

# MULTIPLEXORES

Son circuitos combinacionales que poseen las siguientes entradas y salidas:

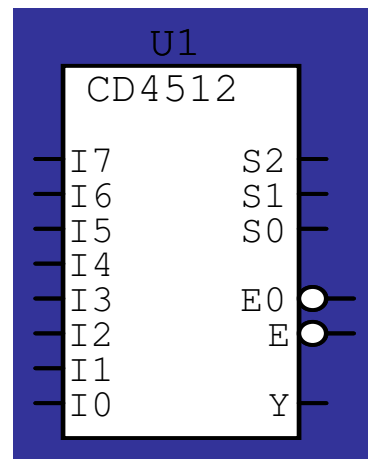
- **m** entradas de informacion o de datos
- **n** entradas de seleccion o control
- Una salida de información
- Una entrada de autorización o enable

La relación entre las entradas de datos y las entradas de selección es:

$$\text{Numero de entradas} = 2^n$$

En los esquemas representativos de estos circuitos se suele llamar a dichas entradas y salidas con los simbolos que se exponen a continuación

- I0 a In a las entradas de datos.
- S0 a Sn a las entradas de selección.
- E a la entrada de autorización.
- Y a la salida.



E	S <sub>1</sub>	S <sub>0</sub>	Z
0	0	0	I <sub>0</sub>
0	0	1	I <sub>1</sub>
0	1	0	I <sub>2</sub>
0	1	1	I <sub>3</sub>
1	0	0	Z
1	0	1	Z
1	1	0	Z
1	1	1	Z

El enable es el encargado de habilitar el dispositivo según el valor que tome este funcionara como multiplexor ( $E=0$ ) o bien dando en la salida una alta impedancia ( $E=1$ ).

Siempre que  $E=0$  el funcionamiento del multiplexor es el siguiente:

Cuando una combinación binaria aparece en las entradas de selección, la información de entrada presente en el canal por ella definido aparece en la salida

La función que nos informa de la salida es

$$Z = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3$$

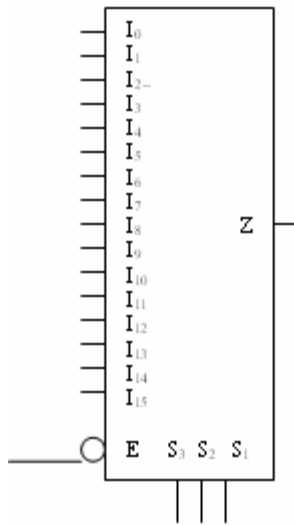
Por que podemos considerar a un multiplexor como un conmutador de múltiples entradas y cuya única salida se controla mediante las entradas de selección.

- **TIPOS DE MULTIPLEXORES**

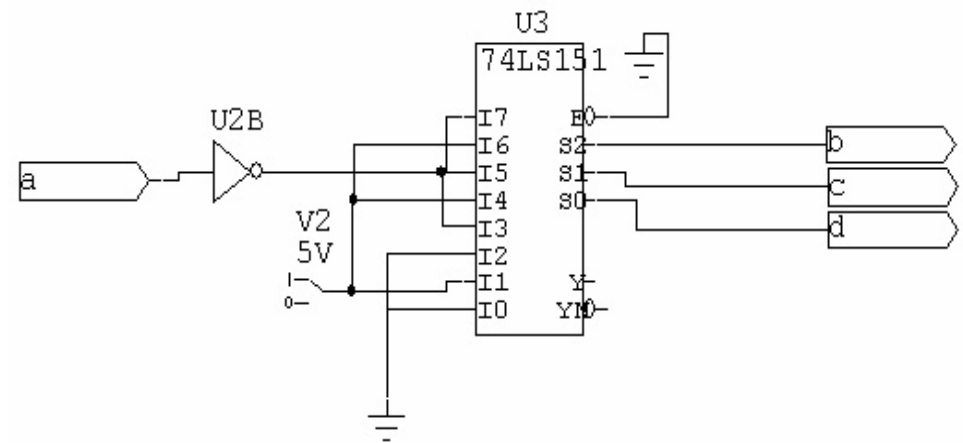
- Los más comerciales son de 4 a 1, de 8 a 3 y de 16 a 4, pudiendo conectar entre si dichas configuraciones para hacer circuitos más económico.
- **Forma de utilizar los multiplexores**
  - - ***Empleando multiplexores de igual número de entradas de selección que de variables a implementar.***
  - - ***Empleando multiplexores con número de entradas de selección inferior en una al de las variables a implementar.***

