Solutions PDF Generated from: solutions-openai-generated/labs/lab-04-solutions-set-03.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.

```
Inputs: Left, Right, and Hazard Switches
Outputs: 6 or 8 LEDs (Chosen)

Next State Logic:

   Q0 (Starting State): All LEDs are off
      Q1 (Left On): Left LEDs progressively light up, then go dark, repeating until left is no
          longer applied
      Q2 (Right On): Right LEDs progressively light up, then go dark, repeating until right is no
          longer applied
      Q3 (Hazard On): All LEDs blink on and off until the Hazard Switch is turned off.

Output Logic:

Q0 (Starting State): All LEDs are off

Q1 (Left On): Left LEDs will progressively light up, then go dark.

Q2 (Right On): Right LEDs will progressively light up, then go dark.

Q3 (Hazard On): All LEDs blink on and off alternately in a pattern.
```

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.

```
//Left and Right logic signals
reg [1:0] LEFT_EN, RIGHT_EN;

// Left, Right, and Hazard control signals
wire [2:0] CTRL;
```

```
// Finite State Machine
parameter S0 = 2'b00,
         S1 = 2'b01,
         S2 = 2'b10,
         S3 = 2'b11;

// Internal Control logic
wire [2:0] in_ctr;
assign in_ctr[1] = LEFT_EN[1] | RIGHT_EN[0];
assign in_ctr[2] = LEFT_EN[0] | RIGHT_EN[1];
assign in_ctr[0] = CTRL[2];

// FSM output
reg [3:0] LEDA, LEDB;

always @ (in_ctr)
begin
  case (in_ctr)
    S0: {LEDA,LEDB} = 4'b0000;
    S1: {LEDA,LEDB} = 4'b1111;
    S2: {LEDA,LEDB} = 4'b1100;
    S3: {LEDA,LEDB} = 4'b0011;
  endcase
end

// FSM inputs
reg [2:0] next_state;

// FSM logic
always@ (in_ctr or LEFT_EN or RIGHT_EN)
begin
  case (in_ctr)
    S0: next_state = {LEFT_EN[0], RIGHT_EN[0], CTRL[2]};
    S1: if (LEFT_EN[0] == 1)
          next_state = S2;
        else if (RIGHT_EN[0] == 1)
          next_state = S3;
        else
          next_state = S0;
    S2: if (LEFT_EN[1] == 1)
          next_state = S3;
        else
          next_state = S0;
    S3: if (RIGHT_EN[1] == 1)
          next_state = S2;
        else
          next_state = S0;
  endcase
end

// FSM control signals
always@ (next_state)
begin
  CTRL = next_state;
end
```

Execution Time

```
0:00:22.005618
```

OpenAI Parameters

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