ECE437: Sensors and Instrumentation

Lab 2 – Finite State Machines and XSim Simulator

Introduction

In this lab, you will learn how to write finite state machines (FSM) and how to debug FSM using XSim simulator in Vivado. There are two tutorial documents on course website located on the reference webpage as well as on labs webpage.

The first tutorial document explains how to write a Moore FSM as well as provides detail description various Verilog commands and syntax. You will need to read over the entire tutorial document before the lab so that you get acquainted with this material. The prelab questions are directly related to the material presented in the FSM tutorial.

The second tutorial document guides you through the XSim simulator in Vivado. This tutorial will show you how to setup a test bench file, simulate the behavior of the module you wrote and display the results on the screen. You will need to follow all the steps outlined in the tutorial so that you can simulate your own Verilog program.

Prelab Questions (10 points):

1. What is the difference between Mealy and Moore finite state machines?
2. What is the difference between `locaparam` and `parameter` in Verilog? When would you use either one of these declarations?
3. What is the difference between `wires` and `registers` in Verilog? List several syntax rules for both wires and registers?
4. When designing finite state machine module, when would you use a `wire` and when would you use `register`?
5. What is the difference between blocking (=) and non-blocking assignment?
Checkpoint 1 (50 points)

You will write Verilog code necessary to implement the state machine shown in the figure below. Each state corresponds to the number of LEDs that should be lit up (i.e. "state 7" means 7 LEDs should be on, doesn't matter which ones). B refers to the button vector, or the four on board buttons (remember, they are active low!). Lastly, if a state transition has no value shown, treat it like an else condition. Demonstrate your results to the TA once you are finished.

Checkpoint 2 (40 points)

In this exercise, you will simulate the Verilog code that you wrote in part 1 of this lab using XSim simulator in Vivado. Make sure you go through the XSim tutorial located on the course website. Simulate your circuit for 200 nsec and make sure you write a test bench that exhaustively tests your Verilog code (i.e. it show that your code transitions through all possible states).