

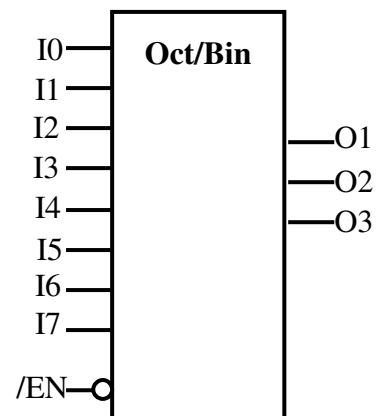
CODIFICADORES

Son los dispositivos MSI que realizan la operación inversa a la realizada por los decodificadores. Generalmente, poseen 2^n entradas y n salidas.

Cuando solo una de las entradas está activa para cada combinación de salida, se le denomina **codificador completo**.

Por ejemplo, el siguiente circuito proporciona a la salida la combinación binaria de la entrada que se encuentra activada. En este caso se trata de un codificador completo de **8** bits, o también llamado codificador de 8 a 3 líneas:

/EN	I0	I1	I2	I3	I4	I5	I6	I7	O1	O2	O3
1	X	X	X	X	X	X	X	X	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	0	1	1	1	1

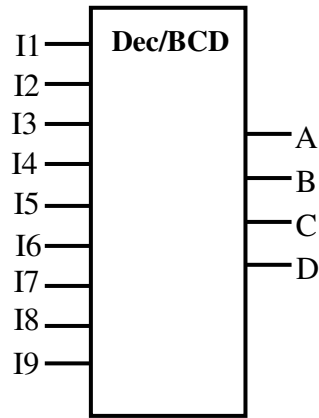


Las salidas codificadas, generalmente se usan para controlar un conjunto de 2^n dispositivos, suponiendo claro está que sólo uno de ellos está activo en cualquier momento. Sin embargo cuando nos encontremos con que se deben controlar dispositivos que pueden estar activos al mismo tiempo, problema que se suelen encontrar los sistemas microprocesadores, es preciso usar un dispositivo que nos proporcione a la salida el código del dispositivo que tenga más alta prioridad.

En la siguiente página podemos ver una sencilla comparación.

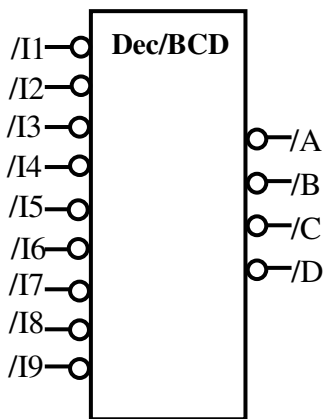
En la siguiente figura se representa el diagrama lógico de un codificador completo de Decimal a BCD natural, junto a su tabla de funcionamiento.

I1	I2	I3	I4	I5	I6	I7	I8	I9	A	B	C	D
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	1

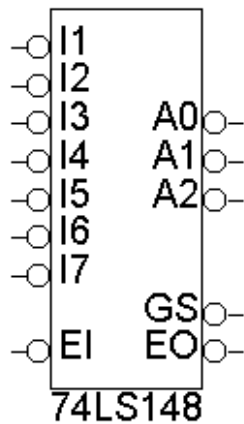


Por otro lado la figura siguiente representa el diagrama lógico del circuito 74147, que es un codificador de **prioridad** de Decimal a BCD natural; en la tabla de funcionamiento adjunta se puede notar la diferencia con el anterior.

