**BCD to Seven Segment Decoder**

**Lab no# 04**

****

Spring 2022

CSE-308L Digital Systems Design lab

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Dr: Ma’am Madeha sheer**

**April** 24, 2022

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**Objective:**

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

**Description:**

* Enable input in Spartan 6 is active low so we will keep it low (0) in order to turn on the particular seven segment display (LED).
* Similarly each segment of led is active low so we will keeps 0 that segments which we want to on.

**Task01:**

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

**Source Code:**

module BCD\_TO\_Seven\_Segment(in,en,out,enable1,enable2,enable3);

input [3:0]in;

input en; //input to enable1

output [6:0]out;

output enable1,enable2,enable3;

//enable2 and 3 are used for closing led2 and 3 which are not useable in this lab.

assign out=(in==4'b0000)? 7'b1000000 : //7'b(gfedcba) names of segments.

(in==4'b0001)? 7'b1111001 : //outputs are active low

(in==4'b0010)? 7'b0100100 : //nested if-else statements

(in==4'b0011)? 7'b0110000 :

(in==4'b0100)? 7'b0011001 :

(in==4'b0101)? 7'b0010010 :

(in==4'b0110)? 7'b0000010 :

(in==4'b0111)? 7'b1111000 :

(in==4'b1000)? 7'b0000000 :

(in==4'b1001)? 7'b0010000 : 7'b1111111;

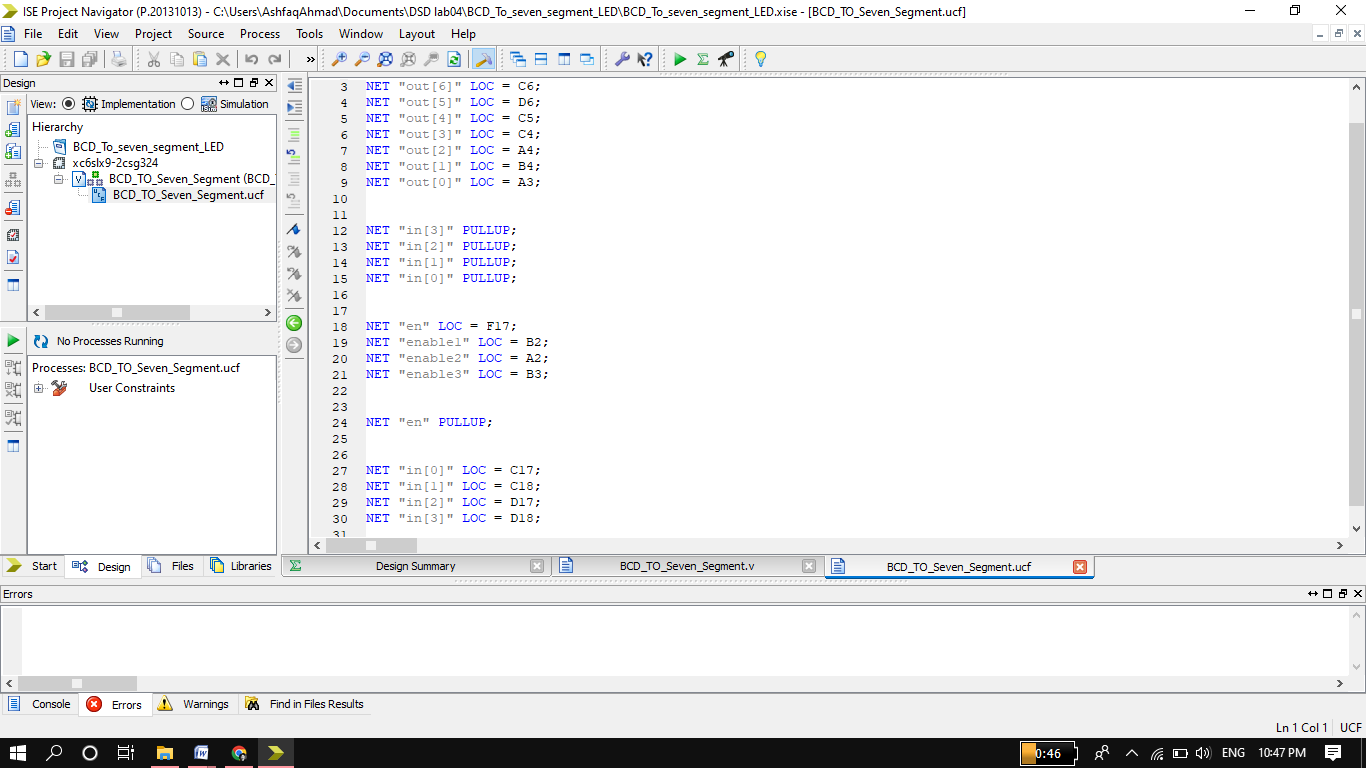
assign enable1=en; //on off led from circuit by switch.

assign enable2=1'b1; //off led 2

assign enable3=1'b1; //off led 3

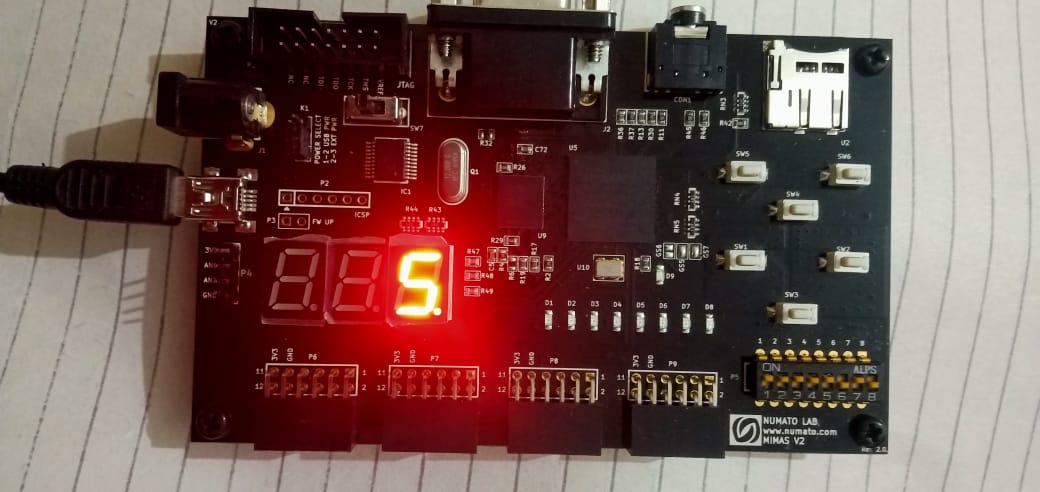
endmodule

**UCF file:**

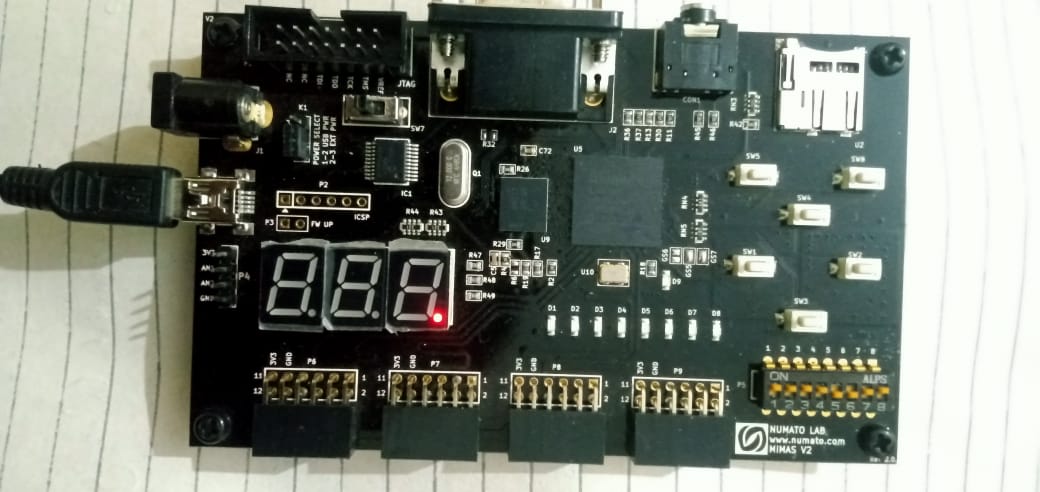


**Output:**

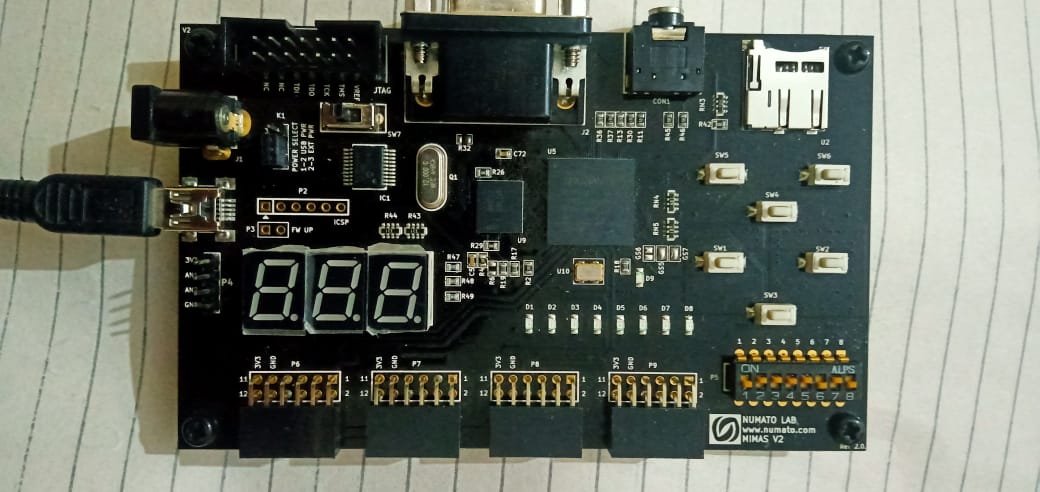
Pin 5,6,7 and 8 are input pins and pin 1 is enable pin. Pin 6 and 8 are set to 1 while pin 7 is set to 0 I,e (101==5 in decimal).



In this picture invalid input 1100 which is not BCD input so no led on.



In this picture Enable input (dip switch 1) is set to 1 so while enable input is active low input so the circuit becomes disable.



**Task02:**

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

**Source Code:**

**Full Adder:**

module FA(sum,cout,A,B,cin);

input A,B,cin;

output sum,cout;

wire w1,w2,w3;

xor (w1,A,B);

xor (sum,w1,cin);

and (w2,A,B);

and (w3,w1,cin);

or (cout,w2,w3);

endmodule

**4 Bits Binary Adder:**

module binary\_4\_Bits\_adder(out,enable1,enable2,enable3,sum,cout,A,B,en);

input [3:0] A,B;

input en;

output [3:0]sum;

output cout;

output [6:0]out;

output enable1,enable2,enable3;

wire [2:0]w;

FA f0(sum[0],w[0],A[0],B[0],1'b0);

FA f1(sum[1],w[1],A[1],B[1],w[0]);

FA f2(sum[2],w[2],A[2],B[2],w[1]);

FA f3(sum[3],cout,A[3],B[3],w[2]);

assign out =(sum==4'b0000)? 7'b1000000 :

(sum==4'b0001)? 7'b1111001 :

(sum==4'b0010)? 7'b0100100 :

(sum==4'b0011)? 7'b0110000 :

(sum==4'b0100)? 7'b0011001 :

(sum==4'b0101)? 7'b0010010 :

(sum==4'b0110)? 7'b0000010 :

(sum==4'b0111)? 7'b1111000 :

(sum==4'b1000)? 7'b0000000 :

(sum==4'b1001)? 7'b0010000 : 7'b1111111;

assign enable1=~en;

assign enable2=1'b1;

assign enable3=1'b1;

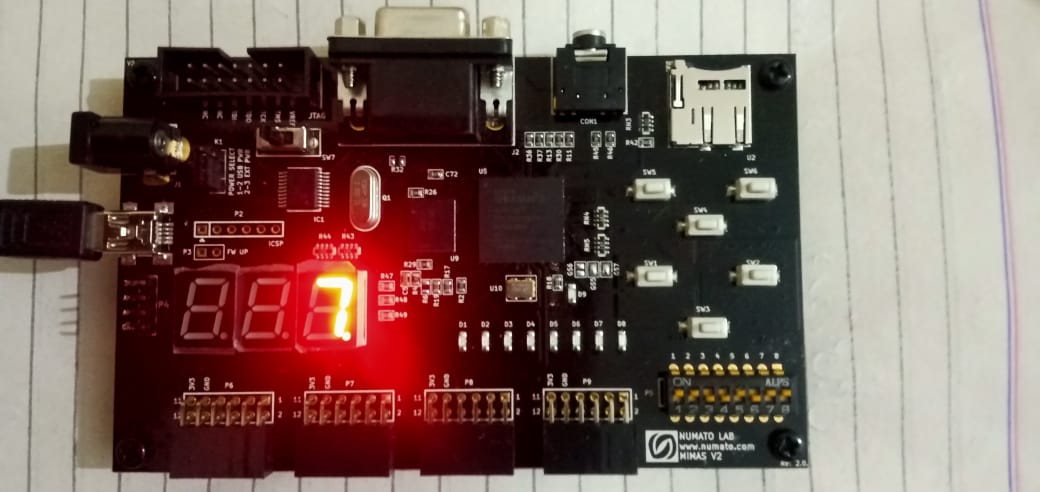
endmodule

**Output:**

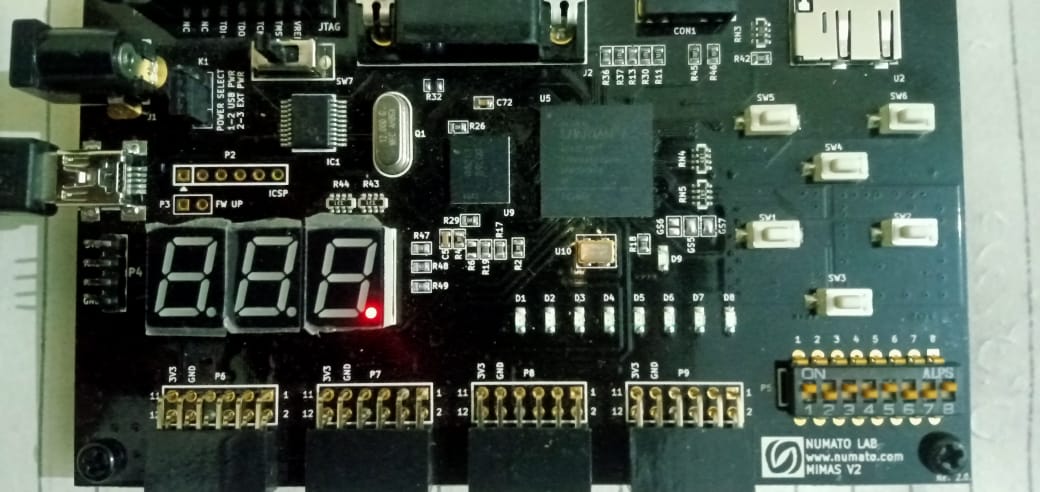
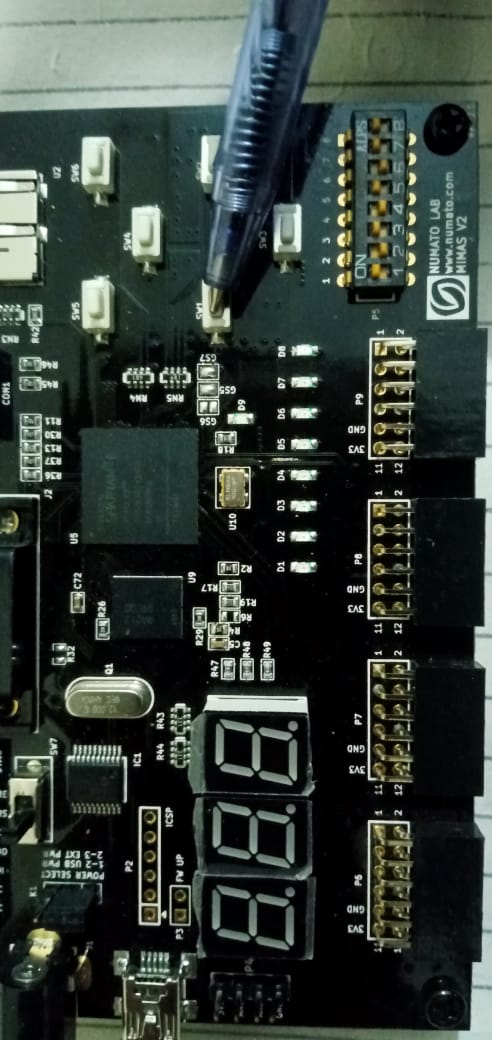
Bit 1 to 4 represent A. bit-4 is LSB.

