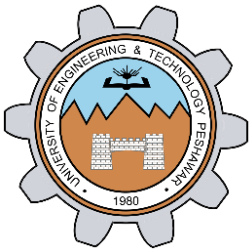
INTRODUCTION TO XILINX ISE AND S3BOARD

LAB # 01



Spring 2021

CSE-308L Digital System Design Lab

Submitted by: **ASHIQ ULLAH**

Registration No. : **18PWCSE1695**

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:

Engr. Madiha Sher

April 17, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

OBJECTIVES:

Introduction to FPGA and Xilinx

TASK01:

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

CODE:

```
ln #
 1 module buffer(A0,A1,A2,A3,A4,A5,A6,A7,O1,O2,O3,O4,O5,O6,O7,O8);
 2          input A0,A1,A2,A3,A4,A5,A6,A7;
 3          output O1,O2,O3,O4,O5,O6,O7,O8;
 4          buf b0(O1,A0);
 5          buf b1(O2,A1);
 6          buf b2(O3,A2);
 7          buf b3(O4,A3);
 8          buf b4(O5,A4);
 9          buf b5(O6,A5);
10          buf b6(O7,A6);
11          buf b7(O8,A7);
12 endmodule
13
```

TestBench:

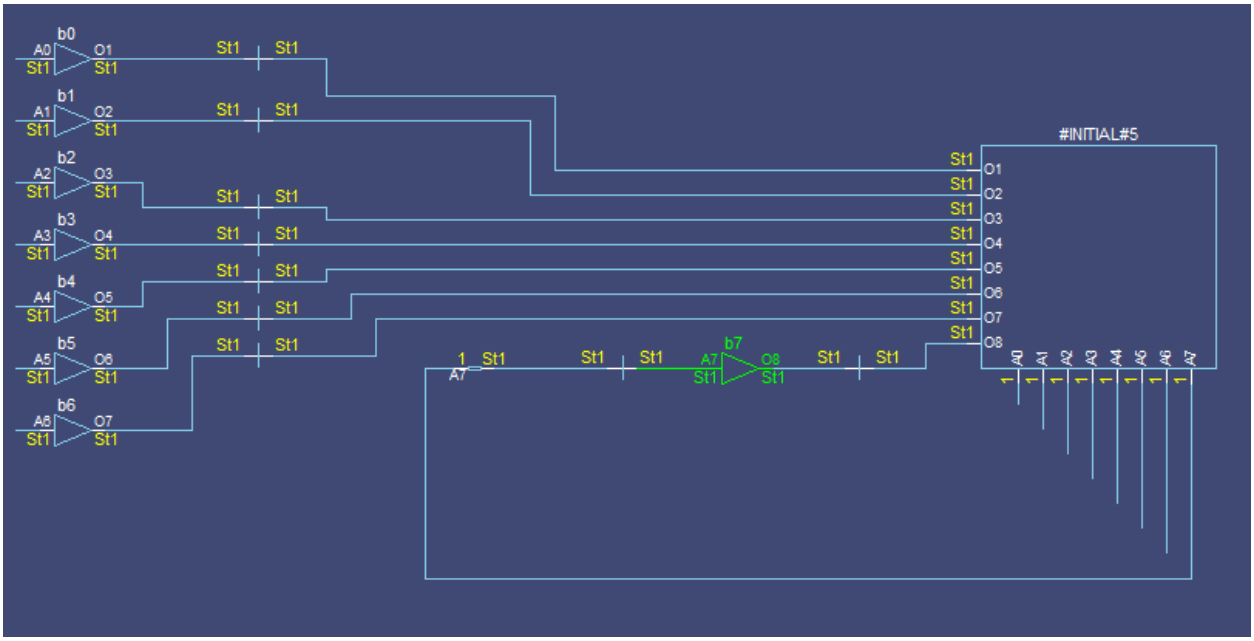
```
ln #                                                                    E:/6TH SEMESTER/DSD LAB/Lab1/Buffer_test.v
 1 module test_buff;
 2         reg A0,A1,A2,A3,A4,A5,A6,A7;
 3         wire O1,O2,O3,O4,O5,O6,O7,O8;
 4         buffer Buf1(A0,A1,A2,A3,A4,A5,A6,A7,O1,O2,O3,O4,O5,O6,O7,O8);
 5         initial
 6         begin
 7                 $display("       Inputs        |       Outputs");