/I1	/I2	/I3	/I4	/I5	/I6	/I7	/I8	/I9	/A	/B	/C	/D
X	X	X	X	X	X	X	X	0	0	1	1	0
X	X	X	X	X	X	X	0	1	0	1	1	1
X	X	X	X	X	X	0	1	1	1	0	0	0
X	X	X	X	X	0	1	1	1	1	0	0	1
X	X	X	X	0	1	1	1	1	1	0	1	0
X	X	X	0	1	1	1	1	1	1	0	1	1
X	X	0	1	1	1	1	1	1	1	1	0	0
X	0	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1



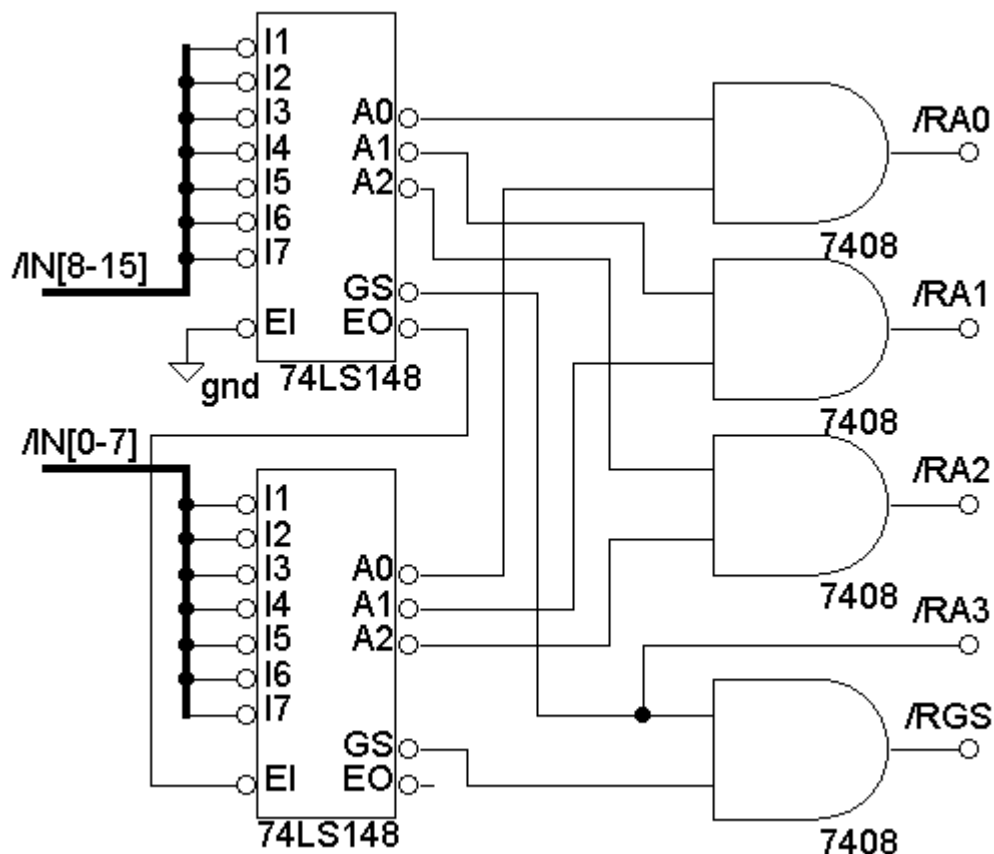
Cuando se trata de establecer la prioridad con mayor número de bits, es preciso recurrir a la asociación de codificadores. El siguiente diagrama muestra un codificador de prioridad de 16 líneas a 4, usando codificadores de prioridad 74148, de 8 a 3 líneas.



/EI: Habilitación

/GS: es 0 cuando el dispositivo está habilitado y una o más de sus entradas está activa

/EO: salida para habilitar otro decodificador de más baja prioridad



MULTIPLEXORES

Multiplexar es pasar información de “muchos” canales o líneas a “pocos” canales o líneas.

Un **MULTIPLEXOR (MUX)** es un circuito combinacional que selecciona una entrada y la transfiere a la salida. La selección de la entrada, o dato, se realiza según un conjunto de valores de las variables de control.

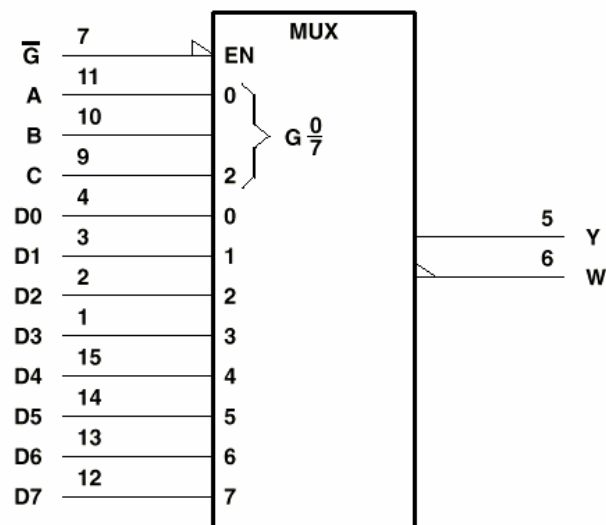
Poseen por tanto, **n** entradas de selección, para **2ⁿ** entrada de datos, proporcionando **dos salidas**: una para el dato directo y otra para el dato negado.

A continuación se presenta la tabla de funcionamiento y el Símbolo lógico estándar para un multiplexor de 8 a 1 líneas. Se trata del circuito 74151, con entrada de habilitación activa a nivel bajo.

FUNCTION TABLE

INPUTS				OUTPUTS	
SELECT			STROBE	Y	W
C	B	A	G		
X	X	X	H	L	H
L	L	L	L	D0	D0
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

H = high level, L = low level, X = irrelevant
D0, D1, . . . D7 = the level of the respective D input



Existen en el mercado diferentes multiplexores.

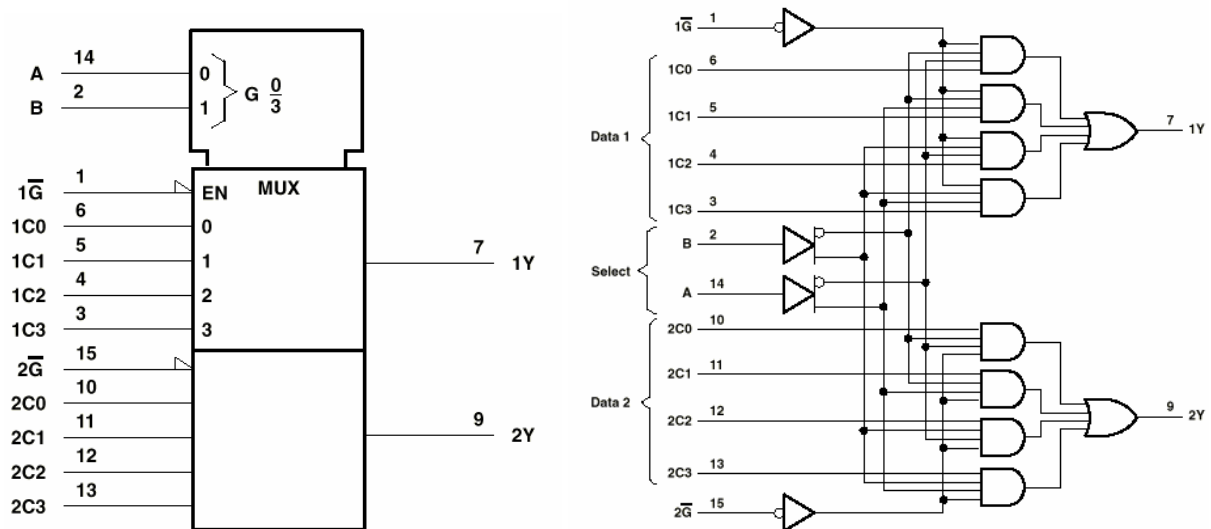
El siguiente corresponde al 74153, que es un circuito MSI con dos Mux de 4 a 1 líneas.

A los Mux se les suele llamar también selectores de datos. Así este sería un selector de datos 1-de-4 (ya que selecciona un dato de cuatro disponibles).

FUNCTION TABLE

SELECT		DATA				STROBE \overline{G}	OUTPUT Y
B	A	C0	C1	C2	C3		
X	X	X	X	X	X	H	L
L	L	L	X	X	X	L	L
L	L	H	X	X	X	L	H
L	H	X	L	X	X	L	L
L	H	X	H	X	X	L	H
H	L	X	X	L	X	L	L
H	L	X	X	H	X	L	H
H	H	X	X	X	L	L	L
H	H	X	X	X	H	L	H

Select inputs A and B are common to both sections.

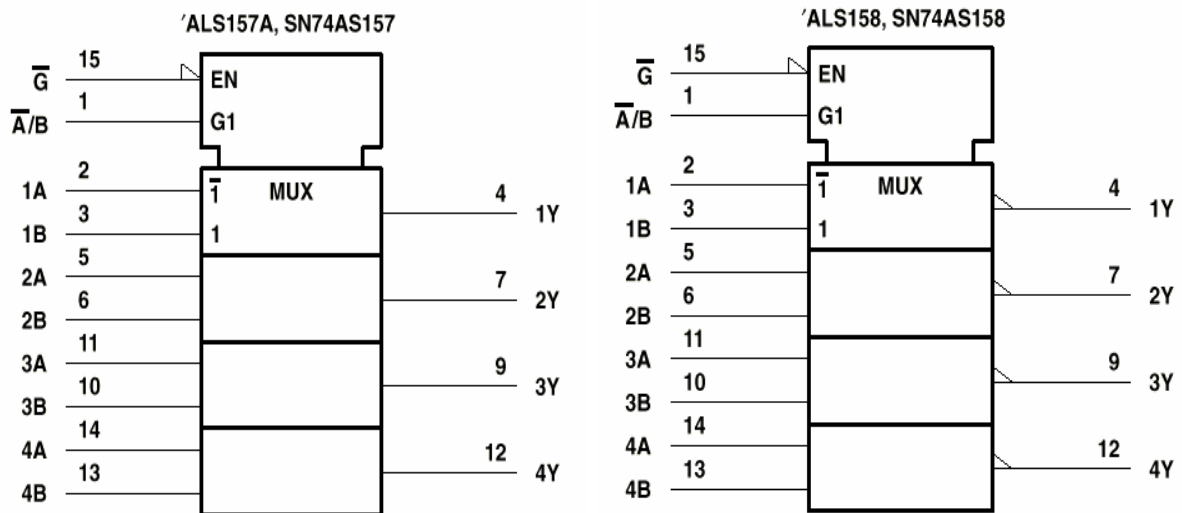


El diagrama lógico sobre estas líneas corresponde a la configuración interior del dispositivo arriba descrito, lo que nos puede dar una idea de la diferencia entre los circuitos SSI y los MSI.

Un multiplexor muy usado es el 74157 (El 74158 es una versión del primero con las salidas activas a nivel bajo). Se trata de un circuito con cuádruple multiplexor de dos a una líneas:

FUNCTION TABLE

INPUTS				OUTPUT Y	
\overline{G}	$\overline{A/B}$	DATA		'ALS157A SN74AS157	'ALS158 SN74AS158
		A	B		
H	X	X	X	L	H
L	L	L	X	L	H
L	L	H	X	H	L
L	H	X	L	L	H
L	H	X	H	H	L



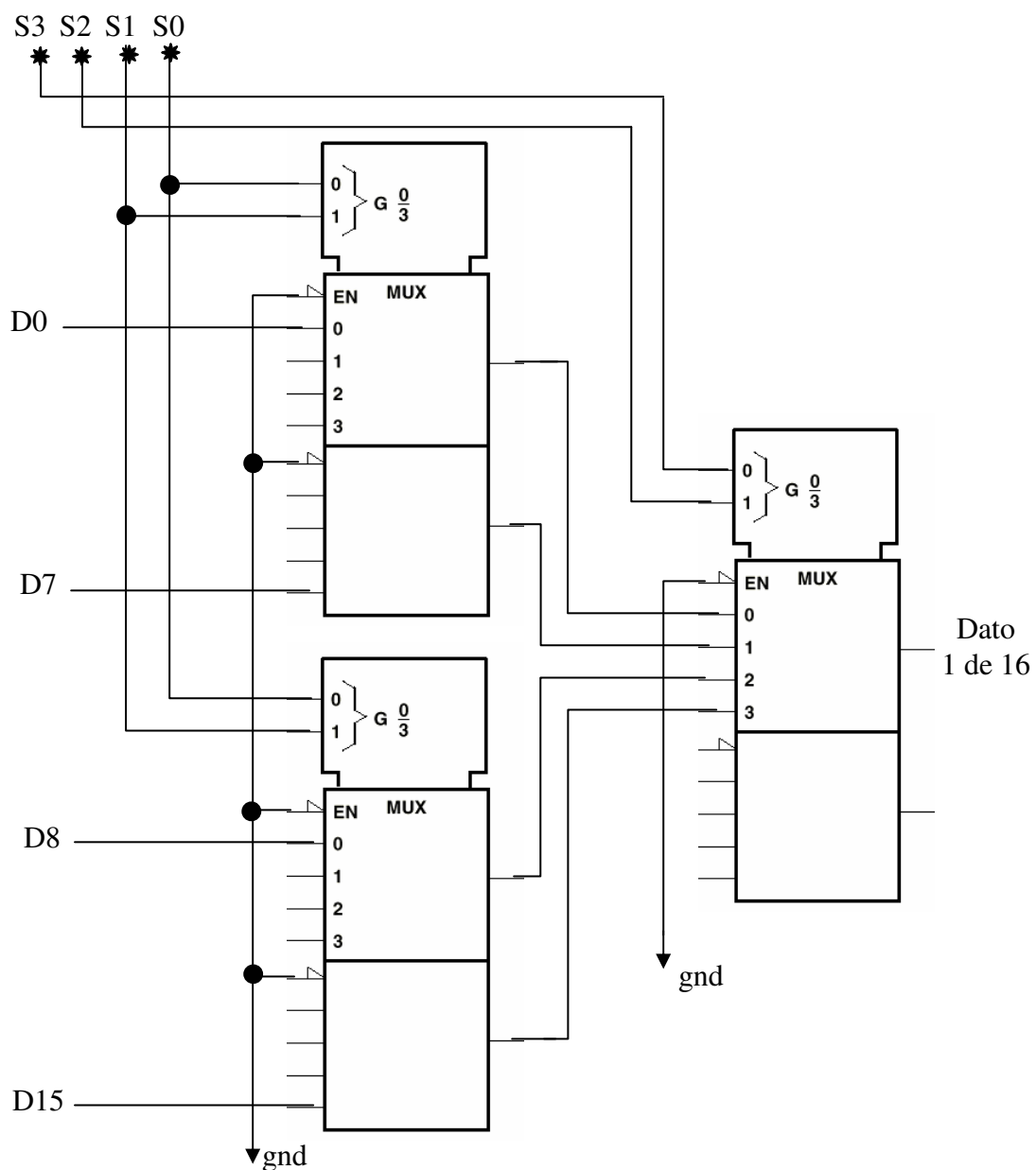
Este Mux funciona como selector de palabras: según sea el valor de selección $G1$, en las cuatro salidas aparece $A[1-4]$ ó $B[1-4]$, lo que lo hace muy útil en las aplicaciones aritméticas que se verán en el tema siguiente.

EXTENSIÓN DE MULTIPLEXORES

Como siempre, no siempre se dispone del número necesario de entradas en un solo dispositivo, con lo que hay que recurrir a asociar dispositivos de menor número de entradas para obtener uno mayor.

Vamos a construir un MUX de 16:1 usando MUX de 4:1.

La asociación es muy similar a la que se hace con los decodificadores, pero cambian la disposición de los MSB y LSB (bits más significativos y menos significativos).



REALIZACIÓN DE FUNCIONES COMBINACIONALES CON MUX

Partimos de la siguiente afirmación: un multiplexor de 2^n entradas puede realizar cualquier función lógica de $n+1$ variables.

Se pueden usar dos métodos:

- de forma algebraica;
- de forma tabular.

Nos centraremos en este último, y lo seguiremos mediante un ejemplo:

1. A partir de la expresión canónica y se escoge un Mux determinado:

Ej: Sea $f(A,B,C,D) = \sum_4 m(0,2,3,7,8,13,15)$

Al ser una función de 4 variables necesitamos un MUX de 8 a 1 líneas (o sea, con tres variables de control)

2. Se crea un mapa de Karnaugh de manera que la numeración en las columnas, coincida con la entrada que se pretende seleccionar.

Así, las columnas, vendrán determinadas por las variables de control del MUX, y las filas por el dato o los datos que se quieren transmitir. Las variables de control deben ser las de menor peso. Evaluando cada columna identificamos el valor que hay que colocar en cada entrada.

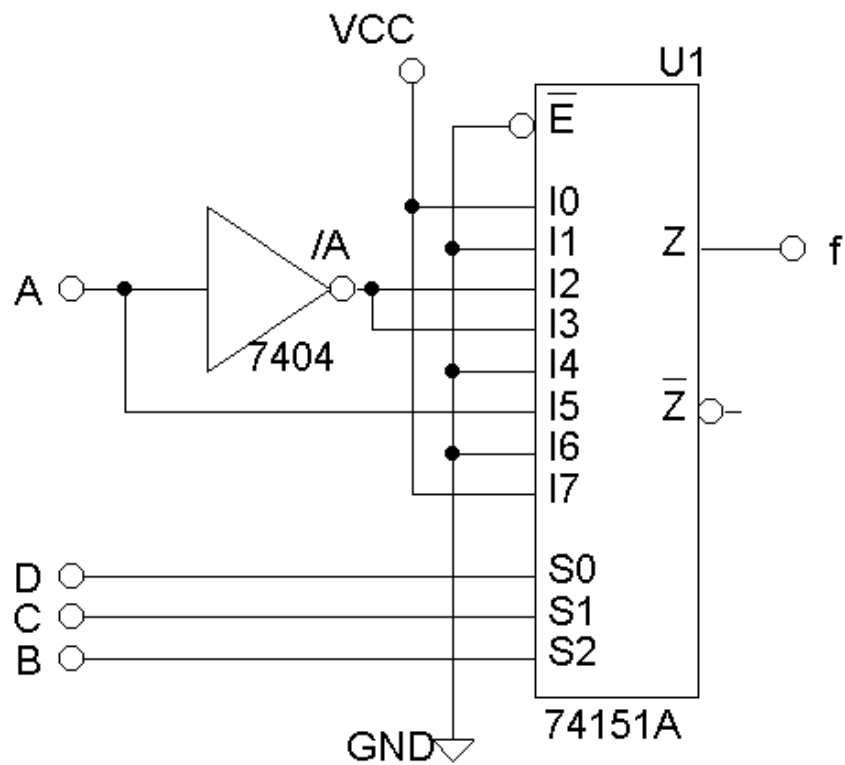
Ej: Realizamos dicho mapa para nuestra función:

		BCD							
		I0 000	I1 001	I2 010	I3 011	I4 100	I5 101	I6 110	I7 111
A	0	1 0	0 1	1 2	1 3	0 4	0 5	0 6	1 7
	1	1 8	0 9	0 10	0 11	0 12	1 13	0 14	1 15
		1	0	/A	/A	0	A	0	1

3. Hacemos el diagrama lógico del circuito colocando en las entradas de datos lo que la tabla nos indique.

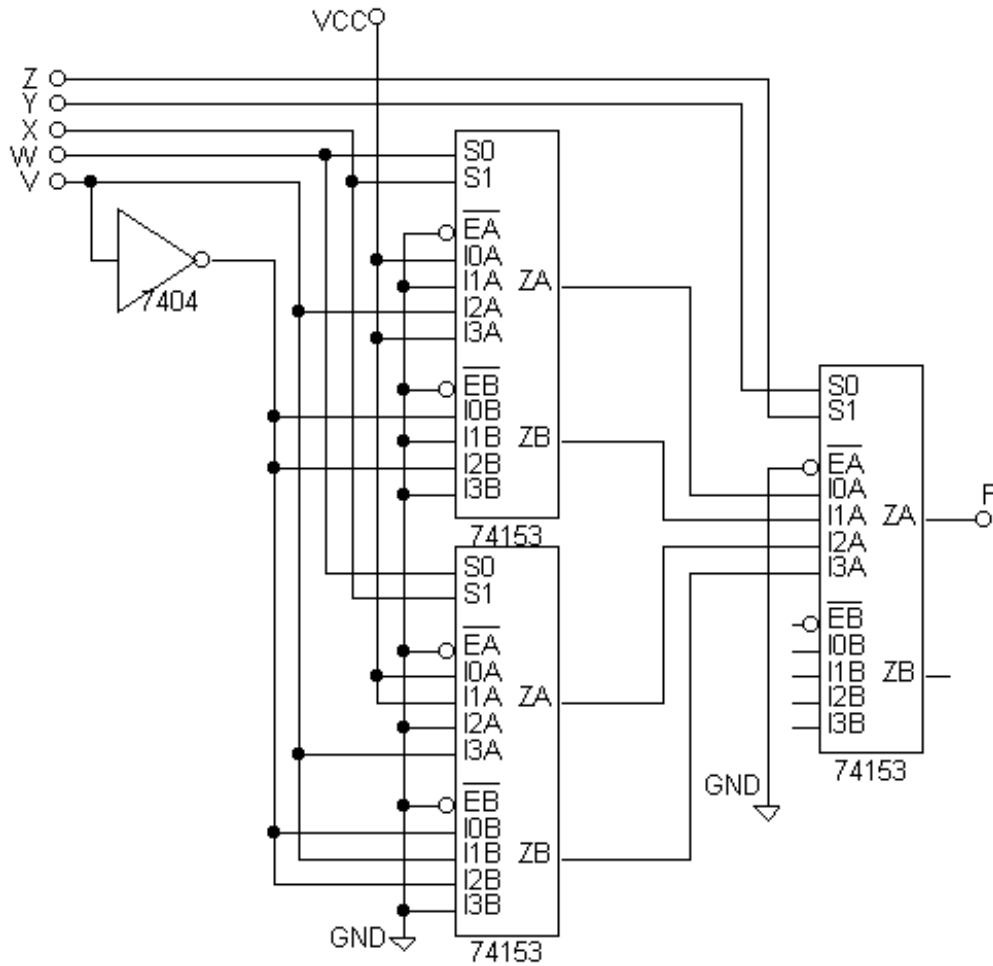
Ej: Nuestra tabla nos dice que en la entrada **I0** de nuestro MUX, debemos colocar un **1**; que en la **I1**, un **0**; ...

Nos debe quedar un circuito como el de la figura:



EJERCICIO DE MUX

Para familiarizarnos con el empleo de MUX, deduzcamos la función que realiza el siguiente circuito



DEMULTIPLEXORES

En realidad no existen como tales, sino que vienen definidos por los decodificadores/demultiplexores.

La función que debe realizar es la **inversa** de la que realiza el MUX, o sea, debemos seleccionar una salida por donde transmitir el dato de la entrada.

Por tanto, el circuito constará de **1** entrada de datos, **n** entradas de selección de salida, y **2ⁿ** salidas.

El Decodificador/DEMUX 74138 que ya conocemos utiliza su entrada de habilitación **G1** para entrada de Datos:

FUNCTION TABLE

INPUTS						OUTPUTS							
ENABLE			SELECT										
G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

