CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

Department of Electrical and Computer Engineering

ECE 526L

LAB – 7 :

SCALABLE MULTIPLEXER

Written By : G.TARUN KUMAR

Professor : Orod Haghighiara

INTRODUCTION:

In this lab we will create a Verilog model of a scalable multiplexer using Synopsys VCS in Linux OS environment and different terminal commands

METHODOLOGY:

In this lab we will create a Verilog model of a scalable multiplexer using the given module and steps in assignment we should write a design code for the scalable multiplexer and save it as “ mux.v ” in a new folder named “lab7”

The module name and file name should be consistent

mux - Notepad

File Edit Format View Help

```
`timescale 1ns / 1ns
module mux #(parameter SIZE=1) (input [SIZE-1:0] A, B, input SEL, output [SIZE-1:0] OUT);
reg[SIZE-1:0] OUT;
integer i;
  always @(*) begin
    if (SEL == 1'b0) begin
      OUT = A;
    end else if (SEL == 1'b1) begin
      OUT = B;
    end else begin
      for (i = 0; i < SIZE; i=i+1) begin
        if (A[i] == B[i]) begin
          OUT[i] = 1'bx;
        end else if (SEL == 1'b0) begin
          OUT[i] = A[i];
        end else begin
          OUT[i] = B[i];
        end
      end
    end
  end
endmodule
```

This is the fig of **Design code**

Now we should write the test bench code for the design code for that we should create four instances of scalable multiplexer by using 1,4,5,6 for different widths and consider the points given in assignment and that should be saved as **“ tb_mux.v** ” shown in below figure.

tb_mux - Notepad

File Edit Format View Help

```
`timescale 1ns / 1ns
module tb_mux;
  reg [0:0] A_1, B_1;
  reg [3:0] A_4, B_4;
  reg [4:0] A_5, B_5;
  reg [5:0] A_6, B_6;
  reg SEL;
  wire [0:0] OUT_1;
  wire [3:0] OUT_4;
  wire [4:0] OUT_5;
  wire [5:0] OUT_6;
  // Instantiate the scalable multiplexers
  mux # (1) mux_1 (A_1, B_1, SEL, OUT_1);
  mux # (4) mux_4 (A_4, B_4, SEL, OUT_4);
  mux # (5) mux_5 (A_5, B_5, SEL, OUT_5);
  mux # (6) mux_6 (A_6, B_6, SEL, OUT_6);

  initial begin
    $vcdpluson;
    // Test case 1: 1-bit wide multiplexer
    A_1 = 1'b0;
    B_1 = 1'b1;
    SEL = 1'b0;
    #10;
    $display(" 1-bit wide MUX - SEL = 0: OUT = %b", OUT_1);
    // Test case 2: 4-bit wide multiplexer
    A_4 = 4'b0000;
    B_4 = 4'b1111;
    SEL = 1'b1;
    #10;
    $display (" 4-bit wide MUX - SEL = 1: OUT = %b", OUT_4);

    // Test case 3: 5-bit wide multiplexer
    A_5 = 5'b10111;
    B_5 = 5'b10111;
    SEL = 1'bx;
    #10;
    $display(" 5-bit wide MUX - SEL = x: OUT = %b", OUT_5);

    // Test case 4: 6-bit wide multiplexer
    A_6 = 6'b101011;
    B_6 = 6'b010100;
    SEL = 1'bx;
    #10;
    $display(" 6-bit wide MUX - SEL = x: OUT = %b", OUT_6);
    $finish;
  end
endmodule
```

Now compile the two files by using the command line as follows

“vcs -debug_access+all mux.v tb_mux.v”

To start the simulator type simv -l lab1.log ,by using this command we will get logfile and simv outcome.

lab7 - Notepad

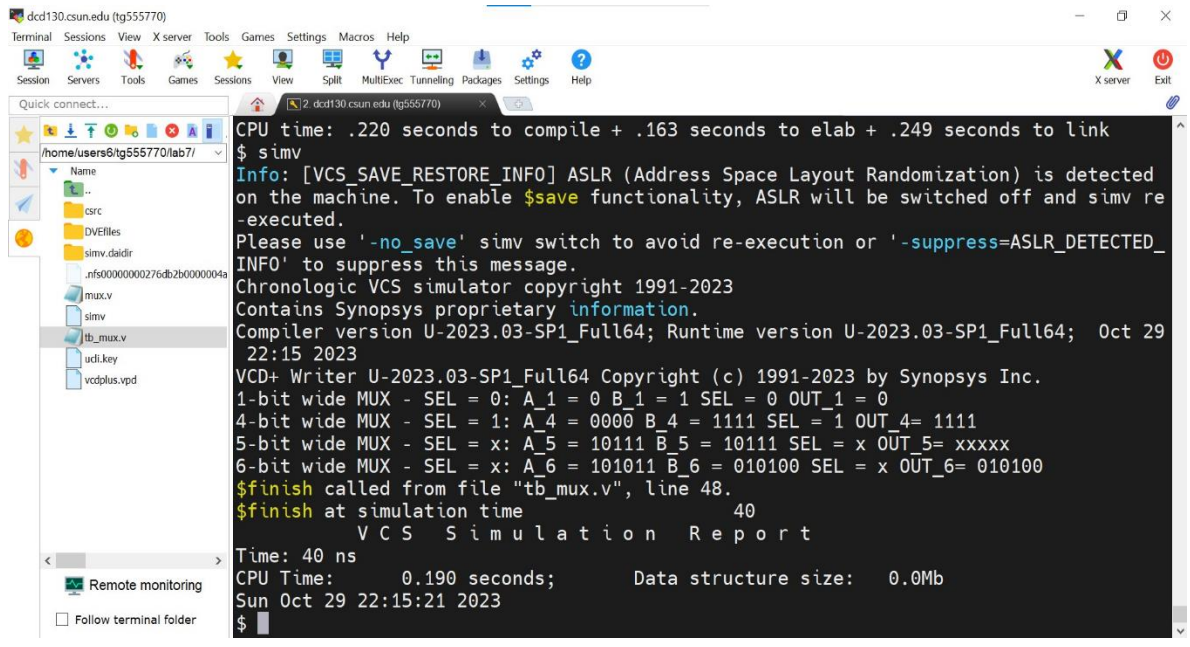
File Edit Format View Help

```
Command: /home/users6/tg555770/lab7/./simv -l lab7.log
Chronologic VCS simulator copyright 1991-2023
Contains Synopsys proprietary information.
Compiler version U-2023.03-SP1_Full64; Runtime version U-2023.03-SP1_Full64;  Oct 29 22:16 2023
VCD+ Writer U-2023.03-SP1_Full64 Copyright (c) 1991-2023 by Synopsys Inc.
1-bit wide MUX - SEL = 0: A_1 = 0 B_1 = 1 SEL = 0 OUT_1 = 0
4-bit wide MUX - SEL = 1: A_4 = 0000 B_4 = 1111 SEL = 1 OUT_4= 1111
5-bit wide MUX - SEL = x: A_5 = 10111 B_5 = 10111 SEL = x OUT_5= xxxxx
6-bit wide MUX - SEL = x: A_6 = 101011 B_6 = 010100 SEL = x OUT_6= 010100
$finish called from file "tb_mux.v", line 48.
$finish at simulation time                   40
          V C S   S i m u l a t i o n   R e p o r t
Time: 40 ns
CPU Time:      0.190 seconds;       Data structure size:   0.0Mb
Sun Oct 29 22:16:20 2023
```

This the **Log File** which I created as shown in the above figure.

If we don’t see any error messages if everything is typed correctly then output should look like this as shown in below figure.

