

**FIGURA 4-23**  
Codificador de cuatro entradas con prioridad

Los mapas para simplificar las salidas  $x$  y  $y$  aparecen en la figura 4-22. Los minitérminos para las dos funciones se deducen a partir de la tabla 4-8. Aunque la tabla sólo tiene cinco filas, al sustituir cada X de una fila, primero por 0 y después por 1, obtendremos las 16 posibles combinaciones de entrada. Por ejemplo, la cuarta fila de la tabla, que tiene XX10, representa los cuatro minitérminos 0010, 0110, 1010 y 1110. Las expresiones booleanas simplificadas para el codificador con prioridad se obtienen de los mapas. La condición para la salida  $V$  es una función OR de todas las variables de entrada. El codificador con prioridad se implementa en la figura 4-23 de acuerdo con las funciones booleanas siguientes:

$$x = D_2 + D_3$$

$$y = D_3 + D_1 D_2'$$

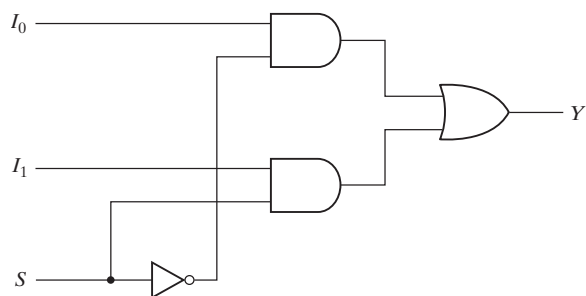
$$V = D_0 + D_1 + D_2 + D_3$$

## 4-10 MULTIPLEXORES

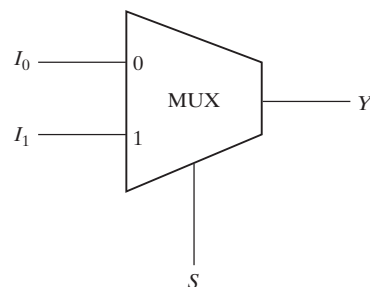
Un multiplexor es un circuito combinacional que selecciona información binaria de una de muchas líneas de entrada y la envía a una sola línea de salida. La selección de una línea de entrada dada se controla con un conjunto de líneas de selección. Normalmente, hay  $2^n$  líneas de entrada y  $n$  líneas de selección cuyas combinaciones de bits determinan cuál entrada se selecciona.

Un multiplexor de 2 líneas a 1 conecta una de dos fuentes de un bit a un destino común, como se indica en la figura 4-24. El circuito tiene dos líneas de entrada de datos, una línea de salida y una línea de selección  $S$ . Cuando  $S = 0$ , se habilita la compuerta AND de arriba e  $I_0$  cuenta con una trayectoria hacia la salida. Cuando  $S = 1$ , la compuerta AND inferior está habilitada e  $I_1$  tiene una trayectoria hacia la salida. El multiplexor actúa como un interruptor electrónico que selecciona una de dos fuentes. El diagrama de bloques de un multiplexor a veces se representa con un símbolo en forma de cuña, como en la figura 4-24b). Esto sugiere visualmente cómo una fuente de datos, seleccionada de entre varias, se dirige a un solo destino. En los diagramas de bloques es común rotular los multiplexores como MUX.

En la figura 4-25 se presenta un multiplexor de 4 líneas a 1. Cada una de las cuatro entradas,  $I_0$  a  $I_3$ , 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 determinada. Las salidas de las compuertas AND se aplican a una sola compuerta OR que genera la salida de una sola línea. La tabla de la función indica qué entrada se pasa a la salida con cada combinación de los valores binarios de selección. Para ilustrar el funcionamiento del circuito, consideremos el caso en que

