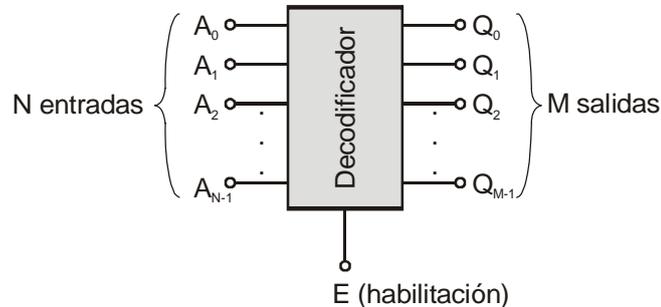




1. CIRCUITOS DECODIFICADORES

Los circuitos decodificadores son circuitos lógicos que poseen un número determinado de entradas y otro número determinado de salidas. Si N es la cantidad de entradas, la cantidad máxima de salidas que podrían existir serían $M = 2^N$. Sin embargo existen muchos decodificadores que presentan menos salidas que las que podrían obtenerse en función de la cantidad de entradas.

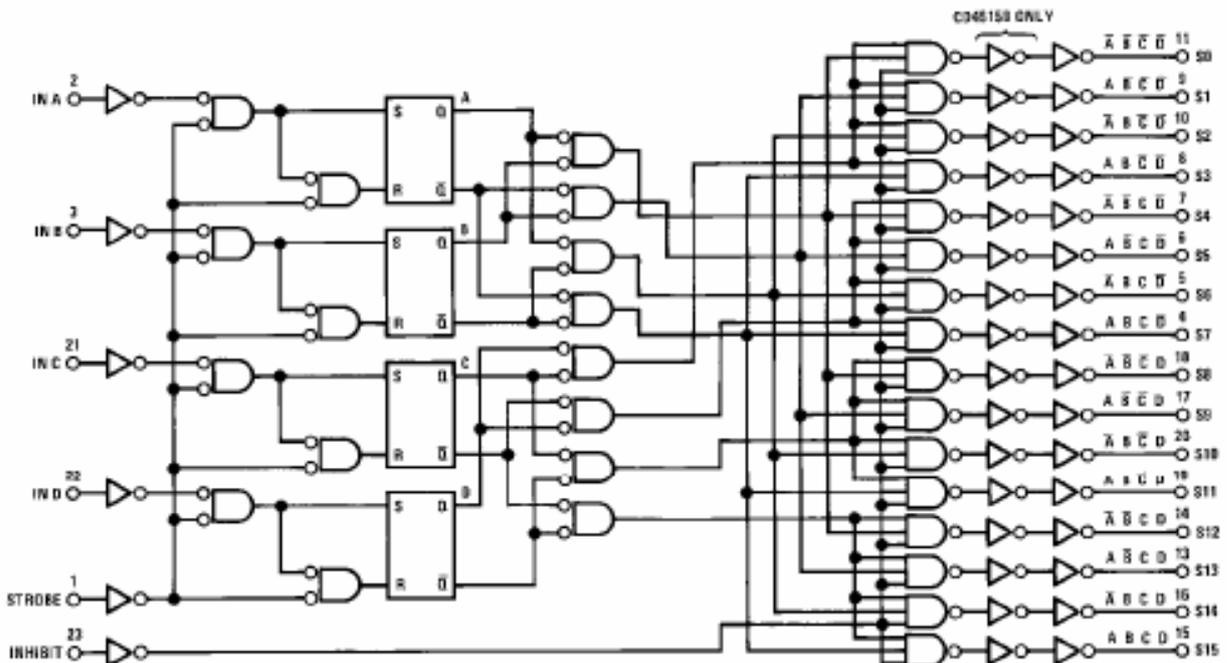
Los decodificadores también suelen poseer una entrada de *habilitación* que habilita o deshabilita el proceso de decodificación.



Veamos algunos decodificadores típicos:

1.1. Decodificador 4 a 16 líneas de 4 bits

Este tipo de decodificador posee 4 bits de entrada, por lo que puede direccionar 16 salidas ($2^4 = 16$). Por cada combinación de las entradas, se activará una y solo una de las salidas. Ejemplos de circuitos integrados comerciales que realizan esta función son los CD4514 y CD4515 ambos de tecnología CMOS. La diferencia entre ambos radica en los niveles de activación de las salidas, en el CD4514 las salidas son activas en **nivel alto** (todas las salidas en 0, menos la activa), y en el CD4515 las salidas son activas en **nivel bajo** (todas las salidas en 1, excepto la activa que pasa 0).

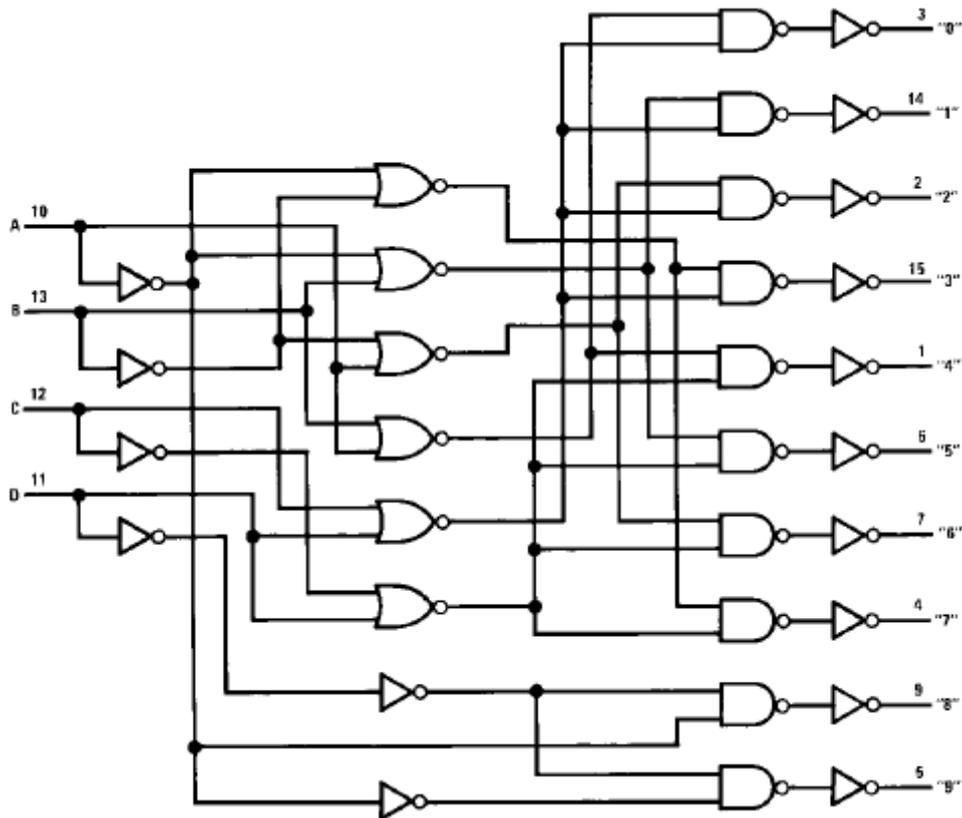


Ver la hoja de datos en archivo pdf.

1.2. Decodificador de BCD a decimal

Este decodificador posee 4 entradas (los 4 bits BCD) y 10 salidas. Según la combinación BCD de entrada, se activará la correspondiente salida. Si la entrada no corresponde a un código BCD válido, el decodificador no activa ninguna de sus salidas.

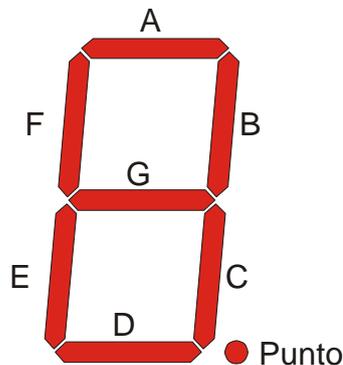
Ejemplo comercial de este decodificador es el CD4028.



