California State University, Northridge

Department of Electrical & Computer Engineering

ECE 526L

Lab Report 1

4:1 Multiplexer

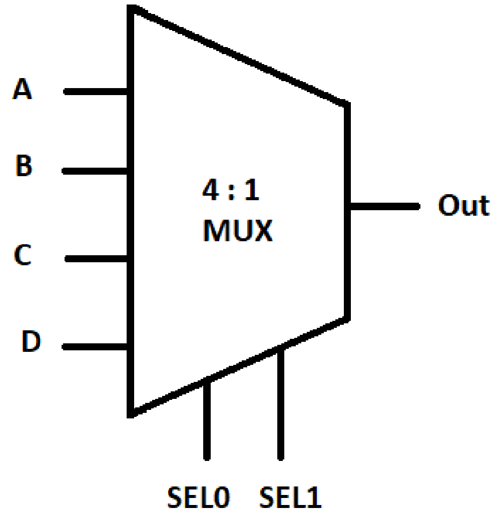
By

Avinash Damse

CSUN ID- 203131064

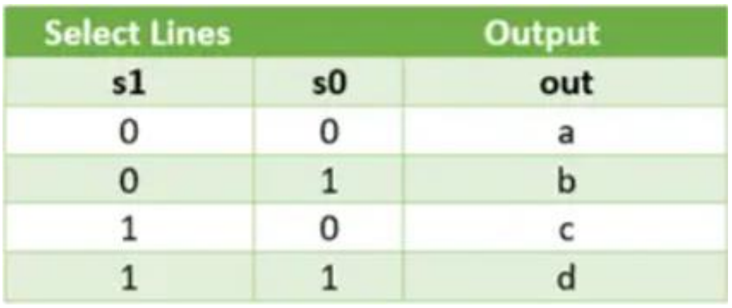
1: Introduction

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 2^n 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.

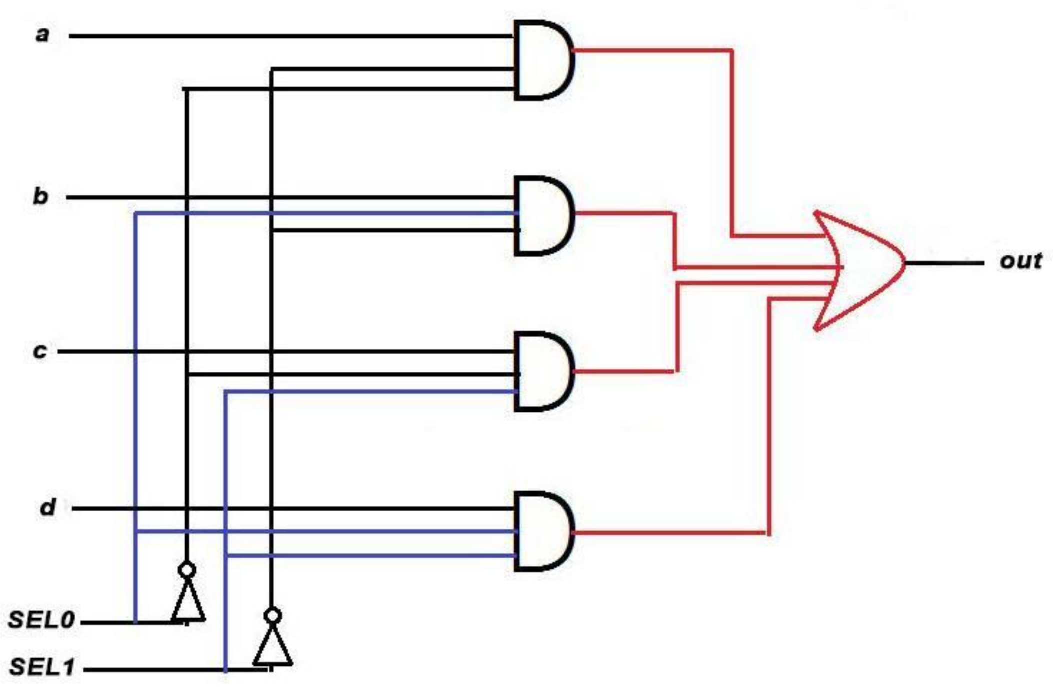


Select Lines		Output
s1	s0	out
0	0	a
0	1	b
1	0	c
1	1	d

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

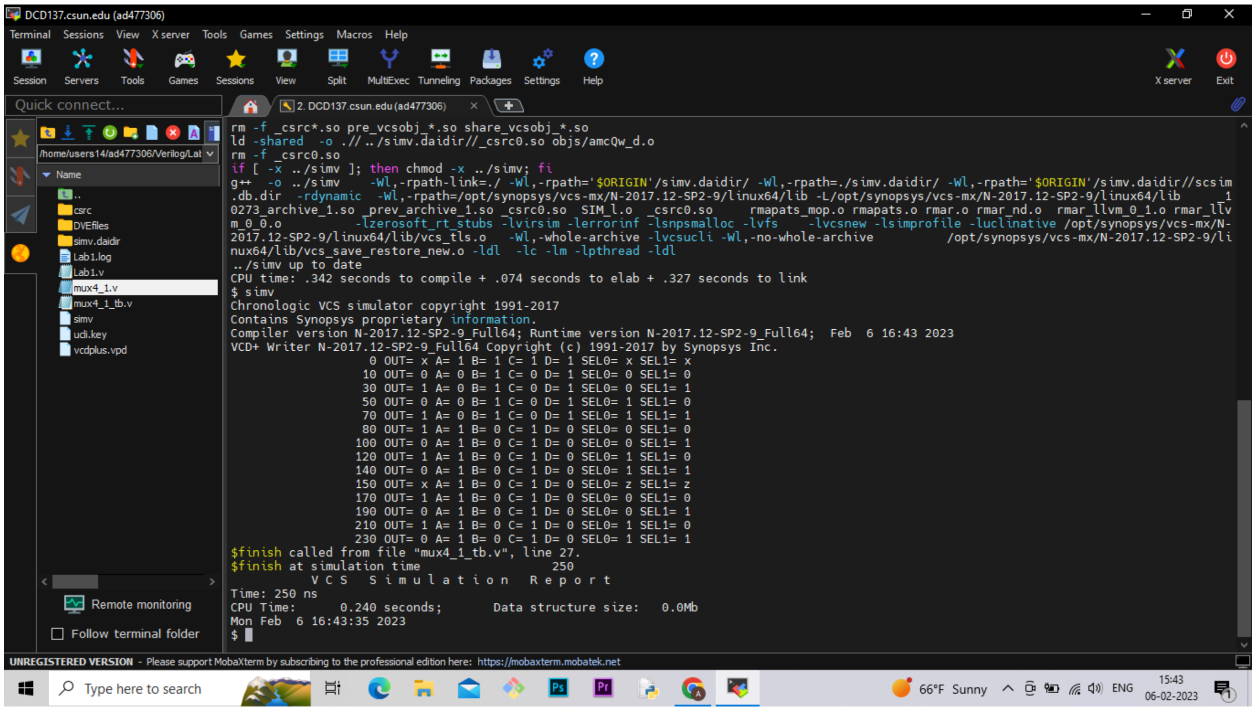
I have written the testbench for the mux4_1 module code after creating my module. We require testbench just to make sure that the module we have created is working properly. In this lab, I have written the testbench and saved the file as “mux4_1_tb.v”.

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.


DCD137.csun.edu (ad477306)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
2. DCD137.csun.edu (ad477306)
/home/users14/ad477306/Verilog/Lat
Name
..
csrc
DVEfiles
simv.daidir
Lab1.log
Lab1.v
mux4_1.v
mux4_1_tb.v
simv
ucli.key
vcdplus.vpd
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
Type here to search
66°F Sunny
ENG
15:43
06-02-2023

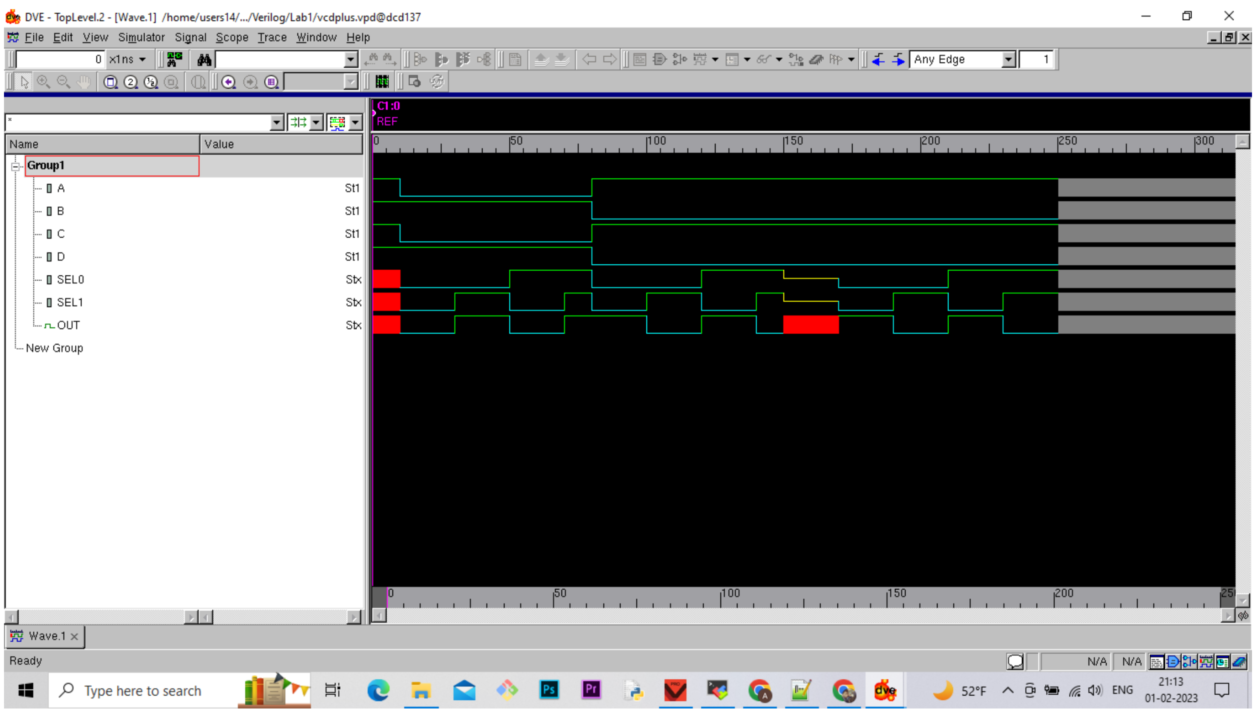
```
rm -f _csrc*.so pre_vcsobj_*.so share_vcsobj_*.so
ld -shared  -o .//../simv.daidir//_csrc0.so objs/amcQw_d.o
rm -f _csrc0.so
if [ -x ../simv ]; then chmod -x ../simv; fi
g++  -o ../simv    -Wl,-rpath-link=./ -Wl,-rpath='$ORIGIN'/simv.daidir/ -Wl,-rpath=./simv.daidir/ -Wl,-rpath='$ORIGIN'/simv.daidir//scsim
.db.dir  -rdynamic  -Wl,-rpath=/opt/synopsys/vcs-mx/N-2017.12-SP2-9/linux64/lib -L/opt/synopsys/vcs-mx/N-2017.12-SP2-9/linux64/lib     _1
0273_archive_1.so _prev_archive_1.so _csrc0.so  SIM_l.o  _csrc0.so     rmapats_mop.o rmapats.o rmar.o rmar_nd.o  rmar_llvm_0_1.o rmar_llv
m_0_0.o          -lzerosoft_rt_stubs -lvirsim -lerrorinf -lsnpsmalloc -lvfs    -lvcsnew -lsimprofile -luclinative /opt/synopsys/vcs-mx/N-
2017.12-SP2-9/linux64/lib/vcs_tls.o   -Wl,-whole-archive -lvcsucli -Wl,-no-whole-archive         /opt/synopsys/vcs-mx/N-2017.12-SP2-9/li
nux64/lib/vcs_save_restore_new.o -ldl  -lc -lm -lpthread -ldl
../simv up to date
CPU time: .342 seconds to compile + .074 seconds to elab + .327 seconds to link
$ simv
Chronologic VCS simulator copyright 1991-2017
Contains Synopsys proprietary information.
Compiler version N-2017.12-SP2-9_Full64; Runtime version N-2017.12-SP2-9_Full64;  Feb  6 16:43 2023
VCD+ Writer N-2017.12-SP2-9_Full64 Copyright (c) 1991-2017 by Synopsys Inc.
                   0 OUT= x A= 1 B= 1 C= 1 D= 1 SEL0= x SEL1= x
                  10 OUT= 0 A= 0 B= 1 C= 0 D= 1 SEL0= 0 SEL1= 0
                  30 OUT= 1 A= 0 B= 1 C= 0 D= 1 SEL0= 0 SEL1= 1
                  50 OUT= 0 A= 0 B= 1 C= 0 D= 1 SEL0= 1 SEL1= 0
                  70 OUT= 1 A= 0 B= 1 C= 0 D= 1 SEL0= 1 SEL1= 1
                  80 OUT= 1 A= 1 B= 0 C= 1 D= 0 SEL0= 0 SEL1= 0
                 100 OUT= 0 A= 1 B= 0 C= 1 D= 0 SEL0= 0 SEL1= 1
                 120 OUT= 1 A= 1 B= 0 C= 1 D= 0 SEL0= 1 SEL1= 0
                 140 OUT= 0 A= 1 B= 0 C= 1 D= 0 SEL0= 1 SEL1= 1
                 150 OUT= x A= 1 B= 0 C= 1 D= 0 SEL0= z SEL1= z
                 170 OUT= 1 A= 1 B= 0 C= 1 D= 0 SEL0= 0 SEL1= 0
                 190 OUT= 0 A= 1 B= 0 C= 1 D= 0 SEL0= 0 SEL1= 1
                 210 OUT= 1 A= 1 B= 0 C= 1 D= 0 SEL0= 1 SEL1= 0
                 230 OUT= 0 A= 1 B= 0 C= 1 D= 0 SEL0= 1 SEL1= 1
$finish called from file "mux4_1_tb.v", line 27.
$finish at simulation time                  250
           V C S   S i m u l a t i o n   R e p o r t
Time: 250 ns
CPU Time:      0.240 seconds;       Data structure size:   0.0Mb
Mon Feb  6 16:43:35 2023
$
```

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) [illegible] Date : 09-02-2023