CPEN 211 Introduction to Microcomputers, 2021
Lab Proficiency Test #1

Question 1 [3 marks]: Create a file named "**q1.v**" and inside it write **synthesizable** Verilog that implements the Moore finite state machine illustrated in the figure 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 3-bits wide. The output for each state is shown in square brackets. For example, when in state A, `out` should be 000, and if `in` is 01, 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 as follows:

```
module top_module(clk,reset,in,out);
  input clk, reset;
  input [1:0] in;
  output [2:0] out;
```

Your q1.v 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.v, but testbench modules inside q1.v will be ignored by the autograder. **Upload your Verilog file named "q1.v" by attaching it as your solution to Question 1 under "Lab Proficiency Test #1" on Canvas <u>before</u> 6:45 pm.** In case your computer's time differs from Canvas submit <u>early</u> and resubmit as needed. Marks will be deducted for submissions made after 6:45 pm and no submissions will be accepted after 6:50 pm **(emailed submissions will <u>not</u> be accepted).** Do **NOT** "zip" your submission. The file you upload for this question **must** be called "q1.v" 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.v",
2. Your q1.v file does not compile using ModelSim (e.g., due to syntax errors),
3. Your q1.v file does not contain a module called `top_module` with inputs/outputs as above,
4. Your `top_module` cannot be simulated (e.g., due to missing module definitions in q1.v),
5. The Verilog used by your `top_module` is not synthesizable by Quartus.
6. The Verilog used by your `top_module` has any inferred latches.
7. Your `top_module` output "`out`" does not exactly match the output of the state machine below for some sequence of values for inputs "`clk`", "`reset`" and "`in`".

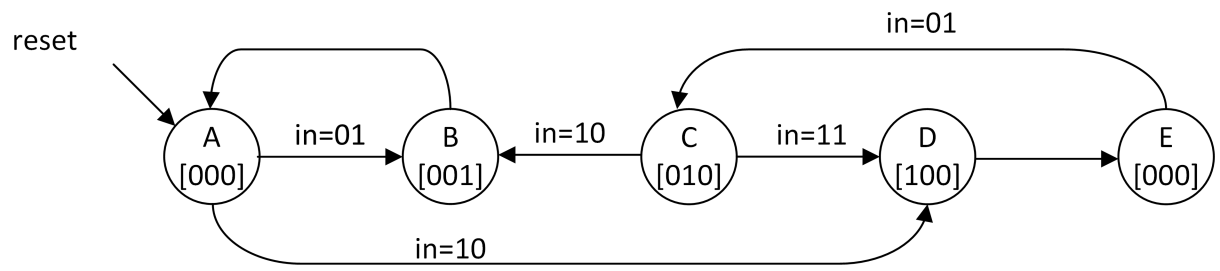

Figure 1

CPEN 211 Introduction to Microcomputers, 2021
Lab Proficiency Test #1

Question 2 [2 marks]: This question is not related to Question 1. Create a file named "**q2.v**" and inside it write **synthesizable** Verilog for the purely combinational logic block labeled "MealyDec" in Figure 2 (block on right) matching the following specification. The expected behavior of the overall circuit in Figure 2 is shown in Figure 3. You do **not** need to write Verilog for the Moore state machine in Figure 2 (block on left) and code for it is **not** provided. You do **not** need to use your solution to Question 1 to answer this question. Instead, your task is writing synthesizable Verilog for only the MealyDec combinational logic block. Upload your q2.v to Canvas as your answer for Question 2 in **"Lab Proficiency Test #1"**.

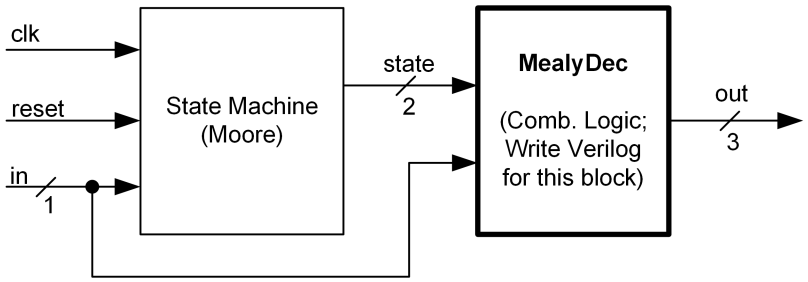

Figure 2

The inputs to `MealyDec` are a 2-bit signal *state* and one-bit single *in.* The signal *state* indicates the present state of the Moore state machine in Figure 2 (block on left). Assume the Moore machine in Figure 2 has the same states and transitions as the Mealy machine in Figure 3 below. Recall the state transitions of a Mealy machine are labeled with <input>/<output>. For example, when in State A if the input is 0 the output should be 3'b111 and the next state will be B. Bits on the left have higher index value. Assume the output *state* of the Moore state machine in Figure 2 is the Moore machine's present state. Specifically, assume *state* is 2'b00 when in State A, 2'b01 when in State B, 2'b10 when in State C, and 2'b11 when in State D. The 3-bit output *out* of `MealyDec` should be the 3-bit output corresponding to each transition in the diagram below. The label `x` in Figure 3 matches either a 0 or 1 value for input *in*.

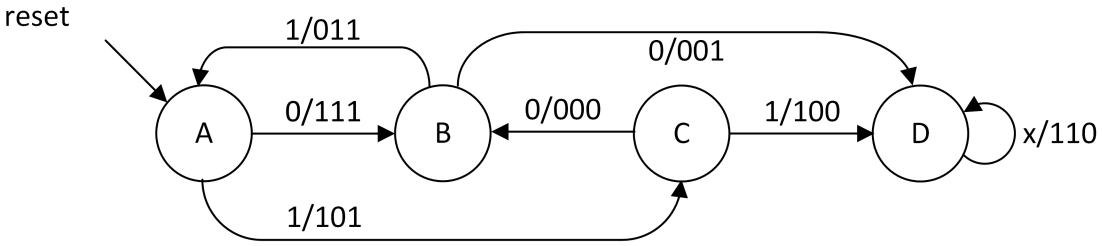

Figure 3

The autograder used to mark your answer will assume your top-level module is called "`MealyDec`" with inputs "`state`" and "`in`", and output "`out`" declared as follows:

```
module MealyDec(state,in,out);
  input [1:0] state;
  input in;
  output [2:0] out;
```

Ensure your final attempt for "Lab Proficiency Test #1" includes both q1.v and q2.v.