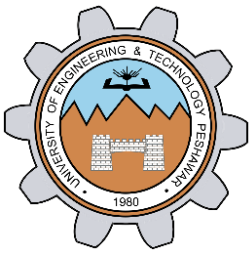
Binary Counter

LAB # 09



Spring 2021

CSE308L Digital System Design Lab

Submitted by: **Hurair M**

Registration No.: **18PWCSE1657**

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 10th, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Task:

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

Code:

```
`timescale 1ns / 1ps

module clock_div(clk_in, rst, clk_out);
   input clk_in, rst;
        output clk_out;
        reg clk_out;
        reg [39:0] count;
        always@(posedge clk_in)
        begin
           if(~rst)
                begin
                   clk_out = 0;
                                count = 0;
                end
                else
                begin
                   count = count + 1;
                   if(count == 100000000)
                                begin
                                   clk_out = ~clk_out;
                                        count = 0;
                                end
                end
        end
endmodule
```

```
module bcdtossd(output dp, output [6:0]s, input [3:0]bcd);


assign {dp, s} = (bcd == 4'b1001)?8'b10000100:(bcd == 4'b1000)?8'b10000000:(bcd == 4'b0111)?8'b10001111:
            (bcd == 4'b0110)?8'b10100000:(bcd == 4'b0101)?8'b10100100:(bcd == 4'b0100)?8'b11001100:
            (bcd == 4'b0011)?8'b10001100:(bcd == 4'b0010)?8'b10010010:(bcd == 4'b0001)?8'b11001111:
            (bcd == 4'b0000)?8'b10000001:8'b01111111;


endmodule



module lab09_1(clk, rst, count, ssd, dp);


  input clk, rst;
        wire clk_out;
        output [3:0] count;
        output [6:0] ssd;
        output dp;
        reg [3:0] count;
        bcdtossd inst1(dp, ssd, count);
        clock_div cd1(clk, rst, clk_out);
        always@(posedge clk_out)
        begin
        if(rst == 0)
           count = 0;
        else
           count = count + 1'b1;
        end
```

endmodule

UCF File:

```
NET "count[0]" LOC = P15 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
NET "count[1]" LOC = P16 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
NET "count[2]" LOC = N15 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
NET "count[3]" LOC = N16 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
NET "clk" LOC = V10 | IOSTANDARD = LVCMOS33 | PERIOD = 100MHZ;
NET "rst" LOC = M18 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST | PULLUP;
NET "ssd[6]" LOC = A3 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
NET "ssd[5]" LOC = B4 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
NET "ssd[4]" LOC = A4 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
NET "ssd[3]" LOC = C4 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
NET "ssd[2]" LOC = C5 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
NET "ssd[1]" LOC = D6 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
NET "ssd[0]" LOC = C6 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
NET "dp" LOC = A5 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
NET "dp" LOC = B3 | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = FAST;
```