California Polytechnic State University Pomona

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Dgtl Circuit Dsgn Verilog Lab

ECE 3300L Section E01

Report # 4

Prepared by

Arvin Ghaloosian (017153422)
Vittorio Huizar (016826784)
Group H

Presented to

Mohamed Aly (Mohamed El-Hadedy)

Due - July 11th, 2025

DESIGN OVERVIEW:

This lab implements a complete Switch-to-7-Segment Display interface on the Nexys A7-100T FPGA board. The design reads 16 slide switch inputs (SW[15:0]), mirrors them to LEDs, and displays the corresponding hexadecimal values on the 8-digit 7-segment display using a time-multiplexed driver.

A seg7_driver module was used to manage digit selection and hexadecimal-to-segment decoding. Each 4-bit nibble of the 32-bit input is displayed in its corresponding position on the display, with SW[15:0] showing up on the right-most four digits.

MODULE HIERARCHY:

- **Top Module**: lab4_top
- **7-Segment Display Driver**: seg7_driver

HOW THE DESIGN WORKS:

Inputs:

- clk – 100 MHz system clock
- rst_n – active-low reset
- SW[15:0] – slide switch input

Outputs:

- LED[15:0] – mirrors the values of switches
- AN[7:0] – digit select (active-low)
- Cnode[6:0] – segment lines (active-low)
- dp – decimal point (always off)

The design operates by:

1. Assigning each nibble of SW[15:0] to a digit position on the 7-segment display.
2. Cycling through each of the 8 digits using a counter (multiplexing).
3. Using active-low logic to illuminate the correct segment pattern for each 4-bit value.

Top Module:

```
`timescale 1ns/1ps
module lab4_top
(
    input  wire        CLK100MHZ,
    input  wire        BTN_C,
    input  wire [15:0] SW,
    output wire [15:0] LED,
    output wire [6:0]  Cnode,
    output wire        DP,
    output wire [7:0]  AN
);

    // Drive on-board LEDs
    assign LED = SW;        // Simple mirror

    // 7-segment driver instance
    seg7_driver u_seg7_driver
    (
        .clk   (CLK100MHZ),
        .rst_n (~BTN_C),
        .SW    ({16'h0000, SW}),
        .Cnode (Cnode),
        .dp    (DP),
        .AN    (AN)
    );

endmodule
```

Explanation:
The top-level module connects inputs from the slide switches to both the LED outputs and the

seg7_driver module. It extends the 16-bit switch input to 32-bits (padding with zeros) to match the driver's interface. The driver handles digit multiplexing and decoding internally.

---

seg7_driver Module:

```
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;

// Decimal point disabled
assign dp = 1'b1;

// Segment patterns (active-low)
always @(digit)
    case(digit)
        4'd0:  Cnode = 7'b1000000;
        4'd1:  Cnode = 7'b1111001;
        4'd2:  Cnode = 7'b0100100;
        4'd3:  Cnode = 7'b0110000;
        4'd4:  Cnode = 7'b0011001;
        4'd5:  Cnode = 7'b0010010;
        4'd6:  Cnode = 7'b0000010;
        4'd7:  Cnode = 7'b1111000;
        4'd8:  Cnode = 7'b0000000;
        4'd9:  Cnode = 7'b0010000;
        4'd10: Cnode = 7'b0001000;
        4'd11: Cnode = 7'b0000011;
        4'd12: Cnode = 7'b1000110;
        4'd13: Cnode = 7'b0100001;
        4'd14: Cnode = 7'b0000110;
        4'd15: Cnode = 7'b0001110;
        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
```

Explanation:

This module handles:

- 7-segment decoding from 4-bit hex digits.
- Rotating through all 8 digits using a slow counter for persistence of vision.
- Providing proper active-low outputs to both the segment lines and digit enable pins.

Special care was taken to correct the segment encoding for 0 to avoid the visual artifact of it appearing as an “A.”

XDC Snippet:

```
## This file is a general .xdc for the Nexys A7-100T
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project

# Clock signal
set_property -dict { PACKAGE_PIN E3    IOSTANDARD LVCMOS33 } [get_ports { CLK100MHZ }]; #IO_L12P_T1_MRCC_35 Sch=clk100mhz
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports {CLK100MHZ}];


##Switches

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]

#set_property -dict { PACKAGE_PIN R12   IOSTANDARD LVCMOS33 } [get_ports { LED16_B }]; #IO_L5P_T0_D06_14 Sch=led16_b
#set_property -dict { PACKAGE_PIN M16   IOSTANDARD LVCMOS33 } [get_ports { LED16_G }]; #IO_L10P_T1_D14_14 Sch=led16_g
#set_property -dict { PACKAGE_PIN N15   IOSTANDARD LVCMOS33 } [get_ports { LED16_R }]; #IO_L11P_T1_SRCC_14 Sch=led16_r
#set_property -dict { PACKAGE_PIN G14   IOSTANDARD LVCMOS33 } [get_ports { LED17_B }]; #IO_L15N_T2_DQS_ADV_B_15 Sch=led17_b
#set_property -dict { PACKAGE_PIN R11   IOSTANDARD LVCMOS33 } [get_ports { LED17_G }]; #IO_0_14 Sch=led17_g
#set_property -dict { PACKAGE_PIN N16   IOSTANDARD LVCMOS33 } [get_ports { LED17_R }]; #IO_L11N_T1_SRCC_14 Sch=led17_r


#7 segment display

set_property -dict { PACKAGE_PIN T10   IOSTANDARD LVCMOS33 } [get_ports {  Cnode[0] }]; #IO_L24N_T3_A00_D16_14 Sch=ca
set_property -dict { PACKAGE_PIN R10   IOSTANDARD LVCMOS33 } [get_ports {  Cnode[1] }]; #IO_25_14 Sch=cb
set_property -dict { PACKAGE_PIN K16   IOSTANDARD LVCMOS33 } [get_ports {  Cnode[2] }]; #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[4] }]; #IO_L13P_T2_MRCC_14 Sch=ce
set_property -dict { PACKAGE_PIN T11   IOSTANDARD LVCMOS33 } [get_ports {  Cnode[5] }]; #IO_L19P_T3_A10_D26_14 Sch=cf
set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports {  Cnode[6] }]; #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 { BTN_C }]; #IO_L9P_T1_DQS_14 Sch=btnc
#set_property -dict { PACKAGE_PIN M18   IOSTANDARD LVCMOS33 } [get_ports { BTNU }]; #IO_L4N_T0_D05_14 Sch=btnu
#set_property -dict { PACKAGE_PIN P17   IOSTANDARD LVCMOS33 } [get_ports { BTNL }]; #IO_L12P_T1_MRCC_14 Sch=btnl
#set_property -dict { PACKAGE_PIN M17   IOSTANDARD LVCMOS33 } [get_ports { BTNR }]; #IO_L10N_T1_D15_14 Sch=btnr
#set_property -dict { PACKAGE_PIN P18   IOSTANDARD LVCMOS33 } [get_ports { BTND }]; #IO_L9N_T1_DQS_D13_14 Sch=btnd


##Pmod Headers


##Pmod Header JA

#set_property -dict { PACKAGE_PIN C17   IOSTANDARD LVCMOS33 } [get_ports { JA[1] }]; #IO_L20N_T3_A19_15 Sch=ja[1]
#set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports { JA[2] }]; #IO_L21N_T3_DQS_A18_15 Sch=ja[2]
```

```
#set_property -dict { PACKAGE_PIN E18   IOSTANDARD LVCMOS33 } [get_ports { JA[3] }];
#IO_L21P_T3_DQS_15 Sch=ja[3]
#set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVCMOS33 } [get_ports { JA[4] }];
#IO_L18N_T2_A23_15 Sch=ja[4]
#set_property -dict { PACKAGE_PIN D17   IOSTANDARD LVCMOS33 } [get_ports { JA[7] }];
#IO_L16N_T2_A27_15 Sch=ja[7]
#set_property -dict { PACKAGE_PIN E17   IOSTANDARD LVCMOS33 } [get_ports { JA[8] }];
#IO_L16P_T2_A28_15 Sch=ja[8]
#set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports { JA[9] }];
#IO_L22N_T3_A16_15 Sch=ja[9]
#set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports { JA[10] }];
#IO_L22P_T3_A17_15 Sch=ja[10]


##Pmod Header JB

#set_property -dict { PACKAGE_PIN D14   IOSTANDARD LVCMOS33 } [get_ports { JB[1] }];
#IO_L1P_T0_AD0P_15 Sch=jb[1]
#set_property -dict { PACKAGE_PIN F16   IOSTANDARD LVCMOS33 } [get_ports { JB[2] }];
#IO_L14N_T2_SRCC_15 Sch=jb[2]
#set_property -dict { PACKAGE_PIN G16   IOSTANDARD LVCMOS33 } [get_ports { JB[3] }];
#IO_L13N_T2_MRCC_15 Sch=jb[3]
#set_property -dict { PACKAGE_PIN H14   IOSTANDARD LVCMOS33 } [get_ports { JB[4] }];
