

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

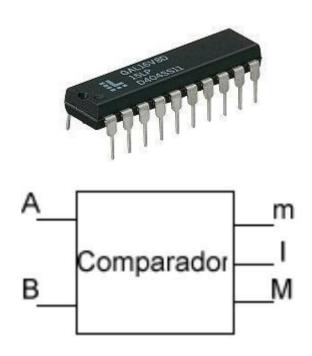


DEPARTAMENTO DE INGENIERÍA EN SISTEMAS

COMPUTACIONALES

PRACTICA No. 9

Comparador de magnitud de 4 bits



Comparador de magnitud

1) Objetivo general.

Al terminar la sesión, los integrantes del equipo diseñaran e implementaran un circuito comparador de magnitud complementado con un decodificador.

2) Material empleado.

- ✓ 1 Circuito Integrado GAL22V10.
- ✓ 7 Resistores de 220Ω .
- ✓ 8 Resistores de 1K Ω .
- ✓ 1 Dip switch de 10.
- ✓ Alambre telefónico.
- ✓ 1 Display de 7 segmentos de ánodo común.
- ✓ 1 Tablilla de Prueba (Protoboard).
- ✓ Pinzas de punta.
- ✓ Pinzas de corte.
- ✓ Cables Banana-Caimán (para alimentar el circuito).

3) Equipo empleado.

- ✓ Multímetro.
- ✓ Fuente de Alimentación de 5 Volts.
- ✓ Programador Universal.

4) Desarrollo Experimental y Actividades.

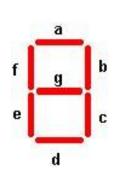
4.1.- Comparador de magnitud de 4 bits.

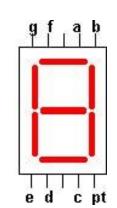
Diseñe un comparador de magnitud de dos cantidades de 4 bits, cuya salida va a ser la entrada a un decodificador, para al final conectarse a un display de 7 segmentos que desplegara el dato, como se muestra en la figura siguiente:

DESARROLLO

Para el desarrollo de esta practica se investigo a fondo el funcionamiento de un comparador y se basa básicamente en comparar la entrada A y la entrada B resgresando cual es mayor o menor y en caso de ser iguales se regresaría un igual.

Si bien hay varias formas de implementar este comparador se decidió implementar el mas fácil que es comparar directamente las entradas A y B y la salida que arrojaran estos se irían para el display





El display encenderá los campos a y d en caso de que el resultado del comparador nos de que estos números sean iguales

El display encenderá los campos a,b,c,d en caso de que el valor A sea mayor a la entrada B

El display encenderá los campos a,d,e,f en caso de que a sea menor que B

TABLA DE VERDAD

					(muy mu	iy corta)					
	A3	A2	A 1	A0	В3	B2	B1	В0	A > B	A <b< td=""><td>A =B</td></b<>	A =B
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1	0	0	1	0
2	0	0	1	1	0	0	1	0	1	0	0
3	0	1	0	0	0	0	1	1	1	0	0
4	0	1	0	0	0	1	0	1	0	1	0
5	0	1	0	1	0	1	0	1	0	0	1
6	0	1	1	1	0	1	1	0	1	0	0
7	0	1	1	1	1	0	0	0	0	1	0
8	1	0	0	0	1	0	0	0	0	0	1
9	1	1	1	1	1	1	1	1	0	0	1

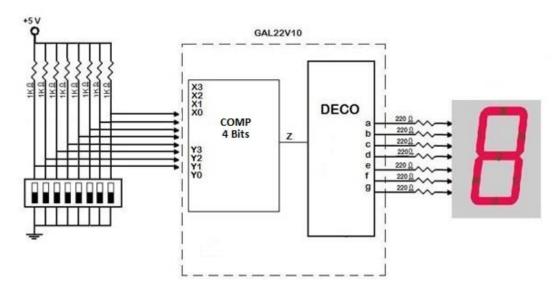


Figura. Desarrollo del comparador de magnitud de 4 bits.

4.2.- Implemente su solución en VHDL y coloque su informe de pines RPT y su código.

```
1 library ieee;
 2 use ieee.std logic 1164.all;
 4 entity comparador is
 5
      port( a,b : in std logic vector(3 downto 0);
       display : out std logic vector (6 downto 0)
 6
 7
 8 end comparador;
10 architecture comparadoor of comparador is
11 begin
12
13 process(a, b)
14
15
      begin
       if(a > b) then
16
17
           display <= "0000111";
18
       elsif(a < b)then
19
           display <= "0110001";
20
21
           display <= "0110111";
22
       end if:
```

Comparador de magnitud Value Stimulator Name ¹⁰ 100 ns ± - a o <= 0000 Ø ⊕ ⊳- Ь <= 0000 Œ 0 🖃 🔤 display (ZZ)(25)(37 $\chi_{17} \chi_{07}$ display(6) 0 display(5) 1 display(4) 1 display(3) 0 display(2) 1 display(1) 1 display(0) . . 50 . . . 100 . . . 150 . . . 20<mark>200 ns</mark> Name Value Stimulator (O + **⊳** a 1 <= 0001 <= 0000 + ⊳ Ь 0 🖃 🗢 display (ZZ)(25)(37 X23 X3 07 display(6) display(5) 0 display(4) 0 display(3) 0 display(2) 1 display(1) 1 display(0) 1 Value 1 · 200 · 1 · 250 · 1 · 300 · 1 · 350 · 400 ns 400 Stimulator Name ± ⊳ a <= 0001 + ⊳- Ь 1 <= 0001 X27 X37 🖃 🗢 display 37 display(6) 0 display(5) display(4) 1 display(3) 0 display(2) 1 display(1) display(0) 1

Comparador de magnitud 250 · · · 300 · · · 350 · · · 400 · · · 450 · 500 ns 500 Value ∑ame Stimulator ⊕ r a <= 0001 χ4 ⊕ ⊳ Ь 4 <= 0100 X33 X31 🖃 🗢 display 7 31 display(6) 0 display(5) 1 display(4) 1 display(3) display(2) 0 display(1) 0 display(0) 1 350 . , . 400 . , . 450 . , . 500 . , . 550 . 600 ns Name Value Stimulator ⊕ ¤ a 5 <= 0101 χ4 ⊕ ⊳ Ь 4 <= 0100 X33 X31 (15)(07 🖃 🗢 display 07 display(6) 0 display(5) 0 display(4) 0 display(3) 0 display(2) display(1) 1 display(0) 1

Name	Value	Stimulator	450 · · · 500 · · · 550 · · · 600 · · · 650 · 700 ns 7
+ ⊳a	5	<= 0101	
+ ⊳ Ь	5	<= 0101	X5
🗏 🗢 display	37		1 (15)(07)(27)(37
🗢 display(6)	0		
🗢 display(5)	1		
🗢 display(4)	1		
🗢 display(3)	0		
🗢 display(2)	1		
display(1)	1		
display(0)	1		

Comparador de magnitud 700 · · · 750 · 800 ns 800 Value 600 . . . 650 . Name Stimulator χ8 ± ⊳ a 8 <= 1000 χ5 ⊕ ⊳ Ь 7 <= 0111 X27 X37 X17 X07 🖃 🗢 display 07 display(6) 0 display(5) 0 display(4) 0 display(3) 0 display(2) 1 display(1) 1 display(0) 1 Stimulator Name Value 700 · · · 750 · · · 800 · · · 850 · 900 ns 900 χ8 ⊕ o a <= 1000 **∓** ⊳ Ь <= 1000 X17 X07 X27 X37 🖃 🗢 display 37 display(6) 0 display(5) display(4) 1 display(3) 0 display(2) 1 display(1) display(0) 800 · · · 850 · · · 900 · · · 950 1000 ns 1000 Value Stimulator Name ± ⊳ a 8 <= 1000 **∓** ⊳ Ь <= 1110 Ε X27 X37 X33 X31 17 🖃 🗢 display 31 display(6) 0 display(5) display(4) 1 display(3) 0 display(2) 0 display(1) 0 display(0) 1

Comparador de magnitud

Name	Value	Stimulator	850 900 950 .	1 - 1000 - 1 - 1050 1100 ns 1100
∓ • a	F	<= 1111		XF
⊕ ⊳ Ь	F	<= 1111	XE	XF.
🗏 🗢 display	37		7 (33)(31	(35)(37
🗢 display(6)	0			
🗢 display(5)	1			
display(4)	1			
🗢 display(3)	0			
🗢 display(2)	1			
🗢 display(1)	1			
display(0)	1			

Comparador de magnitud

5) Conclusiones Individuales.

Los comparadores siempre son utilizados en programación y ahora tenemos la oportunidad de codificar uno... y si estuviéramos en presencial tendríamos la oportunidad de armarlo lo cual es genial ya que no solo usamos codigo para poder obtener ese fucionamiento si no también podemos entender como es su forma de operar y ver tal cual el circuito en físico esto nos permitiría una percepción super clara acerca de estos dispositivos.

El comparador es sumamente sencillo de implementar en vhd con solo unas cuantas líneas de codigo pero su función es sumamente útil nos sirve para comparar señales o voltajes con unas cuantas modificaciones.

6) Bibliografía.