1. Fill in the timing diagram for the following circuit. Note that the DFF with the inverter bubble on the clock input is a negedge flipflop, and the other is a normal posedge flipflop. Also, gate delays are very small on the timing diagram shown.
Note: All machines must be Mealy machines, like we taught in class. That is, all outputs are given on the transitions (edges between states).

2. Design a string recognizer whose output will be true if the last four input values seen are “1000” or “1010”, including the current input. For this circuit (a) Draw the state diagram and (b) implement it with D flip-flops.

3. Create an implementation of the following finite state machine. You can use any basic gates, and D flip-flops.

![State Diagram](image)

4. We wish to design a soda vending machine which charges $.25 cents per soda (very economical). There is one coin slot, which will accept only quarters, nickels, and dimes. The controller has two outputs, Give_Soda and Return_Last_Coin. Draw the corresponding state diagram assuming the vending machine only accepts the correct change. You do NOT need to provide the state table or implementation. Notes: If the user provides more than $.25, you should return the last coin and wait for proper change. If you get the correct amount, you should dispense a soda and be ready for the next user.

5. When submarines fire torpedoes, there is a real danger of the weapon turning around and targeting the submarine. To deal with this, a failsafe is built into the device that blows up the torpedo. Build a failsafe. Assume the submarine is facing north when it fires the torpedo. Your circuit gets two signals, “L” if the torpedo command tells the torpedo to turn 90° left, and “R” if the torpedo command tells the torpedo to turn 90° right. If both are true, or neither, the torpedo keeps going straight. If the torpedo ever starts moving south, the Explode signal should be true, and stay true from then on. Draw the state diagram for this design.