

DECODIFICADOR

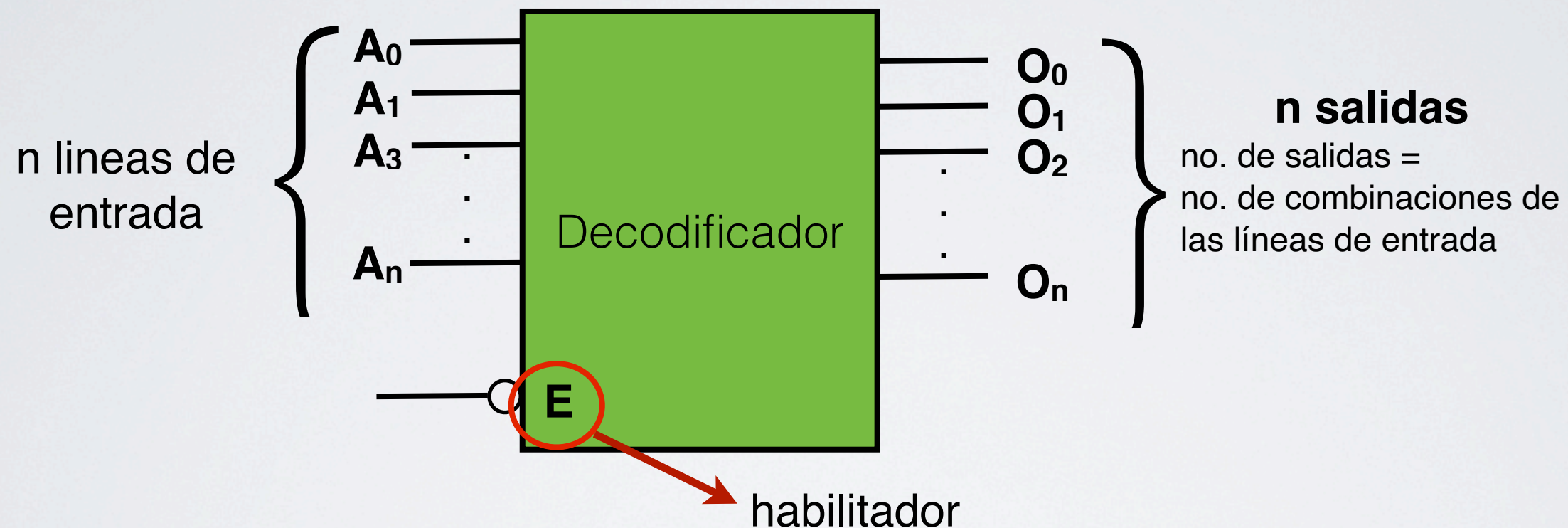
FUNDAMENTOS DE DISEÑO DIGITAL
OPTATIVA I. ISISA

Un ***decodificador*** es un circuito lógico que tiene n bits de entrada que representan un código binario y en la salida puede:

1. Activar sólo una salida, la cual corresponderá a la combinación del código de entrada, en este caso el número de salidas depende en casi todos los casos del número de combinaciones de las entradas.
 2. Activar una salida compuesta de n bits que conectada a algún dispositivo electrónico para presentar información, por ejemplo, muestra un número o letra específico.
-

TIPO 1

Activa sólo una salida, la cual corresponderá a la combinación del código de entrada, en este caso el número de salidas depende en casi todos los casos del número de combinaciones de las entradas.

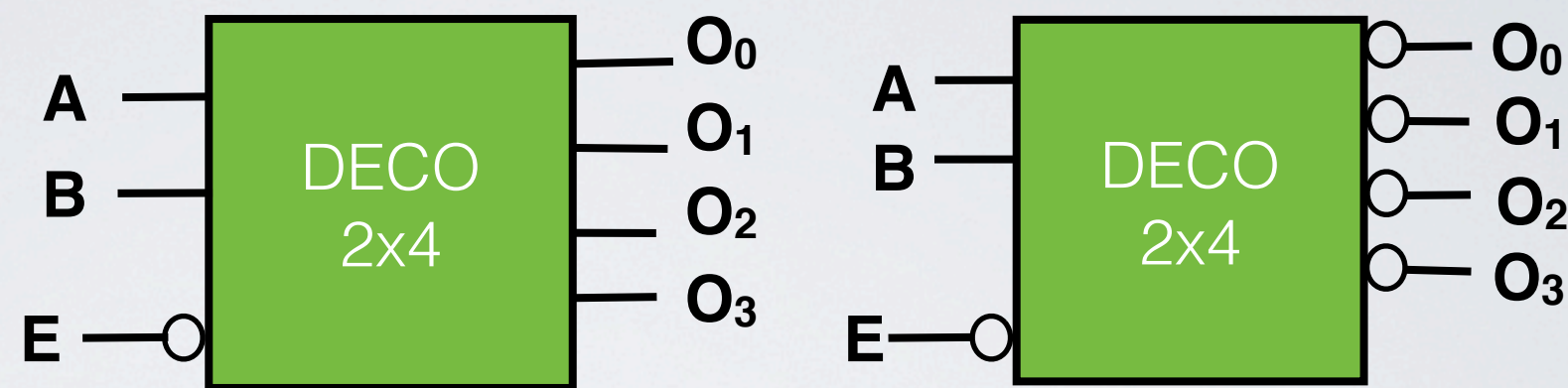


El habilitador o “enable” es una entrada de un bit que estando en valor 1 log (5VCC) las salidas se colocan en alta impedancia “Z”, esto es como si estuvieran apagado el decodificador. Cuando el habilitador vale 0 log (GND) las salidas del decodificador serán las correspondientes a la tabla de verdad.

DECODIFICADOR



Tabla de verdad de un decodificador 2x4 con salida activa en alto, en bajo y con habilitador.



