

CENTRO DE CIENCIAS EXACTAS E INGENIERÍAS  
(CUCEI)

Alumna: Mariana Soto Almaguer

Profesor: Jorge Ernesto Lopez Arce Delgado

Curso: Arquitectura de computadoras

Actividad 02- Compuertas lógicas

Fecha de realización: 02/02/2026

## INTRODUCCIÓN

La Unidad Aritmético Lógica (ALU) es el componente del procesador que ejecuta las operaciones esenciales para el procesamiento de datos. Actualmente, estas unidades se implementan en los dispositivos de FPGA (Field Programmable Gate Arrays), que son circuitos integrados reconfigurables capaces de materializar arquitecturas digitales mediante lenguajes de descripción de hardware o HDL.

En este curso utilizaremos el lenguaje Verilog, que a diferencia de la programación convencional, describe la estructura física y el comportamiento del hardware en lugar de una secuencia de instrucciones.

Para esta práctica, se utilizó la sintaxis bitwise de Verilog, la cual permite manipular señales bit a bit para implementar las siete compuertas lógicas principales.

Estos operadores permiten manipular señales bit a bit, traduciendo ecuaciones booleanas directamente en conexiones de transistores dentro de la FPGA. A continuación, se detallan los operadores utilizados.

- AND (&): Realiza una conjunción lógica; la salida es alta sólo si ambas entradas lo son.
- OR (|): Realiza una disyunción; la salida es alta si al menos una entrada es alta.
- NOT (~): Operador de negación o complemento que invierte el estado lógico del bit.
- XOR (^): Puerta de desigualdad; activa la salida sólo si las entradas son diferentes.
- NAND, NOR y XNOR: Se implementan combinando los operadores anteriores con la negación ( $\sim\&$ ,  $\sim|$ ,  $\sim^$ ), permitiendo una descripción concisa de la lógica inversa.

## Las tablas de verdad, simbología y arreglos de transistores

Un **FPGA** o Field Programmable Gate Array es un conjunto de circuitos integrados, como puede ser cualquier chip que no tiene una función fija cuando ha salido de fábrica, es decir, está pensado para que el cliente lo personalice y programe según la tarea que necesite que haga esa placa. Pasaron de ser memoria rígidas (ROM) a dispositivos totalmente reconfigurables.

## VERILOG y su relación con FPGA

Verilog es un lenguaje de descripción de hardware (HDL) estandarizado como IEEE 1364 que permite modelar, simular y sintetizar sistemas digitales a través de diferentes niveles de abstracción. Es ideal para la programación de FPGAs, ya que permite

es la herramienta clave para transformar diseños lógicos en arquitecturas de silicio funcionales, soportando tanto la verificación rigurosa como la implementación final de Circuitos Complejos.

## TABLAS DE VERDAD

Entrada	Salida	
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

Símbolo para la compuerta AND

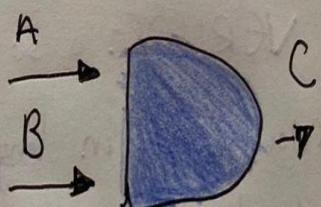


Tabla de verdad AND

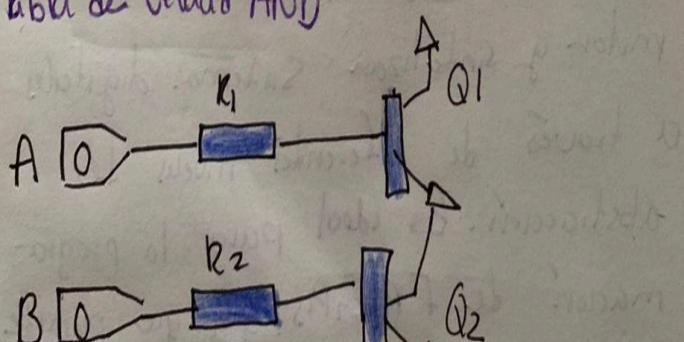
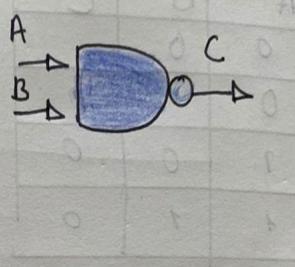


Tabla de verdad NAND

Entrada	Salida	
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Símbolo para la compuerta NAND



Arreglo de transistores NAND

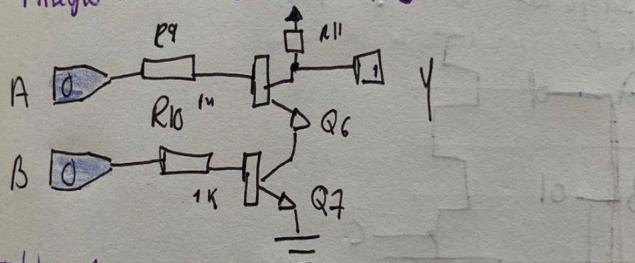


Tabla de verdad OR

Entrada	Salida	
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

Símbolo para compuerta OR

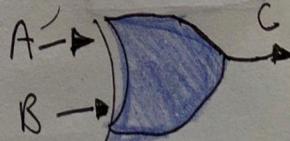
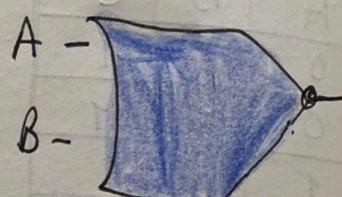


Tabla de verdad NOR

Entrada	Salida	
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

Compuerta NOR



Arreglo de transistores NOR

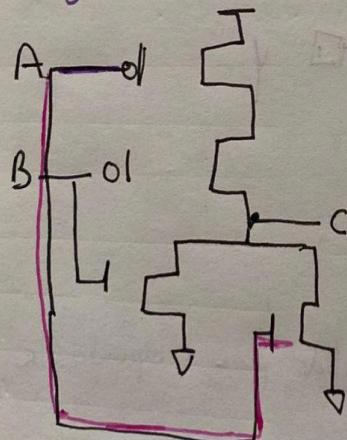


Tabla de verdad NOT

Entrada	Salida
A	C
0	1
1	0

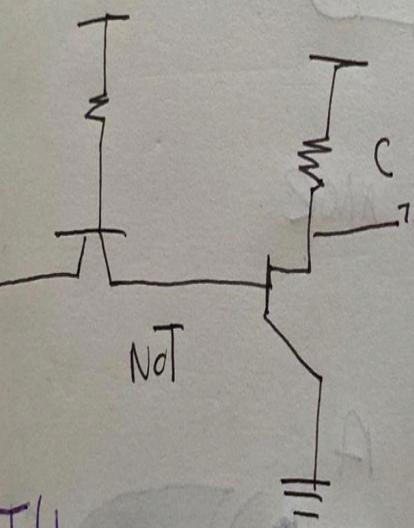
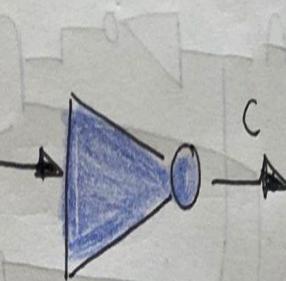
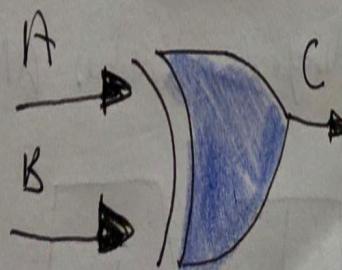


Tabla de verdad NOR

Entrada		Salida
A	B	C
0	0	0
0	1	1
1	0	1
1	1	0



Arreglo de transistores NOR

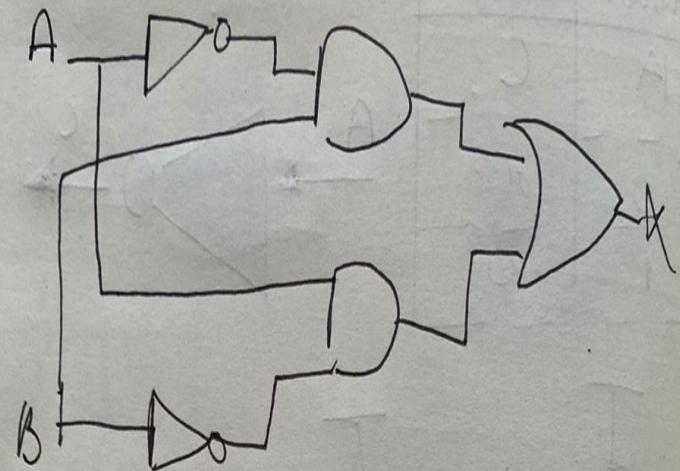
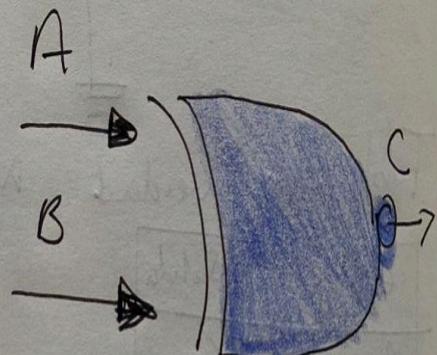
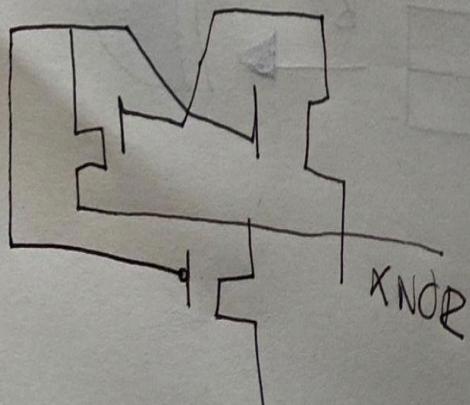


Tabla de verdad ANOR

Entrada		Salida
A	B	C
0	0	1
0	1	0
1	0	0
1	1	1



Arreglo de transistores ANOR



-Objetivo general:

Comprobar la instalación y funcionamiento del compilador y simulador de HDL, así como comprender la implementación de operaciones lógicas básicas mediante el uso del lenguaje de descripción de hardware Verilog.

-Objetivos específicos:

- 1.Verificar la correcta instalación del software Modelsim
2. Investigar la sintaxis de los operadores de lógica combinacional en Verilog.
3. Analizar el funcionamiento de las compuertas lógicas AND,NAND, OR, NOR, NOT, XOR Y XNOR.
- 4.Implementar las siete compuertas lógicas en un solo módulo de Verilog, utilizando dos entradas y siete salidas, con operaciones a nivel de bit.
5. Relacionar el funcionamiento de las compuertas lógicas con la estructura interna de la ALU.

## DESARROLLO / IMPLEMENTACIÓN

Se utilizó Notepad++ para crear un único módulo denominado compuertas principales. El diseño de entrada (A, B) y siete salidas (c1 a c7) para representar las funciones AND, NAND, OR, NOR, NOT, XOR Y XNOR mediante el comando assign y operadores bitwise codificado en Verilog.

Se instaló Modelsim para compilar el archivo.v asegurando la ausencia de errores sintácticos.

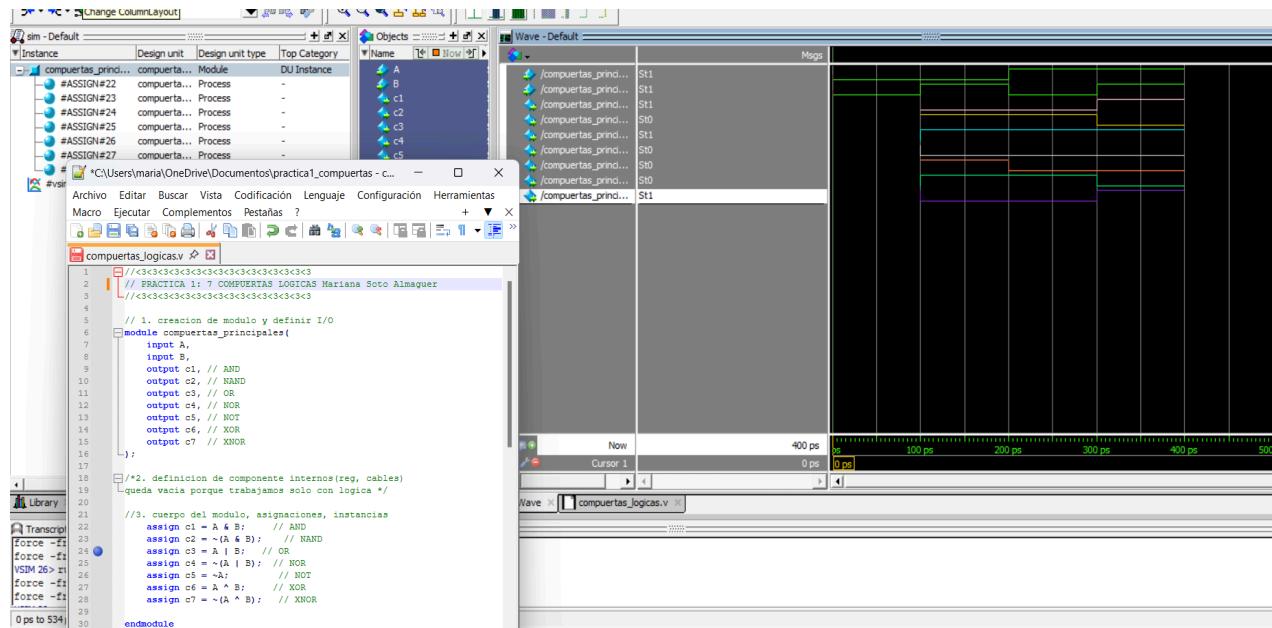
Una vez compilado todo, se añadió una ventana Wave para cada instancia y las entradas en donde se aplicaron estímulos manuales (force) para probar las cuatro combinaciones de tabla de verdad. Se verificó visualmente que las siete salidas cambiaran de estado simultáneamente según su lógica correspondiente, validando el funcionamiento del módulo completo.

Caso 1 (0, 0): Se forzaron A=0 y B=0. En este estado, solo las compuertas de salida negada (NAND, NOR, XNOR) y la NOT de A deben mostrar un "1" lógico, mientras que las demás permanecen en "0".

Caso 2 (0, 1): Se forzó A=0 y B=1. Aquí se activan las compuertas que detectan diferencias (XOR) o presencia de señal (OR), además de las que mantienen su lógica inversa (NAND).

Caso 3 (1, 0): Se forzó A=1 y B=0. Similar al caso anterior, pero sirve para verificar que la compuerta NOT de A ahora cambie a "0".

Caso 4 (1, 1): Se forzaron ambas entradas a 1.. En este punto, la compuerta AND y la OR muestran un "1", mientras que la NAND y la NOR caen a "0".



## CONCLUSIÓN:

Esta actividad, me resultó un desafío técnico, pues no acostumbraba a realizar reportes para mis prácticas por lo que fortalecí mi capacidad para comunicar mis aprendizajes de manera clara y organizada, comprendiendo cómo los conceptos estudiados se traducen en funcionamiento real dentro de un dispositivo digital como la FPGA.

La realización de esta práctica me permitió aplicar los conocimientos de lógica matemática en la implementación de compuertas lógicas con Verilog, reforzando la relación entre teoría y hardware.

## REFERENCIAS:

Virginia, P. B. (s. f.). *Lenguajes de descripción hardware para la síntesis de circuitos: VHDL y Verilog: analogías y diferencias: aplicación a un caso práctico* / Archivo Digital UPM. <https://oa.upm.es/48895/>

Compuertas lógicas. (2020). María Teresa Jímenez Ramírez. <https://repositorio.utn.ac.cr/server/api/core/bitstreams/fe938ab3-7bd0-46e2-b4a4-43f435c50bd3/content>

Rucci, E. (2017, 1 diciembre). *FPGAs: ¿los procesadores del futuro?* <https://sedici.unlp.edu.ar/handle/10915/64231>