```
`define Y 2  // Yellow duration
`define G 5  // Green duration

module TLC(x, clk, clr, hg, sr);
    input x, clk, clr;
    output reg [1:0] hg, sr;
    reg [1:0] state, n_state;

    parameter S0 = 2'd0, S1 = 2'd1, S2 = 2'd2, S3 = 2'd3;

    // State register
    always @(posedge clk) begin
        if (clr)
            state <= S0;
        else
            state <= n_state;
    end

    // Output logic based on current state
    always @(state or x) begin
        case (state)
            S0: begin
                hg = 2'b10;   // Green
                sr = 2'b00;   // Red
            end
            S1: begin
                hg = 2'b01;   // Yellow
                sr = 2'b00;   // Red
            end
            S2: begin
                hg = 2'b00;   // Red
                sr = 2'b10;   // Green
            end
            S3: begin
                hg = 2'b00;   // Red
                sr = 2'b01;   // Yellow
            end
            default: begin
                hg = 2'b00;
                sr = 2'b00;
            end
        endcase
    end

    // Next state logic with delays
    always @(state or x) begin
        case (state)
            S0: begin
                if (x)
                    n_state = S1;
                else
                    n_state = S0;
            end
```

```
            S1: begin
                repeat (`Y) @(posedge clk);
                n_state = S2;
            end

            S2: begin
                if (x)
                    n_state = S2;
                else
                    n_state = S3;
            end

            S3: begin
                repeat (`G) @(posedge clk);
                n_state = S0;
            end

            default: n_state = S0;
        endcase
    end
endmodule
```