Problem 1

Design a Mealy state machine to recognize strings that contain the substring "001". As symbols are processed, the machine should transition between states based on encountered symbols. Define the following states:

- State indicating that you haven't seen any symbols of the pattern "001."
- State indicating that you have just seen a "0."
- State indicating that you have just seen "00."
- State indicating that you have seen the entire pattern "001."

Assign transitions between these states based on symbol input (0 or 1). The machine should start in the initial state and transition accordingly, aiming to reach a final state upon recognizing the complete pattern. Provide a clear diagram illustrating the states and transitions of your Mealy state machine.

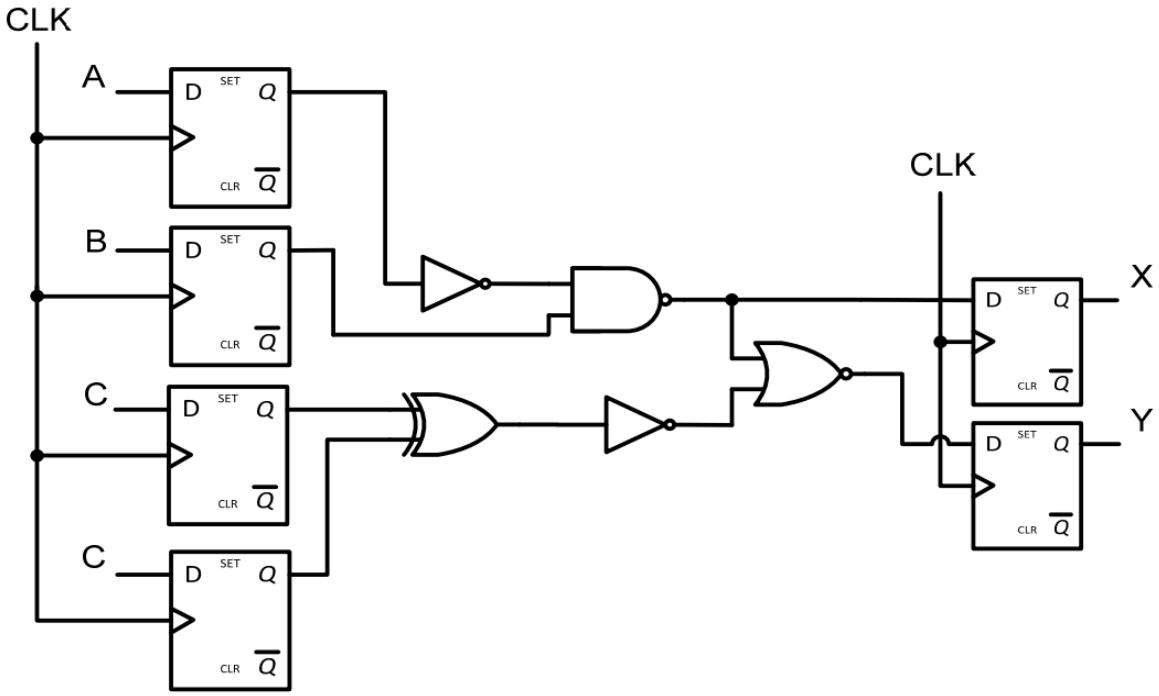
Problem 2

Design a finite state machine (FSM) for a counter that counts through the 3-bit prime numbers downwards. Assume the counter starts with the initial prime value set to 010 as its first 3 bit prime number. You need to provide the state transition table and the state transition diagram. Assume that the state is stored in three D-FFs. Hint: The set of all 3-bit prime numbers includes 2, 3, 5 and 7.

Design the circuit for the least significant bit (LSB) of the next state.

Problem 3

A sequential circuit design is shown in the following diagram.



- D-FF clk-to-q propagation delay t_pcq = 15 ps
- D-FF clk-to-q contamination delay t_ccq = 10 ps
- D-FF data setup time t_s = 15 ps
- D-FF data hold time t_h = 10 ps

Gate	T_{pd} (ps)	T_{cd} (ps)
2-input NAND	15	10
2-input NOR	25	15
2-input XOR	35	25
NOT	10	5

Calculate the maximum clock frequency for reliable operation assuming there is no clock skew.
(Note: Clock skew occurs when the clock signal does not arrive simultaneously at all the components due to differences in propagation delays.)

