**Experiment no: 12 Date:**

**BINARY COUNTER**

**AIM:**

To write a verilog HDL program for binary counter and verify its output.

**ALGORITHM:**

Step1: Define the specifications and initialize the design.   
Step2: Write the source code in VERILOG.  
Step3: Check the syntax and perform synthesis.  
Step4: Write different combinations of input using the test bench.   
Step5: Verify the output by simulating the source code.

**VERILOG SOURCE CODE:**

module counter ( input clk, input rstn, output reg[3:0] out);

always @ (posedge clk) begin

if (rstn <= 0)

out <= 0;

else

out <= out + 1;

end

endmodule

**TESTBENCH:**

module tb\_counter;

reg clk;

reg rstn;

wire [3:0] out;

counter uut(clk,rstn,out);

always #5 clk = ~clk;

// This initial block forms the stimulus of the testbench

initial

begin

clk <= 0;

rstn <= 0;

#20

rstn <= 1;

#80

rstn <= 0;

#50

rstn <= 1;

#200

$finish;

end

initial begin

$dumpvars(0,uut);

$dumpfile("dump.vcd");

end

endmodule

**SIMULATION OUTPUT:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**RESULT:**

Thus a verilog HDL program was written for binary counter and its output was verified.