#IO_L15P_T2_DQS_15 Sch=jb[4]
#set_property -dict { PACKAGE_PIN E16   IOSTANDARD LVCMOS33 } [get_ports { JB[7] }];
#IO_L11N_T1_SRCC_15 Sch=jb[7]
#set_property -dict { PACKAGE_PIN F13   IOSTANDARD LVCMOS33 } [get_ports { JB[8] }];
#IO_L5P_T0_AD9P_15 Sch=jb[8]
#set_property -dict { PACKAGE_PIN G13   IOSTANDARD LVCMOS33 } [get_ports { JB[9] }];
#IO_0_15 Sch=jb[9]
#set_property -dict { PACKAGE_PIN H16   IOSTANDARD LVCMOS33 } [get_ports { JB[10] }];
#IO_L13P_T2_MRCC_15 Sch=jb[10]


##Pmod Header JC

#set_property -dict { PACKAGE_PIN K1    IOSTANDARD LVCMOS33 } [get_ports { JC[1] }];
#IO_L23N_T3_35 Sch=jc[1]
#set_property -dict { PACKAGE_PIN F6    IOSTANDARD LVCMOS33 } [get_ports { JC[2] }];
#IO_L19N_T3_VREF_35 Sch=jc[2]
#set_property -dict { PACKAGE_PIN J2    IOSTANDARD LVCMOS33 } [get_ports { JC[3] }];
#IO_L22N_T3_35 Sch=jc[3]
#set_property -dict { PACKAGE_PIN G6    IOSTANDARD LVCMOS33 } [get_ports { JC[4] }];
#IO_L19P_T3_35 Sch=jc[4]
```

```
#set_property -dict { PACKAGE_PIN E7    IOSTANDARD LVCMOS33 } [get_ports { JC[7] }];
#IO_L6P_T0_35 Sch=jc[7]
#set_property -dict { PACKAGE_PIN J3    IOSTANDARD LVCMOS33 } [get_ports { JC[8] }];
#IO_L22P_T3_35 Sch=jc[8]
#set_property -dict { PACKAGE_PIN J4    IOSTANDARD LVCMOS33 } [get_ports { JC[9] }];
#IO_L21P_T3_DQS_35 Sch=jc[9]
#set_property -dict { PACKAGE_PIN E6    IOSTANDARD LVCMOS33 } [get_ports { JC[10] }];
#IO_L5P_T0_AD13P_35 Sch=jc[10]


##Pmod Header JD

#set_property -dict { PACKAGE_PIN H4    IOSTANDARD LVCMOS33 } [get_ports { JD[1] }];
#IO_L21N_T3_DQS_35 Sch=jd[1]
#set_property -dict { PACKAGE_PIN H1    IOSTANDARD LVCMOS33 } [get_ports { JD[2] }];
#IO_L17P_T2_35 Sch=jd[2]
#set_property -dict { PACKAGE_PIN G1    IOSTANDARD LVCMOS33 } [get_ports { JD[3] }];
#IO_L17N_T2_35 Sch=jd[3]
#set_property -dict { PACKAGE_PIN G3    IOSTANDARD LVCMOS33 } [get_ports { JD[4] }];
#IO_L20N_T3_35 Sch=jd[4]
#set_property -dict { PACKAGE_PIN H2    IOSTANDARD LVCMOS33 } [get_ports { JD[7] }];
#IO_L15P_T2_DQS_35 Sch=jd[7]
#set_property -dict { PACKAGE_PIN G4    IOSTANDARD LVCMOS33 } [get_ports { JD[8] }];
#IO_L20P_T3_35 Sch=jd[8]
#set_property -dict { PACKAGE_PIN G2    IOSTANDARD LVCMOS33 } [get_ports { JD[9] }];
#IO_L15N_T2_DQS_35 Sch=jd[9]
#set_property -dict { PACKAGE_PIN F3    IOSTANDARD LVCMOS33 } [get_ports { JD[10] }];
#IO_L13N_T2_MRCC_35 Sch=jd[10]


#Pmod Header JXADC

#set_property -dict { PACKAGE_PIN A14   IOSTANDARD LVCMOS33     } [get_ports { vauxn3
}]; #IO_L9N_T1_DQS_AD3N_15 Sch=xa_n[1]
#set_property -dict { PACKAGE_PIN A13   IOSTANDARD LVCMOS33     } [get_ports { vauxp3
}]; #IO_L9P_T1_DQS_AD3P_15 Sch=xa_p[1]
#set_property -dict { PACKAGE_PIN A16   IOSTANDARD LVCMOS33     } [get_ports {
vauxn10}]; #IO_L8N_T1_AD10N_15 Sch=xa_n[2]
#set_property -dict { PACKAGE_PIN A15   IOSTANDARD LVCMOS33     } [get_ports { vauxp10
}]; #IO_L8P_T1_AD10P_15 Sch=xa_p[2]
#set_property -dict { PACKAGE_PIN B17   IOSTANDARD LVCMOS33     } [get_ports { vauxn2
}]; #IO_L7N_T1_AD2N_15 Sch=xa_n[3]
#set_property -dict { PACKAGE_PIN B16   IOSTANDARD LVCMOS33     } [get_ports { vauxp2
}]; #IO_L7P_T1_AD2P_15 Sch=xa_p[3]
```

```
#set_property -dict { PACKAGE_PIN A18   IOSTANDARD LVCMOS33     } [get_ports { vauxn11 }]; #IO_L10N_T1_AD11N_15 Sch=xa_n[4]
#set_property -dict { PACKAGE_PIN B18   IOSTANDARD LVCMOS33     } [get_ports { vauxp11 }]; #IO_L10P_T1_AD11P_15 Sch=xa_p[4]


##VGA Connector

#set_property -dict { PACKAGE_PIN A3    IOSTANDARD LVCMOS33 } [get_ports { VGA_R[0] }]; #IO_L8N_T1_AD14N_35 Sch=vga_r[0]
#set_property -dict { PACKAGE_PIN B4    IOSTANDARD LVCMOS33 } [get_ports { VGA_R[1] }]; #IO_L7N_T1_AD6N_35 Sch=vga_r[1]
