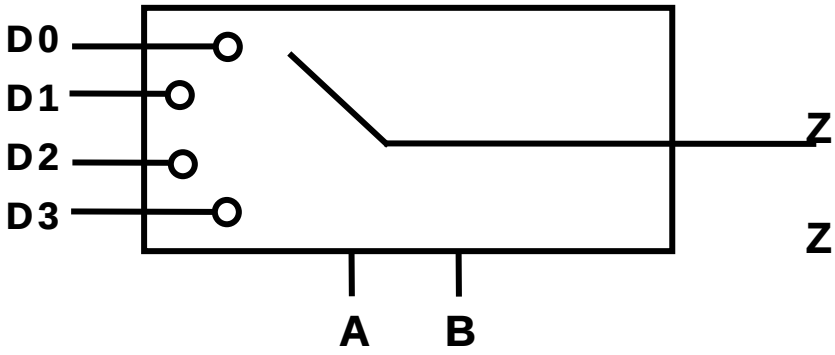
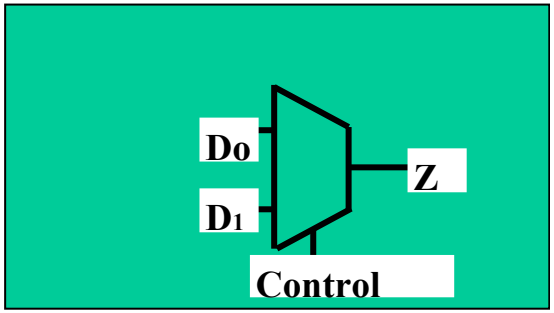
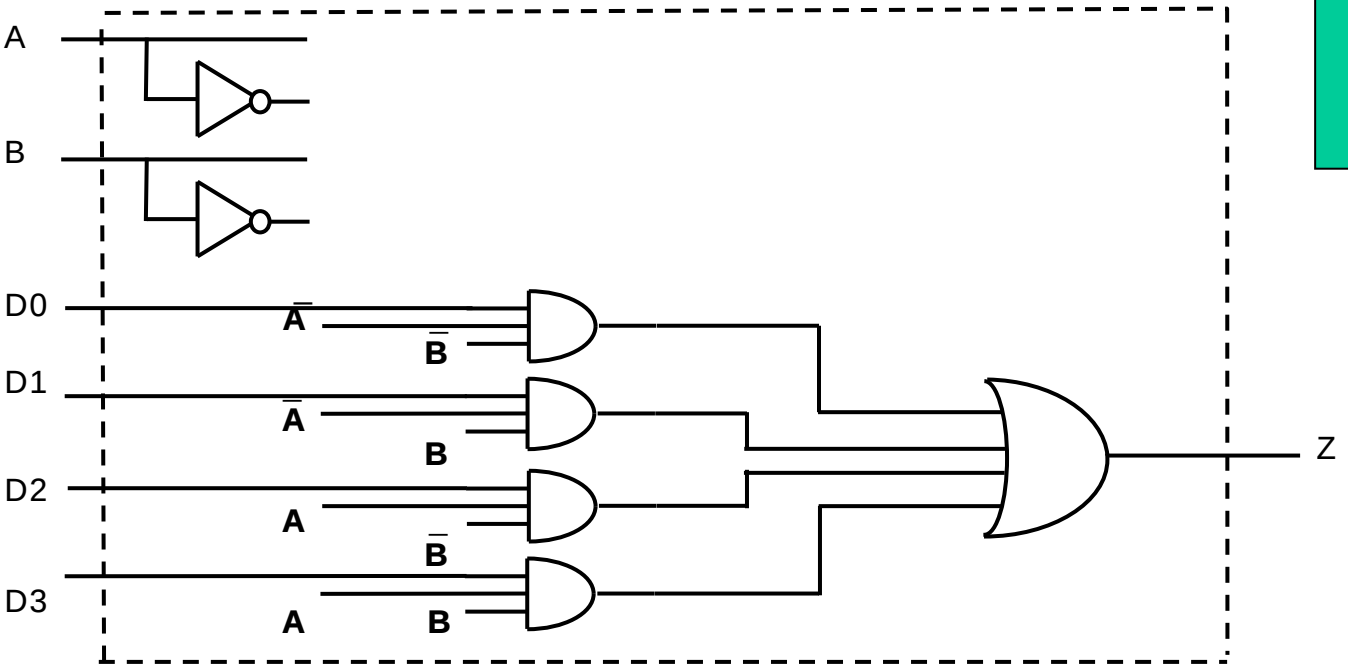
MULTIPLEXOR/SELECTOR/SERIALIZADOR



$$Z = D0\,\bar{A}\,\bar{B} + D1\,\bar{A}\,B + D2\,A\,\bar{B} + D3\,A\,B$$

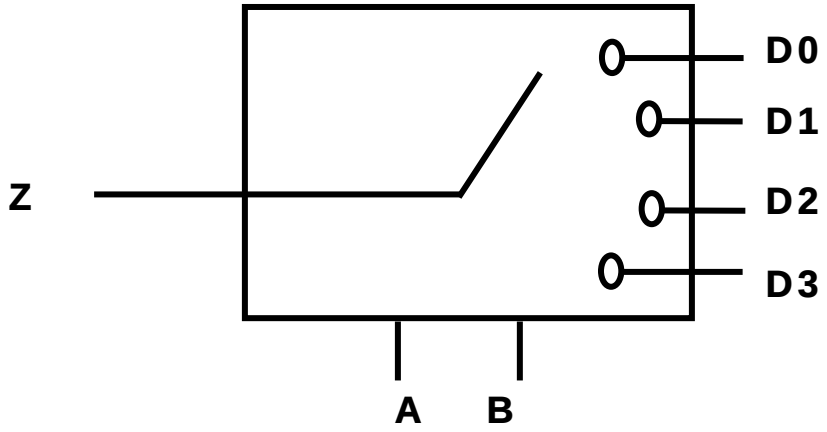
$$Z = D0\,m0 + D1\,m1 + D2\,m2 + D3\,m3$$

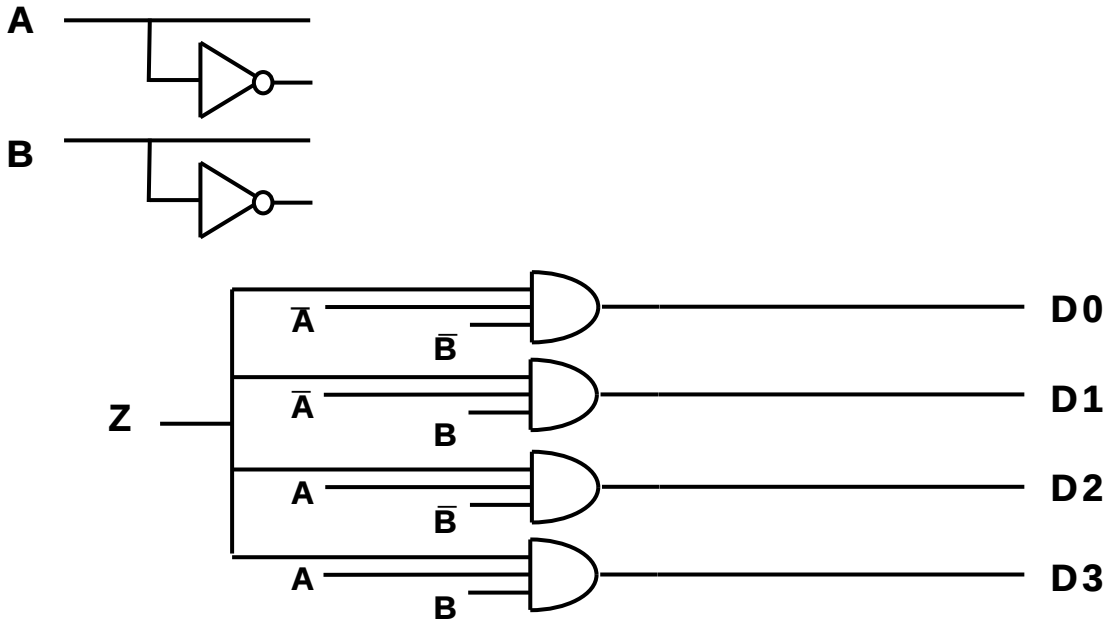




CONTROL		SALIDA
A	B	Z
0	0	D0
0	1	D1
1	0	D2
1	1	D3

DEMULTIPLEXOR/DISTRIBUIDOR/PARALELO





CONTROL		SALIDA			
A	B	D0	D1	D2	D3
0	0	Z	0	0	0
0	1	0	Z	0	0
1	0	0	0	Z	0
1	1	0	0	0	Z

MULTIPLEXOR COMO GENERADOR DE FUNCIONES

ECUACION CARACTERISTICA:

$$Z = D0 \cdot \overline{A}\,\overline{B} + D1 \cdot \overline{A}\,B + D2 \cdot A\,\overline{B} + D3 \cdot A\,B$$

$$Z = D0 \cdot m0 + D1 \cdot m1 + D2 \cdot m2 + D3 \cdot m3$$

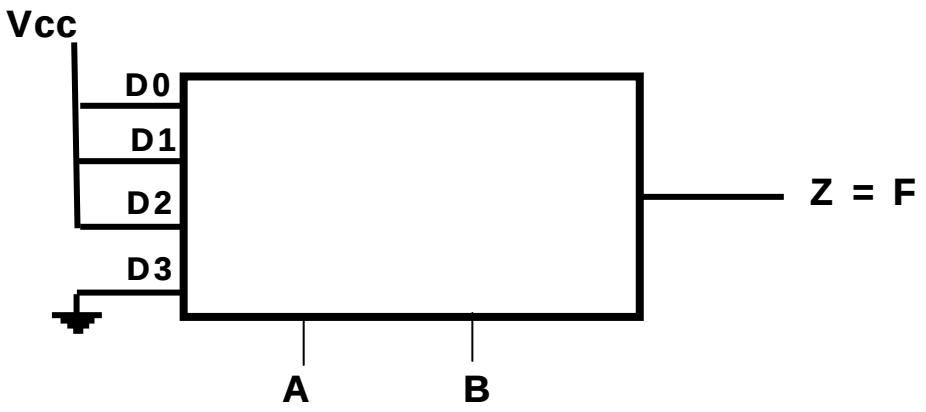
EJEMPLO: SUPONGAMOS LA SIGUIENTE FUNCION...

$$F = \sum 0, 1, 2$$

COMPAREMOS LA ECUACION CARACTERISTICA E IGUALEMOS:

PARA Z = F ⇨

- D0 = 1
- D1 = 1
- D2 = 1
- D3 = 0



EL MULTIPLEXOR PUEDE GENERAR UNA FUNCION CON N + 1 VARIABLES

EJ: $f = \sum 0, 1, 3, 6$

$$f = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C$$
$$m0 \quad . \quad m1 \quad . \quad m3 \quad . \quad m2$$

$$Z = D0 \cdot \bar{B}\bar{C} + D1 \cdot \bar{B}C + D3 \cdot BC + D2 \cdot B\bar{C}$$

SI $f = Z$

$D0 = \bar{A}$

$D1 = \bar{A}$

$D2 = A$

$D3 = \bar{A}$

A \ BC	00	01	10	11
0	1	1	0	1
1	0	0	1	0

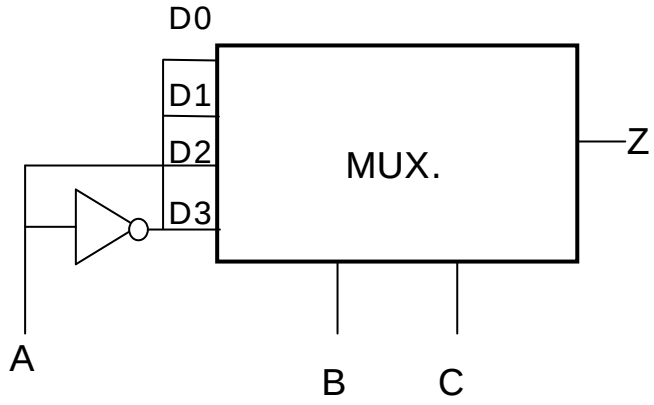
BC	00	01	10	11
	D0	D1	D2	D3

$D0 = \bar{A}$

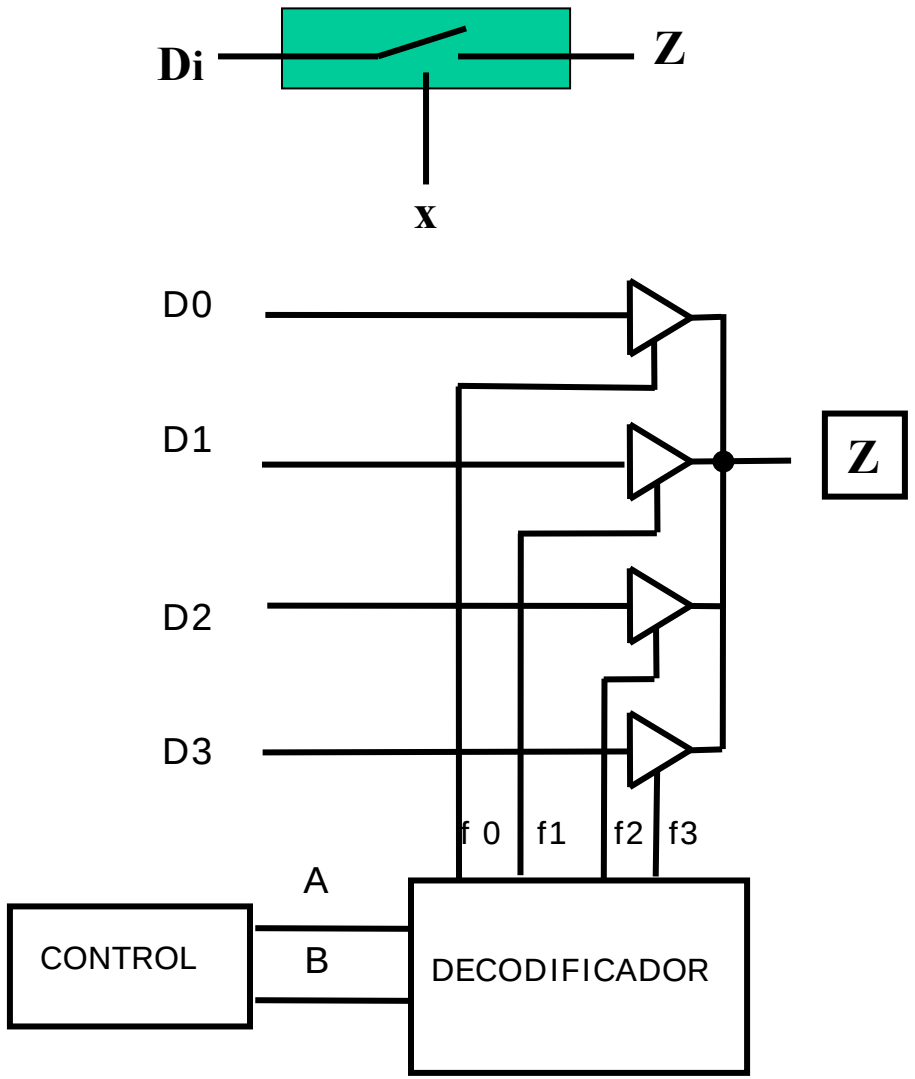
$D1 = \bar{A}$

$D2 = A$

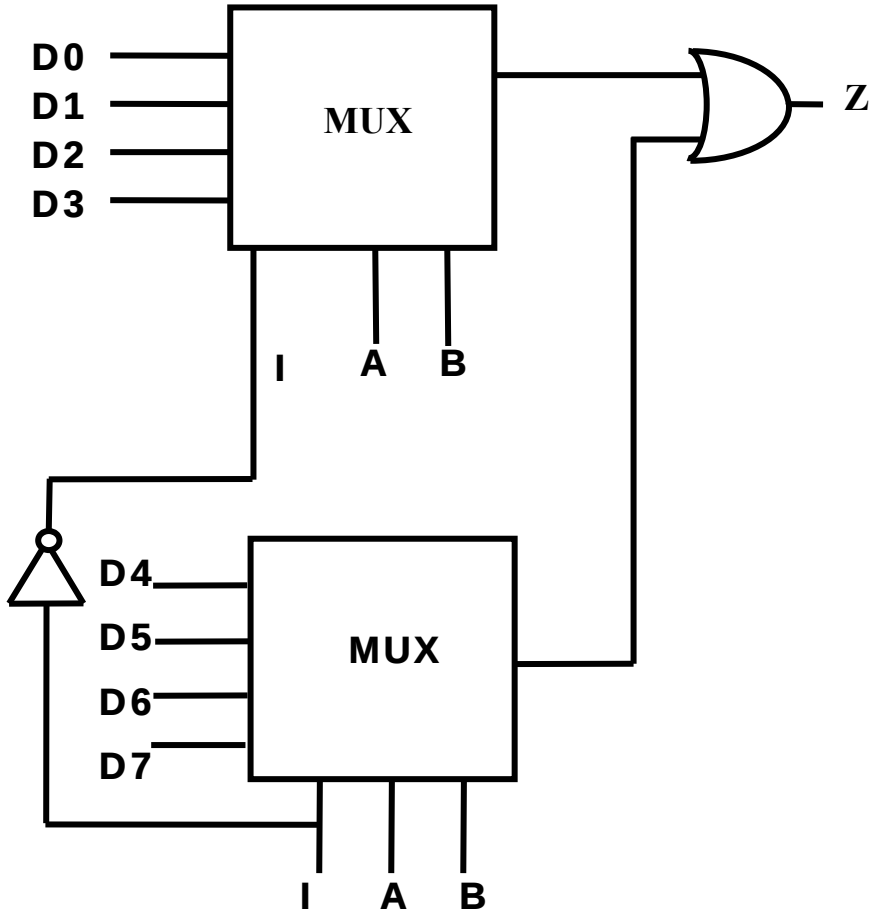
$D3 = \bar{A}$



MULTIPLEXOR CON LOGICA DE TRES ESTADOS



EXPANSION CON MULTIPLEXORES

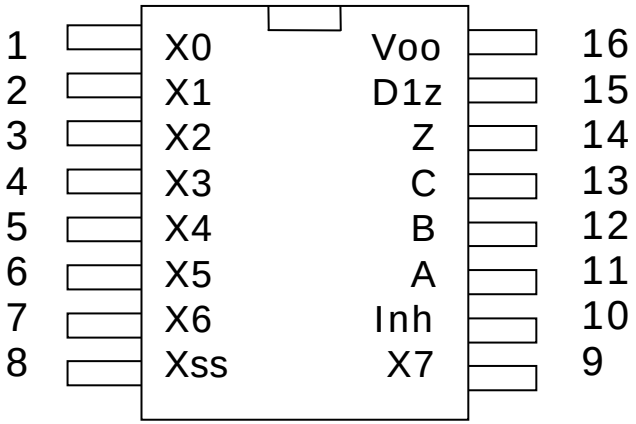


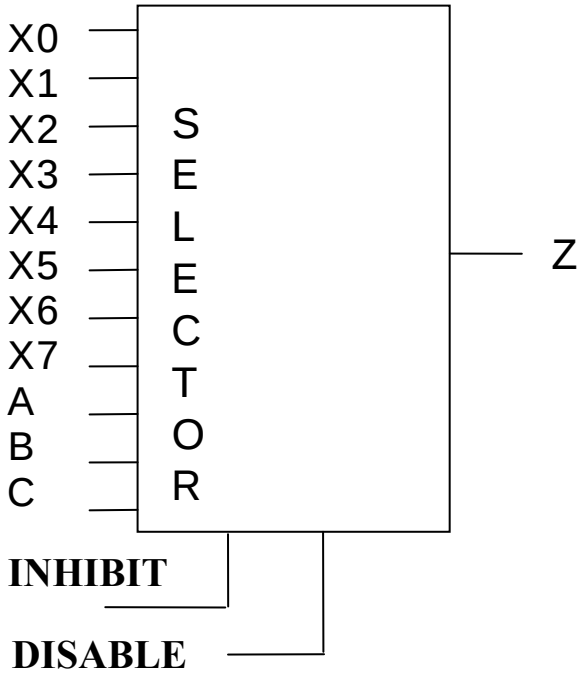
ENTRADAS			SAL.
I	A	B	Z
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3
1	0	0	D4
1	0	1	D5
1	1	0	D6
1	1	1	D7

I : ENTRDA DE HABILITACION

CI 4512 – 8 – CHANNEL DATA SELECTOR

PIN ASSIGNAMENT





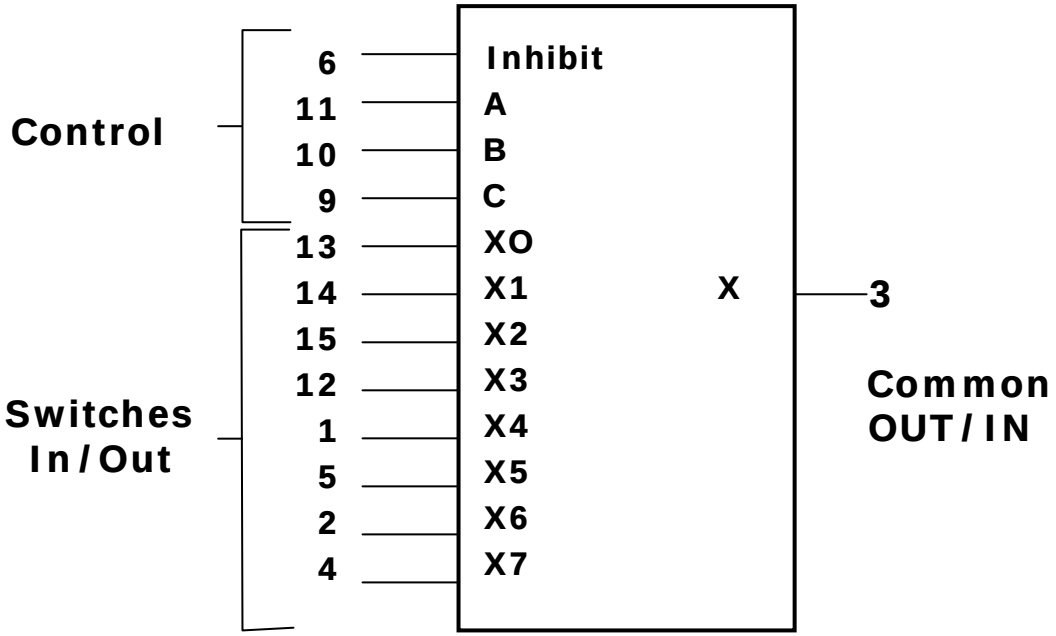
TRUTH TABLE

C	B	A	INHIBIT	DISABLE	Z
0	0	0	0	0	X0
0	0	1	0	0	X1
0	1	0	0	0	X2
0	1	1	0	0	X3
1	0	0	0	0	X4
1	0	1	0	0	X5
1	1	0	0	0	X6
1	1	1	0	0	X7
X	X	X	1	0	0
X	X	X	X	1	High Impedance

X = Don't Care

CI 4051 – 8-CHANEL ANALOG MULTIPLEXER / DEMULTIPLEXER

PINOUT

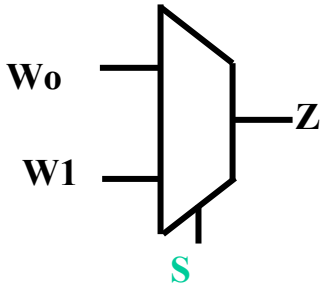


Vdd = Pin 16
Vss = Pin 8
Vee = Pin 7

CONTROL IMPUTS				
	SELECT			ON SWITCHES
INHIBIT	C	B	A	MC14051B
0	0	0	0	X0
0	0	0	1	X1
0	0	1	0	X2
0	0	1	1	X3
0	0	0	0	X4
0	0	0	1	X5
0	0	1	0	X6
0	0	1	1	X7
1	X	X	X	NONE

MODELO VHDL

Describimos el código de un multiplexor de 2 a 1. La selección de la señal se realiza mediante la entrada *S*




```
LIBRARY ieee;
USE ieee.std_logic_1164.all ;

ENTITY mux2to1 IS
        PORT ( w0, w1,S      : IN     STD_LOGIC;
                     f       : OUT  STD_LOGIC ) ;
END mux2to1 ;

ARCHITECTURE Behavior OF mux2to1 IS
BEGIN
        WITH S SELECT
            f <=   w0 WHEN '0' ,
                   w1 WHEN OTHERS;
END Behavior ;
```

Código VHDL para un multiplexor 2 a 1

Observamos dentro de la architectura la palabra clave WITH, la que especifica que *S* va ha ser usada para la selección. Hay dos clausulas WHEN, las que establecen que a la salida *f* le sea asignada el valor de w0 cuando *S*= 0, u otro si *S* = 1 (others: 1, z, -)

MODELO VHDL MUX 4:1

Mostramos el codigo para un multiplexor de 4 a 1.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all ;

ENTITY mux4to1 IS
        PORT ( w0, w1, w2, w3   : IN    STD_LOGIC ;
                    S           : IN    STD_LOGIC-_VECTOR(1 DOWNTO 0) ;
                    f           : OUT STD_LOGIC ) ;
END mux4to1 ;

ARCHITECTURE Behavior OF mux4to1 IS
BEGIN
        WITH S SELECT
            f <=    w0 WHEN “00”,
                    w1 WHEN “01”,
                    w2 WHEN “10”.
                    w3 WHEN OTHERS ;
END Behavior ;
```

MODELO VHDL

```
entity multi is port(                    entidad del multiplexor
a, b, c :in bit_vector(3 downto 0);      puertos del multiplexor
enable  :in bit;
control :in bit_vector(1 downto 0);
d   :out bit_vector(3 downto 0)
);
end multi;                               finaliza la entidad


architecture archmul of multi is         arquitectura del multiplexor
begin
process (a, b, c, control, enable)
begin
if enable='1' then d<="1111";            si enable es 1 entonces d="1111"
elsif enable='0' then                    si enable no es 1 y es 0 entonces
  case control is                        sentencia case dentro del if
    when "00" => d <= a;
    when "01" => d <= b;
    when "10" => d <= c;
    when "11" => d <= "1111";
  end case;                              se cierra la sentencia case
end if;                                  se cierra la sentencia if con end
end process;
end archmul;                             finaliza la arquitectura
```

MODELO VHDL - PROCCESS

```
LIBRARY ieee;
USE ieee.std_logic_1164.all ;

ENTITY mux2to1 IS
        PORT ( w0, w1, s    : IN    STD_LOGIC ;
                    f           : OUT    STD_LOGIC- ;
END mux2to1 ;

ARCHITECTURE Behavior OF mux2to1 IS
BEGIN
    PROCESS (w0, w1, s )
    BEGIN
        IF s = '0' THEN
            f <= w0 ;
        ELSE
            f <= w1 ;
        END IF ;
    END PROCESS ;
END Behavior ;
```

Figure 6.38 **A 2-to-1 multiplexer specified using the if-then-else statement**.