Lab2

Hardware Trojan Insertion

Hardware Hacking

Goals:

In this lab you will examine the behavior of a traffic light controller circuit, and then attack it with a hardware Trojan.

Items to be submitted:

- Attacked top-level module
- Trojan-free behavioral waveform
- Trojan-impacted behavioral waveform
- FPGA Demonstration
- Answers to attached questions

Part 1: Examining the Traffic Light Controller.

Make a new project and all the included “.vhd”, “.v”, and “.ucf” files. The s298 circuit and the dff circuit it depends upon are written in verilog, but it is not necessary for you to read these files. The s298 circuit is an ISCAS benchmark circuit which is distributed for circuit testing purposes.

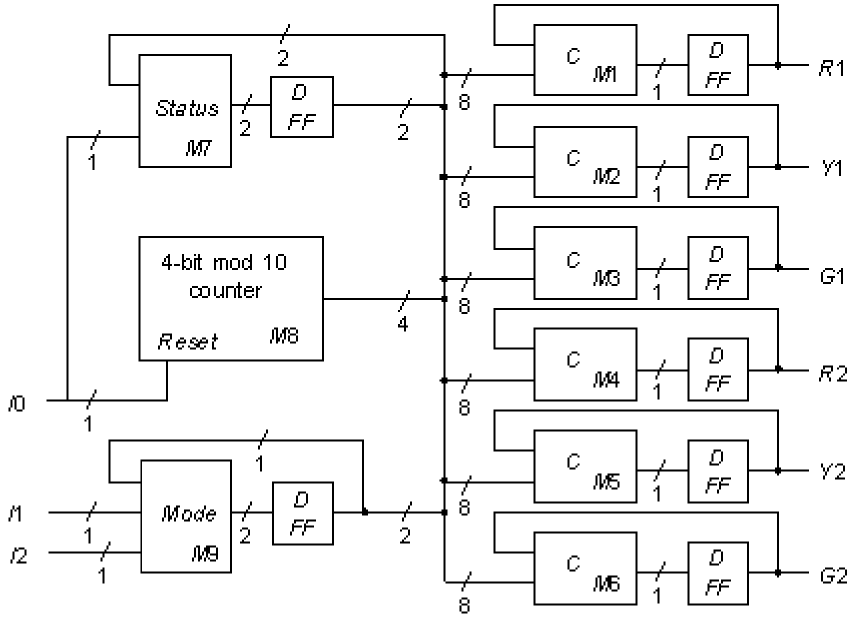

Fig1: s298 Circuit Diagram

It has three modes, but only one will be used for this lab. Each light has a controller and DFF to determine its state. While functionally correct it should signal only one direction with a green light at a time, and transitions should be buffered by yellow lights during which the other direction still has a red light. Simulate the top level module and examine its normal behavior. Save a screenshot of the waveform for submission.

Part 2: Trojan Insertion.

Open the top level module and notice the commented lines with a partial instantiation of a module called “crash_inducing_trojan.” Complete the instantiation of this module. You may need to remove the lines that assign Green2, Yellow2, and Red2. Simulate the top level module now that the circuit has been attacked. Make sure you are running the simulation long enough to see the change. Save a screenshot of the part of the waveform that has been changed.

Part3: FPGA Demonstration.

Demonstrate the traffic light controller on an FPGA. Use an LED for each output signal of the traffic light controller. The design assumes a long clock period so you will not be able to observe the behavior if you use the on-board clock. Instead, use a switch for the clock signal. Verify that at least one cycle of the correctly functioning pattern works and demonstrate this to the TA.

Part 4: Questions.
Attach your responses to the following questions to your submission.

1) If you were to demonstrate the malicious behavior of the Trojan on the FPGA in the same way you demonstrated the correct behavior in part 3, how many times would you have to flip the switch before the Trojan is activated? If the clock period were 15 seconds, how long would it take to activate the Trojan?

2) Why is this Trojan harmful? Describe qualitatively what it does upon activation. Refer to the waveform from part 2.

3) How would you best classify this Trojan using the Taxonomy shown in class? Be sure to include its physical, activation, and action characteristics.

4) Estimate the number of gates required to implement this Trojan in addition to the counter. (It may be helpful to first sketch what you expect the gate-level implementation of the Trojan to look like.)