LAB 4

INTRODUCTION TO XILINX ISE AND Spartan 6 BOARD and implementation of MUX and DECODERS

Spring 2025

Submitted by: **Mohsin Sajjad**

Registration No: **22pwsce2149**

Class Section: **A**

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

Mohsin Sajjad

Student Signature: _____

Submitted to:

Engr. Faheem Jan

Month Day, Year (16 03, 2025)

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Objectives:

- Introduction to FPGA
- Introuction Xilinx ISE

Lab Task:

Task 01:

Develop a program to control on Board LED using On board available switch.

Code:

```
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module buffer(O,I);
input [7:0] I;
output [7:0] O;
assign O=I;
endmodule
```

Output:

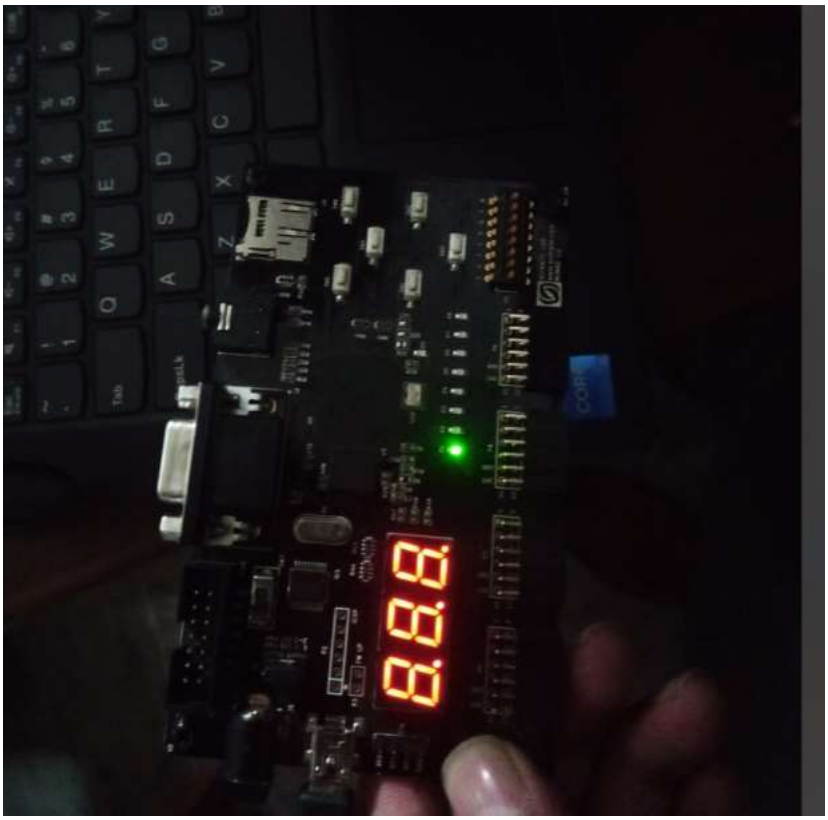
Task 02:

Develop a program that implements a 2x1 multiplexer on the board. Connect the inputs to switches and output to led.

Code:

```
20  //////////////////////////////////////////////////////////////////////////////////
21  module mux2X1(I, sel, out);
22      input [1:0] I;  // 2 input lines
23      input sel;      // Selection line
24      output out;
25      assign out = sel ? I[1] : I[0];  // MUX selects I[1] when sel=1, I[0] when sel=0
26  endmodule
```

Output:



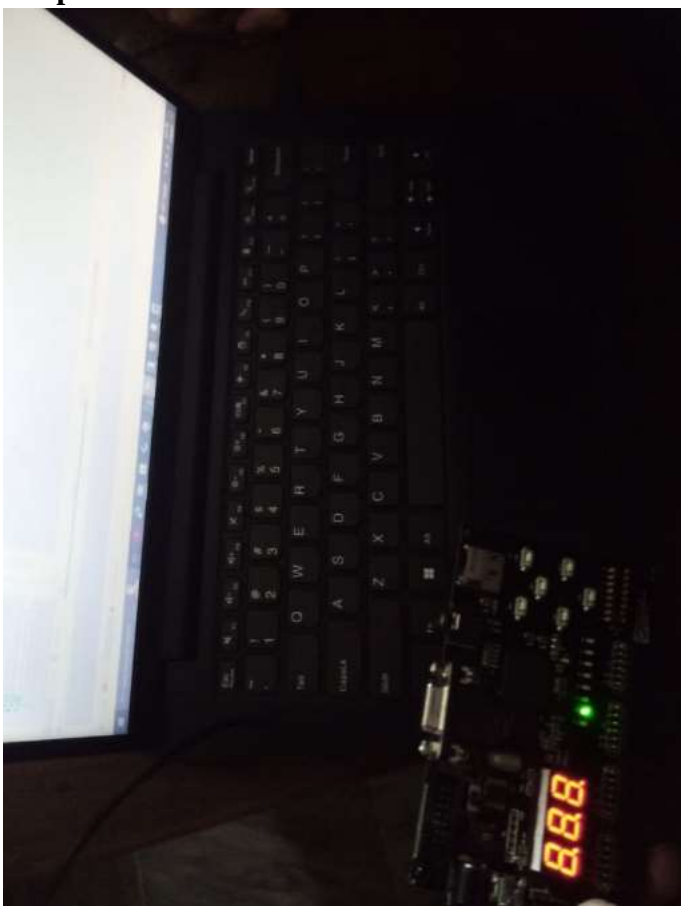
Task 03:

Develop a program that implements a 4x1 multiplexer on the board. Connect the inputs to switches and output to led.

Code:

```
20  //////////////////////////////////////////////////////////////////////////////////
21  module mux4x1(I, sel, out);
22      input [3:0] I;   // 4 input lines
23      input [1:0] sel; // 2-bit selection line
24      output reg out;  // Output declared as reg for procedural assignment
25      always @(*) begin
26          if (sel == 2'b00)
27              out = I[0];
28          else if (sel == 2'b01)
29              out = I[1];
30          else if (sel == 2'b10)
31              out = I[2];
32          else
33              out = I[3];
34      end
35  endmodule
```

Output:



Task 04:

Implement 2X4 Decoder and 3x8 Decoder on the Board.

2X4:

Code:

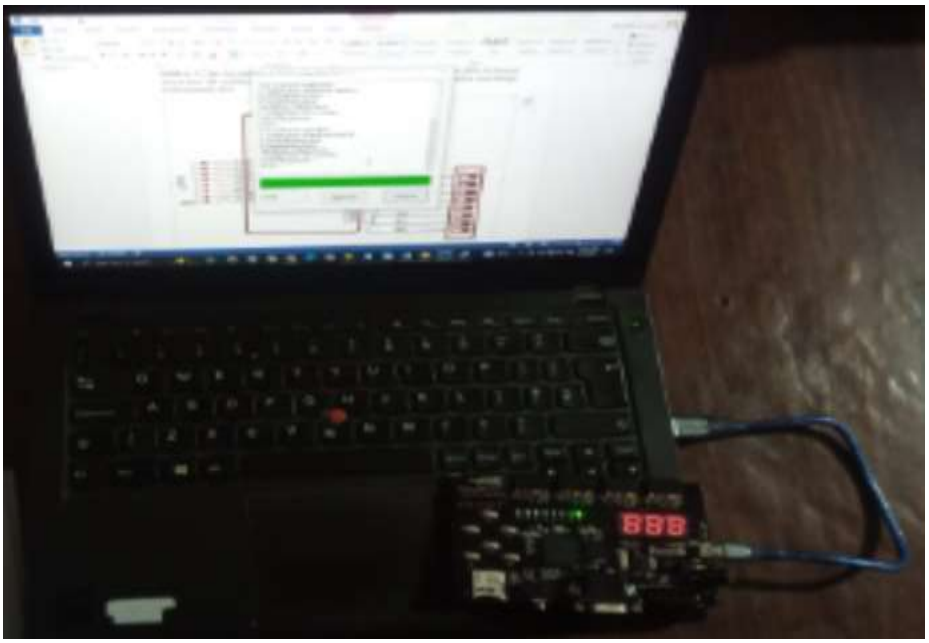
```
20  //////////////////////////////////////////////////////////////////////////////////
21  module decoder2X4(A, B, D);
22      input A, B;
23      output [3:0] D;
24
25      assign D[0] = ~A & ~B;
26      assign D[1] = ~A & B;
27      assign D[2] = A & ~B;
28      assign D[3] = A & B;
29  endmodule
```

3X8:

Code:

```
20  //////////////////////////////////////////////////////////////////////////////////
21  module decoder3x8(A, B, C, D);
22      input A, B, C;  // 3 input lines
23      output [7:0] D; // 8 output lines
24
25      assign D[0] = ~A & ~B & ~C;
26      assign D[1] = ~A & ~B & C;
27      assign D[2] = ~A & B & ~C;
28      assign D[3] = ~A & B & C;
29      assign D[4] = A & ~B & ~C;
30      assign D[5] = A & ~B & C;
31      assign D[6] = A & B & ~C;
32      assign D[7] = A & B & C;
33  endmodule
```

Output:



Task 05:

Implement NOT gate using MUX

Code:

```
20  //////////////////////////////////////////////////////////////////////////////////
21  module not_using_mux(A, Y);
22      input A;      // Input
23      output Y;     // Output
24
25      assign Y = A ? 1'b0 : 1'b1;  // MUX functionality: Y = (A) ? 0 : 1
26  endmodule
```

Output on Xilinx: