Cal Poly Pomona

College of Engineering

California Polytechnic State University, Pomona

ECE3300L

Experiment #4

GROUP K
Hoang, Dalton - 016062800
Siu, Andy - 016205137

July 10th, 2025

Introduction:

The purpose of this lab was to explore the process of interfacing digital inputs with a seven-segment display. The objective is to implement a switch-to-display system that translates 4-bit binary values from slide switches (SW[15:0]) into hexadecimal digits, displayed on the eight-digit seven-segment display. This involved designing and simulating structural Verilog modules such as decoders and multiplexers, as well as understanding how to control display outputs through multiplexing.

Design:

Topmodule.v: This top-level module instantiates seg7_driver.v and maps 16 input switches into a 32-bit pattern where each 4-bit group is duplicated to control two 7-segment digits.

```
21
22
23  module topmodule(
24      input clk,
25      input rst_n,
26      input [15:0] SW,
27      output [15:0] led,
28      output [6:0] Cnode,
29      output dp,
30      output [7:0] AN
31      );
32
33      assign led = SW;
34
35      wire [31:0] sw = {
36          SW[15:12], // digit 8 (AN[7])
37          SW[11:8],  // digit 7 (AN[6])
38          SW[7:4],   // digit 6 (AN[5])
39          SW[3:0],   // digit 5 (AN[4])
40          SW[15:12], // digit 4 (AN[3])
41          SW[11:8],  // digit 3 (AN[2])
42          SW[7:4],   // digit 2 (AN[1])
43          SW[3:0]    // digit 1 (AN[0])
44      };
45
46      seg7_driver seg7_inst (
47          .clk(clk),
48          .rst_n(~rst_n),
49          .SW(sw),
50          .Cnode(Cnode),
51          .dp(dp),
52          .AN(AN)
53      );
54  endmodule
55
56
```

Seg7_driver.v: This module drives an 8-digit 7-segment display by time-multiplexing the active digit and decoding corresponding 4-bit values from a 32-bit input to show hexadecimal digits, with reset and clock control.

```
22  `timescale 1ns / 1ps
23  module seg7_driver(
24      input clk,
25      input rst_n,
26      input [31:0] SW,
27      output reg [6:0] Cnode,
28      output dp,
29      output [7:0] AN
30      );
31      reg [19:0] tmp;
32      reg [3:0] digit;
33
34      assign dp = 1'b1;
35
36      always@(digit)
37          case(digit)
38              4'd0: Cnode=7'b0000001; 4'd1: Cnode=7'b1001111; 4'd2: Cnode=7'b0010010;
39              4'd3: Cnode=7'b0000110; 4'd4: Cnode=7'b1001100; 4'd5: Cnode=7'b0100100;
40              4'd6: Cnode=7'b0100000; 4'd7: Cnode=7'b0001111; 4'd8: Cnode=7'b0000000;
41              4'd9: Cnode=7'b0001100; 4'd10:Cnode=7'b0001000;4'd11:Cnode=7'b1100000;
42              4'd12:Cnode=7'b0110001;4'd13:Cnode=7'b1000010;4'd14:Cnode=7'b0110000;
43              4'd15:Cnode=7'b0111000;default: Cnode=7'b1111111;
44          endcase
45      always@(posedge clk or negedge rst_n)
46          if(!rst_n) tmp<=0;
47          else tmp<=tmp+1;
48
49      wire [2:0] s = tmp[19:17];
50
51      always@(s, SW)
52          case (s)
53              3'd0:digit=SW[3:0]; 3'd1:digit=SW[7:4];
54              3'd2:digit=SW[11:8]; 3'd3:digit=SW[15:12];
55              3'd4:digit=SW[19:16];3'd5:digit=SW[23:20];
56              3'd6:digit=SW[27:24];3'd7:digit=SW[31:28];
57              default:digit=4'b0000;
```

```
58          endcase
59
60      reg [7:0] AN_tmp;
61      always@(s)
62          case(s)
63              3'd0:AN_tmp=8'b11111110;3'd1:AN_tmp=8'b11111101;
64              3'd2:AN_tmp=8'b11111011;3'd3:AN_tmp=8'b11110111;
65              3'd4:AN_tmp=8'b11101111;3'd5:AN_tmp=8'b11011111;
66              3'd6:AN_tmp=8'b10111111;3'd7:AN_tmp=8'b01111111;
67              default:AN_tmp=8'b11111111;
68          endcase
69
70       assign AN=AN_tmp;
71  endmodule
```

