

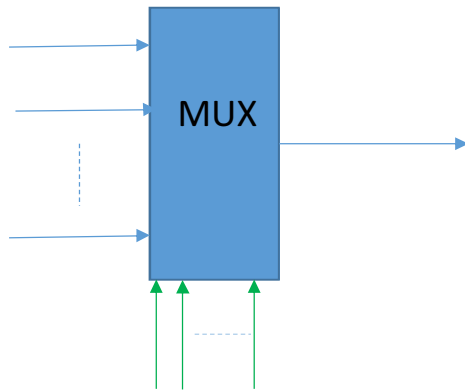
6.-DISEÑO DE MULTIPLEXORES

a)Conceptos

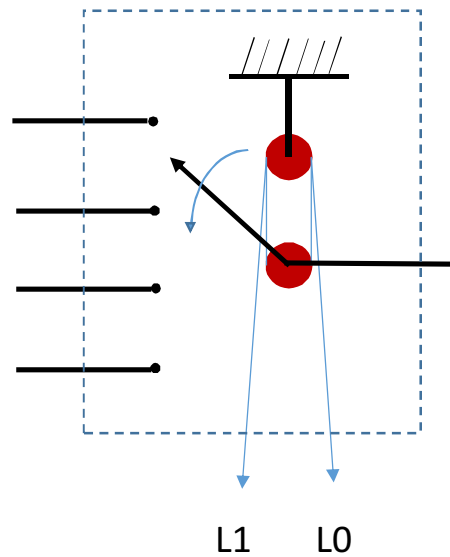
-ES UN SELECTOR DE LÍNEAS DE ENTRADA, SELECCIONA INFORMACION BINARIA PROVENIENTE DE DIFERENTES DISPOSITIVOS

-Multiplexa señales binarias

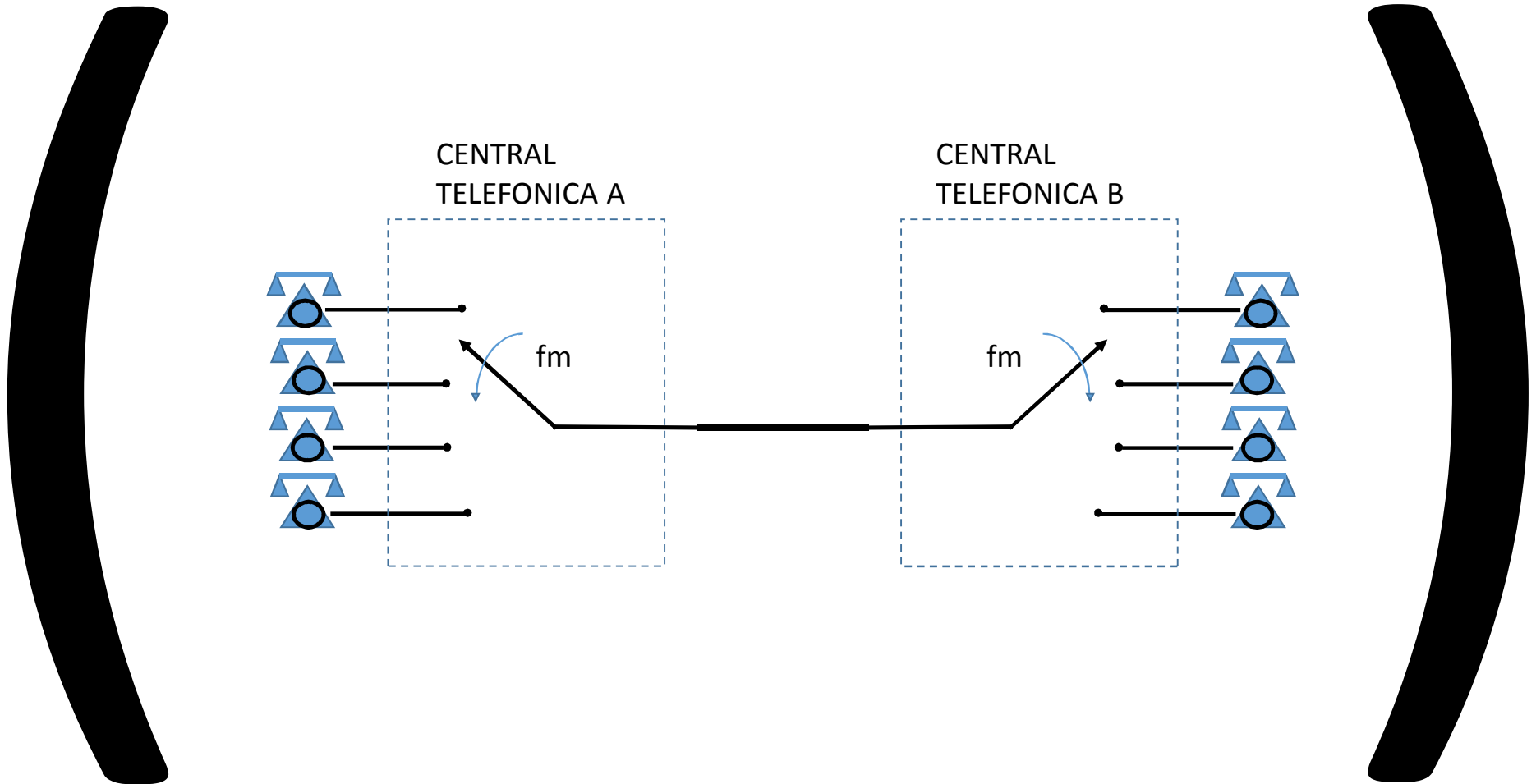
-Es un convertidor paralelo-serie con la arquitectura complementaria



