5.1 (a) Construct an SM chart equivalent to the following state table. Test only one variable in each decision box. Try to minimize the number of decision boxes.

(b) Write a VHDL description of the state machine based on the SM chart.

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>Output Z_1 Z_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X_1 X_2 =</td>
<td>X_1 X_2 =</td>
<td></td>
</tr>
<tr>
<td>S0</td>
<td>S3 S2 S1 S0</td>
<td>00 00 10 11</td>
</tr>
<tr>
<td>S1</td>
<td>S0 S1 S2 S3</td>
<td>01 10 10 11 11</td>
</tr>
<tr>
<td>S2</td>
<td>S3 S0 S1 S1</td>
<td>11 00 00 01</td>
</tr>
<tr>
<td>S3</td>
<td>S2 S2 S1 S0</td>
<td>01 01 00 00</td>
</tr>
</tbody>
</table>

(a)
entity 5_1 is
  port (CLK, X1, X2 : in bit;
        Z1, Z2 : out bit);
end 5_1;

architecture Smbehave of 5_1 is
  type states is {S0, S1, S2, S3};
  signal state, next_state : states;
begin
  process (X1, X2, state)
  begin
    Z1 <= '0';  Z2 <= '0';
    case state is
      when S0 => if (X1 = '0') then
        if (X2 = '0') then
          next_state <= S3;
        else
          next_state <= S2;
        end if;
      else
        if (X2 = '0') then
          Z1 <= '1';
          Z2 <= '1';
          next_state <= S1;
        else
          Z2 <= '1';
          next_state <= S0;
        end if;
      end if;
      when S1 => if (X1 = '0') then
        Z1 <= '1';
        if (X2 = '0') then
          next_state <= S0;
        else
          next_state <= S1;
        end if;
      else
        Z1 <= '1'; Z2 <= '1';
        if (X2 = '0') then
          next_state <= S2;
        else
          next_state <= S3;
        end if;
      end if;
      when S2 => if (X1 = '0') then
        if (X2 = '0') then
          next_state <= S3;
        else
          Z1 <= '1';
          next_state <= S0;
        end if;
      else
        next_state <= S1;
        Z2 <= '1';
        if (X2 = '0') then
          Z1 <= '1';
        end if;
      end if;
    end case;
  end process;
end Smbehave;
end if
end if;
when S3 => if (X1 = '0') then
next_state <= S2;
elself
if (X2 = '0') then
Z2 <= '1';
next_state <= S1;
elself
Z2 <= '1';
next_state <= S0;
end if
end if;
end case;
end process;
process (CLK)
begin
if (CLK = '1' and CLK'event) then
state <= next_state;
end if;
end process;
end Smbehave;

5.6 For the given SM chart:
(a) Complete the following timing diagram (assume that X1 = 1, X2 = 0, X3 = 0, X5 = 1, and X4 is as shown)/ Flip-flops change state on falling edge of clock.
10.1 (a) Determine the necessary inputs to the following network to test for \( u \) stuck-at-0.
(b) For this set of inputs, determine which other stuck-at faults can be tested.
(c) Repeat (a) and (b) for \( r \) stuck-at-1.

(a) & (b) A B C D Faults Tested
\[
\begin{array}{cccc}
  d & 1 & 0 & 1 & u \text{ s-a-0}, b \text{ s-a-0}, d \text{ s-a-0}, w \text{ s-a-0} \\
  d & 1 & 1 & 0 & u \text{ s-a-0}, b \text{ s-a-0}, c \text{ s-a-0}, w \text{ s-a-0} \\
  d & 1 & 1 & 1 & u \text{ s-a-0} \\
\end{array}
\]

(c) A B C D Faults Tested
\[
\begin{array}{cccc}
  0 & 1 & 0 & 0 & a \text{ s-a-1}, c \text{ s-a-1}, d \text{ s-a-1}, q \text{ s-a-1}, r \text{ s-a-1}, u \text{ s-a-1}, v \text{ s-a-1}, w \text{ s-a-1}, p \text{ s-a-0} \\
\end{array}
\]

10.3 For the following network, find a minimum number of test vectors that will test all \( \text{s-a-0} \) and \( \text{s-a-1} \) faults at the AND and OR gate inputs. For each test vector, specify the values of A, B, C, and D, and the stuck-at faults that are tested.

A B C D Faults Tested
\[
\begin{array}{cccc}
  0 & 1 & d & 1 & k \text{ s-a-0}, l \text{ s-a-0}, m \text{ s-a-0}, r \text{ s-a-0}, z \text{ s-a-0} \\
  1 & 0 & 1 & d & h \text{ s-a-0}, i \text{ s-a-0}, j \text{ s-a-0}, q \text{ s-a-0}, z \text{ s-a-0} \\
  0 & 0 & 1 & 1 & h \text{ s-a-1}, i \text{ s-a-1}, p \text{ s-a-1}, q \text{ s-a-1}, r \text{ s-a-1}, z \text{ s-a-1} \\
  0 & 1 & 0 & 0 & e \text{ s-a-1}, m \text{ s-a-1}, p \text{ s-a-1}, q \text{ s-a-1}, r \text{ s-a-1}, z \text{ s-a-1} \\
  1 & 0 & 0 & d & f \text{ s-a-1}, j \text{ s-a-1}, p \text{ s-a-1}, q \text{ s-a-1}, r \text{ s-a-1}, z \text{ s-a-1} \\
  1 & 1 & 1 & 1 & g \text{ s-a-1}, i \text{ s-a-1}, k \text{ s-a-1}, p \text{ s-a-1}, q \text{ s-a-1}, r \text{ s-a-1}, z \text{ s-a-1} \\
\end{array}
\]

10.7 State graphs for two sequential machines are given below. The first graph represents a correctly functioning machine, and the second represents the same machine with a malfunction. Assuming that the two machines can be reset to their starting states (S0 and T0), determine the shortest input sequence that will distinguish the two machines.

Input: 0 0 0 1 1 1 1 1
Correct Output: 1 1 1 0 0 0 1
Incorrect Output: 1 1 1 0 0 0 0