Digital Design Techniques and Verification PROJECT - Stuck at Fault Detection

Objectives:

- To familiarize yourself with identifying a stuck at fault in a circuit
- To learn improve your verification skills via simulation
- To familiarize students with implementing HDL designs for a target technology

Theory:

Integrated circuit manufacturing processes can sometimes result in defects due to a number of reasons. These defects can manifest themselves in a number of ways, such as in minor faulty behaviors, total circuit failure, or they can even be undetectable. In this lab, we will analyze a circuit that exhibits a stuck-at-1 fault, such as that seen in the lectures. This lab assignment will help solidify the concepts covered in "Topic 5: Fault and Defect Modeling."

Assignment:

You are to instantiate a pre-written module that has been designed to behave as the circuit shown in Figure 1. However, the circuit has been modified to contain a single stuck-at-1 fault on one line. You are to design a series of test cases to determine where the fault is occurring, and demonstrate your results via simulation.

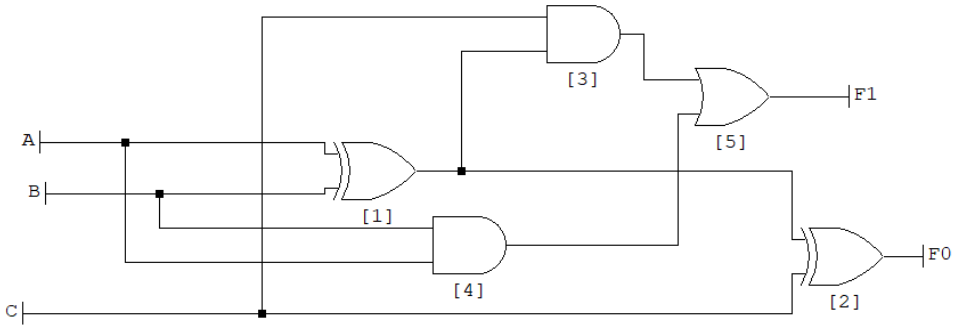

Figure 1. Expected Circuit Schematic

You must record the test cases that you created in a table, and include them in your report. Lastly, you are to synthesize and implement a top module that checks the outputs of the provided module, and shows when errors are detected by turning on the RGB light.

Steps:

1. Create a project and name it lab5.
2. Select the Nexys A7-100T (xc7a100tcsg324-1) board when prompted to select a target device.
3. Start by adding the provided circuit from BeachBoard (Circuit_SA1.v)
4. Create a testbench for this circuit and observe the functionality. Create a truth table for the circuit in Figure 1 and compare your results.
5. Identify the possible locations where a stuck at fault may occur and label them (Hint: there are 12 possible locations).
6. Create test cases for these locations that can identify if a stuck-at-1 fault is occurring. **Tabulate your test cases with the appropriate labels and include it in your lab report.**
7. Create another testbench with the test cases that you defined. Observe the results, and identify the location of the stuck-at-1 fault.
8. Create a new module and call it "Top Module". Instantiate the provided module (Circuit_SA1.v) inside, and also create the logic of the expected circuit (structurally or behaviorally, your choice). XOR the output F0 of the provided module with that of the expected result for F0, and XOR the output F1 of the provided module with the expected result of F1. Then OR this two outputs together. Whenever the output of this final OR is detected to be true based on the input values (meaning that an inconsistency has been found), turn on the RGB lights on your board to cycle through the following pattern: Red for 0.5 seconds, Green for 0.5 seconds, Blue for 0.5 seconds, cycling back to red and beginning the pattern all over again (hint: use a synchronous counter to calculate the duration of each stage of the RGB light output). Whenever the output of the final OR is false, leave the RGB light turned off. Your module must have the following inputs/outputs:
    a. clk: 1-bit input that will be driven by the on-board clock.
    b. reset: 1-bit input that will reset the value of the counter. Driven by BTNC.
    c. A: 1-bit input driven by SW0.
    d. B: 1-bit input driven by SW1.
    e. C: 1-bit input driven by SW2.
    f. RGB: 3-bit output that will control the onboard RGB light.
9. Add the NexysA7-100t.xdc file that has been provided on BeachBoard. Then, uncomment the lines for the pins that you are required to use and define the constraints for the top module, following the input/output port descriptions in step 8.
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.