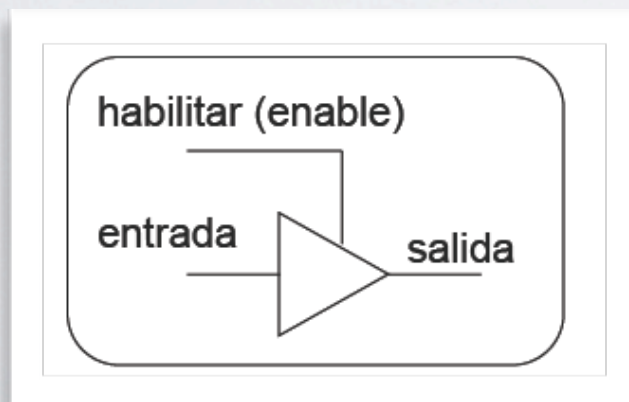
Enable	Entradas		Activación en alto				Activación en bajo			
E	A	B	O ₀	O ₁	O ₂	O ₃	O ₀	O ₁	O ₂	O ₃
0	0	0	1	0	0	0	0	1	1	1
0	0	1	0	1	0	0	1	0	1	1
0	1	0	0	0	1	0	1	1	0	1
0	1	1	0	0	0	1	1	1	1	0
1	X	X	Z	Z	Z	Z	Z	Z	Z	Z

Buffer de tercer estado

Para describir en VHDL la acción de inhabilitación o enable de un decodificador se utiliza el buffer de tercer estado.

Un buffer de tres estados es un circuito que funciona como un interruptor que controla el paso de una señal lógica de la entrada a la salida a través de una segunda entrada llamada de habilitación o enable. La salida puede tomar los siguientes valores Alto (1), Bajo (0) y de Alta Impedancia o Z.

Enable	Entrada	Salida
0	x	Z
1	0	0
1	1	1

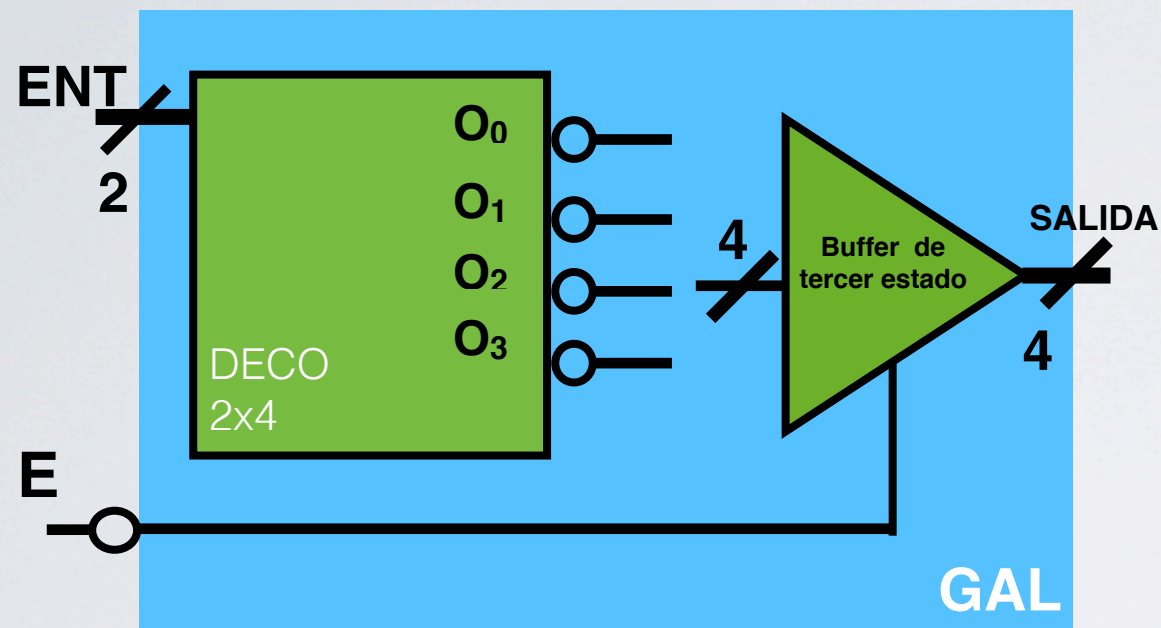


Tipos Lógicos Estándares	
'U'	Valor No-Inicializado
'X'	Valor Fuerte Desconocido
'0'	0 Fuerte
'1'	1 Fuerte
'Z'	Alta Impedancia
'W'	Valor Débil Desconocido
'L'	0 Débil
'H'	1 Débil
'-'	No Importa (Don't Care)

```
1  LIBRARY IEEE;
2  USE IEEE.STD_LOGIC_1164.ALL;
3
4  ENTITY ENE IS
5
6  PORT ( HAB, ENT: IN STD_LOGIC;
7         SAL:OUT STD_LOGIC
8         );
9
10 END ENE;
11
12 ARCHITECTURE A_ENE OF ENE IS
13 BEGIN
14     PROCESS (HAB, ENT)
15     BEGIN
16
17         IF (HAB='0') THEN
18             SAL<= 'Z';
19         ELSE
20             SAL<= ENT;
21         END IF;
22     END PROCESS;
23
24 END A_ENE;
```

DECODIFICADOR

Descripción en VHDL de un Deco 2x4 con salida en bajo y habilitador activado en bajo, a partir de su tabla de verdad.



E	ENT(1)	ENT(0)	SALIDA
0	0	0	1110
0	0	1	1101
0	1	0	1011
0	1	1	0111
1	0	0	Z
1	0	1	Z
1	1	0	Z
1	1	1	Z

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

ENTITY DECO IS
PORT
    ( ENT: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
      E: IN STD_LOGIC;
      SAL: OUT STD_LOGIC_VECTOR (3 DOWNTO 0);
    );
END ENTITY;

ARCHITECTURE A_DECO OF DECO IS
    SIGNAL O: STD_LOGIC_VECTOR (3 DOWNTO 0);
BEGIN

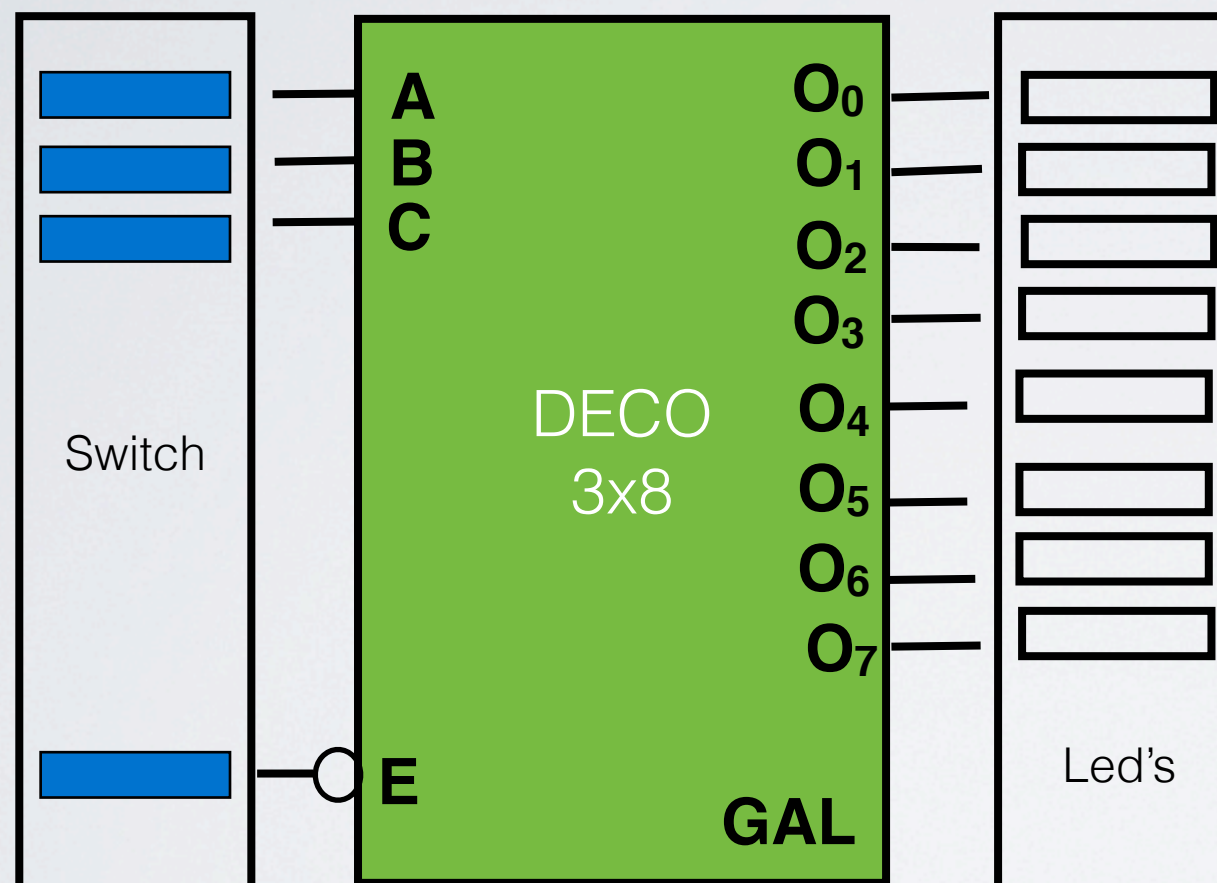
    WITH ENT SELECT
    O<= "1110" WHEN "00",
        "1101" WHEN "01",
        "1011" WHEN "10",
        "0111" WHEN OTHERS;

