Cover Page

Name: Ethan Lingad

Student ID: 862268541

Course Title: CS120A

Lab Section: 22

Lab Title: Lab 01. Introduction to EDA Playground

Overview:

In this lab, I created an EDA account in which I then created a new playground to code in Verilog. Then I copy and pasted the code given in the lab exercise in which the goal is to create a circuit that has two inputs and four outputs. I had to check if the code given in the lab exercise produced the correct output and to look at the output to see if the circuits received the right output. I then was prompted to fill in the rest of the combinations of inputs which is when bot_reg and top_reg had different input values in which they also passed the test.

Analysis:

1. copy code from lab and paste it
2. save and run the code.
3. verify if the input gives the correct outputs for each output.
4. If not, make adjustments to code so that it works
5. If so, add the rest of the test cases, 3 and 4.
6. For the test cases, add the rest of the combinations for the input which is when both inputs are 0 and 1, switching for both test cases.
7. Run the code and verify the result is still correct

Records:

testbench.sv

```
`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

bot_reg = 1'b1;
top_reg = 1'b0;
#40
if ( {a,b,c,d} != 4'b0110)
  $display("\t Result is wrong %b ");
else
  $display("\t Testcase #3 successful");

bot_reg = 1'b0;
top_reg = 1'b1;
#40
  if({a,b,c,d} != 4'b0110)
    $display("\t Result is wrong %b ");
  else
    $display("\t Testcase #4 successful");

end
endmodule
```

design.sv

```
`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
```

Discussion:

The system does work since it produces the expected output when given certain inputs.

From the code given:

bot_reg	top_reg	expected output {a,b,c,d}	a	b	c	d	Passed?
1	1	{1,0,1,0}	1	0	1	0	Yes
0	0	{0,1,0,1}	0	1	0	1	Yes

From the code I added:

bot_reg	top_reg	expected output {a,b,c,d}	a	b	c	d	Passed?
1	0	{0,1,1,0}	0	1	1	0	Yes
0	1	{0,1,1,0}	0	1	1	0	Yes

I can conclude that the code given gave the correct outputs by verifying the outputs after running the code.

Conclusion:

The purpose of this lab was to get a little better understanding on the syntax of Verliog and how to create logic gates. It also taught me how to understand reading the output signals after running

the Verilog code. Overall, it helped get a better general understanding on how Verilog can be used to represent logic gates and how to connect input and outputs.

Question:

Yes, the code given from the lab does produce the correct outputs. I came to the conclusion because looking at the signals after running the code showed that each output had the correct value when given a certain input signal.