EECS 151/251A Homework 4

Due Monday, Feb 24th, 2020

For this HW Assignment

You will be asked to write several Verilog modules as part of this HW assignment. You will need to test your modules by running them through a simulator. As shown in discussion 2, a highly suggested simulator is [https://www.edaplayground.com](https://www.edaplayground.com) which is a free, online, Verilog simulator.

Warning: As per our EECS151 “no register inference policy”, you will need to use the register library in EECS151.v when using registers in your Verilog. EECS151.v is located at [https://inst.eecs.berkeley.edu/~eecs151/sp20/files/lib/EECS151.v](https://inst.eecs.berkeley.edu/~eecs151/sp20/files/lib/EECS151.v). We will only accept solutions using this library.

You can import this library by adding `include "EECS151.v"` to your Verilog code.

Examples of creating a testbench can be found in discussion 2 slides [https://inst.eecs.berkeley.edu/~eecs151/sp20/files/discussion2.pdf](https://inst.eecs.berkeley.edu/~eecs151/sp20/files/discussion2.pdf).

Problem 1. Vending Machine FSM

You need to design a vending machine controller that should exhibit the following behavior:

- The vending machine delivers a package of gum after it has received 15 cents in coins.
- The machine has a single coin slot that accepts only nickels and dimes, one coin at a time.
- A sensor indicates to the controller whether a nickel or dime has been inserted in the coin slot.
- The controller’s output causes a single package of gum to be released when it has received at least 15 cents, and the controller is reset.
- The machine does not give change. For example, if two dimes are inserted, it will release the gum and reset.

For this problem:

(a) Design a Moore finite state machine for the vending machine.
(b) Draw a state transition diagram for your FSM. Define your states, inputs, and outputs.
(c) Draw a circuit diagram consisting of logic gates and registers for your system. Don’t worry about optimal state encoding. Assume your Flip-Flop registers have RESET pins and can reset to any value. Don’t worry about optimal state encoding.
Problem 2. Traffic Light Controller FSM

There is a busy intersection in Berkeley at Martin Luther King Jr Way (MLK Way) and University Avenue (Uni Ave), as shown below. There are detectors on MLK Way that give a signal C if there is a vehicle waiting to cross Uni Ave. The Traffic Light Controller should exhibit the following behavior:

- The lights in the Uni Ave direction should be green as long as there are no cars on MLK Way.
- When a car is detected on MLK Way, the Uni Ave lights should go from yellow to red. MLK Way will be green when Uni Ave is red.
- MLK Way will stay green as long as there are cars detected or until a set interval has passed, since Uni Ave is busier and needs to return to green. If these conditions are met, MLK Way will go from green to yellow to red. Uni Ave will be green when MLK Way is red.
- Even if there are cars detected on MLK Way, Uni Ave will should stay green for a set interval.
- There is an external timer outside the controller that takes in a control signal ST (set timer) that asserts a signal TS after a short interval (for yellow to red timing) and a signal TL after a longer time interval (for green lights). The timer is reset when ST is asserted.

For this problem:

(a) Design a Mealy finite state machine for the Traffic Light Controller.
(b) Draw a state transition diagram for your FSM. Define your states, inputs, and outputs.
(c) Draw a circuit diagram consisting of logic gates and registers for your system. Assume your Flip-Flop registers have RESET pins and can reset to any value. Don’t worry about optimal state encoding.
(d) Design it in behavioral Verilog (using case statements), and provide a testbench and test results.
(e) Re-do part (c), For your state encoding, use one-hot encoding. Re-draw a circuit diagram consisting of logic gates and registers for your system accordingly.

Problem 3. Moore vs. Mealy Machine

Design a finite state machine that asserts a single output whenever its input string has at least two 1’s in a row. The input takes in 1 bit at a time. For example, an input sequence of 0110111 would yield an output sequence of 0010011. Since the 2nd and 3rd bits are both 1’s, the 3rd output would be 1. Since the 4th, 5th, and 6th bits are 1’s, the 5th and 6th output would be 1’s. For this problem:
(a) Draw diagrams for both a Moore and Mealy machine.

(b) Define your states, inputs, and outputs.

(c) Draw an expected waveform diagram.

(d) Design both in Verilog, and provide a testbench and test results.

**Problem 4: State Reduction**

Given the state transition diagram below, determine which states should be combined to determine the reduced state diagram. Use the implication chart method.

![State Transition Diagram](image-url)
Problem 5: State Assignment

Given the state transition diagram below, select a good state assignment, justifying your answer in terms of the state assignment heuristics presented in class.

Problem 6: FSM Complexity: 251A only — Optional Challenge Question for 151

(a) For a 12 state FSM, how many unique state encoding assignments exist assuming that the state register holds no fewer than 4 bits, and no greater than 12? You may write out the answer in an analytical form, a numerical answer is not necessary.

(b) How many unique FSMs of 1 input, 1 output, and 5 states exist? You may write out the answer in an analytical form, a numerical answer is not necessary.