Traffic light controller

RTL code

```
module traffic_light_controller(
    input clk,
    input reset,
    output reg ns_red, ns_yellow, ns_green,
    output reg ew_red, ew_yellow, ew_green
);
    reg [2:0] state;
    reg [3:0] count;
    parameter S0 = 3'd0,  // NS Green, EW Red    (5s)
              S1 = 3'd1,  // NS Yellow, EW Red   (1s)
              S2 = 3'd2,  // NS Red, EW Red      (1s)
              S3 = 3'd3,  // NS Red, EW Green    (5s)
              S4 = 3'd4,  // NS Red, EW Yellow   (1s)
              S5 = 3'd5;  // NS Red, EW Red      (1s)
    always @(posedge clk or posedge reset) begin
        if (reset) begin
            state <= S0;
            count <= 0;
        end else begin
            count <= count + 1;
            case (state)
                S0: if (count == 5) begin state <= S1; count <= 0; end
                S1: if (count == 1) begin state <= S2; count <= 0; end
                S2: if (count == 1) begin state <= S3; count <= 0; end
                S3: if (count == 5) begin state <= S4; count <= 0; end
                S4: if (count == 1) begin state <= S5; count <= 0; end
                S5: if (count == 1) begin state <= S0; count <= 0; end
                default: begin state <= S0; count <= 0; end
```

```
            endcase
        end
    end
    always @(*) begin
        // Default all lights off
        ns_red = 0; ns_yellow = 0; ns_green = 0;
        ew_red = 0; ew_yellow = 0; ew_green = 0;
        case (state)
            S0: begin ns_green = 1; ew_red = 1; end
            S1: begin ns_yellow = 1; ew_red = 1; end
            S2: begin ns_red = 1; ew_red = 1; end
            S3: begin ns_red = 1; ew_green = 1; end
            S4: begin ns_red = 1; ew_yellow = 1; end
            S5: begin ns_red = 1; ew_red = 1; end
        endcase
    end
endmodule
```

Test Bench

```
`timescale 1ns / 1ps
module traffic_light_tb;
    reg clk;
    reg reset;
    wire ns_red, ns_yellow, ns_green;
    wire ew_red, ew_yellow, ew_green;
    traffic_light_controller uut (
        .clk(clk),
        .reset(reset),
        .ns_red(ns_red), .ns_yellow(ns_yellow), .ns_green(ns_green),
        .ew_red(ew_red), .ew_yellow(ew_yellow), .ew_green(ew_green)
```

```
    );
    // Clock generator (period = 10ns for simulation)
    always #5 clk = ~clk;
    initial begin
        clk = 0;
        reset = 1;
        #20;
        reset = 0;
        // Run long enough to cover full state cycle multiple times
        #300;
        $finish;
    end
    initial begin
        $display("Time\tState\tNS (R Y G)\tEW (R Y G)");
        $monitor("%0t\t\t%b %b %b\t\t%b %b %b",
            $time, ns_red, ns_yellow, ns_green,
                ew_red, ew_yellow, ew_green);
    end

endmodule
```

Output Waveforms