CPEN 211 Introduction to Microcomputers, 2024
Lab Proficiency Test #1

Question 1 [3 marks]: Create a file named "**q1.sv**" and inside it write **synthesizable** Verilog that implements the Moore finite state machine illustrated in Figure 1 below. Like the examples in class, state transitions occur on the rising edge of input "`clk`" and the reset is **synchronous** (occurs on the rising edge of `clk`) and is active high (`reset` equal to `1'b1` means reset). Bits on the left are most significant (have higher index value). Transitions from a state to itself are not explicitly shown. The condition for an unlabeled edge is always true. The input "`in`" is 2-bits wide. The output "`out`" is 2-bits wide. The output for each state is shown in square brackets. For example, when in state A, `out` should be 00, and if `in` is 10, then the next state should be B. The autograder used to mark your answer will assume your top-level module is called "`top_module`" with inputs "`clk`", "`reset`" and "`in`", output "`out`" declared **exactly** as follows:

```
module top_module(clk,reset,in,out);
  input clk, reset;
  input [1:0] in;
  output [1:0] out;
```

Your q1.sv file **must** include definitions for any modules instantiated inside `top_module` (even those from the slides or textbook). You should test your code and you can include testbench modules in q1.sv, but testbench modules inside q1.sv will be ignored by the autograder. **Upload your Verilog file named "q1.sv" by attaching it as your solution to Question 1 under "Lab Proficiency Test #1" on Canvas before 5:45 pm.** In case your computer's time differs from Canvas submit early and resubmit as needed. Marks will be deducted for submissions made after 5:45 pm and no submissions will be accepted after 5:50 pm **(emailed submissions will not be accepted).** Do **NOT** "zip" your submission. The file you upload for this question **must** be called "q1.sv" or the autograder script will not mark it.

Your solution for Question 1 will get zero if **any** of the following are true:

1. Your **last** "Lab Proficiency Test #1" attempt on Canvas does not include "q1.sv",
2. Your q1.sv file does not compile using ModelSim (e.g., due to syntax errors),
3. The Verilog used by your `top_module` is not synthesizable by Quartus,
4. Your q1.sv file does not contain a module `top_module` with inputs/outputs as above,
5. Your `top_module` cannot be simulated (e.g., due to missing module definitions in q1.sv),
6. The Verilog used by your `top_module` has inferred latches,
7. Your `top_module` output "`out`" does not exactly match the output of the state machine below for some sequence of "`clk`", "`reset`" and "`in`" values using **gate-level simulation**.

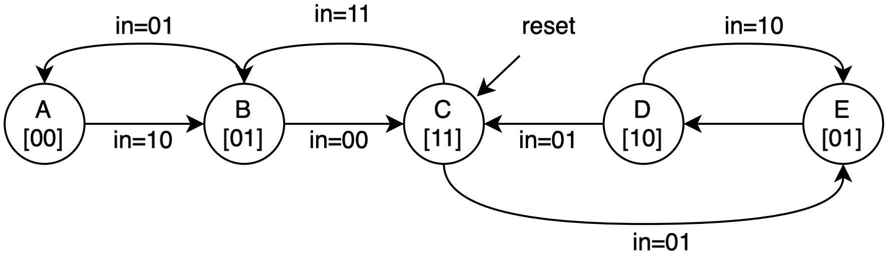

Figure 1

Question 2 [2 marks]: Note that this question is unrelated to Question 1. Create a file "**q2.sv**" and inside it write a testbench module "q2_tb". Your module **must** be called "q2_tb" should have no inputs or outputs and it must declare internal signals "clk", "reset" and "in" as shown below:

```
module q2_tb;
  reg clk, reset;
  reg [3:0] in;
```

Your testbench should generate the waveform illustrated in **Figure 2** for clk, reset and in when "q2_tb" is simulated in ModelSim and then stop at time 55 ps. Your q2.sv will get zero marks if any of the following are true:

1. Your **last** "Lab Proficiency Test #1" attempt on Canvas does not include "q2.sv",
2. Your q2.sv file does not compile using ModelSim (e.g., due to syntax errors),
3. Your q2.sv does not contain a module named q2_tb,
4. Your q2_tb cannot be simulated, or
5. The signal clk, reset and in in your q2_tb does not *exactly* match the following waveform from time 0 ps to time 55 ps.
6. Your testbench does not stop when time 55 ps.

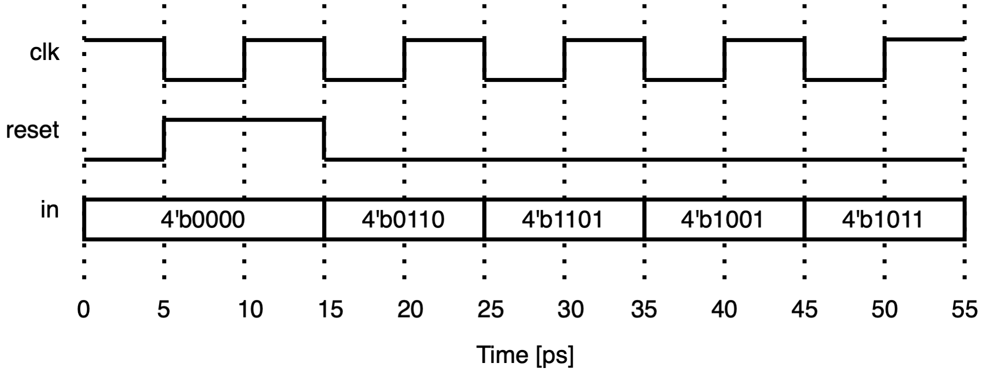

Figure 2