California State University, Northridge

Department of Electrical & Computer Engineering



ECE 526L

Lab Report 1

4:1 Multiplexer

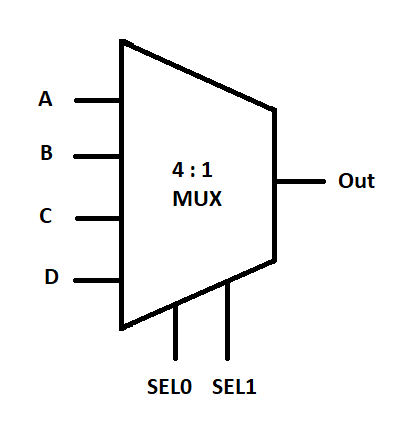
By

Avinash Damse

CSUN ID- 203131064

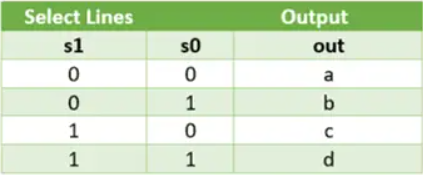
**1: Introduction**

The objective of this lab is to build a 4:1 Multiplexer using our Verilog skills. A multiplexer is a data selector device that selects one input from several input lines, depending upon the enabled, select lines, and yields one single output. A multiplexer of 2n inputs has n select lines and are used to select which input line to send to the output. There is only one output in the multiplexer, no matter what’s its configuration.



### **Truth table**

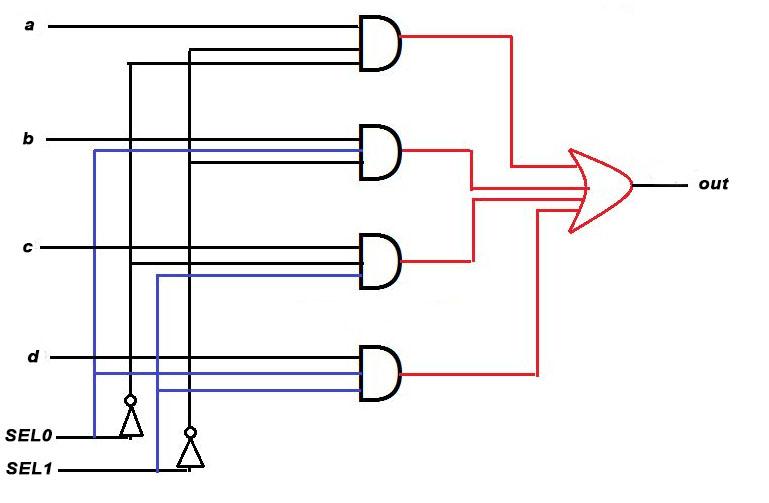
The truth table of the 4:1 MUX has six input variables, out of which two are select lines((SEL0,SEL1), and one is the output signal(Out). The input data lines A, B, C, D are selected depending on the values of the select lines.



**2: Procedure**

**a. Part 1: Creating Mux4\_1 Module**

In this lab I have created a module for a 4:1 multiplexer . Inside the module I have assigned “a”, “b”,“c”, “d” as input variables and “out” as output variables and “SEL0, SEL1” as select variables. Then I performed “AND”,”NOT” and “OR” operations. According to the circuit diagram shown below. After completing the code I ended the module using and saved the file with name “mux4\_1.v”.



