UNIVERSITY OF GHANA

Name: Boniface Delali Dakey

Student ID: 10969344

Date of Submission: 16th July, 2023

Lab Index: LAB 4

Design and Simulation of a Multiplexer and Demultiplexer in VHDL

- **AIM OF EXPERIMENT**

The objective of this study is to create and simulate a multiplexer and demultiplexer using VHDL (VHSIC Hardware Description Language). The main purpose is to comprehend the principles behind designing and implementing these digital logic components. By conducting this experiment, students will acquire practical skills in VHDL coding, simulation, and analyzing waveforms to observe how the multiplexer and demultiplexer circuits behave.

- **ABSTRACT**

This lab report details the development and simulation of a multiplexer and demultiplexer using VHDL (VHSIC Hardware Description Language). The experiment's objective is to grasp the concepts and methodologies involved in designing digital logic components with VHDL. The report presents a step-by-step explanation of the VHDL code for both the multiplexer and demultiplexer, including their respective architectures. ModelSim is employed as the simulation tool to verify the designs' functionality and analyze the waveforms. This experiment allows students to acquire hands-on experience in VHDL coding, simulation configuration, and waveform analysis. The results showcase the accurate operation of the multiplexer and demultiplexer circuits, affirming their reliability and relevance in digital systems.

- **INTRODUCTION**

Multiplexers and demultiplexers are fundamental 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 input signal from multiple options and transmits it as a single output based on a control signal. Conversely, 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 vital role in data transmission, addressing, and signal routing within digital systems.

The lab report focuses on designing and simulating a multiplexer and demultiplexer using VHDL, a hardware description language that enables 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 hold significant importance for several reasons. Firstly, it offers students practical experience in designing digital logic components using VHDL, which is widely utilized in the industry as a hardware description language. It acquaints them with VHDL coding syntax, structures, and techniques. Secondly, the lab allows students to understand the operation and behavior of multiplexers and demultiplexers through hands-on experimentation. By simulating the designs and analyzing waveform outputs, students can verify the functionality of these components and observe their impact on data routing and control. Lastly, this lab encourages critical thinking and problem-solving skills as students learn to translate logic requirements into VHDL code and interpret simulation results.

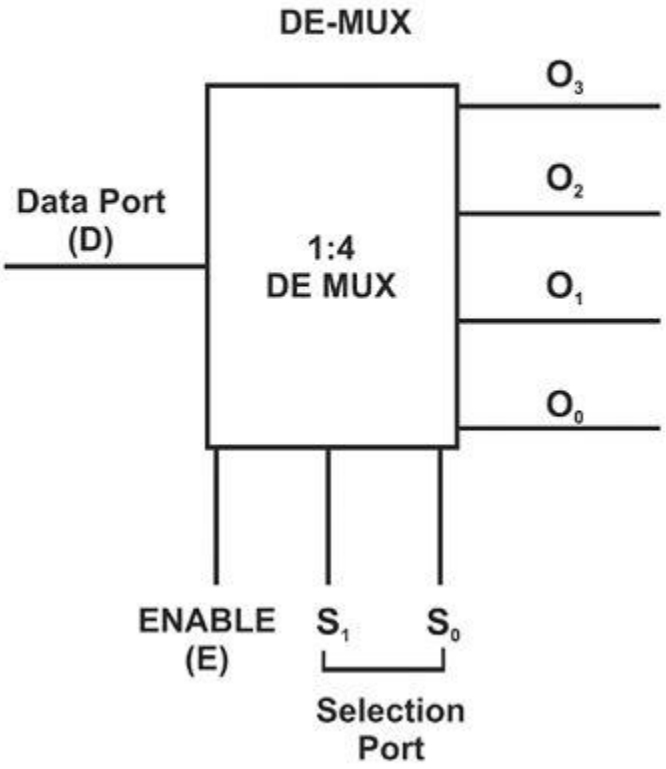
- **METHODOLOGY**

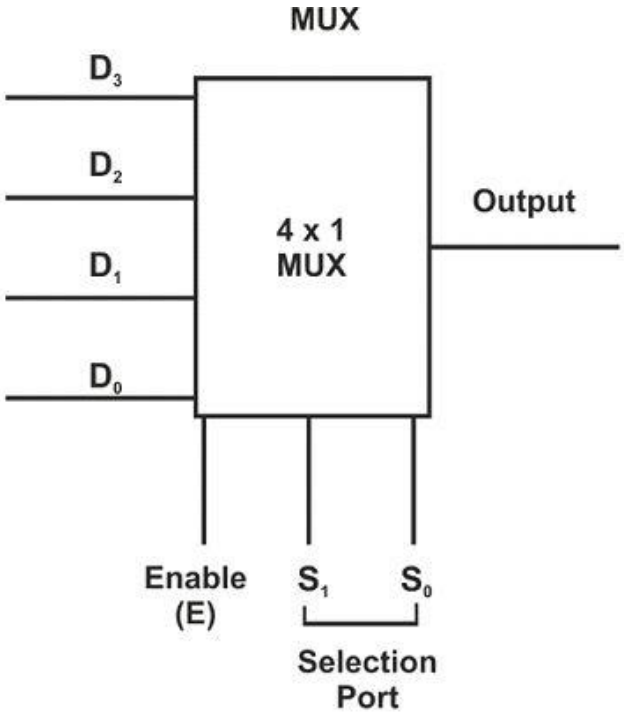
a. Block Diagram of Entity:

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 presents the implementation details and behavior of the multiplexer and demultiplexer.

Multiplexer (mux_41): The "sim" architecture includes a process that evaluates the input signals select_line and enables to determine the output signal output. By considering the value of the select line, 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 "sim" architecture consists of a process that examines the input signals select_line and enable. Depending on the value of the select line, 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 (such as if-else and case) to handle different combinations of control signals. This ensures that the output assignment follows the specified logic correctly.

RESULTS AND DISCUSSION

a. Truth Table of Logic Gates:

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

Multiplexer (mux_41) Truth Table:

When the Enable signal is "0," indicating an "OFF" state, regardless of the input and Selection Port values, the output Port (Y) is designated as "X." The symbol "X" denotes an undefined or unknown value. When a circuit is in an "OFF" state, its values cannot be determined, thus they are declared as undefined. When the Enable signal is "1," signifying an "ON" state, the circuit becomes operational, selects or switches data, and maps the output port based on the corresponding input and selection port (S0, S1) values, as shown below:

Data (Input)	Enable	S0	S1	Y(Output)
D 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:

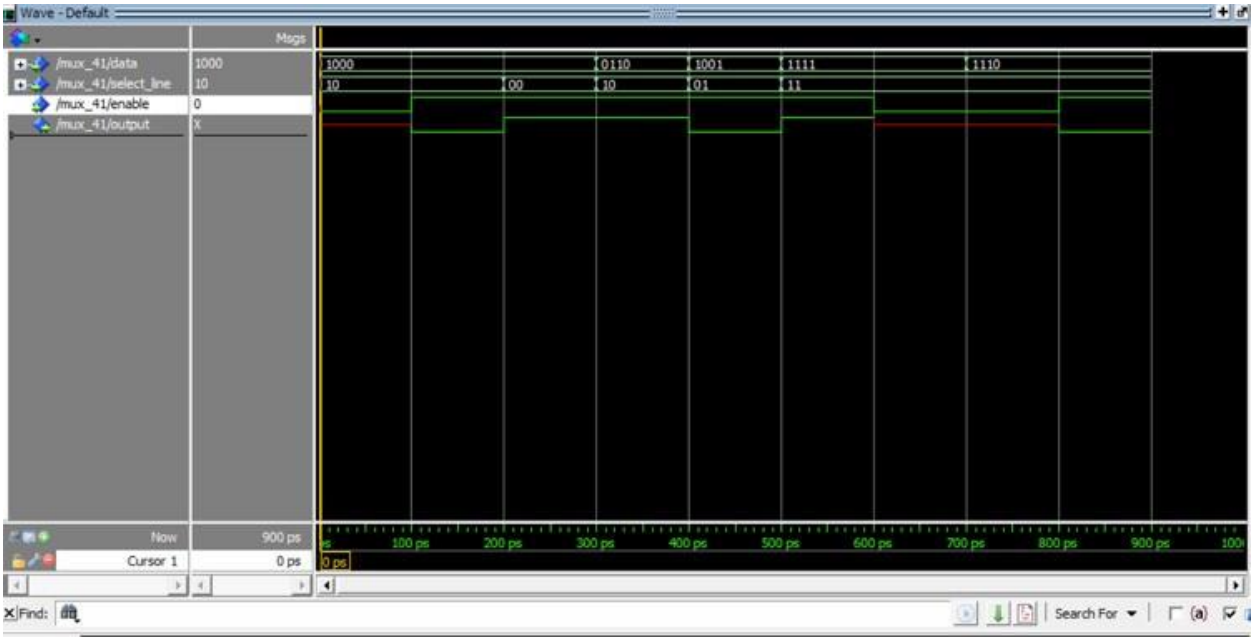
In the case of a Demultiplexer (De-mux), there is only one Input Port (Data Line), multiple output Ports, and selection lines. The general equation is mentioned below for reference. The Data Line values can be either 0 or 1, denoted as "D" to keep track of the output.

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.

MULTIPLEXOR WAVEFORM SIMULATION

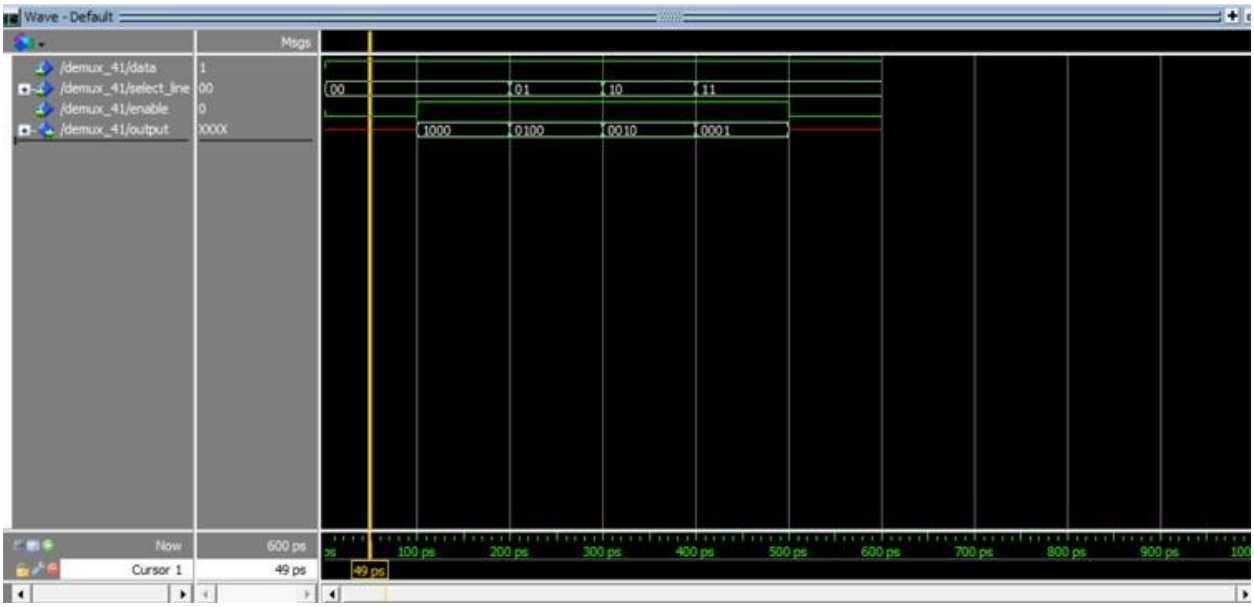


Timing(ps)	Data(D3,D2,D1,D0)	Selction Lines(S1,S0)	Enable	Y(output)
0-100	1000	10	0	X
100-200	1000	10	1	0
200-300	1000	00	1	1
300-400	0110	10	1	1
400-500	1001	01	1	0
500-600	1111	11	1	1
600-700	1111	11	0	X
700-800	1110	11	0	X
800-900	1110	11	1	0

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

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.

DEMULTIPLEXOR 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

CONCLUSION

In summary, the lab experiment revolved around the design and simulation of a multiplexer and demultiplexer using VHDL. Throughout the lab, we obtained hands-on experience in VHDL coding and deepened our understanding of the functionalities of multiplexers and demultiplexers. We utilized ModelSim for simulation and waveform analysis, ensuring the accuracy of our designs. Troubleshooting skills were honed to address any discrepancies observed between the truth table and simulation outcomes. Overall, the lab underscored the importance of these digital logic components in data routing and control. The knowledge and skills acquired in VHDL design and simulation will prove valuable for future endeavors in the realm of digital electronics.