```
/*-------------------------------------------------------------
Test Bench for Sequence_100_Detector_Mealy Finite State Machine
CLass: EE417 Summer 2024
Lesson 04 HW Question 1
Group: Ron Kalin/ Lamin Jammeh
-------------------------------------------------------------*/

module Sequence_100_Detector_Mealy_TB ();

//define the input and outputs as wires and registers
wire  z2, z1;
reg   clk, S_reset;
reg   x_in;

//define the internal probes as wires
wire  [2:0] present_state, next_state;

//define the unit under test (UUT)
Sequence_100_Detector_Mealy UUT (z2, z1, clk, S_reset, x_in);

//internal probes to make it easy to track the logic and for troubleshooting
assign present_state = UUT.present_state;
assign next_state = UUT.next_state;

//generate the clk cycle with a period of 5 ns
initial
   begin
      clk = 1'b0;
      forever
      begin
         #5 clk = ~clk;
      end
   end

//initialize the reset cycle
initial
   begin
      S_reset = 1'b1;
      #30  S_reset = 1'b0;
      #200 S_reset = 1'b1;
      #30  S_reset = 1'b0;
   end

//test the possible input sequence combination to form the different states
initial
   begin
      x_in = 1'b0; #15

      begin

         x_in = 1'b0;   #10   x_in = 1'b0;   #10 x_in = 1'b0;  #10;  //S_000  sequence
         x_in = 1'b0;   #10   x_in = 1'b0;   #10 x_in = 1'b1;  #10;  //S_001  sequence
         x_in = 1'b0;   #10   x_in = 1'b1;   #10 x_in = 1'b0;  #10;  //S_010  sequence = z1
         x_in = 1'b0;   #10   x_in = 1'b1;   #10 x_in = 1'b1;  #10;  //S_011  sequence
         x_in = 1'b1;   #10   x_in = 1'b0;   #10 x_in = 1'b0;  #10;  //S_100  sequence = z2
         x_in = 1'b1;   #10   x_in = 1'b0;   #10 x_in = 1'b1;  #10;  //S_101  sequence
         x_in = 1'b1;   #10   x_in = 1'b1;   #10 x_in = 1'b0;  #10;  //S_110  sequence
         x_in = 1'b1;   #10   x_in = 1'b1;   #10 x_in = 1'b1;  #10;  //S_111  sequence
      end
   end

   // Monitor outputs
    initial begin
        $display("x_in____Present_State_____Next_State____Z1____Z1____");
        $monitor("%b,      %b,      %b,      %b,    %b", x_in, present_state, next_state,
z1, z2);
    end
endmodule
```