 8                 A0 = 0; A1 = 0; A2 = 0; A3 = 0;
 9                 A4 = 0; A5 = 0; A6 = 0; A7 = 0;
10                 #5;
11                 $monitor ("%b %b %b %b %b %b %b %b | %b %b %b %b %b %b %b %b",A0,A1,A2,A3,A4,A5,A6,A7,O1,O2,O3,O4,O5,O6,O7,O8);
12                 A0 = 0; A1 = 1; A2 = 0; A3 = 1;
13                 A4 = 0; A5 = 1; A6 = 0; A7 = 1;
14                 #5;
15
16                 A0 = 0; A1 = 0; A2 = 1; A3 = 1;
17                 A4 = 0; A5 = 0; A6 = 1; A7 = 1;
18                 #5;
19
20                 A0 = 1; A1 = 1; A2 = 0; A3 = 0;
21                 A4 = 1; A5 = 1; A6 = 0; A7 = 0;
22                 #5;
23
24                 A0 = 0; A1 = 0; A2 = 1; A3 = 1;
25                 A4 = 1; A5 = 1; A6 = 0; A7 = 0;
26                 #5;
27
28                 A0 = 1; A1 = 1; A2 = 0; A3 = 0;
29                 A4 = 0; A5 = 0; A6 = 1; A7 = 1;
30                 #5;
31
32                 A0 = 1; A1 = 1; A2 = 1; A3 = 1;
33                 A4 = 0; A5 = 0; A6 = 0; A7 = 0;
34                 #5;
35
36                 A0 = 0; A1 = 0; A2 = 0; A3 = 0;
37                 A4 = 1; A5 = 1; A6 = 1; A7 = 1;
38                 #5;
39
40                 A0 = 1; A1 = 1; A2 = 1; A3 = 1;
41                 A4 = 1; A5 = 1; A6 = 1; A7 = 1;
42                 #5;
43         end
44 endmodule
45
```

