cuyas combinaciones de bits determinan qué entrada se selecciona.

En la figura 3-22 se ilustra un multiplexor de 4 líneas en 1 línea. Cada una de las cuatro entradas, de la D_0 a la D_3 , se aplica a una entrada de una compuerta AND. Las líneas de selección S_1 y S_0 se decodifican para seleccionar una compuerta AND específica. Las salidas de las compuertas AND se aplican a una compuerta OR para generar la salida de una línea. Para demostrar la operación del circuito, considérese el caso cuando $S_1S_0 = 10$. La compuerta AND asociada con la entrada D_2 tiene dos de sus entradas iguales a 1 y la tercera entrada conectada a D_2 . Las otras tres compuertas AND tienen cuando menos una entrada igual a 0, lo que hace que sus salidas sean iguales a 0. La salida de la compuerta OR es ahora igual al valor de D_2 , lo que genera una trayectoria de la entrada seleccionada hacia la salida. La tabla de funciones de la figura cita la entrada que genera la trayectoria hacia la salida para cada combinación de las variables de selección binarias. Un multiplexor recibe también el nombre de *selector de datos* porque selecciona una de las muchas entradas y conduce la información binaria hacia la línea de salida.

Las compuertas e inversores AND del multiplexor se parecen a un circuito decodificador y, en realidad, decodifican las líneas de selección de entradas. En general, un multiplexor de 2^n líneas en 1 línea se construye a partir de un decodificador de n en 2^n sumando 2^n líneas de entrada a éste, una por cada entrada de datos. El tamaño del multiplexor lo especifica el número 2^n de sus líneas de entrada de datos y la línea de salida individual. Después, esto implica que contiene también n líneas de selección. Multiplexor se abrevia a menudo como MUX.

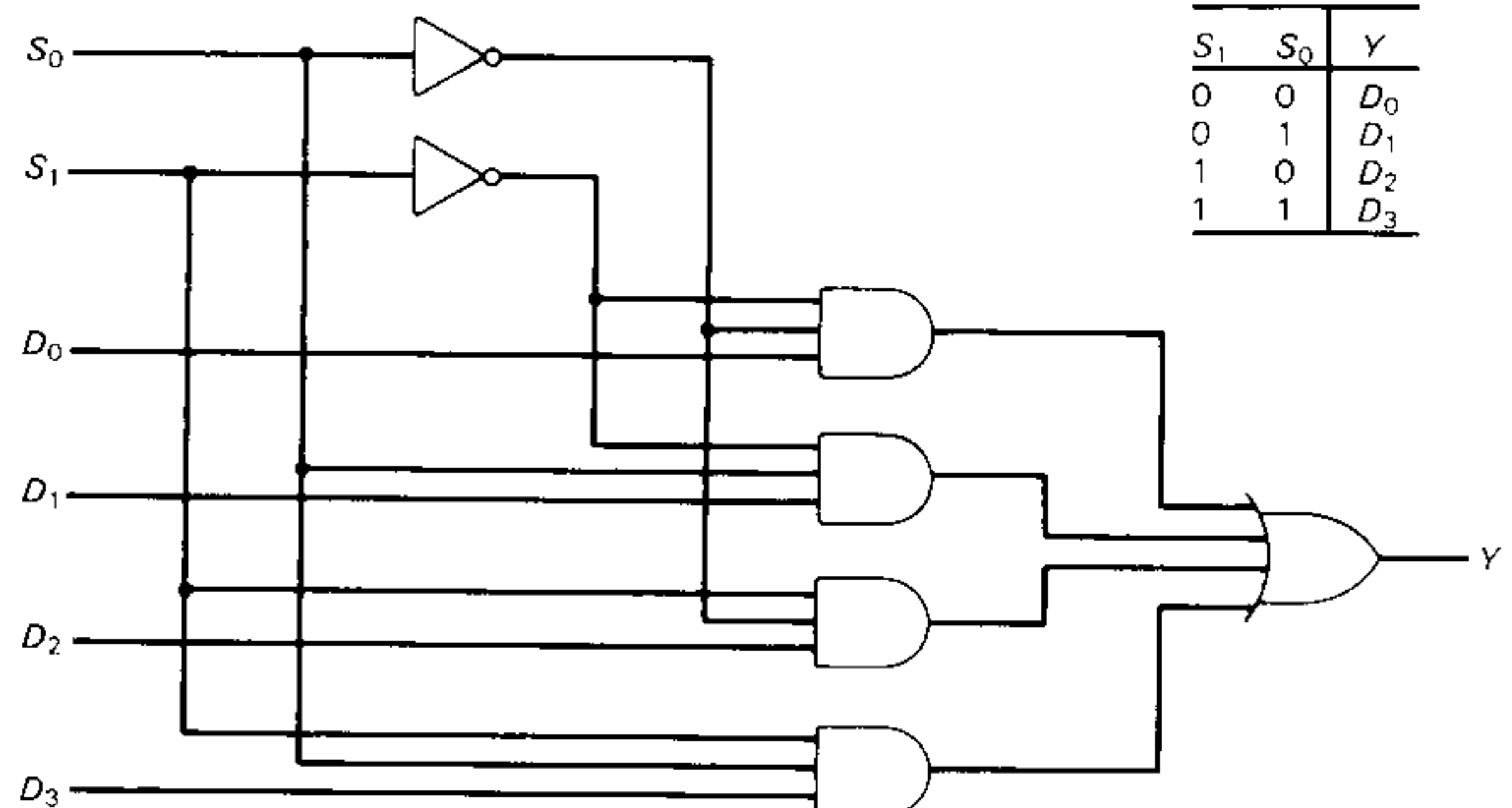


FIGURA 3—22

Multiplexor de 4 líneas en 1 línea

Un multiplexor se puede construir con compuertas de transmisión. Estas se pueden obtener sólo con circuitos integrados del tipo CMOS, y su operación se explica junto con la figura 2-37. En la figura 3-23 se presenta un multiplexor de 4 líneas en 1 línea con compuertas de transmisión (TG). El circuito TG produce una trayectoria de transmisión entre las líneas de entrada y salida horizontales cuando dos entradas de control verticales tienen el valor de 1 en la terminal no encerrada en un círculo y 0 en la que sí está encerrada en un círculo. Con una polaridad opuesta en las entradas de control, se desconecta la trayectoria y el circuito se comporta como un interruptor abierto. Las dos entradas de selección S_1 y S_0 controlan la trayectoria de transmisión en los circuitos TG. Dentro de cada cuadro está marcada la condición para que el circuito sea cerrado. Por lo tanto, si $S_0 = 0$ y $S_1 = 0$ hay una trayectoria cerrada de la entrada D_0 hacia la salida Y , y las otras tres entradas son desconectadas por uno de los otros circuitos TG.

Como los codificadores, los multiplexores pueden tener una entrada habilitadora para controlar la operación de la unidad. Cuando la entrada habilitadora se encuentra en el estado inactivo, las salidas se desactivan; y cuando está en el estado activo, el circuito funciona como un multiplexor normal. La entrada habilitadora es útil para extender dos o más multiplexores en un solo multiplexor con un mayor número de entradas.

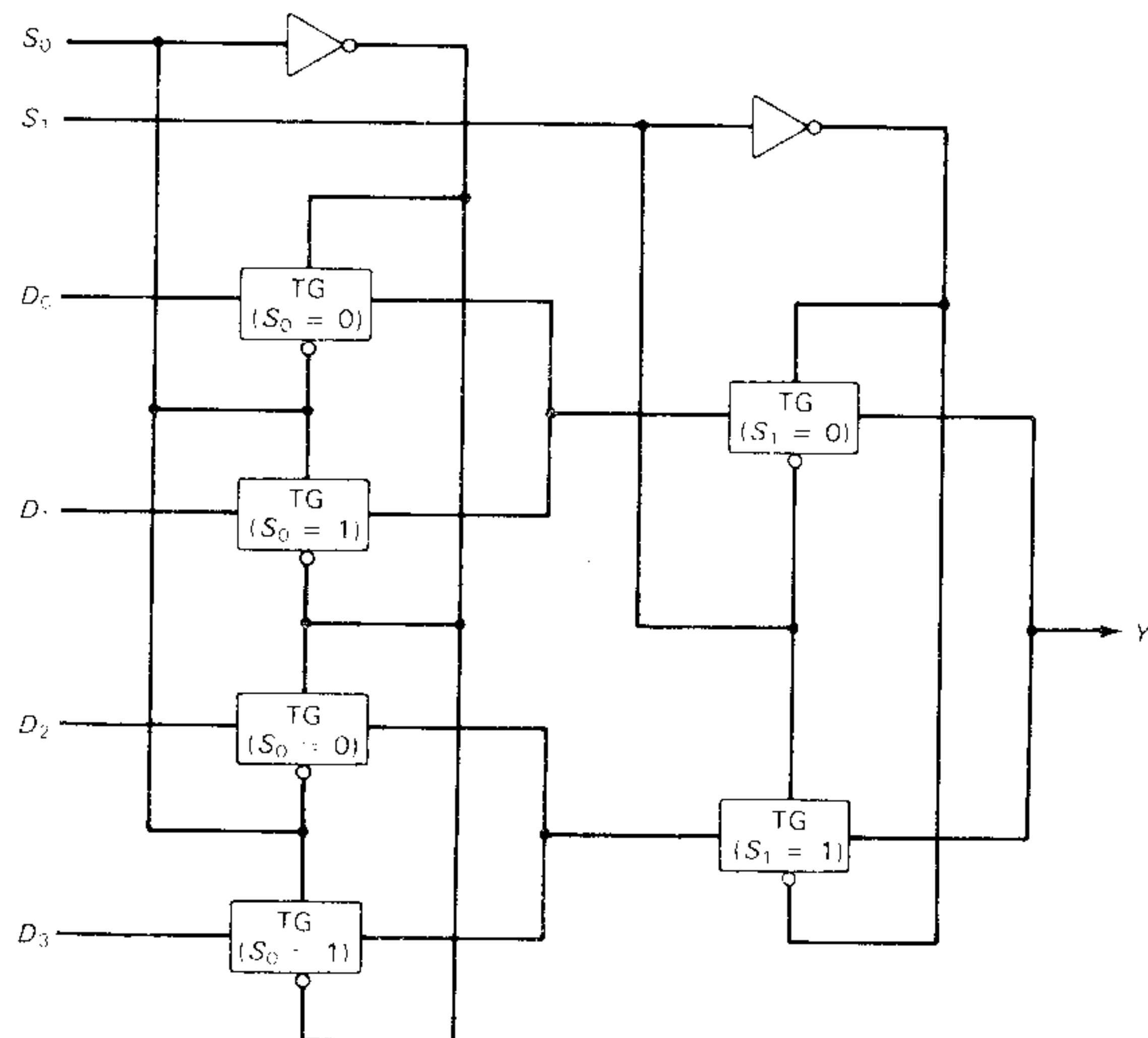


FIGURA 3—23

Multiplexor con compuertas de transmisión (TG)

