ECE 3300 – Lab 4 Report
Switch to 7 Segment Display
Team Name: Group V
Date: July 11, 2025

Group Members

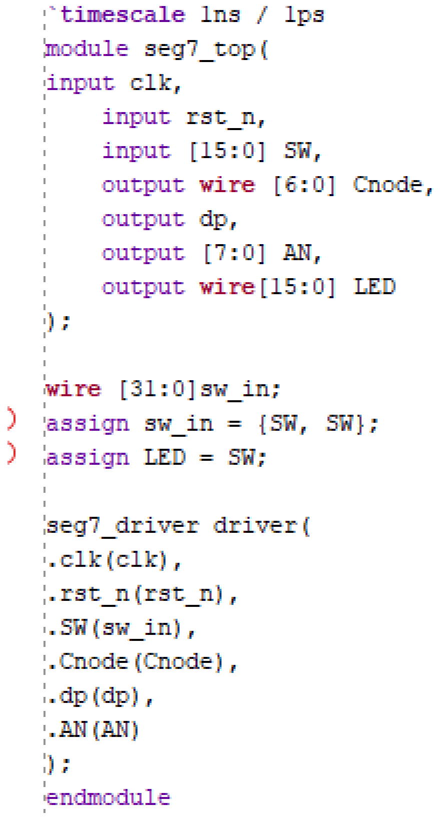
Nathan Marlow, Khristian Chan

1. Objective

The objective of this lab was to implement a 7 segment display interface that utilized 16 switches as input, and a pair of 4 digit 7 segment displays (8 total). Every 4 switches (bits) will correspond to two 7 segment displays of the 8 (1 and 5, 2 and 6, 3 and 7, 4 and 8). Given a 7 segment driver module, we were tasked to create a top module and create a testbench.

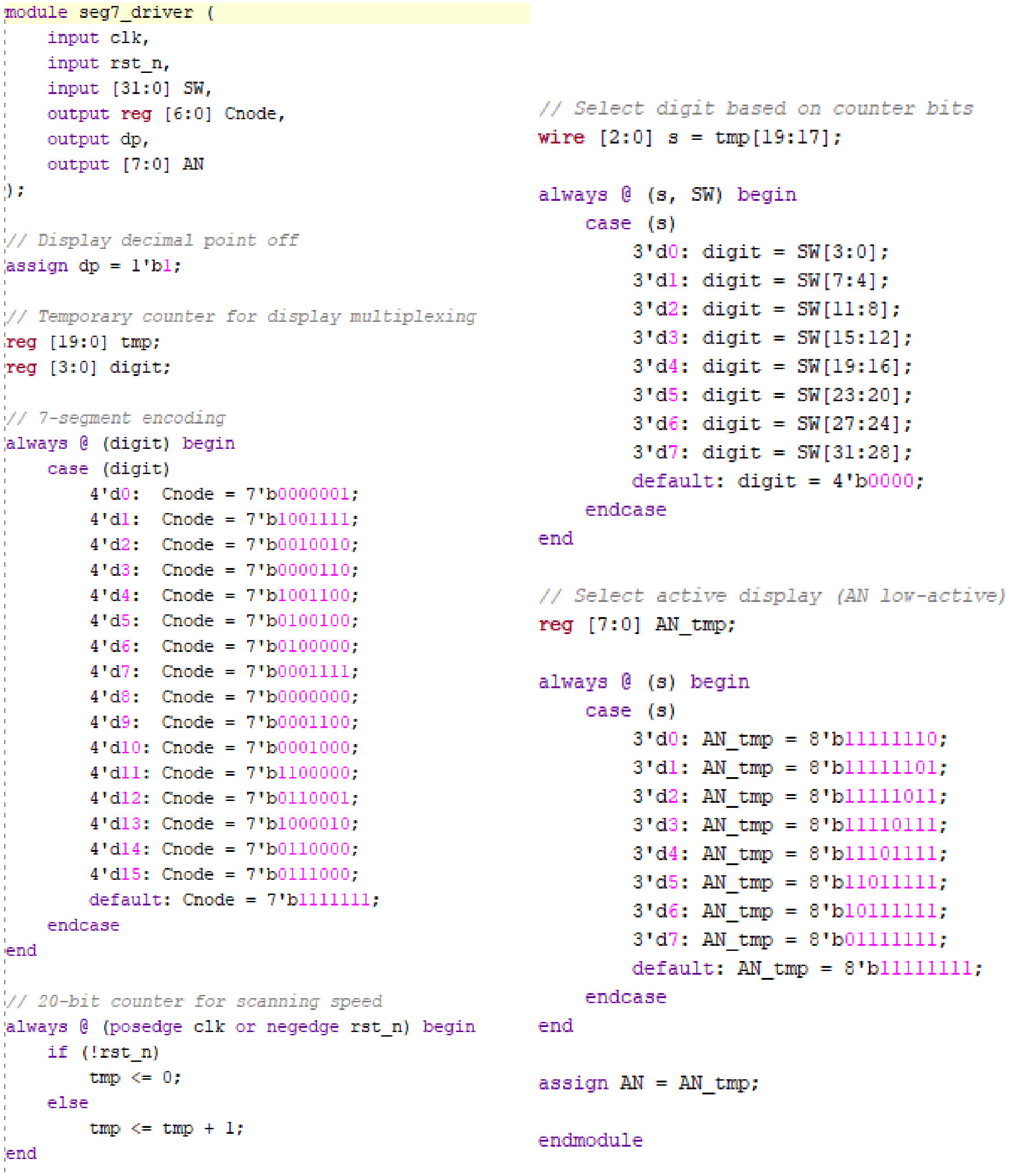
2. Verilog Code

Top Module:



```
`timescale 1ns / 1ps
module seg7_top(
input clk,
    input rst_n,
    input [15:0] SW,
    output wire [6:0] Cnode,
    output dp,
    output [7:0] AN,
    output wire[15:0] LED
);

wire [31:0]sw_in;
assign sw_in = {SW, SW};
assign LED = SW;

seg7_driver driver(
.clk(clk),
.rst_n(rst_n),
.SW(sw_in),
.Cnode(Cnode),
.dp(dp),
.AN(AN)
);
endmodule
```

7Seg Driver:

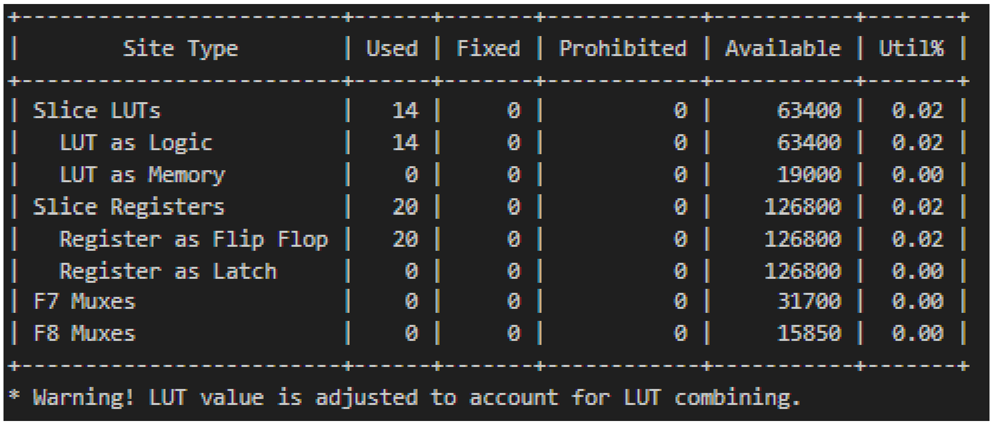


```
module seg7_driver (
    input clk,
    input rst_n,
    input [31:0] SW,
    output reg [6:0] Cnode,
    output dp,
    output [7:0] AN
);

// Display decimal point off
assign dp = 1'b1;

// Temporary counter for display multiplexing
reg [19:0] tmp;
reg [3:0] digit;

// 7-segment encoding
always @ (digit) begin
    case (digit)
        4'd0:  Cnode = 7'b0000001;
        4'd1:  Cnode = 7'b1001111;
        4'd2:  Cnode = 7'b0010010;
        4'd3:  Cnode = 7'b0000110;
        4'd4:  Cnode = 7'b1001100;
        4'd5:  Cnode = 7'b0100100;
        4'd6:  Cnode = 7'b0100000;
        4'd7:  Cnode = 7'b0001111;
        4'd8:  Cnode = 7'b0000000;
        4'd9:  Cnode = 7'b0001100;
        4'd10: Cnode = 7'b0001000;
        4'd11: Cnode = 7'b1100000;
        4'd12: Cnode = 7'b0110001;
        4'd13: Cnode = 7'b1000010;
        4'd14: Cnode = 7'b0110000;
        4'd15: Cnode = 7'b0111000;
        default: Cnode = 7'b1111111;
    endcase
end

// 20-bit counter for scanning speed
always @ (posedge clk or negedge rst_n) begin
    if (!rst_n)
        tmp <= 0;
    else
        tmp <= tmp + 1;
end

// Select digit based on counter bits
wire [2:0] s = tmp[19:17];

always @ (s, SW) begin
    case (s)
        3'd0: digit = SW[3:0];
        3'd1: digit = SW[7:4];
        3'd2: digit = SW[11:8];
        3'd3: digit = SW[15:12];
        3'd4: digit = SW[19:16];
        3'd5: digit = SW[23:20];
        3'd6: digit = SW[27:24];
        3'd7: digit = SW[31:28];
        default: digit = 4'b0000;
    endcase
end

// Select active display (AN low-active)
reg [7:0] AN_tmp;

always @ (s) begin
    case (s)
        3'd0: AN_tmp = 8'b11111110;
        3'd1: AN_tmp = 8'b11111101;
        3'd2: AN_tmp = 8'b11111011;
        3'd3: AN_tmp = 8'b11110111;
        3'd4: AN_tmp = 8'b11101111;
        3'd5: AN_tmp = 8'b11011111;
        3'd6: AN_tmp = 8'b10111111;
        3'd7: AN_tmp = 8'b01111111;
        default: AN_tmp = 8'b11111111;
    endcase
end

assign AN = AN_tmp;

endmodule
```

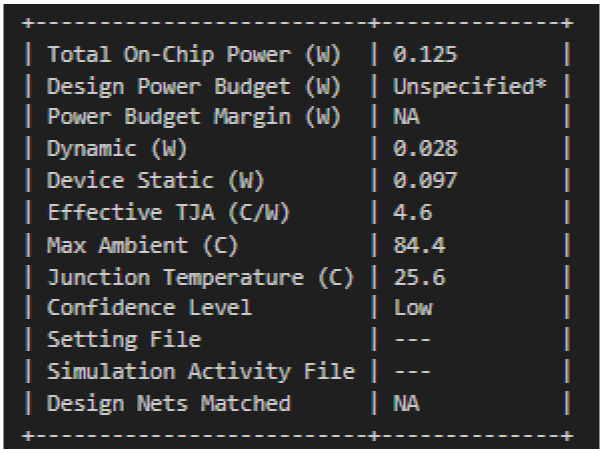
We map our top module switches twice to our 7 segment driver module. We do this so we can give a 32 bit input to our 7 segment driver. Additionally, we added the LED functionality.

3. Implementation Results

After synthesizing and simulating the design in Vivado, the following results were observed:



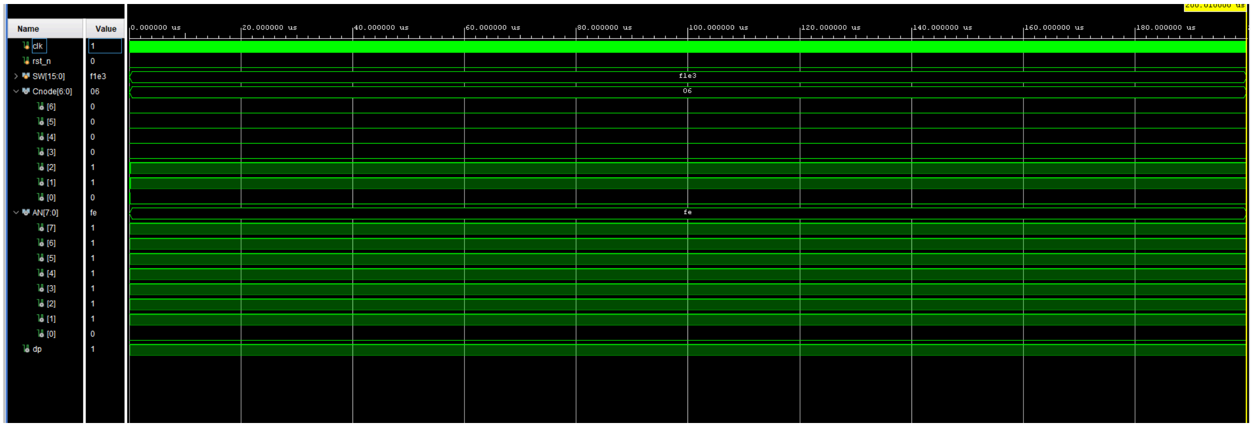
```
+-------------------------+------+-------+------------+-----------+-------+
|        Site Type        | Used | Fixed | Prohibited | Available | Util% |
+-------------------------+------+-------+------------+-----------+-------+
| Slice LUTs              |   14 |     0 |          0 |     63400 |  0.02 |
|   LUT as Logic          |   14 |     0 |          0 |     63400 |  0.02 |
|   LUT as Memory         |    0 |     0 |          0 |     19000 |  0.00 |
| Slice Registers         |   20 |     0 |          0 |    126800 |  0.02 |
|   Register as Flip Flop |   20 |     0 |          0 |    126800 |  0.02 |
|   Register as Latch     |    0 |     0 |          0 |    126800 |  0.00 |
| F7 Muxes                |    0 |     0 |          0 |     31700 |  0.00 |
| F8 Muxes                |    0 |     0 |          0 |     15850 |  0.00 |
+-------------------------+------+-------+------------+-----------+-------+
* Warning! LUT value is adjusted to account for LUT combining.
```



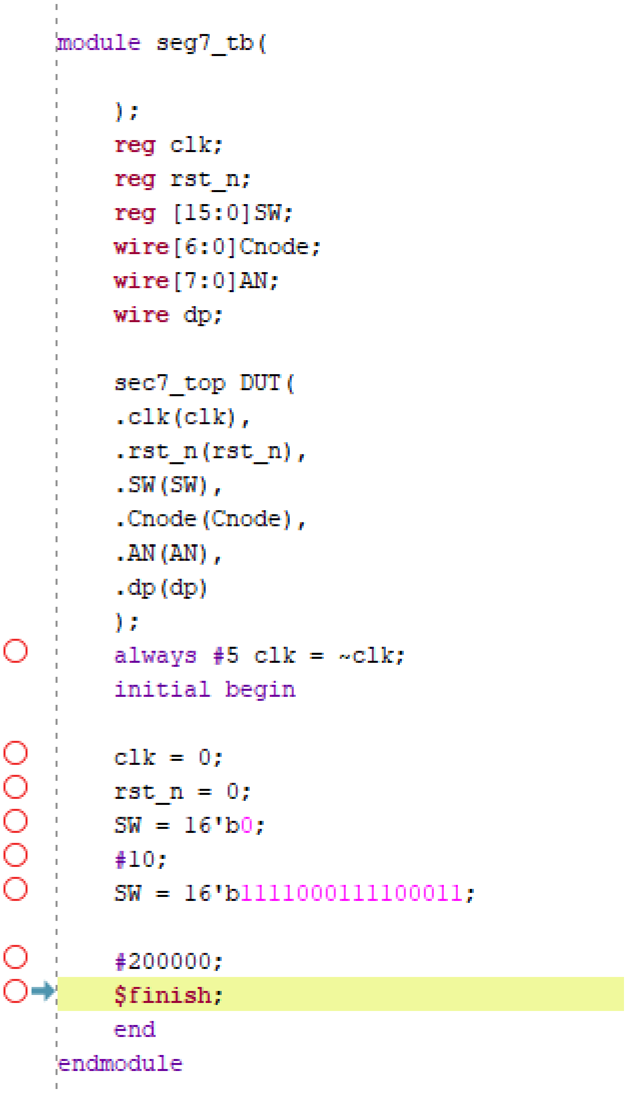
```
+--------------------------+--------------+
| Total On-Chip Power (W)  | 0.125        |
| Design Power Budget (W)  | Unspecified* |
| Power Budget Margin (W)  | NA           |
| Dynamic (W)              | 0.028        |
| Device Static (W)        | 0.097        |
| Effective TJA (C/W)      | 4.6          |
| Max Ambient (C)          | 84.4         |
| Junction Temperature (C) | 25.6         |
| Confidence Level         | Low          |
| Setting File             | ---          |
| Simulation Activity File | ---          |
| Design Nets Matched      | NA           |
+--------------------------+--------------+
```

Simulation Results:

Since the simulation would take a very long time to go through all possible values, we decided to just simulate one switch input. The testbench is able to validate our fpga code.



Name	Value
clk	1
rst_n	0
SW[15:0]	f1e3
Cnode[6:0]	06
[6]	0
[5]	0
[4]	0
[3]	0
[2]	1
[1]	1
[0]	0
AN[7:0]	fe
[7]	1
[6]	1
[5]	1
[4]	1
[3]	1
[2]	1
[1]	1
[0]	0
dp	1



```
module seg7_tb(

    );
    reg clk;
    reg rst_n;
    reg [15:0]SW;
    wire[6:0]Cnode;
    wire[7:0]AN;
    wire dp;

    sec7_top DUT(
    .clk(clk),
    .rst_n(rst_n),
    .SW(SW),
    .Cnode(Cnode),
    .AN(AN),
    .dp(dp)
    );
    always #5 clk = ~clk;
    initial begin

    clk = 0;
    rst_n = 0;
    SW = 16'b0;
    #10;
    SW = 16'b1111000111100011;

    #200000;
    $finish;
    end
endmodule
```

6. Demonstration Video

https://youtu.be/tcdEHrQKYs8

Contributions:

Khristian Chan- 50%: Testbench, Simulation, Report, Demo
Nathan Marlow- 50%: Implementation, XDC, Report