```
 1  module lights(clk, reset, in, out);
 2     input logic clk, reset;
 3     input logic [1:0] in;
 4     output logic [2:0] out;
 5
 6     logic [1:0] ps, in1, in2;
 7
 8     always begin
 9        case (ps)
10           2'b0:    out = 3'b101;
11           2'b1:    out = 3'b100;
12           2'b10:   out = 3'b001;
13           2'b11:   out = 3'b010;
14           default: out = 3'bX;
15        endcase
16     end
17
18     always_ff @(posedge clk) begin
19        in1 <= in;
20     end
21
22     always_ff @(posedge clk) begin
23        in2 <= in1;
24     end
25
26     //DFF
27     always_ff @(posedge clk) begin
28        if (reset)
29           ps <= 2'b11;
30        else
31           case (ps)
32              2'b0:    ps <= 2'b11;
33              2'b1:    if (in2 == 2'b1)  ps <= 2'b10;
34                       else              ps <= 2'b11;
35              2'b10:   if (in2 == 2'b10) ps <= 2'b1;
36                       else              ps <= 2'b11;
37              2'b11:   ps <= in2;
38              default: ps <= 2'bX;
39           endcase
40     end
41
42  endmodule
43
44  module lights_testbench();
45     logic clk, reset;
46     logic [1:0] in;
47     logic [2:0] out;
48
49     lights dut(clk, reset, in, out);
50
51     // Set up the clock.
52     parameter CLOCK_PERIOD=100;
53     initial begin
54        clk <= 0;
55        forever #(CLOCK_PERIOD/2) clk <= ~clk;
56     end
57
58     // Set up the inputs to the design. Each line is a clock cycle.
59     initial begin
60                                  @(posedge clk);
61        reset <= 1;               @(posedge clk);
62        reset <=0; in <= 2'b0;    @(posedge clk);
63                                  @(posedge clk);
64                                  @(posedge clk);
65        in <= 2'b10;              @(posedge clk);
66                                  @(posedge clk);
67                                  @(posedge clk);
68                                  @(posedge clk);
69                                  @(posedge clk);
70                                  @(posedge clk);
71                                  @(posedge clk);
72        in <= 2'b01;              @(posedge clk);
73                                  @(posedge clk);
74                                  @(posedge clk);
```

```
75                                  @(posedge clk);
76                                  @(posedge clk);
77                                  @(posedge clk);
78          in <= 2'b0;             @(posedge clk);
79          in <= 2'b1;             @(posedge clk);
80          in <= 2'b10;            @(posedge clk);
81          $stop;   // End the simulation
82       end
83    endmodule
84
85
```