OUTPUTS

Truth Table:

```
#     Inputs     |    Outputs
#01010101|01010101
#00110011|00110011
#11001100|11001100
#00111100|00111100
#11000011|11000011
#11110000|11110000
#00001111|00001111
#11111111|11111111
```

Circuit Design:



TASK02:

a) Develop a program that implements a 2x1 multiplexer using gate level modeling.
   i) Simulate the multiplexer with a test bench.

CODE:

```
ln #
 1 module mux_2to1(a,b,sel,f);
 2          input a,b,sel;
 3          output f;
 4          wire nsel,f1,f2;
 5          not g1(nsel,sel);
 6          and g2(f1,a,nsel);
 7          and g3(f2,b,sel);
 8          or g4(f,f1,f2);
 9 endmodule
10
11
```

TestBench:

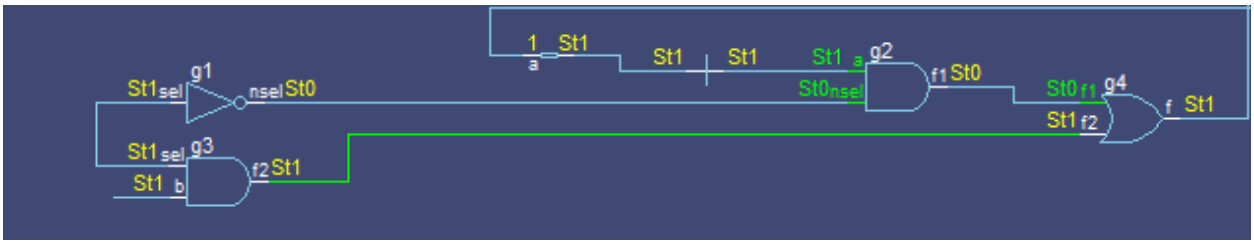
```
ln #                                                    E:/6TH SE
 1 module test_2to1();
 2         reg a,b,sel;
 3         wire f;
 4         mux_2to1 M(a,b,sel,f);
 5         initial
 6         begin
 7                 $display("A B SEL F");
 8                 a=0;
 9                 b=0;
10                 sel=0;
11                 $monitor ("%b %b %b %b",a,b,sel,f);
12                 #5;
13                 a=0; b=0; sel=1;
14                 #5;
15
16                 a=0; b=1; sel=0;
17                 #5;
18
19                 a=0; b=1; sel=1;
20                 #5;
21
22                 a=1; b=0; sel=0;
23                 #5;
24
25                 a=1; b=0; sel=1;
26                 #5;
27
28                 a=1; b=1; sel=0;
29                 #5;
30
31                 a=1; b=1; sel=1;
32                 #5;
33         end
34 endmodule
35
```

OUTPUTS

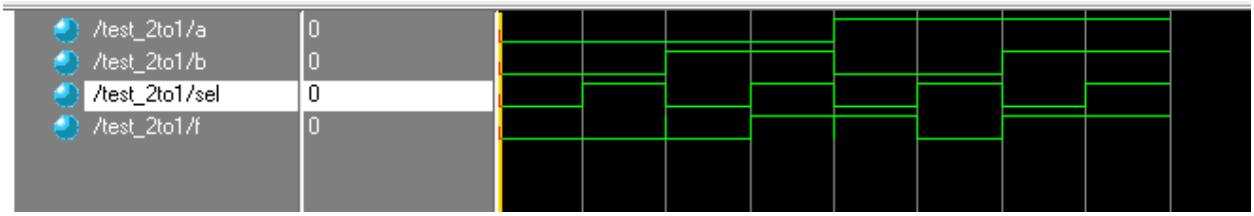
Truth Table:

```
# A B SEL F
# 0 0 0 0
# 0 0 1 0
# 0 1 0 0
# 0 1 1 1
# 1 0 0 1
# 1 0 1 0
# 1 1 0 1
# 1 1 1 1
```

Circuit Design:



Wave:



b) Implement 4x1 mux using modelsim.

i) Simulate the multiplexer with a test bench.

CODE:

```
ln #
 1 module mux_4to1(s1,s2,d3,d2,d1,d0,f);
 2         input s1,s2,d3,d2,d1,d0;
 3         output f;
 4         wire ns1,ns2,a0,a1,a2,a3;
 5         not n1(ns1,s1);
 6         not n2(ns2,s2);
 7         and g1(a0,ns1,ns2,d0);
 8         and g2(a1,ns1,s2,d1);
 9         and g3(a2,s1,ns2,d2);
10         and g4(a3,s1,s2,d3);
11         or o1(f,a0,a1,a2,a3);
12 endmodule
```

TestBench:

```
module test_4to1;
        reg s1,s2,d3,d2,d1,d0;
        wire f;
        mux_4to1 multi(s1,s2,d3,d2,d1,d0,f);
        initial
        begin
                $display("SEL1 SEL2 D3 D2 D1 D0 f");

                s1=0;
                s2=0;
                d3=0;d2=1;d1=0;d0=0;
                #1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
                #5;

                s1=0;
                s2=0;
```

