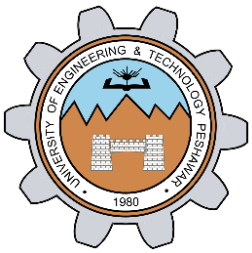
DSD LAB REPORT

LAB # 03



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

May 1, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

OBJECTIVES:

This lab will enable students to:

- Learn top down and bottom up design methodologies
- Use seven segment display available on the S3board
- Data flow level modeling

TASK:

BCD to Seven Segment Decoder

CODE01 (Active Low):

```
module BCD_Segment(A, dp, Out);
        input [0:3] A;
        output [0:6] Out;
        output dp;

        assign Out[0] = (~A[0]&~A[1]&~A[2]&A[3]) | (A[1]&~A[2]&~A[3])
                                                 | (A[0]&A[2]) | (A[0]&A[1]);

        assign Out[1] = (A[1]&A[2]&~A[3]) | (A[1]&~A[2]&A[3]) | (A[0]&A[2])
                                                              | (A[0]&A[1]);

        assign Out[2] = (~A[1]&A[2]&~A[3]) | (A[0]&A[2]) | (A[0]&A[1]);

        assign Out[3] = (A[1]&~A[2]&~A[3]) | (~A[0]&~A[1]&~A[2]&A[3])
                       | (A[1]&A[2]&A[3]) | (A[0]&A[2]) | (A[0]&A[1]);

        assign Out[4] = A[3] | (A[1]&~A[2]) | (A[0]&A[2]) | (A[0]&A[1]);

        assign Out[5] = (~A[1]&A[2]) | (~A[0]&~A[1]&A[3]) | (A[0]&A[2])
                                     | (A[0]&A[1]) | (A[2]&A[3]);

        assign Out[6] = (~A[0]&~A[1]&~A[2]) | (A[1]&A[2]&A[3])
                                            | (A[0]&A[2]) | (A[0]&A[1]);

        assign dp = ~((A[0]&A[1]) | (A[0]&A[2]));

endmodule
```

- A [0] = Most Significant bit
- A [3] = Least Significant bit

TestBench:

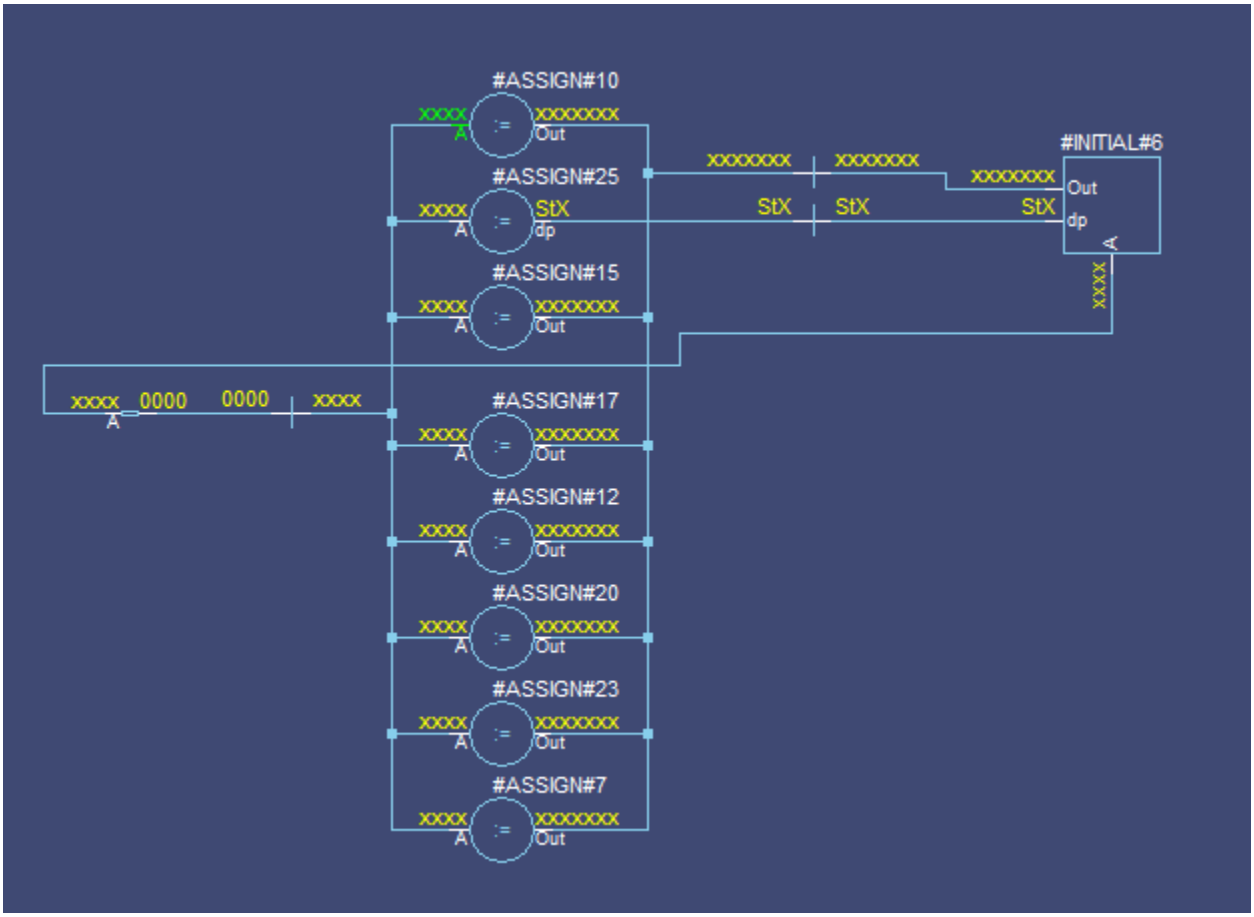
```
module test_segment;
        reg [0:3] A;
        wire [0:6] Out;
        wire dp;
        BCD_Segment BCD(A, dp, Out);
        initial
        begin
                $display ("Inputs  |  Outputs    |  dp");
                A = 4'b0000;
                $monitor ("%b    |  %b  |       %b",A,Out,dp);
                #5;
                A = 4'b0001;
                #5;
                A = 4'b0010;
                #5;
                A = 4'b0011;
                #5;
                A = 4'b0100;
                #5;
                #5;
                A = 4'b0101;
                #5;
                A = 4'b0110;
                #5;
                A = 4'b0111;
                #5;
                A = 4'b1000;
                #5;
                A = 4'b1001;
                #5;
                A = 4'b1010;
                #5;
                A = 4'b1011;
                #5;
                A = 4'b1100;
                #5;
                A = 4'b1101;
                #5;
                A = 4'b1110;
                #5;
                A = 4'b1111;

        end
endmodule
```