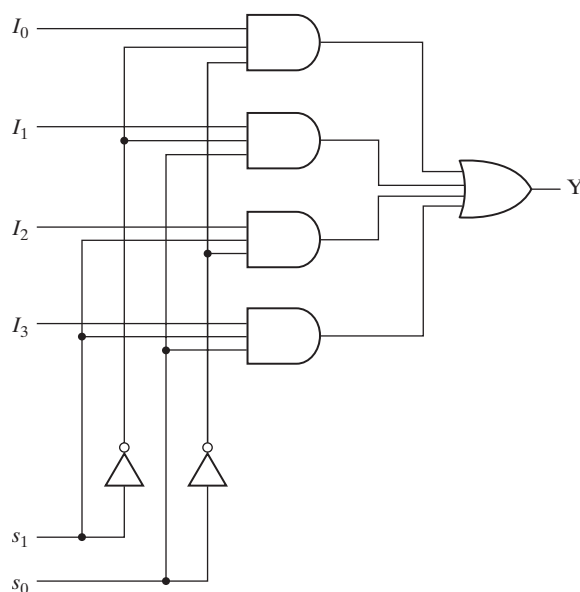


a) Diagrama lógico



b) Diagrama de bloque

**FIGURA 4-24**  
Multiplexor de 2 líneas a 1



a) Diagrama lógico

$s_1$	$s_0$	$Y$
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$

b) Tabla de función

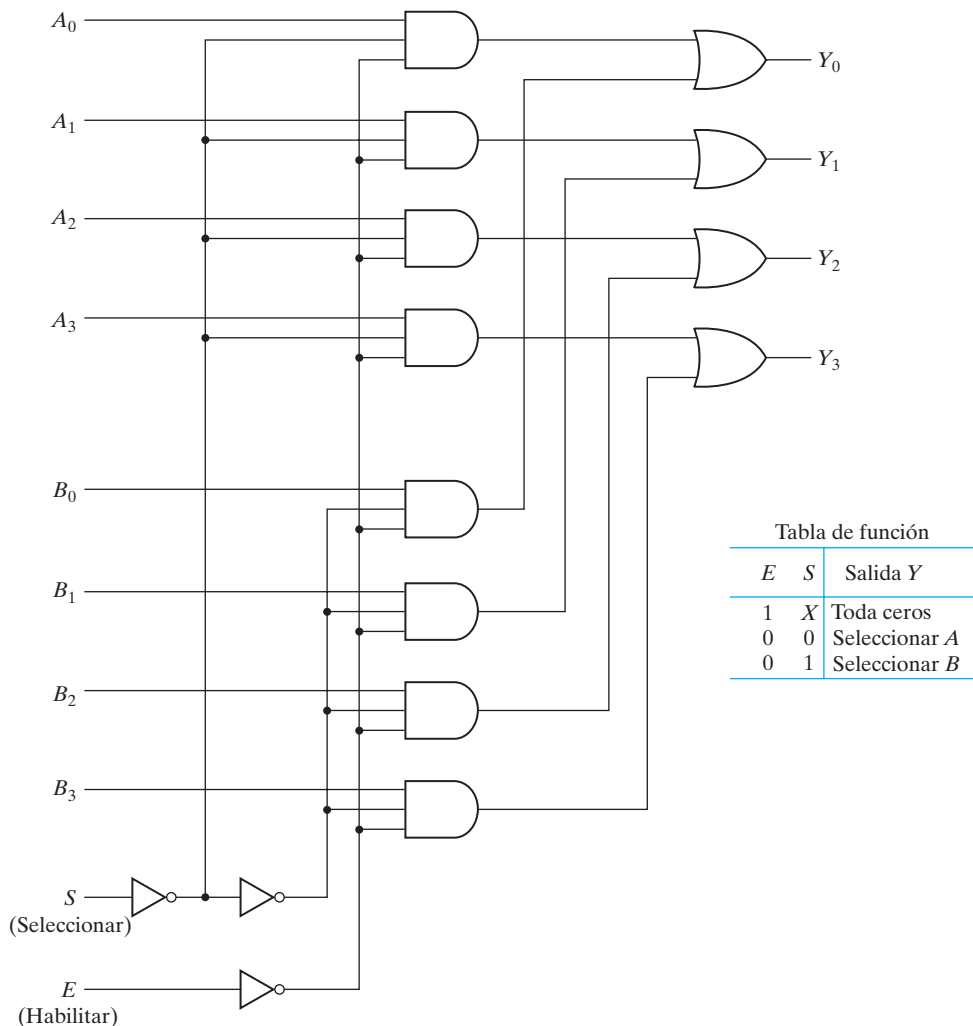
**FIGURA 4-25**  
Multiplexor de 4 líneas a 1

$S_1 S_0 = 10$ . La compuerta AND asociada a la entrada  $I_2$  tiene 1 en dos de sus entradas, y la tercera conectada a  $I_2$ . Las otras tres compuertas AND tienen 0 en por lo menos una de sus entradas, lo que hace que produzcan 0 como salida. Así, la salida de la compuerta OR tiene el mismo valor que  $I_2$ , así que constituye un camino de la entrada seleccionada hasta la salida. Los multiplexores también se denominan *selectores de datos*, pues seleccionan una de varias entradas y dirigen la información binaria a la línea de salida.

