```
module Light(clk, reset, is_center, l_in, r_in, l_on, r_on, out);
   input logic    clk, reset, is_center, 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   out;

   typedef enum logic {ON, OFF} state;
   state ps, ns;

   always_comb begin
      case (ps)
         ON:   if ((in == 4'b0100) | (in == 4'b1000)) ns = OFF;
               else                                   ns = ON;
         OFF:  if ((in == 4'b0110) | (in == 4'b1001)) ns = ON;
               else                                   ns = OFF;
      endcase
   end

   assign out = (ps == ON);

   always_ff @(posedge clk) begin
      if (reset)
         if (is_center) ps <= ON;
         else           ps <= OFF;
      else
         ps <= ns;
   end

endmodule

module CenterLight_testbench();

   logic       clk, reset, is_center, l_in, r_in, l_on, r_on;
   logic       out;

   assign is_center = 1;

   Light dut(clk, reset, is_center, 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);
         reset <= 1;                                              @(posedge clk);
         reset <= 0; l_in <= 0; r_in <= 0; l_on <= 0; r_on <= 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);
         l_in <= 1;                                               @(posedge clk);
         l_in <= 0; r_on <= 1;                                    @(posedge clk);
                                                                  @(posedge clk);
                                                                  @(posedge clk);
         l_in <= 1;                                               @(posedge clk);
         l_in <= 0; r_on <= 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);
         r_in <= 1;                                               @(posedge clk);
         r_in <= 0; l_on <= 1;                                    @(posedge clk);
                                                                  @(posedge clk);
                                                                  @(posedge clk);
         r_in <= 1;                                               @(posedge clk);
         r_in <= 0; l_on <= 0;                                    @(posedge clk);
                                                                  @(posedge clk);
                                                                  @(posedge clk);
         reset <= 1;                                              @(posedge clk);
         reset <= 0;                                              @(posedge clk);
      $stop;
   end
endmodule

module NormalLight_testbench();

   logic       clk, reset, is_center, l_in, r_in, l_on, r_on;
   logic       out;

   assign is_center = 0;

   Light dut(clk, reset, is_center, 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);
         reset <= 1;                                              @(posedge clk);
         reset <= 0; l_in <= 0; r_in <= 0; l_on <= 0; r_on <= 0;  @(posedge clk);
                                                                  @(posedge clk);
                                                                  @(posedge clk);
         r_in <= 1;                                               @(posedge clk);
         r_in <= 0; l_on <= 1;                                    @(posedge clk);
                                                                  @(posedge clk);
                                                                  @(posedge clk);
         r_in <= 1;                                               @(posedge clk);
         r_in <= 0; l_on <= 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);
         l_in <= 1;                                               @(posedge clk);
         l_in <= 0; r_on <= 1;                                    @(posedge clk);
                                                                  @(posedge clk);
                                                                  @(posedge clk);
         l_in <= 1;                                               @(posedge clk);
         l_in <= 0; r_on <= 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);
         reset <= 1;                                              @(posedge clk);
         reset <= 0;                                              @(posedge clk);
      $stop;
   end
endmodule
```