```
1   /*
2      Caitlin DeShazo-Couchot and Ashley Guillard
3      EE 271 Au23: Professor Hussein
4      November 9, 2023
5      Lab 4: Center LED (LEDR 5)
6   */
7
8   // Handles the LED5 (Center LED) during the game of Tug-of-War. Also starting or reset LED.
9   // Inputs: CLOCK, Reset, Left Button, Right Button, Left Light, and Right Light.
10  // Outputs: The state of the LED this is referring to, either On or Off.
11  module centerLight (CLOCK, Reset, L, R, NL, NR, lightOn);
12
13     // INPUT LOGIC
14     input logic CLOCK, Reset;
15
16     // GAME LOGIC
17     // L is true when left key is pressed, R is true when the right key is pressed.
18     // NL is true when the light on the left is on, and NR is true when the light on the
       right is on.
19     input logic L, R, NL, NR;
20
21     // OUTPUT LOGIC: When lightOn is true, the normal light should be on.
22     output logic lightOn;
23
24
25     // State Variables
26     enum logic [1:0] { OFF, ON } ps, ns;
27
28     // Next State Logic
29     always_comb begin
30        case (ps)
31           OFF:
32              if ((NL & R & ~L) | (NR & L & ~R))
33                 ns = ON;
34              else
35                 ns = OFF;
36           ON:
37              if ((R & ~L) | (L &~R))
38                 ns = OFF;
39              else
40                 ns = ON;
41        endcase
42     end
43
44     // Output Logic
45     always_comb begin
46        case (ps)
47           OFF:
48              lightOn = 1'b0;
49           ON:
50              lightOn = 1'b1;
51        endcase
52     end
53
54     // DFFs
55     always_ff @(posedge CLOCK) begin
56        if (Reset)
57           ps <= ON;
58        else
59           ps <= ns;
60     end
61
62  endmodule
63  //Tests all possible combinations for the normalLight
64  module centerLight_testbench ();
65
66     // Test inputs and outputs
67     logic CLOCK, Reset, L, R, NL, NR;
68     logic lightOn;
69
70     // Instantiate the hazard_lights module
71     centerLight dut (.CLOCK, .Reset, .L, .R, .NL, .NR, .lightOn);
72
73      //CLOCK setup
74     parameter CLOCK_period = 10;
75
```

```
76      initial begin
77         CLOCK <= 0;
78         forever #(CLOCK_period) CLOCK <= ~CLOCK;
79
80      end //initial
81
82      // Test cases for Tug of War game
83      initial begin
84         Reset <= 1;                                          @(posedge CLOCK); //Reset
85         Reset <= 0;    L = 0; R = 0; NL = 0; NR = 0; @(posedge CLOCK);
86                                                 NR = 1; @(posedge CLOCK);
87                                       NL = 1; NR = 0; @(posedge CLOCK);
88                                                 NR = 1; @(posedge CLOCK);
89                               R = 1; NL = 0; NR = 0; @(posedge CLOCK);
90                                                 NR = 1; @(posedge CLOCK);
91                                       NL = 1; NR = 0; @(posedge CLOCK);
92                                                 NR = 1; @(posedge CLOCK);
93                         L = 1; R = 0; NL = 0; NR = 0; @(posedge CLOCK);
94                                                 NR = 1; @(posedge CLOCK);
95                                       NL = 1; NR = 0; @(posedge CLOCK);
96                                                 NR = 1; @(posedge CLOCK);
97                               R = 1; NL = 0; NR = 0; @(posedge CLOCK);
98                                                 NR = 1; @(posedge CLOCK);
99                                       NL = 1; NR = 0; @(posedge CLOCK);
100                                                NR = 1; @(posedge CLOCK);
101        $stop;
102     end //initial
103  endmodule
```