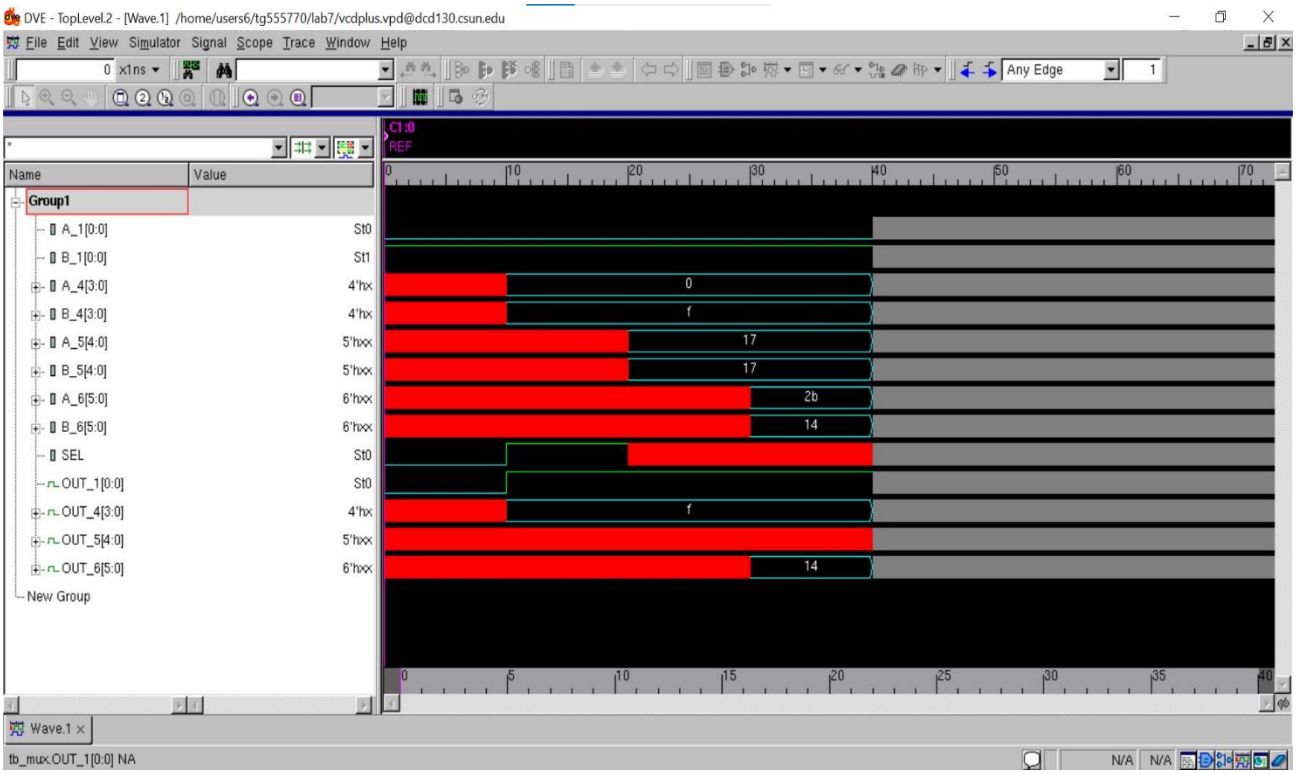
SIMV:



To see the waveforms we should use the software named “DVE”

To see the wave form first type “dve” command line then we will get the blank simulation window then go to file and open database then choose vcdplus.vpd and open the file.

Then finally select all the signals with the mouse ,then right click and select to “ add to wave->new wave view”. Then we will get the waveform as shown in below figure.



Analysis of Result:

scalable multiplexer refers to a digital circuit that can select one of multiple input signals and route it to the output based on a control

signal. The result for a scalable multiplexer in Verilog is typically defined using a case statement or an if-else construct in given lab we taken 4 instances in different cases depend upon on the inputs value we provide and the sel value the output value varies. By increasing the number of input signals and utilizing the proper control signal width, we may scale this idea to construct larger multiplexers, such as 4:1, 8:1, or even larger multiplexers.

Conclusions: In this lab we simulated scalable multiplexer using synopsys Vcs in Linux OS environment and got the required waveform.

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 I will anyone to copy my work.

Name(printed) Gunupudi Tarun Kumar

Name(signed) G.Tarun Date 10/29/2023