Lab 7

Simple Finite State Machines

Introduction

In this assignment, you will be looking at different characteristics of finite state machine design for a simple modulo-5 counter (0, 1, 2, 3, 4, 0, 1…). First, you will create both Moore and Mealy state designs. Next, you will use behavioral Verilog to create and simulate both versions of the counter. Finally, you will put your designs on the Nexys A7 board and display the counter value on a seven-segment display.

Collaboration Policy

Lab groups may collaborate in completing the lab but may not consult other lab groups. Violating this policy is considered a violation of the Duke Community Standard.

Equipment

- Computer with Icarus and Vivado
- Nexys A7

Lab Tasks

Your lab TA must sign-off on the completion of each task in Sakai. You are not required to finish all tasks by the end of lab. However, you may not leave until all the provided tasks are completed or the lab block ends, whichever comes first.

1. Design both a Moore and Mealy style modulo-5 counter with a state diagram.
2. Implement the Moore and Mealy modulo-5 counter in Verilog and simulate in Icarus.
3. Upload the Moore counter to the Nexys A7.
4. Upload the Mealy counter to the Nexys A7.
5. Display the modulo-5 counter on the built-in seven-segment display.

Lab Instructions

State Machine Design

Design **both a Moore and Mealy design** modulo-5 counter as a state table or diagram. To do so, modify the standard mod-8 synchronous counter found in the lecture slides. Draw.io is a great website for creating flowcharts and state machine diagrams. Your output should be a single binary value that is high on the last state for the Moore machine and high on the last state when the input is also high for the Mealy machine. You should use separate inputs for the clock and signal to advance to the next state.

Task 1: Design both a Moore and Mealy style modulo-5 counter with a state diagram.

Implementing the Counter in Verilog

Implement both the Moore and the Mealy style modulo-5 counters. You will use one of the buttons on the Nexys A7 board as your input; therefore, your input to the state machine will be one-bit in width. You may use JK or D (or T if you're feeling adventurous) flip flops to implement your design. If you need help, refer to the modulo-8 counter in the lecture notes. In the simulation, have your counter trigger an output whenever your count reaches "4" (in the sequence 0-1-2-3-4-0-1-2-3-4...). Simulate your design in Icarus.

Task 2: Implement the Moore and Mealy modulo-5 counter in Verilog and simulate in Icarus.

Implement the Design on the Nexys A7

Set the Moore machine as your top-level module and open the I/O pin planner. Use one on-board LED to signify every time the input (i.e. button) is high, three to display the current state, and another LED to signify the output reaches four. You must use Pin P17 as your clock input (this button is special on the board as it is electrically configured to function as a clock (remember ECE 270 to understand why)), which is the left button on the board. Upload your Moore modulo-5 counter to the Nexys A7.

Task 3: Upload the Moore counter to the Nexys A7.

Set the Mealy machine as your top-level module and follow the same steps shown above.

Task 4: Upload the Mealy counter to the Nexys A7.

Implement the Design's Output on the Seven-Segment Display

Design a module so the counter value (current state) can be shown on the built-in seven-segment display. Create a project in Vivado that holds both your Moore and Mealy machines.

You will need to create a decoder that selects the right pins to light up on the seven-segment display. The interface for the seven segment display is described here.

Task 5: Display the modulo-5 counter on the built-in seven-segment display.

Grading

- Completing Lab Tasks: 100 points (pass/fail)