```
module Victory2(clk, reset, in, out);
   input  logic clk, reset;
   // Left On, Right On, Left In, Right In
   input  logic [3:0] in;
   output logic [1:0] 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;
         default                             ns = 2'b00;
      endcase
   end

   always_ff @(posedge clk) begin
      if (reset)
         ps <= 2'b00;
      else
         ps <= ns;
      out <= ns;
   end
endmodule

module Victory_testbench();
   logic clk, reset, l_in, r_in, l_on, r_on;
   logic [1:0] out;

   Victory2 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);
      r_in <= 1;                                               @(posedge clk);
      r_in <= 0;                                               @(posedge clk);
                                                               @(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);
      r_in <= 1;                                               @(posedge clk);
      r_in <= 0;                                               @(posedge clk);
                                                               @(posedge clk);
                                                               @(posedge clk);
      r_in <= 1;                                               @(posedge clk);
      r_in <= 0; r_on <= 1;                                    @(posedge clk);
                                                               @(posedge clk);
                                                               @(posedge clk);
      r_in <= 1;                                               @(posedge clk);
      r_in <= 0; r_on <= 0;                                    @(posedge clk);
                                                               @(posedge clk);
                                                               @(posedge clk);
      r_in <= 1;                                               @(posedge clk);
      r_in <= 0;                                               @(posedge clk);
                                                               @(posedge clk);
                                                               @(posedge clk);
      l_in <= 1;                                               @(posedge clk);
      l_in <= 0;                                               @(posedge clk);
                                                               @(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);
                                                               @(posedge clk);
      $stop;
   end
endmodule

```