**b. Part 2: Creating mux4\_1\_tb Module**

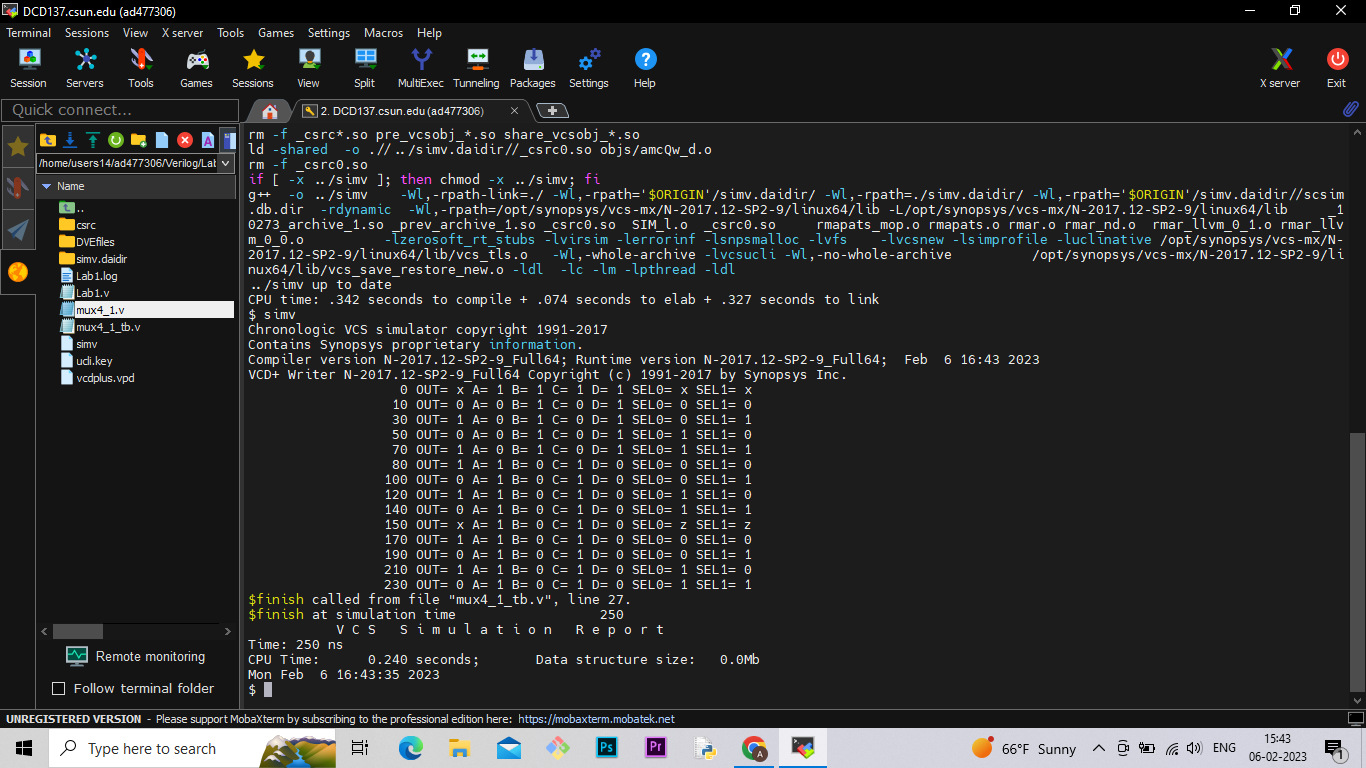
I have written the testbench for the mux4\_1 module code after creating my module. We require testbench just to make sure that the module we have created is working properly. In this lab, I have written the testbench and saved the file as “mux4\_1\_tb.v”.

**c. Part 3: Checking the 4:1 mux and Test Bench Module**

After completing the 4:1 Mux module and testbench module , we have to check if the code is running properly or not. To make sure that the code is running perfectly or not I have run the VCS command followed by the mux4\_1.v and mux4\_1\_tb.v module. I have used the command “vcs -debug -full64 mux4\_1.v mux4\_1\_tb.v” and checked if any error occurs. I found one error in the code, with the help of the “gedit” command. I edited the code and re-run again. This time code executed without any error.

**d. Part 4: Simulation**

After completing the code and testbench for 4:1 Mux I have run the command “simv” for simulation.

