**Code:**

* **Design Module**

// Code your design here

module SD\_101(clock,reset,in,out);

input clock,reset,in;

output out;

localparam S0 = 2'b00, // local parameters can only be set from with the design file

S1 = 2'b01,

S2 = 2'b10,

S3 = 2'b11;

reg [1:0] current\_state;

reg [1:0] next\_state;

// state memory

always @(posedge clock)

begin

if (reset)

current\_state <= S0;

else

current\_state <= next\_state;

end

//transition logic

always @ ( in or current\_state)

begin

case(current\_state)

S0 :

begin

if(in == 0)

next\_state <= S0;

else

next\_state <= S1;

end

S1 :

begin

if(in == 0)

next\_state <= S2;

else

next\_state <= S1;

end

// S2: next\_state <= in ? S3 : S2; //ternary form

// S3: next\_state <= in? S1:S2;

S2 :

begin

if(in == 0)

next\_state <= S0;

else

next\_state <= S3;

end

S3 :

begin

if(in == 0)

next\_state <= S2;

else

next\_state <= S1;

end

default: next\_state <= S0;

endcase

end

//output logic

assign out = (current\_state == S3) ? 1 : 0 ;

endmodule

* **TestBench**

// Code your testbench here

// or browse Examples

module tb();

reg clock,reset,in;

wire out;

SD\_101 dut(clock,reset,in,out);

always #10 clock = ~clock; //forever clock

initial // initial system value handling

begin

clock <=1;

reset <=0;

end

initial // 500 times random input @ negative edge of clock

begin

for (int i = 0; i<=500 ; i++) begin @(negedge clock)

in = $random;

end

end

// only for eda playground wave generation

// initial // wave generation

// begin

// $dumpfile("dump.vcd");

// $dumpvars;

// #1000

// $finish;

// end

endmodule

**Output:**

