```
module Victory(clk, reset, l_in, r_in, l_on, r_on, out);
   input  logic clk, reset, l_in, r_in, l_on, r_on;
   logic[3:0] in;
   assign in[0] = r_on;
   assign in[1] = l_on;
   assign in[2] = r_in;
   assign in[3] = l_in;

   output logic [6:0] out;

   logic [1:0] ps, ns;

   always_comb begin
      case (ps)
         2'b00:   if       (in == 4'b1010)   ns = 2'b10;
                  else if  (in == 4'b0101)   ns = 2'b01;
                  else                       ns = 2'b00;
         2'b01:                              ns = 2'b01;
         2'b10:                              ns = 2'b10;
      endcase
   end

   always_ff @(posedge clk) begin
      if (reset)
         begin
            ps    <= 2'b00;
            out   <= 7'b0;
         end
      else
         begin
            ps <= ns;
            case (ps)
               2'b01:   out <= 7'b1111001;
               2'b10:   out <= 7'b0100100;
               2'b00:   out <= 7'b0;
            endcase
         end
   end
endmodule

module Victory_testbench();
   logic clk, reset, l_in, r_in, l_on, r_on;
   logic [6:0] out;

   Victory dut(clk, reset, l_in, r_in, l_on, r_on, out);

   parameter CLOCK_PERIOD=100;

   initial begin
      clk <= 0;
      forever #(CLOCK_PERIOD/2) clk <= ~clk;
   end

   initial begin                                                    @(posedge clk);
                                                                    @(posedge clk);
      reset <= 1;                                                   @(posedge clk);
      reset <= 0; l_in <= 0; r_in <= 0; l_on <= 0; r_on <= 0;       @(posedge clk);
                                                                    @(posedge clk);
      l_in <= 1;                                                    @(posedge clk);
      l_in <= 0;                                                    @(posedge clk);
                                                                    @(posedge clk);
                                                                    @(posedge clk);
      l_in <= 1;                                                    @(posedge clk);
      l_in <= 0; l_on <= 1;                                         @(posedge clk);
                                                                    @(posedge clk);
                                                                    @(posedge clk);
      l_in <= 1;                                                    @(posedge clk);
      l_in <= 0; l_on <= 0;                                         @(posedge clk);
                                                                    @(posedge clk);
                                                                    @(posedge clk);
      l_in <= 1;                                                    @(posedge clk);
      l_in <= 0;                                                    @(posedge clk);
                                                                    @(posedge clk);
                                                                    @(posedge clk);
```

```
75          r_in <= 1;                                                          @(posedge clk);
76          r_in <= 0;                                                          @(posedge clk);
77                                                                              @(posedge clk);
78                                                                              @(posedge clk);
79          reset <= 1;                                                         @(posedge clk);
80          reset <= 0; l_in <= 0; r_in <= 0; l_on <= 0; r_on <= 0;             @(posedge clk);
81                                                                              @(posedge clk);
82          r_in <= 1;                                                          @(posedge clk);
83          r_in <= 0;                                                          @(posedge clk);
84                                                                              @(posedge clk);
85                                                                              @(posedge clk);
86          r_in <= 1;                                                          @(posedge clk);
87          r_in <= 0; r_on <= 1;                                               @(posedge clk);
88                                                                              @(posedge clk);
89                                                                              @(posedge clk);
90          r_in <= 1;                                                          @(posedge clk);
91          r_in <= 0; r_on <= 0;                                               @(posedge clk);
92                                                                              @(posedge clk);
93                                                                              @(posedge clk);
94          r_in <= 1;                                                          @(posedge clk);
95          r_in <= 0;                                                          @(posedge clk);
96                                                                              @(posedge clk);
97                                                                              @(posedge clk);
98          l_in <= 1;                                                          @(posedge clk);
99          l_in <= 0;                                                          @(posedge clk);
100                                                                             @(posedge clk);
101                                                                             @(posedge clk);
102         reset <= 1;                                                         @(posedge clk);
103         reset <= 0; l_in <= 0; r_in <= 0; l_on <= 0; r_on <= 0;             @(posedge clk);
104                                                                             @(posedge clk);
105                                                                             @(posedge clk);
106         $stop;
107      end
108   endmodule
109
```