OUTPUTS

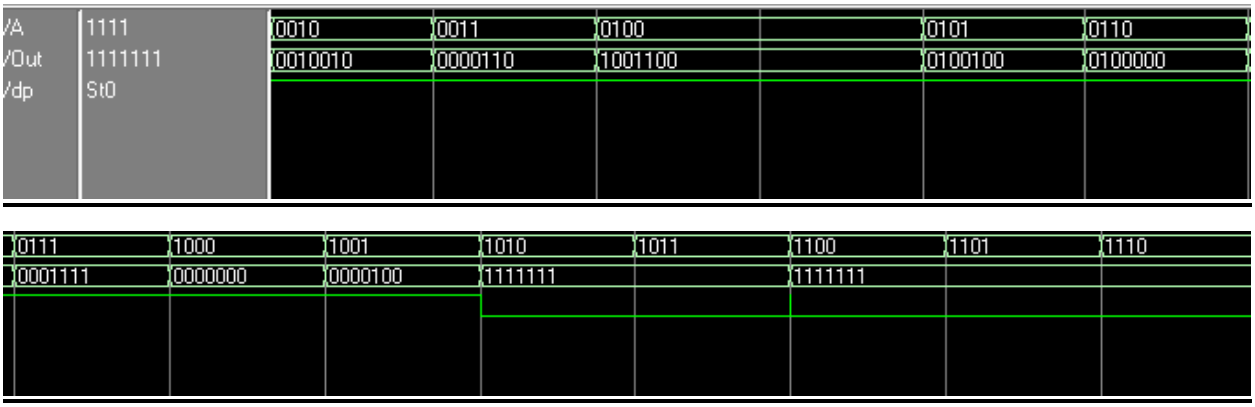
Truth Table:

```
run
# Inputs | Outputs  | dp
# 0000   | 0000001 | 1
# 0001   | 1001111 | 1
# 0010   | 0010010 | 1
# 0011   | 0000110 | 1
# 0100   | 1001100 | 1
# 0101   | 0100100 | 1
# 0110   | 0100000 | 1
# 0111   | 0001111 | 1
# 1000   | 0000000 | 1
# 1001   | 0000100 | 1
# 1010   | 1111111 | 0
# 1011   | 1111111 | 0
# 1100   | 1111111 | 0
# 1101   | 1111111 | 0
# 1110   | 1111111 | 0
# 1111   | 1111111 | 0
```

Data Flow:



Wave Form:



CODE02 (Perform in Lab):

```
ln #
 1 module sev_seg (A, Out, Dp);
 2          input [3:0] A;
 3          output [6:0] Out;
 4          output Dp;
 5          assign {Dp,Out} = (A==4'b0000)?(8'b01111110):
 6                            (A==4'b0001)?(8'b00110000):
 7                            (A==4'b0010)?(8'b01101101):
 8                            (A==4'b0011)?(8'b01111001):
 9                            (A==4'b0100)?(8'b00110011):
10                            (A==4'b0101)?(8'b01011011):
11                            (A==4'b0110)?(8'b01011111):
12                            (A==4'b0111)?(8'b01110000):
13                            (A==4'b1000)?(8'b01111111):
14                            (A==4'b1001)?(8'b01111011):
15                                         (8'b10000000);
16 endmodule
```

TestBench:

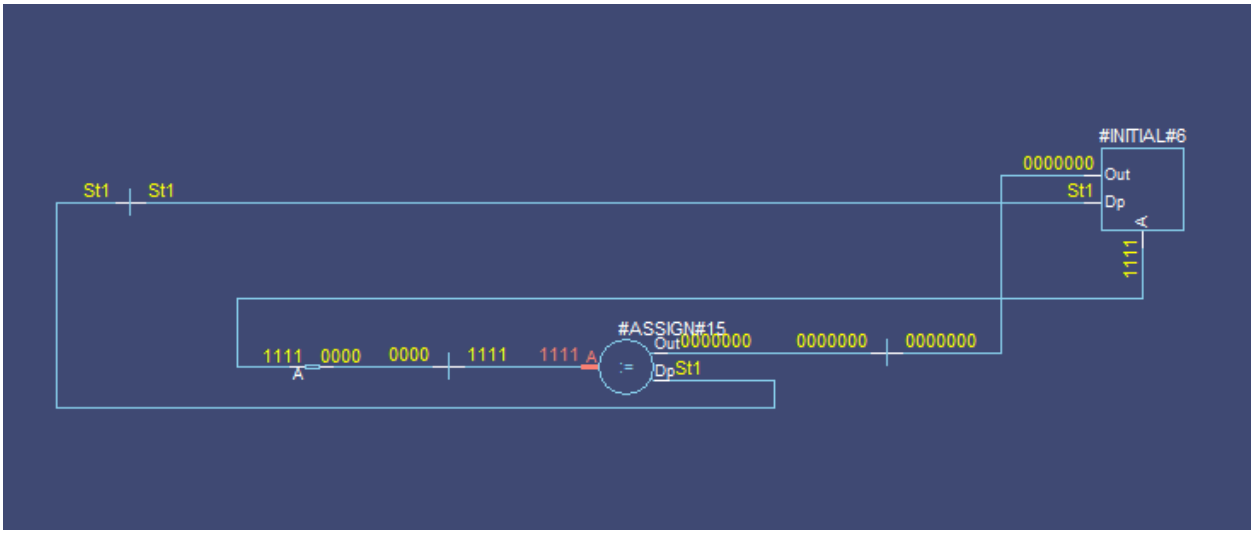
```
 1 module test_7;
 2         reg [3:0] A;
 3         wire [6:0] Out;
 4         wire Dp;
 5         sev_seg BCD2(A, Out, Dp);
 6         initial
 7         begin
 8                 $display ("Inputs  |  Outputs    |  Dp");
 9                 A = 4'b0000;
10                 $monitor ("%b    |  %b  |       %b",A,Out,Dp);
11                 #5;
12                 A = 4'b0001;
13                 #5;
14                 A = 4'b0010;
15                 #5;
16                 A = 4'b0011;
17                 #5;
18                 A = 4'b0100;
19                 #5;
20                 #5;
21                 A = 4'b0101;
22                 #5;
23                 A = 4'b0110;
24                 #5;
25                 A = 4'b0111;
26                 #5;
27                 A = 4'b1000;
28                 #5;
29                 A = 4'b1001;
30                 #5;
31                 A = 4'b1010;
32                 #5;
33                 A = 4'b1011;
34                 #5;
35                 A = 4'b1100;
36                 #5;
37                 A = 4'b1101;
38                 #5;
39                 A = 4'b1110;
40                 #5;
41                 A = 4'b1111;
42
43         end
44 endmodule
```

OUTPUTS

Truth Table:

# Inputs	Outputs	Dp
# 0000	1111110	0
# 0001	0110000	0
# 0010	1101101	0
# 0011	1111001	0
# 0100	0110011	0
# 0101	1011011	0
# 0110	1011111	0
# 0111	1110000	0
# 1000	1111111	0
# 1001	1111011	0
# 1010	0000000	1
# 1011	0000000	1
# 1100	0000000	1
# 1101	0000000	1
# 1110	0000000	1
# 1111	0000000	1

Data Flow:



Wave Form: