



# Instituto Politécnico Nacional

## Escuela Superior de Cómputo

### Práctica 3

### Compuertas lógicas usando VHDL

Materia: Fundamentos de Diseño Digital

Alumnos: Ávila López Víctor Iván  
Ruiz Gonzales Carlos Emilio  
Cazares Cruz Jeremy Sajid

Grupo: 3CM5

# Objetivo

El alumno realizará las compuertas lógicas AND, OR, NAND, NOR, XOR y XNOR programando en el lenguaje VHDL y programará su GAL 22V10 para verificar el resultado.

## Introducción Teórica

Las Compuertas Lógicas son circuitos electrónicos conformados internamente por transistores que se encuentran con arreglos especiales con los que otorgan señales de voltaje como resultado o una salida de forma booleana, están obtenidos por operaciones lógicas binarias (suma, multiplicación). También niegan, afirman, incluyen o excluyen según sus propiedades lógicas. Estas compuertas se pueden aplicar en otras áreas de la ciencia como mecánica, hidráulica o neumática.

Existen diferentes tipos de compuertas y algunas de estas son más complejas, con la posibilidad de ser simuladas por compuertas más sencillas. Todas estas tienen tablas de verdad que explican los comportamientos en los resultados que otorga, dependiendo del valor booleano que tenga en cada una de sus entradas.

### Materiales empleados

1 circuito Integrado GAL22V10

5 LEDS de colores diferentes

5 resistores de  $330\Omega$

5 resistores de  $1K\Omega$

1 Dip switch de 8

Alambre telefónico

1 tablilla de Prueba (Protoboard)

1 pinzas de punta

1 pinzas de corte

Cables Banana-Caimán (para alimentar el circuito) Materiales

### Equipo empleado

Multímetro

Fuente de Alimentación de 5 Volts

Manual de MOTOROLA, "FAST and LS TTL"

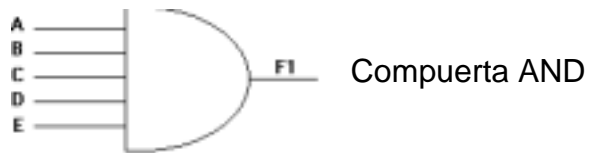
Programador Universal

Desarrollo Experimental y Actividades

1.- Implementar las siguientes compuertas lógicas usando VHDL.

2.- Llene su tabla de verdad para todas las compuertas.

3.- Coloque el código final de su programa junto con el archivo de asignación de pines RPT. 4.- Arme su circuito y programe la GAL para verificar sus tablas de verdad.



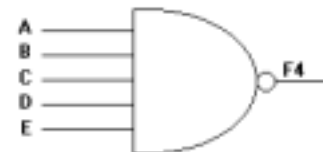
Compuerta AND



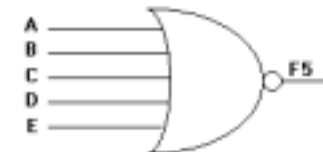
Compuerta OR



Compuerta XOR



Compuerta NAND



Compuerta NOR



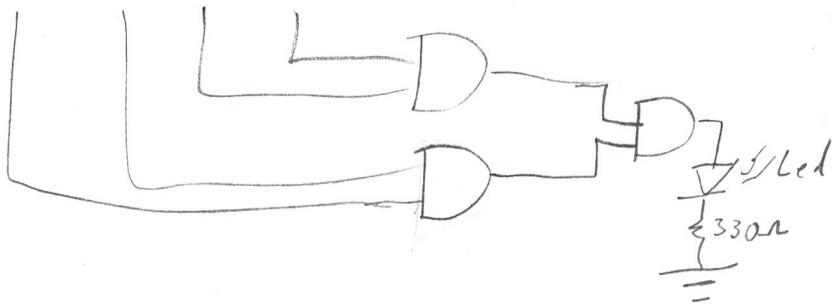
Compuerta XNOR

# Tabla de Verdad

#	A	B	C	D	E	F1 AND	F2 OR	F3 XOR	F4 NAND	F5 NOR	F6 NORE
0	0	0	0	0	0	0	0	0	1	1	1
1	0	0	0	0	1	0	1	1	1	0	0
2	0	0	0	1	0	0	1	1	1	0	0
3	0	0	0	1	1	0	1	0	1	0	1
4	0	0	1	0	0	0	1	1	1	0	0
5	0	0	1	0	1	0	1	0	1	0	1
6	0	0	1	1	0	0	1	0	1	0	0
7	0	0	1	1	1	0	1	1	1	0	0
8	0	1	0	0	0	0	1	1	1	0	0
9	0	1	0	0	1	0	1	0	1	0	1
10	0	1	0	1	0	0	1	0	1	0	1
11	0	1	0	1	1	0	1	1	1	0	0
12	0	1	1	0	0	0	1	0	1	0	1
13	0	1	1	0	1	0	1	1	1	0	0
14	0	1	1	1	0	0	1	1	1	0	0
15	0	1	1	1	1	1	1	0	0	0	1

## Compuerta AND

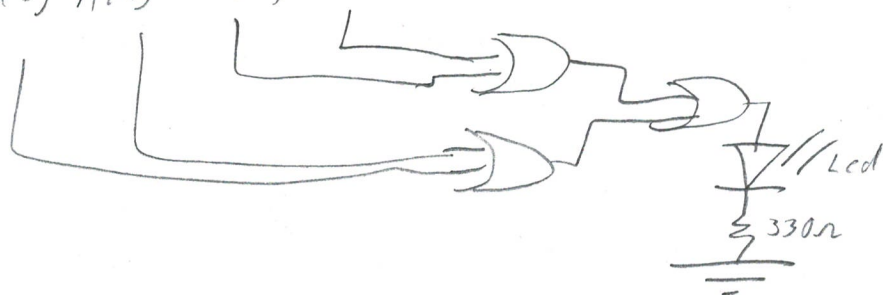
A(3) A(2) A(1) A(0)



$$\text{Andout} \Leftarrow (A(0) \text{ and } A(1)) \text{ and } (A(2) \text{ and } A(3))$$

## Compuerta Or

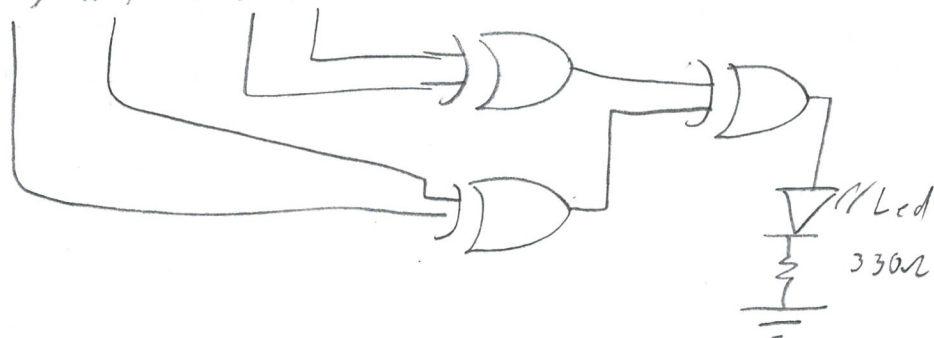
A(3) A(2) A(1) A(0)



$$\text{Orout} \Leftarrow (A(0) \text{ or } A(1)) \text{ or } (A(2) \text{ or } A(3))$$

## Compuerta XOR

A(3) A(2) A(1) A(0)

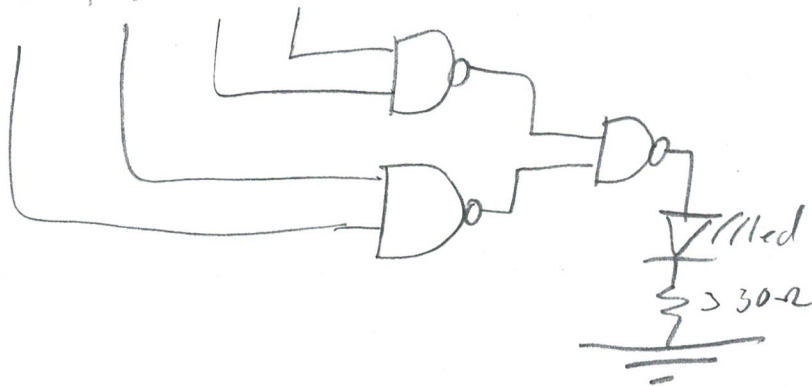


$$\text{Xorout} \Leftarrow (A(0) \text{ xor } A(1)) \text{ xor } (A(2) \text{ xor } A(3))$$



Compuerta NAND

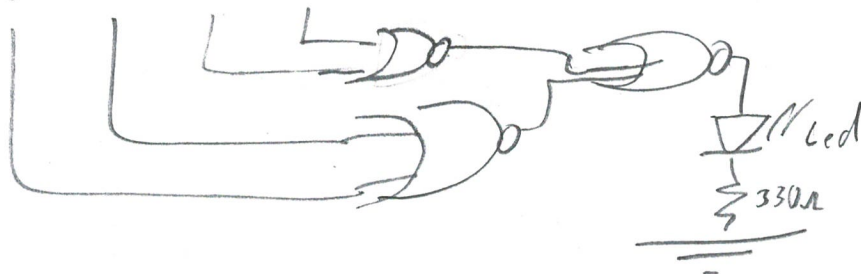
A(3) A(2) A(1) A(0)



$$NAND_{out} = \text{Not} ((A(0) \text{ and } A(1)) \text{ and } (A(2) \text{ and } A(3)))$$

Compuerta Nor

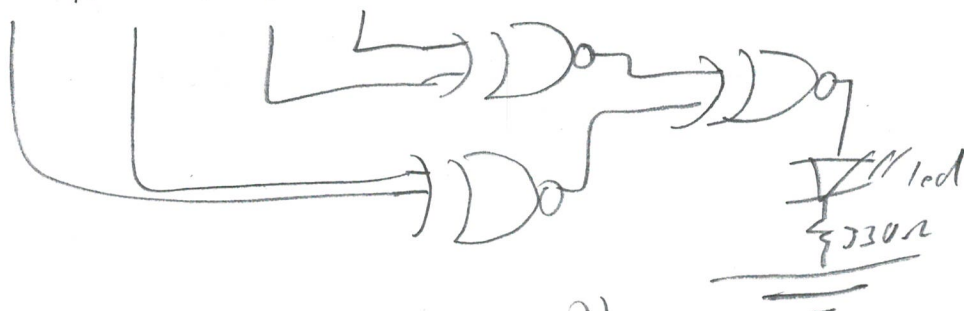
A(3) A(2) A(1) A(0)



$$NOR_{out} = \text{Not} ((A(0) \text{ or } A(1)) \text{ or } (A(2) \text{ or } A(3)))$$

Compuerta XNOR

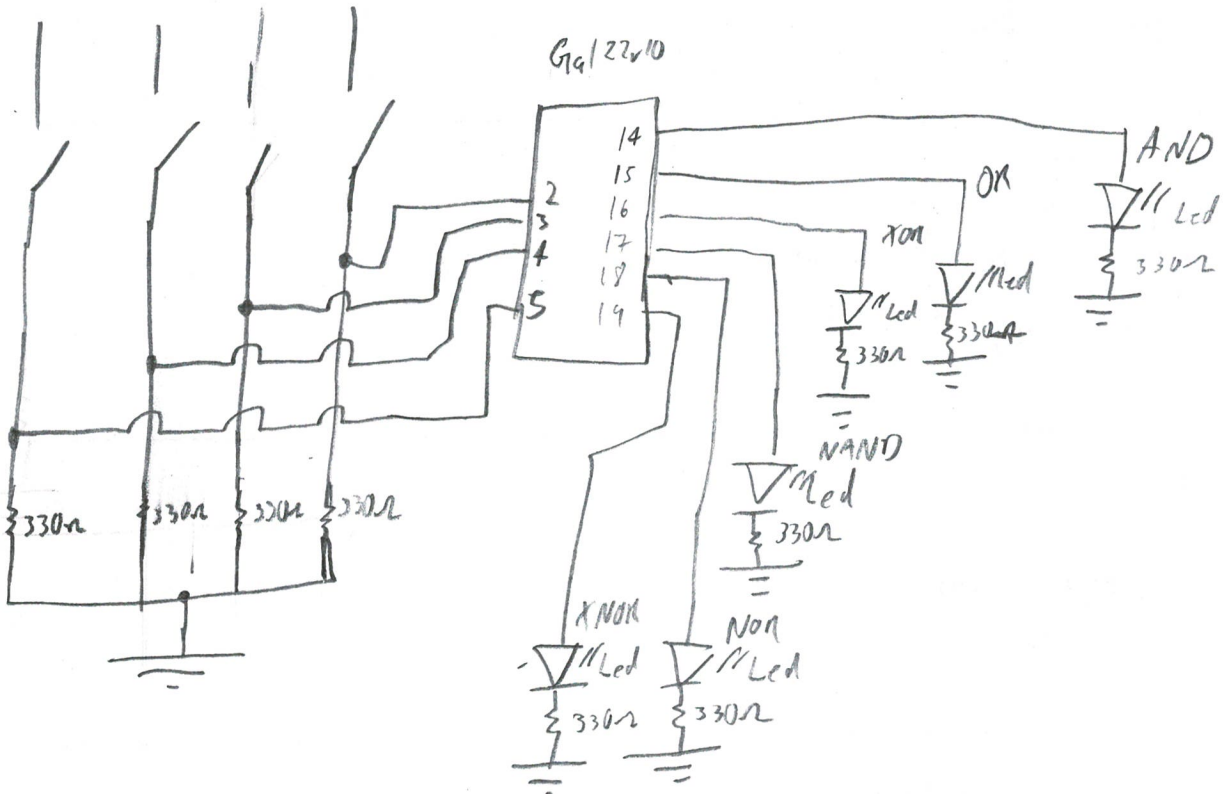
A(3) A(2) A(1) A(0)



$$XNOR_{out} = (A(0) \text{ xnor } A(1)) \text{ xnor } (A(2) \text{ xnor } A(3))$$

# Diagrama del circuito físico

A(3) A(2) A(1) A(0)



## Código

```

1  LIBRARY IEEE;
2  USE IEEE.STD_LOGIC_1164.ALL;
3
4  ENTITY COMPUERTAS IS
5      PORT (A : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
6            Y : OUT STD_LOGIC_VECTOR(5 DOWNTO 0));
7
8  ATTRIBUTE PIN_NUMBERS OF COMPUERTAS: ENTITY IS
9      " A(0):2 A(1):3 A(2):4 A(3):5 "
10     & " Y(0):14 Y(1):15 Y(2):16 Y(3):17 "
11     & " Y(4):18 Y(5):19 ";
12
13 END COMPUERTAS;
14
15 ARCHITECTURE CODIGO OF COMPUERTAS IS
16 BEGIN
17
18     Y(0) <= (a(0) and a(1)) and (a(2) and a(3));
19     Y(1) <= (a(0) or a(1)) or (a(2) or a(3));
20     Y(2) <= not((a(0) and a(1)) and (a(2) and a(3)));
21     Y(3) <= not((a(0) or a(1)) or (a(2) or a(3)));
22     Y(4) <= (a(0) xor a(1)) xor (a(2) xor a(3));
23     Y(5) <= (a(0) xnor a(1)) xnor (a(2) xnor a(3));
24
25 END CODIGO;

```

## Pines asignados

205					
206				C22V10	
207					
208	not	used	*	1	24  * not used
209		a(0)	=	2	23  * not used
210		a(1)	=	3	22  * not used
211		a(2)	=	4	21  * not used
212		a(3)	=	5	20  * not used
213	not	used	*	6	19 = y(5)
214	not	used	*	7	18 = y(4)
215	not	used	*	8	17 = y(3)
216	not	used	*	9	16 = y(2)
217	not	used	*	10	15 = y(1)
218	not	used	*	11	14 = y(0)
219	not	used	*	12	13  * not used
220					
221					
222					



## **Conclusiones Individuales.**

Emilio

En esta práctica entendimos el comportamiento de las compuertas NOR, AND, XOR, OR, XNAN y XNOR, así como el funcionamiento del código de VHDL, al momento de programar una GAL, si bien la práctica era un tanto sencilla, se tuvieron problemas al momento de programar la GAL, pero al final solo fue un error de conexión en la protoboard, esto quiere decir que para que no tengamos ningún error, es necesario checar bien donde y en qué puerto está conectado todo.

Iván

Con la implementación del programa en la GAL y el desarrollo de código y con ayuda de los leds como medio de salida, pudimos comprobar el correcto funcionamiento de la GAL22v10 para la programación de compuertas lógicas.

Jeremy

Mediante la realización de la práctica se logró observar y corroborar el uso de compuertas lógicas mediante el uso de un lenguaje descriptivo de circuitos tal como es el VHDL, así como su comprobación física siguiendo la lógica de las compuertas básicas AND, OR, XOR, NAND, NOR y XNOR

## **Bibliografía.**

L. (s. f.). *Las Compuertas Lógicas y sus Operaciones Lógicas (AND, OR, NOT, NAND, NOR, XOR, XNOR)*. Logicbus S.A. de C.V. Recuperado 15 de mayo de 2022, de <https://www.logicbus.com.mx/compuertas-logicas.php>

## **ANEXOS.**



DIP

