Diane Shan

862148900

EE120A Section 23

Lab 1 – Intro to EDA Playground

**Overview**

In this lab, I implemented a design using Verilog and also created a Verilog testbench which tested the functionality of the design and verified that the design produced the results we were expecting. In doing these things, I familiarized myself with Xilinx software, and learned about the FPGA board and that Verilog is an HDL. I successfully coded and, nand, or, and nor gates with the Xilinx software. They were represented in a timing diagram and programmed onto FPGA board. I wrote two testbenches by myself and after running them they turned out to be successful further confirming the functionality of the design. Xilinx is the learning environment we will be using in this course and this lab helped us get practice with using it.

**New Concepts**

FPGA – Field Programmable Gate Array, HDL/Software programmable, medium-fast speed, it is an array of LUTs (Logic Elements), often uses off-chip memory and an integrated microprocessor, can also be used as a testbench

VHDL – very popular in government/education/Aerospace and European companies

Verilog – mostly used in semiconductor industry, most widely known and used

AND/NAND/OR/NOR gates – basic logic gates

**Analysis**

Procedure:

1. Set up the EDA Playground UI
2. Copy the structural implementation of the circuit to the design window of the EDA Playground UI
3. Save and look over code for understanding
4. Copy the given testbench code into the testbench window of the EDA Playground UI
5. Click “Run” to run the testbench
6. When the EPWave window pops up, click on “Get Signal” and add the signals to the waveform
7. Then add two more tests to the testbench and generate the final waveform

**Records**

*Design:*

`timescale 1ns / 1ps

module lab1\_structural (

// Ports I/O

input wire Bot,

input wire Top,

output wire A,

output wire B,

output wire C,

output wire D

);

// Place gates and connect I/O

// Format: gate\_type arbitrary\_label (output, input1, input2)

and gate1 (A, Top, Bot);

nand gate2 (B, Top, Bot) ;

or gate3 (C, Top, Bot) ;

nor gate4 (D, Top, Bot) ;

endmodule

*Testbench*:

`timescale 1ns / 1ps

module lab1\_structural\_tb();

// Inputs

reg bot\_reg = 1'b0;

reg top\_reg = 1'b0;

// Outputs

wire a;

wire b;

wire c;

wire d;

// Instantiate your circuit (lab1\_structural)

lab1\_structural UUT (

// Connect your testbench to your circuit

.A(a),

.B(b),

.C(c),

.D(d),

.Bot(bot\_reg),

.Top(top\_reg)

);

// Assign inputs to your circuit

// Test for expected output

initial begin

// The following line is EDA Playground specific.

// Make sure to add it to all your future testbenches

$dumpfile("dump.vcd"); $dumpvars;

// Testcase #1

$display("Testcase #1");

// Assign inputs

bot\_reg = 1'b1;

top\_reg = 1'b1;

#40; // Wait 40 ns

// Check for expected output

if ( {a,b,c,d} != 4'b1010 )

$display ("\t Result is wrong %b ", {a,b,c,d});

else

$display ("\t Testcase #1 successful");

// Testcase #2

$display("Testcase #2");

// Assign inputs

bot\_reg = 1'b0;

top\_reg = 1'b0;

#40;

// Check for expected output

if ( {a,b,c,d} != 4'b0101 )

$display ("\t Result is wrong %b ", {a,b,c,d});

else

$display ("\t Testcase #2 successful");

// Add testcase #3 and #4 below

//Testcase #3

$display("Testcase #3");

//Assign inputs

bot\_reg = 1'b1;

bot\_reg = 1'b0;

#40;

//Check for expected output

if ( {a,b,c,d} != 4'b0101 )

$display ("\t Result is wrong %b ", {a,b,c,d});

else

$display ("\t Testcase #3 successful");

//Testcase #4

$display("Testcase #4");

//Assign inputs

bot\_reg = 1'b0;

bot\_reg = 1'b1;

#40;

//Check for expected output

if ( {a,b,c,d} != 4'b0110 )

$display ("\t Result is wrong %b ", {a,b,c,d});

else

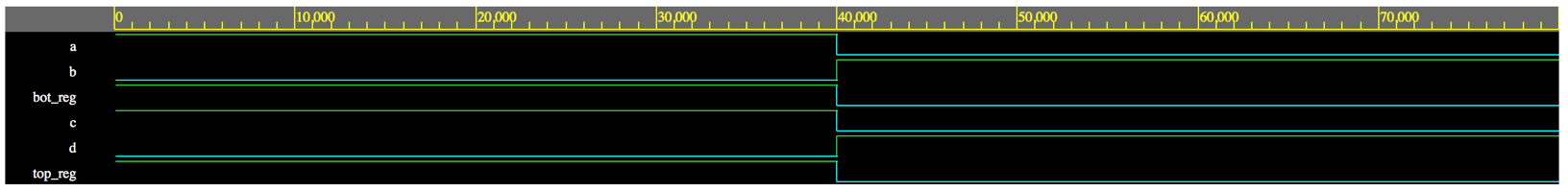
$display ("\t Testcase #4 successful");

end

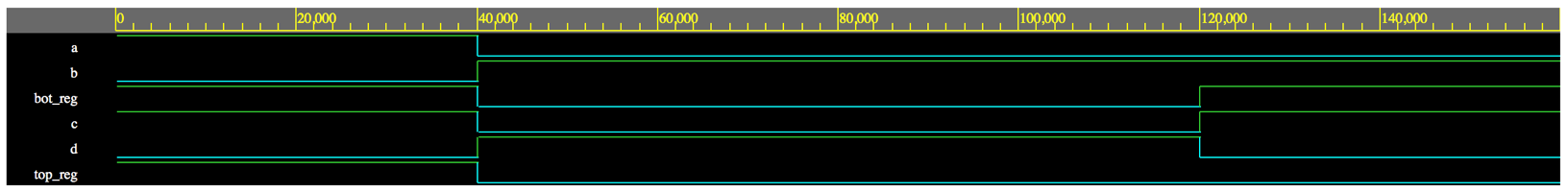
endmodule

*Simulation Results (waveform):*

First time:



Final Waveform (after adding testbench 3 and 4):



**Discussion**

After running the first simulation with only testbench 1 and testbench 2, the system worked. It successfully passed the first two testbenches. The results seen in the waveform are the expected results. The circuit produces the correct results. Then after adding the next 2 testbenches, the system passes those as well. This means the system works according to the provided specifications. There were no problems or technical issues encountered and the system is good as is since it passes all the tests with flying colors. There are not really any improvements to be made to this system.

**Conclusion**

The purpose of the lab is to implement a design using the Xilinx environment and test it with testbenches. It was to get us familiar with the software and learn how to use testbenches properly. Also, it helped us with looking at waveforms and how to use Verilog. The results were that the design passed all the testbenches and the simulated waveforms were correct.

**Questions**

Does the simulated waveform look correct?

The simulated waveform is correct. It produces the correct output (a, b, c, d) and therefore passes the testbench. This means the design functions correctly and as expected.