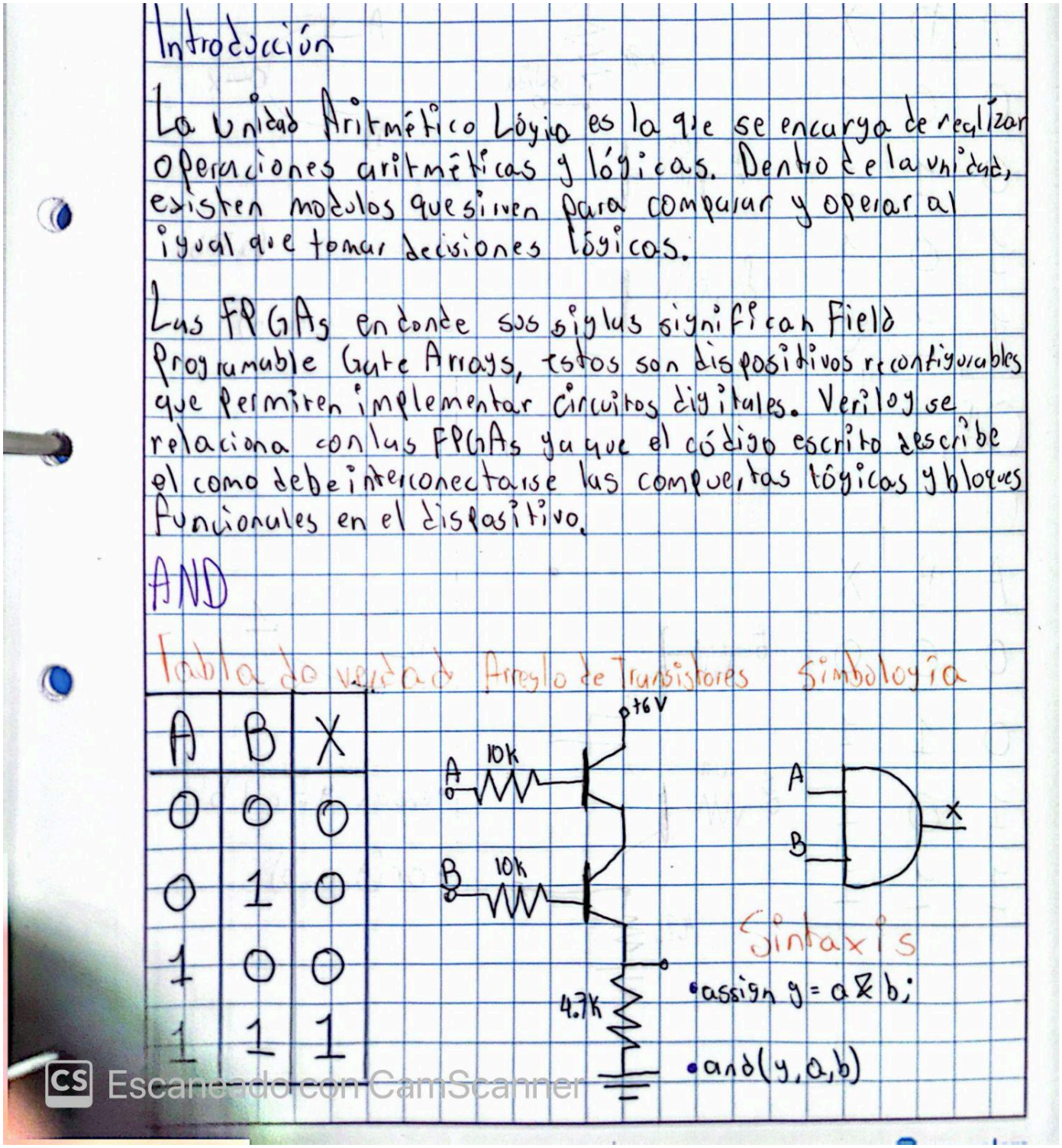
Actividad 02

Unidad Lógica – Implementación de Compuertas Lógicas

Emilio Gomez Breschi

Introducción



Introducción

La Unidad Aritmético Lógica es la que se encarga de realizar operaciones aritméticas y lógicas. Dentro de la unidad, existen modulos que sirven para comparar y operar al igual que tomar decisiones lógicas.