#set_property -dict { PACKAGE_PIN C5    IOSTANDARD LVCMOS33 } [get_ports { VGA_R[2] }]; #IO_L1N_T0_AD4N_35 Sch=vga_r[2]
#set_property -dict { PACKAGE_PIN A4    IOSTANDARD LVCMOS33 } [get_ports { VGA_R[3] }]; #IO_L8P_T1_AD14P_35 Sch=vga_r[3]

#set_property -dict { PACKAGE_PIN C6    IOSTANDARD LVCMOS33 } [get_ports { VGA_G[0] }]; #IO_L1P_T0_AD4P_35 Sch=vga_g[0]
#set_property -dict { PACKAGE_PIN A5    IOSTANDARD LVCMOS33 } [get_ports { VGA_G[1] }]; #IO_L3N_T0_DQS_AD5N_35 Sch=vga_g[1]
#set_property -dict { PACKAGE_PIN B6    IOSTANDARD LVCMOS33 } [get_ports { VGA_G[2] }]; #IO_L2N_T0_AD12N_35 Sch=vga_g[2]
#set_property -dict { PACKAGE_PIN A6    IOSTANDARD LVCMOS33 } [get_ports { VGA_G[3] }]; #IO_L3P_T0_DQS_AD5P_35 Sch=vga_g[3]

#set_property -dict { PACKAGE_PIN B7    IOSTANDARD LVCMOS33 } [get_ports { VGA_B[0] }]; #IO_L2P_T0_AD12P_35 Sch=vga_b[0]
#set_property -dict { PACKAGE_PIN C7    IOSTANDARD LVCMOS33 } [get_ports { VGA_B[1] }]; #IO_L4N_T0_35 Sch=vga_b[1]
#set_property -dict { PACKAGE_PIN D7    IOSTANDARD LVCMOS33 } [get_ports { VGA_B[2] }]; #IO_L6N_T0_VREF_35 Sch=vga_b[2]
#set_property -dict { PACKAGE_PIN D8    IOSTANDARD LVCMOS33 } [get_ports { VGA_B[3] }]; #IO_L4P_T0_35 Sch=vga_b[3]

#set_property -dict { PACKAGE_PIN B11   IOSTANDARD LVCMOS33 } [get_ports { VGA_HS }]; #IO_L4P_T0_15 Sch=vga_hs
#set_property -dict { PACKAGE_PIN B12   IOSTANDARD LVCMOS33 } [get_ports { VGA_VS }]; #IO_L3N_T0_DQS_AD1N_15 Sch=vga_vs


##Micro SD Connector
```

```
#set_property -dict { PACKAGE_PIN E2    IOSTANDARD LVCMOS33 } [get_ports { SD_RESET }]; #IO_L14P_T2_SRCC_35 Sch=sd_reset
#set_property -dict { PACKAGE_PIN A1    IOSTANDARD LVCMOS33 } [get_ports { SD_CD }]; #IO_L9N_T1_DQS_AD7N_35 Sch=sd_cd
#set_property -dict { PACKAGE_PIN B1    IOSTANDARD LVCMOS33 } [get_ports { SD_SCK }]; #IO_L9P_T1_DQS_AD7P_35 Sch=sd_sck
#set_property -dict { PACKAGE_PIN C1    IOSTANDARD LVCMOS33 } [get_ports { SD_CMD }]; #IO_L16N_T2_35 Sch=sd_cmd
#set_property -dict { PACKAGE_PIN C2    IOSTANDARD LVCMOS33 } [get_ports { SD_DAT[0] }]; #IO_L16P_T2_35 Sch=sd_dat[0]
#set_property -dict { PACKAGE_PIN E1    IOSTANDARD LVCMOS33 } [get_ports { SD_DAT[1] }]; #IO_L18N_T2_35 Sch=sd_dat[1]
#set_property -dict { PACKAGE_PIN F1    IOSTANDARD LVCMOS33 } [get_ports { SD_DAT[2] }]; #IO_L18P_T2_35 Sch=sd_dat[2]
#set_property -dict { PACKAGE_PIN D2    IOSTANDARD LVCMOS33 } [get_ports { SD_DAT[3] }]; #IO_L14N_T2_SRCC_35 Sch=sd_dat[3]


##Accelerometer

#set_property -dict { PACKAGE_PIN E15   IOSTANDARD LVCMOS33 } [get_ports { ACL_MISO }]; #IO_L11P_T1_SRCC_15 Sch=acl_miso
#set_property -dict { PACKAGE_PIN F14   IOSTANDARD LVCMOS33 } [get_ports { ACL_MOSI }]; #IO_L5N_T0_AD9N_15 Sch=acl_mosi
#set_property -dict { PACKAGE_PIN F15   IOSTANDARD LVCMOS33 } [get_ports { ACL_SCLK }]; #IO_L14P_T2_SRCC_15 Sch=acl_sclk
#set_property -dict { PACKAGE_PIN D15   IOSTANDARD LVCMOS33 } [get_ports { ACL_CSN }]; #IO_L12P_T1_MRCC_15 Sch=acl_csn
#set_property -dict { PACKAGE_PIN B13   IOSTANDARD LVCMOS33 } [get_ports { ACL_INT[1] }]; #IO_L2P_T0_AD8P_15 Sch=acl_int[1]
#set_property -dict { PACKAGE_PIN C16   IOSTANDARD LVCMOS33 } [get_ports { ACL_INT[2] }]; #IO_L20P_T3_A20_15 Sch=acl_int[2]


##Temperature Sensor

#set_property -dict { PACKAGE_PIN C14   IOSTANDARD LVCMOS33 } [get_ports { TMP_SCL }]; #IO_L1N_T0_AD0N_15 Sch=tmp_scl
#set_property -dict { PACKAGE_PIN C15   IOSTANDARD LVCMOS33 } [get_ports { TMP_SDA }]; #IO_L12N_T1_MRCC_15 Sch=tmp_sda
#set_property -dict { PACKAGE_PIN D13   IOSTANDARD LVCMOS33 } [get_ports { TMP_INT }]; #IO_L6N_T0_VREF_15 Sch=tmp_int
#set_property -dict { PACKAGE_PIN B14   IOSTANDARD LVCMOS33 } [get_ports { TMP_CT }]; #IO_L2N_T0_AD8N_15 Sch=tmp_ct
```