# Ejemplo de uso

Utilizando mismo número de entradas de selección que de variables a implementar



Utilizando un numero de entradas de selección inferior al de variables a implementar



# DESCRIPCIÓN DE MULTIPLEXORES EN VHDL

- Este ejemplo simula un multiplexor de dos entradas. Es un ejemplo sencillo que muestra como describir un elemento a partir de su funcionamiento.

```
entity MUX2a1 is  
    port(a: in std_logic;  
          b: in std_logic;  
          sel: in std_logic;  
          z: out std_logic);  
end entity  
architecture dataflow of MUX2a1 is  
    begin  
        z <= a when sel='0' else b;  
end dataflow;
```

**Un ejemplo algo más complejo es el de un multiplexor de cuatro entradas. Este ejemplo trabaja con vectores para controlar la entrada activa a través de la entrada *sel*.**

```
entity MUX4a1 is  
    port(a: in std_logic;  
        b: in std_logic;  
        c: in std_logic;  
        d: in std_logic;  
        z: out std_logic;  
        sel: in std_logic_vector(1 downto 0)); end entity;  
architecture dataflow of MUX4a1 is  
    begin  
        process(a,b,c,d,sel) begin  
            case sel is  
                when "00" => z <= a;  
                when "01" => z <= b;  
                when "10" => z <= c;  
                when "11" => z <= d;  
            end case;  
        end process;  
    end dataflow;
```

