Stanford University
Department of Electrical Engineering

EE 121
Winter 2001-2002

Lab 4
Bicycle Rear Taillight

Prelab Due Date: Tuesday, February 5th, 2002 at 9:30am

Note Midterm #1 is on the due date!

1 Overview

This week's laboratory assignment is to design and build a synchronous sequential circuit: a "bicycle light" comprising of a button and a number of red LEDs. Push the button once and the LEDs shine steadily. Push again and they go off. Push a third time and the LEDs flash on and off a few times a second. Push one more time and they go off again. The system clock is ***4 Hz*** but the flashing must happen at ***1 Hz***. We will use the bar led on the Xilinx board as the bike LEDs and a DIP switch as the button to push. As you might expect, use the Xilinx Foundation software to implement your design and demonstrate it using the XC2S100 FPGA in the lab.

In order to gain facility in both binary and one-hot state machine design you will design the taillight with both methodologies. For the one-hot state machine design, use modified one-hot where the power on state is all zeros and the normal operating states are one-hot.

2 Methodology

1. Determination of inputs and outputs.
2. Determination of machine states.
3. Create State/Bubble Diagram—should this be a Mealy or Moore machine?
4. State Assignment—assign each state a particular value.
5. Create Transition/Output Table (Note: since we will only use D-flip flops, the transition and excitation tables are the same. For a DFF, Q* = D).
6. Derive Next State Logic for each state element—using K-maps as necessary.
7. Derive Output logic.
8. Implement in Xilinx.

Laboratory Requirements

For this laboratory assignment, partners should work together.

A top-level schematic will be provided on the course web page that includes signal names and pin assignments. Please make sure that all signals defined in this file are still in your design.

NOTE: For all your logic blocks, you are only allowed to use macros you build yourself and the following basic gates:

```
AND2, AND3, AND4, NAND2, NAND3, NAND4, NOR2, NOR3, NOR4,
INV, BUF, OR2, OR3, OR4, XOR2, XNOR2, M2_1, GND, VCC
```

For this assignment use the FDR (D-flipflop with synchronous reset) as the basic state element.

All macros you build previously, including the seven-segment decoder and three-to-eight decoder built in Lab 3, are available for use. Hint: Feed the current binary state (the output of the state flip-flops) through the seven-segment decoder connected to the seven-segment LED on the boards. This will allow you to quickly tell which state the state machine is in while working on the hardware. Similarly connect the one-hot state (the output of the state flip-flops) directly to the seven segment led outputs so you can know which state it is in. If your one-hot machine has more than seven states then use two seven segment LEDs.

What to turn in for this week's Prelab:

1. Complete Foundation Project with schematics of all logic blocks in your design (submitted electronically)
2. Test script demonstrating that your design works (submitted electronically—the top level script should be the same since there should be no difference between the two designs, but feel free to have separate scripts for lower modules.)
3. A report file that specifies resources used in the FPGA for the design—ie, the number of CLBs/IOBs/etc used. Look under the "reports" tab in Xilinx foundation. Summarize and discuss the results in your README in one paragraph (submitted electronically—again this needs to be done twice)
4. Extract timing information for the design as demonstrated in lecture and submit the file. The "Analyze against Auto Generated Design constraints…" button in the Xilinx Timing Analyzer tool will provide the values needed. Summarize and discuss the results in your README in one paragraph (submitted electronically—twice again)
5. Look at your design in the Xilinx FPGA Editor—be impressed it did all that wiring and not you! ☺ (JPEG screen dump submitted electronically for each design)
6. Readme file (submitted electronically—see below). Be sure to include comments about using binary state encoding versus modified one-hot.

Lab Exercise

During lab time, you will get a chance to compile your design, download it onto your Xilinx chip, and test it with real signals. There is no lab write-up for this lab. However, you do need to demo your circuit to your TA, using any test values your TA desires. Be prepared to demo both the binary encoded version and the one-hot encoded version.

Submission

Same rules as Lab3.

Remember, you must submit your project electronically, whether or not it works correctly, before 9:30 AM on the due date or you will receive no credit.