**University of Engineering and Technology, Peshawar**
Department of Computer Systems Engineering.
*Course Lab: CSE-308 Digital System Design*

**Zubair Khan** **19** PWCSE **1797**
Section A
Batch 21 (Spring_2022)
**Submitted to** Engr. Madiha Sher

# LAB # 4 TITLE

## BCD TO SEVEN SEGMENT DECODER

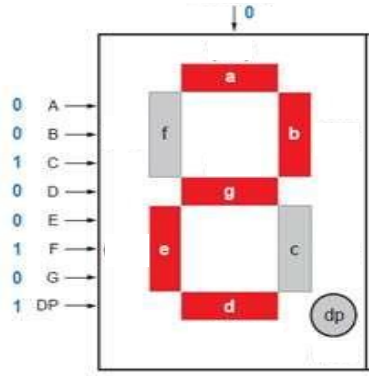
*Objective:*

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

## Block Diagram:

In This lab you are required to take a BCD input from the user and display that number on the seven segment display. Following diagram shows the 7 bit code for displaying "2" on the seven segment display all the input are active low signals. Note that enable signal should be held low in order to turn on the particular seven segment display. There are 4 seven segment displays on the Spartan 6 board. In later labs you will learn how to use time multiplexing techniques to turn on all the four seven segment displays as the input A,B,C,D,E,F,G, Dp are shared by all the four seven segment displays.

**I/O Connections:** your module will have 4 bit input and 8 bit output. Connect input to switches and output to seven segment display.




## Lab Task:

1- Using switches enter a BCD number and show the resulting number on the seven segment display.

Code:

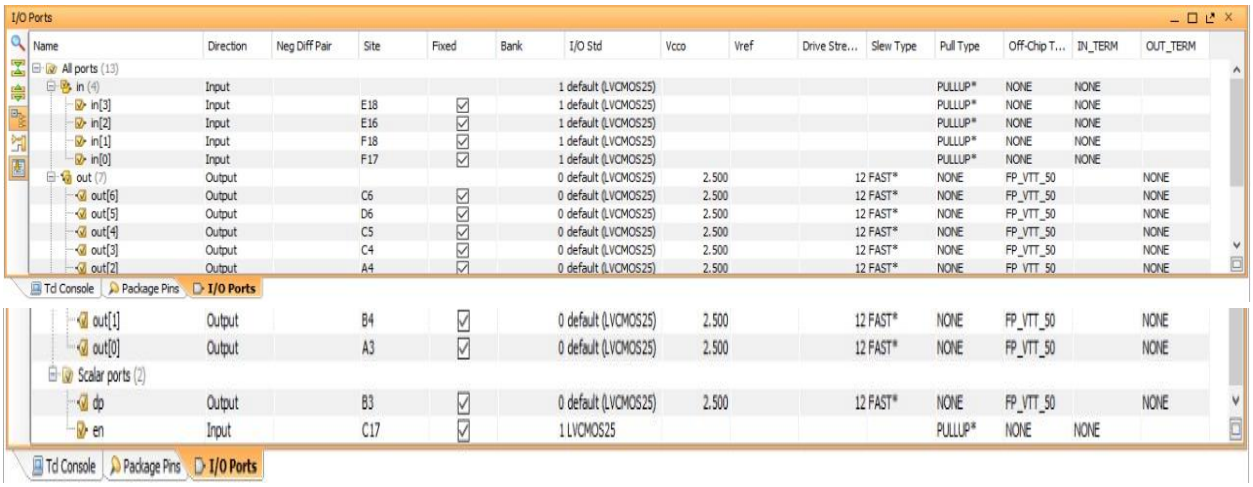
```
//////////////////////////////////////////////////////////
module task1(
    output dp,
    input en,
    input [3:0] in,
    output [6:0] out
    );

assign dp=en;
assign out=(in==0)?7'b1000000:
           (in==1)?7'b1111001:
           (in==2)?7'b0100100:
           (in==3)?7'b0110000:
           (in==4)?7'b0011001:
           (in==5)?7'b0010010:
           (in==6)?7'b0000010:
           (in==7)?7'b1111000:
           (in==8)?7'b0000000:
           (in==9)?7'b0010000:7'b1111111;
endmodule
```

UCF File:

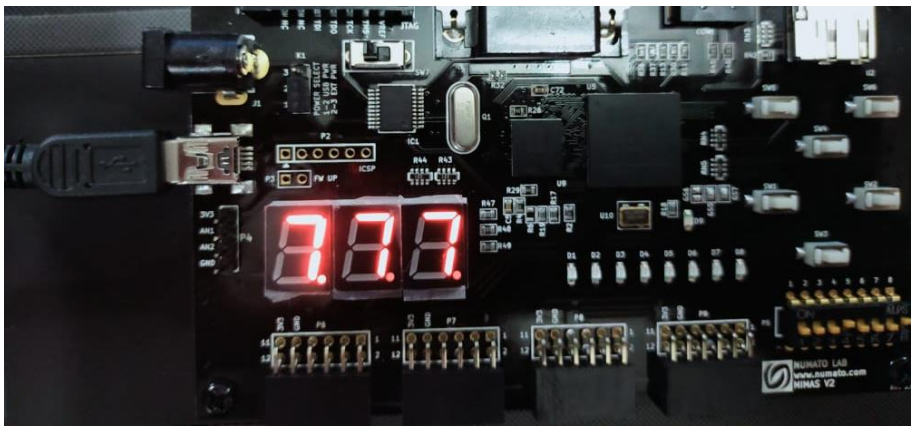
```
1
2   NET "in[3]" PULLUP;
3   NET "in[2]" PULLUP;
4   NET "in[1]" PULLUP;
5   NET "in[0]" PULLUP;
6
7
8   NET "out[0]" LOC = A3;
9   NET "out[3]" LOC = C4;
10  NET "out[2]" LOC = A4;
11  NET "out[5]" LOC = D6;
12  NET "out[1]" LOC = B4;
13  NET "out[4]" LOC = C5;
14  NET "en" LOC = C17;
15  NET "dp" LOC = B3;
16  NET "in[3]" LOC = E18;
17
18
19  NET "out[6]" SLEW = FAST;
20  NET "out[5]" SLEW = FAST;
21  NET "out[4]" SLEW = FAST;
22  NET "out[3]" SLEW = FAST;
23  NET "out[2]" SLEW = FAST;
24  NET "out[1]" SLEW = FAST;
25  NET "out[0]" SLEW = FAST;
```

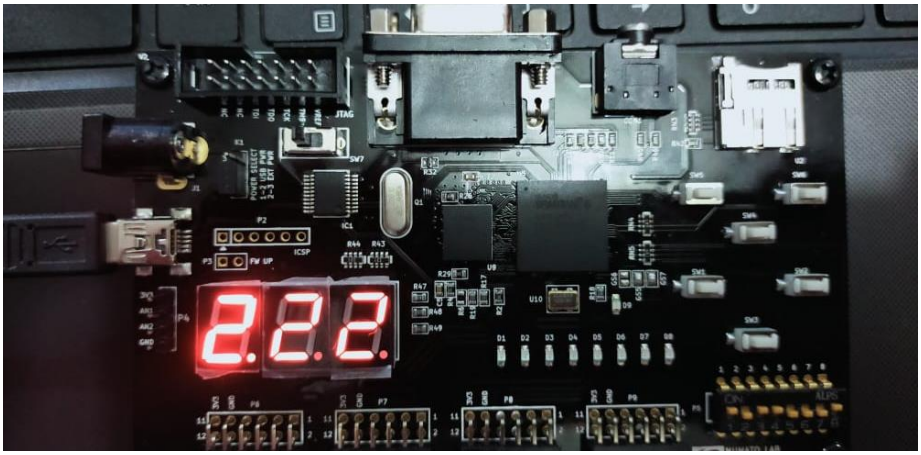
```
26
27
28  NET "out[6]" LOC = C6;
29
30
31  NET "dp" SLEW = FAST;
32  NET "en" PULLUP;
33
34
35  NET "in[2]" LOC = E16;
36  NET "in[1]" LOC = F18;
37  NET "in[0]" LOC = F17;
38
```

## I/O Pin Planning:




## Output:



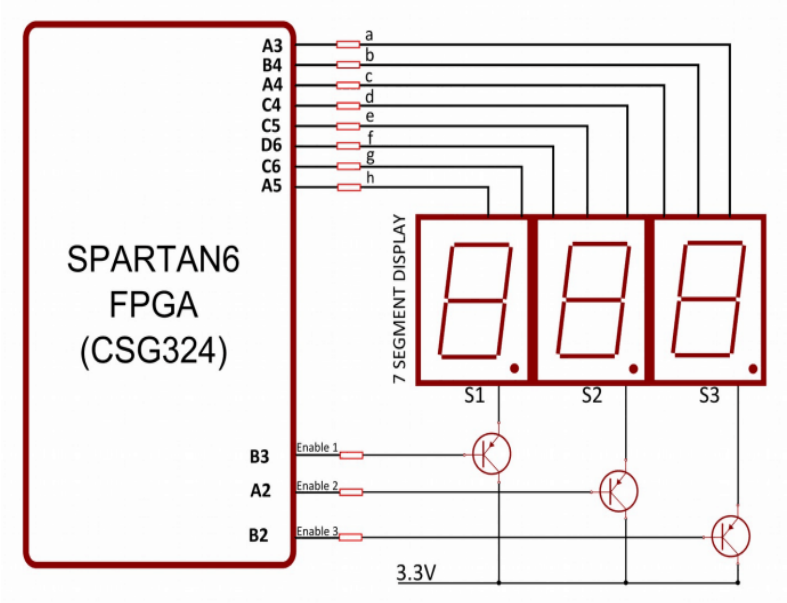


2- Connect the output of your lab 02 (4 bit adder) to the seven segment display. Note that number above 1001 are not valid BCD numbers. In this situation keep the seven segment display off and just the dp on.

## 7 Segment Led Display On Spartan 6 Board

This board features three 7-segment LED display multiplexed for low pin count operation. Each module can be separately turned on and off with the three switching transistors.

Note: All signals (a, b, c, d, e, f, g, dot, enable 1, enable 2, enable 3) used for controlling 7-Segment display are active-low signals. So, for example, for displaying “8” in display-2, users need to drive Enable 2 to 0 as well as drive signals a, b, c, d, e, f to 0. All other signals need to be driven to 1.



Code:

```
module FullAdderr(Sum, Cout, X, Y, Cin);
input X, Y, Cin;
output Sum, Cout;
//Behavoural Code
assign {Cout,Sum}=X+Y+Cin;
endmodule

module RippleCarryAdder (Cout, S, X, Y);
output Cout;
output [3:0] S;
input [3:0] X, Y;
wire [2:0] C; //Intermediate/Internal Carries
// FA1 (Sum, Cout, X, Y, Cin); //FA1's Interface (I/O Pins)
FullAdderr fulla0 (S[0], C[0], X[0], Y[0], 1'b0);
FullAdderr fulla1 (S[1], C[1], X[1], Y[1], C[0]);
FullAdderr fulla2 (S[2], C[2], X[2], Y[2], C[1]);
FullAdderr fulla3 (S[3], Cout, X[3], Y[3], C[2]);
endmodule

module BCD_out(out,dp,en,Cout,X,Y);

output Cout;
wire [3:0] S;

input [3:0] X, Y;
RippleCarryAdder rcl(Cout,S,X,Y);
output [6:0] out;
output dp;
input en;

assign dp=en;
    assign out =
  (S==0)?(7'b1000000):
    (S==1)?(7'b1111001):
    (S==2)?(7'b0100100):
    (S==3)?(7'b0110000):
    (S==4)?(7'b0011001):
    (S==5)?(7'b0010010):
    (S==6)?(7'b0000010):
    (S==7)?(7'b1111000):
    (S==8)?(7'b0000000):
    (S==9)?(7'b0011000):(7'b1111111);

endmodule
```

UCF File:

```
# PlanAhead Generated IO constraints

NET "out[6]" SLEW = FAST;
NET "out[5]" SLEW = FAST;
NET "out[4]" SLEW = FAST;
NET "out[3]" SLEW = FAST;
NET "out[2]" SLEW = FAST;
NET "out[1]" SLEW = FAST;
NET "out[0]" SLEW = FAST;

# PlanAhead Generated physical constraints

NET "out[6]" LOC = C6;
NET "out[5]" LOC = D6;
NET "out[4]" LOC = C5;
NET "out[3]" LOC = C4;
NET "out[2]" LOC = A4;
NET "out[1]" LOC = B4;
NET "out[0]" LOC = A3;
NET "dp" LOC = B3;

# PlanAhead Generated IO constraints

NET "X[3]" PULLUP;
NET "X[2]" PULLUP;
NET "X[1]" PULLUP;

NET "X[0]" PULLUP;
NET "Y[3]" PULLUP;
NET "Y[2]" PULLUP;
NET "Y[1]" PULLUP;
NET "Y[0]" PULLUP;

# PlanAhead Generated physical constraints

NET "Cout" LOC = T18;
NET "en" LOC = M18;
NET "Y[3]" LOC = D18;
NET "Y[2]" LOC = D17;
NET "Y[1]" LOC = C18;
NET "Y[0]" LOC = C17;
NET "X[3]" LOC = F17;
NET "X[2]" LOC = F18;
NET "X[1]" LOC = E16;
NET "X[0]" LOC = E18;

# PlanAhead Generated IO constraints

NET "en" PULLUP;
```

## I/O Pin Planning:

I/O Ports

| Name | Direction | Neg Diff Pair | Site | Fixed | Bank | I/O Std | Vcco | Vref | Drive Stre... | Slew T... | Pull Type | Off-Chip T... | IN_TERM | OUT_TERM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| All ports (18) | | | | | | | | | | | | | | |
| out (7) | Output | | | | | 0 default (LVCMOS25) | 2.500 | | 12 | FAST* | NONE | FP_VTT_50 | | NONE |
| out[6] | Output | | C6 | ☑ | | 0 default (LVCMOS25) | 2.500 | | 12 | FAST* | NONE | FP_VTT_50 | | NONE |
| out[5] | Output | | D6 | ☑ | | 0 default (LVCMOS25) | 2.500 | | 12 | FAST* | NONE | FP_VTT_50 | | NONE |
| out[4] | Output | | C5 | ☑ | | 0 default (LVCMOS25) | 2.500 | | 12 | FAST* | NONE | FP_VTT_50 | | NONE |
| out[3] | Output | | C4 | ☑ | | 0 default (LVCMOS25) | 2.500 | | 12 | FAST* | NONE | FP_VTT_50 | | NONE |
| out[2] | Output | | A4 | ☑ | | 0 default (LVCMOS25) | 2.500 | | 12 | FAST* | NONE | FP_VTT_50 | | NONE |
| out[1] | Output | | B4 | ☑ | | 0 default (LVCMOS25) | 2.500 | | 12 | FAST* | NONE | FP_VTT_50 | | NONE |
| out[0] | Output | | A3 | ☑ | | 0 default (LVCMOS25) | 2.500 | | 12 | FAST* | NONE | FP_VTT_50 | | NONE |
| X (4) | Input | | | | | 1 default (LVCMOS25) | | | | | PULLUP* | NONE | NONE | |
| X[3] | Input | | F17 | ☑ | | 1 default (LVCMOS25) | | | | | PULLUP* | NONE | NONE | |
| X[2] | Input | | F18 | ☑ | | 1 default (LVCMOS25) | | | | | PULLUP* | NONE | NONE | |
| X[1] | Input | | E16 | ☑ | | 1 default (LVCMOS25) | | | | | PULLUP* | NONE | NONE | |
| X[0] | Input | | E18 | ☑ | | 1 default (LVCMOS25) | | | | | PULLUP* | NONE | NONE | |
| Y (4) | Input | | | | | 1 default (LVCMOS25) | | | | | PULLUP* | NONE | NONE | |
| Y[3] | Input | | D18 | ☑ | | 1 default (LVCMOS25) | | | | | PULLUP* | NONE | NONE | |
| Y[2] | Input | | D17 | ☑ | | 1 default (LVCMOS25) | | | | | PULLUP* | NONE | NONE | |
| Y[1] | Input | | C18 | ☑ | | 1 default (LVCMOS25) | | | | | PULLUP* | NONE | NONE | |
| Y[0] | Input | | C17 | ☑ | | 1 default (LVCMOS25) | | | | | PULLUP* | NONE | NONE | |
| Scalar ports (3) | | | | | | | | | | | | | | |
| Cout | Output | | T18 | ☑ | | 1 default (LVCMOS25) | 2.500 | | 12 | SLOW | NONE | FP_VTT_50 | | NONE |
| dp | Output | | B3 | ☑ | | 0 default (LVCMOS25) | 2.500 | | 12 | SLOW | NONE | FP_VTT_50 | | NONE |
| en | Input | | M18 | ☑ | | 1 default (LVCMOS25) | | | | | PULLUP* | NONE | NONE | |