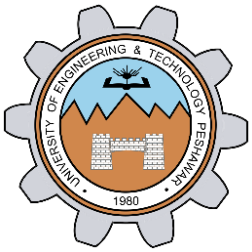
DIGITAL SYSTEM DESIGN LAB

LAB #09



Spring 2021

CSE308L DSD LAB

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

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

Saturday, July 10, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Task # 01:

Implement 4-bit binary counter and display the result on LEDs and 7-segment display.

Code:

Counter:

```
module Counter(clk,rst,dp,out,count);
	input clk,rst;
	output [6:0] out;
	output dp;
	output reg [3:0] count = 0;
	wire oclk;

	BINARYto7SEG c1(count,out,dp);

	Divider d1(clk,oclk,rst);

	always @(posedge oclk)
		if(rst==0)
			count =0;
		else
			count = count + 1;

endmodule
```

Divider:

```
module Divider(input iclk,output reg oclk, input rst);
	reg [100:0] count;
	always @(posedge iclk)
		if(rst==0)
		begin
			oclk = 0;
			count = 0;
		end
		else
		begin
			count = count+1;
			if(count==100000000)
			begin
				oclk = ~oclk;
				count = 0;
			end
		end

endmodule
```

BINARYto7SEG:

```
module BINARYto7SEG (A, Out, Dp);
        input [3:0] A;
        output [6:0] Out;
        output Dp;

                assign {Dp,Out} = (A==4'b0000)?(8'b00000001):
                                  (A==4'b0001)?(8'b01001111):
                                  (A==4'b0010)?(8'b00010010):
                                  (A==4'b0011)?(8'b00000110):
                                  (A==4'b0100)?(8'b01001100):
                                  (A==4'b0101)?(8'b00100100):
                                  (A==4'b0110)?(8'b00100000):
                                  (A==4'b0111)?(8'b00001111):
                                  (A==4'b1000)?(8'b00000000):
                                  (A==4'b1001)?(8'b00000100):
                                  (A==4'b1010)?(8'b00001000):
                                  (A==4'b1011)?(8'b01100000):
                                  (A==4'b1100)?(8'b00110001):
                                  (A==4'b1101)?(8'b01000010):
                                  (A==4'b1110)?(8'b00110000):
                                  (A==4'b1111)?(8'b00111000): (8'b10000000);
endmodule
```

UCF File:

```
net "rst" LOC =m18 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST | PULLUP;
net "clk" LOC =v10 | IOSTANDARD = LVCMOS33 | period = 100MHz;

net "count[0]" LOC =P16 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "count[1]" LOC =N15 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "count[2]" LOC =N16 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "count[3]" LOC =U17 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;


net "dp" LOC =B3 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "dp" LOC =A5 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "out[0]" LOC =C6 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "out[1]" LOC =D6 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "out[2]" LOC =C5 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "out[3]" LOC =C4 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "out[4]" LOC =A4 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "out[5]" LOC =B4 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
net "out[6]" LOC =A3 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
```

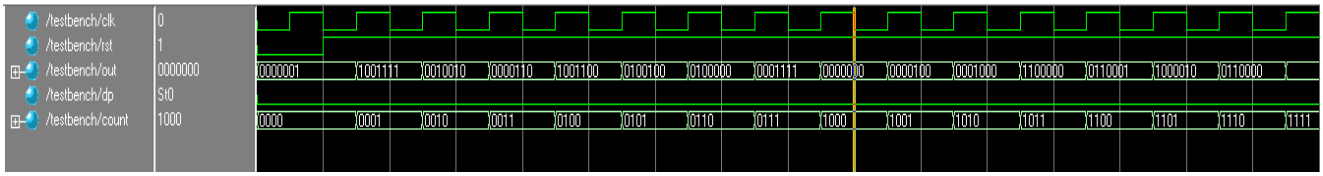
TestBench:

```
module testbench;
        reg clk,rst;
        wire [6:0] out;
        wire dp;
        wire [3:0] count;

        Counter c1(clk,rst,dp,out,count);

        always
                #5 clk = ~clk;

        initial
        begin
                $display("RST   Count");
                clk = 0;
                rst = 0;
                #10
                rst = 1;
                $monitor("%b    %d",rst,count);
        end
endmodule
```

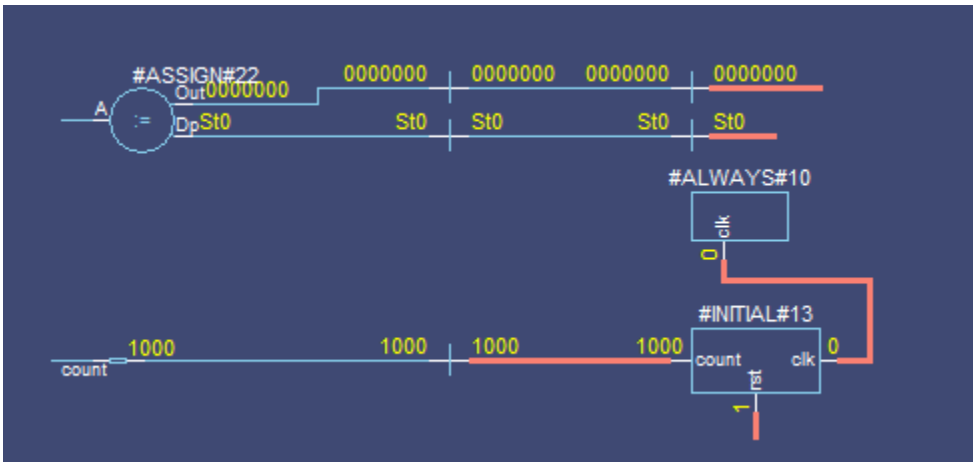
Output:

```
# RST   Count
# 1    0
# 1    1
# 1    2
# 1    3
# 1    4
# 1    5
# 1    6
# 1    7
# 1    8
# 1    9
# 1   10
# 1   11
# 1   12
# 1   13
# 1   14
# 1   15
```

Waveform:



Dataflow:


#ASSIGN#22
A
:=
Out0000000
Dp St0
0000000
0000000
0000000
0000000
St0
St0
St0
St0
#ALWAYS#10
clk
0
#INITIAL#13
count
1000
1000
1000
1000
count
clk
0
1st
1