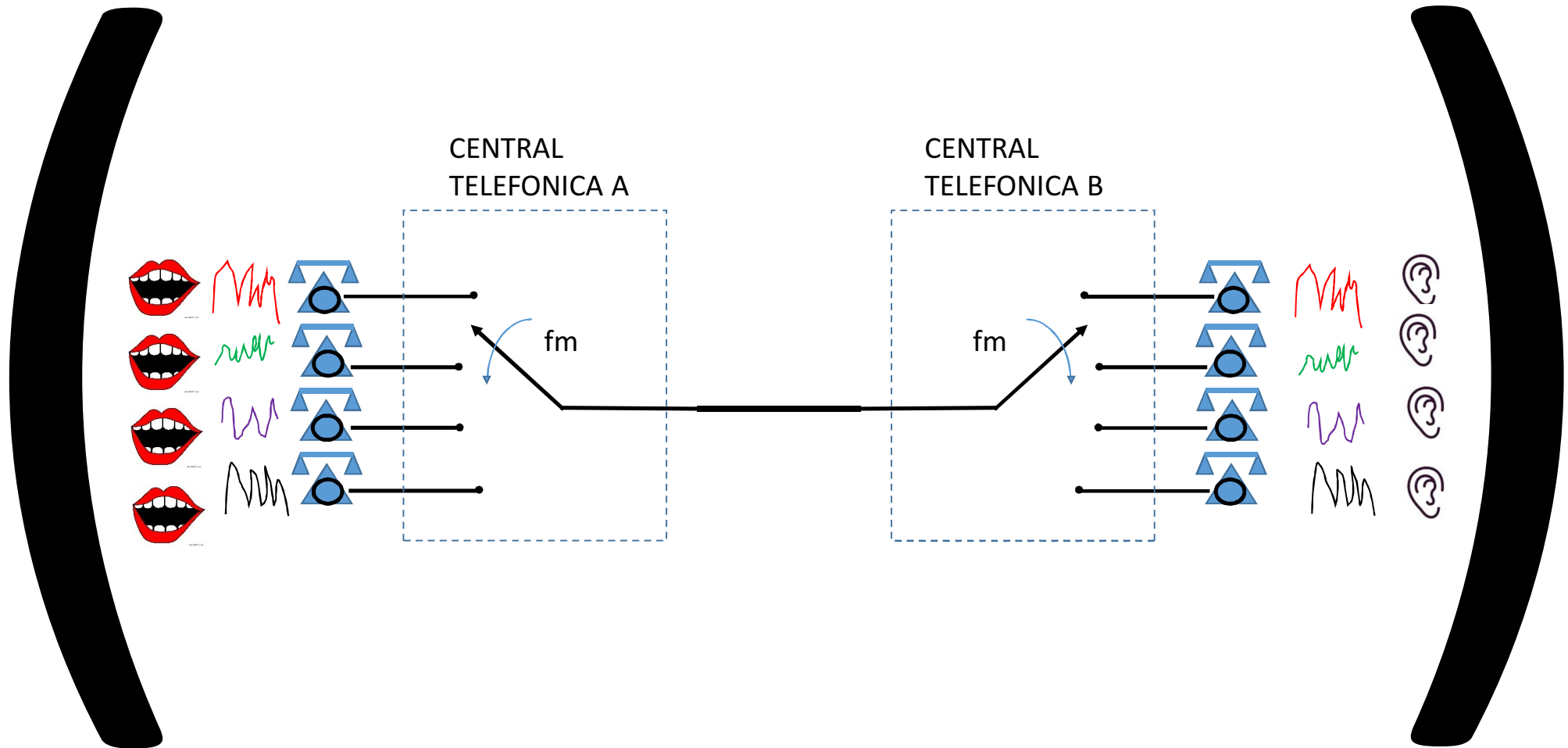
MULTIPLEXOR MECANICO



PARENTESIS



PARENTHESIS



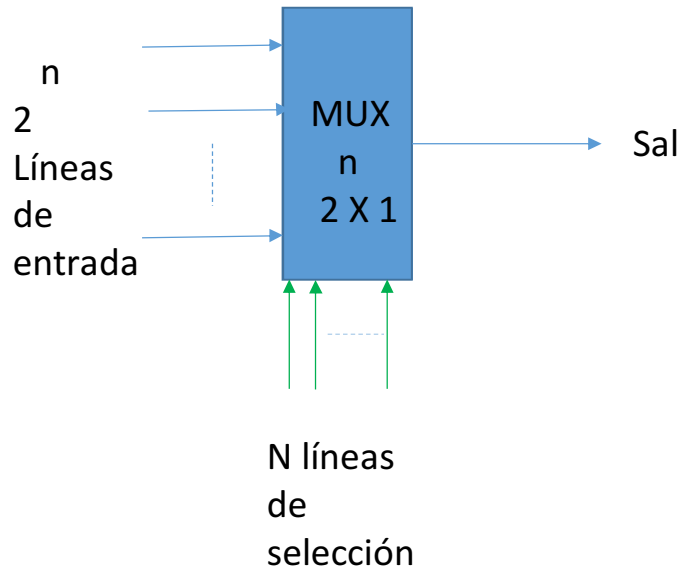
6.-DISEÑO DE MULTIPLEXORES

a)Conceptos

-**ES UN SELECTOR DE LÍNEAS DE ENTRADA, SELECCIONA INFORMACION BINARIA PROVENIENTE DE DIFERENTES DISPOSITIVOS**

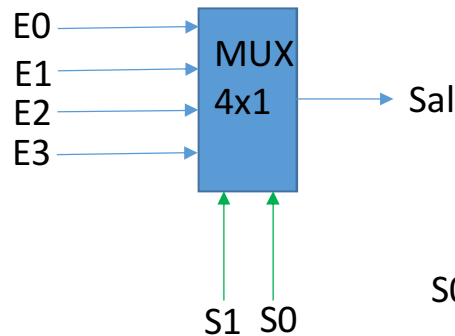
-Multiplexa señales binarias

-Es un convertidor paralelo-serie con la arquitectura complementaria



Ejemplo: Diseñar un MUX 4X1

PRIMER PASO (Núm de var E/S)



SEGUNDO PASO (Tab. De verdad)

SI	SO	SaL
0	0	E0
0	1	E1
1	0	E2
1	1	E3

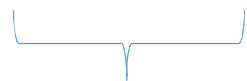
TERCER PASO
(minimización)

	S1		
S0	<div></div>	<div>0</div>	<div>1</div>
	<div>0</div>	E0	E2
	<div>1</div>	E1	E3

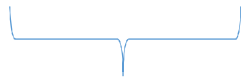
$$Sal = E0S1'S0' + E1S1'S0 + E2S1S0' + E3S1S0$$

¿QUE OBSERVAN?

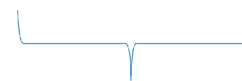
$$SaI = E0S1'S0' + E1S1'S0 + E2S1S0' + E3S1S0$$



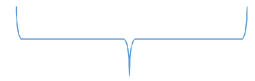
min 0



min 1



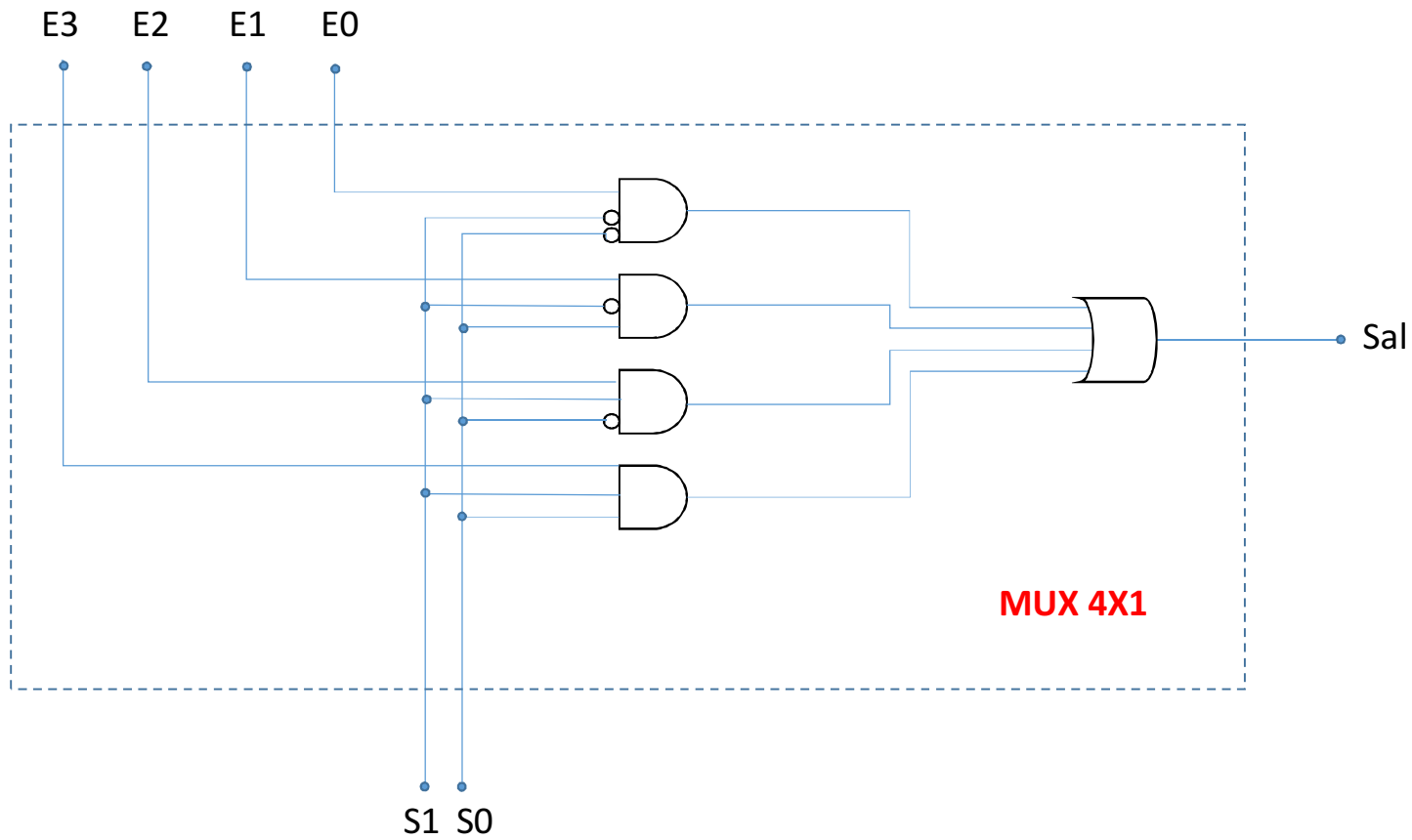
min 2



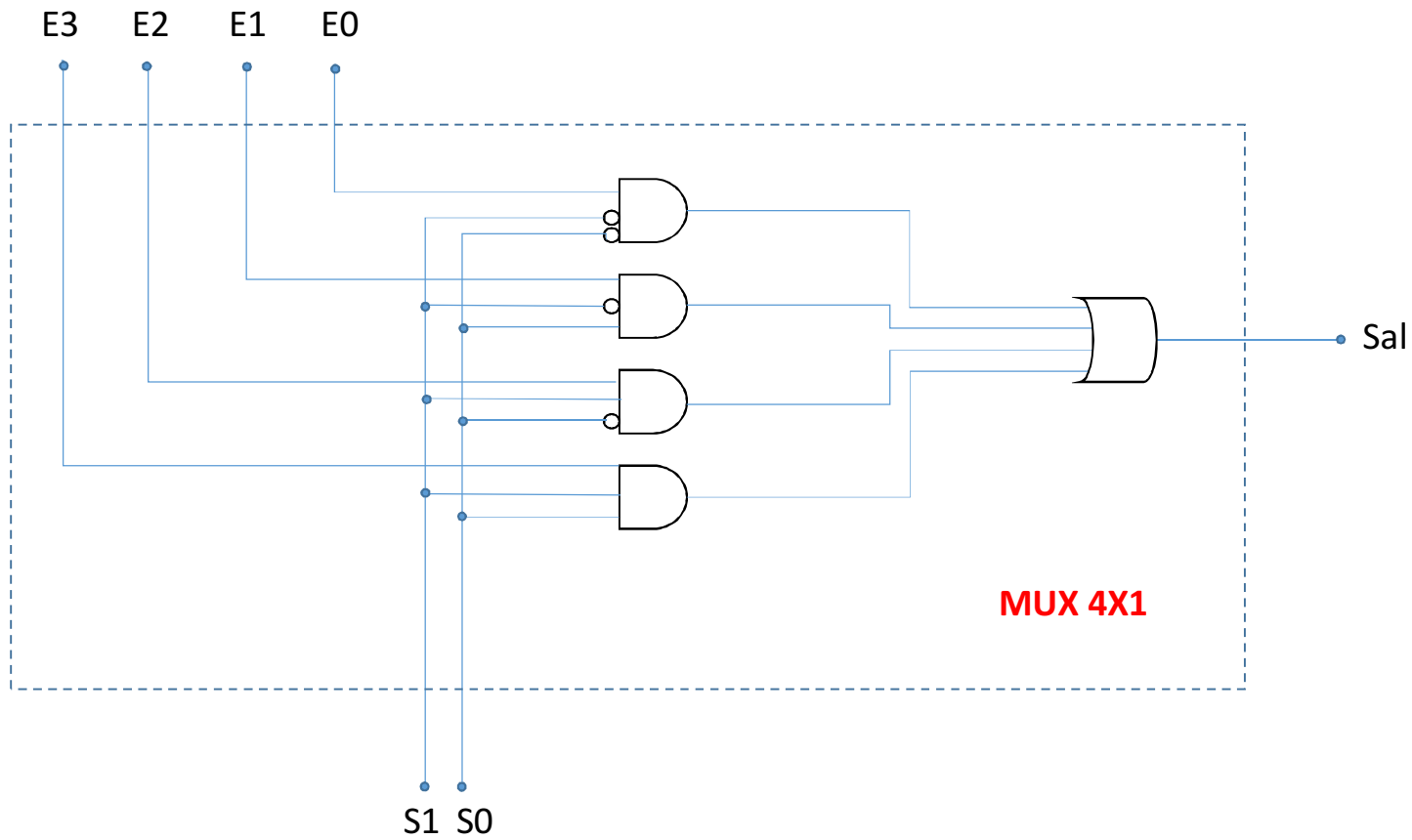
min 3

“SON MINTERMINOS”

CUARTO PASO
(implementación)



CUARTO PASO
(implementación)

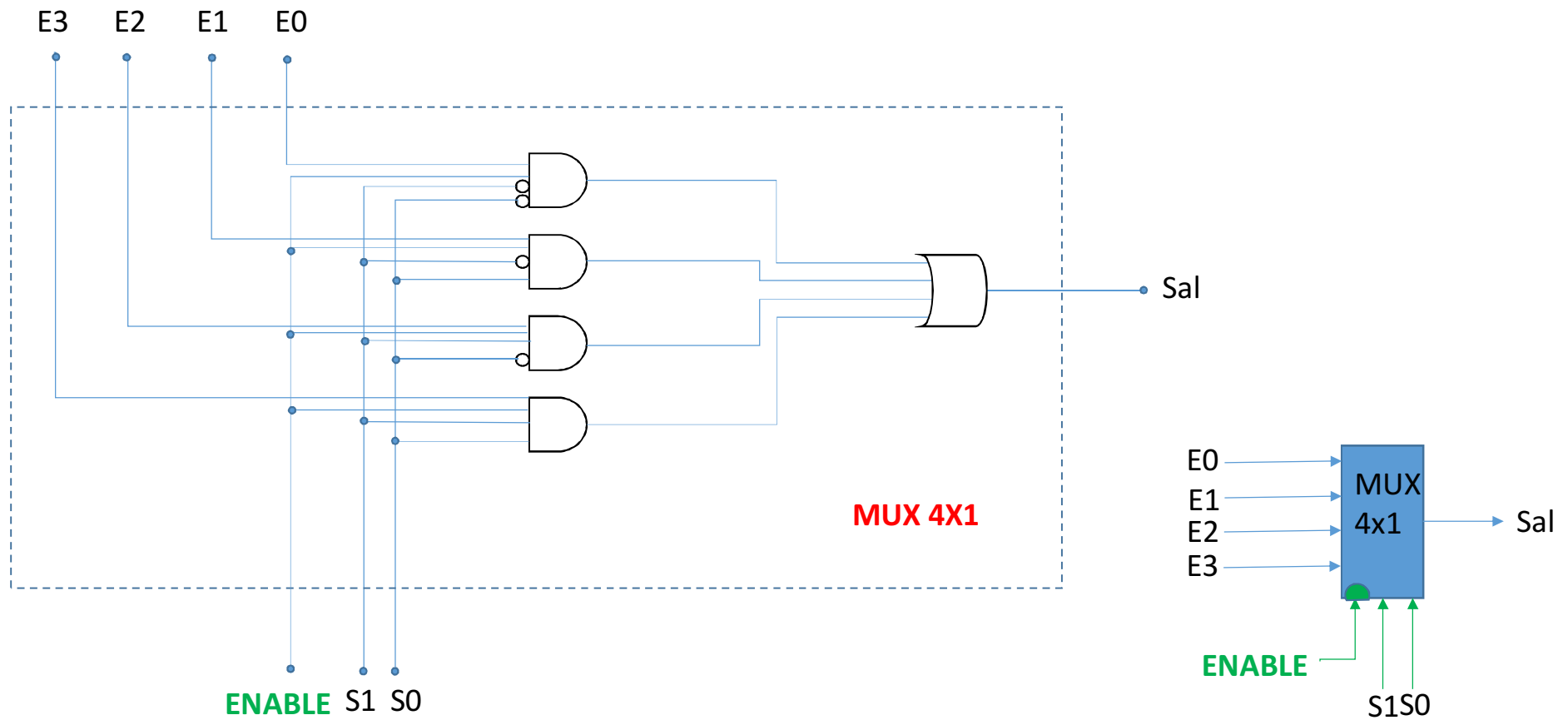


EJEMPLO 2: DISEÑAR UN MUX 8X1

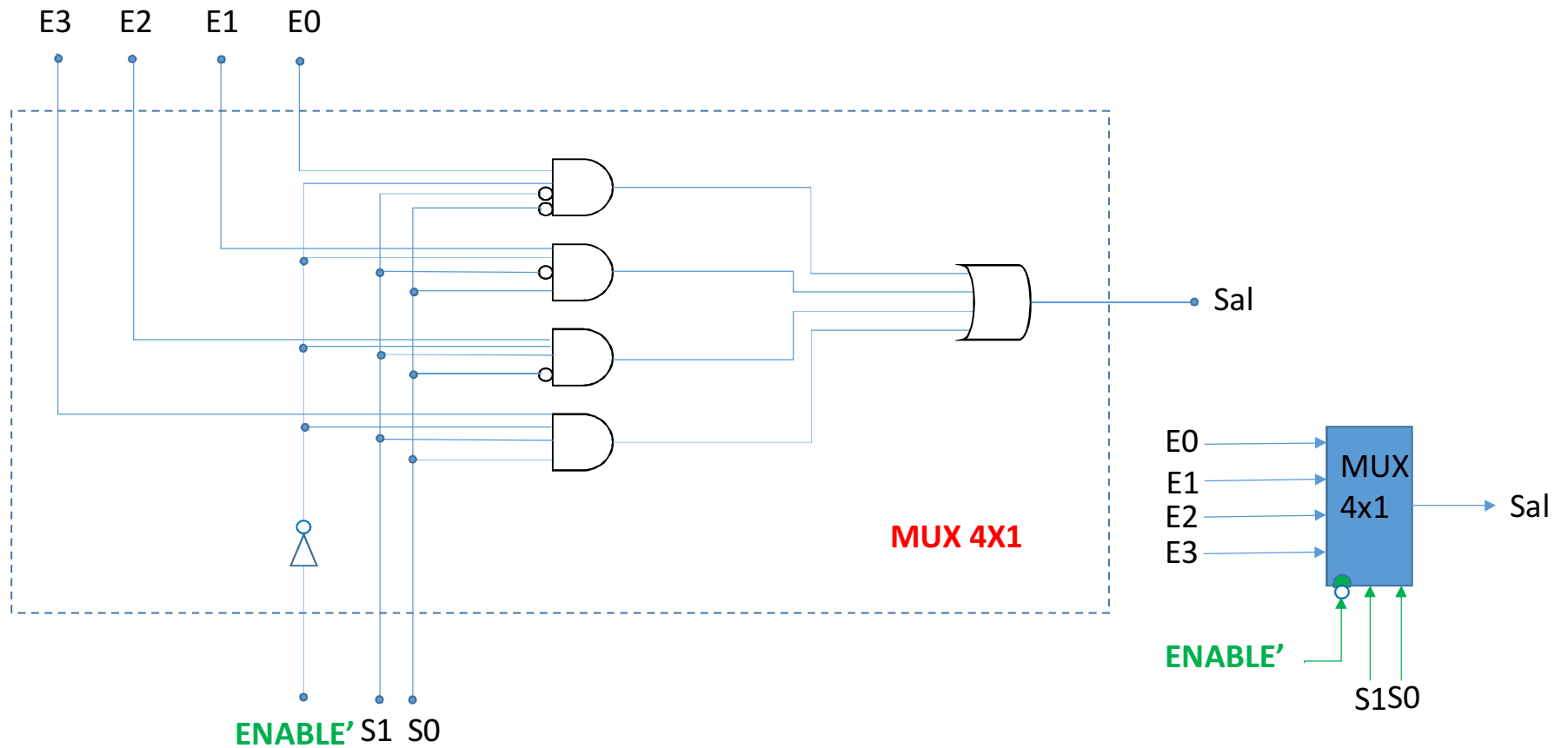
TAREA!!!!!!!!!!!!

PROF: ING. ROBERTO FEDERICO MANDUJANO WILD

EJEMPLO 3: DISEÑAR UN MUX 4X1 CON SEÑAL DE ENABLE



EJEMPLO 3: DISEÑAR UN MUX 4X1 CON SEÑAL DE ENABLE O BIEN ENABLE NEGADA

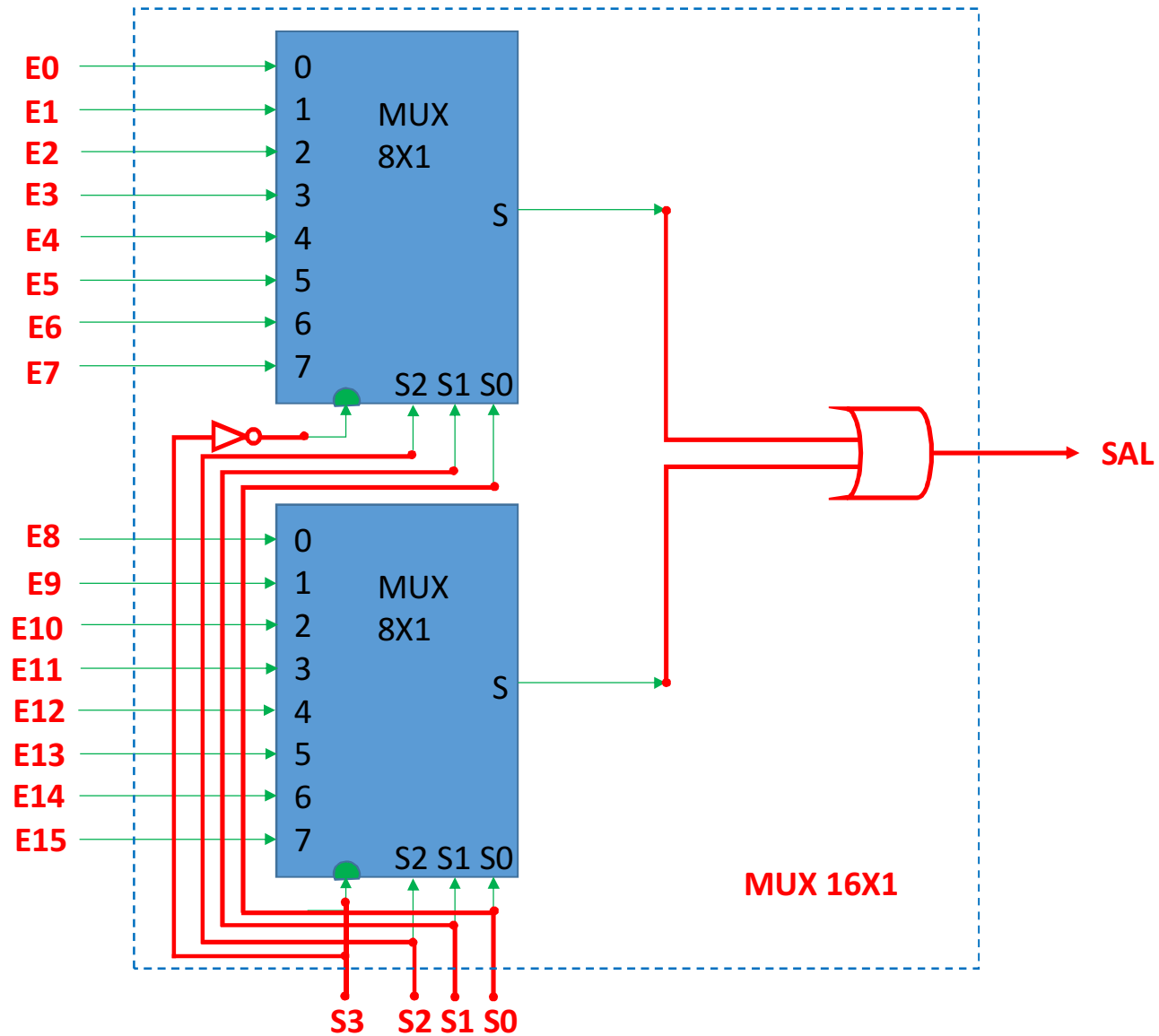


b)Diseño con Multiplexores

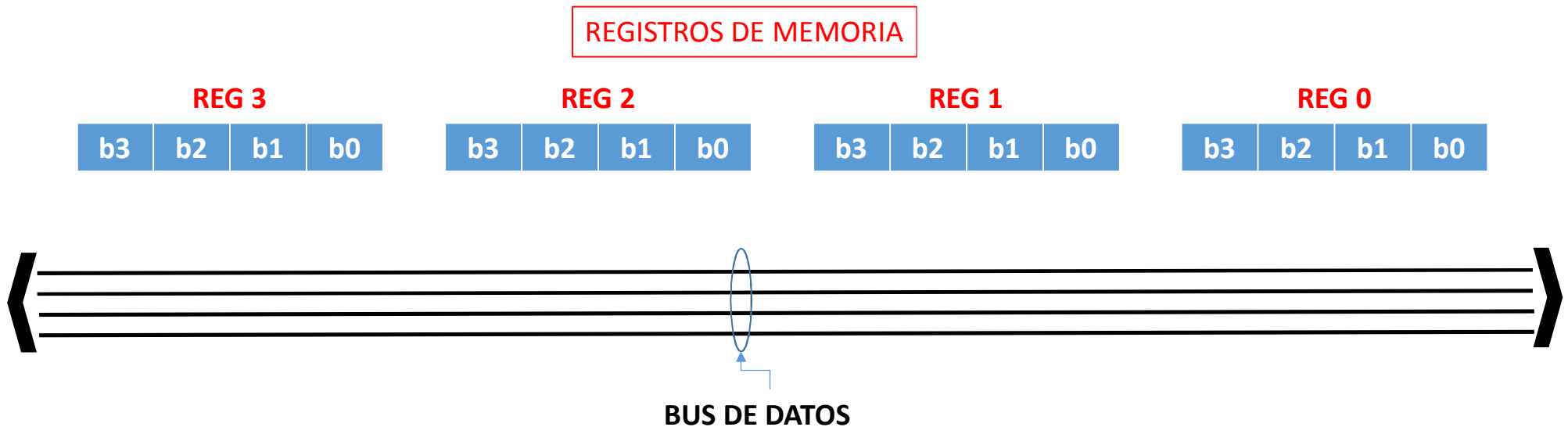
Puesto que los Multiplexores los podemos ya encontrar integrados en una sola componente (CHIP), y pertenecen a los circuitos MSI, podemos diseñar **CON** ellos.

EJEMPLO 1: DISEÑAR UN MUX 16X1 CON DOS MUX 8X1 CON SEÑAL DE ENABLE

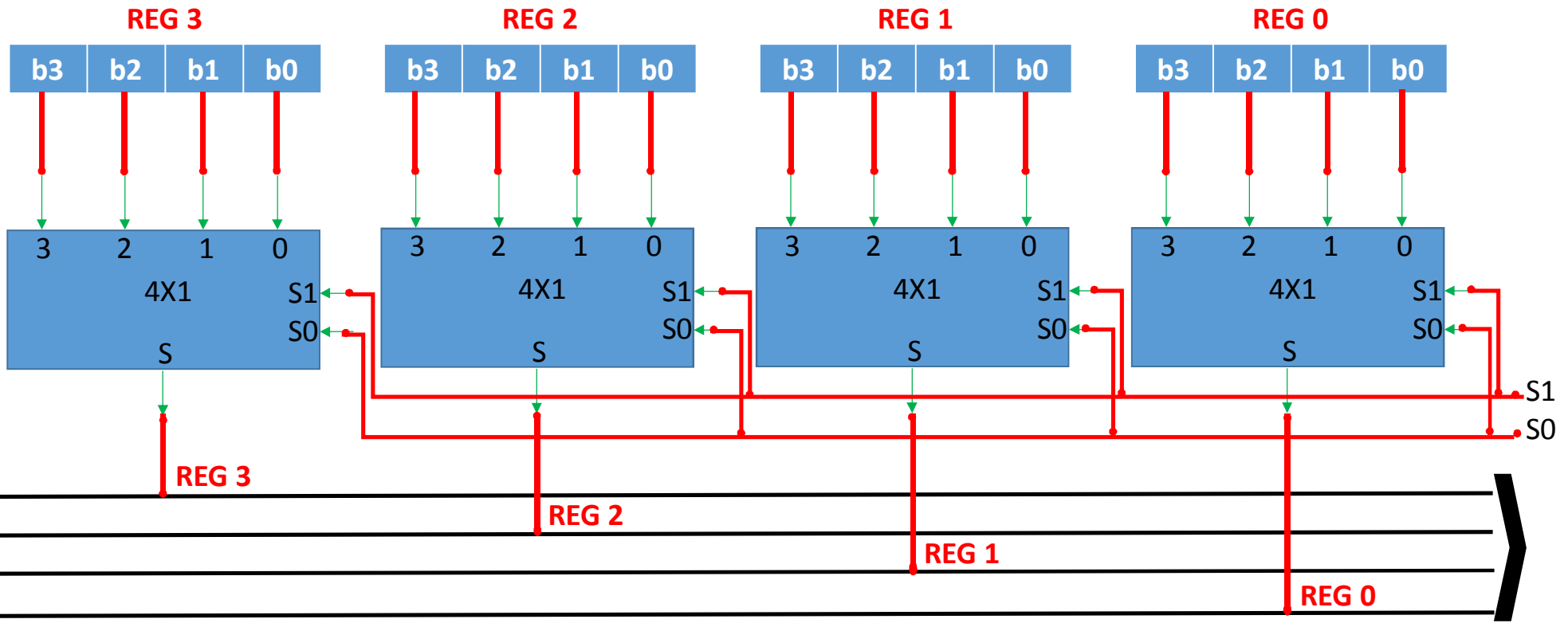
IMPLEMENTACION:



EJEMPLO 2: CONECTAR CUATRO REGISTROS DE MEMORIA DE 4 BITS CADA UNO A UN SOLO BUS DE DATOS DE 4 LINEAS,
a) TRANSMITIENDO LA INFORMACION DE LOS CUATRO REGISTROS A LA VEZ.
b) TRANSMITIENDO LA INFORMACION DE UN REGISTRO A LA VEZ



EJEMPLO 2: a) TRANSMITIENDO LA INFORMACION DE LOS CUATRO REGISTROS A LA VEZ.

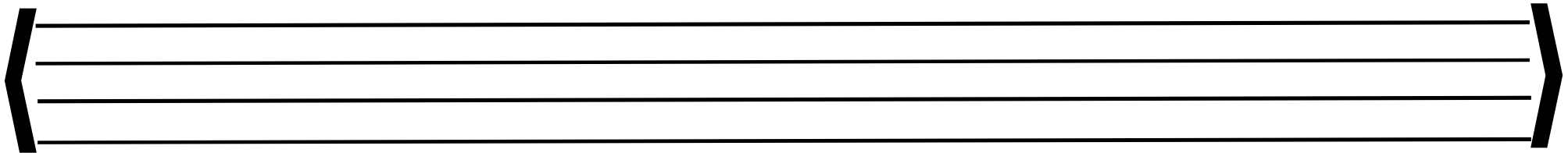


NOTA: SI PONEMOS EN LAS LINEAS DE SELECCION S1S0, UN CONTADOR CON LA SECUENCIA: 00,01,10,11 SE CONVIERTE EN UN CONVERTIDOR PARALELO-SERIE

EJEMPLO 2: b) TRANSMITIENDO LA INFORMACION DE UN REGISTRO A LA VEZ



TAREA!!!!!!!!!!!!!!



EJEMPLO 3: DISEÑAR UN SISTEMA DE TRANSFERENCIA LOGICA ENTRE REGISTROS DE UNA UNIDAD DE MEMORIA DE 4 REGISTROS DE 2 bits C/U MEDIANTE UN BUS COMUN

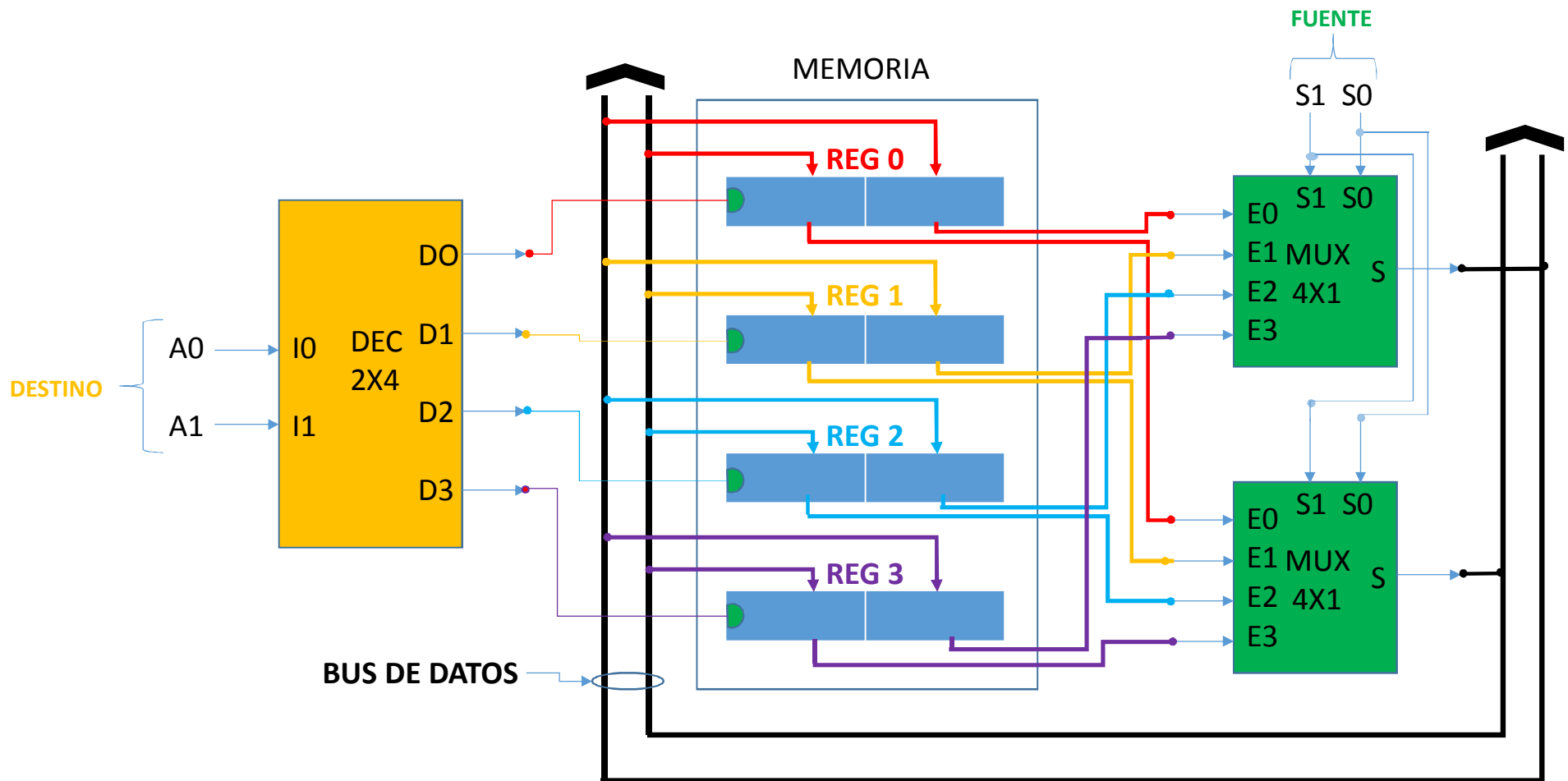


NOTA: LA OPERACIÓN A REALIZAR SE LLAMA “FUNCION TRANSFERENCIA” Y SE REPRESENTA:

$\text{REGx} \leq \text{REGy}$ (Se transfiere al REGx el contenido del REGy)

ESTA SIMBOLOGIA ES USADA EN EL LLAMADO “LENGUAJE DE TRANSFERENCIA LOGICA” Y SE UTILIZA EN EL DISEÑO DE SISTEMAS DIGITALES AVANZADOS

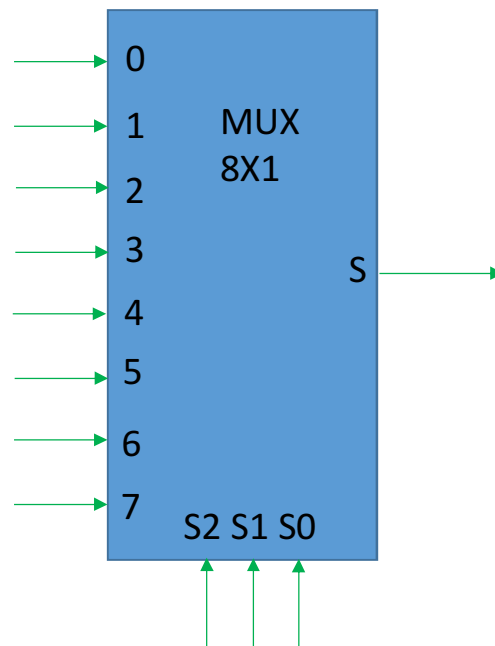
EJEMPLO 3: DISEÑAR UN SISTEMA DE TRANSFERENCIA LOGICA ENTRE REGISTROS DE UNA UNIDAD DE MEMORIA DE 4 REGISTROS DE 2 bits C/U MEDIANTE UN BUS COMUN



EJEMPLO 4: IMPLEMENTAR LA SIGUIENTE FUNCION BOOLEANA UTILIZANDO:

a) USANDO UN MUX 8X1

X	Y	Z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



PODEMOS OBSERVAR QUE LA FUNCION BOOLEANA ES DEL TAMAÑO DEL MULTIPLEXOR MUX 8X1, ES DECIR:

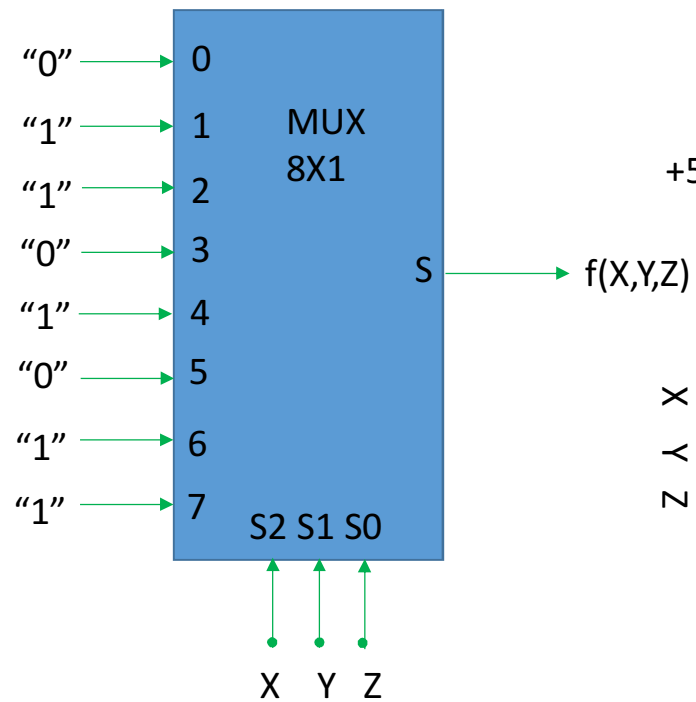
- 1.-NUMERO DE MINTERMINOS 8 = NUMERO DE ENTRADAS DEL MUX 8X1**
- 2.-NUMERO DE VARIABLES 3 = NUMERO DE LINEAS DE SELECCIÓN DEL MUX 8X1**
- 3.-NUMERO DE FUNCIONES A IMPLEMENTAR 1 = UNA SOLA SALIDA DEL MUX 8X1**

EJEMPLO 4: IMPLEMENTAR LA SIGUIENTE FUNCION BOOLEANA UTILIZANDO:

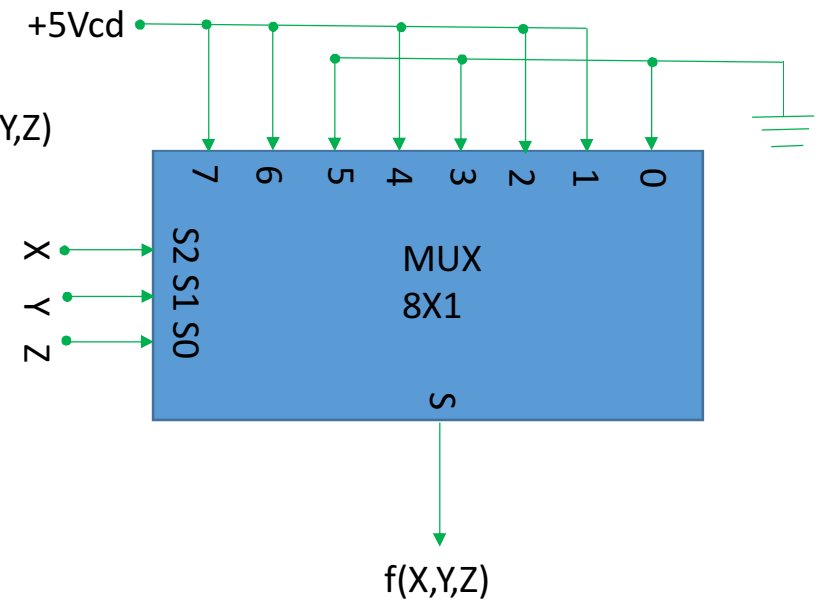
a) USANDO UN MUX 8X1

POR LO TANTO LA IMPLEMENTACION SERIA:

X	Y	Z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



O BIEN: "FISICAMENTE", SERIA:

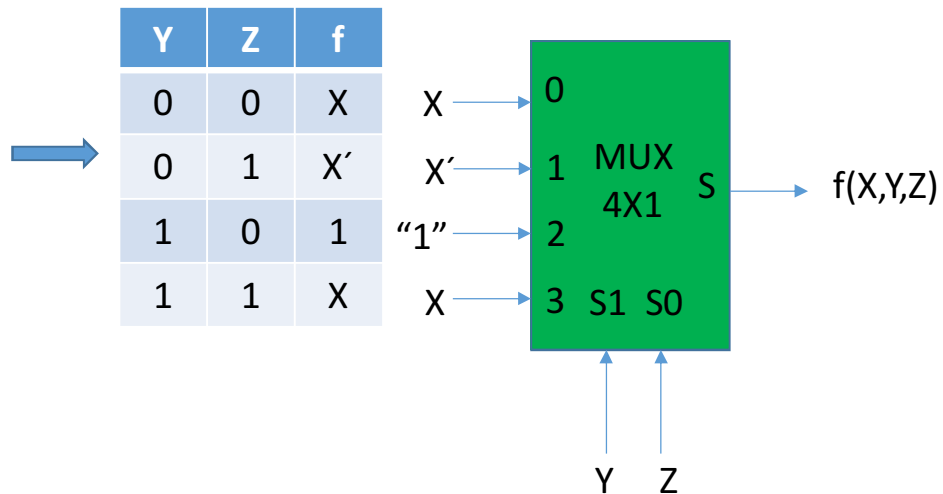


EJEMPLO 4: IMPLEMENTAR LA SIGUIENTE FUNCION BOOLEANA UTILIZANDO:

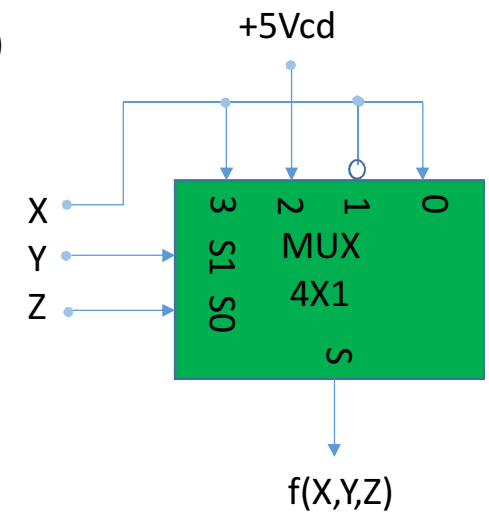
b) USANDO UN MUX 4X1

¡AH!....REDUZCO LA TABLA AL TAMANO DEL MUX 4X1
....Y CONTINUO CON LA IMPLEMENTACION

X	Y	Z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



Y "FISICAMENTE", SERIA:

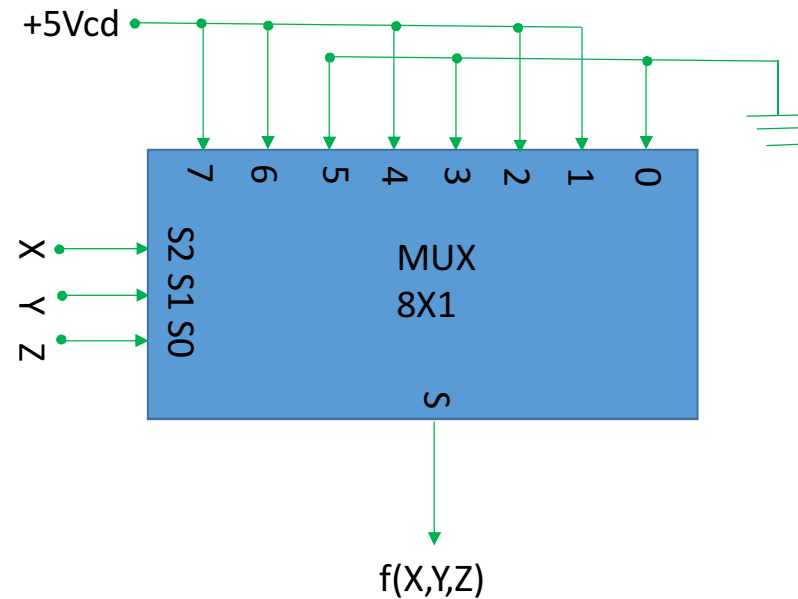


EJEMPLO 5: DISEÑAR UN DETECTOR DE MULTIPLOS DE 3 EN PALABRAS DE 4 BITS (A,B,C,D)
UTILIZANDO UN MUX 4X1.
a) USANDO COMO LINEAS DE SELECCION LAS VARIABLES A Y B
b) USANDO COMO LINEAS DE SELECCION LAS VARIABLES C Y D

TAREA!!!!!!!!!!!!!!

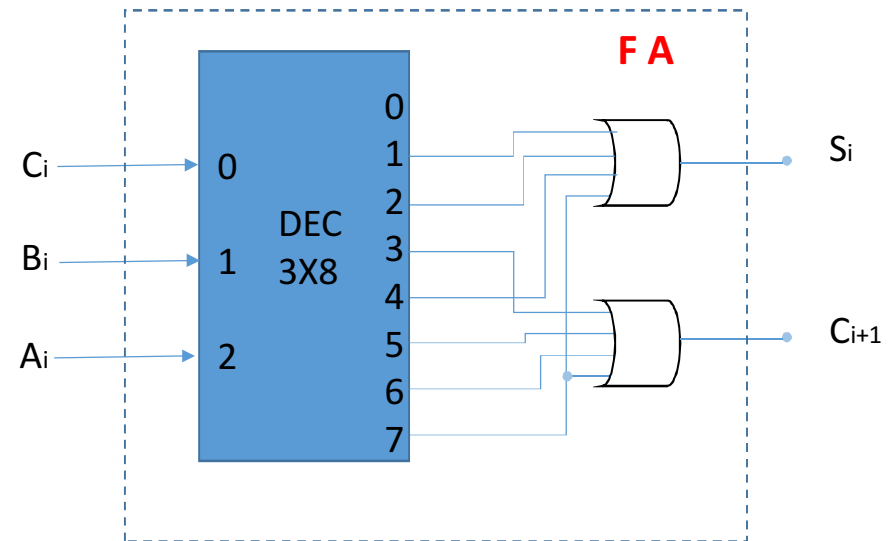
¡OJO!, PODEMOS OBSERVAR QUE LOS MULTIPLEXORES ME SIRVEN PARA IMPLEMENTAR UNA SOLA FUNCION BOOLEANA, MIENTRAS QUE LOS DECODIFICADORES, ME SIRVEN PARA IMPLEMENTAR VARIAS FUNCIONES BOOLEANAS

X	Y	Z	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



SI RECUERDAN EL FA IMPLEMENTADO CON UN DEC 3X8

A _i	B _i	C _i	C _{i+1}	S _i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



$$C_{i+1} = \sum m(3, 5, 6, 7)$$

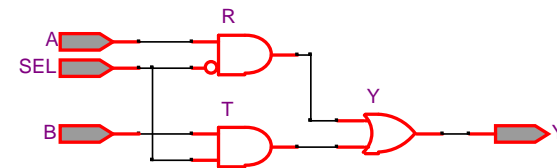
$$S_i = \sum m(1, 2, 4, 7)$$

EJEMPLOS CON LA TARJETA MAX (ALTERA)

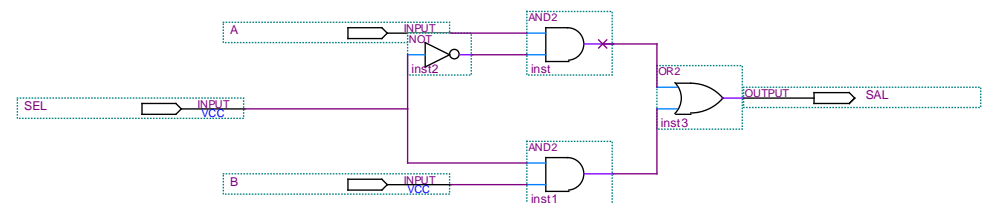
EJEMPLO 1: MUX 2X1

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY MUX2X1 IS
    PORT(A,B,SEL: IN STD_LOGIC;
         Y: OUT STD_LOGIC);
END ENTITY;
ARCHITECTURE MUX2X1 OF MUX2X1 IS
    SIGNAL R,T: STD_LOGIC;
BEGIN
    R<=A AND (NOT SEL);
    T<=B AND SEL;
    Y<=R OR T;
END ARCHITECTURE;
```

IMPLEMENTACION CON VHDL



IMPLEMENTACION GRAFICA



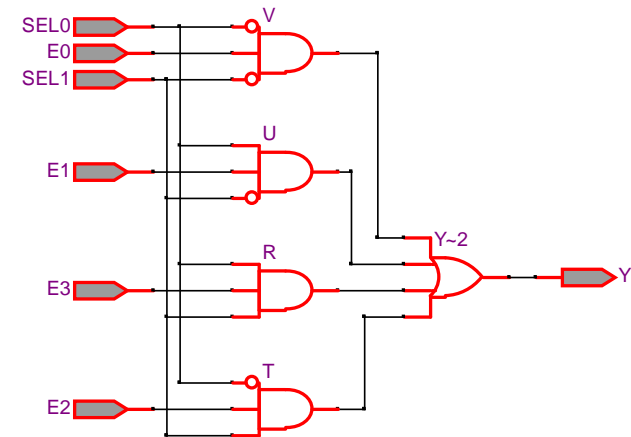
EJEMPLO 2: MUX 4X1

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

ENTITY MUX4X1 IS
    PORT(E3,E2,E1,E0,SEL1,SEL0: IN STD_LOGIC;
          Y: OUT STD_LOGIC);
END ENTITY;

ARCHITECTURE MUX4X1 OF MUX4X1 IS
    SIGNAL R,T,U,V: STD_LOGIC;
BEGIN
    R<=E3 AND SEL1 AND SEL0;
    T<=E2 AND SEL1 AND (NOT SEL0);
    U<=E1 AND (NOT SEL1) AND SEL0;
    V<=E0 AND (NOT SEL1) AND (NOT SEL0);
    Y<=R OR T OR U OR V;
END ARCHITECTURE;
```

IMPLEMENTACION CON VHDL



EJEMPLO 3: MUX 4X1 DE 4 BITS

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

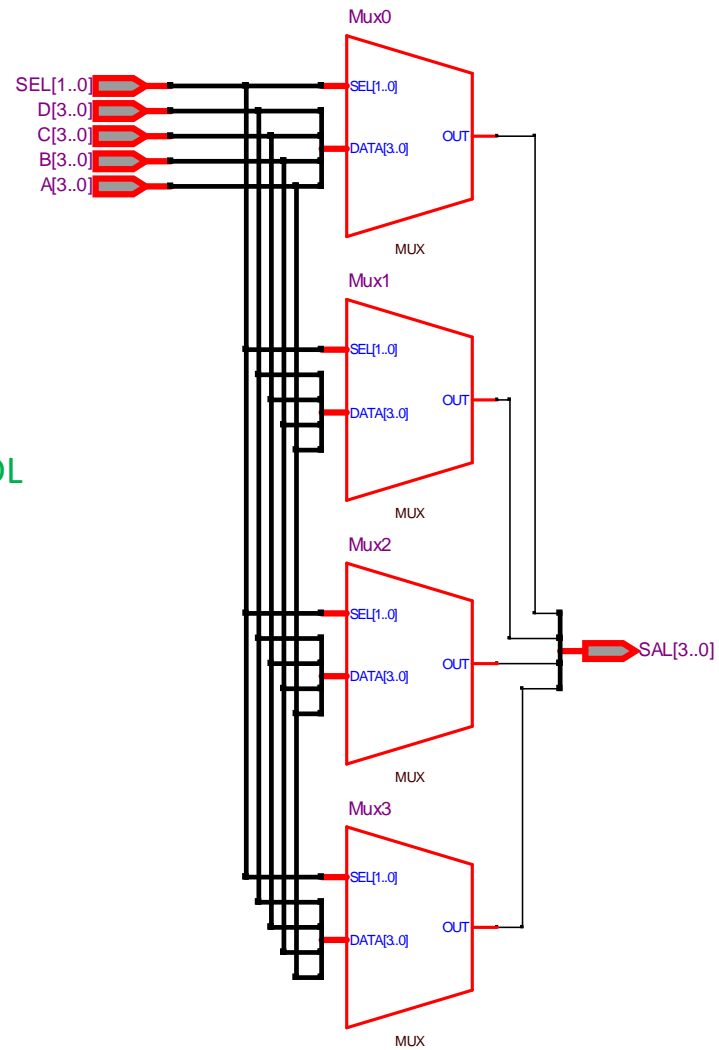
--TIPO VECTOR
--ENTITY MUX4X1_4BITS IS
--      PORT(A,B,C,D: IN STD_LOGIC_VECTOR (3 DOWNT0 0);
--      SAL: OUT STD_LOGIC_VECTOR (3 DOWNT0 0);
--      SEL: IN STD_LOGIC_VECTOR (1 DOWNT0 0));
--END ENTITY;
--ARCHITECTURE MUX4X1_4BITD OF MUX4X1_4BITS IS
--      BEGIN
--
--          SAL<=A WHEN SEL="00" ELSE
--          B WHEN SEL="01" ELSE
--          C WHEN SEL="10" ELSE
--          D;
--END ARCHITECTURE;
```

 continúa

sigue 

```
--SEÑALES TIPO ENTERO
ENTITY MUX4X1_4BITS IS
      PORT(A,B,C,D: IN INTEGER RANGE 0 TO 15;
      SAL: OUT INTEGER RANGE 0 TO 15;
      SEL: IN INTEGER RANGE 0 TO 3);
END ENTITY;
ARCHITECTURE MUX4X1_4BITS OF MUX4X1_4BITS IS
      BEGIN
          WITH SEL SELECT
              SAL<=A WHEN 0,
              B WHEN 1,
              C WHEN 2,
              D WHEN OTHERS;
END ARCHITECTURE;
```

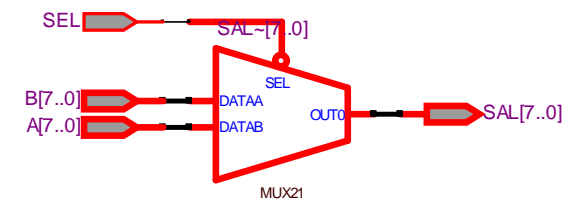
IMPLEMENTACION CON VHDL



EJEMPLO 4: MUX 2X1 DE 8 BITS

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY MUX2X1_8BITS IS
    PORT (A,B: IN STD_LOGIC_VECTOR(7 DOWNTO 0);
          SEL: IN STD_LOGIC;
          SAL: OUT STD_LOGIC_VECTOR(7 DOWNTO
0));
END ENTITY;
ARCHITECTURE MEX2X1_8BITS OF MUX2X1_8BITS IS
    BEGIN
        SAL<=A WHEN SEL='0' ELSE B;
END ARCHITECTURE;
```

IMPLEMENTACION CON VHDL



This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.