Ver la hoja de datos en archivo pdf.

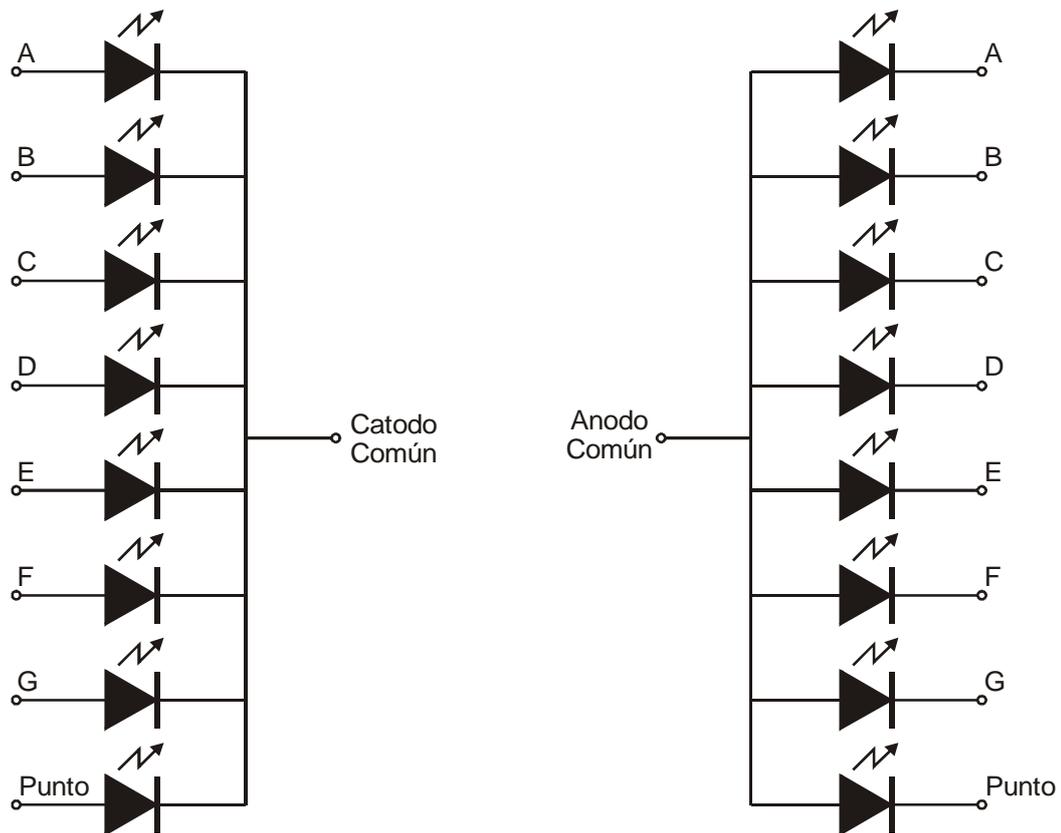
1.3. Decodificador de BCD a 7 segmentos

Este decodificador permite representar un número BCD en un display de 7 segmentos. Un display de 7 segmentos es una matriz de LEDs (diodos emisores de luz) o también de elementos de cristal líquido, ordenados de tal manera que permiten “dibujar” un número decimal. Un display tiene la siguiente forma:

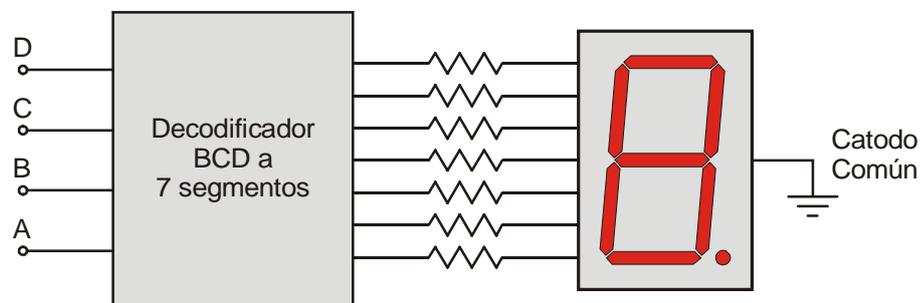


Donde cada segmento está implementado por un led o por un elemento de cristal líquido. Se dice que es un display de 7 segmentos porque justamente posee 7 elementos (sin contar el punto decimal) que permiten “dibujar” un número decimal. Cada segmento se identifica por una letra de la **A** a la **F**, y justamente estas son las salidas del decodificador. Por ejemplo, si la entrada BCD del decodificador fuera **0001**, el decodificador tendría que activar las salidas **B** y **C** para así dibujar el decimal 1. Al igual que el decodificador anterior, si la entrada BCD es inválida, ninguna de las salidas se activaría con lo que no se visualizaría ningún número en el display.

Con respecto a los displays de LEDs, los hay en dos configuraciones: cátodo común y ánodo común. Esto se refiere a cómo se encuentran conectados internamente los leds que conforman el display. En el caso del display cátodo común, todos los cátodos de los leds están unidos y conectados a un pin del display. Análogamente, para los displays de ánodo común, todos los leds tienen unidos sus ánodos y conectados a un pin del display:



En el caso de los displays de leds, ya sean de ánodo o de cátodo común, entre la salida del decodificador y cada segmento del display debe intercalarse una resistencia para limitar la corriente consumida por cada segmento del display, caso contrario podría dañarse tanto el decodificador como el display.

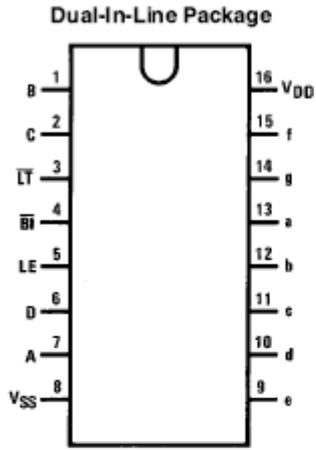


Ejemplos comerciales de estos decodificadores son los integrados CD4511 (para displays de leds) y CD4543 (para displays de cristal líquido).

Ver hojas de datos en archivos pdf.

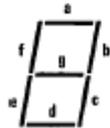


Connection Diagram



Top View
 Order Number CD4511B

Segment Identification



TL/F/5991-3

Truth Table

Inputs							Outputs							
LE	BI	LT	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	B
X	0	1	X	X	X	X	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	0	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	*
0	1	1	1	0	1	1	0	0	0	0	0	0	0	*
0	1	1	1	1	0	0	0	0	0	0	0	0	0	*
0	1	1	1	1	0	1	0	0	0	0	0	0	0	*
0	1	1	1	1	1	0	0	0	0	0	0	0	0	*
0	1	1	1	1	1	1	0	0	0	0	0	0	0	*
1	1	1	X	X	X	X					*			*

X = Don't Care

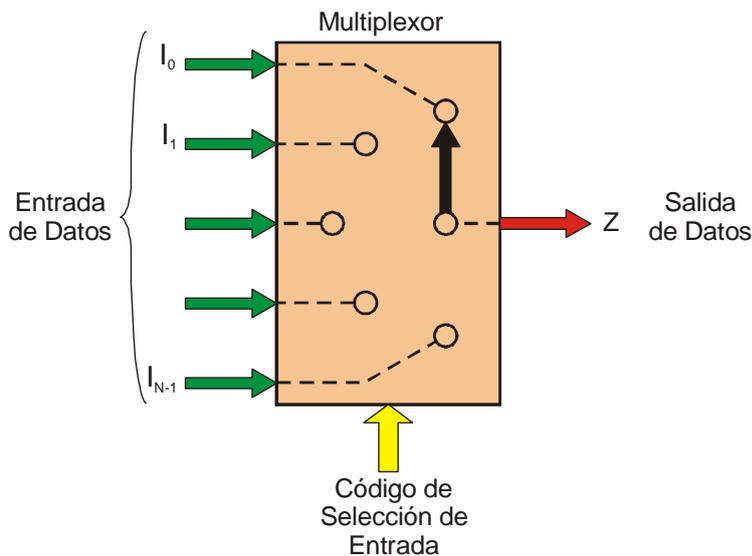
*Depends upon the BCD code applied during the 0 to 1 transition of LE.

Display



2. MULTIPLEXORES Y DEMULTIPLEXORES DIGITALES

Un multiplexor (o selector de datos) es un circuito lógico que posee varias entradas de datos digitales, pero sólo permite alcanzar la salida a uno de ellos. El control respecto a cual entrada se conecta a la salida, se realiza mediante unas líneas de selección del multiplexor (también llamadas entradas de dirección), de forma que el código binario en estas líneas de control corresponderá a la entrada que se desea seleccionar.



El multiplexor actúa como un interruptor de posiciones múltiples controlado digitalmente a través de las entradas de selección. Un multiplexor selecciona una entre N fuentes de datos de entrada y transmite la entrada seleccionada a la salida.



2.1. Multiplexor de dos entradas

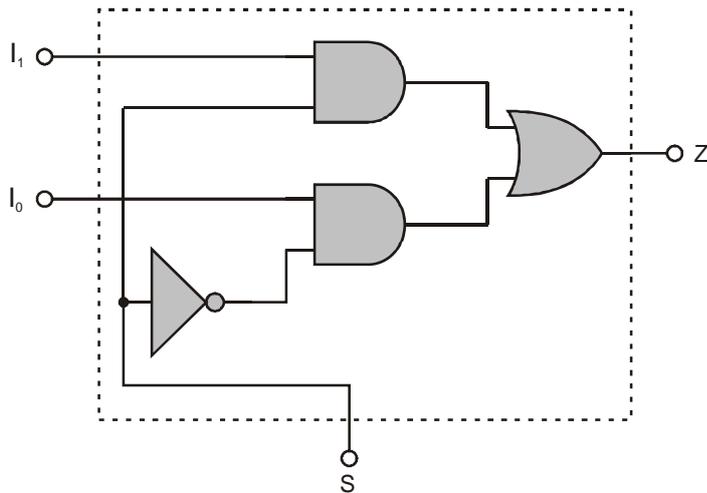
Veamos como podríamos implementar un multiplexor de dos entradas de datos, haciendo uso de lógica combinacional. Tenemos entonces dos entradas de datos I_0 e I_1 , una salida Z , y una entrada de selección S . La tabla de verdad que representa a este circuito sería entonces:

Selección S	Salida Z
0	$Z = I_0$
1	$Z = I_1$

Tenemos entonces que la función lógica del circuito resulta:

$$Y = I_0 \cdot \bar{S} + I_1 \cdot S$$

Es decir, cuando la entrada de selección S sea 0, la salida Z tomará el valor del dato presente en I_0 . Análogamente, cuando S sea 1, la salida Z tomará el valor del dato presente en I_1 .

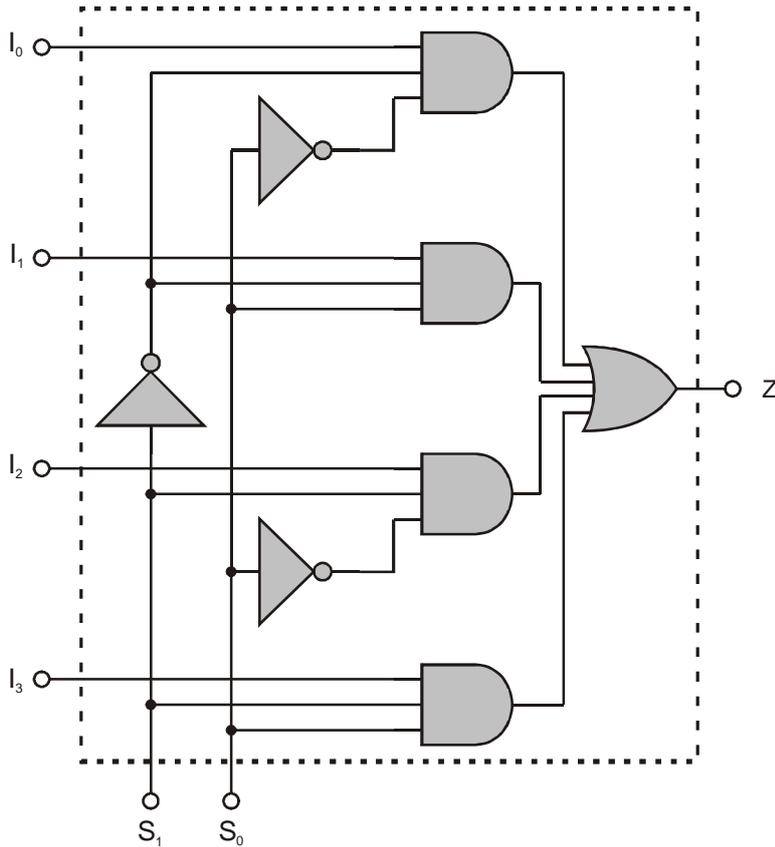


2.2. Multiplexor de cuatro entradas

Similar al caso anterior, aquí tendremos 4 entradas de datos (I_0 , I_1 , I_2 e I_3), una salida de datos Z , pero a diferencia del multiplexor anterior, aquí necesitaremos dos líneas de selección de datos y no una. La tabla de verdad es entonces:

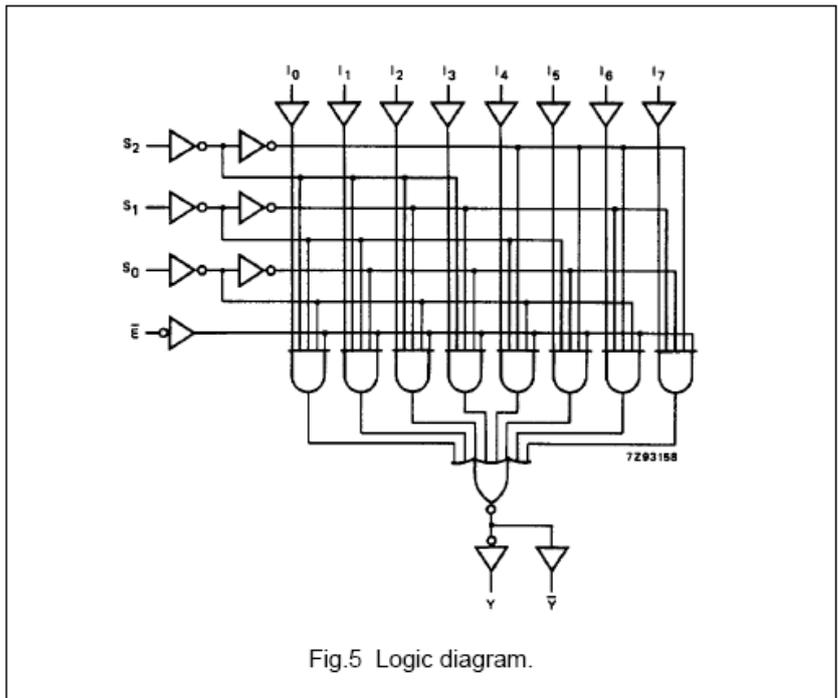
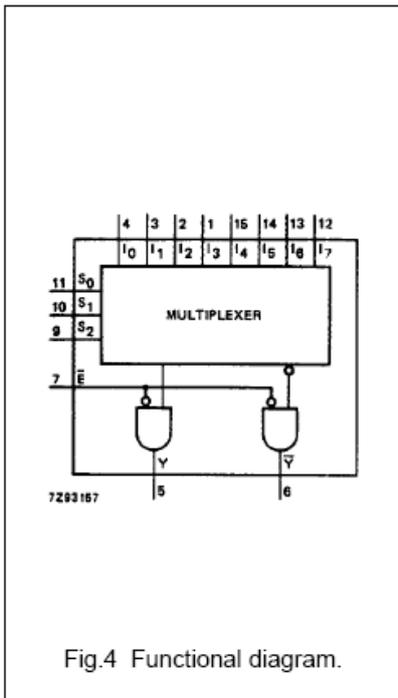
S_1	S_0	Salida Z
0	0	$Z = I_0$
0	1	$Z = I_1$
1	0	$Z = I_2$
1	1	$Z = I_3$

