Solutions PDF Generated from: solutions-openai-generated/labs/lab-04-solutions-set-01.json

Question A

Design a Finite State Machine to mimic the performance of a 1965 Ford Thunderbirds taillights. There are three inputs left, right, and hazard. If left is the input, the left set of LEDs should progressively light up and then go dark, repeating until the signal is no longer applied. Similar logic will apply to the right LEDs. If the hazard input switch is on, all of the lights should blink on and off until the input is changed. Write out a Next State Logic and Output Logic circuit. The three inputs will be the left, right, and hazard switches. The outputs will be the 6 or 8 LEDs that you choose to use for displaying the patterns.

```
Next State Logic

Left:
If left switch = 0   next state = off
If left switch = 1   next state = left_led1

Right:
If right switch = 0  next state = off
If right switch = 1  next state = right_led1

Hazard:
If hazard switch = 0 next state = off
If hazard switch = 1 next state = led_all

Output Logic

off:
 LEDs = 0

left_led1:
 LEDs = Left_led1

right_led1:
 LEDs = Right_led1

led_all:
 If cycle = even:
 LEDs= left_led1, left_led2, left_led3, right_led1, right_led2, right_led3
 If cycle = odd:
 LEDs= 0
```

Question B

Write Verilog code to realize the Finite State Machine for the taillights operation. There are three inputs left, right, and hazard. If left is the input, the left set of LEDs should progressively light up and then go dark, repeating until the signal is no longer applied. Similar logic will apply to the right LEDs. If the hazard input switch is on, all of the lights should blink on and off until the input is changed.

```
module DualTailLights_FSM (Left, Right, Hazard, LeftLight, RightLight);

  input Left, Right, Hazard;
  output LeftLight, RightLight;

  // Declare states
  reg [1:0] state;

  // Assign states
  parameter
    STATE_INIT = 2'b00,
    STATE_LEFT_SEQ1 = 2'b01,
    STATE_LEFT_SEQ2 = 2'b10,
    STATE_HAZARD = 2'b11;

  // Start of FSM
  always @* begin
    case(state)
      STATE_INIT: begin
        LeftLight = 1'b0;
        RightLight = 1'b0;
        if(Left == 1'b1)
          state <= STATE_LEFT_SEQ1;
        elseif(Right == 1'b1)
          state <= STATE_LEFT_SEQ2;
        elseif(Hazard == 1'b1)
          state <= STATE_HAZARD;
        else
          state <= STATE_INIT;
      end

      STATE_LEFT_SEQ1: begin
        LeftLight = 1'b1;
        RightLight = 1'b0;
        if(Left == 1'b1)
          state <= STATE_LEFT_SEQ2;
        elseif(Hazard == 1'b1)
          state <= STATE_HAZARD;
        else
          state <= STATE_INIT;
      end

      STATE_LEFT_SEQ2: begin
        LeftLight = 1'b0;
        RightLight = 1'b0;
        if(Left == 1'b1)
          state <= STATE_LEFT_SEQ1;
        elseif(Hazard == 1'b1)
          state <= STATE_HAZARD;
        else
          state <= STATE_INIT;
      end
      STATE_HAZARD: begin
        LeftLight = !LeftLight;
        RightLight = !RightLight;
        if(Hazard == 1'b1)
          state <= STATE_HAZARD;
        else
          state <= STATE_INIT;
      end

      default:
        // default state
        LeftLight = 1'b0;
        RightLight = 1'b0;
        state <= STATE_INIT;
    endcase
  end

endmodule
```

Execution Time

0:00:25.940905

OpenAI Parameters

Model: text-davinci-003, Max. Tokens: 1024, Temperature: 1, N: 1