1. (15 points) Verilog and State diagram

Draw the state diagram specified by following Verilog program. Briefly comment what it is.

```verilog
module Guess_what_prob1(CLK, RESET, X, Z)
input CLK, RESET, X;
output Z;
reg[1:0] state, next_state;

always@(posedge CLK or posedge RESET)
begin
  if (RESET == 1)
    state <= 2'b00;
  else
    state <= next_state;
end

always @(X or state)
begin
  case (state)
    2'b00: if (X == 1) next_state <= 2'b10;
           else next_state <= 2'b00;
    2'b01: if (X == 0) next_state <= 2'b00;
           else next_state <= 2'b01;
    2'b10: if (X == 1) next_state <= 2'b11;
           else next_state <= 2'b00;
    2'b11: if (X == 0) next_state <= 2'b00;
           else next_state <= 2'b01;
    default: next_state <= 2'bxx;
  end

endmodule
```

**Answer:** This can be either a Moore state diagram of pattern “111” recognizer or a 2-bit gray code counter with enable (X = EN).
2. (20 points) Synchronous circuit analysis

Given a synchronous sequential circuit that contains two state variables \( A(t) \) and \( B(t) \). State variable \( A(t) \) is the output of a T flip-flop, and state variable \( B(t) \) is the output of a SR flip-flop. \( Z(t) \) is the output of the circuit.

(a) (10 points) Derive the next state equations for \( A(t+1) \) and \( B(t+1) \).

**Give the equations in simplified SOP format.** You must show your work to receive credit.

**Answers:** Using a characteristic equation for \( Q(t+1) = T(t) \oplus Q(t) \) and \( T(t) = A(t) \oplus B(t) \cdot X(t) \) . \( A(t+1) = A(t) \oplus (B(t) \cdot X(t)) \oplus A(t) = (A(t) \oplus A(t)) \oplus (B(t) \cdot X(t)) = 0 \oplus (B(t) \cdot X(t)) = B(t) \cdot X(t) . \)

Using a characteristic equation for \( Q(t+1) = S(t) + \overline{R}(t) \cdot Q(t) \) and \( S(t) = A(t) \oplus B(t) \cdot X(t) . \)

and \( R(t) = B(t) \cdot X(t) . \) \( B(t+1) = A(t) \oplus B(t) \cdot X(t) + \overline{B}(t) \cdot X(t) \cdot B(t) = \)
\[ A(t) \cdot B(t) \cdot X(t) + A(t) \cdot B(t) \cdot X(t) + (B(t) + X(t)) \cdot B(t) = \\
A(t) \cdot \overline{B(t)} + A(t) \cdot B(t) + \overline{X(t)} \cdot B(t) . \]

\[
A(t+1) = \frac{B(t) \cdot X(t)}{A(t) \cdot \overline{B(t)} + A(t) \cdot B(t) + \overline{X(t)} \cdot B(t)}
\]

\[
B(t+1) = \frac{A(t)}{A(t) \cdot \overline{B(t)} + A(t) \cdot B(t) + \overline{X(t)} \cdot B(t)}
\]

(b) (10 points) Assume the initial state is \( A(t)B(t) = 00 \), where \( A(t)B(t) \) is the binary value of the state in two binary digits. Fill in the blanks of the following table.

<table>
<thead>
<tr>
<th>( t )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X(t) )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( Z(t) )</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>( A(t)B(t) )</td>
<td>01</td>
<td><strong>11</strong></td>
<td><strong>01</strong></td>
<td><strong>01</strong></td>
<td><strong>11</strong></td>
</tr>
</tbody>
</table>

Notice that if you use the characteristic equation \( Q(t+1) = S(t) + \overline{R(t)} \cdot Q(t) \), which was presented in class, you are filling in 1’s for the two don’t care conditions (QSR=100 and QSR=111). In other words, you are treating them as set dominant FFs to drive the characteristic equation above. If you are assuming reset-dominant FFs, you may get a slightly different implementation for \( B(t+1) \) in part (a). If you are not using characteristic equations, then you will need to decide how to complete the next state entry when \( X=1, A=0 \) and \( B=1 \). This may give different results than the provided solutions in (a) and (b).

3. (15 points) Sequential circuits and timing analysis

The clock (CLK) signal is connected to a following sequential circuit with output \( Y \).

The clock (CLK) signal is connected to a sequential circuit with output \( Y \). Complete the timing diagram given below. Assume that the gate delays are negligible.

**Answer:** The circuit shown above is simply a negative edge triggered JK’ flip-flop. Notice that it consists of one SR-latch (a 4-NAND gate network), one D-latch (SR-latch with an input feeding S and inverted input feeding R), and simple three gate network to implement,
hold, reset, set and toggle. The timing diagram of a negative edge triggered JK’ flip-flop should be as follows.

4. (20 points) State Diagram Synthesis
   (a) (15 points) A sequential circuit has one input X(t), and one output Z(t). Whenever the input sequence consists of a stream of 010 or 0001 (three 0’s in a row then 1), the output will become 1 during the clock cycle when the last symbol is received. Otherwise, the output equals to 0. Derive a state diagram of this synchronous sequential circuit.

   **Answer:** A is initial state, C is detected-010 state, and E is detected-0001 state.

   (b) (5 points) Is your implementation of state diagram of part (a) Mealy or Moore?
   **Answer:** Mealy

   Is it possible to implement your state diagram in part (a) using the other type (Mealy/Moore) to satisfy the timing requirement and keep the number of states same? Explain why or why not using only a sentence or two!

   **Answer:** No, you will need seven states with the Moore implementation.

5. (30 points) State diagram, state table and synchronous circuit design
   The state diagram shown below is for a synchronous sequential circuit with unknown function. This sequential circuit has two state variables S1(t) and S0(t), and an input C. State variables S1(t) and S0(t) are to be implemented with JK flip-flops.
(a) (10 points) Complete the state table below.

<table>
<thead>
<tr>
<th>C</th>
<th>S1(t)</th>
<th>S0(t)</th>
<th>S1(t+1)</th>
<th>S0(t+1)</th>
<th>S1J</th>
<th>S1K</th>
<th>S0J</th>
<th>S0K</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(b) (10 points) Express the result in a minimized SOP standard format. You must show your work to receive full credit!

\[
S_{1S} = C \cdot S_0 \\
S_{1R} = C \cdot S_0 \\
S_{0S} = C \\
S_{0R} = C \\
Z = C \cdot S_1 \cdot S_0
\]
(c) (10 points) Suppose you are to use a pair of SR flip-flops to implement the circuit with a minimum logic. Would this change your implementation/design?

**Answer:** (circle only one **Yes** or **No**)

If Yes, list all implementation/design changes. If No, show why it would not change your implementation. You must show your work to get credit.

With the SR-FFs, green X’s will become 0’s and red X’s will become 0’s. Output Z will not change.

\[
\begin{align*}
S_{1s} &= C \cdot S_0 + \overline{C} \cdot \overline{S_1} \cdot S_0 \\
S_{0s} &= C \cdot \overline{S_0} \\
S_{1r} &= C \cdot S_0 + \overline{C} \cdot S_1 \cdot S_0 \\
S_{0r} &= C \cdot \overline{S_0} 
\end{align*}
\]