Bit 5 to 8 represent B. bit-8 is LSB.

In given figure A=0011=3 and B=0100=4 so the result is 7.



In given figure A=0101=5 and B=1001=9. As the result is 1110=14 which is not in BCD range so LED is off.



Bush button 1 act as an enable. As it is active low so when button is not pressed its value is 1 the circuit will disable but I toggled the enable in program for keeping away from continuously pressing. So in given figure I pressed the button the value of button became 0 but I have toggled it in program so it again become 1 and due low active the circuit is disable.

**Task02:** (ModelSim)

**Source Code:**

module FA(sum,cout,A,B,cin);

input A,B,cin;

output sum,cout;

wire w1,w2,w3;

xor (w1,A,B);

xor (sum,w1,cin);

and (w2,A,B);

and (w3,w1,cin);

or (cout,w2,w3);

endmodule

module binary\_4\_Bits\_adder(out,en,sum,cout,A,B);

input [3:0] A,B;

output [3:0]sum;

output cout;

output [6:0]out;

output en;

wire [2:0]w;

FA f0(sum[0],w[0],A[0],B[0],1'b0);

FA f1(sum[1],w[1],A[1],B[1],w[0]);

FA f2(sum[2],w[2],A[2],B[2],w[1]);

FA f3(sum[3],cout,A[3],B[3],w[2]);

assign {en,out} =(sum==4'b0000)? 8'b10111111 :

(sum==4'b0001)? 8'b10000110 :

(sum==4'b0010)? 8'b11011011 :

(sum==4'b0011)? 8'b11001111 :

(sum==4'b0100)? 8'b11100110 :

(sum==4'b0101)? 8'b11101101 :

(sum==4'b0110)? 8'b11111101 :

(sum==4'b0111)? 8'b10000111 :

(sum==4'b1000)? 8'b11111111 :

(sum==4'b1001)? 8'b11101111 : 8'b00000000;

endmodule

**Test Bench:**

module test\_4\_bits\_adder;

reg [3:0]a,b;

wire en;

wire [3:0]sum;

wire [6:0]out;

wire cout;

binary\_4\_Bits\_adder td(out,en,sum,cout,a,b);

initial

begin

$display("sum en g f e d c b a");

$monitor("%b %b %b %b %b %b %b %b %b",sum,en,out[6],out[5],out[4],out[3],out[2],out[2],out[0]);

#5 a=4'b0000; b=4'b0001;

#5 a=4'b0100; b=4'b0011;

#5 a=4'b0010; b=4'b0001;

