Ivan Neto

862228249

Logic Design | CS120A | Section 021

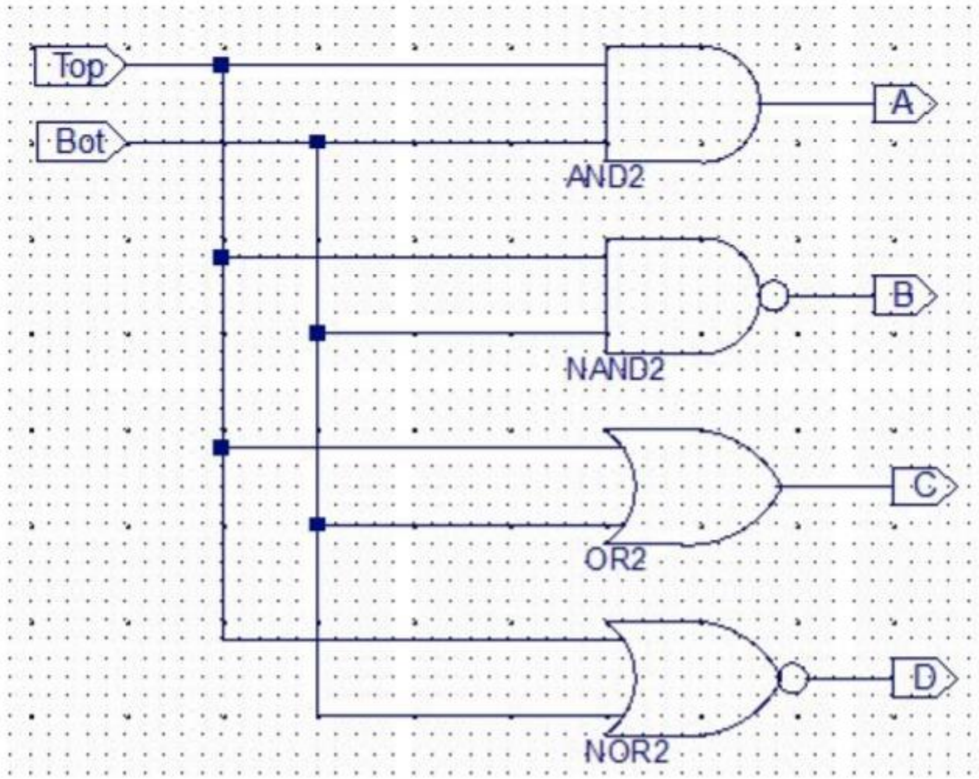
Lab 1 – Intro to EDA Playground

Shreya Chitturi

Overview

Brief Overview

Given Logic Circuit Design



- We implemented the given algorithm to test the logic circuit design given to us. Using Verilog and EDA Playground, we implemented test cases using a testbench file to determine whether our implementation of a given circuit design was correct. Further, we analyzed the EPWave output to determine whether the output of our logic circuit design implementation was correct.

Summary of Results

- The test cases we implemented to test the implementation of the given logic circuit design were all successful. We tested with binary inputs b01 and b10, and asserted that both of these inputs would output b0110, given the logic circuit design.

Conclusion From the Results

- We conclude that our implementation of the logic circuit design was correct, and the logic circuit has been successfully implemented using Verilog.

New Concepts

Verilog, Development, and Lab Exercise

- How to declare variables
- Test benching a main program
- Logic circuit design
    - Logic gates
- Implementation of logic circuit designs with Verilog
- EDA Playground Environment
- Xilinx Environment

Analysis

Procedure

- Implement test case
  - Write Verilog program that will test `design.sv`
- Save file
  - Save the `testbench.sv` file so that the correct program will be executed.
- Run/Test
  - Run, and observe the output to assert the correctness of implementation
- Repeat
  - Implement a new test bench, save the file, run/test program

