

//Tbird_fsm.sv source code

```
typedef enum logic [2:0] {  
    IDLE = 3'b000,  
    L1  = 3'b001,  
    L2  = 3'b011,  
    L3  = 3'b010,  
    R1  = 3'b101,  
    R2  = 3'b111,  
    R3  = 3'b110,  
    LR3 = 3'b100  
} t_tbird_lights_state;
```

```
module tbird_fsm (  
    input logic    clk,  
    input logic    rst_b,  
    input logic    left,  
    input logic    right,  
    input logic    haz,  
    output logic [2:0] l_lights,  
    output logic [2:0] r_lights  
);  
    t_tbird_lights_state state;
```

```
always_ff @(posedge clk or negedge rst_b) begin  
    if (~rst_b)  
        state <= IDLE;  
    else  
        case (state)  
            IDLE: begin  
                if ( (left & right) | haz)  
                    state <= L1;  
                else if (left)  
                    state <= LR3;  
                else if (right)  
                    state <= R1;  
                else  
                    state <= IDLE;  
            end  
            L1: state <= haz? LR3 : L2;  
            L2: state <= haz? LR3 : L3;  
            L3: state <= IDLE;  
            R1: state <= haz? LR3 : R2;
```

```
    R2: state <= haz? LR3 : R3;
    R3: state <= IDLE;
    LR3: state <= IDLE;
endcase
end
```

```
always_comb begin
  case (state)
    IDLE: begin
      l_lights = 3'b000;
      r_lights = 3'b000;
    end
    L1: begin
      l_lights = 3'b001;
      r_lights = 3'b000;
    end
    L2: begin
      l_lights = 3'b011;
      r_lights = 3'b000;
    end

    L3: begin
      l_lights = 3'b111;
      r_lights = 3'b000;

    end
    R1: begin
      l_lights = 3'b000;
      r_lights = 3'b001;
    end
    R2: begin
      l_lights = 3'b000;
      r_lights = 3'b011;
    end
    R3: begin
      l_lights = 3'b000;
      r_lights = 3'b111;
    end
    LR3: begin
      l_lights = 3'b111;
      r_lights = 3'b111;
    end
  endcase
end
```

```
endmodule
```

```
//testbench_hw8 source code
```

```
module testbench_hw8 ();
```

```
    logic clk, rst_b, left, right, haz;
```

```
    logic [2:0] l_lights, r_lights;
```

```
    tbird_fsm UUT (  
.clk(clk),  
.rst_b(rst_b),  
.left(left),  
.right(right),  
.haz(haz),  
.l_lights(l_lights),  
.r_lights(r_lights)  
    );
```

```
    initial begin
```

```
        clk  = 1'b0;
```

```
        rst_b = 1'b1;
```

```
        left  = 1'b0;
```

```
        right = 1'b0;
```

```
        haz   = 1'b0;
```

```
        forever #5 clk = ~clk;
```

```
    end
```

```
    initial begin
```

```
        #10;
```

```
        rst_b = 1'b0;
```

```
        #20;
```

```
        rst_b = 1'b1;
```

```
        #20;
```

```
        right = 1'b1;
```

```
        #40;
```

```
        right = 1'b0;
```

```
        left  = 1'b1;
```

```
        #40;
```

```
        left  = 1'b0;
```

```
        right = 1'b1;
```

```
        #10;
```

```
    right = 1'b0;
    haz  = 1'b1;
    #10;
    haz  = 1'b0;
    #10;
    right = 1'b1;
    #20;
    right = 1'b0;
    haz  = 1'b1;
    #10;
    haz  = 1'b0;
    #10;
    left = 1'b1;
    #10;
    left = 1'b0;
    haz  = 1'b1;
    #10;
    haz  = 1'b0;
    #10;
    left = 1'b1;
    #20;
    left = 1'b0;
    haz  = 1'b1;
    #10;
    haz  = 1'b0;
    #10;
    $finish();
end
endmodule
```

Tagged Waveform Screenshot:

