**Shift register code**

module shift\_register(

input wire clk,

input wire reset,

input wire x\_i, //serial input

output wire [3:0] sr\_o

);

logic [3:0]sr\_ff;

logic [3:0]nxt\_sr;

always\_ff@(posedge clk or posedge reset)

if(reset)

sr\_ff <= 4'h0;

else

sr\_ff <= nxt\_sr;

assign nxt\_sr = {sr\_ff[2:0],x\_i};

assign sr\_o = sr\_ff;

endmodule

**Test Bench Code**

module tb\_shift\_register();

logic clk,reset;

logic x\_i;

logic [3:0]sr\_o;

shift\_register day6(.\*);

always begin

clk=1'b0;

#5;

clk=1'b1;

#5;

end

initial begin

reset <= 1'b1;

x\_i <= 1'b0;

@(posedge clk);

reset <= 1'b0;

@(posedge clk);

for (int i=0; i<16; i=i+1) begin

x\_i <= $random%2;

@(posedge clk);

end

$finish();

end

// Dump VCD

initial begin

$dumpfile("day6.vcd");

$dumpvars(2, tb\_shift\_register);

end

endmodule

**Simualtion**