Y el circuito correspondiente:



2.3. Circuitos multiplexores comerciales

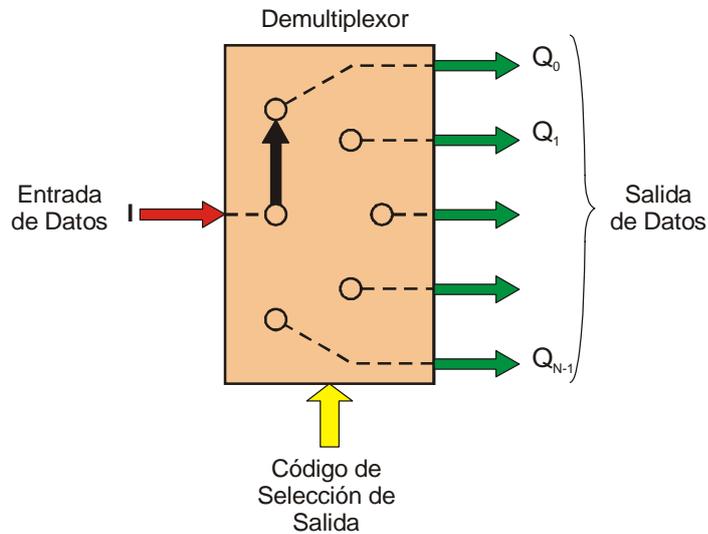
Comercialmente, los fabricantes suelen desarrollar multiplexores de 8 entradas de datos. Tal es el caso del 74HC151 por ejemplo, del cual se adjunta su hoja de datos. Observar que en los multiplexores comerciales se suele disponer de dos salidas, una normal y otra negada. Asimismo suele existir una línea de control adicional **E** (enable, o habilitación) con la cual se pueden deshabilitar todas las salidas.





2.4. Demultiplexores

Los demultiplexores son circuitos que realizan la operación inversa al multiplexor. Es decir, es un sistema que posee una sola entrada de datos y varios canales de salidas. También posee una serie de entradas de control mediante las cuales se selecciona el canal de salida al cual se desea redirigir la entrada de datos. Esquemáticamente, un demultiplexor sería:



Circuitualmente, la síntesis de un demultiplexor es muy similar a la del multiplexor. Tanto los multiplexores como los demultiplexores se utilizan mayormente en sistemas de transmisión de datos digitales y en sistemas de conversión serie-paralelo y paralelo-serie. En el caso de los multiplexores, también se los suele utilizar como una alternativa a los circuitos combinatoriales con compuertas, un uso poco difundido para un multiplexor pero que puede ser una solución muy útil. Veamos en el siguiente punto como sería esto.

