University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS150, Spring 2012

Homework Assignment 3: Behavioral Verilog, Synthesis and Simulation and MIPS ISA Review

Due Feburary 9^{th}, 2pm

1. You have seen how a D flip-flop can be constructed from two latches. Use Verilog to describe a latch, and then instantiate two latches to build a D flip flop.

2. In homework 2, you designed a 3-bit $\overline{Up}/Down$ counter, now you are to implement it as an FSM. You can assume the counter output wraps around when it overflows/underflows.

   (a) Draw the state transition diagram

   (b) Implement your FSM using Verilog.

3. You are to design (using behavioral Verilog) a serial one hot code to binary converter which converts 4-bit one hot code to binary numbers. Instead of having all 4 bits of the one hot code at the same time, you would see one bit each clock cycle. Your circuit should have two outputs, one for the binary number, and another one indicating if the current binary number is valid. The valid signal should only be high after all 4 bits are seen and they constitute a legal one hot code. After the circuit decodes one 4-bit code or detect an invalid code, it would be reset before the next code is presented.

4. You are given three registers R_1, R_2 and R_3, each of which has an enable signal. A register would keep its old value at positive edge of the clock if the enable signal is not high. You will design a circuit capable of swapping the values in any two of the registers.

   (a) How would you connect the registers such that you can swap the values between any two of them? Make sure you minimize the number of clock cycles needed for the swap. Feel free to use any additional components. Draw the circuit diagram and label clearly the signals you would use to control the actions of each component.

   (b) Briefly describe the sequence of actions needed to swap the values in R_1 and R_2, referring to control signals you have in part 4a.

5. You have four registers each holding a number, you are to design a circuit to find the greatest number and put it into all four registers. You can use comparators, registers and other components in your implementation. Input s would start the operation and when the task is completed, a $done$ signal should be asserted.

   (a) For the first version of the circuit your are allowed to employ only one comparator.

      i. Arrange the four registers, the comparator, and any other necessary components into a datapath that can accomplish the described task. Draw a circuit diagram and label the signals you would use to control the actions of the datapath.

ii. Draw the state transition diagram for a FSM to control your circuit in part 5(a)i to accomplish the desired task. State clearly the value of each control signal in every state.

(b) In this part, devise a circuit that minimizes the number of clock cycles necessary. You are allowed to use more comparators and other components as necessary. (Hint: It is possible in one clock cycle.) Draw your resulting circuit.

6. Create a testbench for the full adder circuit you created during lab 0, make sure you cover all possible input combinations. If you are given a 16-bit adder, how many test cases would be needed for exhaustive testing?

7. Create a testbench for the FSM in question 2, make sure all state transitions are covered.

8. Using MIPS assembly, write a program which would allow you to determine the endian-ness of the machine. Explain how it works.

9. Translate the following MIPS code to machine code, write the instructions in hexadecimal.

```
0x7C                addi  $t2, $0, 0
0x80                addi  $t1, $0, 0x10
0x84      L1:       lb    $t3, 0x20($t1)
0x88                add   $t2, $t2, $t3
0x8C                addi  $t1, $t1, -1
0x90                bne   $t1, $0, L1
```