California State University, Long Beach
Computer Engineering and Computer Science Department
CECS 361 - Computer Logic Design II
Lab 3 - Circuit Equivalence Checking

Objectives:

- To identify the behavior of an unknown circuit via simulation
- To create an equivalent circuit based on the behavior of another circuit
- To familiarize students with implementing HDL designs for a target technology

Theory:

As digital logic designs become more and more complex, functional verification techniques such as verification via simulation become less feasible. As such we must turn to formal verification techniques, such as equivalence checking, to ensure that our designs work as expected. In this lab, we will attempt to mimic equivalence checking of an unknown module by creating a circuit of your own based on the observed behavior of the original circuit. This lab assignment will help solidify the concepts covered in "Topic 3: Formal Design Verification."

Assignment:

You are to instantiate a module with unknown behavior and you are to create a testbench for it to observe its functionality. Then, you are to create a Boolean equation based on the observed behavior, simplify it, and create a circuit to implement that equation. Finally, you are to instantiate both designs in a top module that will check their equivalence and implement that module on your board.

Steps:

1. Create a project and name it lab3.
2. Select the Nexys A7-100T (xc7a100tcsg324-1) board when prompted to select a target device.
3. Start by adding the Lab3 module that has been provided for you on BeachBoard (Lab3.v). This module contains an obfuscated case statement that assigns a value to output "Out" based on different combinations of its inputs (A, B, C, D, E).
4. Create a simulation to test all 32 possible combinations of inputs and observe the behavior of the design.

5. Create a truth table based on the results of your simulation and formulate a Boolean equation. Simplify the Boolean equation using any technique of your choosing. **Show your work**.
6. Create a new module and name it Lab3_eq. Implement the simplified Boolean formula and create a testbench for this module to ensure that it works as expected.
7. Create another module and name it top. Create an instance of the provided Lab3.v file and name it "original". Create an instance of your module called Lab3_eq and name it "equivalent". Your top module should have the following global input/output ports:
    a. A: 1-bit input that will drive the A port of the "original" and "equivalent" circuits
    b. B: 1-bit input that will drive the B port of the "original" and "equivalent" circuits
    c. C: 1-bit input that will drive the C port of the "original" and "equivalent" circuits
    d. D: 1-bit input that will drive the D port of the "original" and "equivalent" circuits
    e. E: 1-bit input that will drive the E port of the "original" and "equivalent" circuits
    f. Out1: 1-bit output that will be driven by the output of the "original" circuit
    g. Out2: 1-bit output that will be driven by the output of the "equivalent" circuit
    h. eq: 1-bit output that will be the result of an XOR between Out1 and Out2

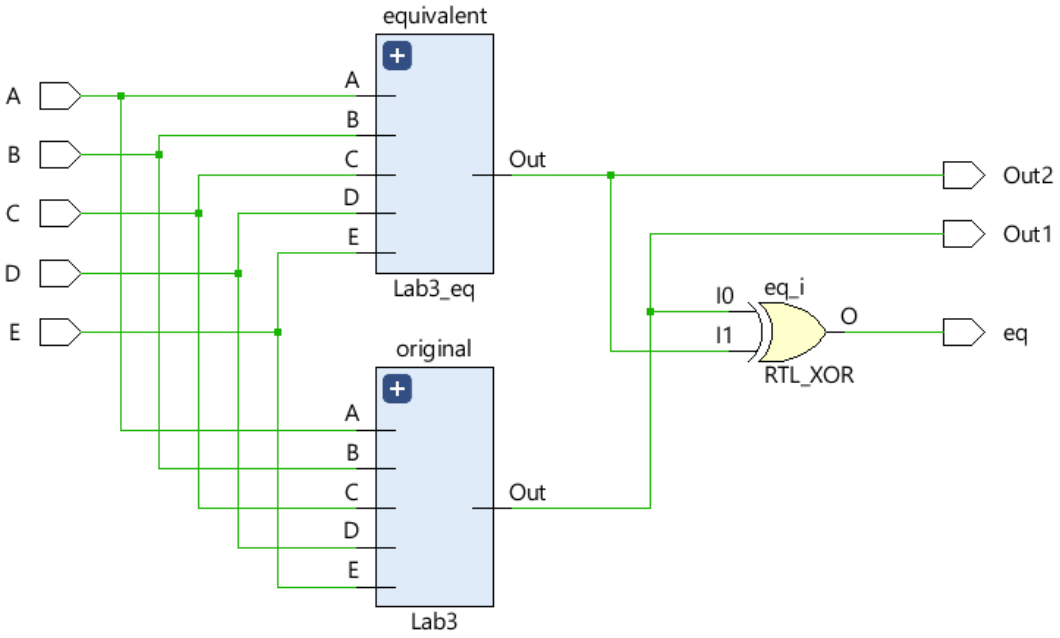

Figure 1. Schematic of the top module

8. Create a testbench for the top module to verify its functionality through simulation. You may get inspired by Lab2 to check if the created top module is satisfiable (SAT) or not satisfiable (UNSAT). If the top module is **not satisfiable**, it means that Lab3 and Lab3_eq modules are **equivalent**.
9. Add the NexysA7-100T.xdc constraints file from Beachboard into your project. The constraints file maps the various inputs and outputs of your top module to the

components for which they are meant to drive. Modify the constraints file to make the following connections on your board: (Make sure that you uncomment the lines for the pins that you are required to use)

a. A through E will be driven by the right most switches (R17 through J15), with A being the most significant switch and E being the least significant.
b. Out1 will be connected to LED0 (H17)
c. Out2 will be connected to LED1(K15)
d. eq will be connected to LED2(J13)

10. Generate the bitstream. Under the Flow Navigator pane, click on **Generate Bitstream** to generate the bitstream that will map your design to the FPGA. You will be prompted to run the synthesis and implementation steps that are required to generate the bitstream. Leave all the prompts with the default selections and continue.
11. Program your device. Connect your Nexys board to a USB port on your computer and power it on. Connect to the board by clicking on **Program and Debug > Open Hardware Manager > Open Target > Auto Connect** to automatically connect to your board. Once connection has been achieved, click on **Program Device** to program the board with the bitstream that you have generated. Observe behavior of your design on the board.

Lab Deliverables:

Submit a brief report including the approach you took to find the equivalent circuit, the snapshots of your simulation waveforms, and the snapshots of your running board. Also, you need to demonstrate your work and the completed steps in the demo time. Upload your solutions (one .zip file) including a lab report (.pdf file) and Lab3 module and testbench, Lab3_eq.v module and testbench, and top module and testbench (.v files) to the Dropbox labeled Lab 3.

Submission (Report & Verilog files) due date: Oct 19th 2020, 11:30PM

Demo: Oct 20th 2020, 6:00PM - 7:15PM & Oct 22nd 2020, 6:00PM - 7:15PM