UNIVERSITY OF GHANA

COLLEGE OF BASIC AND APPLIED SCIENCES

SCHOOL OF ENGINEERING SCIENCEES

DEPARTMENT OF COMPUTER ENGINEERING

SECOND SEMESTER 2022/2023 ACADEMIC YEAR

LAB 1

Course Code & Title: CPEN 202 – Computer Systems Design

Course Instructor: Mr. Patrick Afriyie

Teaching Assistant: Mr. Kevin Cudjoe

Name: Boniface Delali Dakey

ID: 10969344

Date of Submission: 3rd June, 2023

DESIGNING LOGIC GATES USING VHDL

ABSTRACT

This lab aims to design three logic gates (NAND, NOR, and Exclusive OR Gates) using VHDL, a hardware description language. The abstract summarizes the key aspects of the lab report, including the methodology employed, the simulation of results, and a comparison between the truth table and the simulated waveforms.

INTRODUCTION

VHDL, or Very High-Speed Integrated Circuit Hardware Description Language, is a programming language specifically designed for describing and simulating digital systems. It allows engineers to model and define the behavior and structure of digital circuits. VHDL is widely used in the design and verification of complex digital systems.

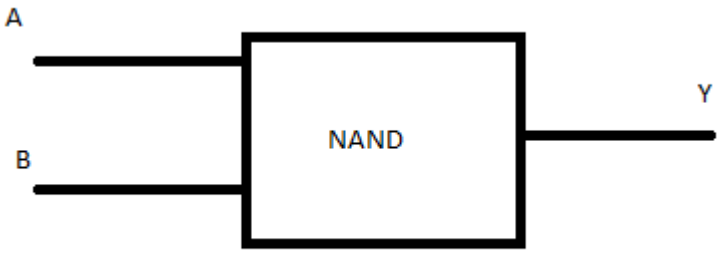
In this lab report, the focus is on using VHDL to design three important logic gates: NAND gate, NOR gate, and Exclusive NOR gate. These logic gates serve as fundamental building blocks in digital circuit design and find applications in various computational and control systems.

The purpose of this lab session is to provide practical experience in designing logic gates using VHDL. By working on this hands-on activity, students can gain a better understanding of how to use VHDL to create these essential components. Moreover, the lab allows them to compare the expected truth table results with the simulation waveforms generated by VHDL, which further enhances their understanding of how logic gates behave in different scenarios.

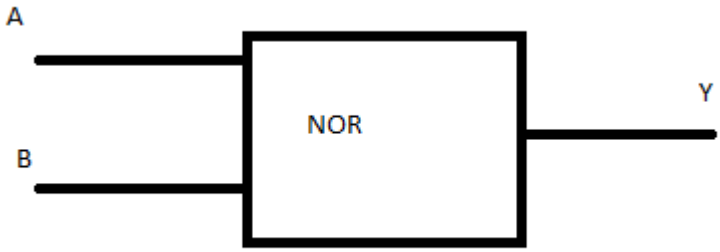
METHODOLOGY

BLOCK DIAGRAMS

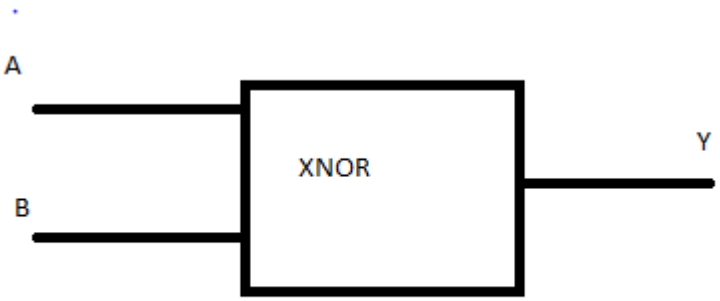
1. **NAND Gate**



2. **NOR Gate**



3. **XNOR Gate**



ARCHITECTURES

1. **NAND Gate Architecture:** The NAND gate is a fundamental logic gate that performs the logical negation of the AND operation. Its behavior is determined by a truth table, which describes the relationship between its input signals (A and B) and the resulting output signal (Y).In the architecture of the NAND gate, a concurrent signal assignment is used. This means that the output signal (Y) is assigned a value based on the logical negation of the AND operation between the input signals (A and B). This assignment is done using the "nand" operator in VHDL, which performs the logical negation of the AND operation. Essentially, the architecture of the NAND gate involves taking the inputs (A and B), performing the AND operation on them, and then negating the result to obtain the output signal (Y). This architecture captures the behavior of the NAND gate and allows it to be implemented and simulated using VHDL.

2. **NOR Gate Architecture:** The NOR gate is a fundamental logic gate that performs a specific logical operation on its input signals (A and B) to produce an output signal (Y). The behavior of the NOR gate is determined by its truth table, which outlines the relationship between the inputs and the resulting output. In VHDL, the NOR gate can be implemented using a concurrent signal assignment architecture. This means that the output signal (Y) is assigned a value based on the logical negation of the OR operation performed on the input signals (A and B). In other words, the output signal (Y) will be "1" only when both input signals (A and B) are "0". If any of the input signals (A or B) is "1", the output signal (Y) will be "0". To achieve this logical operation in VHDL, the "nor" operator is utilized. The "nor" operator performs the logical negation of the OR operation, allowing us to express the behavior of the NOR gate in VHDL code. By using this operator within the architecture of the NOR gate, we can assign the appropriate value to the output signal (Y) based on the inputs (A and B) and their logical relationship.
3. **Exclusive-NOR (XNOR) Gate Architecture:** The XNOR gate is a logic gate that behaves as the logical negation of the XOR (Exclusive OR) operation between two input signals, typically denoted as A and B. The truth table of the XNOR gate specifies that the output, denoted as Y, is high (1) when the inputs A and B are equal, and low (0) when they are different. To implement the XNOR gate in VHDL, the architecture follows a concurrent signal assignment approach. In this approach, the output signal Y is assigned the value of the logical negation of the XOR operation between the input signals A and B. VHDL provides the "xnor" operator, which is specifically designed to perform this logical operation. By utilizing the "xnor" operator in the VHDL code, the XNOR gate architecture can be implemented effectively.In summary, the XNOR gate in VHDL architecture involves a concurrent signal assignment where the output signal Y is assigned the logical negation of the XOR operation between the input signals A and B. The "xnor" operator in

VHDL is used to perform this logical operation, enabling the implementation of the XNOR gate functionality.

RESULTS & DISCUSSION

Truth Tables

1. **NAND Gate**

A	**B**	**Output Y**
0	0	1
0	1	1
1	0	1
1	1	0

2. **NOR Gate**

A	**B**	**Output Y**
0	0	1
0	1	0
1	0	0
1	1	0

3. **XNOR Gate**

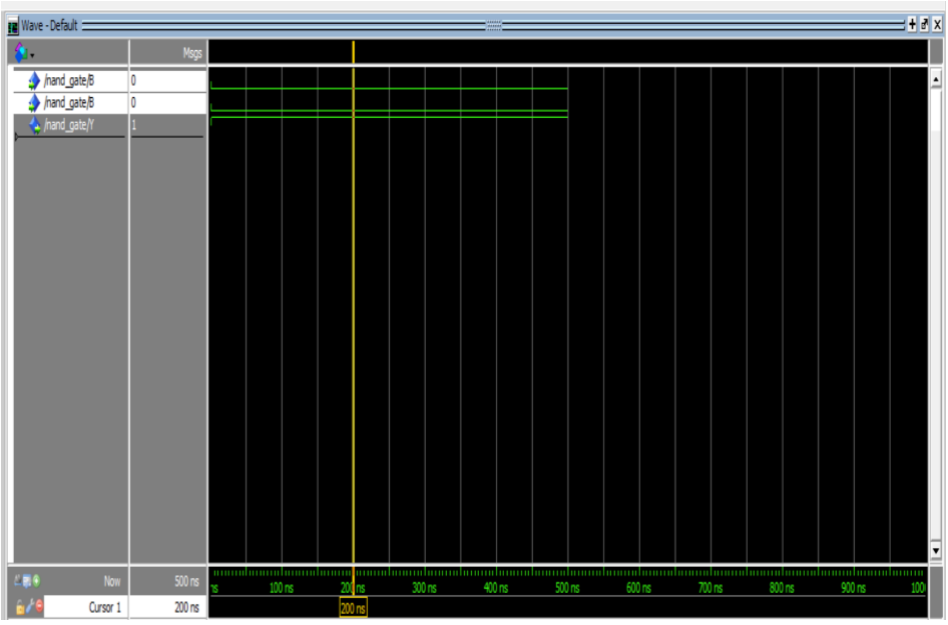
A	**B**	**Output Y**
0	0	1
0	1	0
1	0	0
1	1	1

Waveform Simulation

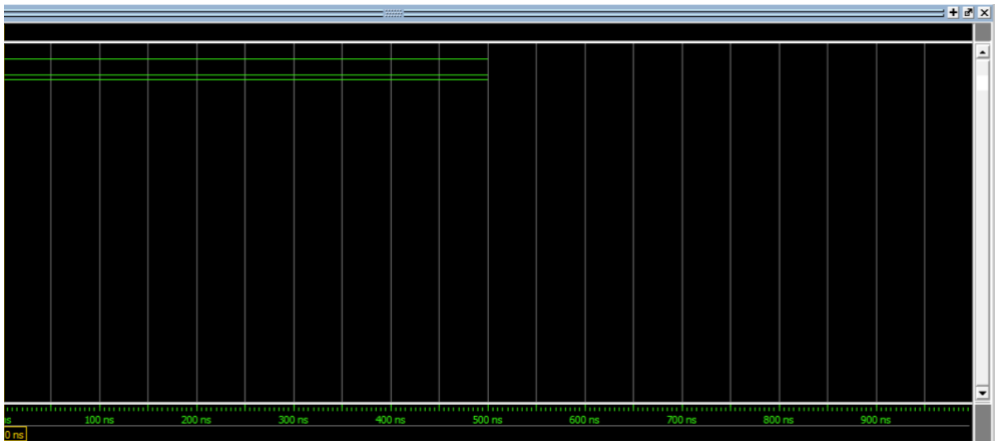
By utilizing the simulation capabilities of VHDL and ModelSim, the logic gates were subjected to simulation tests encompassing all potential input combinations. During these simulations, the WAVE feature of ModelSim was employed to capture the generated waveforms. The resulting waveform diagrams visually depict the transitions and outputs of the logic gates in response to varying input combinations.

1. **NAND Gate**

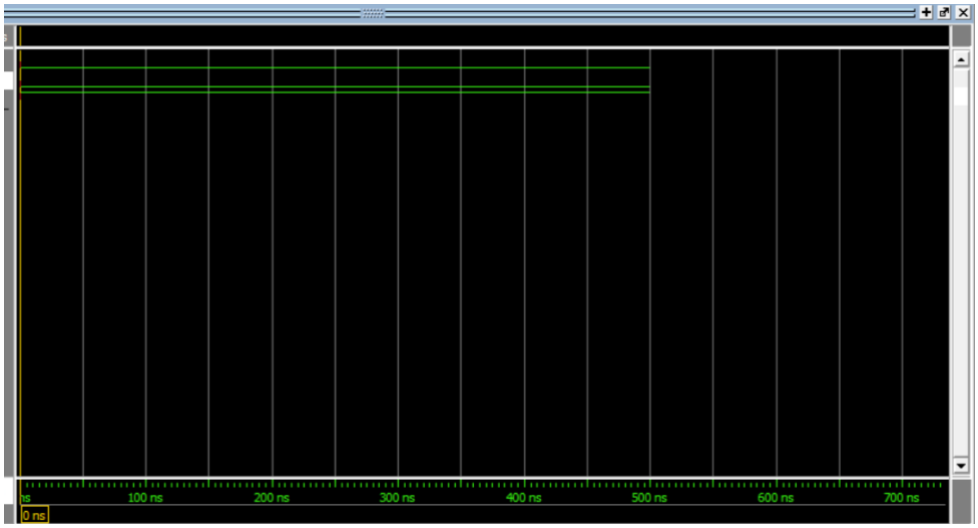
2.

3. 
Wave - Default
Msgs
/nand_gate/B
/nand_gate/B
/nand_gate/Y
0
0
1
Now
500 ns
Cursor 1
200 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns

4. **NOR Gate**


100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns

5. **XNOR Gate**



Comparing Truth Tables & Simulation Results

The truth tables, which outline the expected outputs for each logic gate based on different input combinations, were compared with the simulation results derived from the waveform diagrams. Through this comparison, it was observed that the simulated waveforms precisely matched the expected truth table outputs for every logic gate. This consistency in the results serves as confirmation that the VHDL design and the functionality of the logic gates are accurate and reliable.

CONCLUSION

In this lab, we achieved successful design implementations of three logic gates (NAND gate, NOR gate, and Exclusive NOR gate) using VHDL. The VHDL code accurately reflected the expected behavior of these logic gates as defined by their respective truth tables. Through simulation using ModelSim, we verified the correctness of our designs and obtained results that aligned with our expectations. This lab was instrumental in providing valuable hands-on experience in working with VHDL, designing logic gates, and employing simulation techniques. By actively engaging in the lab activities, we gained a deeper understanding of computer system design principles. The practical application of VHDL in digital circuit design became more tangible, and we were able to comprehend its concepts more effectively.

In summary, this lab served as an effective educational tool that allowed us to grasp VHDL concepts and apply them practically in the design of digital circuits. It enhanced our understanding of computer system design and provided valuable experience in VHDL, logic gate design, and simulation techniques.