Lab report 05

BCD to 7-Segment Decoder

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**308L-Digital Systems Design LAB**

**Department of Computer System Engineering**

**University of Engineering and Technology Peshawar**

Submitted to: **Engr. Shahzada Fahim Jan**

Submitted by: **Muhammad Saad**

Reg ID: **21PWCSE1997**

**DCSE, Batch 23, Section “B”**

Date: **SUN, 31st March, 2024**

**Lab 05**

**BCD to 7-Segment Decoder**

**Objective:**

* To implement a BCD to Seven Segment Decoder on Spartan 6 board

**Components needed for this lab:**

* Xilinx ISE
* Spartan 6 Board

**BCD to 7-Segment Display:**

In this lab, we will take a BCD input from the user and display the corresponding number on a seven-segment display. The diagram shows the 7-bit code for displaying the digit ‘2.’ tTe enable signal must be low to activate a specific seven-segment display. The Spartan 6 board has four such displays, and later labs will cover time multiplexing techniques to control all four displays using shared input signals (A, B, C, D, E, F, G, Dp).

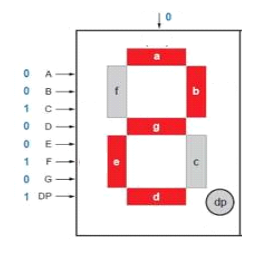


Fig 01: BCD pin and display configuration

**Tasks**

**1:** Implement buffer ON the Kit and attach the snapshot.

**module** disp**(**O**,** dp**,** A**,** en **);**

**output** **reg** **[**6**:**0**]**O**;**

**output** dp**;**

**input** **[**3**:**0**]**A**;**

**input** en**;**

**assign** dp **=** 0**;**

**always@(\*)** **begin**

**if(**A**==**4'b0000**)**

O **<=** 7'b1000000**;**//0

**else** **if(**A**==**4'b0001**)**

O **<=** 7'b1111001**;**//1

**else** **if(**A**==**4'b0010**)**

O **<=** 7'b0100100**;**//2

**else** **if(**A**==**4'b0011**)**

O **<=** 7'b0110000**;**//3

**else** **if(**A**==**4'b0100**)**

O **<=** 7'b0011001**;**//4

**else** **if(**A**==**4'b0101**)**

O **<=** 7'b0010010**;**//5

**else** **if(**A**==**4'b0110**)**

O **<=** 7'b0000010**;**//6

**else** **if(**A**==**4'b0111**)**

O **<=** 7'b1111000**;**//7

**else** **if(**A**==**4'b1000**)**

O **<=** 7'b0000000**;**//8

**else** **if(**A**==**4'b1001**)**

O **<=** 7'b0010000**;**//9

**end**

**endmodule**

Verilog Code for 7-Segment BCD

NET "A[3]" PULLUP;

NET "A[2]" PULLUP;

NET "A[1]" PULLUP;

NET "A[0]" PULLUP;

NET "O[6]" SLEW = FAST;

NET "O[5]" SLEW = FAST;

NET "O[4]" SLEW = FAST;

NET "O[3]" SLEW = FAST;

NET "O[2]" SLEW = FAST;

NET "O[1]" SLEW = FAST;

NET "O[0]" SLEW = FAST;

# PlanAhead Generated physical constraints

NET "A[3]" LOC = F17;

NET "A[2]" LOC = F18;

NET "A[1]" LOC = E16;

NET "A[0]" LOC = E18;

NET "O[0]" LOC = A3;

NET "O[1]" LOC = B4;

NET "O[2]" LOC = A4;

NET "O[3]" LOC = C4;

NET "O[4]" LOC = C5;

NET "O[5]" LOC = D6;

NET "O[6]" LOC = C6;

NET "dp" LOC = B3;

NET "en" LOC = C17;

