```
//Top module for lab4
module State_Machine (tail_lights,clock,left,right,brake,hazards,reset);
//Input Ports
input     clock,left,right,brake,hazards,reset;
//Output Ports
//output rc,rb,ra,la,lb,lc;
output [5:0] tail_lights;
//Input Ports data type
wire      clock,left,right,brake,hazards,reset;
//Output Ports
//reg     rc,rb,ra,la,lb,lc;
reg [5:0] tail_lights;
//Internal Constants
//Declare all the possible states
parameter OFF  = 6'b000000;
parameter ON   = 6'b111111;
parameter LAR0 = 6'b001000;
parameter LBR0 = 6'b011000;
parameter L1R0 = 6'b111000;
parameter L1RA = 6'b111100;
parameter L1RB = 6'b111110;
parameter L0RA = 6'b000100;
parameter L0RB = 6'b000110;
parameter L0R1 = 6'b000111;
parameter LAR1 = 6'b001111;
parameter LBR1 = 6'b011111;

reg [5:0] current_state;
reg [5:0] next_state;

//Here, we start by the having all leds OFF
initial begin
          current_state = OFF;
end

//We are assigning neg clok edge for state changes
//Similiar to : we are using neg clock edge for writing
//and pos edge for reading or checking states
always @(negedge clock)
begin: FSM_SEQ
          current_state = next_state;
          tail_lights = next_state;
end

//Pos edge clock for checking states
always @(posedge clock) begin

            // we start first by checking the reset
            if (reset == 1)
                    next_state <= OFF; // if reset enabled then all leds are off

            //Check for hazard-like request
            else if ((right == 1 && left == 1) || hazards == 1)begin
                    if (brake == 1 || current_state == OFF)
                              next_state = ON; //Brakes given priority over hazards
                    else
```

```
                        next_state = OFF;
        end

        //Handle brake requests not involving turn signals or hazards
        else if (right == 0 && left == 0)begin
                if (brake == 0)
                        next_state = OFF;
                else
                        next_state = ON;
        end

        //Handle turn-signal requests
        //Write your own version of code for handling the turn signals

end

endmodule
```