Lab Report
CSE4010 Computer Architecture

Name: Zachary A. Hampton **Score:** /10

Student ID: 008339494 **Due:** 02-09-2025

Lab: Lab 1 - Introduction to Verilog

Report

- What is Verilog and what is it used for?
  - Verilog is an IEEE-standardized Hardware Description Language (HDL) used for designing and modeling digital electronic systems, particularly in FPGA and ASIC development.
- What is a module and a testbench?
  - A module is the fundamental building block in Verilog; it defines a circuit's behavior, encapsulating functionality with specified inputs, outputs, internal logic, and potentially other modules.
  - A testbench is a Verilog module used for simulation, providing stimulus to a design under test (DUT) and verifying its behavior before actual hardware implementation.
- A brief but comprehensive description of what you've done and what you learned.
  - For Part A, I set up the 'wireTest' module and its corresponding testbench using the provided sample code. I then compiled and executed the simulation, generating a '.vcd' waveform file for analysis. Using GTKWave, I loaded the '.vcd' file and enabled the necessary variables to visualize their respective waveforms. For Part B, I wrote a similar module with an additional input and slightly different logic operations, resulting in a more complex waveform output.

Source Code for Part A

wireTest.v

Listing 1: wireTest.v

```
module wireTest (A,B,C);

    input A;
    output B;
    output C;

    assign B = A;
    assign C = !A;

endmodule
```

wireTest_tb.v

Listing 2: wireTest$_tb.v$

```
`timescale 1ns / 1ns
`include "wireTest.v"

module wireTest_tb;

reg A;
wire B;
wire C;

wireTest uut (A, B, C);

initial begin

   $dumpfile("wireTest_tb.vcd");
   $dumpvars(0, wireTest_tb);

   A = 0;
   #20

   A = 1;
   #20

   A = 0;
   #20

   A = 1;
   #20

   $display("Wire test complete!");

end

endmodule
```

Source Code for Part B

wireTest2.v

Listing 3: wireTest2.v

```
/*
Instructions from Dr. Ahmed:
Accepts two inputs W and X
Gives two outputs Y and Z
Set Y equal to the negation of X
Set Z equal to the negation of Y
*/

// Define the wireTest2 module
module wireTest2 (W,X,Y,Z);

    // Declare the inputs and outputs
    input W;
    input X;
    output Y;
    output Z;

    // Assign Y to the negation of X
    assign Y = !X;
    // Assign Z to the negation of Y
    assign Z = !Y;

// End the wireTest2 module
endmodule
```

wireTest2_tb.v

Listing 4: wireTest2$_tb.v$

```
/*
Instructions from Dr. Ahmed:
This file is used to test the wireTest2 module.
The testbench will set W and X to 0 and 1 in various combinations
and check that Y and Z are set correctly.
The testbench will output "Wire test complete!" when the test is
   done.
*/

// Include the wireTest2 module and timescale of 1ns
`timescale 1ns / 1ns
`include "wireTest2.v"

// Define the wireTest2_tb module
module wireTest2_tb;

```

```
// Declare the inputs and outputs
reg W;
reg X;
wire Y;
wire Z;

// Instantiate the wireTest2 module
wireTest2 uut (W, X, Y, Z);

// Define the initial block
initial begin

    // Dump the waveforms to wireTest2_tb.vcd
    $dumpfile("wireTest2_tb.vcd");
    $dumpvars(0, wireTest2_tb);

    // Set W and X to 0
    W = 0;
    X = 0;
    #20

    // Set W to 1
    W = 1;
    #20

    // Set X to 1
    X = 1;
    #20

    // Set W to 0
    W = 0;
    #20

    // Set X to 0
    X = 0;
    #20

    // Print "Wire test complete!" to the console
    $display("Wire test complete!");

// End the initial block
end

// End the wireTest2_tb module
endmodule
```