##Omnidirectional Microphone

#set_property -dict { PACKAGE_PIN J5   IOSTANDARD LVCMOS33 } [get_ports { M_CLK }]; #IO_25_35 Sch=m_clk
#set_property -dict { PACKAGE_PIN H5   IOSTANDARD LVCMOS33 } [get_ports { M_DATA }]; #IO_L24N_T3_35 Sch=m_data
#set_property -dict { PACKAGE_PIN F5   IOSTANDARD LVCMOS33 } [get_ports { M_LRSEL }]; #IO_0_35 Sch=m_lrsel

##PWM Audio Amplifier

#set_property -dict { PACKAGE_PIN A11   IOSTANDARD LVCMOS33 } [get_ports { AUD_PWM }]; #IO_L4N_T0_15 Sch=aud_pwm
#set_property -dict { PACKAGE_PIN D12   IOSTANDARD LVCMOS33 } [get_ports { AUD_SD }]; #IO_L6P_T0_15 Sch=aud_sd

##USB-RS232 Interface

#set_property -dict { PACKAGE_PIN C4   IOSTANDARD LVCMOS33 } [get_ports { UART_TXD_IN }]; #IO_L7P_T1_AD6P_35 Sch=uart_txd_in
#set_property -dict { PACKAGE_PIN D4   IOSTANDARD LVCMOS33 } [get_ports { UART_RXD_OUT }]; #IO_L11N_T1_SRCC_35 Sch=uart_rxd_out
#set_property -dict { PACKAGE_PIN D3   IOSTANDARD LVCMOS33 } [get_ports { UART_CTS }]; #IO_L12N_T1_MRCC_35 Sch=uart_cts
#set_property -dict { PACKAGE_PIN E5   IOSTANDARD LVCMOS33 } [get_ports { UART_RTS }]; #IO_L5N_T0_AD13N_35 Sch=uart_rts

##USB HID (PS/2)

#set_property -dict { PACKAGE_PIN F4   IOSTANDARD LVCMOS33 } [get_ports { PS2_CLK }]; #IO_L13P_T2_MRCC_35 Sch=ps2_clk
#set_property -dict { PACKAGE_PIN B2   IOSTANDARD LVCMOS33 } [get_ports { PS2_DATA }]; #IO_L10N_T1_AD15N_35 Sch=ps2_data

##SMSC Ethernet PHY

#set_property -dict { PACKAGE_PIN C9   IOSTANDARD LVCMOS33 } [get_ports { ETH_MDC }]; #IO_L11P_T1_SRCC_16 Sch=eth_mdc
#set_property -dict { PACKAGE_PIN A9   IOSTANDARD LVCMOS33 } [get_ports { ETH_MDIO }]; #IO_L14N_T2_SRCC_16 Sch=eth_mdio

#set_property -dict { PACKAGE_PIN B3   IOSTANDARD LVCMOS33 } [get_ports { ETH_RSTN }]; #IO_L10P_T1_AD15P_35 Sch=eth_rstn
#set_property -dict { PACKAGE_PIN D9   IOSTANDARD LVCMOS33 } [get_ports { ETH_CRSDV }]; #IO_L6N_T0_VREF_16 Sch=eth_crsdv
#set_property -dict { PACKAGE_PIN C10   IOSTANDARD LVCMOS33 } [get_ports { ETH_RXERR }]; #IO_L13N_T2_MRCC_16 Sch=eth_rxerr
#set_property -dict { PACKAGE_PIN C11   IOSTANDARD LVCMOS33 } [get_ports { ETH_RXD[0] }]; #IO_L13P_T2_MRCC_16 Sch=eth_rxd[0]
#set_property -dict { PACKAGE_PIN D10   IOSTANDARD LVCMOS33 } [get_ports { ETH_RXD[1] }]; #IO_L19N_T3_VREF_16 Sch=eth_rxd[1]
#set_property -dict { PACKAGE_PIN B9   IOSTANDARD LVCMOS33 } [get_ports { ETH_TXEN }]; #IO_L11N_T1_SRCC_16 Sch=eth_txen
#set_property -dict { PACKAGE_PIN A10   IOSTANDARD LVCMOS33 } [get_ports { ETH_TXD[0] }]; #IO_L14P_T2_SRCC_16 Sch=eth_txd[0]
#set_property -dict { PACKAGE_PIN A8   IOSTANDARD LVCMOS33 } [get_ports { ETH_TXD[1] }]; #IO_L12N_T1_MRCC_16 Sch=eth_txd[1]
#set_property -dict { PACKAGE_PIN D5   IOSTANDARD LVCMOS33 } [get_ports { ETH_REFCLK }]; #IO_L11P_T1_SRCC_35 Sch=eth_refclk
#set_property -dict { PACKAGE_PIN B8   IOSTANDARD LVCMOS33 } [get_ports { ETH_INTN }]; #IO_L12P_T1_MRCC_16 Sch=eth_intn

##Quad SPI Flash

#set_property -dict { PACKAGE_PIN K17   IOSTANDARD LVCMOS33 } [get_ports { QSPI_DQ[0] }]; #IO_L1P_T0_D00_MOSI_14 Sch=qspi_dq[0]
#set_property -dict { PACKAGE_PIN K18   IOSTANDARD LVCMOS33 } [get_ports { QSPI_DQ[1] }]; #IO_L1N_T0_D01_DIN_14 Sch=qspi_dq[1]
#set_property -dict { PACKAGE_PIN L14   IOSTANDARD LVCMOS33 } [get_ports { QSPI_DQ[2] }]; #IO_L2P_T0_D02_14 Sch=qspi_dq[2]
#set_property -dict { PACKAGE_PIN M14   IOSTANDARD LVCMOS33 } [get_ports { QSPI_DQ[3] }]; #IO_L2N_T0_D03_14 Sch=qspi_dq[3]
#set_property -dict { PACKAGE_PIN L13   IOSTANDARD LVCMOS33 } [get_ports { QSPI_CSN }]; #IO_L6P_T0_FCS_B_14 Sch=qspi_csn

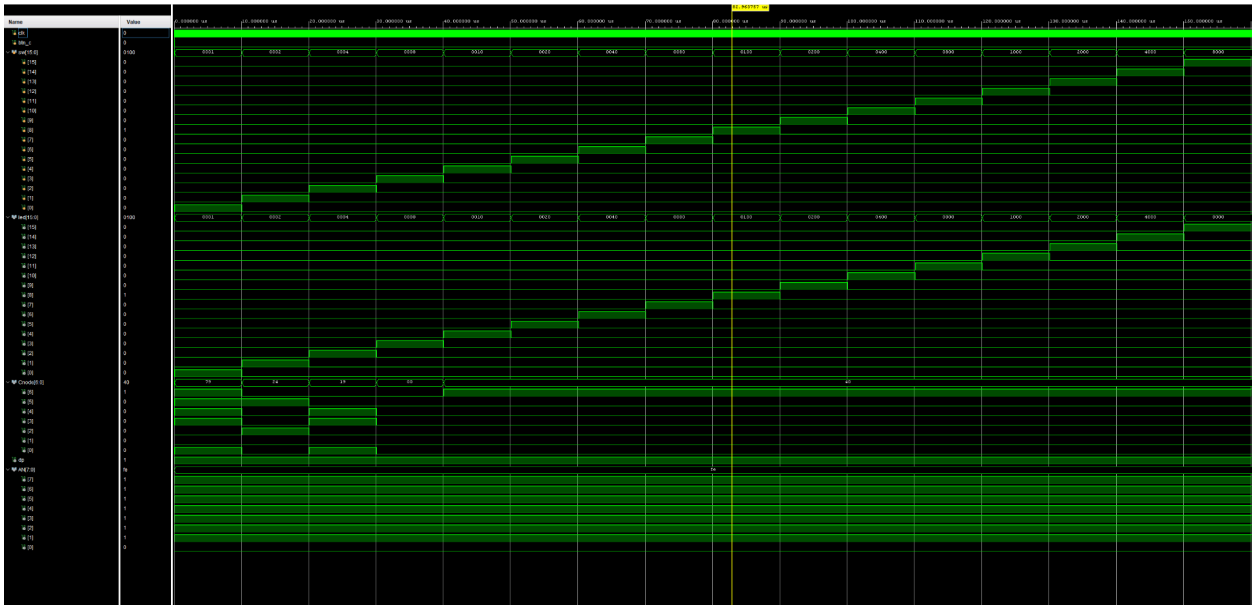
Explanation:
The constraint file was edited to map:

- clk to pin E3
- rst to pin C12
- SW[0]–SW[15] to slide switch pins

- LED[0]–LED[15] to corresponding output pins
- AN[0]–AN[7], Cnode[0]–Cnode[6], and dp to 7-segment display pins

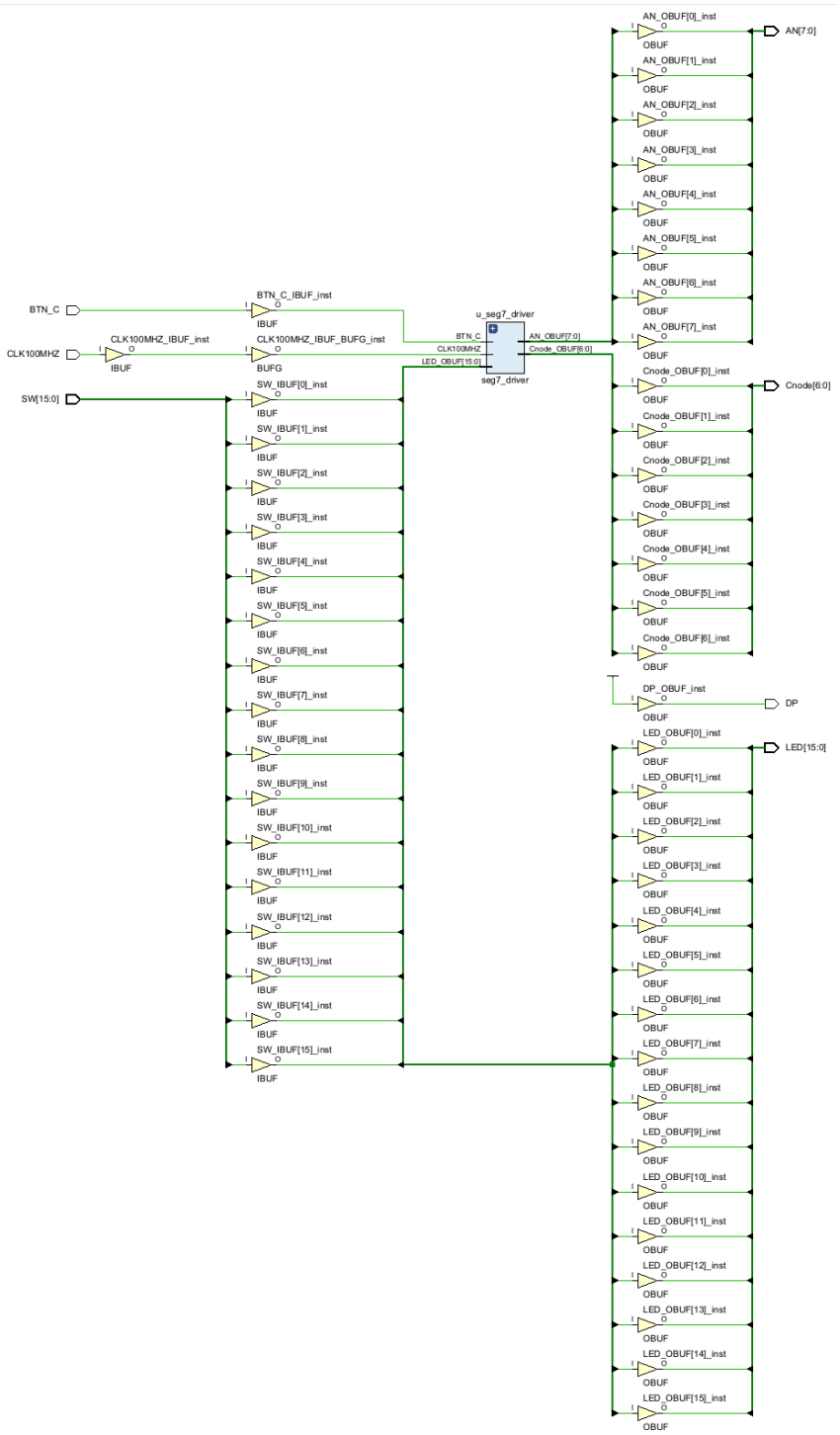
---

TEST BENCH:

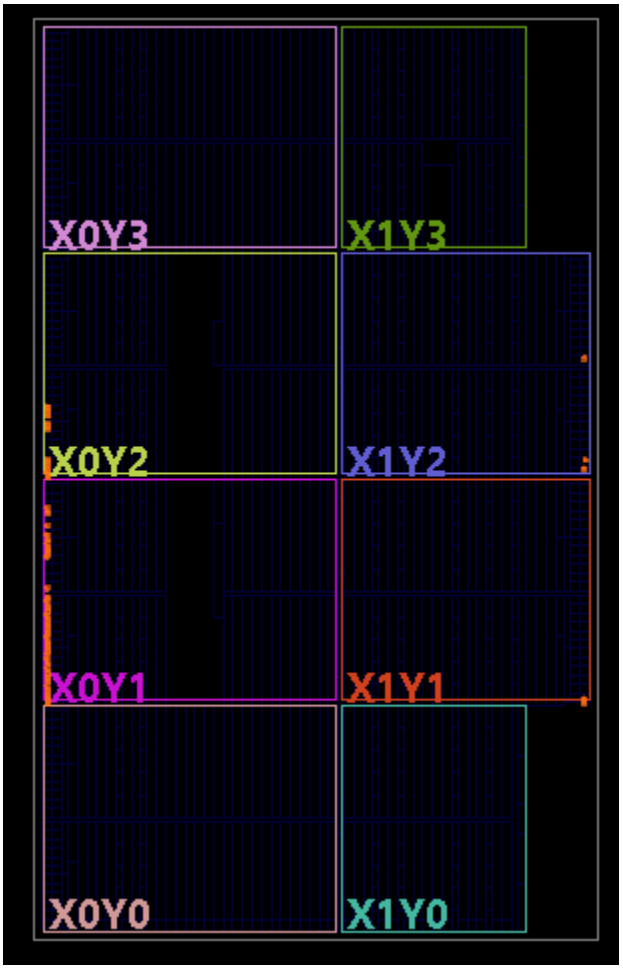


Explanation:
A simulation testbench was used to verify the display of various hex values by manipulating SW inputs and monitoring the Cnode and AN outputs to ensure the correct segments light up in the correct digit positions.

SCHEMATIC:


BTN_C
CLK100MHZ
SW[15:0]
BTN_C_IBUF_inst
IBUF
CLK100MHZ_IBUF_inst
IBUF
CLK100MHZ_IBUF_BUFG_inst
BUFG
SW_IBUF[0]_inst
SW_IBUF[1]_inst
SW_IBUF[2]_inst
SW_IBUF[3]_inst
SW_IBUF[4]_inst
SW_IBUF[5]_inst
SW_IBUF[6]_inst
SW_IBUF[7]_inst
SW_IBUF[8]_inst
SW_IBUF[9]_inst
SW_IBUF[10]_inst
SW_IBUF[11]_inst
SW_IBUF[12]_inst
SW_IBUF[13]_inst
SW_IBUF[14]_inst
SW_IBUF[15]_inst
u_seg7_driver
BTN_C
CLK100MHZ
LED_OBUF[15:0]
AN_OBUF[7:0]
Cnode_OBUF[6:0]
seg7_driver
AN_OBUF[0]_inst
AN_OBUF[1]_inst
AN_OBUF[2]_inst
AN_OBUF[3]_inst
AN_OBUF[4]_inst
AN_OBUF[5]_inst
AN_OBUF[6]_inst
AN_OBUF[7]_inst
AN[7:0]
Cnode_OBUF[0]_inst
Cnode_OBUF[1]_inst
Cnode_OBUF[2]_inst
Cnode_OBUF[3]_inst
Cnode_OBUF[4]_inst
Cnode_OBUF[5]_inst
Cnode_OBUF[6]_inst
Cnode[6:0]
DP_OBUF_inst
DP
LED_OBUF[0]_inst
LED_OBUF[1]_inst
LED_OBUF[2]_inst
LED_OBUF[3]_inst
LED_OBUF[4]_inst
LED_OBUF[5]_inst
LED_OBUF[6]_inst
LED_OBUF[7]_inst
LED_OBUF[8]_inst
LED_OBUF[9]_inst
LED_OBUF[10]_inst
LED_OBUF[11]_inst
LED_OBUF[12]_inst
LED_OBUF[13]_inst
LED_OBUF[14]_inst
LED_OBUF[15]_inst
LED[15:0]
OBUF

SYNTHESIS SCHEMATIC:



RESOURCE UTILIZATION:

Name 1	Slice LUTs (63400)	Slice Registers (126800)	Bonded IOB (210)	BUFGCTRL (32)
N lab4_top	13	20	50	1

CONTRIBUTIONS:

Arvin Ghaloosian (50%)

- Designed and verified the seg7_driver module
- Handled on-board testing and debugging
- Implemented LED mirroring and switch logic

Vittorio Huizar (50%)

- Created the top-level wrapper and simulation testbench
- Edited constraint file for correct pin mapping
- Verified timing and logic in simulation

REFLECTION:

This lab deepened our understanding of display multiplexing and active-low logic systems. Fixing segment patterns and managing counter-based digit selection required careful consideration of how the display hardware interprets signal states. We also learned how small mistakes in logic encoding (like segment g being on for 0) can produce major visual issues. This project gave us valuable experience in both debugging hardware and refining HDL logic to suit physical device behavior.

Link to demo video:

https://youtu.be/QbWb4CQdcSk