UCF File Code

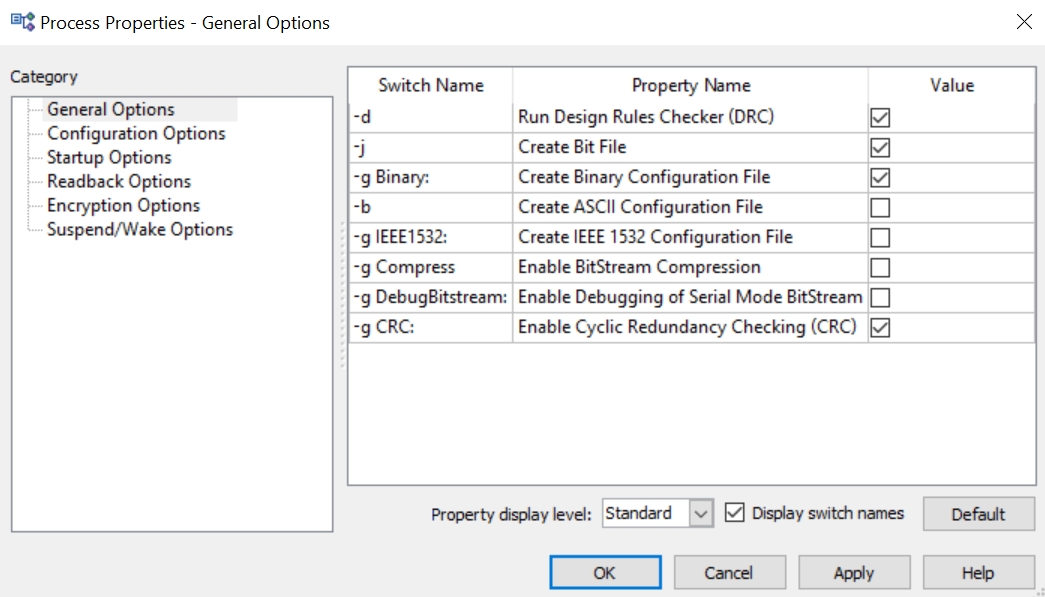


Fig 02: First we turn on ‘Create Binary Configuration File’

