

```

1  module winner (clk, reset, LED9, LED1, L, R, user);
2      input logic clk, reset;
3      input logic LED9, LED1, L, R;
4      output logic [6:0] user;
5
6      enum {off, P1, P2} ps, ns;
7
8      always_comb begin
9          case(ps)
10             off:    if(LED9 & L & ~R) ns = P1;
11
12                     else if(LED1 & R & ~L) ns = P2;
13
14                     else ns = off;
15
16             P1: ns = P1;
17
18             P2: ns = P2;
19
20          endcase
21
22          if(ns == P1) user = 7'b1111001;
23          else if (ns == P2) user = 7'b0100100;
24          else user = 7'b1111111;
25
26      end
27
28      always_ff @(posedge clk) begin
29          if(reset)
30              ps <= off;
31          else
32              ps <= ns;
33      end
34
35  endmodule
36
37  module winner_testbench();
38      logic clk, reset, LED9, LED1, L, R, user;
39
40      winner dut (.clk, .reset, .LED9, .LED1, .L, .R, .user);
41
42      parameter CLOCK_PERIOD = 100;
43      initial begin
44          clk <= 0;
45          forever #(CLOCK_PERIOD / 2)
46              clk <= ~clk;
47      end
48
49      initial begin
50          reset <= 1; @(posedge clk);
51          reset <= 0; @(posedge clk);
52          LED9 <= 0; LED1 <= 0; L <= 0; R <= 0; repeat(4) @(posedge clk);
53                                     R <= 1; repeat(4) @(posedge clk);
54                                     L <= 1; R <= 0; repeat(4) @(posedge clk);
55                                     R <= 1; repeat(4) @(posedge clk);
56          LED1 <= 1; L <= 0; R <= 0; repeat(4) @(posedge clk);
57                                     R <= 1; repeat(4) @(posedge clk);
58          L <= 1; R <= 0; repeat(4) @(posedge clk);
59                                     R <= 1; repeat(4) @(posedge clk);
60          LED9 <= 1; LED1 <= 0; L <= 0; R <= 0; repeat(4) @(posedge clk);
61                                     R <= 1; repeat(4) @(posedge clk);
62          L <= 1; R <= 0; repeat(4) @(posedge clk);
63                                     R <= 1; repeat(4) @(posedge clk);
64          LED1 <= 1; L <= 0; R <= 0; repeat(4) @(posedge clk);
65                                     R <= 1; repeat(4) @(posedge clk);
66          L <= 1; R <= 0; repeat(4) @(posedge clk);
67          R <= 1; repeat(4) @(posedge clk);
68          $stop;
69      end

```