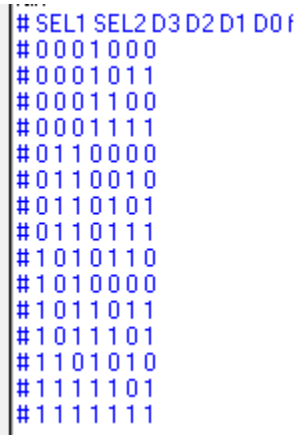
```
d3=0;d2=1;d1=0;d0=1;
#1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
#5;
s1=0;
s2=0;
d3=0;d2=1;d1=1;d0=0;
#1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
#5;
s1=0;
s2=0;
d3=0;d2=1;d1=1;d0=1;
#1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
#5;
s1=0;
s2=1;
d3=1;d2=0;d1=0;d0=0;
#1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
#5;
s1=0;
s2=1;
d3=1;d2=0;d1=0;d0=1;
#1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
#5;
s1=0;
s2=1;
d3=1;d2=0;d1=1;d0=0;
#1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
#5;
```

```
s1=0;
s2=1;
d3=1;d2=0;d1=1;d0=1;
#1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
#5;
s1=1;
s2=0;
d3=1;d2=0;d1=1;d0=1;
#1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
#5;
s1=1;
s2=0;
d3=1;d2=0;d1=0;d0=0;
#1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
#5;
s1=1;
s2=0;
d3=1;d2=1;d1=0;d0=1;
#1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
#5;
s1=1;
s2=0;
d3=1;d2=1;d1=1;d0=0;
#1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
#5;
s1=1;
s2=1;
d3=0;d2=1;d1=0;d0=1;
```

```
        #1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
        #5;
        s1=1;
        s2=1;
        d3=1;d2=1;d1=1;d0=0;
        #1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
        #5;
        s1=1;
        s2=1;
        d3=1;d2=1;d1=1;d0=1;
        #1 $display("%b %b %b %b %b %b %b",s1,s2,d3,d2,d1,d0,f);
        #5;
    end
endmodule
```

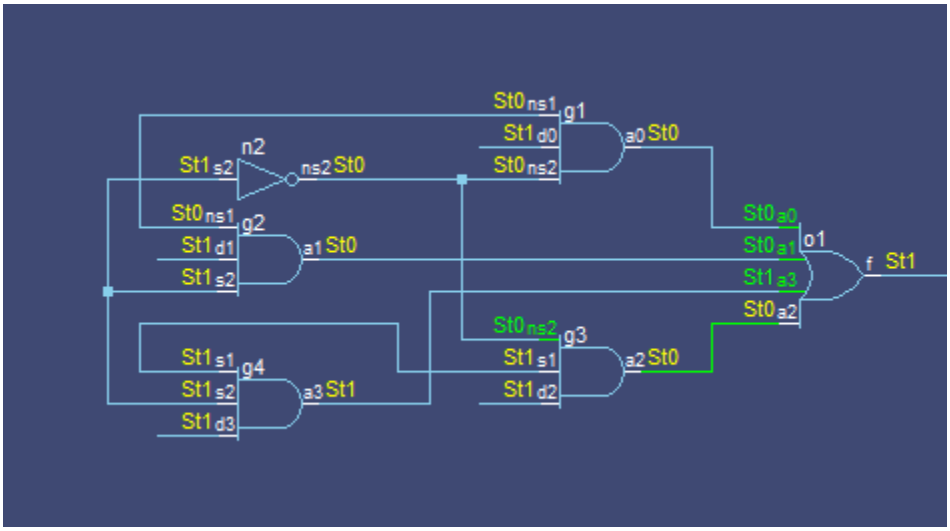
OUTPUTS

Truth Table:



```
# SEL1 SEL2 D3 D2 D1 D0 f
# 0 0 0 1 0 0 0
# 0 0 0 1 0 1 1
# 0 0 0 1 1 0 0
# 0 0 0 1 1 1 1
# 0 1 1 0 0 0 0
# 0 1 1 0 0 1 0
# 0 1 1 0 1 0 1
# 0 1 1 0 1 1 1
# 1 0 1 0 1 1 0
# 1 0 1 0 0 0 0
# 1 0 1 1 0 1 1
# 1 0 1 1 1 0 1
# 1 1 0 1 0 1 0
# 1 1 1 1 1 0 1
# 1 1 1 1 1 1 1
```

Circuit Design:



Wave: