BEE 271 Spring 2017
Homework 6

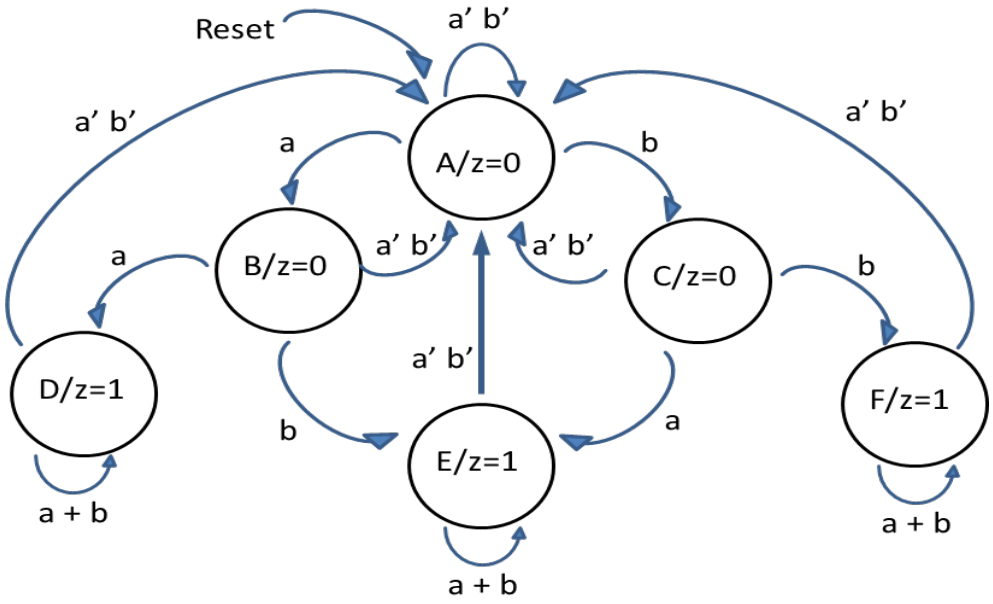
Please answer the following questions. Each is worth 8 points.

1. Create a Verilog module that implements a Moore machine that can report the number of ones on an ***asynchronous*** serial input sampled on the last 128 clocks. Please minimize the critical path from clock to output by avoiding trying to "add up" the ones.

2. Create a Verilog module for a Mealy version of the same machine that can report the number of ones in the current and last 127 bits on a ***synchronous*** serial input. Please minimize the critical paths from input or clock to output.

3. For two states to be equivalent, what has to be true about them?

4. The next several questions relate to this FSM. You may assume a clock and that the reset is synchronous and it's physically impossible for both a and b to be true at once. First, is this a Mealy or a Moore machine?



5. Create a Verilog module that implements this machine.

6. Create a state table for this FSM.

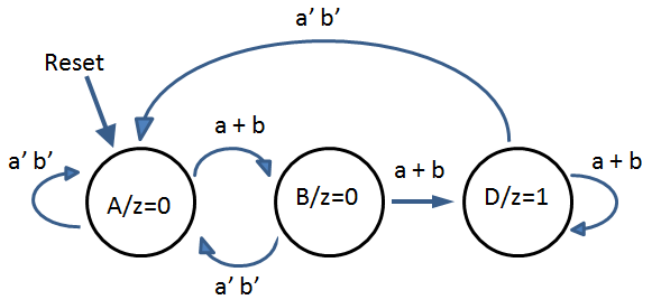
7. Find any equivalent states by partitioning, first by output, then by successors.

8. Create a minimized state table eliminating any equivalent states you discovered.

9. Create a state diagram for your minimized FSM.

10. Create a Verilog module that implements your minimized FSM.

11. Create a state diagram for a Mealy version of this FSM.



12. Create a Verilog module which implements your Mealy version.

13. Suppose you've been given a proposed, but clearly inefficient state diagram and asked to design the most elegant and efficient implementation you can. List three ways you might try to improve your design.