EXPERIMENT NO. 9

Aim: Modelling, simulation and synthesis of a sequence detector which has one input and one output. If the input sequence is “01” or “10” then an output of the detector is ‘1’.

Perform the following activities.

Activity 1: **Draw the Mealy state diagram for a 01/10 sequence detector. Overlapping is allowed.**

Activity 2: Write the VHDL code for the sequence detector.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity mealy_seq_det is
 port(clk,rst,x: in std_logic;
      z: out std_logic);
end mealy_seq_det;

architecture Behavioral of mealy_seq_det is
type mealy_state is (s0,s1,s2);
signal reg_ps,reg_ns: mealy_state;

begin

--- register
 process(rst,clk)
  begin
    if (rst='1') then
      reg_ps<=s0;
   elsif(clk'event and clk='1') then
    reg_ps<=reg_ns;
    end if;
   end process;
```

```
---- Next State Logic
  process(x,reg_ps)
    begin
     case reg_ps is
      when s0 =>
       if(x='0') then
         reg_ns<=s1;
         else
         reg_ns<=s2;
         end if;

      when s1 =>
       if(x='0') then
         reg_ns<=s1;
         else
         reg_ns<=s2;
        end if;
      when s2 =>
       if(x='0') then
         reg_ns<=s1;
         else
         reg_ns<=s2;
        end if;
    when others =>
      reg_ns<=s0;
  end case;
end process;
```

```
--- Output Logic
  process(x,reg_ps)
    begin
     case reg_ps is

      when s1 =>
       if(x='0') then
         z<='0';
         else
         z<='1';
       end if;
      when s2 =>
       if(x='0') then
         z<='1';
         else
         z<='0';
       end if;
    when others =>
      z<='0';
  end case;
 end process;
end Behavioral;
```

Activity 3: Verify the operation of the design using VHDL test bench.

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;


ENTITY mealy_seq_det_tb IS
END mealy_seq_det_tb;

ARCHITECTURE behavior OF mealy_seq_det_tb IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT mealy_seq_det
    PORT(
         clk : IN  std_logic;
         rst : IN  std_logic;
         x : IN  std_logic;
         z : OUT  std_logic
        );
    END COMPONENT;


   --Inputs
   signal clk : std_logic := '0';
   signal rst : std_logic := '0';
   signal x : std_logic := '0';
```

```
   --Outputs
   signal z : std_logic;

   -- Clock period definitions
   constant clk_period : time := 50 ns;

BEGIN

   -- Instantiate the Unit Under Test (UUT)
   uut: mealy_seq_det PORT MAP (
          clk => clk,
          rst => rst,
          x => x,
          z => z
        );

   -- Clock process definitions
   clk_process :process
   begin
      clk <= '0';
      wait for clk_period/2;
      clk <= '1';
      wait for clk_period/2;
   end process;

   -- Stimulus process
   stim_proc: process
   begin
   rst<= '1'; wait for 5 ns;
   rst<= '0';

   x<='0'; wait until falling_edge(clk);
   x<='0'; wait until falling_edge(clk);
   x<='1'; wait until falling_edge(clk);
   x<='1'; wait until falling_edge(clk);
   x<='0'; wait until falling_edge(clk);
   x<='1'; wait until falling_edge(clk);
   x<='0'; wait until falling_edge(clk);
   x<='0'; wait until falling_edge(clk);
   x<='1'; wait until falling_edge(clk);
   x<='0'; wait until falling_edge(clk);



      wait;
   end process;

END;
```

Activity 4: Perform the synthesis of sequence detector design.

Note: Set the FPGA device before the synthesis (FPGA Family- Spartan 3E, Device-XC3S100E).

Number of Slices	
Number of Slice FFs	
Number of LUTs	
IOBs	
Minimum Clock period	