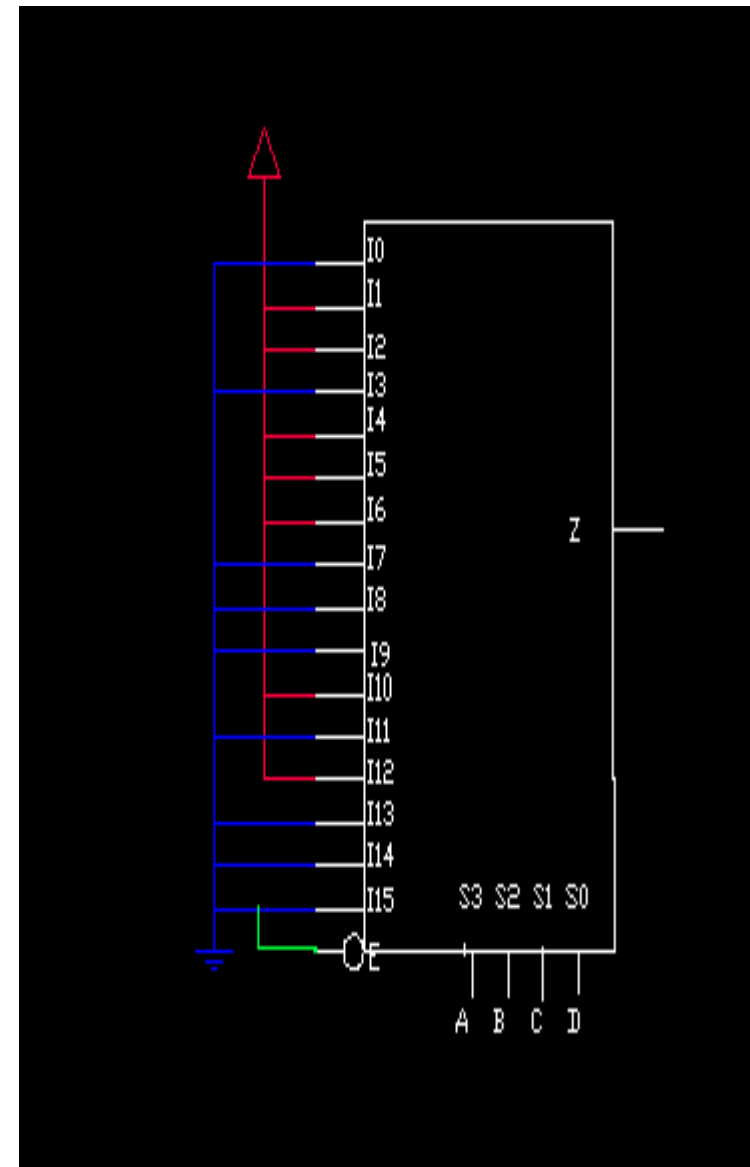
# Ejemplos

- Implementación de funciones lógicas con multiplexores

$$F(A,B,C,D) = \Sigma(2,4,5,6,10,12) + \Sigma_{\emptyset}(1,3,14,15)$$

	00	01	11	10
00		1	1	
01	0	1		
11	0		0	
10	1	1	0	1

Diagram illustrating the Karnaugh map for the function  $F(A,B,C,D)$ . The map shows the function value (0 or 1) for each combination of inputs A, B, C, and D. The map is partitioned into four groups of four cells each, corresponding to the minterms in the sum of products expression. The groups are: (00, 01, 11, 10) for A=0, (01, 11, 10, 00) for B=1, (00, 01, 11, 10) for C=1, and (00, 01, 11, 10) for D=1. The output Z is indicated by a vertical line on the right.

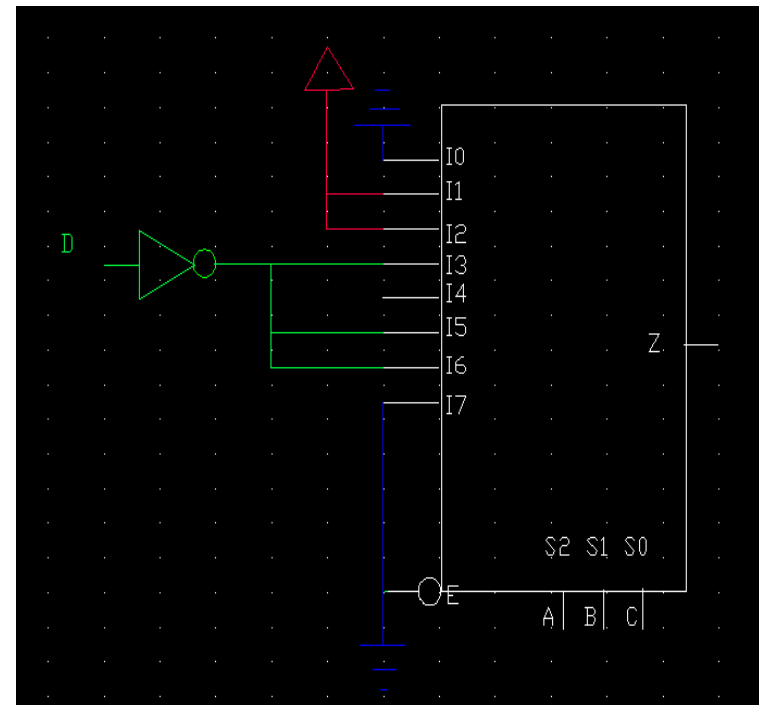




Podemos implementar la función anterior con un multiplexor de 8 a 1 con lo cual abaratamos

