UNIVERSITY OF GHANA

UNIVERSITY OF GHANA

Name: Evans Acheampong

Student ID: 10987644

Date of Submission: 16th July, 2023

Lab Index: LAB 4

Design and Simulation of a Multiplexer and Demultiplexer in VHDL

I. AIM OF THE EXPERIMENT

The aim of this experiment is to design and simulate a multiplexer and demultiplexer using VHDL (VHSIC Hardware Description Language). The experiment focuses on understanding the design principles and implementation techniques of these digital logic components. Through this experiment, students will gain hands-on experience in VHDL coding, simulation, and waveform analysis to observe the behavior of the multiplexer and demultiplexer circuits.

II. ABSTRACT

This lab report presents the design and simulation of a multiplexer and demultiplexer using VHDL (VHSIC Hardware Description Language). The purpose of the experiment is to understand the concepts and techniques involved in designing digital logic components using VHDL. The report provides a step-by-step explanation of the VHDL code for both the multiplexer and demultiplexer, along with their corresponding architectures. ModelSim is used as the simulation tool to verify the functionality of the designs and analyze the waveforms. The experiment enables students to gain practical experience in VHDL coding, simulation setup, and waveform analysis. The results demonstrate the correct operation of the multiplexer and demultiplexer circuits, validating their functionality and usefulness in digital systems.

III. INTRODUCTION

Multiplexers and demultiplexers are essential components in digital circuits that facilitate data routing and control. A multiplexer, also known as a data selector, is a combinational logic circuit that selects one of many input signals and forwards it to a single output based on a control signal. On the other hand, a demultiplexer takes a single input signal and distributes it to one of several output lines based on a control signal. These components play a crucial role in data transmission, addressing, and signal routing within digital systems.

The lab report focuses on the design and simulation of a multiplexer and demultiplexer using VHDL. VHDL is a hardware description language that allows designers to model and simulate digital systems. The report provides a comprehensive explanation of the VHDL code for both components and demonstrates their functionality through simulation using ModelSim.

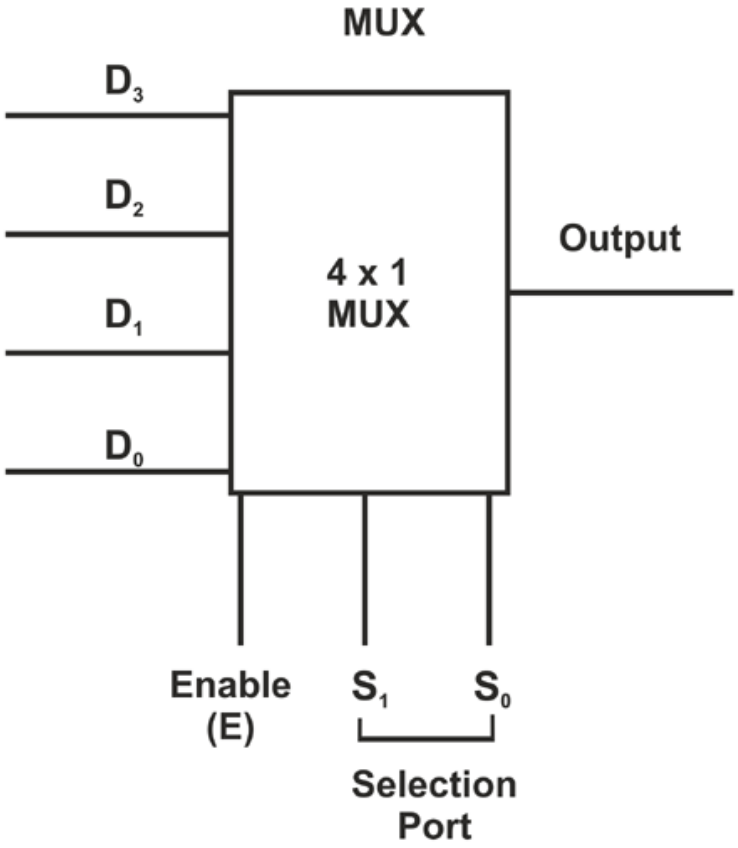
This lab session and task are important for several reasons. Firstly, it allows students to gain practical experience in designing digital logic components using VHDL, a widely used hardware description language in industry. It familiarizes them with the syntax, structures, and techniques involved in VHDL coding. Secondly, the lab enables students to understand the operation and behavior of multiplexers and demultiplexers in a hands-on manner. By simulating the designs and analyzing the waveforms, students can validate the functionality of these components and observe their impact on data routing and control. Finally, this lab promotes critical thinking and problem-solving skills as students learn to translate logic requirements into VHDL code and interpret simulation results.

