**LAB REPORT**

**CSE4010 Computer Architecture**

**Instructor: Lawrence Orijuela**

Name: Daniel Delgado Acosta SCORE: \_\_\_/30

Student ID: 006896598 DUE: 2/14/23

LAB: Lab1

**Report**

**o *What is Verilog and what is it used for?***

Verilog is a computer language used to describe and design circuits and other electronic systems.

**o *What is a module and a testbench?***

A module is a piece of code that does a task such as assigning variables. A test bench is what is used to check and run a module.

**o *A brief but comprehensive description of what you’ve done and what you learned.***

I first created a Verilog file that initialized and assigned 3 different variables. I then created a second Verilog file as a test bench and included the first file to create wires. I ran the code in the terminal and saw the wires over time. The second part this lab was nearly identical to the first part, but I used 4 variable and different negation assignments.

**Source Code for Parts A and B**

**Part A**

*WireTest.v*

//initialize wires A, B, and C

module wireTest (A, B, C);

    //input wire is A

    input A;

    //output wires are B and C

    output B;

    output C;

    //wire B will be the same as wire A

    assign B = A;

    //wire C will be the opposite as wire A

    assign C = !A;

endmodule

*WireTest\_tb.v*

//creates time limit and includes the wireTest.v file

`timescale 1 ns / 1 ns

`include "wireTest.v"

//creates test bench

module wireTest\_tb;

//denotes wires A, B, and C

reg A;

wire B;

wire C;

//creates instance of wireTest

wireTest uut(A, B, C);

initial begin

    //creates a separate .vvp file and .vcd file for outputs

    $dumpfile("wireTest\_tb.vcd");

    $dumpvars(0, wireTest\_tb);

    A = 0; //sets A to 0

    #20 //wait 20 nanoseconds

    A = 1; //sets A to 1

    #20 //wait 20 nanoseconds

    A = 0; //sets A to 0

    #20 //wait 20 nanoseconds

    A = 1; //sets A to 1

    #20 //wait 20 nanoseconds

    //displays message to terminal

    $display("Wire tests complete!");

end

endmodule

**Part B**

*wireTest2.v*

//initialize wires W, X, Y, and Z

module wireTest2 (W, X, Y, Z);

    //input wires are W and X

    input W;

    input X;

    //output wires are Y and Z

    output Y;

    output Z;

    //wire Y will be the opposite as wire X

    assign Y = !X;

    //wire Z will be the opposite as wire Y

    assign Z = !Y;

endmodule

*wireTest2\_tb.v*

//creates time limit and includes the wireTest2.v file

`timescale 1 ns / 1 ns

`include "wireTest2.v"

//creates test bench

module wireTest2\_tb;

//denotes wires W, X, Y, and Z

reg W;

reg X;

wire Y;

wire Z;

//creates instance of wireTest

wireTest2 uut(W, X, Y, Z);

initial begin

    //creates a seperate .vvp file and .vcd file for outputs

    $dumpfile("wireTest2\_tb.vcd");

    $dumpvars(0, wireTest2\_tb);

    W = 0; //sets W to 0

    X = 0; //sets X to 0

    #20 //wait 20 nanoseconds

    W = 1; //sets W to 1

    #20 //wait 20 nanoseconds

    X = 1; //sets X to 1

    #20 //wait 20 nanoseconds

    W = 0; //sets W to 0

    #20 //wait 20 nanoseconds

    X = 0; //sets X to 0

    #20 //wait 20 nanoseconds

    //displays message to terminal

    $display("Wire tests complete!");

end

endmodule

**Screenshots for Parts A and B**

Text

Description automatically generated**Part A**

Graphical user interface, application

Description automatically generated

Text

Description automatically generated**Part B**

Graphical user interface, application

Description automatically generated