

University of California
Department of Electrical and Computer Engineering
ECE 156B
Synthesis & CAD

Homework 2

DUE: In Class, Thursday, Feb 12

Design of Simple Sequential Circuits

For each part, you need to turn in (hardcopy in class, email to lylee@ece.ucsb.edu):

- State diagram (hardcopy)
- RTL Verilog version of the circuit (hardcopy + email)
- Waveform of RTL simulation (hardcopy only)
- Area report by Design Compiler (hardcopy only)
- Gate Verilog version of circuit automatically generated by DC (hardcopy + email)
- Waveform of Gate simulation (hardcopy only)

Part a) Write and simulate an RTL Verilog description of a module (named **state_machine1**) that will watch a synchronous serial input and detect the pattern 0100. Assume that the data is stable at the rising edge of the clock and that the **rightmost-bit** in this pattern arrives first. Also assume an asynchronous reset that resets the machine to the initial state when 1. The machine will output a 1 when it has detected the last bit of the pattern, and 0 elsewhere.

- (i) Develop a state transition graph and implement the module as state machine.
- (ii) Implement the module with a shift register

The simulation results should show all the possible state transitions. Write comments in your code and in the simulation results.

Part b) Write RTL Verilog code for a state machine (named **state_machine2**) for a clocked sequential circuit that analyzes an input X and produces an output Z=1 whenever the input sequence ends with 1101 or 011. X arrives with the **leftmost-bit** first.

Example: For X = 0101101,
 Z = 0000101

Fully simulate your design and write comments throughout your code and in the simulation results.

Note: You can use your ECE 156A HW 4 solutions for the Verilog RTL, but as some of you may have noticed in HW 1, you might not be able to directly use your old code, as the Synopsys Design Compiler is more picky than ModelSim. Also, most of the testbenches in ECE 156A were not adequately showing all possible state transitions, so please include a more complete testbench for this assignment.