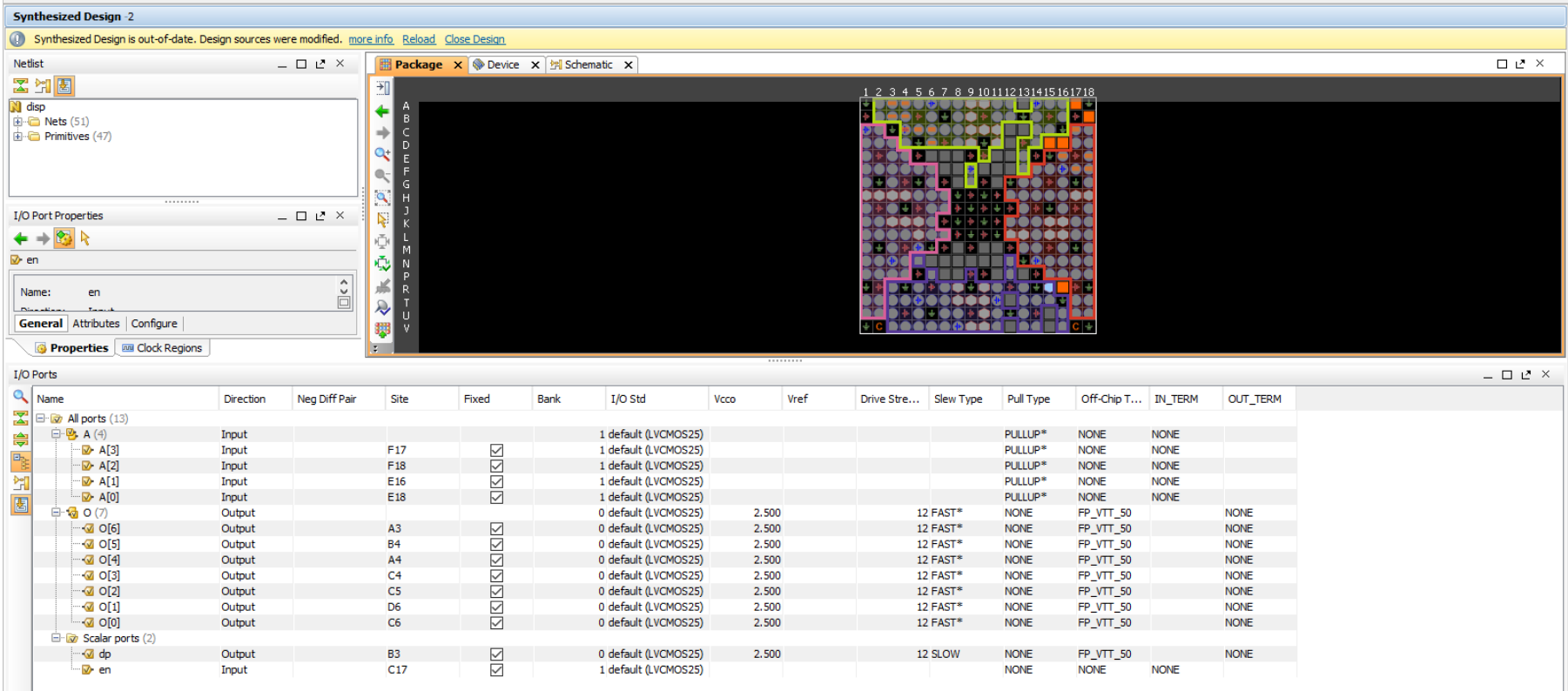


Fig 03: Choosing input and output ports for spartan 6 board

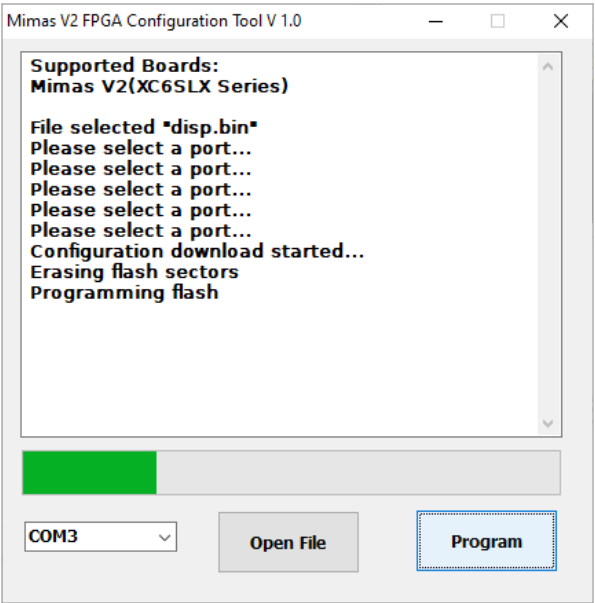


Fig 04: Programming Spartan 6 board with .bin file

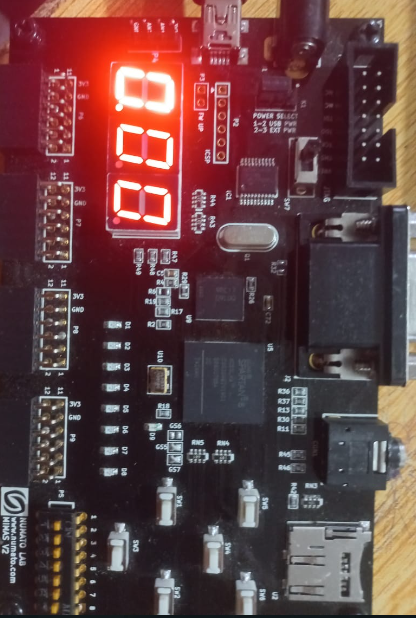


Fig 05: BCD Displaying 0

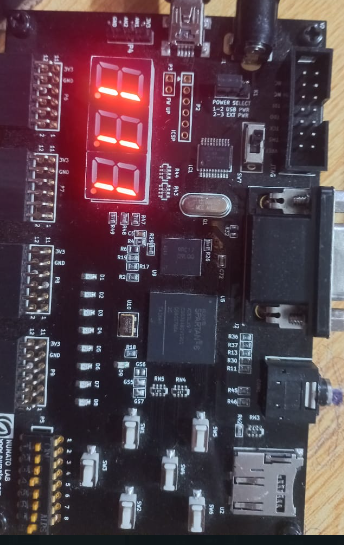


Fig 06: BCD displaying 1

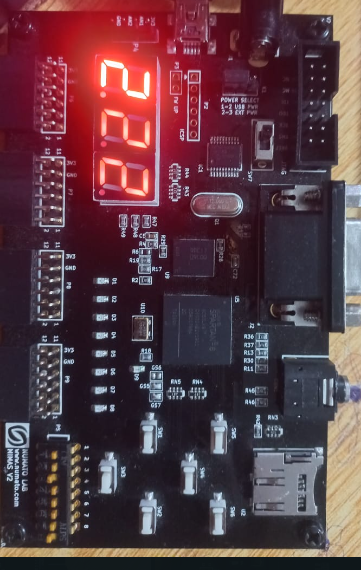


Fig 07: BCD displaying 2

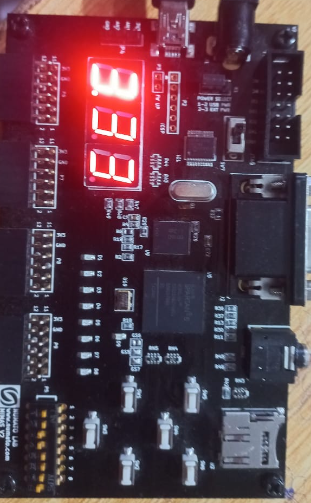


Fig 08: BCD displaying 3

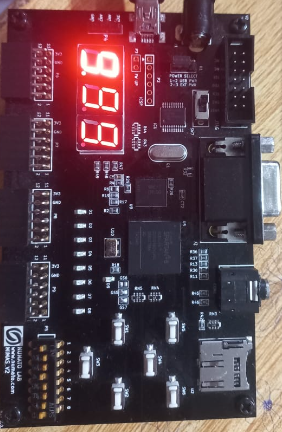


Fig 09: BCD goes upto 9

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2:** Connect the output of your lab 02 (4 bit adder) to the seven segment display. Note that number above 1001 are not valid

BCD numbers. In this situation keep the seven segment display off and just the dp on.

**module** disp**(**O**,** dp**,** A**,** B**,** en **);**

**output** **reg** **[**6**:**0**]**O**;**

**output** **reg** dp**;**

**input** **[**3**:**0**]**A**,** B**;**

**input** en**;**

**wire** **[**3**:**0**]**sum**;**

**assign** sum **=** A**+**B**;**

**always@(\*)** **begin**

**if(**sum**==**4'b0000**)**

O **<=** 7'b1000000**;**//0

**else** **if(**sum**==**4'b0001**)**

O **<=** 7'b1111001**;**//1

**else** **if(**sum**==**4'b0010**)**

O **<=** 7'b0100100**;**//2

**else** **if(**sum**==**4'b0011**)**

O **<=** 7'b0110000**;**//3

**else** **if(**sum**==**4'b0100**)**

O **<=** 7'b0011001**;**//4

**else** **if(**sum**==**4'b0101**)**

O **<=** 7'b0010010**;**//5

**else** **if(**sum**==**4'b0110**)**

O **<=** 7'b0000010**;**//6

**else** **if(**sum**==**4'b0111**)**

O **<=** 7'b1111000**;**//7

**else** **if(**sum**==**4'b1000**)**

O **<=** 7'b0000000**;**//8

**else** **if(**sum**==**4'b1001**)**

O **<=** 7'b0010000**;**//9

**else** **if(**sum**>**4'd9**)**

**begin**

dp **<=** 0**;**

O **<=** 0**;**

**end**

**end**

**endmodule**

Verilog Code for 7-Segment BCD

NET "A[3]" PULLUP;

