**Experiment 1**

**Aim :** Realization of logic gates using Verilog HDL code in all three modelling style .

**Gate Level Modelling**

module LogicGates(a,b,y);

input a,b;

output [0:6]y;

not(y[0],a);

and(y[1], a,b);

or(y[2],a,b);

nand(y[3],a,b);

nor(y[4],a,b);

xor(y[5],a,b);

xnor(y[6],a,b);

endmodule

**Behavioral Modelling**

module gates (a, b, y);

input a, b;

output reg [0:6]y;

always @ (a or b)

begin

y[0]=~a;

y[1]=a&b;

y[2]=a|b;

y[3]=~(a &b);

y[4]=~(a|b);

y[5]=a^b;

y[6]=~(a^b);

end

endmodule

**Data flow modelling**

module gates (a, b, y);

input a, b;

output [0:6]y;

assign y[0]=~a;

assign y[1]=a&b;

assign y[2]=a|b;

assign y[3]=~(a &b);

assign y[4]=~(a|b);

assign y[5]=a^b;

assign y[6]=~(a^b);

end

endmodule

**Test bench**

module TestModule;

// Inputs

reg a;

reg b;

// Outputs

wire [0:6]y;

// Instantiate the Unit Under Test (UUT)

gates uut (a,b,y);

initial begin

// Initialize Inputs

a = 0; b = 0;

// Wait 100 ns for global reset to finish

#1

a = 0;b = 1;

#1

a = 1;

b = 0;

#1

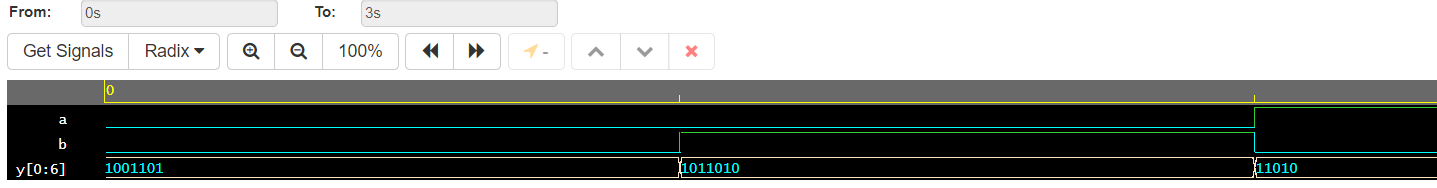
a = 1;

b = 1;

end

endmodule

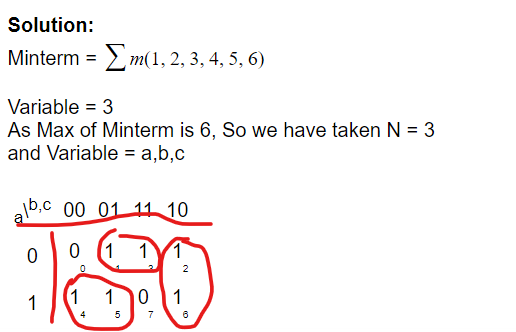
**Waveform**

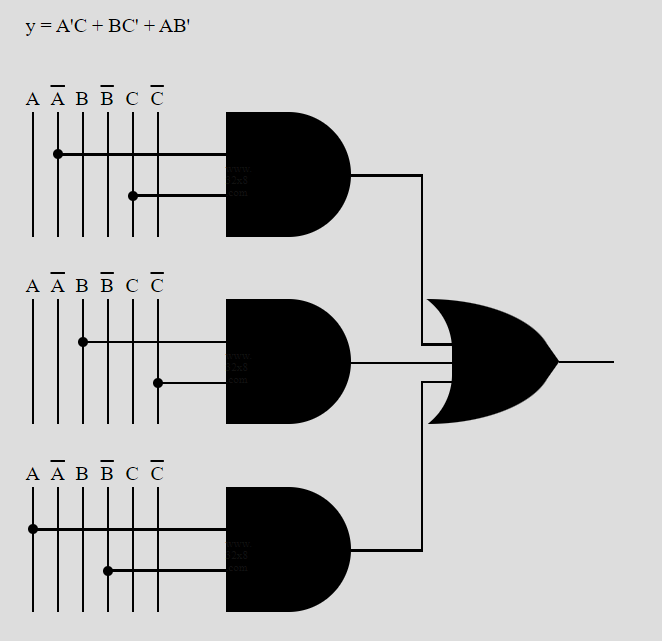
****

**1 b**

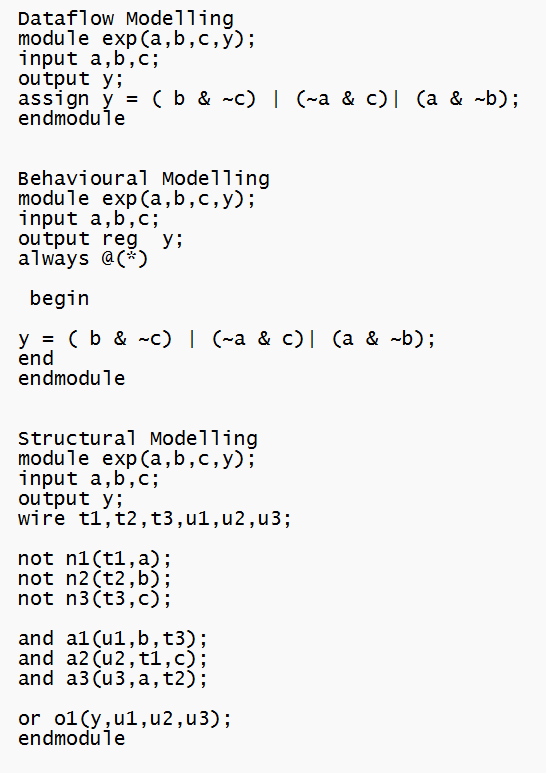
**To design and develop Verilog HDL code to realize the given three variables and four variable functions.**

**(a,b,c) = ∑m(1,2,3,4,5,6)**

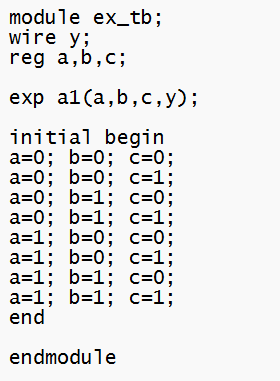
****

****

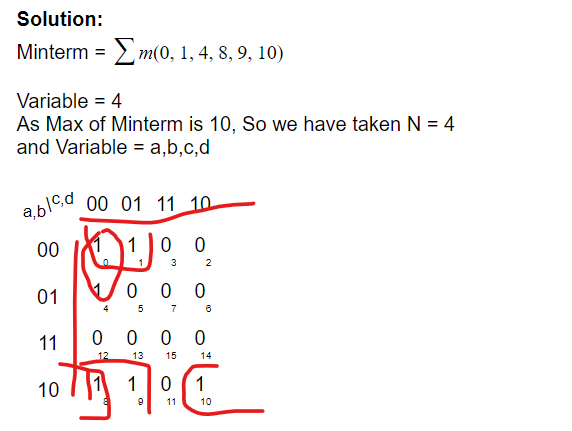
**Code**

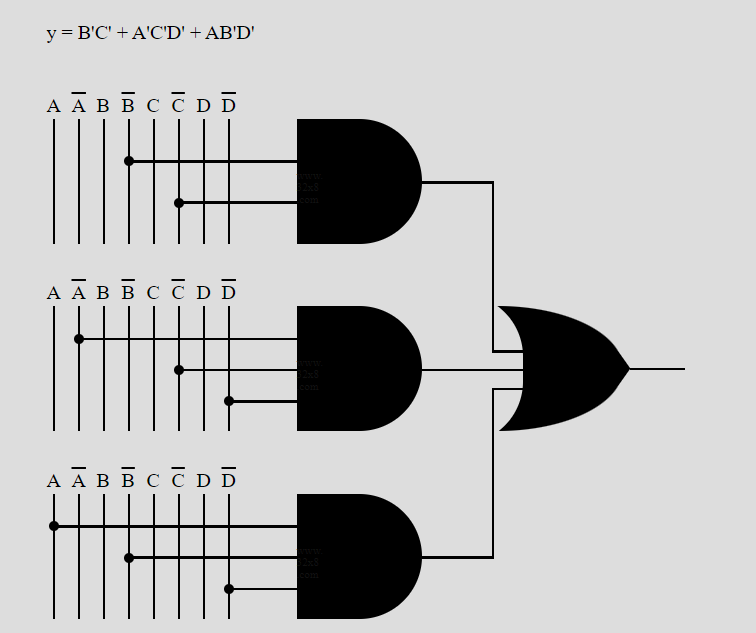
****

**Test bench**

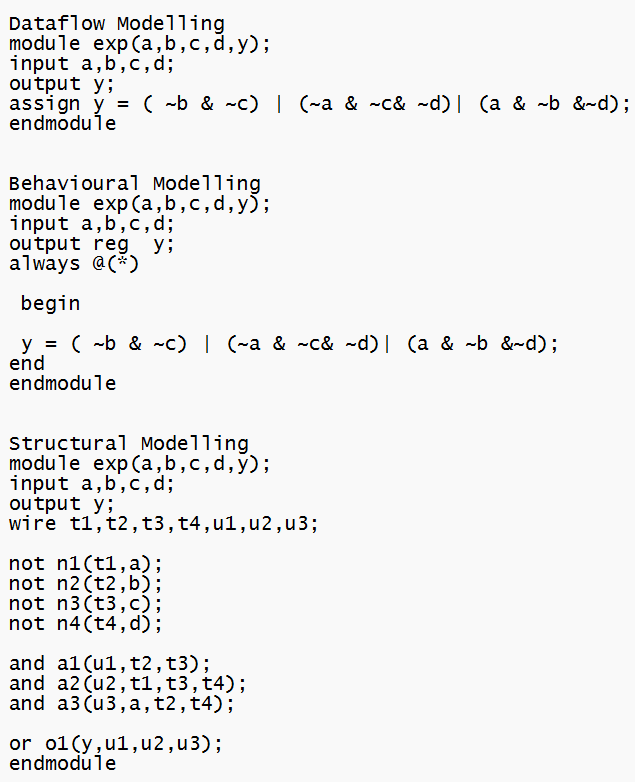
****

**2. f(a,b,c,d) = ∑m(0,1,4,8,9,10)**

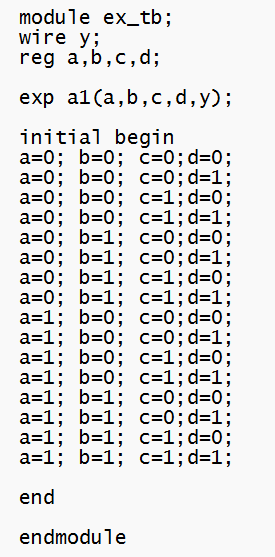
****

****

**Code**

****

**Test Bench**

****