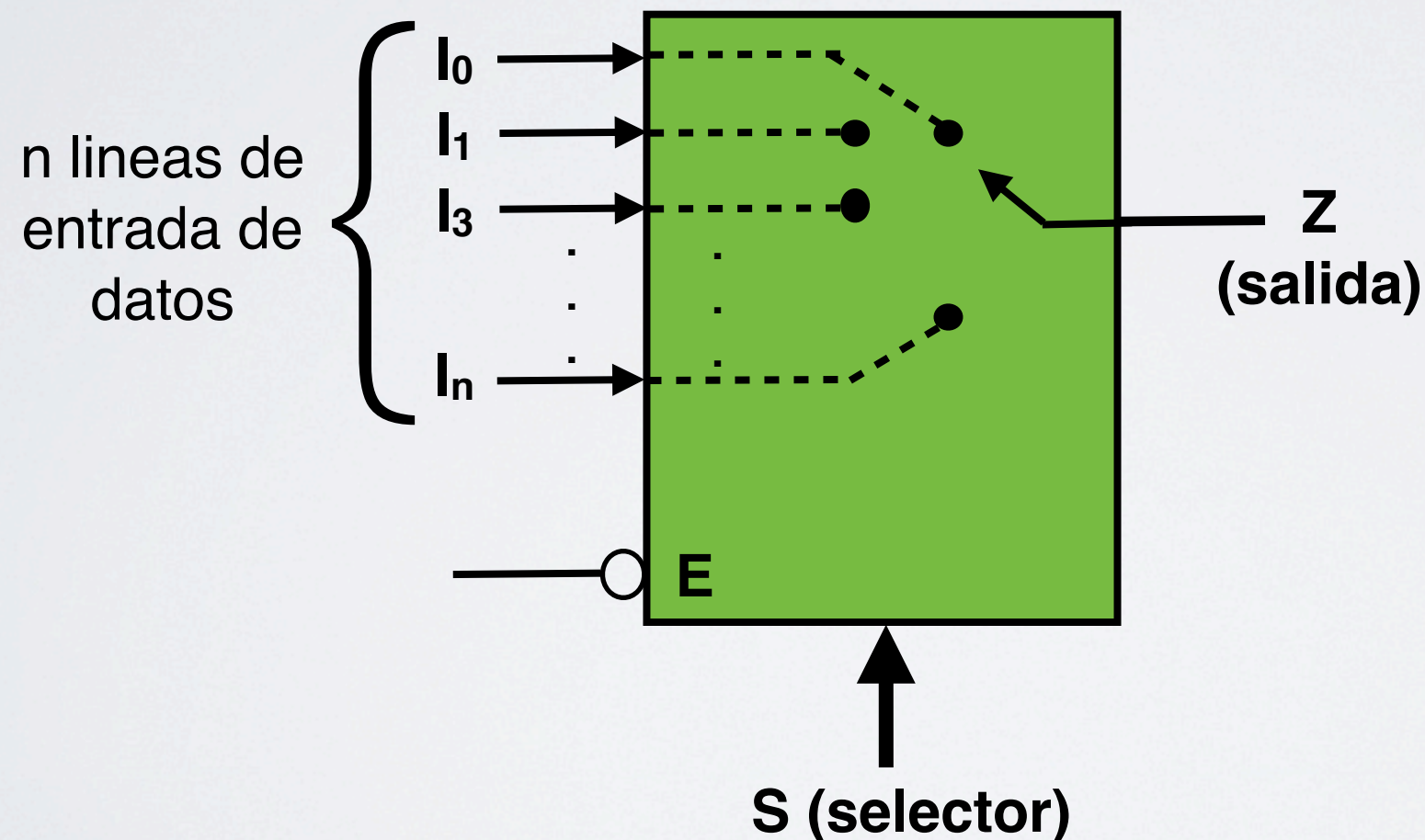


MULTIPLEXOR

FUNDAMENTOS DE DISEÑO DIGITAL
OPTATIVA I. ISISA

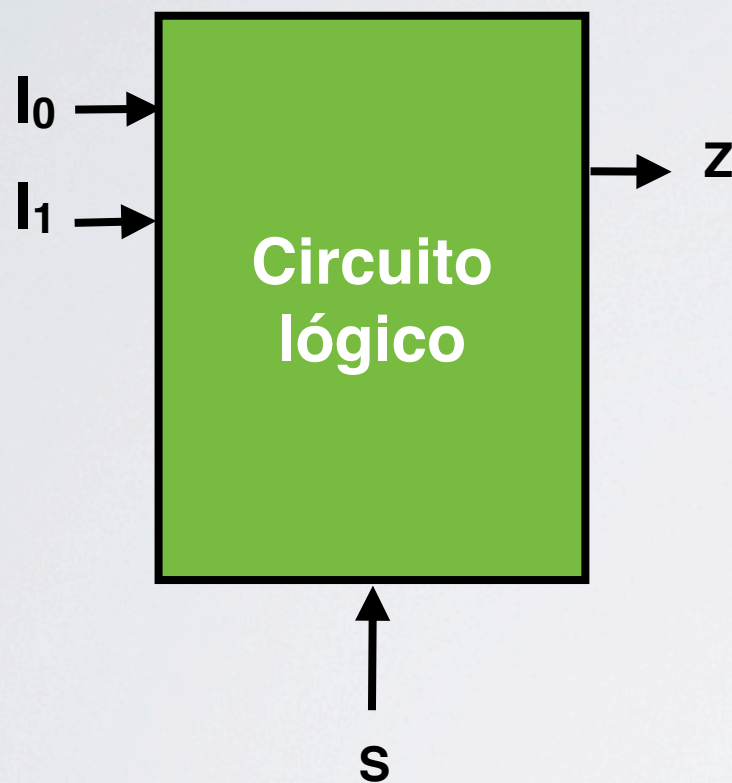
Multiplexor

Es un circuito digital que acepta varias entradas de datos y permite sólo a una de ellas alcanzar la salida. El dato de entrada que se verá en la salida es seleccionado a través de las entradas de "selección".



Multiplexor

Diseñe un circuito lógico que tenga tres entradas I_0, I_1 y S y una salida Z que funcione de acuerdo con la siguiente tabla.



S	Z
0	I_0
1	I_1

Si bien, con experiencia en el diseño de circuitos lógicos se podría obtener directamente la función Booleana y con esto el circuito lógico de la tabla de verdad anterior, es conveniente reescribir la tabla considerando todos los posibles valores lógicos de las entradas I_0, I_1 y S con el fin de encontrar la función Booleana a través del método de minimización de Mapas de Karnaugh.

S	I_1	I_0	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Multiplexor

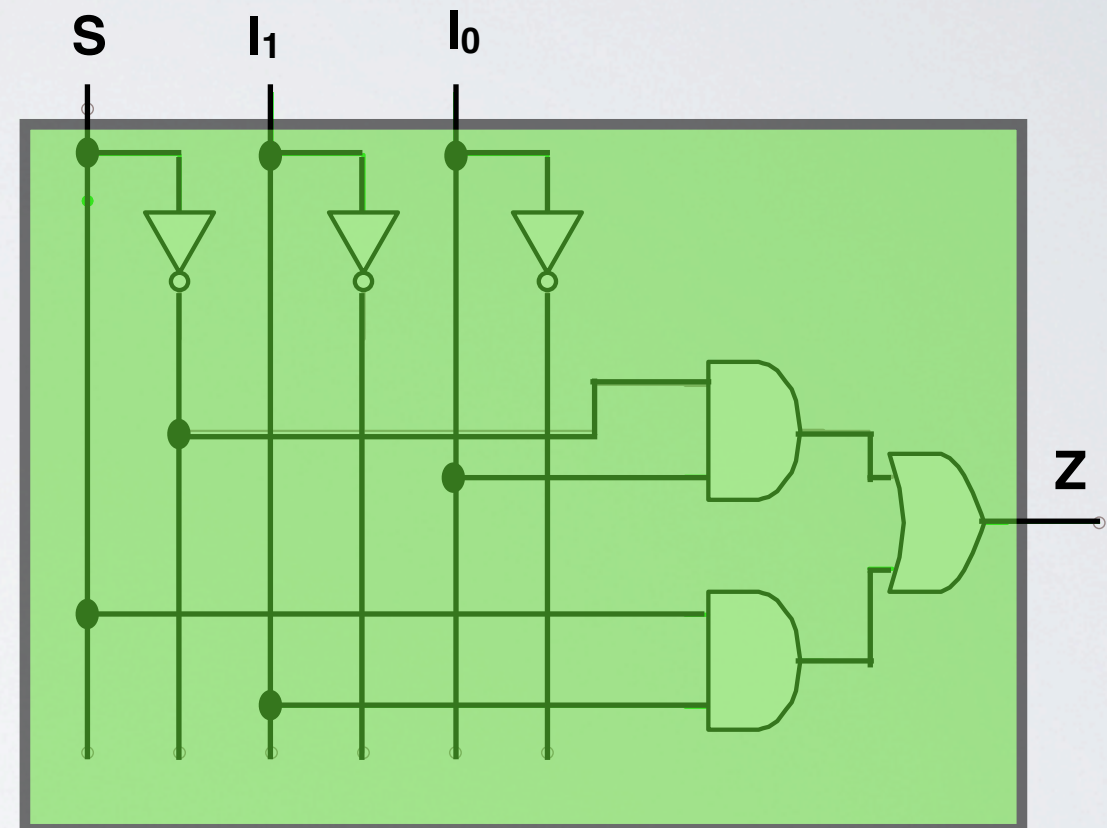
De la tabla de verdad anterior se tiene la siguiente función:

$$Z = \sum m(1, 3, 6, 7)$$

Minimizando la función por Mapas de Karnaugh se tiene:

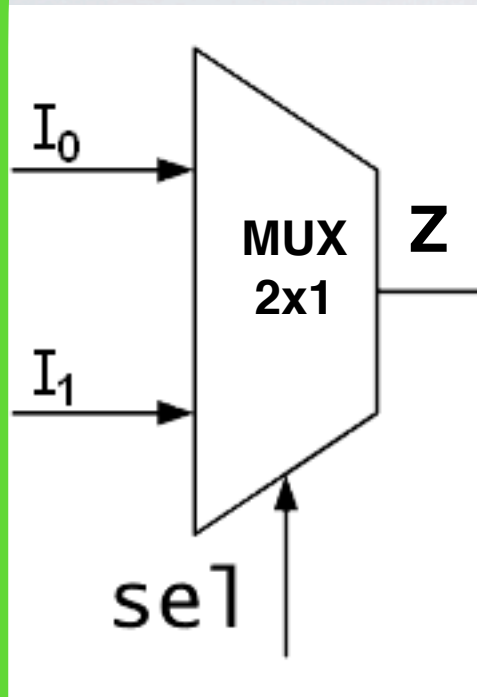
$I_1 I_0$		00	01	11	10
S	0		1	1	
	1			1	1

$$Z = \bar{S}I_0 + SI_1$$



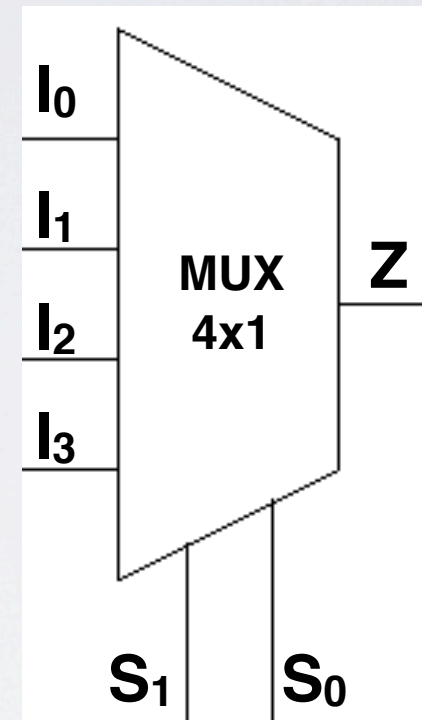
Multiplexor 2x1
2 - entradas ($I_1 I_0$)
1- salida (Z)
1- selector (S)

Tipos de Multiplexores



Sel	Z
0	I_0
1	I_1

Multiplexor 2x1
2 - entradas (I_1 I_0)
1- salida (Z)
1- selector (S)

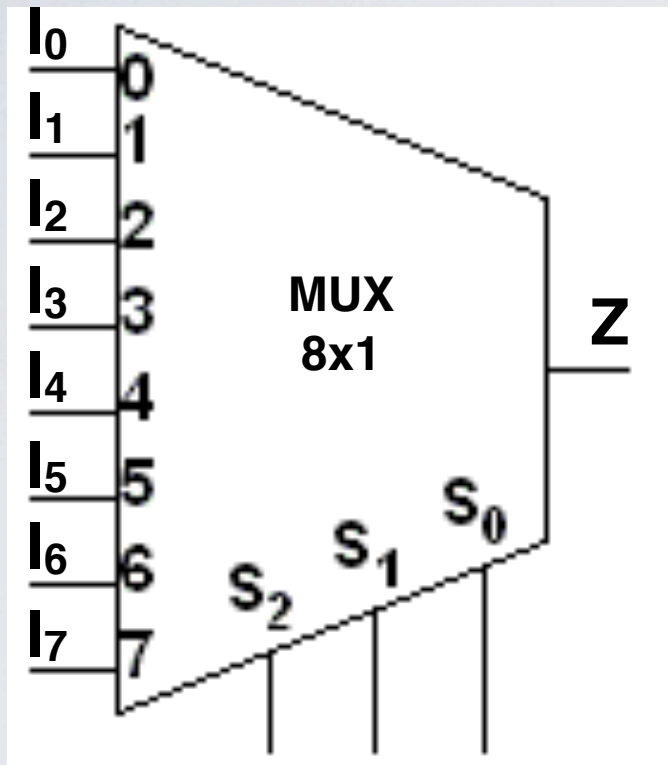


$S_1 S_0$	Z
00	I_0
01	I_1
10	I_2
11	I_3

Multiplexor 4x1
4 - entradas (I_3 I_2 I_1 I_0)
1- salida (Z)
2- selector (S_1 S_0)

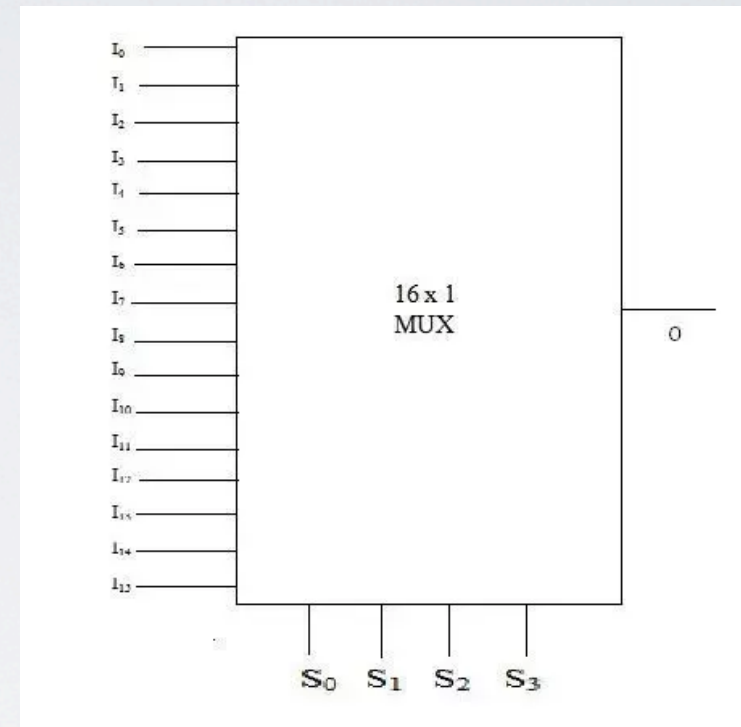
Multiplexor

Tipos de Multiplexores



$S_2 S_1 S_0$	Z
000	I_0
001	I_1
010	I_2
011	I_3
100	I_4
101	I_5
110	I_6
111	I_7

Multiplexor 8x1
8 - entradas ($I_7 I_6 I_5 I_4 I_3 I_2 I_1 I_0$)
1- salida (Z)
3- selector ($S_2 S_1 S_0$)



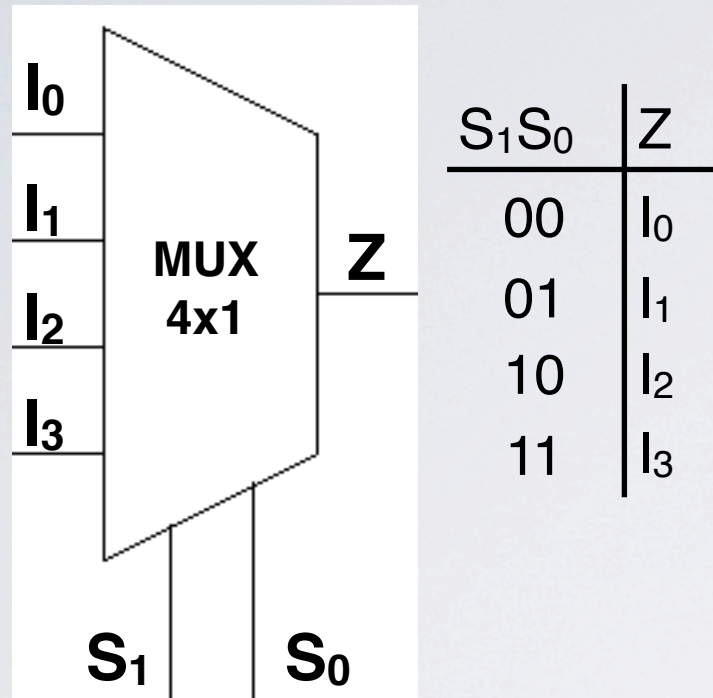
$S_3 S_2 S_1 S_0$	Z
0000	I_0
0001	I_1
0010	I_2
0011	I_3
0100	I_4
0101	I_5
0110	I_6
0111	I_7
1000	I_8
1001	I_9
1010	I_{10}
1011	I_{11}
1100	I_{12}
1101	I_{13}
1110	I_{14}
1111	I_{15}

Multiplexor 16x1
16 - entradas ($I_{15} I_{14} I_{13} I_{12} \dots I_1 I_0$)
1- salida (Z)
4- selector ($S_3 S_2 S_1 S_0$)

Multiplexor



Descripción en VHDL de un multiplexor 4x1



S ₁ S ₀	Z
00	I ₀
01	I ₁
10	I ₂
11	I ₃

Al describir en VHDL cualquier multiplexor lo que se utiliza es la tabla de verdad. A continuación se tiene la descripción de un MUX 4x1 usando primero la instrucción ***With-Select*** y después la instrucción ***When-case***.

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY MUX IS
PORT
    ( S: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
      I0,I1,I2,I3: IN STD_LOGIC;
      Z: OUT STD_LOGIC;
    );
END ENTITY;

ARCHITECTURE A_MUX OF MUX IS
BEGIN

    WITH S SELECT
    Z<=I0  WUEN "00",
        I1  WUEN "01",
        I2  WUEN "10",
        I3  WHEN OTHERS;

END A_MUX;
```

Instrucción ***With-Select***

Instrucción ***When-case***

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY MUX IS
PORT
    ( S: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
      I0,I1,I2,I3: IN STD_LOGIC;
      Z: OUT STD_LOGIC;
    );
END ENTITY;

ARCHITECTURE A_MUX OF MUX IS
BEGIN

    PROCESS (S)
    BEGIN

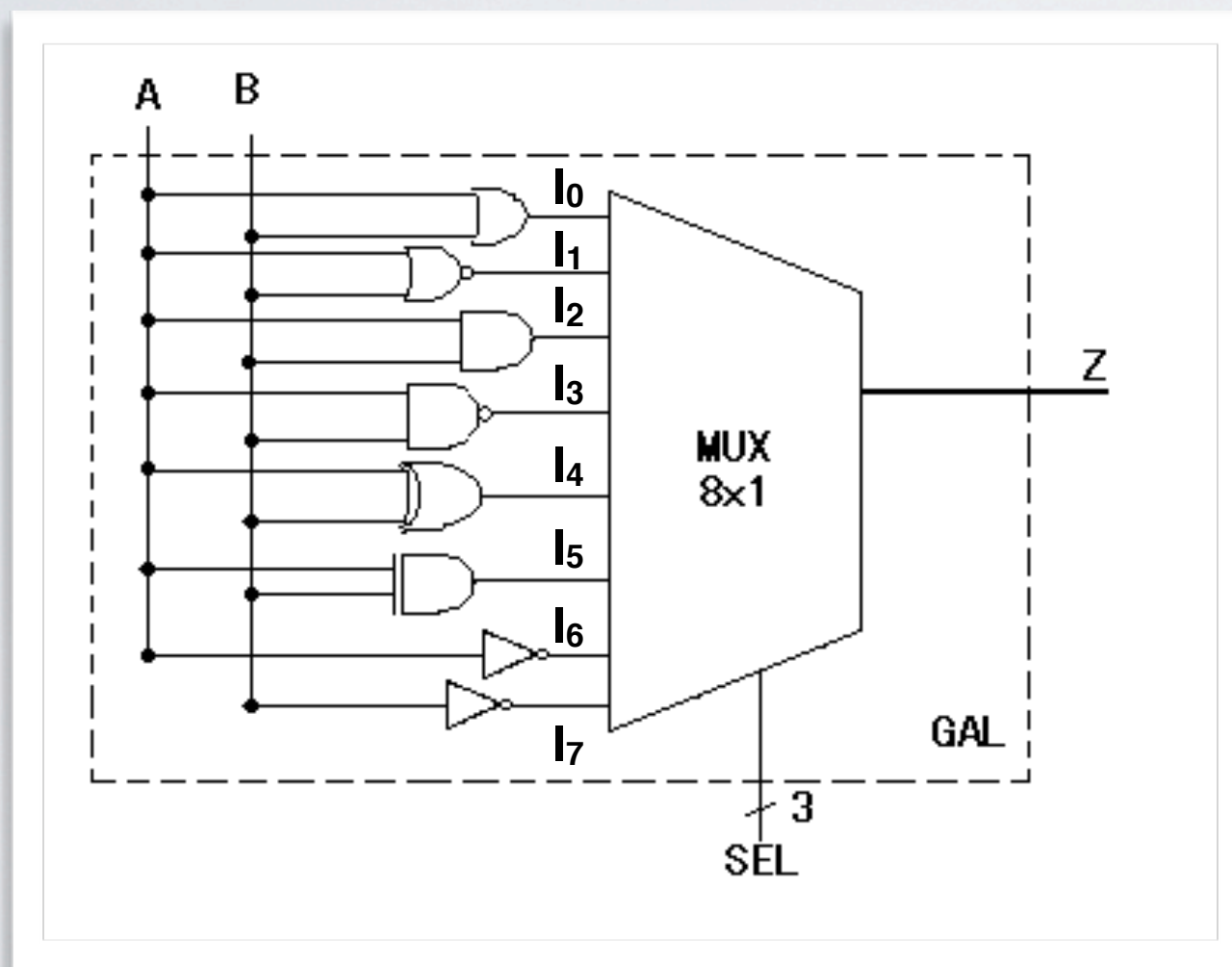
        CASE S IS
        WHEN "00" => Z<=I0;
        WHEN "01" => Z<=I1;
        WHEN "10" => Z<=I2;
        WHEN OTHERS=> Z<=I3;

        END CASE;

    END PROCESS;
END A_MUX;
```


Multiplexor

Práctica: describir en VDHL un MUX 8x1 en cuyas entradas estén conectadas una de las compuestas lógicas básicas. En este caso las entradas serán A y B, necesarias para comprobar la tabla de verdad de cada una de las compuertas.



SEL	Z
000	A OR B
001	A NOR B
010	A AND B
011	A NAND B
100	A XOR B
101	A XNOR B
110	NOT A
111	NOT B

DEMULTIPLEXOR

FUNDAMENTOS DE DISEÑO DIGITAL
OPTATIVA I. ISISA

DEMULTIPLEXOR

- Un **demultiplexor** efectúa la operación contraria al MUX.
- Toma una fuente de datos de entrada y la distribuye selectivamente a uno de los N canales de salida.
- El código de las entradas de selección determina hacia que salida se transmitirá la entrada de datos.