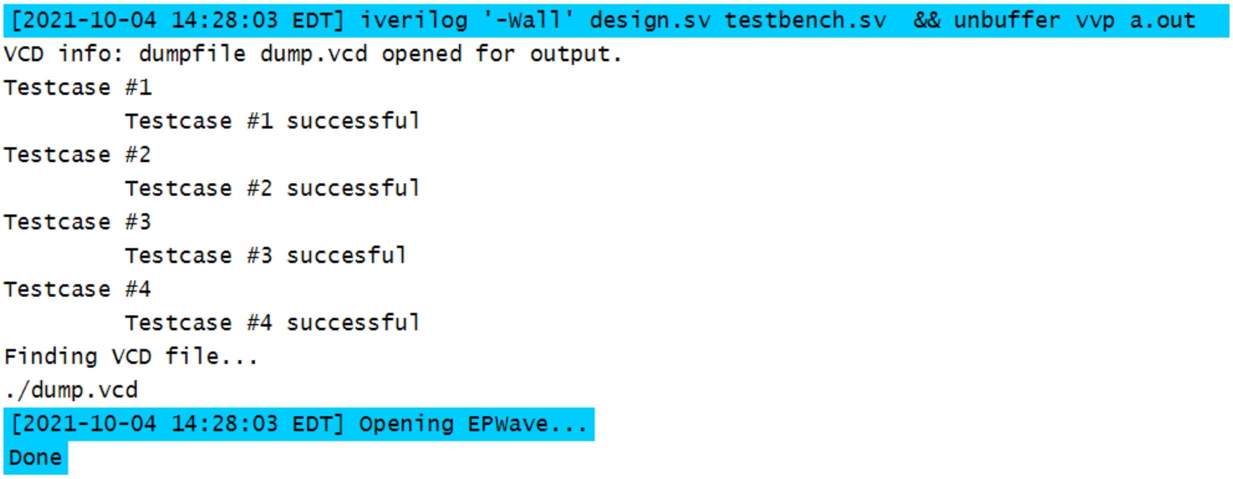
Truth Table for All Possible Inputs

A	B	AND	NAND	OR	NOR
0	0	0	1	0	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	1	0	1	0

Running Test Cases

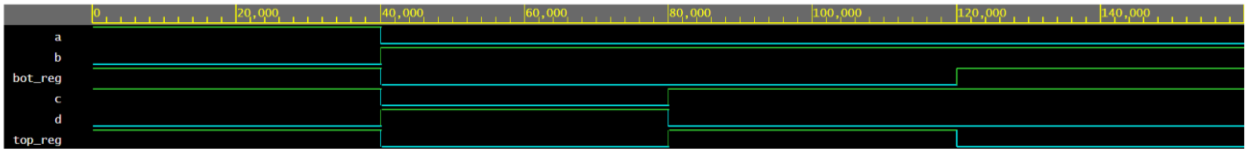
- All test cases were implemented in running the test bench.
  - Cases 1 and 2:
    - top_reg, bot_reg = 0
    - top_reg, bot_reg = 1
  - Cases 3 and 4:
    - top_reg = 1, bot_reg = 0
    - top_reg = 0, bot_reg = 1

- Running the program, we can observe the following console output:



```
[2021-10-04 14:28:03 EDT] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
Testcase #1
         Testcase #1 successful
Testcase #2
         Testcase #2 successful
Testcase #3
         Testcase #3 succesful
Testcase #4
         Testcase #4 successful
Finding VCD file...
./dump.vcd
[2021-10-04 14:28:03 EDT] Opening EPWave...
Done
```

- This output indicates that our main program successfully passed the test benches that we have set up to test the circuit. Test cases 1 and 2 were included by default, while our team implemented test cases 3 and 4.
- We can further observe the EPWave output from the given configuration:

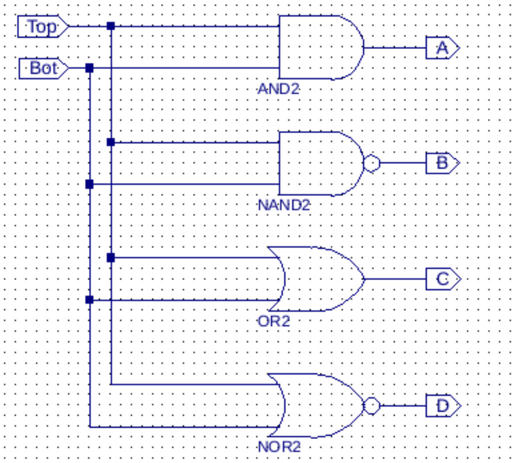


  - A green line for a given variable refers to a 1 signal, while a blue line represents a 0 signal.
  - From this EPWave output we can see that the given inputs match the output variables a, b, c, d.