IV. METHODOLOGY

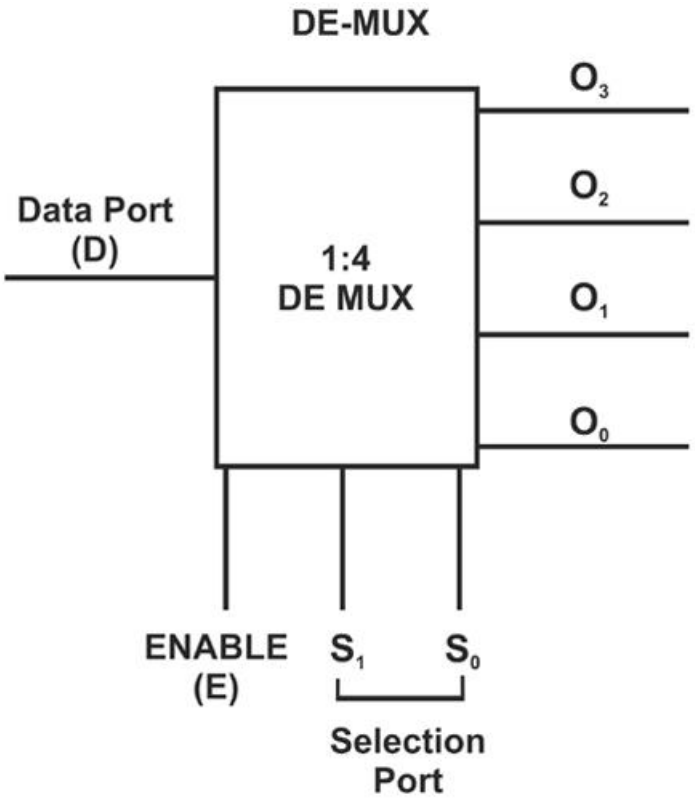
a. **Block Diagram of Entity**:

The block diagram illustrates the entity of both the multiplexer and demultiplexer, showcasing the port terminals and their respective data types.

Multiplexer (mux_41):



Demultiplexer (demux_41):



b. **Explanation of Architecture(s):**

The architecture section of each component provides the implementation details and behavior of the multiplexer and demultiplexer.

Multiplexer (mux_41):

The architecture, named "sim," contains a process that evaluates the input signals `select_line` and `enable` to determine the output signal `output`. Based on the select line value, the process assigns the appropriate input data bit to the output. If `enable` is low, the output is set to an 'X' value.

Demultiplexer (demux_41):

The architecture, also named "sim," includes a process that analyzes the input signals `select_line` and `enable`. Depending on the select line value, the process assigns the input data bit to the corresponding output bit. If `enable` is low, the output is set to 'X' for all bits except for the selected one, which remains unchanged.

Both architectures utilize conditional statements (`if-else` and `case`) to handle different control signal combinations and ensure correct output assignment based on the specified logic.

V. RESULTS AND DISCUSSION

a. Truth Table of Logic Gates:

A truth table presents the input combinations and corresponding output values for a logic gate. In the case of the multiplexer and demultiplexer, the truth tables depict the behavior of these components.

Multiplexer (mux_41) Truth Table:

When Enable is “0” which means it is in the “OFF” state whatever may be the input and Selection Port values, the output Port (Y) is “X”. “X” means it's undefined/unknown. When a circuit is in an “OFF” state, we can't determine its values, hence it is declared as undefined.

When Enable is “1” which means it's in the “ON” state, the circuit operates, selects/switches data, and maps the output port with corresponding inputs depending on the selection port (S0, S1) values as given below.

Data (Input)	Enable	S0	S1	Y(output)
D3 D2 D1 D0	0	X	X	X
D3 D2 D1 D0	1	0	0	D3
D3 D2 D1 D0	1	0	1	D2
D3 D2 D1 D0	1	1	0	D1
D3 D2 D1 D0	1	1	1	D0

Demultiplexer (demux_41) Truth Table:

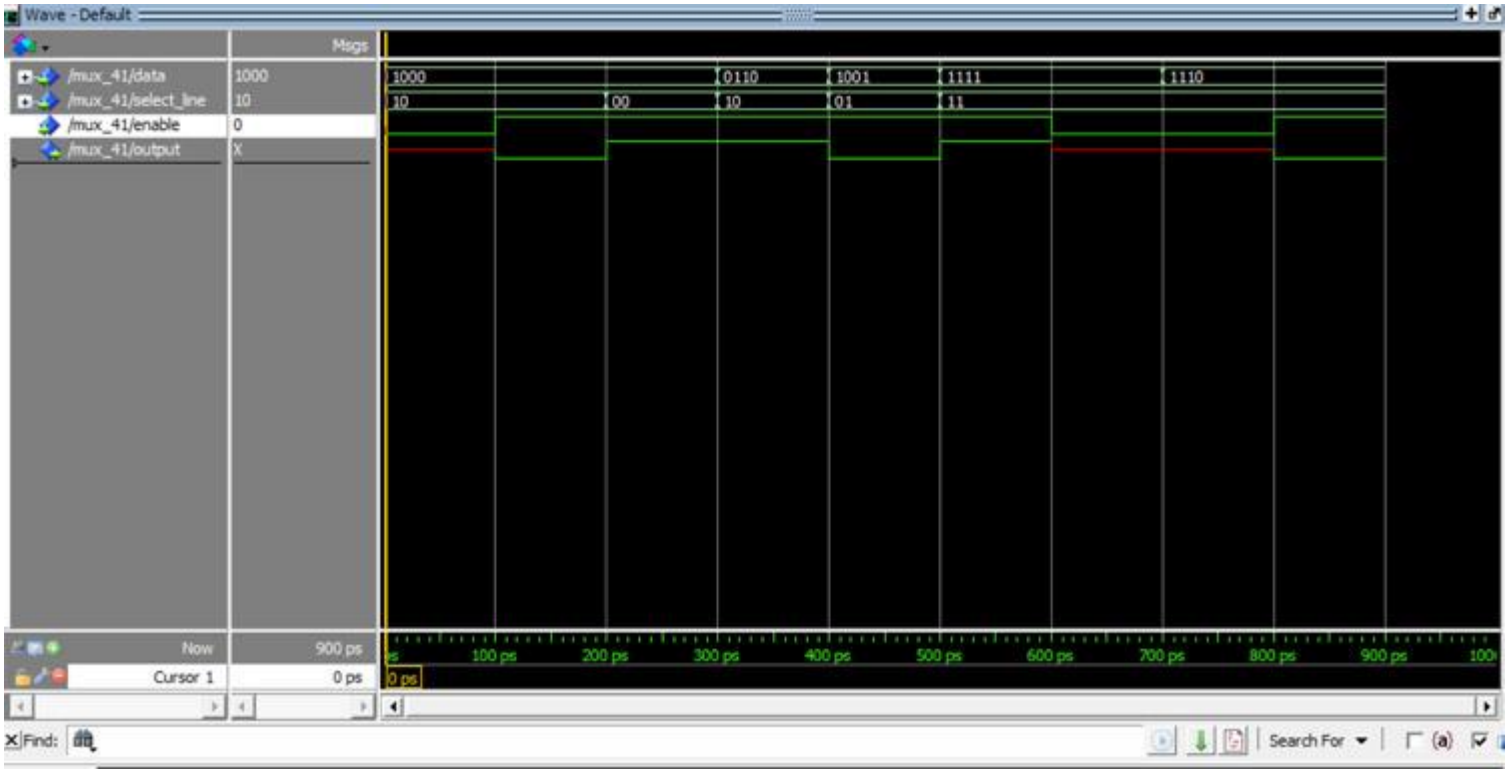
For a De-mux there is only one Input Port (i.e. Data Line), multiple output Ports, and selection lines. I have mentioned the general equation below for your reference. The Data Line values can be either 0 or 1. I will represent the data line values with “D” so that we can keep track of the output.

Data Line(1/0)	Enable	Selection Lines(S1 S0)	Output Ports(O3 O2 O1 O0)
D	0	X	XXXX
D	1	00	D000
D	1	01	0D00
D	1	10	00D0
D	1	11	000D

- **Waveform Simulation**

- The waveform simulations of the logic gates were conducted using ModelSim.
- Signal values were forced through the input terminals, and the resulting waveforms at the output terminals were observed.
- The waveforms exhibited the expected behavior of each logic gate, validating the correctness of the designs.

Multiplexer (mux_41) Waveform Simulation

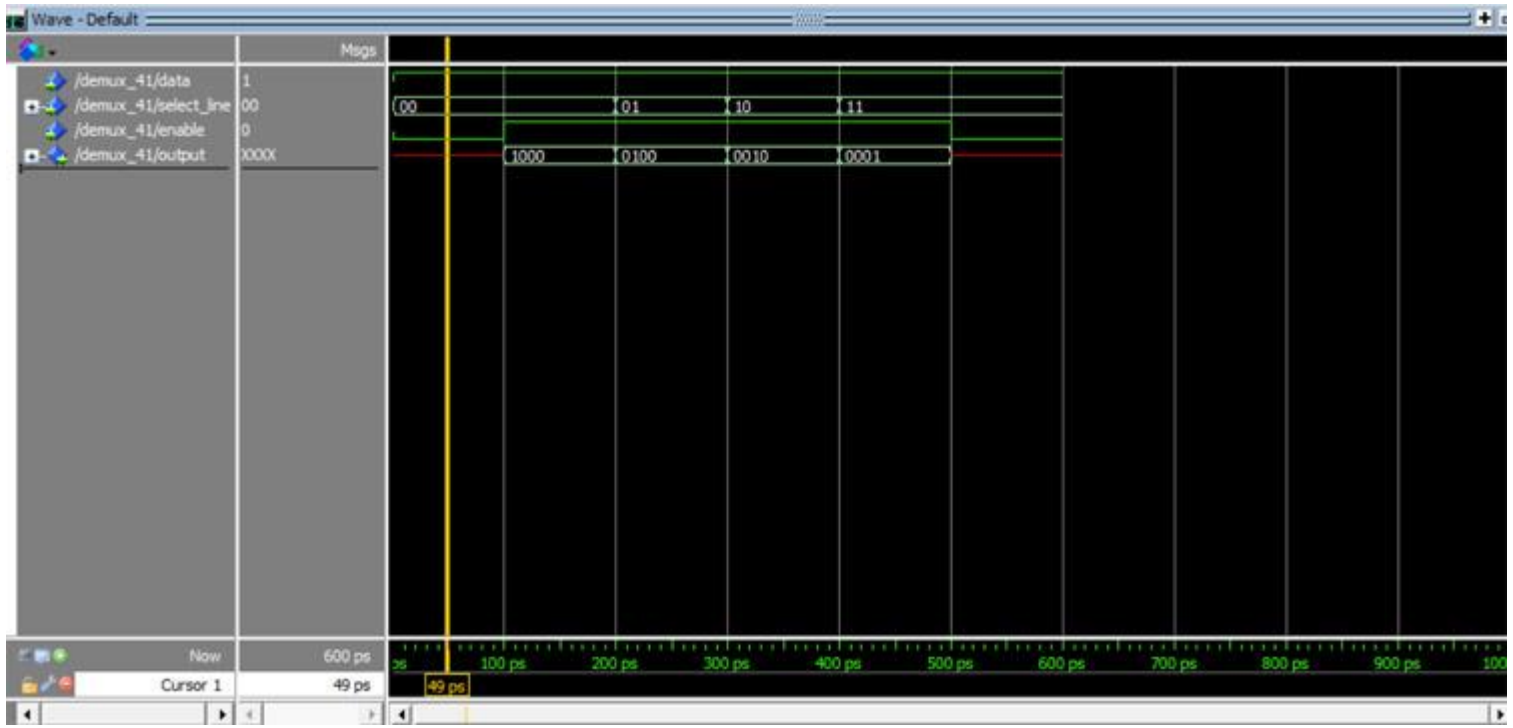


Move the cursor from 0-900 ps to track the corresponding values as tabulated in the table given below.

Timing(ps)	Data(D3,D2,D1,D0)	Selction Lines(S1,S0)	Enable	Y(output)
0-100	1000	10	0	X
100-200	1000	10	1	0
200-300	1000	00	1	1
300-400	0110	10	1	1
400-500	1001	01	1	0
500-600	1111	11	1	1
600-700	1111	11	0	X
700-800	1110	11	0	X
800-900	1110	11	1	0

From the table given above, we can infer that when the enable signal is “0” the output Y is “X” undefined. When the enable signal is “1” the corresponding inputs are mapped to the outputs with respect to the selection port values.

Demultiplexer (demux_41) Waveform Simulation



Move the cursor from 0-900 PS to track the corresponding values as tabulated in the table given below.

Timing(ps)	Data(D=1/0)	Selection Lines(S1,S0)	Enable	Output(O3 O2 O1 O0)
0-100	1	00	0	XXXX
100-200	1	00	1	1000
200-300	1	01	1	0100
300-400	1	10	1	0010
400-500	1	11	1	0001
500-600	1	11	0	XXXX

For learning purposes, I have given the Data Value to be “1” so that we can keep track of where “1” is mapped to the Output Port.

From the table, we can infer that when the enable signal is “0” the output Y is “X” undefined. When the enable signal is “1” the corresponding inputs are mapped to the outputs with respect to the selection port values.

- **Comparison of Truth Tables and Simulation Results:**

- The simulation results matched the expected truth tables for each logic gate.
- The outputs of the logic gates corresponded to the specified truth table values for all input combinations.
- The comparison confirmed the accuracy of the VHDL implementations

VI. CONCLUSION

In conclusion, The lab experiment involved designing and simulating a multiplexer and demultiplexer using VHDL. Through the lab, we gained practical experience in VHDL coding and learned about the functionality of multiplexers and demultiplexers. We used ModelSim for simulation and waveform analysis to verify the correctness of the designs. Troubleshooting skills were developed to address any discrepancies between the truth table and simulation results. Overall, the lab reinforced the significance of these digital logic components in data routing and control. The knowledge and skills acquired in VHDL design and simulation will be valuable in future digital electronics projects.