Seg7constraint.xdc: This constraint file assigns specific physical FPGA pins to the logical ports defined in the top-level module. This file makes sure connections are correct between the board hardware (switches, LEDs, button, and 7-segment display) and the synthesized Verilog design.

```
set_property -dict { PACKAGE_PIN J15   IOSTANDARD LVCMOS33 } [get_ports { SW[0] }]; #IO_L24N_T3_RS0_15 Sch=sw[0]
set_property -dict { PACKAGE_PIN L16   IOSTANDARD LVCMOS33 } [get_ports { SW[1] }]; #IO_L3N_T0_DQS_EMCCLK_14 Sch=sw[1]
set_property -dict { PACKAGE_PIN M13   IOSTANDARD LVCMOS33 } [get_ports { SW[2] }]; #IO_L6N_T0_D08_VREF_14 Sch=sw[2]
set_property -dict { PACKAGE_PIN R15   IOSTANDARD LVCMOS33 } [get_ports { SW[3] }]; #IO_L13N_T2_MRCC_14 Sch=sw[3]
set_property -dict { PACKAGE_PIN R17   IOSTANDARD LVCMOS33 } [get_ports { SW[4] }]; #IO_L12N_T1_MRCC_14 Sch=sw[4]
set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports { SW[5] }]; #IO_L7N_T1_D10_14 Sch=sw[5]
set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports { SW[6] }]; #IO_L17N_T2_A13_D29_14 Sch=sw[6]
set_property -dict { PACKAGE_PIN R13   IOSTANDARD LVCMOS33 } [get_ports { SW[7] }]; #IO_L5N_T0_D07_14 Sch=sw[7]
set_property -dict { PACKAGE_PIN T8    IOSTANDARD LVCMOS18 } [get_ports { SW[8] }]; #IO_L24N_T3_34 Sch=sw[8]
set_property -dict { PACKAGE_PIN U8    IOSTANDARD LVCMOS18 } [get_ports { SW[9] }]; #IO_25_34 Sch=sw[9]
set_property -dict { PACKAGE_PIN R16   IOSTANDARD LVCMOS33 } [get_ports { SW[10] }]; #IO_L15P_T2_DQS_RDWR_B_14 Sch=sw[10]
set_property -dict { PACKAGE_PIN T13   IOSTANDARD LVCMOS33 } [get_ports { SW[11] }]; #IO_L23P_T3_A03_D19_14 Sch=sw[11]
set_property -dict { PACKAGE_PIN H6    IOSTANDARD LVCMOS33 } [get_ports { SW[12] }]; #IO_L24P_T3_35 Sch=sw[12]
set_property -dict { PACKAGE_PIN U12   IOSTANDARD LVCMOS33 } [get_ports { SW[13] }]; #IO_L20P_T3_A08_D24_14 Sch=sw[13]
set_property -dict { PACKAGE_PIN U11   IOSTANDARD LVCMOS33 } [get_ports { SW[14] }]; #IO_L19N_T3_A09_D25_VREF_14 Sch=sw[14]
set_property -dict { PACKAGE_PIN V10   IOSTANDARD LVCMOS33 } [get_ports { SW[15] }]; #IO_L21P_T3_DQS_14 Sch=sw[15]


## LEDs

set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports { led[0] }]; #IO_L18P_T2_A24_15 Sch=led[0]
set_property -dict { PACKAGE_PIN K15   IOSTANDARD LVCMOS33 } [get_ports { led[1] }]; #IO_L24P_T3_RS1_15 Sch=led[1]
set_property -dict { PACKAGE_PIN J13   IOSTANDARD LVCMOS33 } [get_ports { led[2] }]; #IO_L17N_T2_A25_15 Sch=led[2]
set_property -dict { PACKAGE_PIN N14   IOSTANDARD LVCMOS33 } [get_ports { led[3] }]; #IO_L8P_T1_D11_14 Sch=led[3]
set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports { led[4] }]; #IO_L7P_T1_D09_14 Sch=led[4]
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports { led[5] }]; #IO_L18N_T2_A11_D27_14 Sch=led[5]
set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports { led[6] }]; #IO_L17P_T2_A14_D30_14 Sch=led[6]
set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports { led[7] }]; #IO_L18P_T2_A12_D28_14 Sch=led[7]
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports { led[8] }]; #IO_L16N_T2_A15_D31_14 Sch=led[8]
set_property -dict { PACKAGE_PIN T15   IOSTANDARD LVCMOS33 } [get_ports { led[9] }]; #IO_L14N_T2_SRCC_14 Sch=led[9]
set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports { led[10] }]; #IO_L22P_T3_A05_D21_14 Sch=led[10]
set_property -dict { PACKAGE_PIN T16   IOSTANDARD LVCMOS33 } [get_ports { led[11] }]; #IO_L15N_T2_DQS_DOUT_CSO_B_14 Sch=led[11]
set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports { led[12] }]; #IO_L16P_T2_CSI_B_14 Sch=led[12]
set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports { led[13] }]; #IO_L22N_T3_A04_D20_14 Sch=led[13]
set_property -dict { PACKAGE_PIN V12   IOSTANDARD LVCMOS33 } [get_ports { led[14] }]; #IO_L20N_T3_A07_D23_14 Sch=led[14]
set_property -dict { PACKAGE_PIN V11   IOSTANDARD LVCMOS33 } [get_ports { led[15] }]; #IO_L21N_T3_DQS_A06_D22_14 Sch=led[15]
```

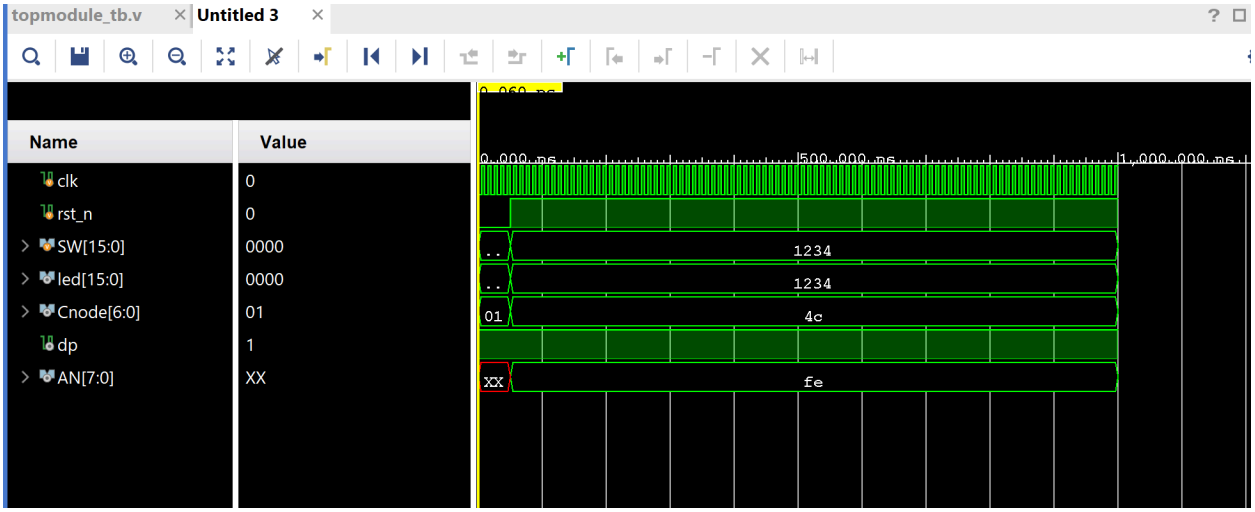
```
##7 segment display

set_property -dict { PACKAGE_PIN T10   IOSTANDARD LVCMOS33 } [get_ports { Cnode[6] }]; #IO_L24N_T3_A00_D16_14 Sch=ca
set_property -dict { PACKAGE_PIN R10   IOSTANDARD LVCMOS33 } [get_ports { Cnode[5] }]; #IO_25_14 Sch=cb
set_property -dict { PACKAGE_PIN K16   IOSTANDARD LVCMOS33 } [get_ports { Cnode[4] }]; #IO_25_15 Sch=cc
set_property -dict { PACKAGE_PIN K13   IOSTANDARD LVCMOS33 } [get_ports { Cnode[3] }]; #IO_L17P_T2_A26_15 Sch=cd
set_property -dict { PACKAGE_PIN P15   IOSTANDARD LVCMOS33 } [get_ports { Cnode[2] }]; #IO_L13P_T2_MRCC_14 Sch=ce
set_property -dict { PACKAGE_PIN T11   IOSTANDARD LVCMOS33 } [get_ports { Cnode[1] }]; #IO_L19P_T3_A10_D26_14 Sch=cf
set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports { Cnode[0] }]; #IO_L4P_T0_D04_14 Sch=cg

set_property -dict { PACKAGE_PIN H15   IOSTANDARD LVCMOS33 } [get_ports { dp }]; #IO_L19N_T3_A21_VREF_15 Sch=dp

set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports { AN[0] }]; #IO_L23P_T3_FOE_B_15 Sch=an[0]
set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports { AN[1] }]; #IO_L23N_T3_FWE_B_15 Sch=an[1]
set_property -dict { PACKAGE_PIN T9    IOSTANDARD LVCMOS33 } [get_ports { AN[2] }]; #IO_L24P_T3_A01_D17_14 Sch=an[2]
set_property -dict { PACKAGE_PIN J14   IOSTANDARD LVCMOS33 } [get_ports { AN[3] }]; #IO_L19P_T3_A22_15 Sch=an[3]
set_property -dict { PACKAGE_PIN P14   IOSTANDARD LVCMOS33 } [get_ports { AN[4] }]; #IO_L8N_T1_D12_14 Sch=an[4]
set_property -dict { PACKAGE_PIN T14   IOSTANDARD LVCMOS33 } [get_ports { AN[5] }]; #IO_L14P_T2_SRCC_14 Sch=an[5]
set_property -dict { PACKAGE_PIN K2    IOSTANDARD LVCMOS33 } [get_ports { AN[6] }]; #IO_L23P_T3_35 Sch=an[6]
set_property -dict { PACKAGE_PIN U13   IOSTANDARD LVCMOS33 } [get_ports { AN[7] }]; #IO_L23N_T3_A02_D18_14 Sch=an[7]


##Buttons

#set_property -dict { PACKAGE_PIN C12   IOSTANDARD LVCMOS33 } [get_ports { CPU_RESETN }]; #IO_L3P_T0_DQS_AD1P_15 Sch=cpu_resetn

set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVCMOS33 } [get_ports { rst_n  }]; #IO_L9P_T1_DQS_14 Sch=btnc
```

Simulation:

```
// Generate 100 MHz clock
always #5 clk = ~clk;

initial begin
  // Initialize signals
  clk = 0;
  rst_n = 0;
  SW = 16'h0000;

  // Reset pulse
  #50;
  rst_n = 1;

  // Cycle through test cases every 1 ms
  SW = 16'h1234; // Test case 1
  #1_000_000;

  SW = 16'hABCD; // Test case 2
  #1_000_000;

  SW = 16'hF2A1; // Test case 3
  #1_000_000;

  SW = 16'h0000; // All off
  #1_000_000;

  SW = 16'h9876; // Test case 4
  #1_000_000;

  SW = 16'h5E0F; // Test case 5
  #1_000_000;

  $stop; // End simulation
end
```

```
// Testbench signals
reg clk;
reg rst_n;
reg [15:0] SW;
wire [15:0] led;
wire [6:0] Cnode;
wire dp;
wire [7:0] AN;

// Instantiate DUT
topmodule uut (
  .clk(clk),
  .rst_n(rst_n),
  .SW(SW),
  .led(led),
  .Cnode(Cnode),
  .dp(dp),
  .AN(AN)
);
```

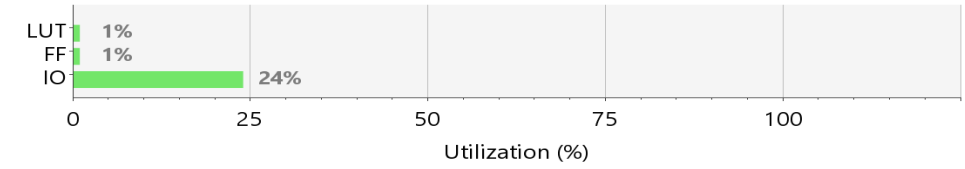
This Verilog testbench simulates the functionality of our topmodule source file by generating a 100 MHz clock and initializing input signals clk, rst_n, and SW. It applies a reset pulse shortly after the start of simulation and then sequentially assigns a series of predefined 16-bit values to the SW input every 1 millisecond to test different cases.



The waveform simulation confirms correct functionality of the topmodule for the Switch-to-7-Segment Display interface. The clock (clk) and reset (rst_n) signals initialize the system properly, and the switch input SW[15:0] is set to 0x1234, which is accurately mirrored on the led[15:0] output. The AN signal transitions from undefined to FE, activating digit 0, while the Cnode output displays the correct 7-segment pattern (0x4C) for the value 4 (from SW[3:0]). The decimal point (dp) remains off as expected. Overall, the waveform shows that the display controller is correctly decoding and multiplexing the switch inputs to the appropriate digit positions on the 7-segment display.

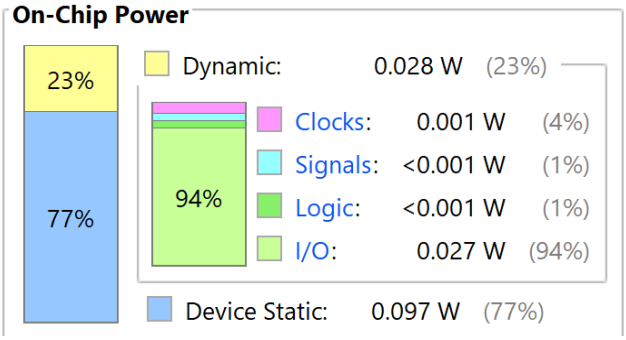
Implementation:

Resource	Utilization	Available	Utilization %
LUT	13	63400	0.02
FF	20	126800	0.02
IO	50	210	23.81



Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**0.126 W**
Design Power Budget:	**Not Specified**
Process:	typical
Power Budget Margin:	**N/A**



Video Link: https://www.youtube.com/watch?v=74gpJK-A0B4

Contributions:

Andy Siu: 50% verilog source files, demo, report

Dalton Hoang: 50% testbench, simulation, report