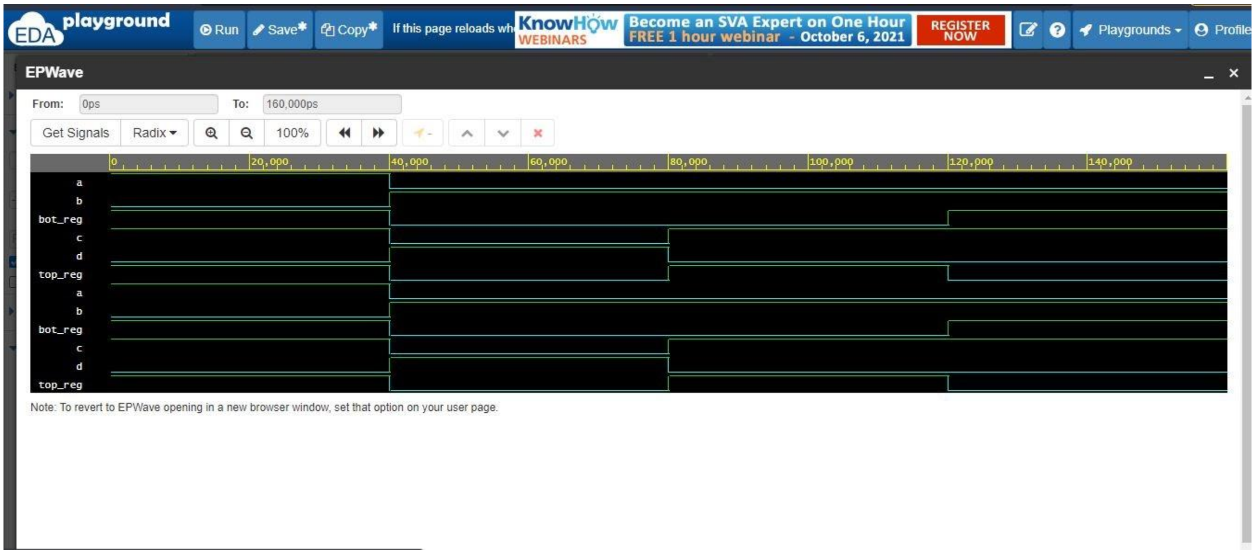
Records

Schematics

- Using Xilinx we created a schematic of the design, though it is the same as was given by the lab instructions.



Simulation



From: 0ps To: 160,000ps

Note: To revert to EPWave opening in a new browser window, set that option on your user page.

- This image portrays the four different possible combinations of bot_reg and top_reg so as to produce different outputs. Each "column" (each distinct block of signals) represents a cycle which illustrates that each output is given the inputs.

UCF

- As we used EDA Playground to complete this lab, we do not have UCF.

Discussion

System Works

- The system works according to the provided specification. For the given input 1’b1 and 1’b1, the output was as expected, 4’b1010. For the input 1’b0 and 1’b0, the output was as expected, 4’b0101. For the other two test cases that we implemented, 1’b1 with 1’b0 and 1’b0 with 1’b1, we got the expected output, 4’b0110 for both.

Problems/Technical Issues

- The only technical issue we had was that we were not all able to install and use Xilinx. Therefore, we resorted to EDA Playground to implement our testbench and circuit. Within EDA Playground, however, we did not have any technical issues that impeded our ability to complete the lab.

Ways to Improve the System

- Some ways in which we could improve our implementation of the testbench would be to:
    - Modularize our development such that we can fully comprehend what each test case is doing.
    - Improve our understanding of Verilog syntax so that we can create more efficient code, as well as code that is readable.
    - Install and learn the full usage of Xilinx for deeper development and testing with Verilog.

Conclusion

Purpose

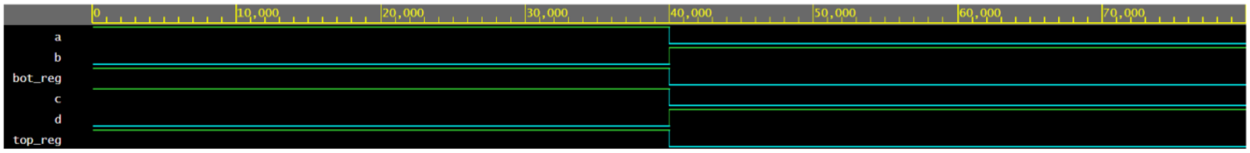
- The purpose of this lab was to get familiar with EDA Playground and the basic Verilog syntax. This lab helped us understand the way that the Verilog Design window and the Verilog Testbench window work together. Once we finished typing in the code that was provided, we learned how to run the Testbench and interpret the EPWave simulation output.

Summary of Results

- We implemented and tested the given test cases (1’b0 with 1’b0 and 1’b1 with 1’b1) and got the expected output. We then implemented two of our own test cases (1’b0 with 1’b1 and 1’b1 with 1’b0). To determine the expected results of our own test cases, we determined what the output would be by manually obtaining the result of each of the gates with our input. For both of our implemented test cases, we got the expected output, 4’b0110.
- This indicates that we successfully implemented the circuit shown in the circuit design image that was given.

Questions

Q1) You should now be able to see the simulation waveform. Does it look correct? i.e. for the given input (bot_reg. top_reg), is the circuit producing the correct output (a, b, c, d)?



- The simulation in question is the one with only test cases 1 and 2 implemented.

- The following is a truth table of the expected output of each gate in the circuit.

A	**B**	**AND**	**NAND**	**OR**	**NOR**
0	**0**	**0**	**1**	**0**	**1**
0	**1**	**0**	**1**	**1**	**0**
1	**0**	**0**	**1**	**1**	**0**
1	**1**	**1**	**0**	**1**	**0**

- We have highlighted the expected outputs for each of the gates in the circuit.
  - A = AND2
  - B = NAND2
  - C = OR2
  - D = NOR2
- In the following table we can summarize the EPWave output in comparison to the expected truth table, for test cases 1 and 2.

Source	top_reg	bot_reg	AND	NAND	OR	NOR
EPWave	0	0	0	1	0	1
Expected	0	0	0	1	0	1
EPWave	1	1	1	0	1	0
Expected	1	1	1	0	1	0

- Since the outputs match for both sources of output (both expected and EPWave results), we know that the implemented circuit both passed the testbench and is correctly implemented in Verilog, using EDA Playground.
- For the given inputs for the bot_reg and top_reg, the circuit is producing the correct output.