Members of group (5 members):

Procedure: After all members of the group have introduced themselves by stating their name, high school they graduated from and what they did over the summer, solve the following state machine design problem; only one solution submitted per group.

Problem: Design a clocked synchronous state machine that asserts \( Z \) (i.e., \( Z=1 \)) when a clocked input variable \( X = 1 \) for two sequential positive clock transitions, after which \( Z = 0 \). This means that if \( X = 1 \) for four sequential positive clock transitions that \( Z \) will be asserted twice, and for six positive transitions, \( Z \) is asserted three times, etc. Define all variables and state all assumptions that are used in your solution.

a) determine a state diagram for your solution.

b) present the state/output table for your state diagram

c) define a state assignment, and present the transition/output table (present state/next state table)

d) assuming the use of D flip-flops, present the transition (excitation) equations for the design, and the output equations

e) present a logic circuit for your design identifying all signal lines
2 new 15 sequence detector

Moore

-state machine design

<table>
<thead>
<tr>
<th>S</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Y1 = \frac{Y0 \cdot X}{\bar{X} + 1}

Y0 = \frac{Y0 \cdot \bar{X}}{X}

Y1, Y0, \bar{Z}

clock

Meady

-clever mealy

-state machine design

<table>
<thead>
<tr>
<th>S</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Y1 = \frac{Y0 \cdot X}{\bar{X} + 1}

Y0 = \frac{Y0 \cdot \bar{X}}{X}

Y1, Y0, \bar{Z}

clock

-inputs

-output

1) problem statement/state diagram
2) state/output table
3) state assignment
4) transition/output table
5) transition equations
6) excitation eqs (D, T, ff)
7) output eqs.
8) logic diagram

-clock

-output
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

entity counter is port ( clk, rst : in std_logic;
q : inout std_logic_vector (1 downto 0));
end counter;

architecture behavioral of counter is
begin
process (clk, rst)
begin
if rst = '1' then q <= "00";
elseif rising_edge (clk) then q <= q + 1;
end if;
end process;
end behavioral;

architecture behavioral of state_machine is
signal COUNT: std_logic_vector (9 downto 0);
signal SLOW_CLK: std_logic;
begin
frequency_divider:
process (CLK)
begin
if rising_edge (CLK) then COUNT <= COUNT + 1;
end if;
SLOW_CLK <= COUNT(9);
end process;

state_machine:
process (SLOW_CLK, RST)
begin
...
end process;
end behavioral;
library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity counter is port (  
  set, shorten, clk : in std_logic;
  q : out std_logic_vector (1 downto 0);
  z1, z2 : out std_logic);
end counter;

architecture behavioral of counter is  
type state_type is (a,b,c,d);
signal ps, ns: state_type;
begin  
  sync_proc:
  process (set, clk)  
  begin  
    if set = '1' then ps <= a;  
    elsif rising_edge (clk) then ps <= ns;  
    end if;  
  end process;
  
  comb_proc:  
  process (ps, shorten)  
  begin  
    z1 <= '0'; z2 <= '0'; -- default values for Moore and Mealy outputs for case statement  
    case ps is  
      when a => q <= "11"; ns <= b;  
      when b => q <= "10"; ns <= c;  
      when c => q <= "01"; if shorten = '1' then z1 <= '1'; ns <= a;  
      else ns <= d;  
      end if;  
      when d => q <= "00"; z2 <= '1'; ns <= a;  
    end case;  
  end process;
end behavioral;