NET "A[2]" PULLUP;

NET "A[1]" PULLUP;

NET "A[0]" PULLUP;

NET "O[6]" SLEW = FAST;

NET "O[5]" SLEW = FAST;

NET "O[4]" SLEW = FAST;

NET "O[3]" SLEW = FAST;

NET "O[2]" SLEW = FAST;

NET "O[1]" SLEW = FAST;

NET "O[0]" SLEW = FAST;

NET "A[3]" LOC = F17;

NET "A[2]" LOC = F18;

NET "A[1]" LOC = E16;

NET "A[0]" LOC = E18;

NET "O[0]" LOC = A3;

NET "O[1]" LOC = B4;

NET "O[2]" LOC = A4;

NET "O[3]" LOC = C4;

NET "O[4]" LOC = C5;

NET "O[5]" LOC = D6;

NET "O[6]" LOC = C6;

NET "dp" LOC = B3;

NET "en" LOC = V16;

NET "B[3]" LOC = D18;

NET "B[2]" LOC = D17;

NET "B[1]" LOC = C18;

NET "B[0]" LOC = C17;

# PlanAhead Generated IO constraints

NET "B[3]" PULLUP;

NET "B[2]" PULLUP;

NET "B[1]" PULLUP;

NET "B[0]" PULLUP;

UCF File Code

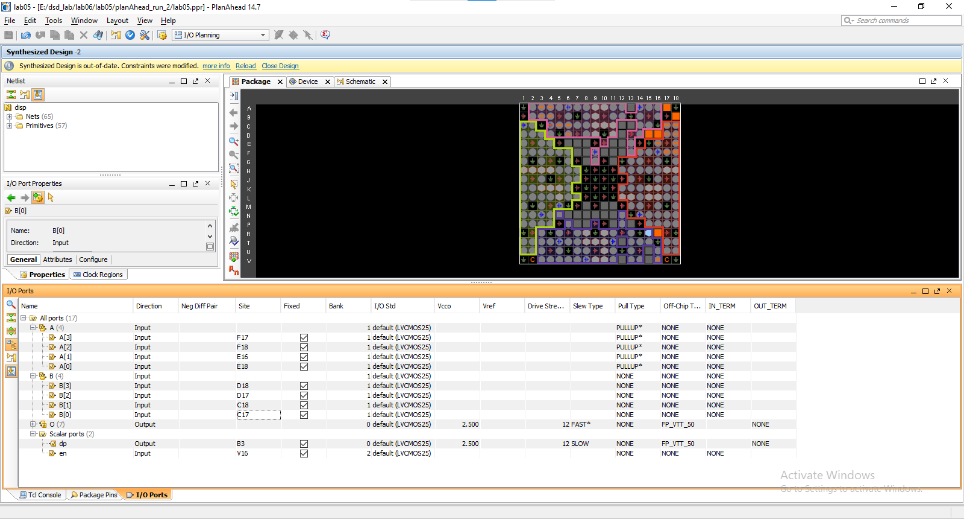


Fig 10: Choosing input and output ports for spartan 6 board

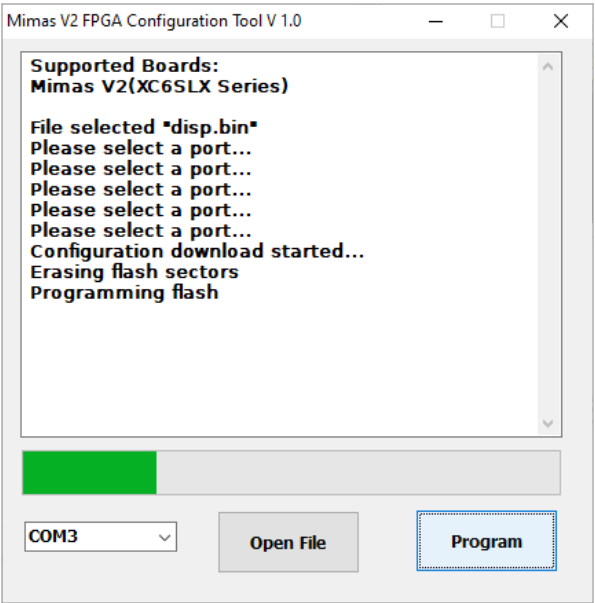


Fig 11: Programming Spartan 6 board with .bin file

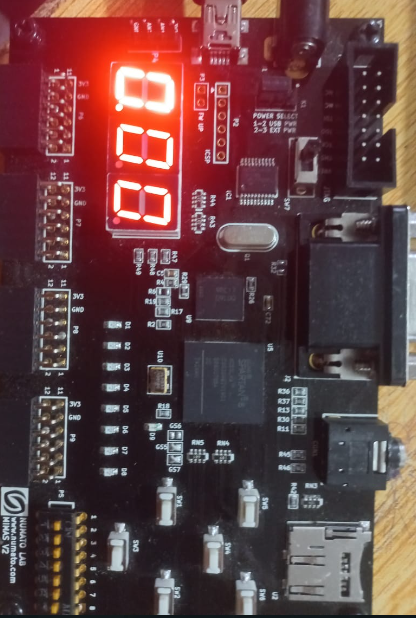


Fig 12: BCD Displaying 0

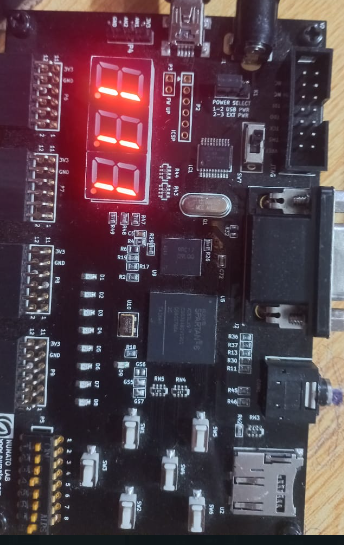


Fig 13: BCD displaying 1

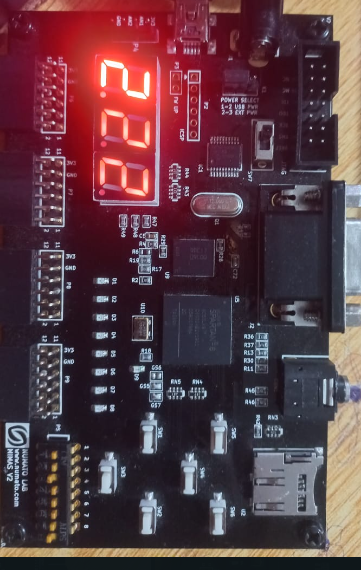


Fig 14: BCD displaying 2

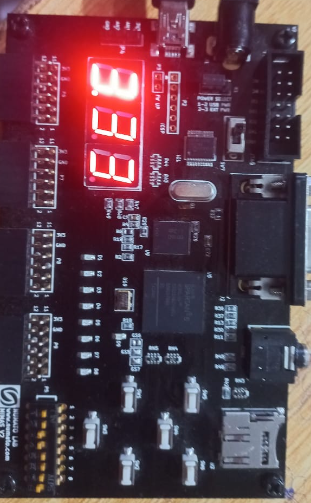


Fig 15: BCD displaying 3

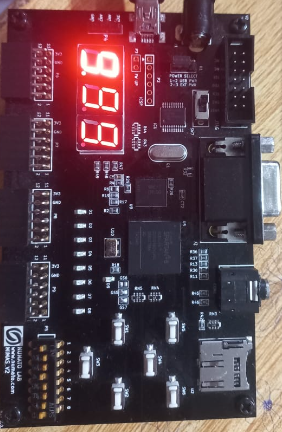


Fig 16: BCD goes upto 9

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Conclusion:**

In this Lab we learned how to program an FPGA (spartan 6) for showing 1 to 9 numbers on 7-segment display in first task and in the second we created a 4-bit adder and showed the result on 7-segment display.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**