    PROCESS (E)
    BEGIN

        IF (E='0') THEN
            SALIDA <= O;
        ELSE
            SALIDA 'Z';
        END IF;
    END PROCESS;
END A-DECO;
```


PRACTICA 1

Describe en VHDL un Decodificador 3x8 con salida en alto y habilitador.

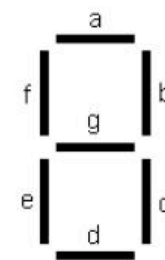
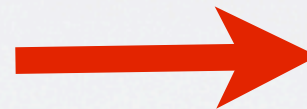
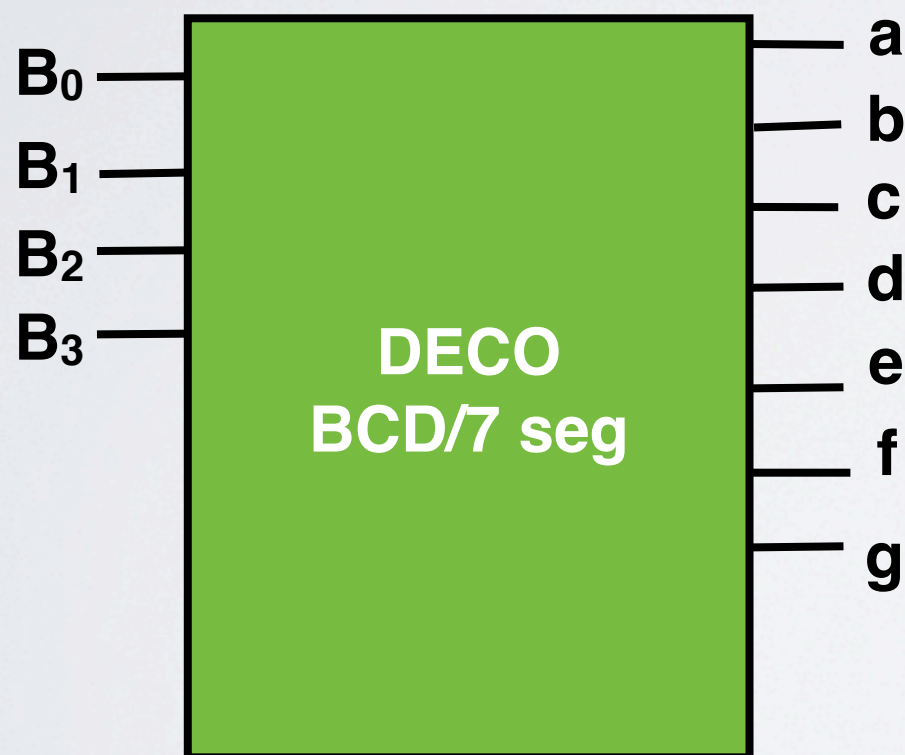


- Escribe la tabla de verdad.
- A partir de la Tabla has la descripción usando instrucciones secuenciales.

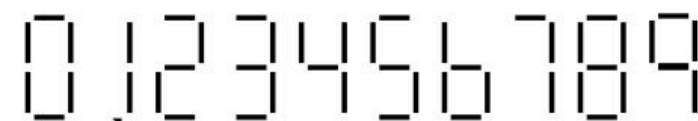
TIPO 2

Activar una salida compuesta de n bits que conectada a algún dispositivo electrónico para presentar o que visualiza información, por ejemplo, muestra un número o carácter específico.

DECODIFICADOR BCD/ 7 SEGMENTOS



(a)



(b)

(a) Configuración de siete segmentos; (b) segmentos activos para cada dígito

DECODIFICADOR

DISPLAY DE 7 SEGMENTOS

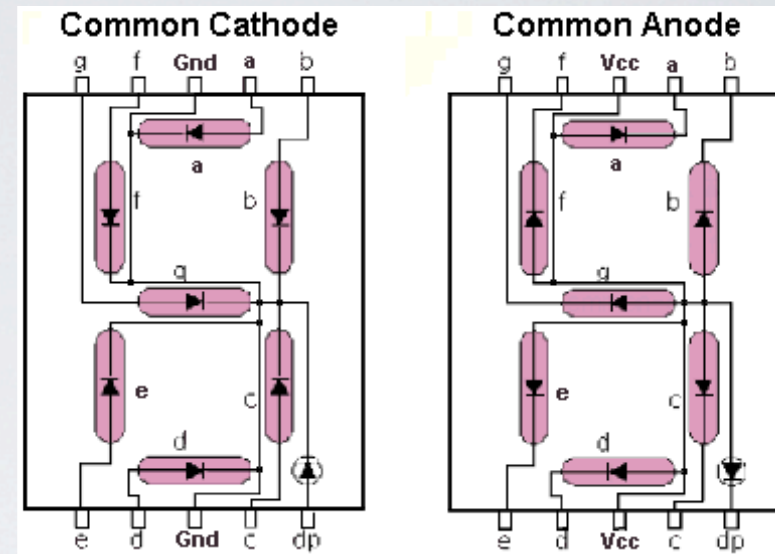
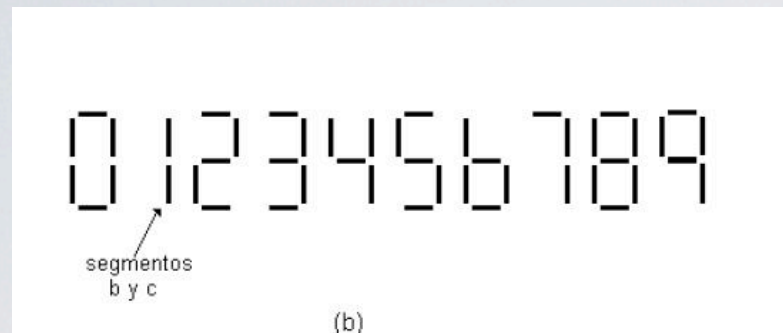
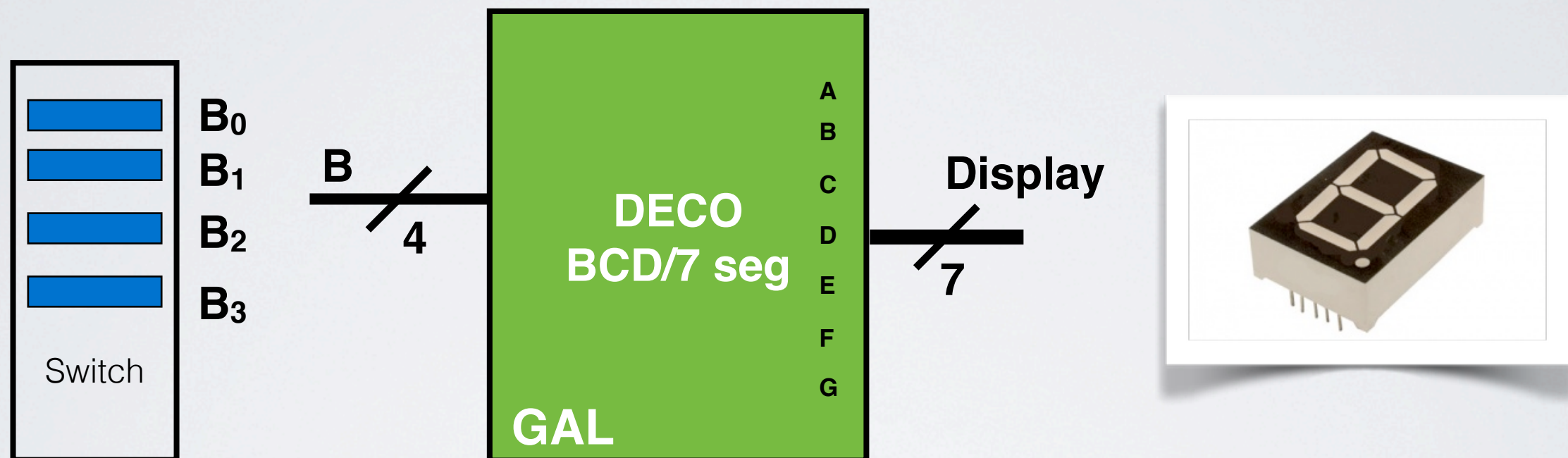


Tabla de verdad para decodificar BCD/ 7 segmentos

DIG	Ánodo común							Cátodo común						
	a	b	c	d	e	f	g	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1	1	1	1	1	1	1	0
1	1	0	0	1	1	1	1	0	1	1	0	0	0	0
2	0	0	1	0	0	1	0	1	1	0	1	1	0	1
3	0	0	0	0	1	1	0	1	1	1	1	0	0	1
4	1	0	0	1	1	0	0	0	1	1	0	0	1	1
5	0	1	0	0	1	0	0	1	0	1	1	0	1	1
6	1	1	0	0	0	0	0	0	0	1	1	1	1	1
7	0	0	0	1	1	1	1	1	1	1	0	0	0	0
8	0	0	0	0	0	0	0	1	1	1	1	1	1	1
9	0	0	0	1	1	0	0	1	1	1	0	0	1	1

PRACTICA 2

Describe en VHDL un decodificador BCD/7 segmentos considerando que se usará un display de cátodo común.



DECODIFICADOR



B ₃	B ₂	B ₁	B ₀	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

De la tabla de verdad anterior usamos la salida para cátodo común.

La salida para las combinaciones de entrada 1010 al 1111 no son validas, ya que sólo se puede representar un dígito en el display de siete segmentos, en estos casos a la salida se le puede asignar el valor de “**estado de no importa o don't care**” que se representan con “X”.

La forma canónica de la salida “a” con estados de no importa sería la siguiente:

$$a = \sum m(4,6) + DC(10,11,12,13,14,15)$$

El valor de “X” puede ser considerado como “1 o 0” log, ya que esta salida no es valida, el valor que se considere dependerá del diseño digital a realizar, en los mapas de Karnaugh se puede utilizar como comodín para obtener una mejor reducción de la función.

B ₁ B ₀		B ₃ B ₂			
		00	01	11	10
00					
01		1			1
11		X	X	X	X
10				X	X

Función mínima

$$a = B_2 \overline{B_0}$$

NOTA: cuando hay estados de no importan que no pueden agruparse con 1's o 0's estos no se deben usar para formar conjuntos.

Tabla de Verdad para un Deco BCD/7 seg de ánodo común


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

ENTITY DECOBCD IS
PORT
    ( B: IN STD_LOGIC_VECTOR (3 DOWNT0 0);
      DISPLAY: OUT STD_LOGIC_VECTOR (6 DOWNT0 0);
    );
END ENTITY;

ARCHITECTURE A_DECOBCD OF DECOBCD IS
BEGIN

PROCESS (B)
BEGIN

    CASE B IS

        WHEN "0000" => DISPLAY <= "0000001";
        WHEN "0001" => DISPLAY <= "1001111";
        WHEN "0010" => DISPLAY <= "0010010";
        WHEN "0011" => DISPLAY <= "0000110";
        WHEN "0100" => DISPLAY <= "1001100";
        WHEN "0101" => DISPLAY <= "0100100";
        WHEN "0110" => DISPLAY <= "1100000";
        WHEN "0111" => DISPLAY <= "0001111";
        WHEN "1000" => DISPLAY <= "0000000";
        WHEN "1001" => DISPLAY <= "0001100";
        WHEN OTHERS => DISPLAY <= "1111111"

    END CASE;

END PROCESS;
END A-DECOBCD;
```

Descripción en VHDL de un decodificador BCD/7 seg con display cátodo común. La descripción se hace utilizando la tabla de verdad anterior, los estados de no importa son sustituidos con la combinación "1111111", la cual apagará el display en las combinaciones 1010 al 1111.

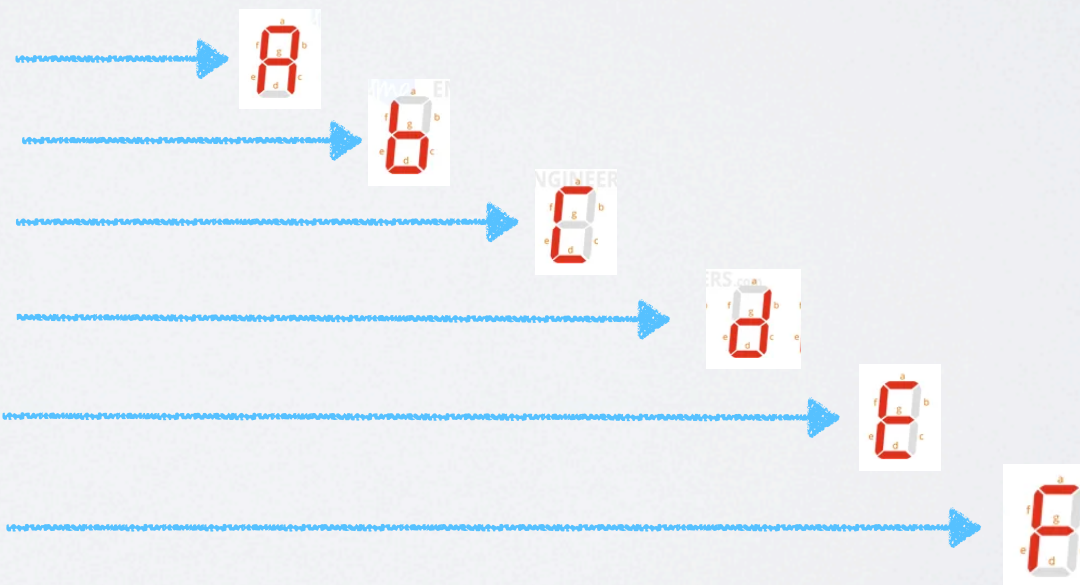
DECODIFICADOR



Binario				Hexadecimal						
B ₃	B ₂	B ₁	B ₀	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0
1	0	1	0	0	0	0	1	0	0	0
1	0	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	0	0	0	0
1	1	0	1	1	0	0	0	0	1	0
1	1	1	0	0	1	1	0	0	0	0
1	1	1	1	0	1	1	1	0	0	0

De manera genera un decodificador es un circuito digital que por cada combinación que se genere en la entrada activa una única salida, esta salida puede ser de un solo bit o un arreglo de bits que representen un carácter, por ejemplo, en un display.

Podemos representar el sistema numérico hexadecimal en un display de 7 segmentos. Del 0 al 9 es la misma codificación que para el código BCD, del 10 al 15 activamos los segmentos correspondientes para visualizar las letras A,b,C,d, E y F. En el caso de b y d se representan en minúscula ya que en mayúscula se confundirían con el 8 y 0, correspondientemente.

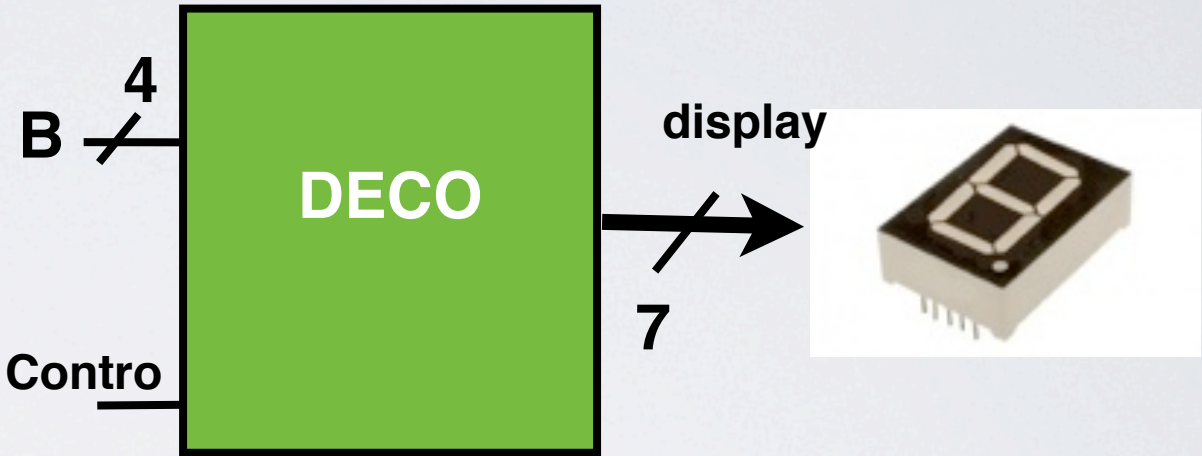


DECODIFICADOR



PRACTICA 3

Describe en VHDL un decodificador que dependiendo de control funcione como un Deco BDC/7seg o un Deco binario/hexadecimal.



control	DECO
0	BCD/ 7 SEG
1	BINARIO/HEXADECIMAL

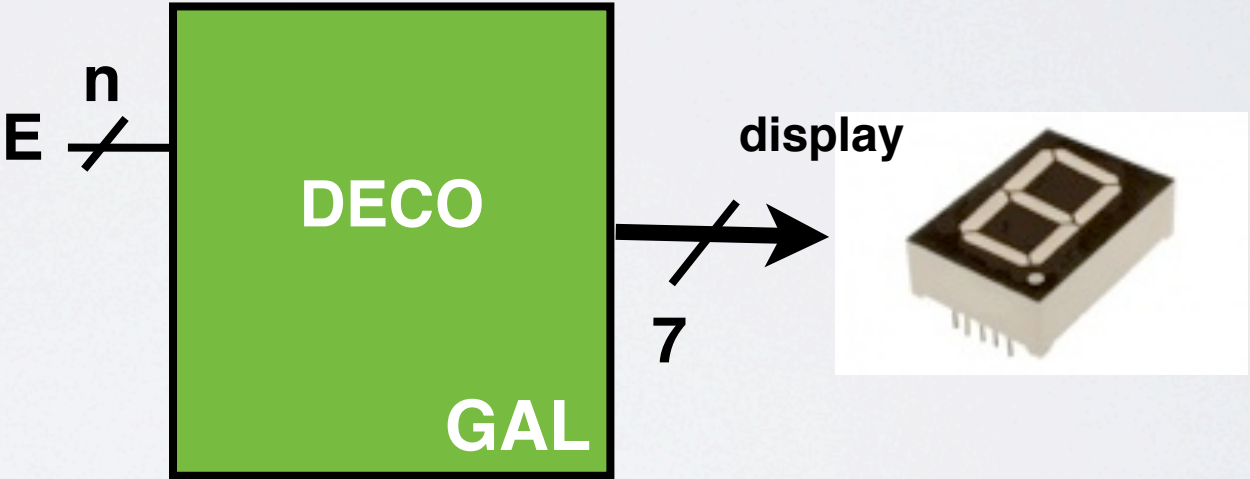
Nota: qué pondrías en el display para representar que del 10 al 15 no hay salida de un dígito en BCD.

B3	B2	B1	B0	display	
				C=0	C=1
0	0	0	0	BCD	
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	HEXADECIMAL	
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		

PRACTICA 4

Entrada	display
n	H
	O
	L
	A
B I T S	—
	P
	A
	L.
S	S
	E
	L
	C.

Describe en VHDL un decodificador que por cada combinación de entrada despliegue en el display letra por letra la palabra: HOLA y una palabra que tu selecciones y que pueda representarse en el display.



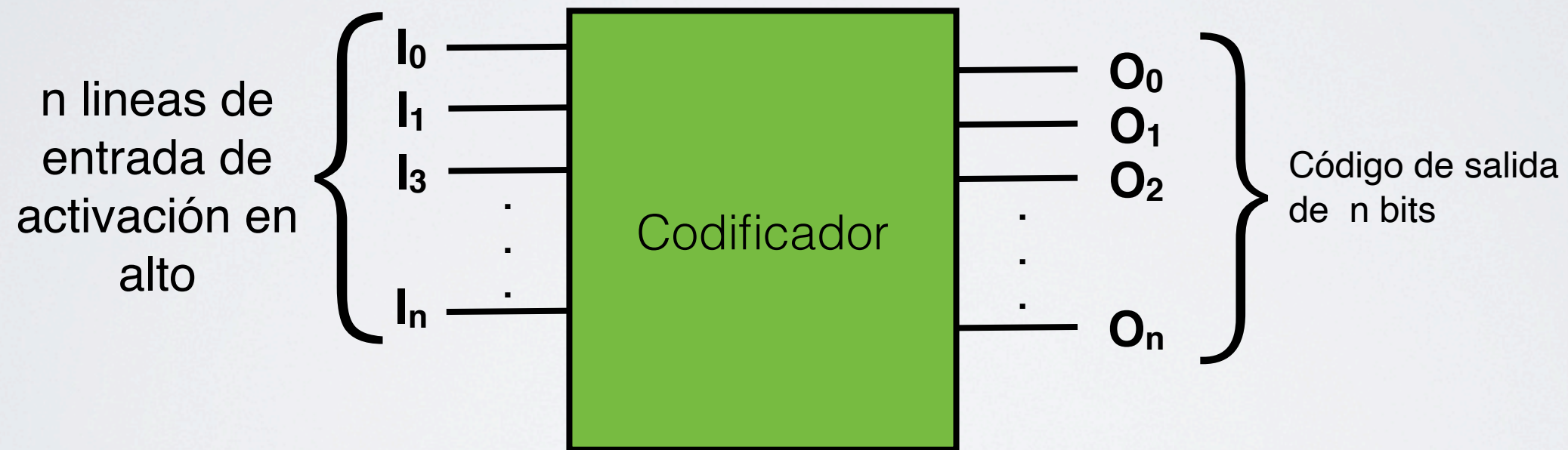
CODIFICADOR

FUNDAMENTOS DE DISEÑO DIGITAL
OPTATIVA I. ISISA

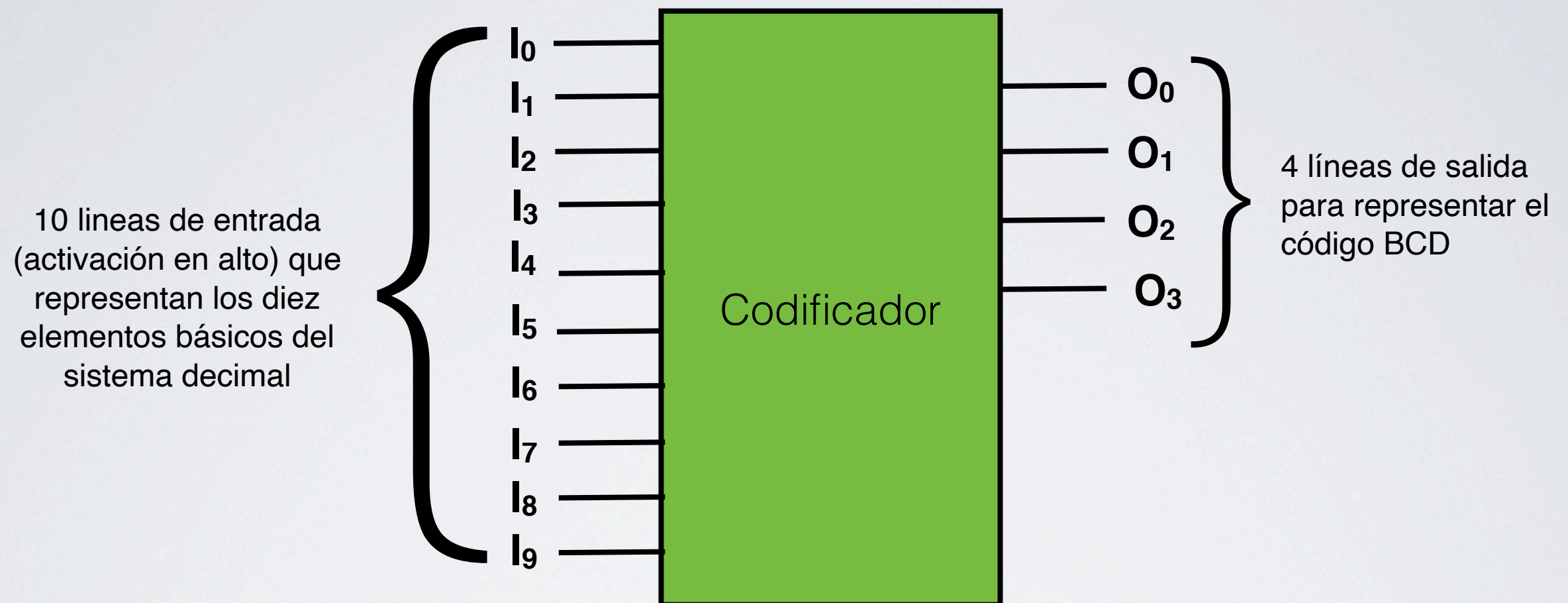
CODIFICADOR



Un ***codificador*** es un circuito lógico que tiene varias líneas de entrada, de las cuales, sólo una de ellas se activa en un momento dado y produce un código de salida de "n" bits que esta relacionado con la entrada que se activó.



CODIFICADOR DE PRIORIDAD DE DECIMAL A BCD



Prioridad: si dos o más entradas se activan al mismo tiempo la salida le dará prioridad a la entrada de mayor valor.

Tabla de verdad

I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	O3	O2	O1	O0
1	0	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	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

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

ENTITY COD IS
PORT
    ( I: IN STD_LOGIC_VECTOR (0 TO 9);
      O: OUT STD_LOGIC_VECTOR (3 DOWNT0 0);
    );
END ENTITY;

ARCHITECTURE A_COD OF COD IS
BEGIN

    PROCESS (I)
    BEGIN

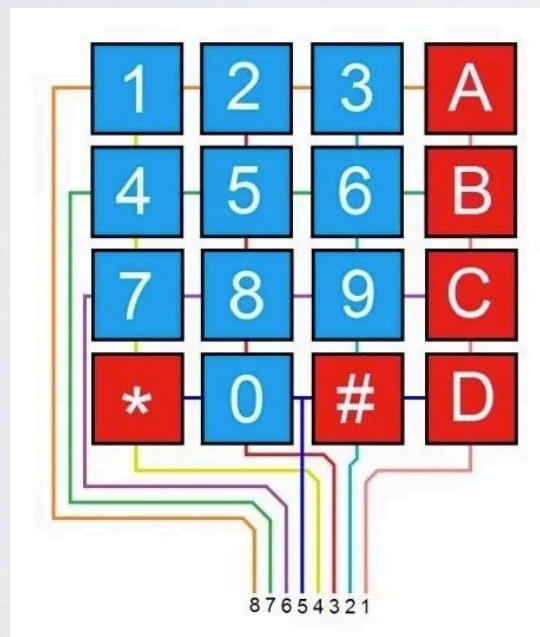
        CASE I IS
            WHEN "1000000000" => O<="0000";
            WHEN "0100000000" => O<="0001";
            WHEN "0010000000" => O<="0010";
            WHEN "0001000000" => O<="0011";
            WHEN "0000100000" => O<="0100";
            WHEN "0000010000" => O<="0101";
            WHEN "0000001000" => O<="0110";
            WHEN "0000000100" => O<="0111";
            WHEN "0000000010" => O<="1000";
            WHEN OTHERS=> O<="1001";

        END CASE;

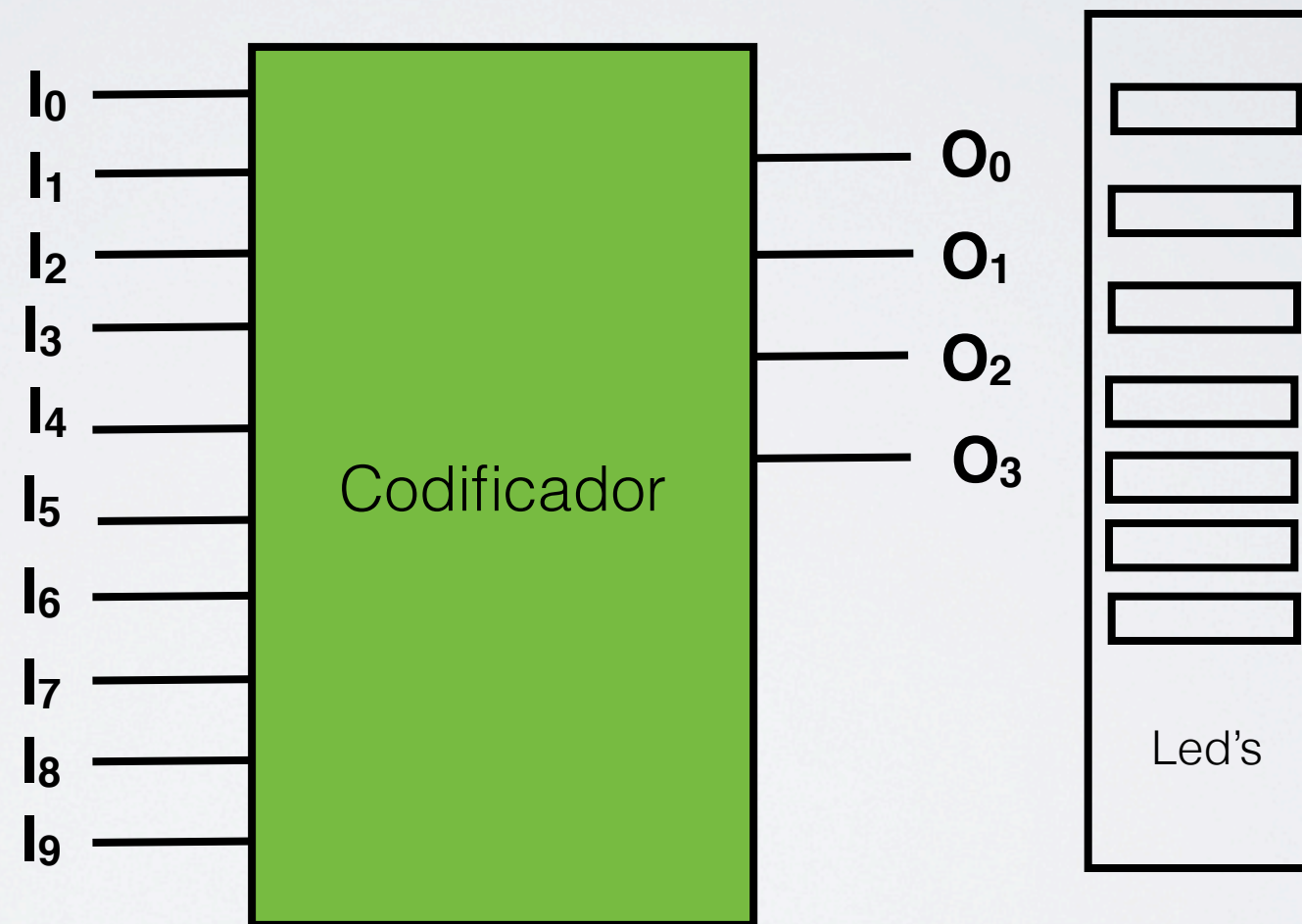
    END PROCESS;
END A_COD;
```

Descripción en VHDL de un codificador de década, de acuerdo a la tabla de verdad anterior. El código usa instrucciones secuenciales

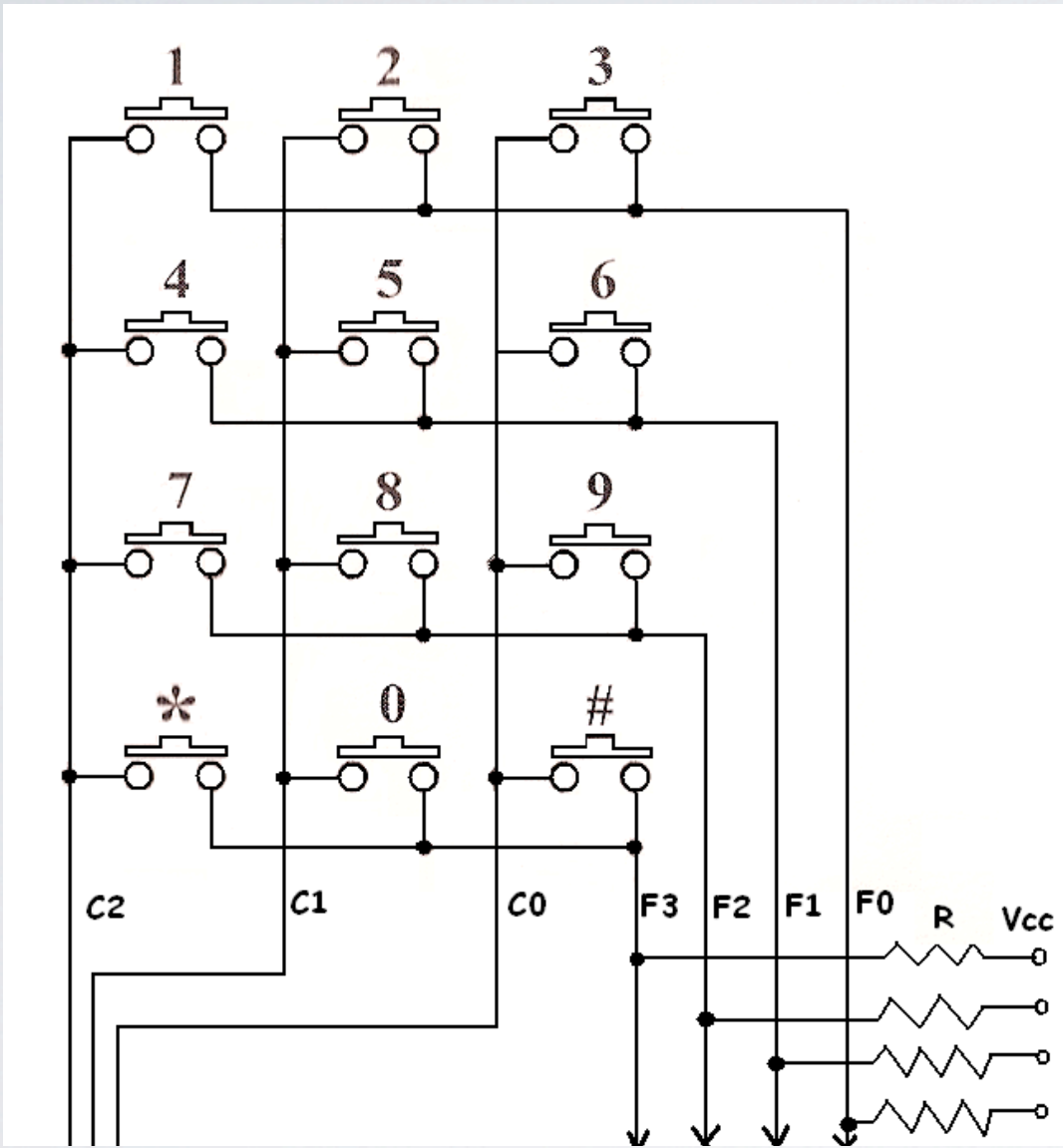
PRACTICA 1



Describe en VHDL un codificador de década de prioridad cuyas entradas se activen con un teclado matricial.



TECLADO MATRICIAL



F3	F2	F1	F0	C2	C1	C0	TECLA
1	1	1	1	X	X	X	NT
1	1	1	0	1	1	0	3
1	1	0	1	1	1	0	6
1	0	1	1	1	1	0	9
0	1	1	1	1	1	0	#
1	1	1	0	1	0	1	2
1	1	0	1	1	0	1	5
1	0	1	1	1	0	1	8
0	1	1	1	1	0	1	0
1	1	1	0	0	1	1	1
1	1	0	1	0	1	1	4
1	0	1	1	0	1	1	7
0	1	1	1	0	1	1	*