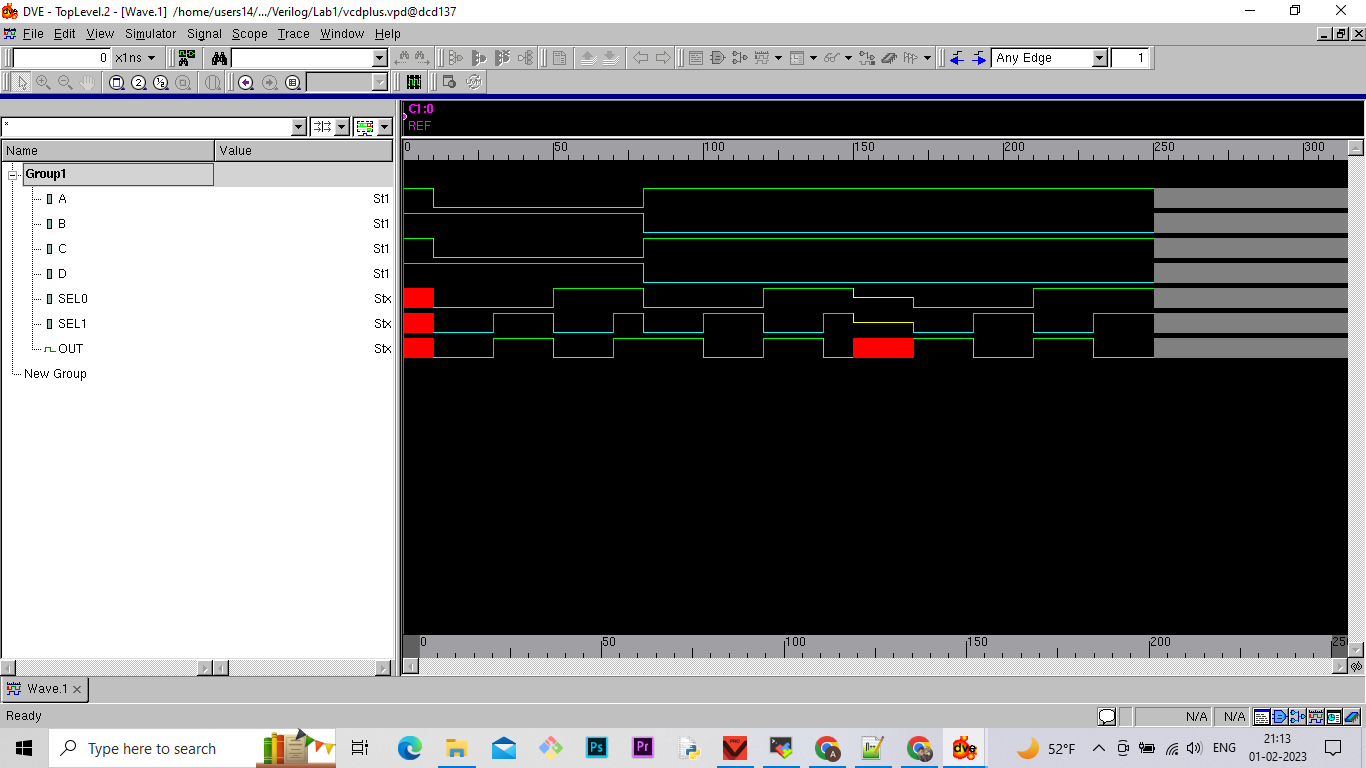


**e. Part 5: Creating Log File**

After running the simulation I created the log file using the “simv -l Lab1.log” command.

**f. Part 6: Seeing the waveform.**

After creating the log file I opened the DVE using “dve -full64 &” command to see the waveform.



**Conclusion :**

In this lab I learned how to design a 4:1 multiplexer in verilog with the help of 4:1 mux circuit. This lab taught me how to run the code and check errors. Moreover, I learned how the simulation works and How to see it in waveform. This lab was very helpful for me to understand the basic knowledge of Verilog language.

**Extra Credit :**

If we consider X and Z as possible inputs, there will be 64 test cases.

I hereby attest that this lab report is entirely my own work. I have not copied either code or text from anyone, nor have I allowed or will I allow anyone to copy my work.

Name (printed) Avinash Damse

Name (signed) Date : 09-02-2023