#5 a=4'b0010; b=4'b0101;

#5 a=4'b0011; b=4'b1001;

#5 a=4'b0101; b=4'b0100;

#5 a=4'b1100; b=4'b0011;

#5 a=4'b1110; b=4'b0011;

#5 a=4'b0100; b=4'b0100;

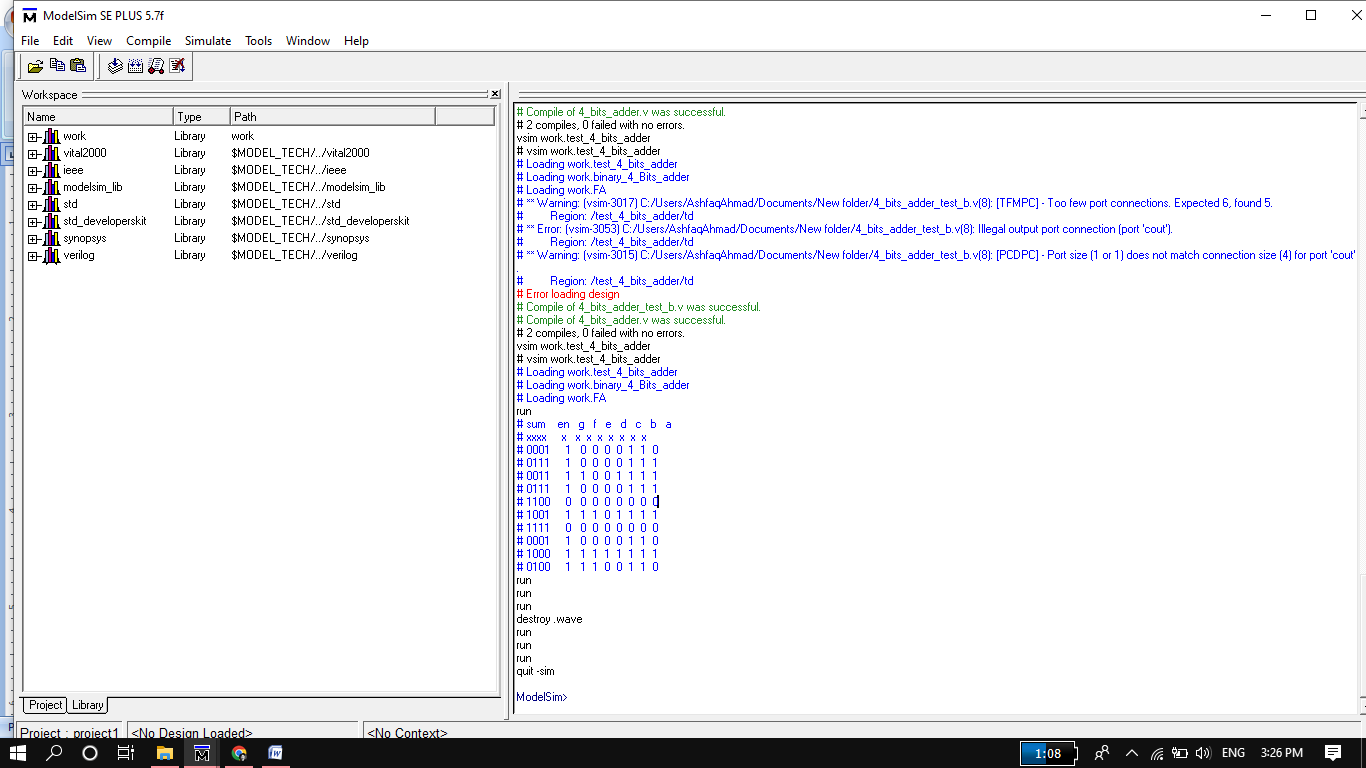
#5 a=4'b0011; b=4'b0001;

end

endmodule

**Truth Table:**

The output (a-to-g) shows the segments of LED. When the value of sum exceed from 9 (1001) the segments of LED become off and enable also become zero.



**Wave Form:**

