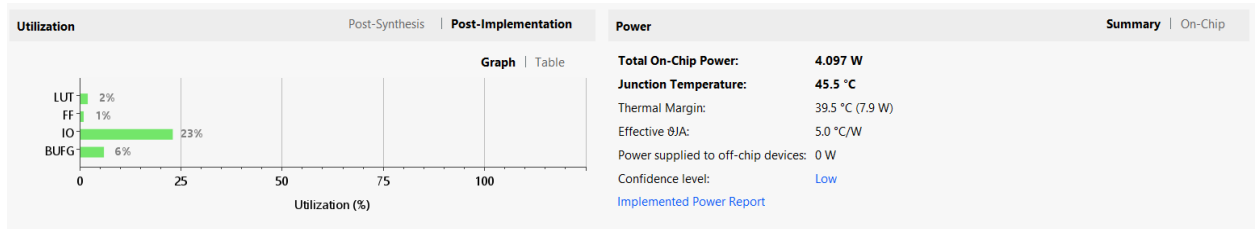


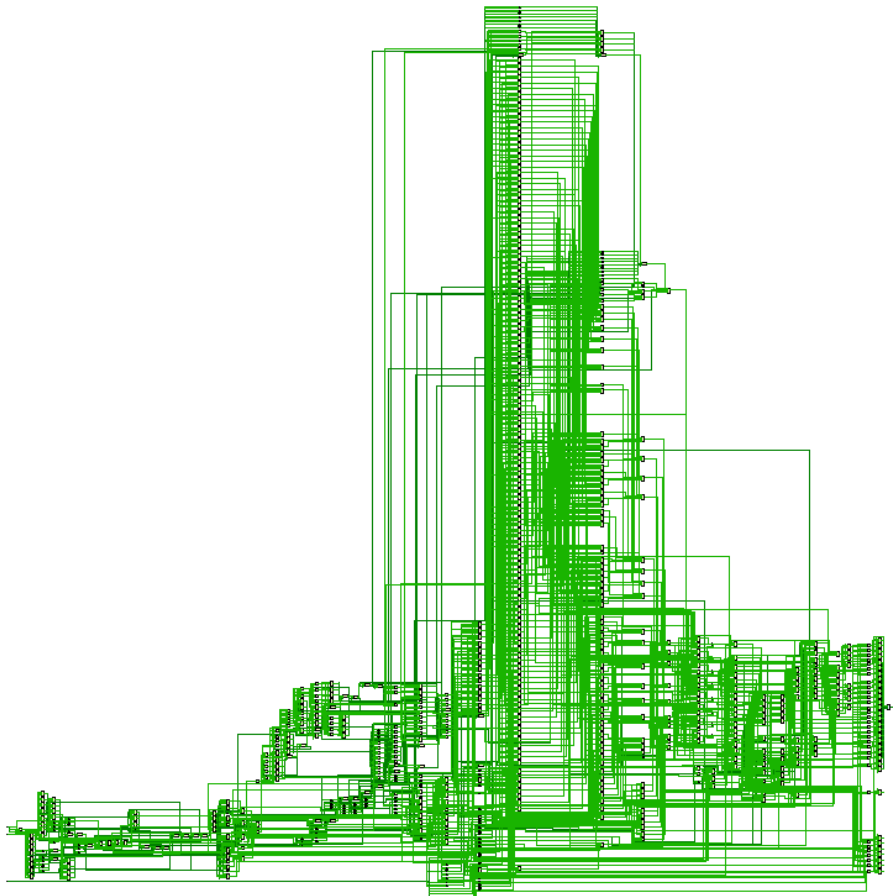
Vincent Han

## Lab 6 Notebook

1. In this lab we created a moving text for the board. It stores a string then displays it across.
2. I had attempted to test different strings, however my testbench and everything else was messed up.



3.



## Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): inf	Worst Hold Slack (WHS): inf	Worst Pulse Width Slack (WPWS): NA
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): NA
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: NA
Total Number of Endpoints: 1049	Total Number of Endpoints: 1049	Total Number of Endpoints: NA

**There are no user specified timing constraints.**

- 5.
6. The design goals were not met, the storing and moving of the text did not work.
7. The cases, as well as the timing reports.

Source Code:

### movingText.v

```
`timescale 1ns / 1ps

module movingText(
    input clk,
    input [2:0] btn,
    input [7:0] sw,
    output [7:0] disp,
    output [3:0] anode
);

    integer charCount = 0;
    integer i;
    integer count = 0;
    reg [7:0] message[0:20];
    reg [7:0] dig1;
    reg [7:0] dig2;
    reg [7:0] dig3;
    reg [7:0] dig4;
    reg [7:0] dig_temp1;
    reg [7:0] dig_temp2;
    reg [7:0] dig_temp3;
    reg [7:0] dig_temp4;
    reg stored = 0;
    wire cout;

    clock_divider C(cout, clk);
    disp D(clk, dig1, dig2, dig3, dig4, disp, anode);

    initial begin
        dig1 <= 8'b11111111;
        dig2 <= 8'b11111111;
        dig3 <= 8'b11111111;
        dig4 <= 8'b11111111;
        dig_temp1 <= 8'b11111111;
        dig_temp2 <= 8'b11111111;
        dig_temp3 <= 8'b11111111;
        dig_temp4 <= 8'b11111111;
        for(i = 0; i <= 20; i=i+1) begin
            message[i] <= 8'b11111111;
        end
    end

    always @ (posedge clk) begin
        case(btn)
            3'b110: begin
```

```

        stored <= 0;
        dig1 <= 8'b10001111;           // L
        dig2 <= 8'b11111111;
        dig3 <= 8'b11111111;
        dig4 <= ~sw;
    end

    3'b111: begin
        dig1 <= 8'b01000111;           // o
        dig2 <= 8'b11111111;
        dig3 <= 8'b11111111;
        dig4 <= ~sw;
        if(charCount < 20 && stored == 0) begin
            message[charCount] <= ~sw;
            charCount <= charCount + 1;
            stored <= 1;
        end
    end

    3'b010: begin
        stored <= 0;
        charCount <= 0;
        dig4 <= 8'b11111111;
        dig3 <= 8'b11111111;
        dig2 <= 8'b11111111;
        dig1 <= 8'b11111111;
        for(i = 0; i <= 19; i=i+1) begin
            message[i] <= 8'b11111111;
        end
    end

    default: begin
        stored <= 0;
        dig1 <= dig_temp1;
        dig2 <= dig_temp2;
        dig3 <= dig_temp3;
        dig4 <= dig_temp4;
    end
endcase
end

always @ (posedge cout) begin
    if (btn == 3'b010) begin
        count <= 0;
        dig_temp4 <= 8'b11111111;
        dig_temp3 <= 8'b11111111;
        dig_temp2 <= 8'b11111111;
        dig_temp1 <= 8'b11111111;
    end

    if (charCount == 0) begin
        dig_temp4 <= 8'b11111111;
        dig_temp3 <= 8'b11111111;
        dig_temp2 <= 8'b11111111;
        dig_temp1 <= 8'b11111111;
    end
    else if (charCount == 1) begin
        dig_temp1 <= 8'b11111111;
        dig_temp2 <= 8'b11111111;
        dig_temp3 <= 8'b11111111;
        dig_temp4 <= message[0];
    end
    else if (charCount == 2) begin
        dig_temp1 <= 8'b11111111;
        dig_temp2 <= 8'b11111111;
        dig_temp4 <= message[1];
        dig_temp3 <= message[0];
    end
end

```

```

end
else if (charCount == 3) begin
    dig_temp1 <= 8'b11111111;
    dig_temp4 <= message[2];
    dig_temp3 <= message[1];
    dig_temp2 <= message[0];
end
else if (charCount == 4) begin
    dig_temp4 <= message[3];
    dig_temp3 <= message[2];
    dig_temp2 <= message[1];
    dig_temp1 <= message[0];
end
else if (charCount > 4 && count == 0) begin
    dig_temp4 <= message[3];
    dig_temp3 <= message[2];
    dig_temp2 <= message[1];
    dig_temp1 <= message[0];
    count <= 1;
end
else begin
    if (count <= (charCount - 3)) begin
        dig_temp4 <= message[count+3];
        dig_temp3 <= dig_temp4;
        dig_temp2 <= dig_temp3;
        dig_temp1 <= dig_temp2;
        count <= count + 1;
    end
    else begin
        dig_temp4 <= message[0];
        dig_temp3 <= dig_temp4;
        dig_temp2 <= dig_temp3;
        dig_temp1 <= dig_temp2;
        count <= -2;
    end
end
end
endmodule

```

## Disp.v

```

module disp(
    input clk,
    input [7:0] dig1,
    input [7:0] dig2,
    input [7:0] dig3,
    input [7:0] dig4,
    output reg [7:0] disp,
    output reg [3:0] anode
);

localparam N = 19;
reg [N-1:0] count = 0;
//reg [6:0] temp1 = 7'b1111111;
//reg [6:0] temp2 = 7'b1111111;

always @ (posedge clk)
begin
    count <= count + 1;
end

always @ (*)
begin
    case(count[N-1:N-2])
        2'b00: begin
            disp = dig4;

```

```
        anode = 4'b1110;
    end
    2'b01: begin
        disp = dig3;
        anode = 4'b1101;
    end
    2'b10: begin
        disp = dig2;
        anode = 4'b1011;
    end
    2'b11: begin
        disp = dig1;
        anode = 4'b0111;
    end
endcase
end
endmodule
```