**ECE 3300L**

**California State Polytechnic University, Pomona**

**Group G**

**Nathaniel Garcia and Mikael Parsmyan**

**Lab Report #4**

**07/10/2025**

**Explanation:**

**Driver:**

A screenshot of a computer program

AI-generated content may be incorrect.

**Seg7\_driver:**

`timescale 1ns / 1ps  
module seg7\_driver(  
input clk,  
input rst\_n,  
input [31:0] SW,  
output reg [6:0] Cnode,  
output dp,

output [7:0] AN  
);  
reg [19:0] tmp;  
reg [3:0] digit;  
assign dp = 1'b1;  
always@(digit)  
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  
always@(posedge clk or negedge rst\_n)  
if(!rst\_n) tmp<=0;  
else tmp<=tmp+1;  
wire [2:0] s = tmp[19:17];  
always@(s, SW)  
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  
reg [7:0] AN\_tmp;  
always@(s)  
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  
assign AN=AN\_tmp;  
endmodule

**Tb\_Driver:**

module tb\_Driver;

// Inputs

reg clk;

reg rst\_n;

reg [15:0] SW\_in;

// Outputs

wire [15:0] LED;

wire [6:0] Cnode;

wire dp;

wire [7:0] AN;

// Instantiate DUT

Driver uut (

.clk(clk),

.rst\_n(rst\_n),

.SW\_in(SW\_in),

.LED(LED),

.Cnode(Cnode),

.dp(dp),

.AN(AN)

);

// 100MHz clock

initial begin

clk = 0;

forever #5 clk = ~clk;

end

// Stimulus

initial begin

rst\_n = 0;

SW\_in = 16'h0000;

// Hold reset

#20 rst\_n = 1;

// Set switches to show 'ABCD'

#50 SW\_in = 16'hABCD;

// Change to '1234'

#50000 SW\_in = 16'h1234;

// Change to 'FFFF'

#50000 SW\_in = 16'hFFFF;

// Finish after some time

#100000 $finish;

end

endmodule

**Contributions:**

Nathaniel: Demo, Xdc file and Implementaion

Mikael: Driver, Testbench and Simulation