**Assignment – 5**

Implementation of Up/Down Counter for 4bits

//Verilog Code

module up\_down\_counter (

input wire clk,

input wire reset,

input wire up\_down,

output reg [3:0] count);

always @(posedge clk) begin

if (reset)

count <= 4'b0000;

else if (up\_down)

count <= count + 1;

else

count <= count - 1;

end

endmodule

//TestBench

timescale 1ns / 1ps

module tb\_up\_down\_counter;

reg clk;

reg reset;

reg up\_down;

wire [3:0] count;

up\_down\_counter uut (

.clk(clk),

.reset(reset),

.up\_down(up\_down),

.count(count)

);

always #5 clk = ~clk;

initial begin

$dumpfile("up\_down\_counter.vcd");

$dumpvars(0, tb\_up\_down\_counter);

clk = 0;

reset = 1;

up\_down = 1;

#10 reset = 0;

#50 up\_down = 0;

#40 up\_down = 1;

#30 reset = 1;

#10 reset = 0;

#20 $finish;

end

endmodule