En algunos casos, dos o más multiplexores están contenidos en un solo paquete de circuitos integrados. Las entradas de selección y habilitadoras en la construcción de múltiples unidades suelen ser comunes a todos los multiplexores. Para poner un ejemplo ilustrativo, en la figura 3-24 se muestra un multiplexor cuádruple de 2 líneas en 1 línea. El circuito tiene cuatro multiplexores, cada uno capaz de seleccionar una de dos líneas de entrada. La salida Y_1 se puede seleccionar proveniente de la entrada A_1 o B_1 . En forma análoga, la salida Y_1 puede tener el valor de A_1 o B_1 , y así sucesivamente. La línea de selección de entrada S selecciona una de las líneas de cada uno de los cuatro multiplexores. La entrada habilitadora E debe estar activa en operación normal. Aunque el circuito contiene cuatro multiplexores, podemos concebirlo también como un circuito que selecciona una de dos líneas de datos de cuatro bits. Como se muestra en la tabla de funciones, la unidad se habilita cuando $E = 1$. En consecuencia, si $S = 0$, las cuatro entradas A tienen una trayectoria hacia las cuatro salidas. Por otra parte, si $S = 1$, las cuatro entradas B se aplican a las salidas. Las salidas tienen sólo ceros cuando $E = 0$, independientemente de los valores de S .

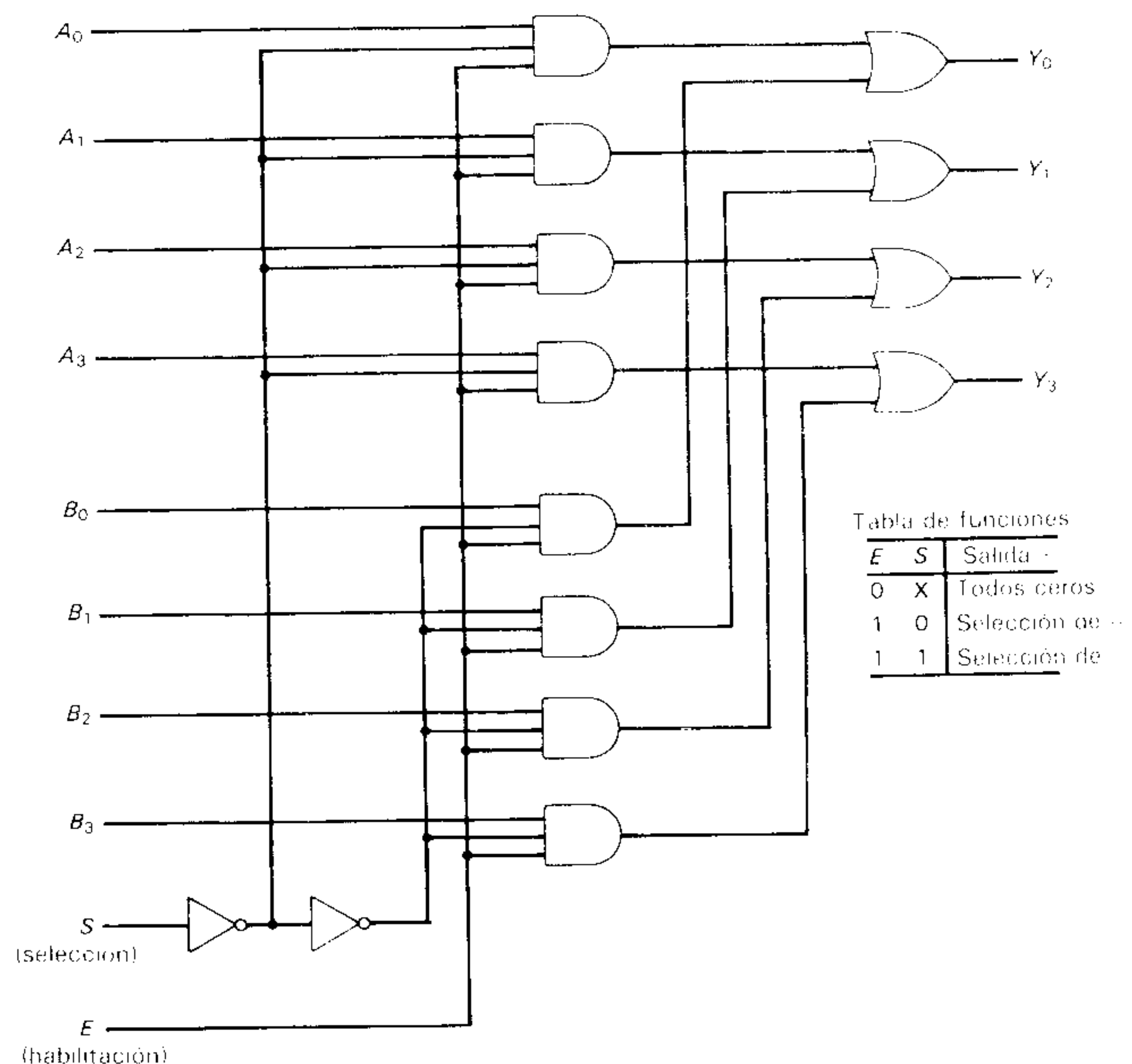


FIGURA 3—24

Multiplexor cuádruple de 2 líneas en 1 línea

Ejecución de funciones booleanas

Se demostró en la sección 3-5 que un decodificador se puede emplear para ejecutar una función booleana usando una compuerta OR externa. Si se consulta el diagrama de lógica de un multiplexor se ve que se trata básicamente de un decodificador que incluye la compuerta OR dentro de la unidad. Los minitérminos de una función se generan en un multiplexor a través del circuito asociado con las entradas de selección. Los términos mínimos individuales pueden ser seleccionados por las entradas de datos. Esto ofrece un método de ejecución de una función booleana de n variables con un multiplexor que tiene n entradas de selección y 2^n entradas de datos, una para cada minitérmino.

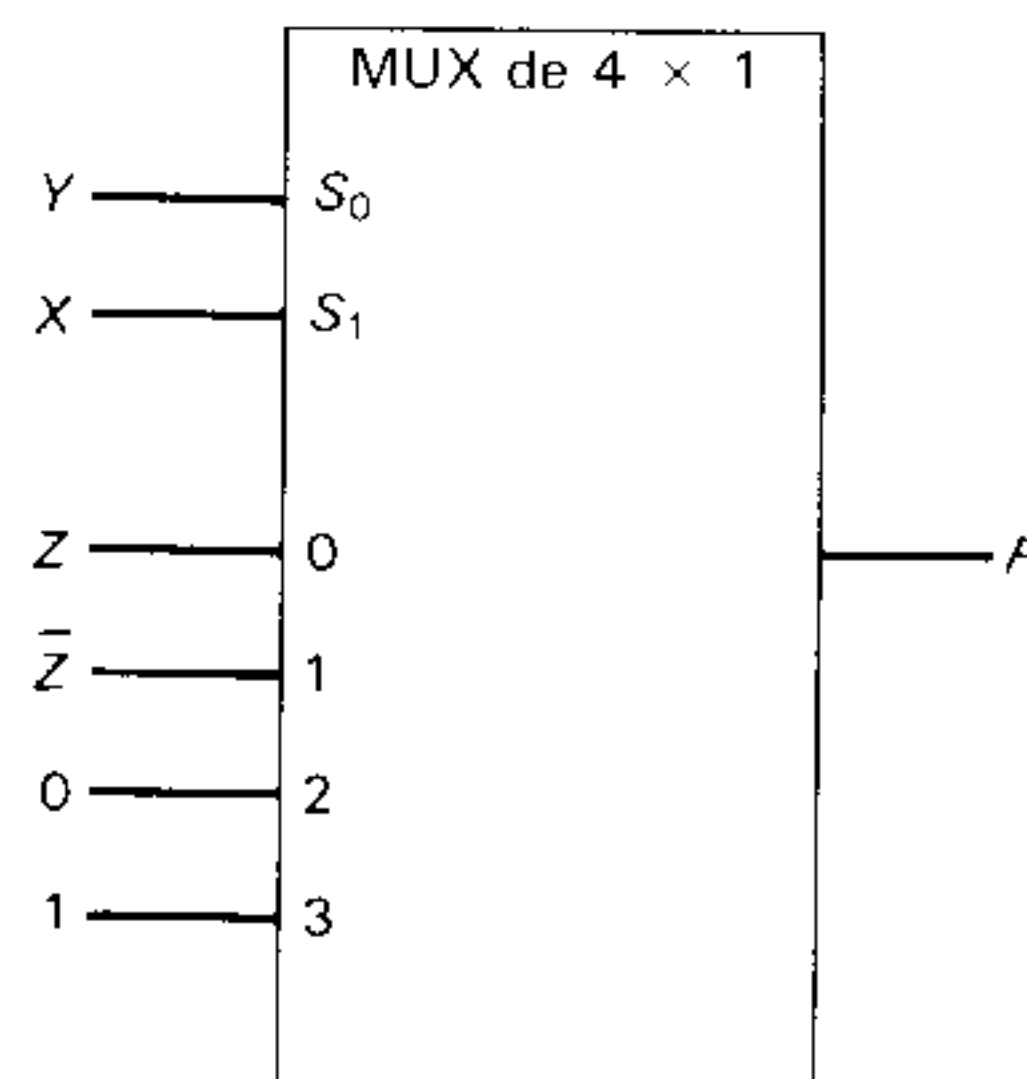
Ahora presentaremos un método más eficiente para ejecutar una función booleana de n variables con un multiplexor que tiene $n - 1$ entradas de selección. Las primeras $n - 1$ variables de la función se conectan a las entradas de selección del multiplexor. La variable que resta de la función se utiliza para las entradas de datos. Si la variable está denotada por Z , las entradas de datos del multiplexor serán $Z, \bar{Z}, 1$ o 0 . Para demostrar este procedimiento, considérese la función booleana de tres variables

$$F(X, Y, Z) = \sum m(1, 2, 6, 7)$$

La función se puede ejecutar con un multiplexor de 4 líneas en 1 línea, como se muestra en la figura 3-25. Las variables X y Y se aplican a las líneas de selección en ese orden; X se conecta a la entrada S_1 y Y a la entrada S_0 . Los valores de las líneas de entrada de datos se determinan de la tabla de verdad de la función. Cuando $XY = 00$, la salida F es igual a Z puesto que $F = 0$ cuando $Z = 0$ y $F = 1$ cuando $Z = 1$. Esto requiere que se aplique la variable Z a la entrada de datos 0. La operación del multiplexor es tal que cuando $XY = 0$, la entrada de datos 0 tiene una trayectoria hacia la salida que hace que F sea igual a Z . En forma semejante podemos determinar la entrada requerida a las líneas 1, 2 y 3 a partir del valor de F cuando $XY = 01, 10$ y 11 , respectivamente. Este ejemplo en particular muestra las cuatro posibilidades que se pueden obtener para las entradas de datos.

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

(a) Tabla de verdad



(b) Ejecución con el multiplexor

FIGURA 3-25

Ejecución de una función booleana con un multiplexor

El procedimiento general para ejecutar cualquier función booleana de n variables con un multiplexor con $n - 1$ entradas de selección y 2^{n-1} entradas de datos se deduce del ejemplo anterior. La función booleana se presenta primero en una tabla de verdad. Las primeras $n - 1$ variables de la tabla se aplican a las entradas de selección del multiplexor. Para cada combinación de las variables de selección, evaluamos la salida como función de la última variable. Esta puede ser 0, 1, la variable o el complemento de la variable. Después se aplican estos valores a las entradas de datos en el orden adecuado. Para poner un segundo ejemplo, considérese la ejecución de la función booleana que sigue:

$$F(A, B, C, D) = \Sigma m(1, 3, 4, 11, 12, 13, 14, 15)$$

Esta se ejecuta con un multiplexor con tres entradas de selección como se muestra en la figura 3-26. Nótese que la primera variable A debe conectarse a la entrada de selección S_2 de tal suerte que A, B, C correspondan a las entradas de selección S_2, S_1 y S_0 , respectivamente. Los valores de las entradas de datos se determinan a partir de la tabla de verdad que se presenta en la figura. El número de la línea de datos correspondiente se determina a partir de la combinación binaria de ABC . Por ejemplo, cuando $ABC = 101$, la tabla muestra que $F = D$; de manera que la variable de entrada D se aplica a la entrada de datos 5. Las constantes binarias 0 y 1 corresponden a dos valores de señales. Si se utilizan circuitos integrados del tipo 1 lógico, entonces el 0 lógico corresponde a la tierra de la señal y el 1 lógico es equivalente a una señal de 5 voltios.

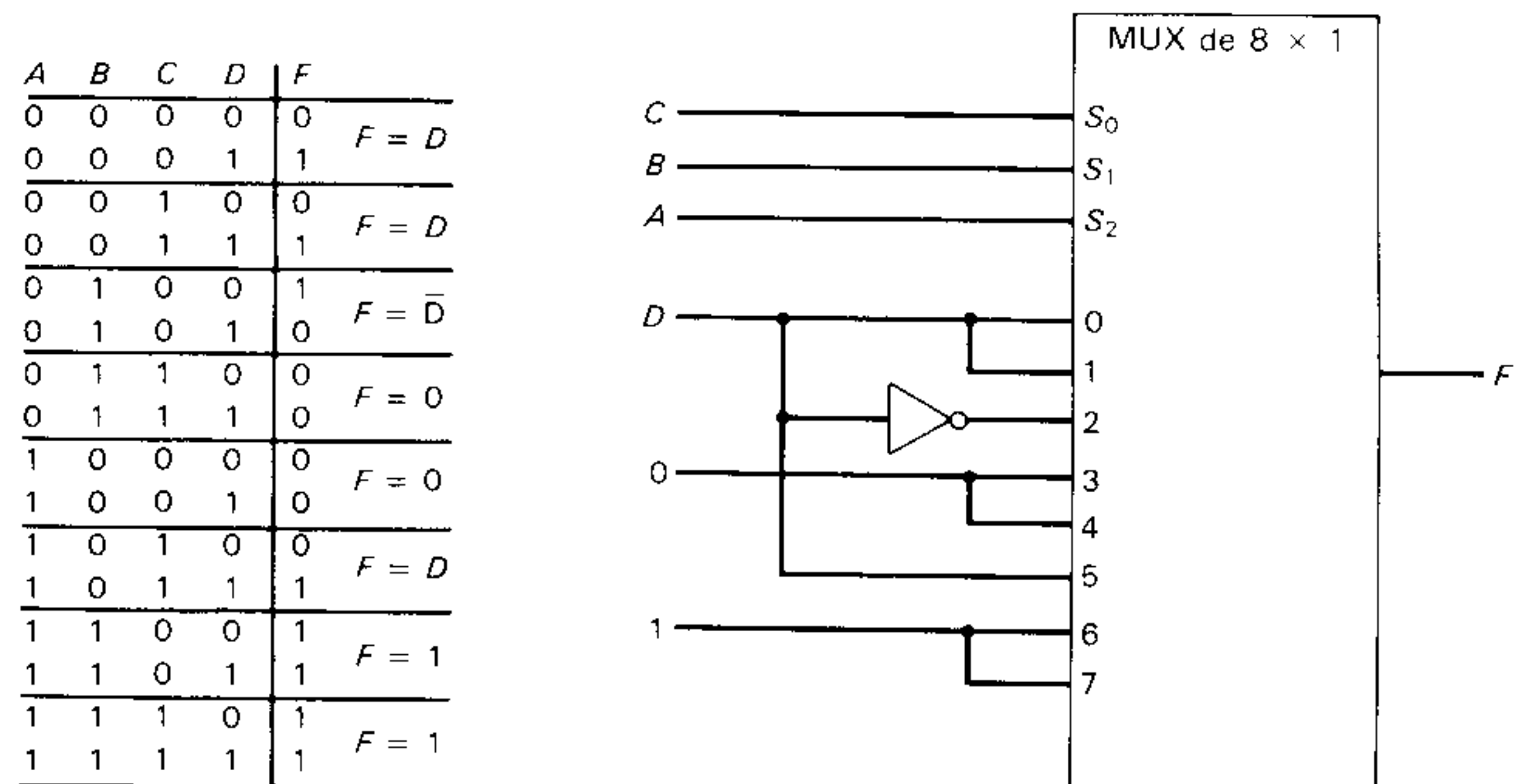


FIGURA 3—26

Ejecución de una función de 4 entradas con un multiplexor

Demultiplexor

Un demultiplexor (o demulticanalizador) es una función digital que realiza la operación inversa de un multiplexor, pues recibe información de una línea y la transmite a una de las 2^n líneas de salida posibles. La selección de la salida específica es controlada por la combinación de bits de n líneas de selección. En la figura 3-27 se muestra

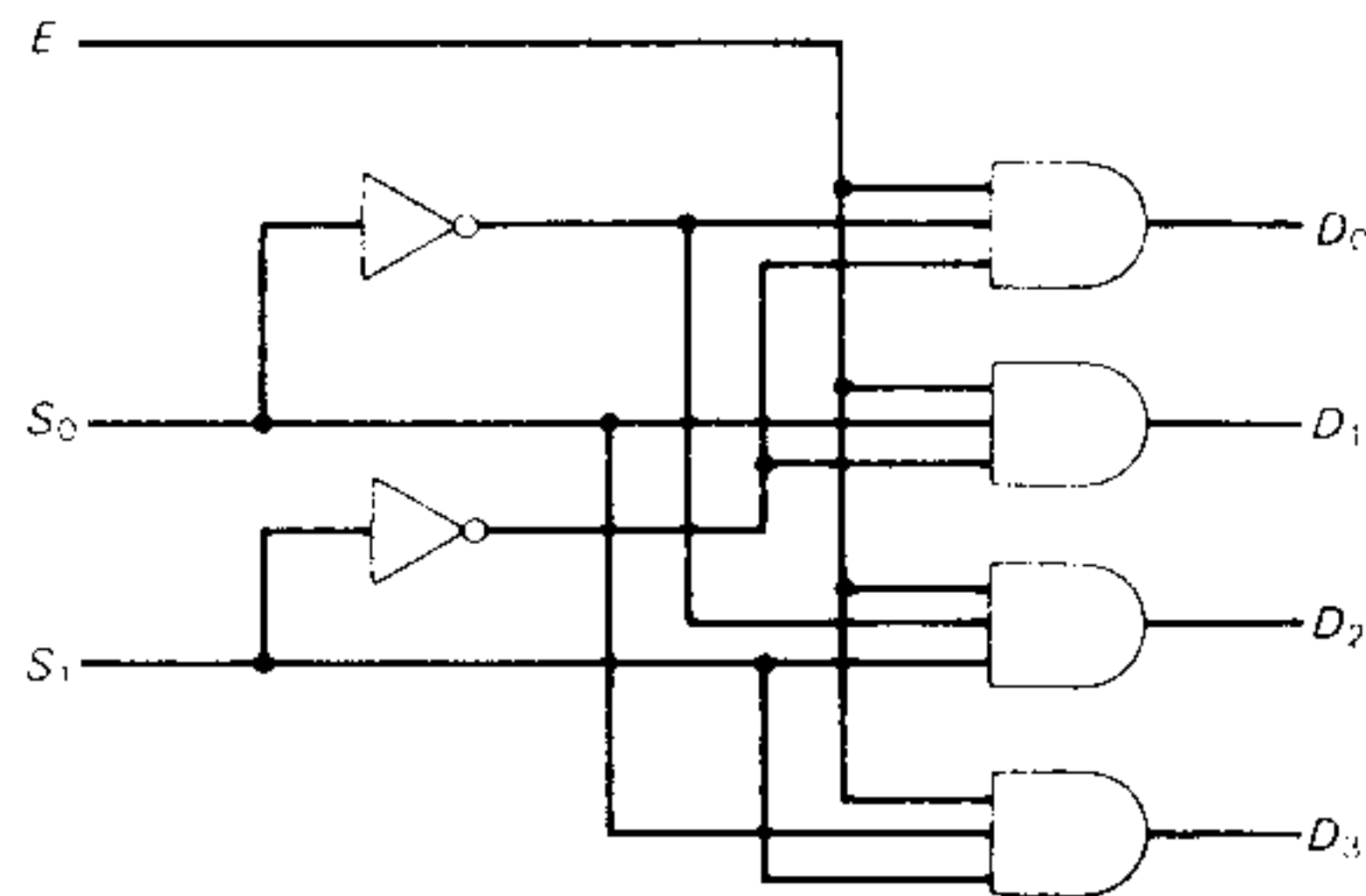


FIGURA 3—27

Demultiplexor de 1 línea en 4 líneas

un demultiplexor de 1 línea en 4 líneas. La entrada de datos E tiene una trayectoria hacia las cuatro salidas, pero la información de entrada va dirigida sólo a una de las salidas, según lo especifican las líneas de selección S_1 y S_0 . Por ejemplo, si $S_1S_0 = 10$, la salida D_2 será la misma que el valor de entrada de E , mientras que todas las otras salidas se mantienen inactivas en el 0 lógico.

Una inspección detenida del circuito demultiplexor indicará que es idéntico a un decodificador de 2 líneas en 4 líneas con entrada habilitadora. Para el decodificador, las entradas de datos son S_1 y S_0 y la habilitadora es la entrada E . En el caso del demultiplexor, la entrada E produce los datos mientras que las otras entradas aceptan las variables de selección. Aunque los dos circuitos tienen aplicaciones diferentes, sus diagramas de lógica son exactamente iguales. Por este motivo, un decodificador con entrada habilitadora se conoce como *decodificador/demultiplexor*.

3-8 SIMBOLOS GRAFICOS ESTANDAR

Componentes digitales que se utilizan con frecuencia como sumadores, decodificadores y multiplexores se consiguen en el mercado en circuitos integrados y se clasifican como funciones MSI. Se han desarrollado símbolos gráficos estándar para éstos y otros componentes de manera que el usuario pueda reconocer cada función por el símbolo gráfico único asignado a ella. Este estándar, conocido como ANSI/IEEE Std. 91-1984, ha sido aprobado por la industria, el gobierno y las organizaciones profesionales, y es compatible con estándares internacionales.

El estándar se vale de una figura rectangular que representa cada función lógica en particular. Dentro de la figura, hay un símbolo calificador general que denota la operación lógica que realiza la unidad. Por ejemplo, el símbolo calificador general de un multiplexor es MUX. El tamaño de la figura es arbitrario y puede ser cuadrada o rectangular con razón arbitraria de largo a ancho. Las líneas de entrada se colocan del lado izquierdo y las de salida en el derecho. La inversión de la dirección del flujo de señales se debe indicar con flechas. Los símbolos de forma rectangular de las compuertas digitales se presentaron en la figura 2-25.

Un ejemplo de un símbolo gráfico estándar es el sumador paralelo de 4 bits que se presenta en la figura 3-28. El símbolo calificador de un sumador es la letra griega Σ . Las letras elegidas para los operandos aritméticos son P y Q . Los símbolos de agrupación de bits en los dos tipos de entradas y la salida de la suma son el equivalente decimal de los valores de los bits a la potencia de 2. Por lo tanto, la entrada rotulada con 3 corresponde al valor de $2^3 = 8$. El acarreo de entrada está designado por CI y el acarreo de salida por CO .

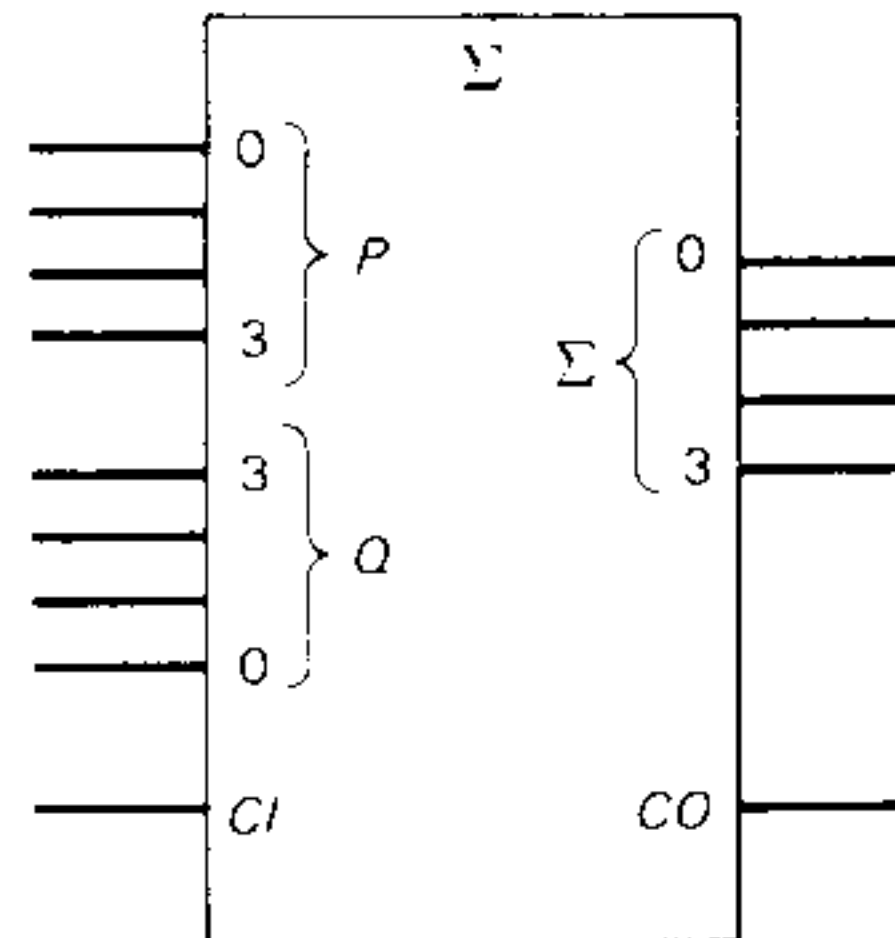


FIGURA 3—28

Símbolo gráfico estándar de un sumador paralelo de 4 bits

Antes de presentar los símbolos gráficos de otros componentes, es necesario repasar algunos términos. Como se mencionó en la sección 2-8, un sistema de lógica positiva define el más positivo de dos niveles de señales (designado por H) como 1 lógico y el nivel más negativo (designado por L) como 0 lógico. La lógica negativa supone la asignación opuesta. Una tercera alternativa es la de emplear una convención de lógica mixta donde las señales se consideran completamente en términos de sus valores de H y L. En cualquier punto del circuito, el usuario puede definir la polaridad lógica asignando un 1 lógico a la señal H o a la L. La notación de lógica mixta utiliza un símbolo gráfico de un triángulo rectángulo pequeño para designar una polaridad de lógica negativa en cualquier terminal de entrada o de salida. (Véase la figura 2-42f.)

Los fabricantes de circuitos integrados especifican la operación de circuitos integrados en términos de señales H y L. Cuando se considera una entrada o una salida en términos de lógica positiva, ellas se definen como *activas altas*. Cuando se consideran en términos de lógica negativa, se supone que son *activas bajas*. Las entradas o salidas activas bajas se reconocen por la presencia del símbolo gráfico indicador de la polaridad, que es un triángulo pequeño. Cuando se utiliza exclusivamente la lógica positiva en todo el sistema, el símbolo de polaridad (triángulo pequeño) es equivalente al círculo pequeño que designa una negación. En este libro suponemos la lógica positiva en todo momento, y emplearemos el círculo pequeño cuando tracemos diagramas de lógica. Cuando una línea de entrada o salida no incluya el círculo pequeño, la definiremos como activa si es 1 lógico. Una línea que incluya el círculo pequeño será considerada activa si se encuentra en el estado 0 lógico. Sin embargo,

utilizaremos el símbolo de polaridad del triángulo pequeño para indicar asignación activa baja en todos los diagramas que representen trazos estándar. Esto coincidirá con manuales de datos de circuitos integrados donde suele utilizarse el símbolo de polaridad. La manera más sencilla de entender lo que se ha afirmado es observar que el símbolo de polaridad del triángulo pequeño es equivalente al símbolo de negación del círculo pequeño cuando se utiliza lógica positiva.

Símbolo gráfico estándar del decodificador

El símbolo gráfico estándar del decodificador se ilustra en la figura 3-29. En (a) tenemos un decodificador de 3 líneas en 8 líneas con una entrada habilitadora designada por *EN*. Las entradas están del lado izquierdo y las salidas del derecho. El símbolo de identificación *X/Y* indica que el circuito convierte el código *X* en código *Y*. A las entradas binarias se les asignan las ponderaciones binarias 1, 2 y 4, que son equivalentes a 2^0 , 2^1 y 2^2 . A las salidas se asignan números del 0 al 7. La suma de las ponderaciones de las entradas, determina la salida que está activa. Por lo tanto, si las dos líneas de entrada con ponderaciones de 1 y 4 son activadas, la ponderación total es $1 + 4 = 5$ y se activa la salida 5. Desde luego, la entrada *EN* debe ser activada para que cualquier salida esté activa. El símbolo de dos letras *EN* está reservado para identificar una entrada que habilita salidas. Si la entrada con el rótulo *EN* está activa, todas las salidas están activas o inactivas de acuerdo con las condiciones de entrada que las afectan. Si la entrada *EN* está inactiva, entonces todas las salidas también están inactivas.

Un decodificador de 2 líneas en 4 líneas con una entrada activa baja y salidas activas bajas se ilustra en la figura 3-29(b). El circuito se habilita si la entrada habilitadora está activa. Esto quiere decir que la señal habilitadora debe estar en un estado de bajo nivel, como lo indica el símbolo de polaridad. Si no se habilita el circuito, todas las salidas están inactivas y en el estado de alto nivel. Cuando se habilita el circuito, las ponderaciones de las entradas determinan qué salida estará activa. Los símbolos de polaridad en las salidas indican que la salida seleccionada estará en un nivel bajo. En el caso de la asignación lógica positiva, las entradas y salidas son como se especifica en la tabla de verdad de la figura 3-17(b).

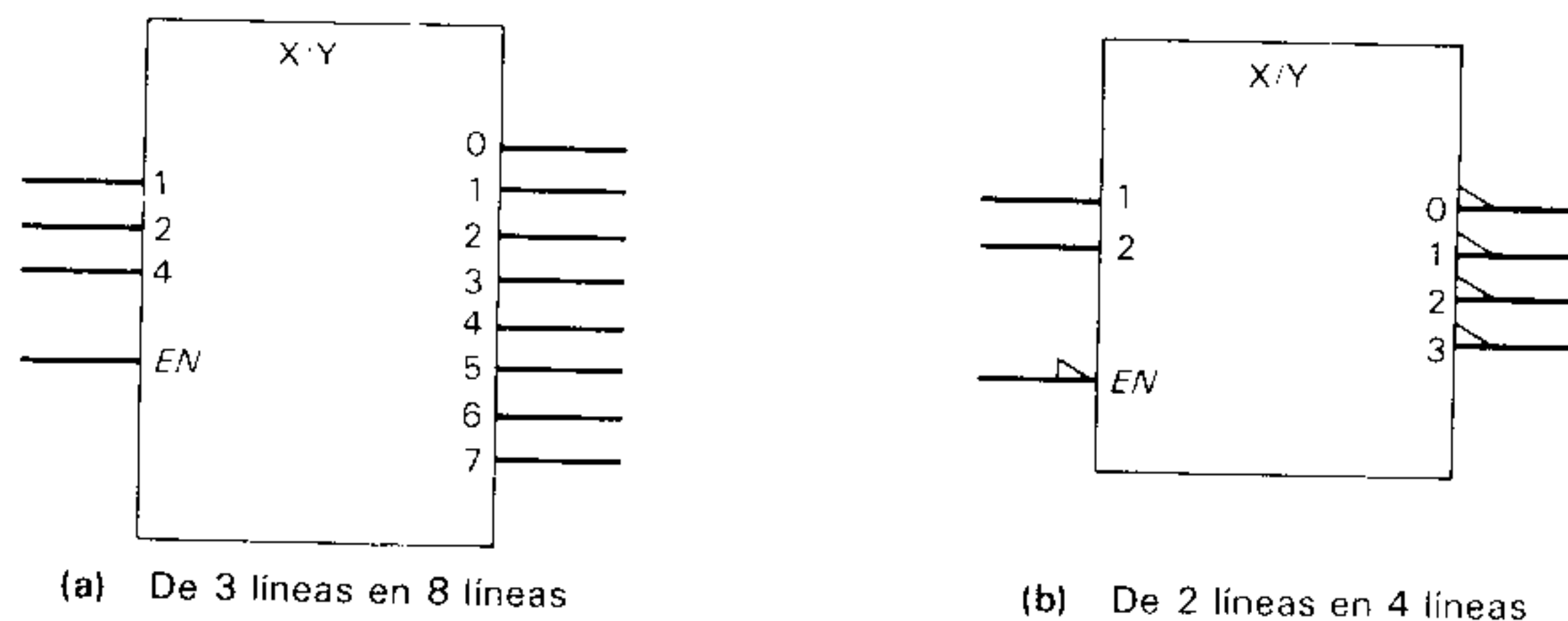


FIGURA 3—29

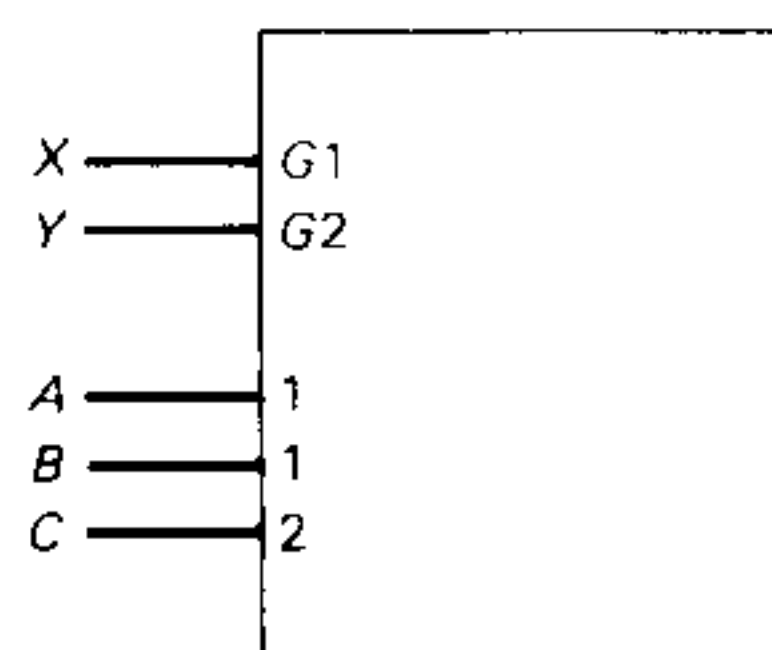
Símbolo gráfico estándar de un decodificador

El decodificador es un caso especial de un componente más general que se conoce como *codificador*. Un codificador es un dispositivo que recibe un código binario de entrada en un número de entradas y produce un código binario diferente en un número de salidas. En vez de utilizar el símbolo calificador X/Y , el codificador se puede especificar por el nombre del código. Por ejemplo el decodificador de 3 líneas en 8 líneas de la figura 3-29(a) puede simbolizarse con el nombre BIN/OCT puesto que el circuito convierte un código binario de tres bits en un valor octal de ocho salidas. Cuando el componente digital representado por la figura es también un circuito integrado comercial, se acostumbra escribir el número de terminal del IC en cada línea de entrada y salida.

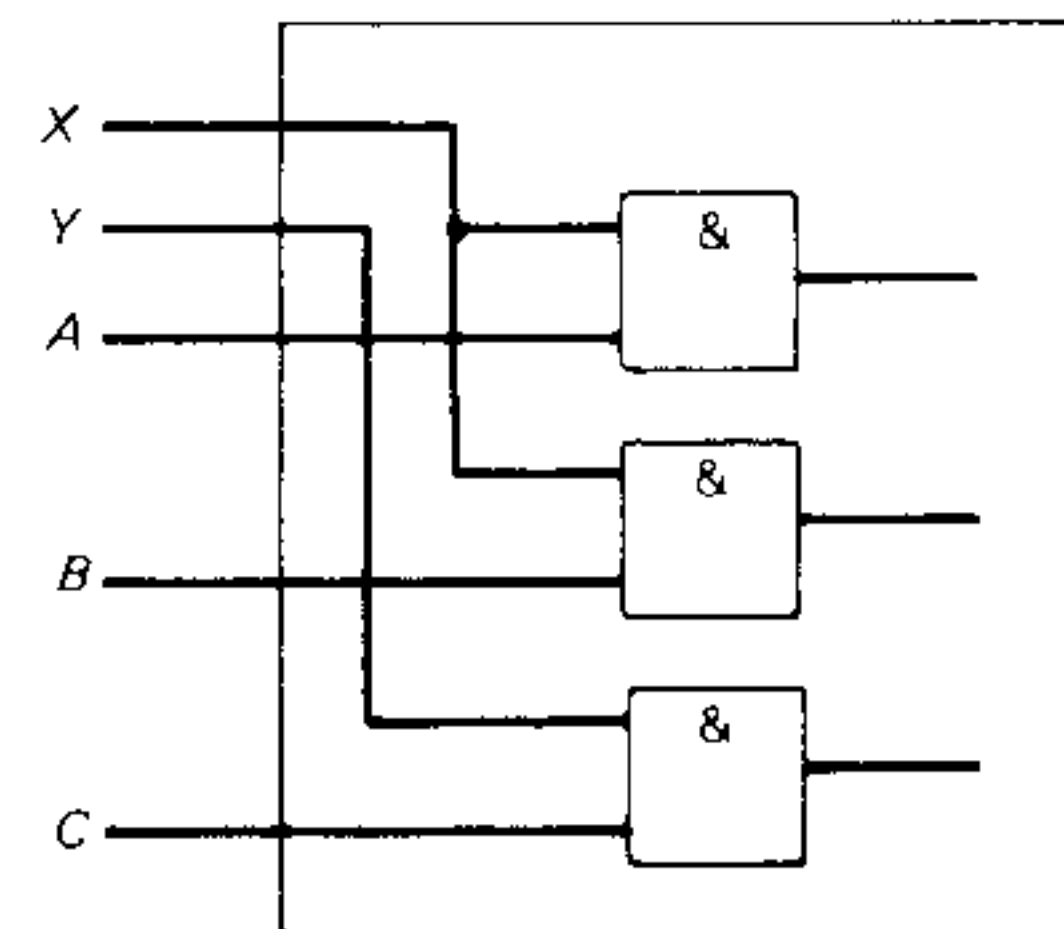
Símbolo gráfico estándar del multiplexor

Antes de presentar el símbolo gráfico de un multiplexor, es necesario definir una notación llamada *dependencia de AND*. La letra G seguida de un número está reservada para especificar dependencia de AND. Cualquier entrada o salida de un diagrama de bloque que esté rotulado con el número asociado con G se considera comparado (con AND) con éste. Por ejemplo, si una entrada del diagrama de bloque tiene el rótulo $G1$ y la otra entrada está rotulada con el número 1, entonces las entradas rotuladas como $G1$ y 1 se consideran comparadas (con AND) en forma interna.

Un ejemplo de dependencia de AND se ilustra en la figura 3-30. En (a) se tiene una parte de un símbolo gráfico con dos rótulos de dependencia de AND, $G1$ y $G2$. Hay dos entradas rotuladas con el número 1 y una entrada rotulada con el número 2. La interpretación equivalente se muestra en la parte (b) de la figura. Nótese que las compuertas AND se representan con el *ampersand* (&) dentro de una figura rectangular para ajustarse a las formas estándar. La entrada X con el rótulo $G1$ se considera comparado (con AND) con las entradas A y B que están rotuladas con un 1. En forma análoga, la entrada Y se compara (con AND) con la entrada C para coincidir con la dependencia entre $G2$ y 2.



(a) Bloque con $G1$ y $G2$



(b) Interpretación equivalente

FIGURA 3—30

Ejemplo de dependencia G (AND)

La dependencia de AND se representa algunas veces con una notación corta como G_7^0 . Este símbolo representa los ocho símbolos de dependencia de AND del 0 al 7, como sigue:

$$G_0, G_1, G_2, G_3, G_4, G_5, G_6, G_7$$

En cualquier momento dado, sólo puede estar activa una de las ocho compuertas AND. La compuerta AND activa está determinada por las entradas asociadas con el símbolo G . Estas entradas se marcan con ponderaciones iguales a las potencias de 2. En el caso de las ocho compuertas AND antes citadas, las ponderaciones son 0, 1 y 2, que corresponden a los números 2^0 , 2^1 y 2^2 , respectivamente. La compuerta AND que está activa en un momento dado cualquiera se determina a partir de la suma de los valores asignados de las entradas activas. Por lo tanto, si las entradas 0 y 2 están activas, la compuerta AND que está activa tiene el número $2^0 + 2^2 = 5$. Esto hace que G_5 esté activa y que las otras siete compuertas AND estén inactivas.

El símbolo gráfico estándar del multiplexor se presenta en la figura 3-31. El rótulo MUX identifica el dispositivo como un multiplexor. Los símbolos contenidos en el bloque son parte de la notación estándar, pero los símbolos marcados afuera son definidos por el usuario. A fin de entender la notación estándar, utilizamos aquí las mismas variables estándar que en los diagramas de lógica de los multiplexores de la sección 3-7.

El diagrama de la figura 3-31(a) representa un multiplexor de 8 líneas en 1 línea con entrada habilitadora (EN). La dependencia de AND está marcada con G_7^0 y está asociada con las entradas incluidas en paréntesis cuadrados. Estas entradas tienen ponderaciones de 0, 1 y 2. Son en realidad lo que hemos llamado entradas de selección. Las ocho entradas de datos están marcadas con números del 0 al 7. La ponderación neta de las entradas activas asociadas con el símbolo G , especifica el número en la entrada de datos que está activa. Por ejemplo, si las entradas de selección $S_2S_1S_0 = 110$, entonces las entradas 1 y 2 asociadas con G están activas. Esto da un

Nota

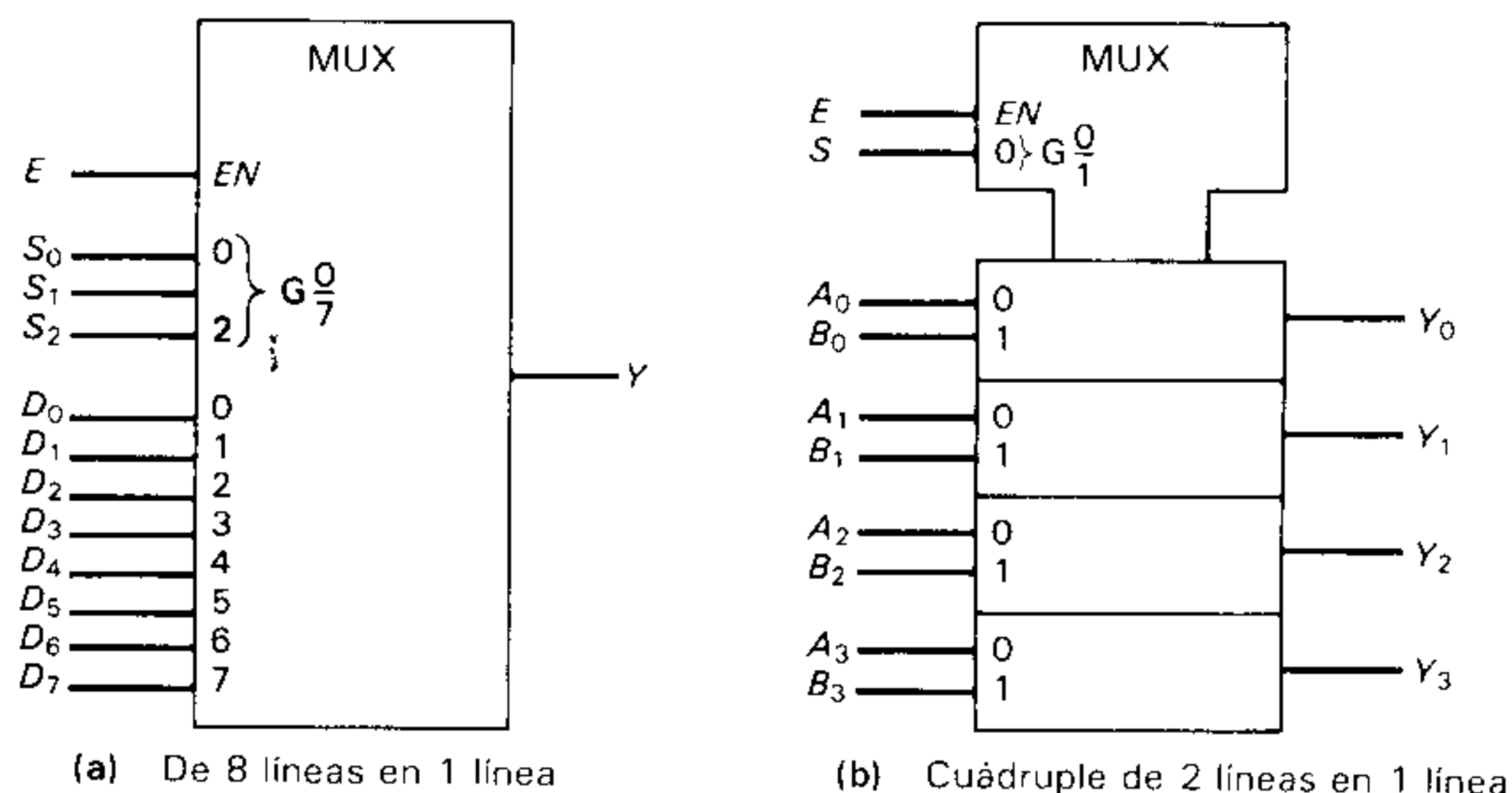


FIGURA 3—31

Símbolos gráficos estándar de multiplexores

valor numérico para la dependencia de AND de $2^2 + 2^1 = 6$, que hace que G_6 esté activo. Como G_6 se compara (con AND) con el número de entrada de datos 6, hace que esta entrada sea activa. Por lo tanto la salida será igual a la entrada de datos D_6 siempre que la entrada habilitadora esté activa.

El diagrama de la figura 3-31(b) representa el multiplexor cuádruple de 2 líneas en 1 línea cuyo diagrama de lógica se muestra en la figura 3-24. Las entradas habilitadora y de selección son comunes a los cuatro multiplexores. Esto se indica en la notación estándar con el cuadro sangrado en la parte superior del diagrama que representa un *bloque de control común*. Las entradas a un bloque de control común controlan todas las secciones inferiores del diagrama. La entrada habilitadora común EN está activa cuando se encuentra en el estado de alto nivel. La dependencia de AND G_1^0 representa dos compuertas AND, G_0 y G_1 . Cuando $S = 0$, G_0 es activa y las entradas marcadas con 0 están activas también. Cuando $S = 1$, G_1 está activa y también lo están las entradas marcadas con 1. Las entradas activas se aplican a las salidas correspondientes si EN está activa.

Notación de dependencia

El aspecto más importante de los símbolos lógicos estándar es la notación de dependencia. Esta se utiliza para ofrecer el medio de representar la relación entre diferentes entradas o salidas sin mostrar en realidad todos los elementos e interconexiones entre ellas. Ya hemos definido la dependencia G (AND) en conexión con el símbolo del multiplexor. El estándar de ANSI/IEEE define otros diez tipos de dependencias. Cada una está denotada por un símbolo de letra (salvo EN). La letra figura en la entrada o salida y va seguida de un número. Cada entrada o salida afectada por esa dependencia está rotulada con ese mismo número.

Las 11 dependencias y sus designaciones alfabéticas correspondientes son las siguientes:

G	—	Representa una relación AND (compuerta)
EN	—	Especifica una acción de habilitación
C	—	Identifica una dependencia de control
S	—	Especifica una acción de inicialización
R	—	Especifica una acción de reinicialización
M	—	Identifica una dependencia de modo
A	—	Identifica una dependencia de dirección
Z	—	Indica una interconexión interna
X	—	Indica una transmisión controlada
V	—	Denota una relación OR
N	—	Denota una relación de negación (OR excluyente).

Las dependencias G y EN se presentaron en esta sección. La dependencia de control C se utiliza para identificar una entrada de reloj en un elemento secuencial y para indicar qué entrada es controlada por ella. Las dependencias de inicialización S y de reinicialización R se usan para especificar estados lógicos internos de un multivibrador biestable (o flip-flop) SR. Las dependencias C , S y R se explican en la sec-

ción 4-3 junto con el circuito multivibrador biestable. La dependencia de modo M sirve para identificar entradas que seleccionan el modo de operación de la unidad. La dependencia de modo se presenta en el capítulo 5 junto con registros y contadores. La dependencia de dirección A se utiliza para identificar la entrada de dirección de una memoria. Esta se presenta en el capítulo 6 junto con la unidad de memoria.

La dependencia Z se emplea para indicar interconexiones dentro de la unidad. Se traduce en la existencia de conexiones lógicas internas entre entradas, salidas, entradas internas y salidas internas, en cualquier combinación. La dependencia X se usa para señalar la trayectoria de transmisión controlada en una compuerta de transmisión semejante a la que se muestra en la figura 2-37.

Las dependencias V y N se utilizan para denotar las relaciones booleanas de operaciones OR y OR excluyente semejantes al símbolo G que representa la operación AND booleana. Un ejemplo de un símbolo gráfico que utiliza la dependencia N se muestra en la figura 3-32. Este es el símbolo del circuito sumador-restador de la figura 3-12. Los operandos aritméticos son P y Q como en el sumador. La entrada S está rotulada con $N4$ y las cuatro entradas Q tienen el rótulo numérico 4. La notación de dependencia implica aquí la existencia de la relación OR excluyente entre la entrada S y cada una de las entradas B . Esto significa que cuando $S = 0$, el operando Q es igual a las entradas B ; pero cuando $S = 1$, las cuatro entradas son negadas por una compuerta OR excluyente y Q se convierte en el complemento a 1's de B .

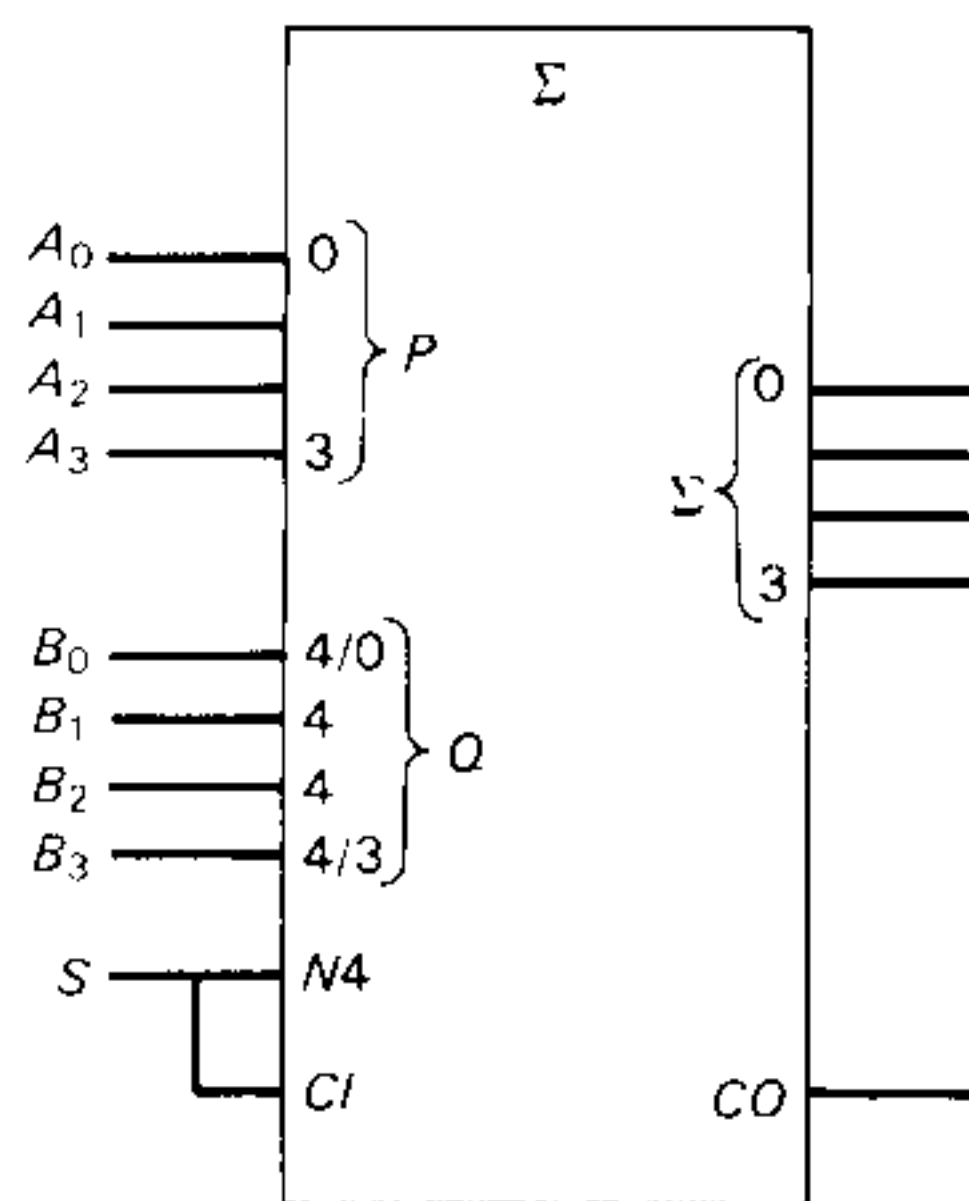


FIGURA 3—32

Símbolo gráfico de un sumador-restador

Símbolos calificativos de entradas y salidas

Además de la notación de dependencia, los símbolos gráficos estándar de ANSI/IEEE para funciones lógicas proporcionan una lista de símbolos calificativos asociados con entradas, salidas y otras conexiones. Estos incluyen a los símbolos de polaridad y negación, símbolos de conexiones internas, símbolos de ubicación fuera de la figura y símbolos de conexiones no lógicas. Toda la lista de símbolos es demasiado grande para incluirla aquí. Algunos de los símbolos de uso más común se citan en la figura 3-33.



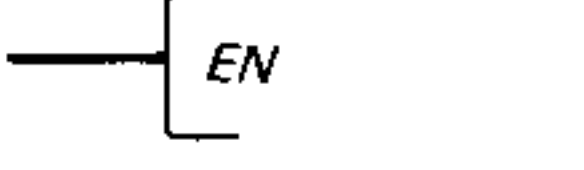
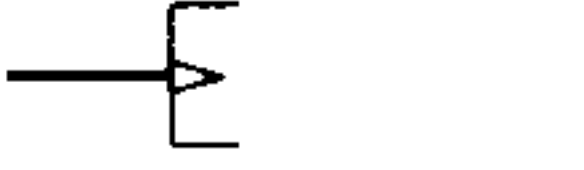
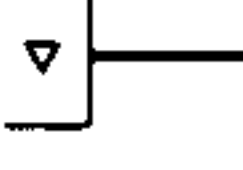

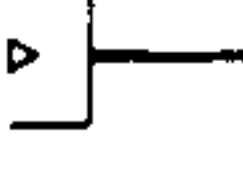
Símbolo	Descripción
	Entrada o salida activas bajas
	Entrada o salida de negación lógica
	Entrada habilitadora: habilita todas las salidas cuando está activa
	Entrada dinámica (véanse la sección 4-3 y la figura 4-12)
	Salida de tres estados (véanse la sección 7-4 y la figura 7-7)
	Salida de circuito abierto
	Salida con amplificación especial

FIGURA 3—33

Símbolos calificadores asociados con entradas y salidas

La entrada o salida activa baja es el indicador de polaridad. Como se dijo antes, éste es equivalente a la negación lógica cuando se supone la lógica positiva. La entrada *EN* es semejante a la dependencia *EN* salvo que no necesita ir seguida de un número. Si no hay un número que le siga, la entrada *EN* simple tiene el efecto de habilitar todas las salidas cuando está activa.

La entrada dinámica está asociada con la entrada de reloj en circuitos flip-flop o multivibradores biestables. Esta indica que la entrada está activa en la transición de una señal de un nivel bajo a uno alto. Esto se explica más a fondo en la sección 4-3. La salida de tres estados tiene un tercer estado externo de alta impedancia que no tiene significado lógico. Esto se explica en la sección 7-4 junto con la construcción de un sistema de bus común.

La salida del circuito abierto se conoce también como salida de colector abierto o de drenaje abierto. Uno de los dos estados lógicos posibles de este tipo de salida corresponde a una condición externa de alta impedancia. A veces se requiere un resistor conectado externamente a fin de producir el nivel lógico adecuado. El símbolo en forma de diamante puede tener una barra encima que represente tipo alto o debajo de él para representar un tipo bajo. El tipo alto o bajo especifica el nivel lógico cuando la salida no está en el estado de alta impedancia. Por ejemplo, los circuitos integrados del tipo TTL tienen una salida especial llamada salida de colector abierto. Estas salidas se reconocen por un símbolo en forma de diamante con una barra debajo de él. Cuando se utilizan como parte de una función de distribución, dos o más salidas de colector abierto conectadas a un resistor común desempeñan una función AND de lógica positiva o una función OR de lógica negativa.

La salida con amplificación especial se utiliza en salidas de compuertas con medios especiales de conducción. Tales compuertas se utilizan en componentes como unidades controladoras de relojes o transmisores orientados por bus.

BIBLIOGRAFIA

1. BLAKESLEE, T. R. *Digital Design with Standard MSI and LSI*. 2a./edic. Nueva York: Wiley, 1979.
2. *High-Speed CMOS Logic Data Book*. Dallas: Texas Instruments, 1984.
3. *IEEE Standard Graphic Symbols for Logic Functions*. (ANSI/IEEE Std. 91-1984). Nueva York: The Institute of Electrical and Electronics Engineers.
4. KAMPEL, I. *A Practical Introduction to the New Logic Symbols*. Boston: Butterworths, 1985.
5. MANO, M. M. *Digital Design*. Englewood Cliffs: Prentice-Hall, 1984.
6. PEATMAN, J. B. *Digital Hardware Design*. Nueva York: McGraw-Hill, 1980.
7. SANDIGE, R. S. *Digital Concepts Using Standard Integrated Circuits*. Nueva York: McGraw-Hill, 1978.
8. *The TTL Data Book*. Vol. 2. Dallas: Texas Instruments, 1985.

PROBLEMAS

- 3-1 Determine las funciones booleanas para las salidas J y K como función de las cuatro entradas del circuito de la figura P3-1.

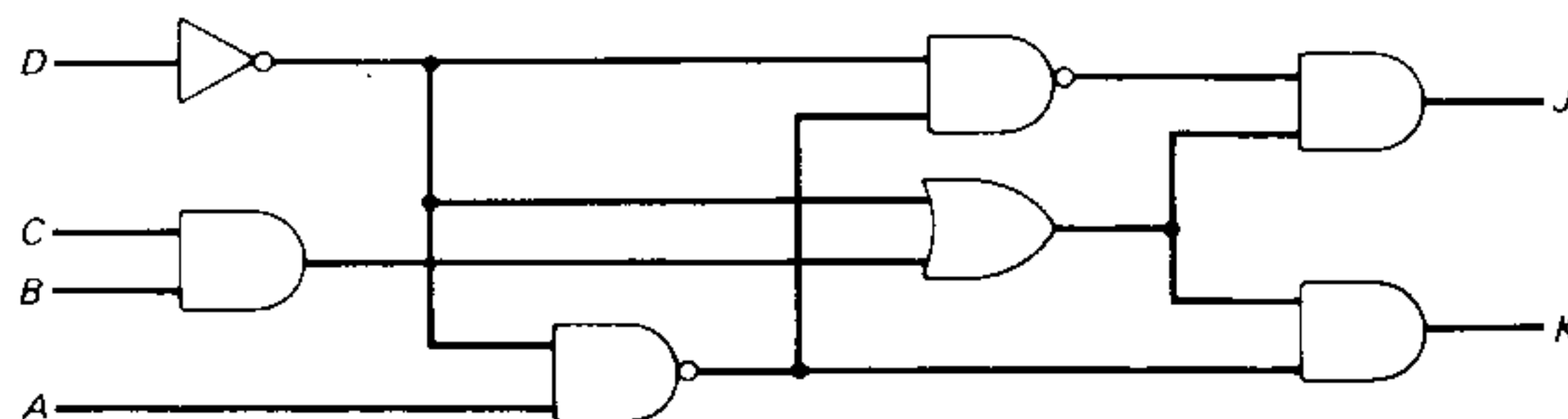


FIGURA P3—1

Circuito del problema 3—1

- 3-2 Obtenga la tabla de verdad del circuito que se ilustra en la figura P3-2. Trace un circuito equivalente de F con menos compuertas NAND.

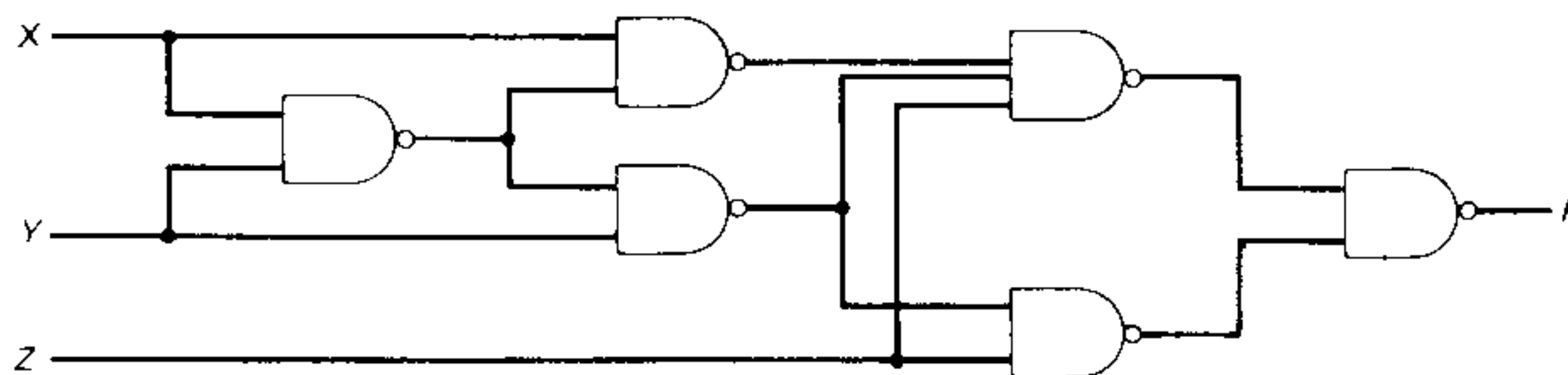


FIGURA P3—2

Circuito del problema 3—2

- 3-3 Verifique que el circuito de la figura P3-3 genere la función NOR excluyente.

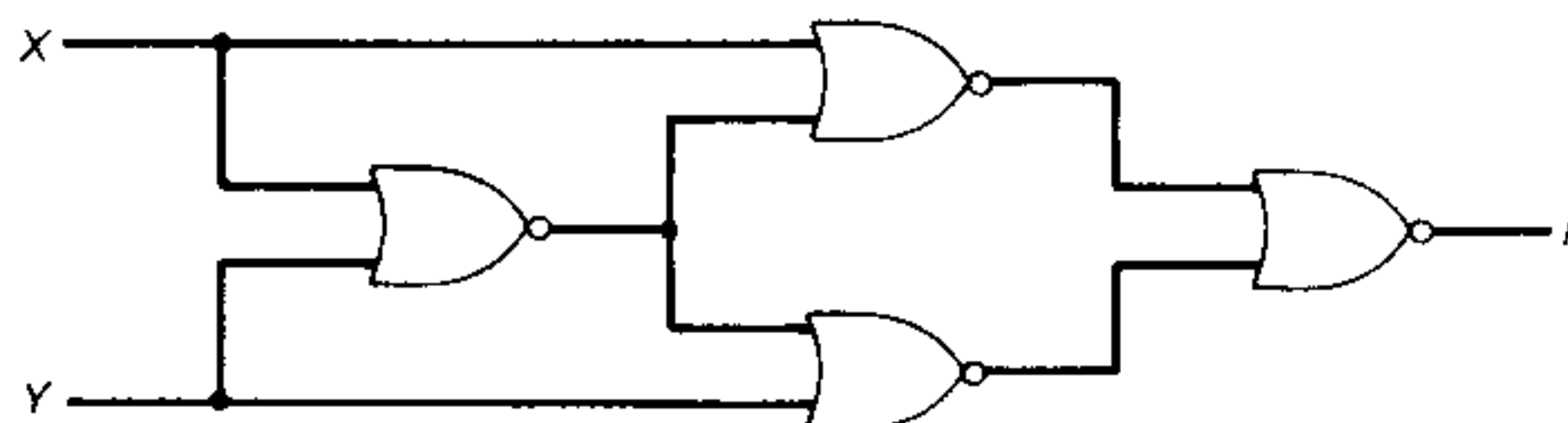


FIGURA P3—3

Circuito del problema 3-3

-
- The logic diagram shows a 1-bit full adder circuit. It has three inputs: A_0 , B_0 , and C_0 . The circuit uses several logic gates: two AND gates for the first stage, two NOT gates, and two more AND gates for the second stage. The outputs are S_0 (the sum) and C_1 (the carry-out). The circuit implements the following logic: $S_0 = A_0 \oplus B_0 \oplus C_0$ and $C_1 = (A_0 \wedge B_0) \vee (A_0 \wedge C_0) \vee (B_0 \wedge C_0)$.

FIGURA P3—13
Circuito del problema 3-13

- 3-14** Construya un convertidor de código BCD a exceso 3 con un sumador de cuatro bits. Recuerde que el dígito del código de exceso 3 se obtiene sumando tres al dígito BCD correspondiente. ¿Qué debe hacerse para cambiar el circuito por un convertidor de código de exceso 3 a BCD?
- 3-15** El circuito sumador-restador de la figura 3-12 tiene los valores siguientes para la selección de entrada S y las entradas de datos A y B . En cada caso, determine los valores de las salidas S_3 , S_2 , S_1 , S_0 y C_4 .

	<u>S</u>	<u>A</u>	<u>B</u>
(a)	0	0111	0110
(b)	0	1000	1001
(c)	1	1100	1000
(d)	1	0101	1010
(e)	1	0000	0001

- 3-16** ¿Cuántas combinaciones de entradas no importa hay en un sumador BCD?
- 3-17** Diseñe un circuito combinatorio que genere el complemento a 9's de un dígito BCD.
- 3-18** Construya un sumador-restador BCD utilizando el sumador BCD de la figura 3-13 y el complemento a 9's del problema 3-17. Utilice diagramas de bloque para ambos componentes mostrando únicamente entradas y salidas.
- 3-19** Es necesario diseñar un sumador decimal para dos dígitos representados en el código de exceso 3. Demuestre que la corrección después de sumar los dos dígitos con un sumador binario de cuatro bits es como sigue:
- (a) El acarreo de salida es igual al acarreo del sumador binario.
- (b) Si el acarreo de salida = 1, entonces sume 0011.
- (c) Si el acarreo de salida = 0, entonces sume 1101.

Construya el sumador decimal con dos sumadores de cuatro bits y un inversor.

- 3-20** Diseñe un multiplicador binario que multiplique dos números de cuatro bits. Utilice compuertas AND y sumadores binarios.
- 3-21** Diseñe un circuito combinatorio que compare dos números de cuatro bits A y B para verificar si son iguales. El circuito tiene una salida X , de manera que $X = 1$ si $A = B$ y $X = 0$ si A no es igual a B .
- 3-22** Diseñe un decodificador de BCD a decimal utilizando las combinaciones no usadas del código BCD como condiciones no importa.
- 3-23** Construya un decodificador de 5 líneas en 32 líneas con cuatro decodificadores de 3 líneas en 8 líneas con entrada habilitadora y un decodificador de 2 líneas en 4 líneas. Utilice diagramas gráficos estándar.
- 3-24** Un circuito combinatorio se define a través de las tres funciones booleanas que siguen. Diseñe el circuito con un decodificador y compuertas externas.

$$F_1 = \overline{X}\overline{Y} + XY\overline{Z}$$

$$F_2 = \overline{X} + Y$$

$$F_3 = XY + \overline{X}\overline{Y}$$

- 3-25** Un circuito combinatorio se especifica por medio de las tres funciones booleanas siguientes. Diseñe el circuito con un decodificador construido con compuertas NAND (semejante a la figura 3-17) y compuertas NAND externas.

$$F_1(A, B, C) = \Sigma m(0, 5, 7)$$

$$F_2(A, B, C) = \Sigma m(2, 3, 4)$$

$$F_3(A, B, C) = \Sigma m(1, 6, 7)$$

- 3-26** Trace el diagrama de lógica de un decodificador de 2 líneas en 4 líneas sólo con compuertas NOR. Incluya una entrada habilitadora.
- 3-27** Diseñe un codificador de prioridad de cuatro entradas con entradas y salidas como en la tabla 3-8, pero que la entrada D_0 tenga la más alta prioridad.
- 3-28** Obtenga la tabla de verdad de un codificador de prioridad de octal a binario.
- 3-29** Construya un multiplexor de 8 líneas en 1 línea con entrada habilitadora utilizando compuertas de transmisión.
- 3-30** Construya un multiplexor de 16 líneas en 1 línea con dos multiplexores de 8 líneas en 1 línea y uno de 2 líneas en 1 línea. Utilice símbolos estándar de diagramas de bloque.
- 3-31** Diseñe un sumador completo con un multiplexor dual de 4 líneas en 1 línea.
- 3-32** Diseñe la siguiente función booleana con un multiplexor.

$$F(A, B, C, D) = \Sigma m(0, 3, 5, 6, 8, 9, 14, 15)$$

- 3-33** Ejecute la función booleana definida en la tabla de verdad de la figura 3-26 con un multiplexor de 4 líneas en 1 línea y compuertas externas. Conecte las entradas A y B a las líneas de selección. Los requisitos de entrada de las cuatro líneas de datos serán función de las variables C y D . Estos valores se obtienen expresando F como función de C y D en cada uno de los cuatro casos cuando $AB = 00, 01, 10$ y 11 . Estas funciones pueden tener que ejecutarse con compuertas externas.
- 3-34** Reordene la tabla de verdad del circuito de la figura 3-17 y verifique que pueda funcionar como demultiplexor.
- 3-35** Trace el símbolo gráfico de un decodificador de BCD a decimal con entrada habilitadora. Esto es, semejante a la figura 3-29 pero con cuatro entradas de datos y diez salidas.
- 3-36** Trace el símbolo gráfico de un decodificador de binario en octal con tres entradas habilitadoras, E_1 , E_2 y E_3 . El circuito se habilita si $E_1 = 1$, $E_2 = 0$ y $E_3 = 0$ (suponiendo lógica positiva). El nombre del símbolo calificador del decodificador es BIN/OCT. El símbolo habilitador EN se coloca en la salida de una compuerta AND con entradas E_1 , E_2 y E_3 . El símbolo rectangular de la compuerta AND se coloca dentro de la figura rectangular del decodificador (este es igual al circuito integrado 74138 del tipo TTL).
- 3-37** Trace el diagrama de símbolo gráfico de un multiplexor dual de 4 líneas en línea con entradas habilitadoras separadas y entradas de selección comunes (este es igual al circuito integrado 74153 del tipo TTL).
- 3-38** Defina con sus palabras:
- | | |
|---------------------------------|------------------------------|
| (a) Lógica positiva y negativa. | (d) Bloque de control común. |
| (b) Activo alto y activo bajo. | (e) Notación de dependencia. |
| (c) Indicador de polaridad. | |
- 3-39** Dé un ejemplo de un símbolo gráfico que tenga las tres dependencias booleanas G , V y N . Trace la interpretación equivalente.