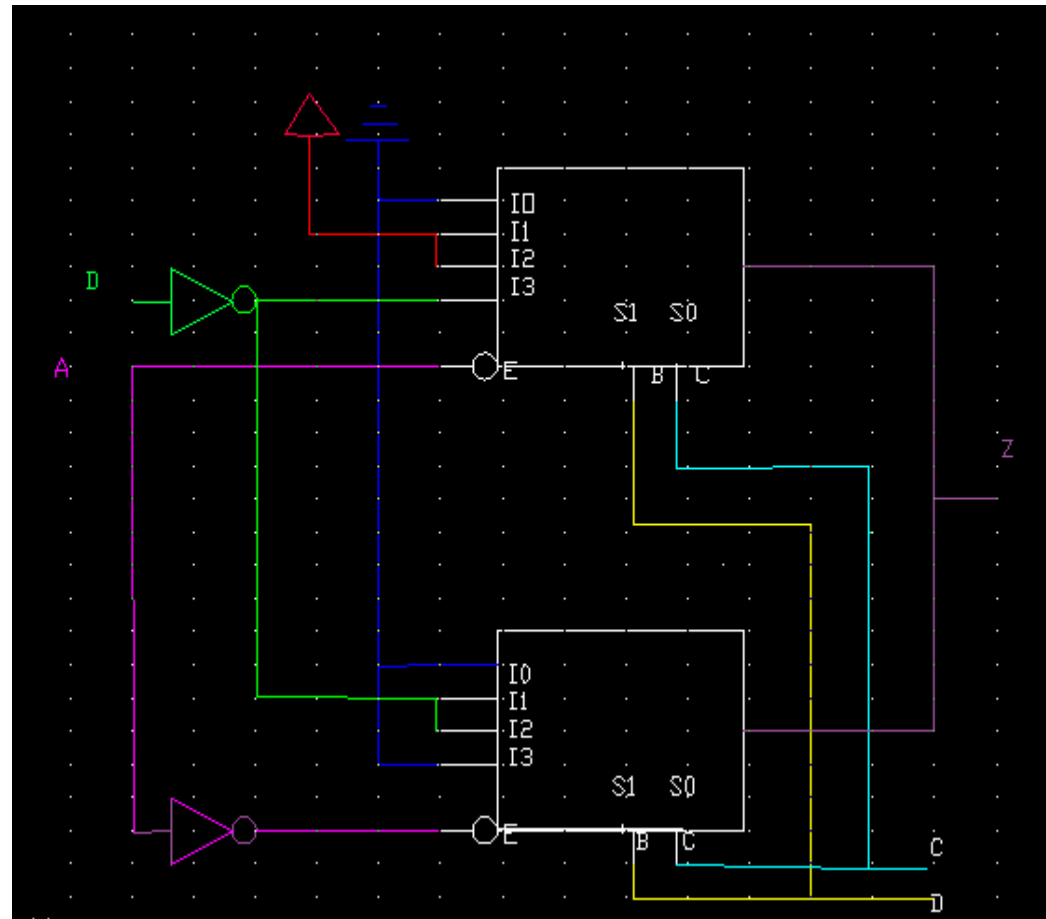
A	B	C	D	F0	F1
0	0	0	0	0	$I_0=0$
0	0	0	1	$\emptyset$	
0	0	1	0	1	$I_1=1$
0	0	1	1	$\emptyset$	
0	1	0	0	1	$I_2=1$
0	1	0	1	1	
0	1	1	0	1	$I_3=\overline{D}$
0	1	1	1	0	
1	0	0	0	0	$I_4=0$
1	0	0	1	0	
1	0	1	0	1	$I_5=D$
1	0	1	1	0	
1	1	0	0	1	$I_6=D$
1	1	0	1	0	
1	1	1	0	$\emptyset$	$I_7=0/1$
1	1	1	1	$\emptyset$	

8 A 1



# Podemos hacerlo también con multiplexores 4 a 1

A	B	C	D	F0	F1
0	0	0	0	0	$I_0=0$
0	0	0	1	$\emptyset$	
0	0	1	0	1	$I_1=1$
0	0	1	1	$\emptyset$	
0	1	0	0	1	$I_2=1$
0	1	0	1	1	
0	1	1	0	1	$I_3=\overline{D}$
0	1	1	1	0	
1	0	0	0	0	$I_0=0$
1	0	0	1	0	
1	0	1	0	1	$I_1=D$
1	0	1	1	0	
1	1	0	0	1	$I_2=D$
1	1	0	1	0	
1	1	1	0	$\emptyset$	$I_3=0/1$
1	1	1	1	$\emptyset$	



# Implementar un multiplexor de cuatro a uno canales con puertas lógicas

$$S = \overline{C_1} \overline{C_0} \overline{D_0} + \overline{C_1} \overline{C_0} D_1 + \overline{C_1} C_0 \overline{D_2} + \overline{C_1} C_0 D_3$$

Entradas de control		Canales de entrada				Salidas
$C_1$	$C_0$	$D_3$	$D_2$	$D_1$	$D_0$	$S$
0	0	X	X	X	0	0
0	0	X	X	X	1	1
0	1	X	X	0	X	0
0	1	X	X	1	X	1
1	0	X	0	X	X	0
1	0	X	1	X	X	1
1	1	0	X	X	X	0
1	1	1	X	X	X	1

