Traffic Light Controller

Ayush Yadav

1 Introduction

This document describes the implementation of a traffic light controller using a finite state machine (FSM) in Verilog. The controller manages the traffic lights for a North-South (NS) and East-West (EW) intersection, cycling through different states to control the traffic flow.

2 Verilog Code

Below is the Verilog code for the traffic light controller:

```
module traffic_light_controller(
    input clk,              // Clock input
    input reset,            // Reset input
    output reg [2:0] light_NS, // North-South traffic light (3 bits
    : Red, Yellow, Green)
    output reg [2:0] light_EW  // East-West traffic light (3 bits:
    Red, Yellow, Green)
);

    // State encoding
    typedef enum reg [1:0] {
        NS_GREEN_EW_RED = 2'b00,
        NS_YELLOW_EW_RED = 2'b01,
        NS_RED_EW_GREEN = 2'b10,
        NS_RED_EW_YELLOW = 2'b11
    } state_t;

    state_t state, next_state;
    reg [31:0] counter; // Counter for timing

    // State transition and output logic
    always @(posedge clk or posedge reset) begin
        if (reset) begin
            state <= NS_GREEN_EW_RED;
            counter <= 0;
        end else begin
            counter <= counter + 1;
            if (counter == 100) begin // Change state every 100
    clock cycles
                state <= next_state;
                counter <= 0;
            end
```

```
        end
    end

    // Next state logic
    always @(*) begin
        case (state)
            NS_GREEN_EW_RED: begin
                light_NS = 3'b001; // Green
                light_EW = 3'b100; // Red
                next_state = NS_YELLOW_EW_RED;
            end
            NS_YELLOW_EW_RED: begin
                light_NS = 3'b010; // Yellow
                light_EW = 3'b100; // Red
                next_state = NS_RED_EW_GREEN;
            end
            NS_RED_EW_GREEN: begin
                light_NS = 3'b100; // Red
                light_EW = 3'b001; // Green
                next_state = NS_RED_EW_YELLOW;
            end
            NS_RED_EW_YELLOW: begin
                light_NS = 3'b100; // Red
                light_EW = 3'b010; // Yellow
                next_state = NS_GREEN_EW_RED;
            end
            default: begin
                light_NS = 3'b100; // Red
                light_EW = 3'b100; // Red
                next_state = NS_GREEN_EW_RED;
            end
        endcase
    end
endmodule
```

Listing 1: Traffic Light Controller

3 Explanation

3.1 Module Declaration

The module `traffic_light_controller` has two inputs: `clk` (clock signal) and `reset` (reset signal). It also has two 3-bit output registers: `light_NS` (North-South traffic light) and `light_EW` (East-West traffic light).

3.2 State Encoding

The state machine uses an enumerated type `state_t` with four states:

- `NS_GREEN_EW_RED`: North-South green, East-West red.
- `NS_YELLOW_EW_RED`: North-South yellow, East-West red.
- `NS_RED_EW_GREEN`: North-South red, East-West green.

- `NS_RED_EW_YELLOW`: North-South red, East-West yellow.

3.3 State Transition and Output Logic

The state transition and output logic are handled in an `always` block sensitive to the positive edge of the clock and the reset signal. When the reset signal is asserted, the state is set to `NS_GREEN_EW_RED` and the counter is reset to 0. Otherwise, the counter increments on each clock cycle. When the counter reaches 100, the state transitions to the next state, and the counter is reset to 0.

3.4 Next State Logic

The next state logic determines the `next_state` and sets the output lights based on the current state using a `case` statement:

- `NS_GREEN_EW_RED`: Sets the North-South light to green and the East-West light to red. Transitions to `NS_YELLOW_EW_RED`.
- `NS_YELLOW_EW_RED`: Sets the North-South light to yellow and the East-West light to red. Transitions to `NS_RED_EW_GREEN`.
- `NS_RED_EW_GREEN`: Sets the North-South light to red and the East-West light to green. Transitions to `NS_RED_EW_YELLOW`.
- `NS_RED_EW_YELLOW`: Sets the North-South light to red and the East-West light to yellow. Transitions to `NS_GREEN_EW_RED`.
- `default`: Sets both lights to red and transitions to `NS_GREEN_EW_RED`.

3.5 Summary

The module implements a simple traffic light controller using a finite state machine. It cycles through four states, controlling the traffic lights for a North-South and East-West intersection. The state transitions occur every 100 clock cycles, simulating the timing of traffic light changes. The lights are represented using 3-bit values where each bit corresponds to Red, Yellow, and Green lights respectively.