Las compuertas AND y los inversores del multiplexor semejan un circuito decodificador y, de hecho, decodifican las líneas de selección de entrada. En general, un multiplexor de  $2^n$  líneas

a 1 se construye a partir de un decodificador de  $n$  a  $2^n$  líneas añadiéndole  $2^n$  líneas de entrada, una para cada compuerta AND. Las salidas de las compuertas AND se aplican a una sola compuerta OR. El tamaño del multiplexor se especifica con el número de líneas de entrada de datos que tiene ( $2^n$ ) y la única línea de salida. El número de líneas de selección ( $n$ ) está implícito en el número de líneas de datos ( $2^n$ ). Al igual que los decodificadores, los multiplexores podrían tener una entrada de habilitación que controla el funcionamiento de la unidad. Si dicha entrada está inactiva, las salidas quedarán inhabilitadas, y si está activa, el circuito funcionará como un multiplexor normal.

Los circuitos multiplexores se pueden combinar con entradas de selección comunes para crear una lógica de selección de múltiples bits. Por ejemplo, en la figura 4-26 se ilustra un multiplexor cuádruple de 2 líneas a 1. El circuito tiene cuatro multiplexores, cada uno de los



**FIGURA 4-26**  
Multiplexor cuádruple de 2 líneas a 1

cuales puede seleccionar una de dos líneas de entrada. Podemos escoger que la salida  $Y_0$  provenga de la entrada  $A_0$  o bien de  $B_0$ . De igual manera, la salida  $Y_1$  podría tener el valor de  $A_1$  o  $B_1$ , y así sucesivamente. La línea de selección de entrada  $S$  selecciona una de las líneas en cada uno de los cuatro multiplexores. La entrada de habilitación  $E$  debe estar activa para que el funcionamiento sea normal. Aunque el circuito contiene cuatro multiplexores de 2 líneas a 1, seguramente lo veremos como un circuito que selecciona uno de dos conjuntos de líneas de datos de cuatro bits. Como indica la tabla de función, la unidad se habilita cuando  $E = 0$ . Entonces, si  $S = 0$ , las cuatro entradas  $A$  tendrán un camino a las cuatro salidas. En cambio, si  $S = 1$ , las cuatro entradas  $B$  se aplicarán a las salidas. Cuando  $E = 1$ , todas las salidas tienen 0, sin importar qué valor tenga  $S$ .

## Implementación de funciones booleanas

En la sección 4-8 se explicó cómo utilizar un decodificador para implementar funciones booleanas añadiendo compuertas OR externas. Un examen del diagrama lógico de un multiplexor revela que básicamente es un decodificador con una compuerta OR incluida en la unidad. Los minitérminos de una función se generan en un multiplexor mediante el circuito asociado a las entradas de selección. Los minitérminos individuales se pueden seleccionar con las entradas de datos. Esto ofrece un método para implementar 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.

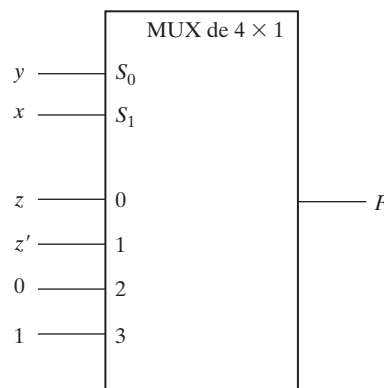
Ahora mostraremos un método más eficiente para implementar una función booleana de  $n$  variables con un multiplexor que tiene  $n - 1$  entradas de selección. Las primeras  $n - 1$  variables de la función se conectan a las entradas de selección del multiplexor. La variable restante de la función se utiliza para las entradas de datos. Si denotamos esa variable con  $z$ , cada entrada de datos del multiplexor será,  $z$ ,  $z'$ , 1 o 0. Para ilustrar este procedimiento, consideremos la función booleana de tres variables:

$$F(x, y, z) = \Sigma(1, 2, 6, 7)$$

La función puede implementarse con un multiplexor de 4 líneas a 1 como se indica en la figura 4-27. Las dos variables  $x$  y  $y$  se aplican a las líneas de selección en ese orden;  $x$  se conecta