2.5. Multiplexor como alternativa a los circuitos combinatoriales

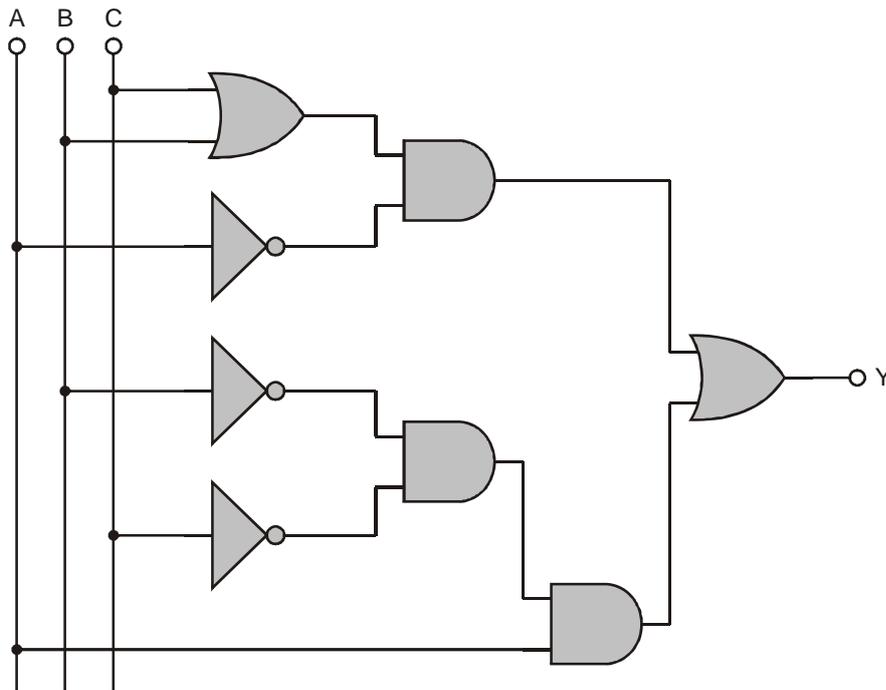
Como con cualquier sistema combinatorial, para lograr la síntesis circuital del mismo debemos partir de su tabla de verdad. Supongamos entonces que tenemos un sistema con tres entradas **A**, **B** y **C**, y con una salida **Y**. Supongamos también que las entradas y la salida están relacionadas por la siguiente tabla de verdad:

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

La función lógica correspondiente sería:

$$Y = \bar{A} \cdot (B + C) + A \cdot \bar{B} \cdot \bar{C}$$

Y de querer implementar dicha función con compuertas estándar, el circuito resultante sería el siguiente:



Hasta aquí hemos aplicado los conceptos de implementación de circuitos combinatoriales con compuertas. Veamos ahora cómo podemos reemplazar el circuito anterior mediante el uso de un multiplexor de 8 entradas de datos. Tomemos como referencia al 74HC151: la entrada \bar{E} (habilitación de salidas) es activa en nivel bajo, como queremos habilitar las salidas entonces a esta entrada la conectamos a masa (0 lógico).

Analicemos ahora las líneas de selección de entrada S_0 , S_1 y S_2 . Si en estas líneas aplicamos las entradas de la tabla de verdad (A , B y C), entonces según el valor que tomen estas variables estaremos seleccionando alguna de las entradas de datos del multiplexor. Supongamos que asignamos el orden de las variables de la siguiente forma:

A $\underline{\hspace{1cm}}$ S_2
 B $\underline{\hspace{1cm}}$ S_1
 C $\underline{\hspace{1cm}}$ S_0

Si por ejemplo $A = B = C = 0$ (primer caso de la tabla de verdad), estaremos seleccionando la entrada de datos I_0 . Según la tabla de verdad, a esta combinación de entrada corresponde un 0 como salida. Si para esta combinación de entrada la salida Y del multiplexor se conectará a su entrada I_0 , ¿que valor debería tener I_0 para que se cumpla la tabla de verdad? Evidentemente debe ser un 0 lógico, lo que significa que I_0 debe conectarse a masa.

Resumiendo, para utilizar un multiplexor como reemplazo a un circuito combinatorial clásico con compuertas, el concepto consiste en:

- Utilizar las entradas de selección de datos del multiplexor (S_0 , S_1 y S_2) como entradas de la tabla de verdad (A , B y C).
- Utilizar la salida del multiplexor (Y) como salida de la tabla de verdad.
- Conectar las entradas de datos del multiplexor (I_0 a I_7) a 0 lógico o a 1 lógico de acuerdo a las distintas combinaciones de la tabla de verdad.

Entonces, de acuerdo a los lineamientos anteriores, para implementar la tabla de verdad anterior con un multiplexor, el conexionado sería el siguiente:

