



**INSTITUTO POLITECNICO NACIONAL**



**UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERIA Y CIENCIAS  
SOCIALES Y ADMINISTRATIVAS**

**Ingeniería en Informática**

**Aplicaciones de Sistemas Digitales**

**Práctica V**

**Decodificadores**

### Practica V

#### Decodificadores

**Objetivo:** Implementar convertidores de códigos binarios a exhibidores de 7 y 16 segmentos, utilizando el método "Tabla de verdad" así como el uso de los operadores aritméticos "+" y "-".

#### Introducción

Un convertidor de código, puede estar compuesto de un codificador, de un decodificador o de ambos conectados entre sí, lo que realiza es pasar de un código numérico a otro.

En resumen, se puede decir que un conversor de código es un elemento lógico que traduce una palabra de "n" bits a otra de "m" bits las cuales se refieren al mismo valor decimal, pero en "distintos códigos".

A continuación se muestran algunos ejemplos de conversión de códigos.

#### *Código de binario a Octal*

A	B	C	Representación
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

#### *Código de binario a BCD*

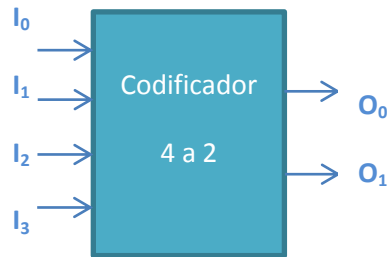
A	B	C	D	Representación
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

#### *Código de binario a Hexadecimal*

A	B	C	D	Representación
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

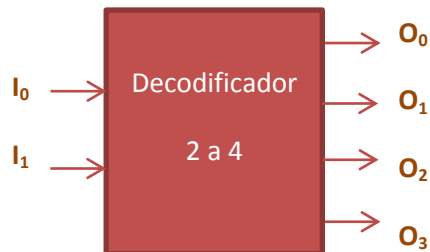
### Codificadores

Los codificadores permiten mostrar en sus "n" salidas, cuál de sus  $2^n$  entradas está activada. Ejemplo, si " $I_2 = 1$ " las salidas  $O_1O_0$  muestran el número 10.



### Decodificador

En el decodificador se interpreta la combinación de "n" entradas y se muestra en una de las  $2^n$  salidas. Ejemplo, si las entradas " $I_1I_0$ " tienen el valor 01, la salida  $O_1$  tendrá el estado lógico definido por el diseñador.

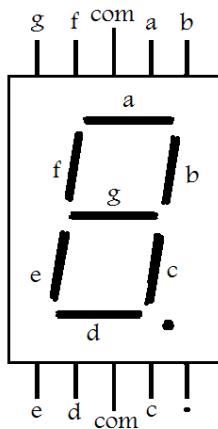


### Exhibidores (displays)

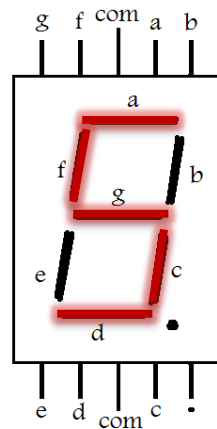
Los exhibidores son elementos para mostrar la información procesada por un sistema digital utilizando imágenes o símbolos alfabéticos y numéricos. Están formados por segmentos.

#### El exhibidor de 7 segmentos

Es un arreglo de 7 segmentos que permite representar los 10 dígitos decimales y las letras de la A a la F. Uno de los más usados es el compuesto por 7 Diodos Emisores de Luz (LED). Se tienen dos versiones, uno de ánodo común y otro de cátodo común. En éstos se controla el flujo de corriente para iluminar o apagar segmentos, por ejemplo si desea visualizar el número 5 debe iluminar los segmentos **a, f, g, c, d** y apagar los segmentos **b y e**.

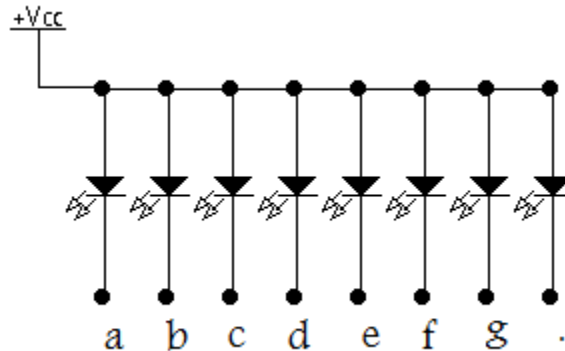


El exhibidor de 7 segmentos

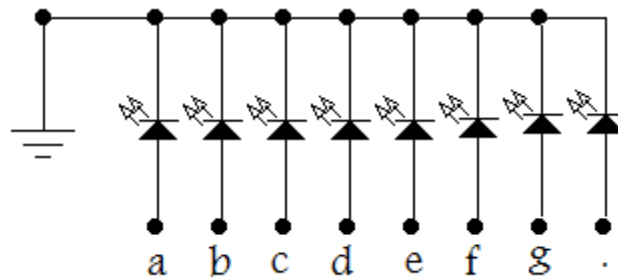


El exhibidor de 7 segmentos mostrando el número 5

Internamente los dos dispositivos están compuestos de 8 leds (incluyendo el punto decimal), los cuales internamente están conectados. Para el caso de Ánodo común, todos los lados positivos (ánodos) estarán conectados a un mismo punto, para que se ilumine cualquiera de los segmentos es necesario aplicar al lado contrario (cátodos), un cero (conexión a negativo) si recibe un 1 se toma como si estuviera conectado inversamente y no prende el segmento al que se le esté aplicando esta condición.

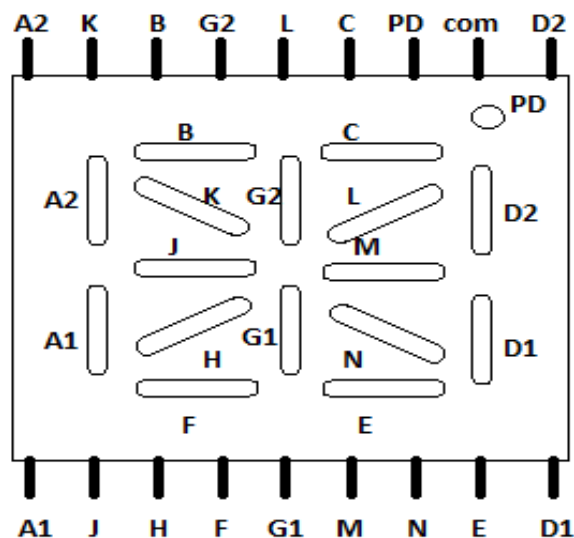


Para el caso de Cátodo común ahora todos los lados negativos (cátodos) están conectados a un mismo punto, que va a negativo, para que se ilumine cualquiera de los segmentos es necesario aplicar al lado contrario (ánodos), un uno (conexión a positivo) si recibe un 0 se toma como si estuviera conectado inversamente y no prende el segmento al que se le esté aplicando esta condición.



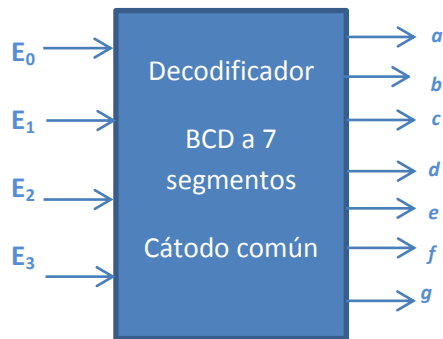
### Exhibidor de 16 segmentos

Funciona de la misma forma que el de 7 segmentos y permite mostrar otro tipo de caracteres alfanuméricos.



### Decodificador de BCD a 7 segmentos

Representar el BCD en un exhibidor de 7 segmentos de cátodo común.



E <sub>3</sub>	E <sub>2</sub>	E <sub>1</sub>	E <sub>0</sub>	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
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

Se obtienen las ecuaciones simplificadas y se realiza el código fuente

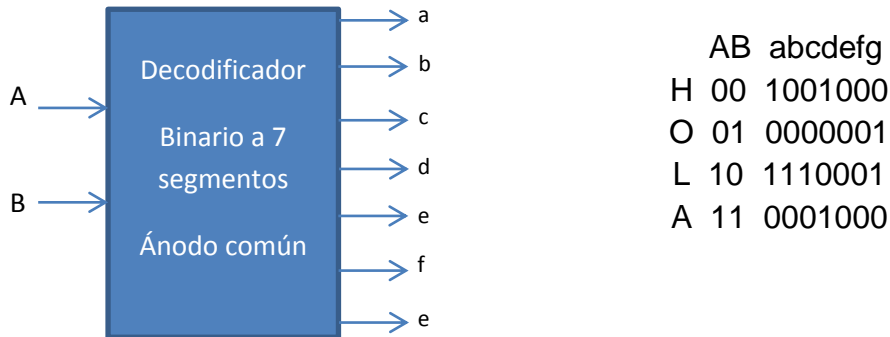
```

1  -----
2  --Decodificador de BCD a display de 7 segmentos de cátodo común
3  -----
4  library IEEE;
5  use IEEE.STD_LOGIC_1164.ALL;
6
7  entity bcd is
8      Port ( e3,e2,e1,e0 : in  STD_LOGIC;
9            a,b,c,d,e,f,g : out STD_LOGIC);
10 end bcd;
11
12 architecture ecuaciones of bcd is
13
14 begin
15
16     a <= e3 or e1 or (e2 xnor e0);
17     b <= not e2 or (e1 xnor e0);
18     c <= e2 or not e1 or e0;
19     d <= e3 or (not e2 and e1) or (not e2 and not e0) or (e1 and not e0) or (e2 and not e1 and e0);
20     e <= (not e2 and not e0) or (e1 and not e0);
21     f <= e3 or (e2 and not e0) or (e2 and not e1) or (not e1 and not e0);
22     g <= e3 or e2 or e1;
23
24 end ecuaciones;
25

```

### Decodificador de binario a 7 segmentos

En este caso utilizaremos el exhibidor de 7 segmentos de ánodo común para mostrar la palabra **HOLA**.



Nótese que por cada letra le corresponde una línea, no importa que la letra se repita recuérdese que es de un sistema a otro, por lo tanto, a cada combinación en binario le corresponde una combinación en 7 segmentos. Cabe señalar que se trata de 4 letras por eso se necesitan 2 variables para la tabla de verdad. Para el código fuente utilizaremos el condicional **case**

#### Case

Este condicional primero determina el valor de la señal a la que es sensible y después busca entre una lista de posibilidades la forma en que el sistema debe responder a determinado estímulo. Este condicional debe incluirse dentro de un **process**, es cual es una porción de código que solamente obedece (es sensible) a determinadas señales que se especifican dentro de un paréntesis y deben separarse por comas. Cabe señalar que las acciones descritas dentro de un process se realizan de forma secuencial, lo que significa que debe respetarse el orden de las mismas.

```

1  -----
2  --HOLA en 7seg AC
3  -----
4  library IEEE;
5  use IEEE.STD_LOGIC_1164.ALL;
6
7  entity hola is
8      Port ( y : in  STD_LOGIC_VECTOR (1 downto 0); --y1 y0 , puede utilizar variables independientes x, y
9            display : out  STD_LOGIC_VECTOR (6 downto 0)); --abcdefg
10 end hola;
11
12 architecture tabla of hola is
13     -- si utiliza variables independientes debe concatenarlas en un signal
14     -- signal entrada: STD_LOGIC_VECTOR (1 downto 0);
15
16 begin
17
18     -- entrada <= x & y;    concatenación de las variables para entrar al "case" como un solo elemento
19
20 process (y)                --si utiliza entradas concatenadas "process (entrada)"
21 begin
22     case (y) is             -- con entradas concatenadas "case (entrada) is"
23         when "00" =>
24             display <= "1001000";
25         when "01" =>
26             display <= "0000001";
27         when "10" =>
28             display <= "1110001";
29         when others =>
30             display <= "0001000";
31     end case;
32 end process;
33
34
35 end tabla;

```

**Sumador binario en paralelo de 3 bits, decodificado a display de 7 segmentos de cátodo común**

En este caso realizaremos la suma con el operador aritmético “+”. El resultado de la suma debe decodificarse en 2 displays de 7 segmentos (cátodo común) y además agregaremos un **led** que indicará si el resultado de la suma es un número **par**.

**Condicional if / elsif / else**

Este condicional permite definir la instrucción a realizar si una proposición es verdadera (IF) o si es falsa (ELSE). Cuando la lista de opciones es mayor a dos se utiliza “ELSIF”.

En este caso utilizaremos el IF/ELSE para encender o apagar el Led que indica si el número es par o no. Recuerde que los números binarios que son pares siempre tienen valor “0” en su LSB.

```

1  -----
2  -- Sumador paralelo 3 bits decodificado a 7seg ac
3  -----
4  library IEEE;
5  use IEEE.std_logic_1164.all;
6  use IEEE.numeric_std.all;
7  use IEEE.std_logic_unsigned.all;
8
9
10 entity sumaDecodificada is
11     Port ( a,b : in  STD_LOGIC_VECTOR (2 downto 0);
12           par : out STD_LOGIC;
13           deco : out  STD_LOGIC_VECTOR (8 downto 0)); --b2c2abcdefg
14 end sumaDecodificada;
15
16 architecture tabla of sumaDecodificada is
17
18     signal suma : STD_LOGIC_VECTOR (3 downto 0); --cs,s2,s1,s0
19
20 begin
21 --suma
22     suma <= ( '0' & a) + ( '0' & b);
23
24 --pares
25 process (suma)
26 begin
27     if suma(0) = '0' then
28         par <= '1';
29     else
30         par <= '0';
31     end if;
32 end process;
33
34 --decodificación
35 process (suma)
36 begin
37     case suma is
38         when "0000" => deco <= "001111110";--decenas apagado y cero en unidades
39         when "0001" => deco <= "000110000";
40         when "0010" => deco <= "001101101";
41         when "0011" => deco <= "001111001";
42         when "0100" => deco <= "000110011";
43         when "0101" => deco <= "001011011";
44         when "0110" => deco <= "001011111";
45         when "0111" => deco <= "001110000";
46         when "1000" => deco <= "001111111";
47         when "1001" => deco <= "001111011";
48         when "1010" => deco <= "111111110";--1 en las decenas y 0 en las unidades
49         when "1011" => deco <= "110110000";
50         when "1100" => deco <= "111101101";
51         when "1101" => deco <= "111111001";
52         when "1110" => deco <= "110110011";
53         when others => deco <= "000000000";--display de cátodo común apagado
54     end case;
55 end process;
56
57 end tabla;
58

```

**Materiales:**

- PC
- Software ISE Design Suite
- Software mojo loader..
- Cable transmisor de datos micro USB.
- Tarjeta Mojo Spartan 6.
- Jumpers.
- Protoboard.
- Display de 16 segmentos.
- DIP Switch.
- Alambre.
- Resistencias de 330 ohms.
- Fuente de alimentación de 5V a 1A.

**Desarrollo:**

1. Diseñar un decodificador de Hexadecimal a 7 segmentos ánodo común (puede utilizar el extremo derecho de su exhibidor de 16 segmentos) Utilice el método ecuaciones.
2. Diseñar un decodificador que muestre su nombre en un exhibidor de 16 segmentos.
3. Diseñar un restador binario de 3 bits, considere que el resultado debe ser decodificado.



**Cuestionario:**

1.- Realizar el decodificador de binario a octal representado a siete segmentos. Obtener las ecuaciones para cada segmento.