$x$	$y$	$z$	$F$
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) Implementación con multiplexor

**FIGURA 4-27**

Implementación de una función booleana con un multiplexor

a la entrada  $S_1$  y y se conecta a  $S_0$ . Los valores de las líneas de entrada de datos se deducen de la tabla de verdad de la función. Cuando  $xy = 00$ , la salida  $F$  es igual a  $z$  porque  $F = 0$  cuando  $z = 0$  y  $F = 1$  cuando  $z = 1$ . Esto requiere aplicar la variable  $z$  a la entrada de datos 0. El funcionamiento del multiplexor es tal que, cuando  $xy = 00$ , la entrada de datos 0 tiene una trayectoria hacia la salida y eso hace que  $F$  sea igual a  $z$ . De forma similar, podemos determinar las entradas que deben recibir las líneas de datos 1, 2 y 3, a partir del valor de  $F$  cuando  $xy = 01, 10$  y  $11$ , respectivamente. Este ejemplo específico muestra las cuatro posibilidades que podemos tener en las entradas de datos.

El procedimiento general para implementar cualquier función booleana de  $n$  variables con un multiplexor de  $n - 1$  entradas de selección y  $2^{n-1}$  entradas de datos se deduce del ejemplo anterior. Primero se enumera la función booleana 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 en función de la última variable. Esta función puede ser 0, 1, la variable o el complemento de la variable. Luego, estos valores se aplican a las entradas de datos en el orden correcto. Como segundo ejemplo, consideremos la implementación de la función booleana

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

Esta función se implementa con un multiplexor con tres entradas de selección, como se ilustra en la figura 4-28. Observe que la primera variable,  $A$ , debe conectarse a la entrada de selección  $S_2$  para que  $A, B$  y  $C$  correspondan a las entradas de selección  $S_2, S_1$  y  $S_0$ , respectivamente. Los valores de las entradas de datos se determinan de la tabla de verdad que se presenta en la figura. El número de línea de datos correspondiente se determina a partir de la combinación binaria de  $ABC$ . Por ejemplo, cuando  $ABC = 101$ , la tabla indica que  $F = D$ , así que se aplica la variable de entrada  $D$  a la entrada de datos 5. Las constantes binarias 0 y 1 corresponden a dos valores de señal fijos. Cuando se usan circuitos integrados, el 0 lógico corresponde a la tierra de señal y el 1 lógico equivale a la señal de potencia, que por lo regular es de 5 volts.

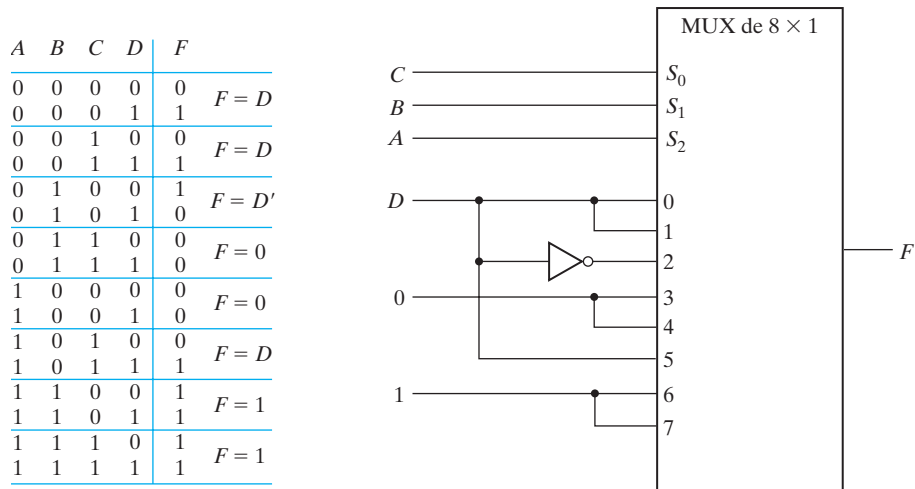


FIGURA 4-28

Implementación de una función de cuatro entradas con un multiplexor