How much clock skew can the circuit tolerate before it experiences a hold time violation?

Problem 4

Draw a latch with two inputs S & R by implementing the functionality in the state transition table below. Use one 4:1 MUX and a minimum number of other gates.

S	R	Q(t)	Q(t+Δ)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Problem 5

Explain the difference between a Moore machine and a Mealy machine.

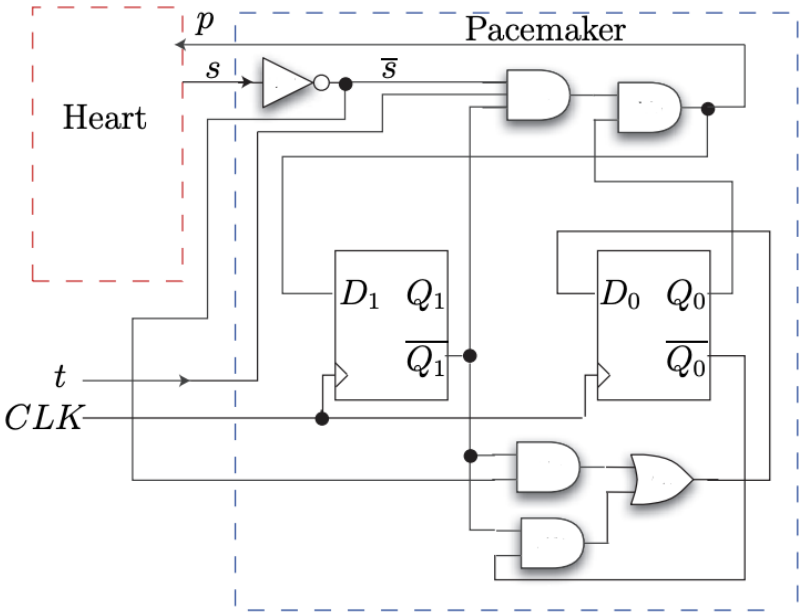
What is the same about both kinds of state machines?

Problem 6

Create a state diagram for a sequence detector that outputs a 1 when it detects the final bit in the serial data stream 1101. Assume overlapping is allowed.

Problem 7

Below is a simplified block diagram of a heart pacemaker. The output p from the pacemaker pulses high if the heart does not contract within a certain time. The input s indicates whether the heart has contracted (s = 1) or not (s = 0). The input t comes from a timer, 6 which counts for the expected time between contractions (approximately 1 second). When t = 1, the timer has counted up to 1 second, and the heart should have contracted. If the heart has not contracted within that time, the pacemaker sends a pulse p = 1. (There is also a timer reset control coming from the pacemaker, which we ignore for this problem.) The inputs to the pacemaker circuit are s (from the heart) and t (from a timer - not shown). The output of the pacemaker circuit is p, which goes to the heart. Arrows indicate the inputs and output.



Note: The black dots in the figure indicate a connection between wires. If there is no black dot, there is no connection between wires that cross.
Is this a Mealy or a Moore machine? Why?

Write the equations for output p and the flip-flop inputs D1 and D0 from the block diagram.

Problem 8

You want to build a finite state machine that will recognize the sequence x = 0110 and output the sequence z = 0001 as this sequence occurs. In other words, output z = 0 when first receiving x = 0. Then output z = 0 if the next bit of x = 1; output z = 0 again if the following bit of x = 1. Finally, if the last (fourth) bit of x = 0, output z = 1. More simply, output z = 0 until the sequence x = 0110 is received, at which time output z = 1.

First, draw and label the transitions in the state bubble diagram below. The states are already labeled (but state bit values have not been assigned). Allow overlap of sequences. Build a Moore machine. Include the input bits of x and output bits of z.

Redraw the state diagram using a Mealy machine design. Be sure to label the transitions and bubbles. You may name your states whatever you like. Again allow overlap of sequences. How many states and flip-flops do you need for the Mealy design?