Screenshots for Parts A and B

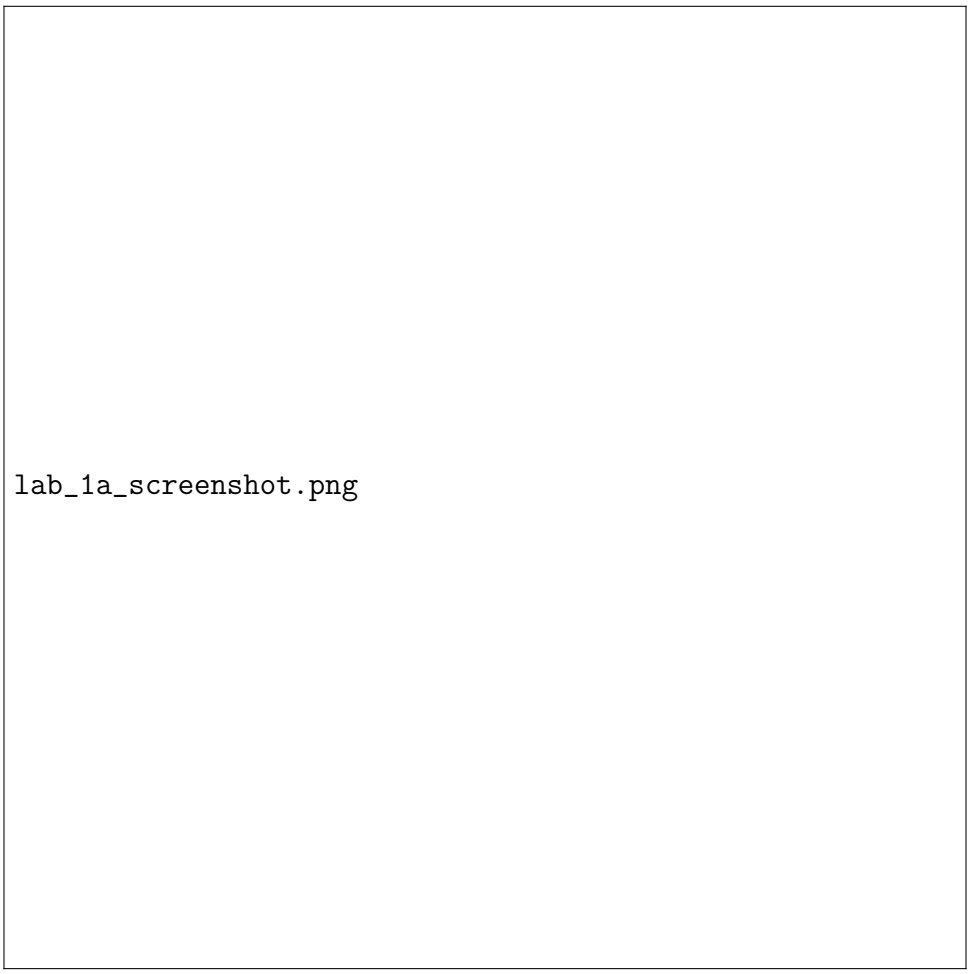

Figure 1: Screenshot for Part A

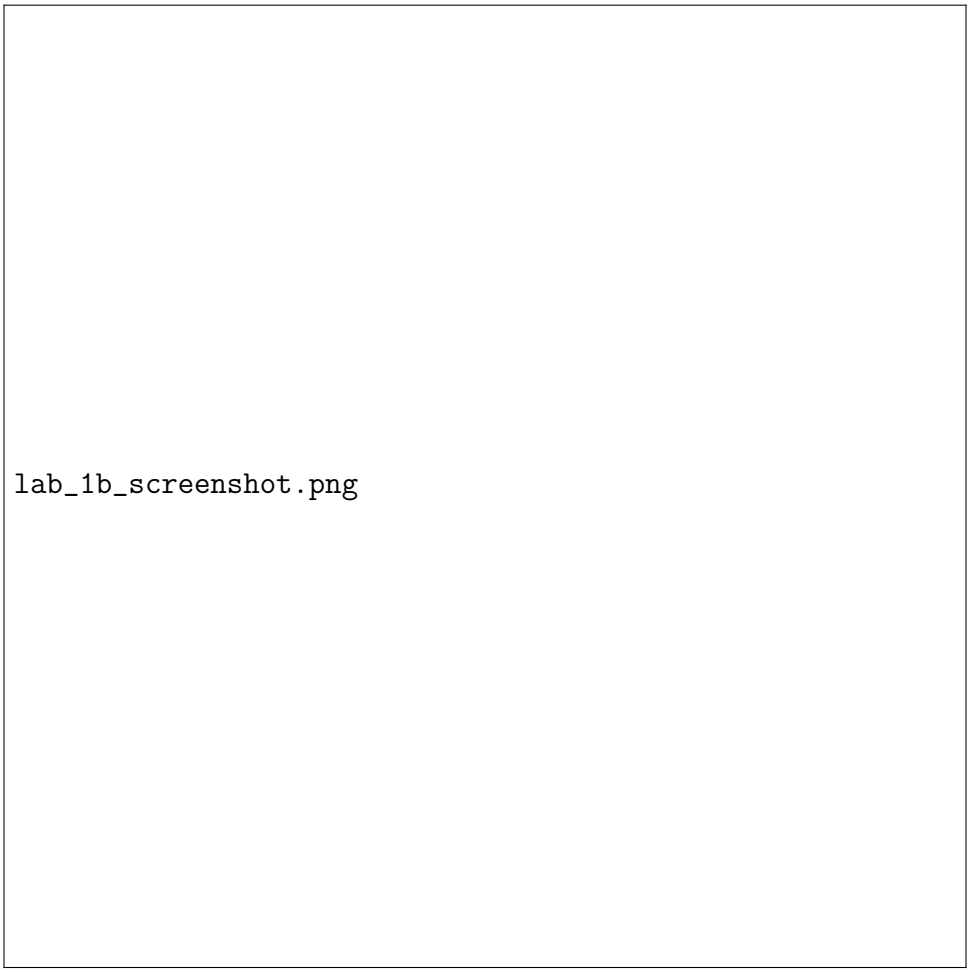

Figure 2: Screenshot for Part B