

```

1  module centerLight (clk, reset, L, R, NL, NR, lightOn);
2      input logic clk, reset;
3      // L is true when left key is pressed, R is true when the right key
4      // is pressed, NL is true when the light on the left is on, and NR
5      // is true when the light on the right is on.
6      input logic L, R, NL, NR;
7      // when lightOn is true, the center light should be on.
8      output logic lightOn;
9      // Your code goes here!!
10
11     enum {on, off} ps, ns;
12     //
13     always_comb begin
14         case(ps)
15             on: lightOn = 1;
16
17             off: lightOn = 0;
18         endcase
19     end
20
21     always_comb begin
22         case(ps)
23             on: if(~R & L | R & ~L) ns = off;
24                 else ns = on;
25
26             off: if(NR & L & ~R | NL & R & ~L) ns = on;
27                 else ns = off;
28
29         endcase
30     end
31
32     always_ff @(posedge clk) begin
33         if(reset)
34             ps <= on;
35         else
36             ps <= ns;
37     end
38 endmodule
39
40 module centerLight_testbench();
41     logic clk, reset, L, R, NL, NR, lightOn;
42
43     normalLight dut (.clk, .reset, .L, .R, .NL, .NR, .lightOn);
44
45     parameter CLOCK_PERIOD = 100;
46     initial begin
47         clk <= 0;
48         forever #(CLOCK_PERIOD / 2)
49             clk <= ~clk;
50     end
51
52     initial begin
53         reset <= 1; @(posedge clk);
54         reset <= 0; @(posedge clk);
55         L <= 0; R <= 0; NL <= 0; NR <= 0; repeat(4) @(posedge clk);
56             NR <= 1; repeat(4) @(posedge clk);
57             NL <= 1; NR <= 0; repeat(4) @(posedge clk);
58             NR <= 1; repeat(4) @(posedge clk);
59             R <= 1; NL <= 0; NR <= 0; repeat(4) @(posedge clk);
60             NR <= 1; repeat(4) @(posedge clk);
61             NL <= 1; NR <= 0; repeat(4) @(posedge clk);
62             NR <= 1; repeat(4) @(posedge clk);
63             L <= 1; R <= 0; NL <= 0; NR <= 0; repeat(4) @(posedge clk);
64             NR <= 1; repeat(4) @(posedge clk);
65             NL <= 1; NR <= 0; repeat(4) @(posedge clk);
66             NR <= 1; repeat(4) @(posedge clk);
67             R <= 1; NL <= 0; NR <= 0; repeat(4) @(posedge clk);
68             NR <= 1; repeat(4) @(posedge clk);
69             NL <= 1; NR <= 0; repeat(4) @(posedge clk);

```