Las FPGAs en donde sus siglas significan Field Programable Gate Arrays, estos son dispositivos reconfigurables que permiten implementar circuitos digitales. Verilog se relaciona con las FPGAs ya que el código escrito describe el como debe interconectarse las compuertas lógicas y bloques funcionales en el dispositivo.

AND

Tabla de verdad

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Arreglo de Transistores

Simbología

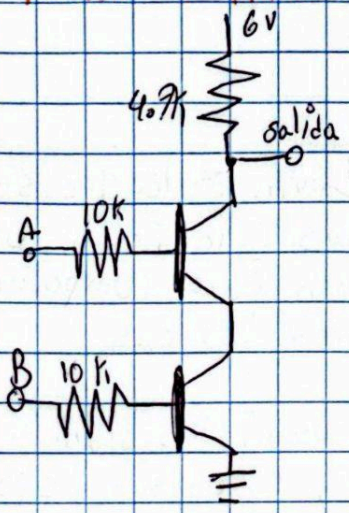
Sintaxis

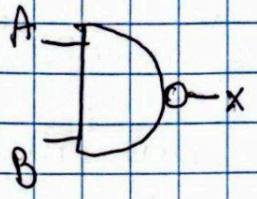
- assign y = a & b;
- and(y, a, b)

NAND

Tabla de Verdad | Arreglo de Transistores | Simbología

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0





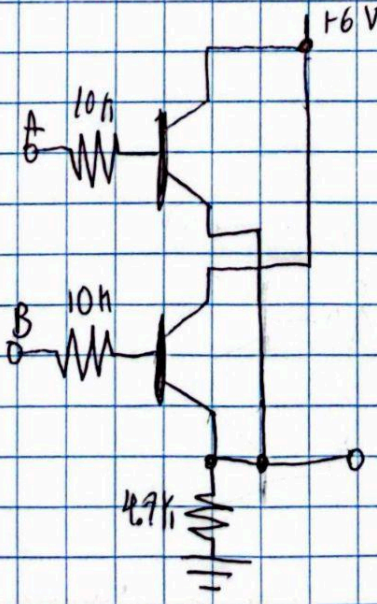
Sintaxis

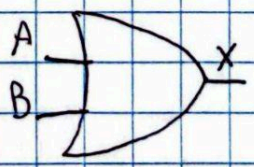
- assign y = ~(a & b)
- nand (y, a, b)

OR

Tabla de Verdad | Arreglo de Transistores | Simbología

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1





Sintaxis

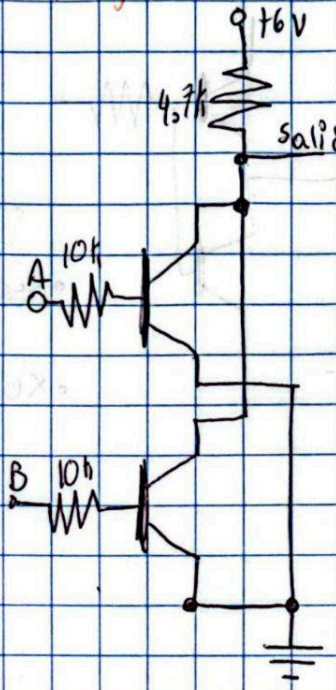
- assign y = a | b
- or (y, a, b)

NOR

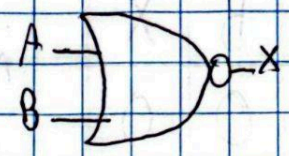
Tabla de Verdad

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Arreglo de Transistores



Simbología



Sintaxis

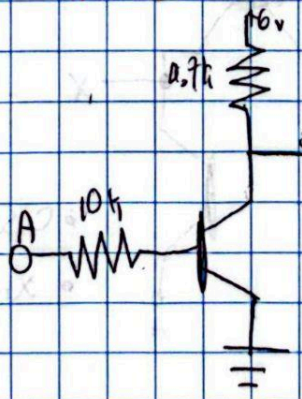
- assign y = ~(a | b)
- nor(y, a, b)

NOT

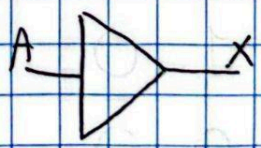
Tabla de Verdad

A	X
0	1
1	0

Arreglo de Transistores



Simbología



Sintaxis

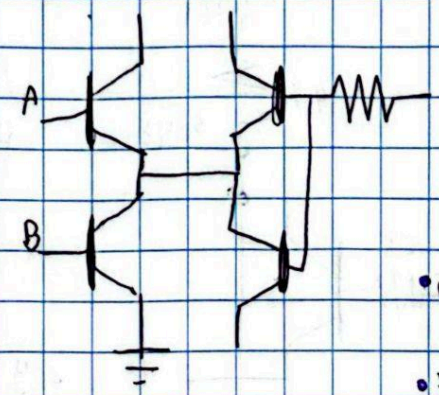
- assign y = ~a;
- not (y, a)

XOR

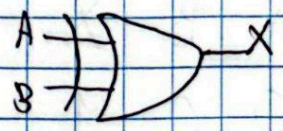
Tabla de Verdad

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Arreglo de Transistores



Simbología



Sintaxis

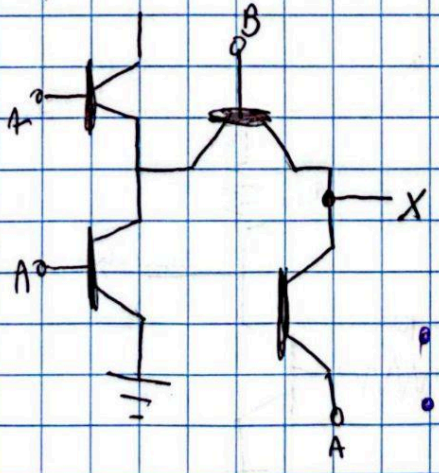
- assign y = a ^ b;
- xor (y, a, b);

XNOR

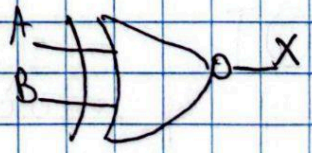
Tabla de Verdad

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

Arreglo de Transistores



Simbología



Sintaxis

- assign y = ~(a ^ b);
- xnor (y, a, b)

Objetivos

Objetivo general

Comprobar la correcta instalación y funcionamiento de ModelSim mediante la programación y simulación de compuertas lógicas usando el lenguaje de Verilog,

Objetivos específicos

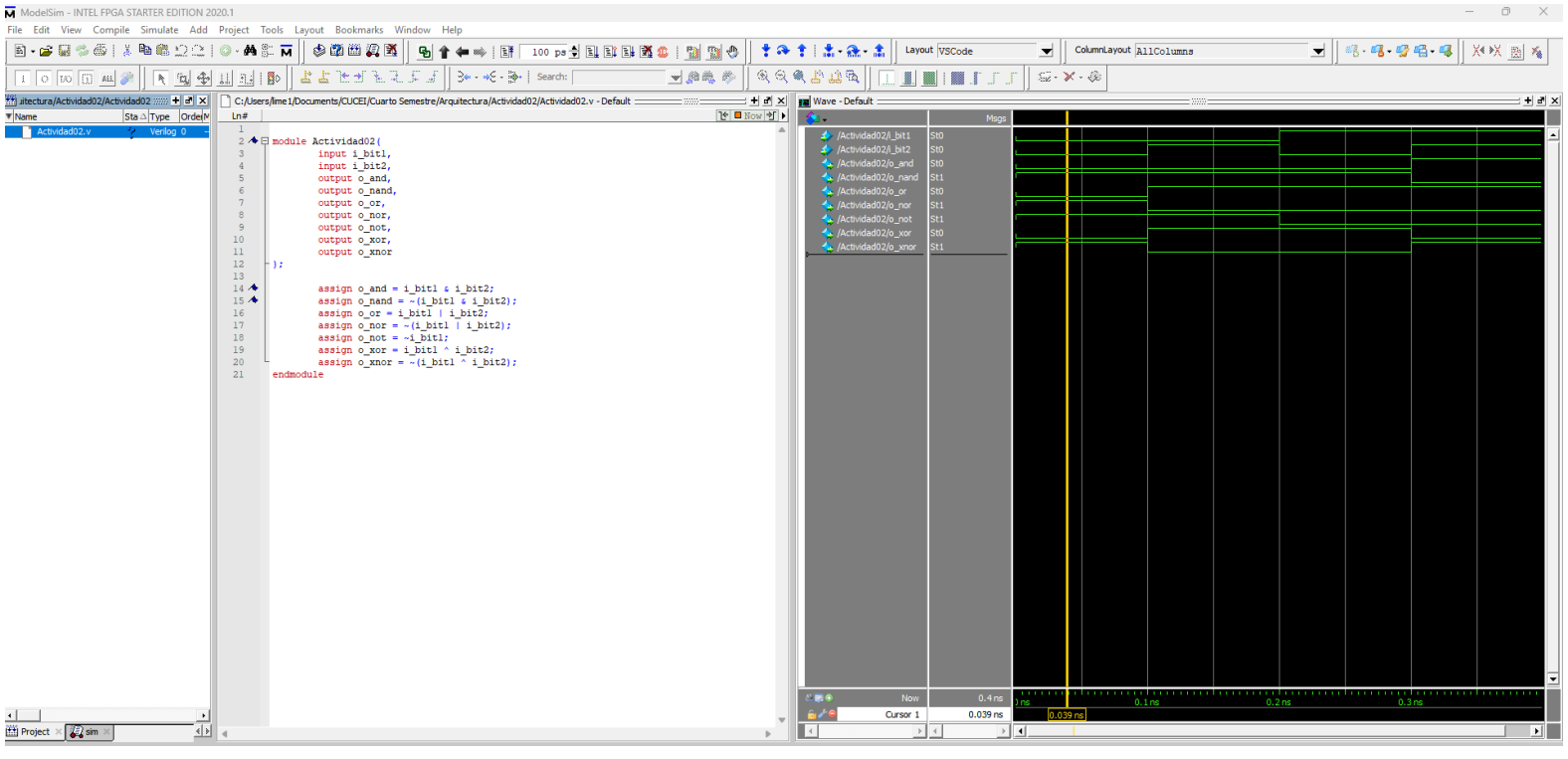
- Investigar el funcionamiento de las compuertas lógicas básicas.
- Analizar la sintaxis de las compuertas lógicas en el lenguaje Verilog.
- Implementar un módulo a nivel de 1 bit con dos entradas y siete salidas.
- Simular el comportamiento de cada compuerta.
- Familiarizarse con el flujo de simulación en ModelSim.

Desarrollo

Para el desarrollo de esta práctica se utilizó el lenguaje Verilog, el cual permite describir circuitos digitales de manera estructurada y clara. Se implementó un solo módulo que contiene dos entradas binarias (i_bit1 e i_bit2) y siete salidas que son las compuertas lógicas.

Cada compuerta fue implementada utilizando los operadores lógicos de Verilog, los cuales permiten describir directamente el comportamiento del circuito. La compuerta NOT utiliza únicamente una de las entradas, mientras que las otras compuertas operan con dos entradas.

El módulo fue simulado en ModelSim, donde se pusieron diferentes combinaciones de entrada para comprobar el comportamiento esperado de cada salida. La simulación permitió observar las ondas, verificando que las salidas coinciden con las tablas de verdad.



Conclusión

La realización de esta práctica me hizo comprender el funcionamiento de las compuertas lógicas y su implementación mediante el lenguaje de verilog. Se aprendieron conceptos de lógica digital y experiencia en el uso de herramientas de simulación como ModelSim.

Aunque la práctica es fácil, es muy importante para entender el funcionamiento interno de sistemas más difíciles como procesadores y compuertas lógicas. Además, permitió identificar la importancia de la simulación previa antes de una implementación física.

Referencias

Compuertas lógicas – Sistemas Digitales. (s. f.).

https://virtual.cuautitlan.unam.mx/intar/sistdig/compuertas-logicas/

TecNM. (s. f.). *TECNM | Tecnológico Nacional de México*. TecNM.

https://www.tecnm.mx/images/areas/docencia01/Apuntes/L%C3%B3gica_Digital.pdf

Wtdtantakatan. (2016, 21 septiembre). *Transistores en compuertas lógicas.* Tantakatanblog.

https://tantakatanblog.wordpress.com/2016/09/18/transistores/