

```

1  // Eugene Ngo
2  // 12/2/2022
3  // EE 271
4  // Lab 5
5
6  // centerLight module takes in clk, reset, L, R, NL, NR, restart as inputs and they are all
  1-Bit and returns
7  // lightOn which is also 1-Bit. This module determines the next state based on the inputs
  and moves the light
8  // to the left if key 3 is pressed and moves the light to the right if key 0 is pressed.
  When the game is reset
9  // or restarted then this module recenters the light for the next game.
10
11 module centerLight (lightOn, clk, reset, L, R, NL, NR, restart);
12     output logic lightOn;
13     input logic clk, reset;
14     input logic L, R, NL, NR, restart;
15
16     enum {on, off} ps, ns;
17
18     always_comb begin
19         case(ps)
20             on:    if(~R & L | R & ~L) ns = off;
21                   else ns = on;
22
23             off:   if(NR & L & ~R | NL & R & ~L) ns = on;
24                   else ns = off;
25
26         endcase
27
28     end
29     always_comb begin
30         case(ps)
31             on: lightOn = 1;
32
33             off: lightOn = 0;
34         endcase
35     end
36
37     always_ff @(posedge clk) begin
38         if(reset | restart)
39             ps <= on;
40         else
41             ps <= ns;
42     end
43
44 endmodule
45
46 // centerLight_testbench tests all expected, unexpected, and edgecase behaviors
47 module centerLight_testbench();
48     logic L, R, NL, NR;
49     logic lightOn;
50     logic clk, reset;
51
52     centerLight dut (.lightOn, .clk, .reset, .L, .R, .NL, .NR);
53
54     //Set up the clock
55     parameter CLOCK_PERIOD = 100;
56     initial begin
57         clk <= 0;
58         forever #(CLOCK_PERIOD / 2)
59             clk <= ~clk;
60     end
61
62     //Set up the inputs to the design. Each line is a clock cycle
63     initial begin
64
65         reset <= 1;
66
67         reset <= 0;
68
69         L <= 1; R <= 0; NL <= 0; NR <= 1;
70

```

```
71      @(posedge clk)
72      @(posedge clk)
73      NL <= 1; NR <= 0;    @(posedge clk)
74      @(posedge clk)
75      @(posedge clk)
76      @(posedge clk)
77      L <= 0; R <= 1;    @(posedge clk)
78      @(posedge clk)
79      @(posedge clk)
80      @(posedge clk)
81      NL <= 0; NR <= 1;    @(posedge clk)
82      @(posedge clk)
83      @(posedge clk)
84      @(posedge clk)
85      L <= 1;      NL <= 1; NR <= 0;    @(posedge clk)
86      @(posedge clk)
87      @(posedge clk)
88      @(posedge clk)
89      reset <= 1;    @(posedge clk)
90      @(posedge clk)
91      @(posedge clk)
92      L <= 0; R <= 1; NL <= 1; NR <= 0;    @(posedge clk)
93      @(posedge clk)
94      @(posedge clk)
95      L <= 1; R <= 0; NL <= 0; NR <= 0;    @(posedge clk)
96      @(posedge clk)
97      @(posedge clk)
98      L <= 0; R <= 1; NL <= 0; NR <= 0;    @(posedge clk)
99      @(posedge clk)
100     @(posedge clk)
101     $stop;
102     end
103 endmodule
```