**University of Engineering and Technology,Taxila**

**Department of Computer Engineering**

# Lab Report 04

**For the Course of DSD lab**

**Submitted By**:Muhammad Ibrahim (21-CP-26)

**Section:** Omega

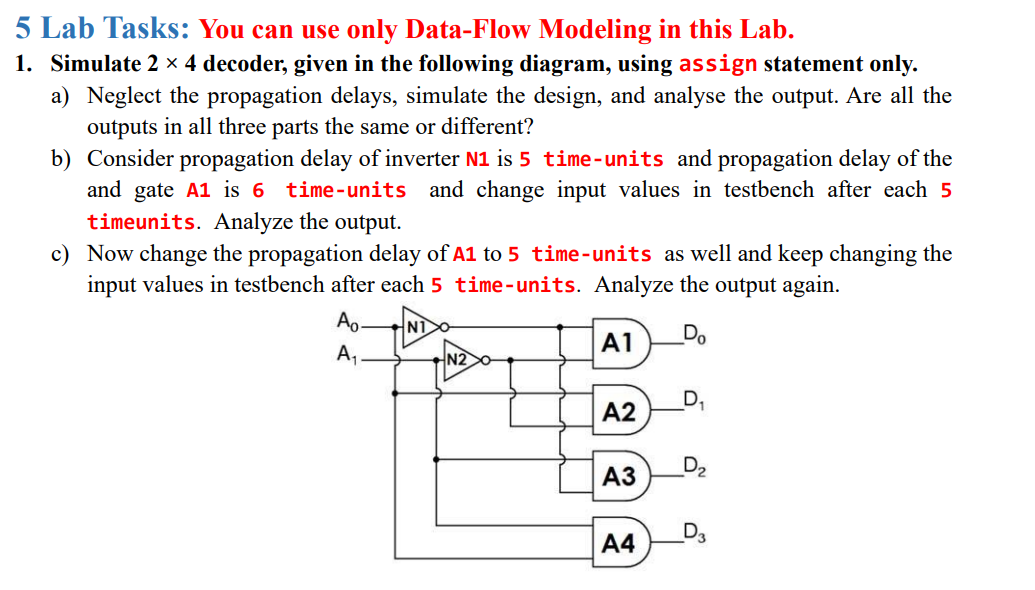
## **Lab Instructor:**Sir Shahid Ali

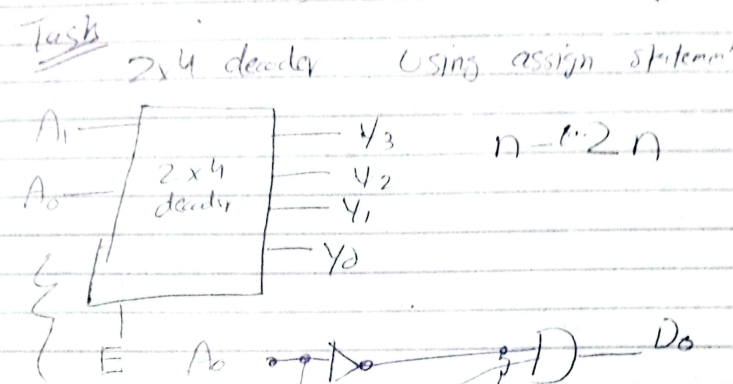
**Course Instructor:**Dr.Abdul rehman aslam

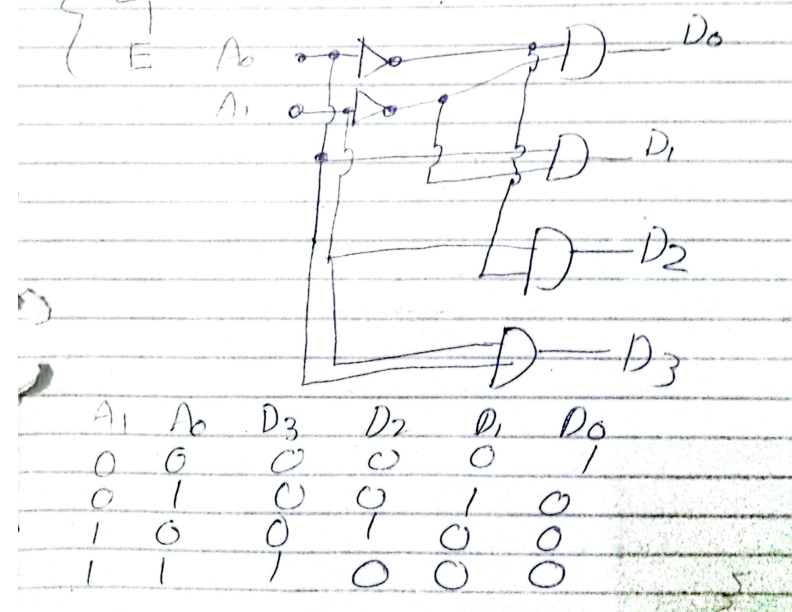
**Date**: 11-02-24.

## **Course Title**: DSD Lab

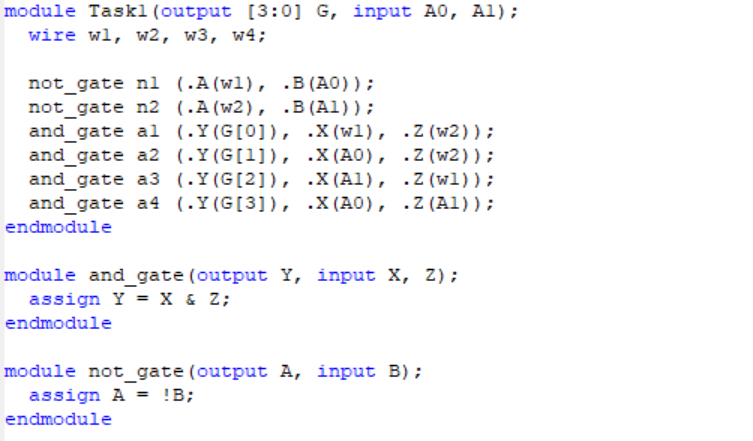
**Question 1:**







**Code:**



module Task1(output [3:0] G, input A0, A1);

wire w1, w2, w3, w4;

not\_gate n1 (.A(w1), .B(A0));

not\_gate n2 (.A(w2), .B(A1));

and\_gate a1 (.Y(G[0]), .X(w1), .Z(w2));

and\_gate a2 (.Y(G[1]), .X(A0), .Z(w2));

and\_gate a3 (.Y(G[2]), .X(A1), .Z(w1));

and\_gate a4 (.Y(G[3]), .X(A0), .Z(A1));

endmodule

module and\_gate(output Y, input X, Z);

assign Y = X & Z;

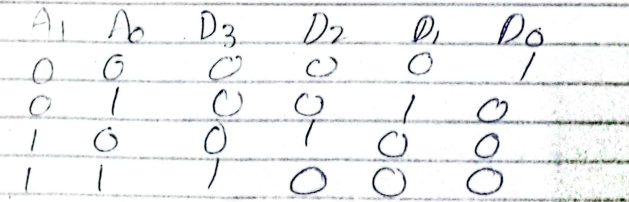
endmodule

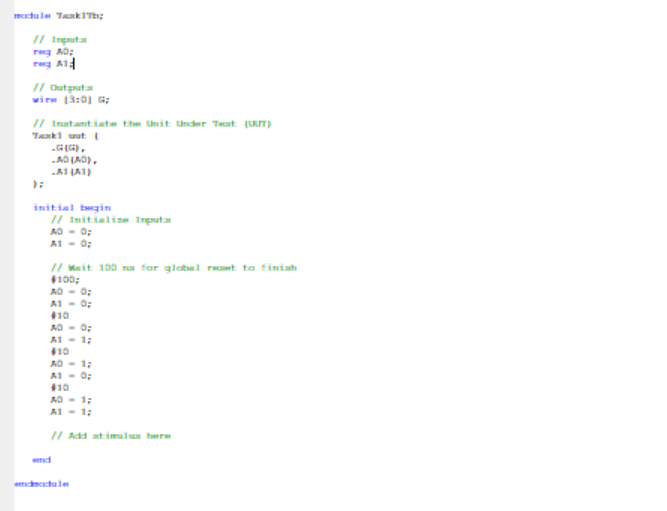
module not\_gate(output A, input B);

assign A = !B;

endmodule

**TestBench:**





module Task1Tb;

// Inputs

reg A0;

reg A1;

// Outputs

wire [3:0] G;

// Instantiate the Unit Under Test (UUT)

Task1 uut (

.G(G),

.A0(A0),

.A1(A1)

);

initial begin

// Initialize Inputs

A0 = 0;

A1 = 0;

// Wait 100 ns for global reset to finish

#100;

A0 = 0;

A1 = 0;

#10

A0 = 0;

A1 = 1;

#10

A0 = 1;

A1 = 0;

#10

A0 = 1;

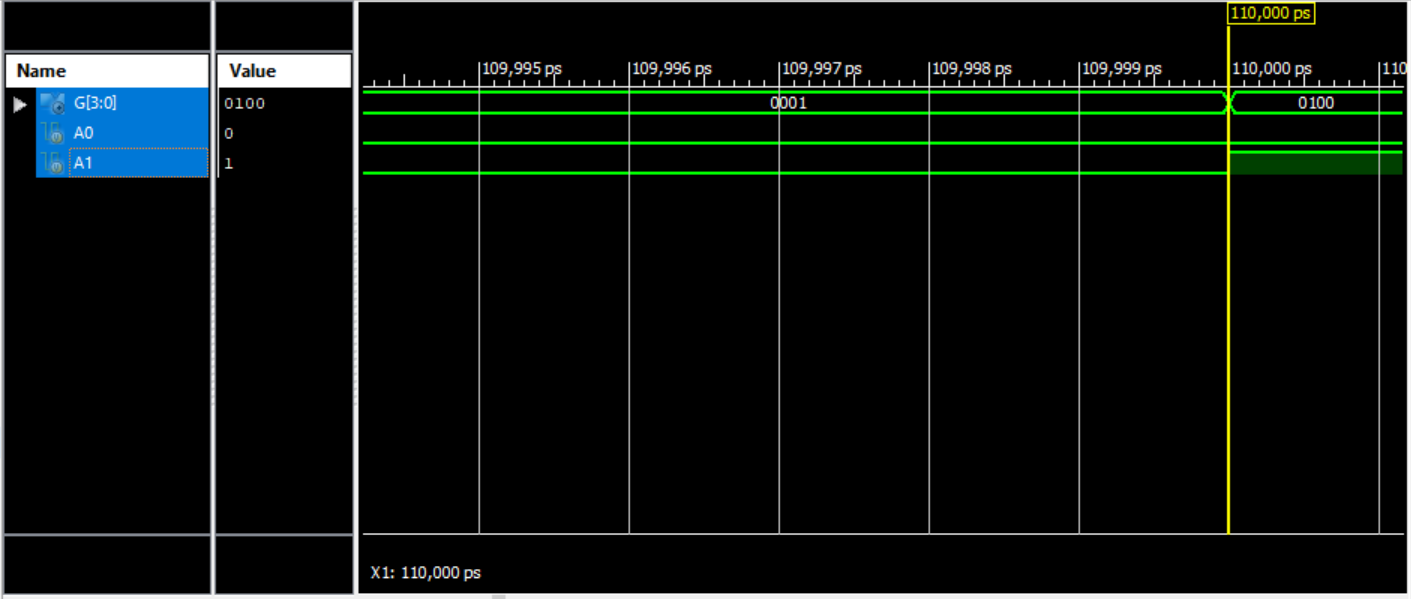
A1 = 1;

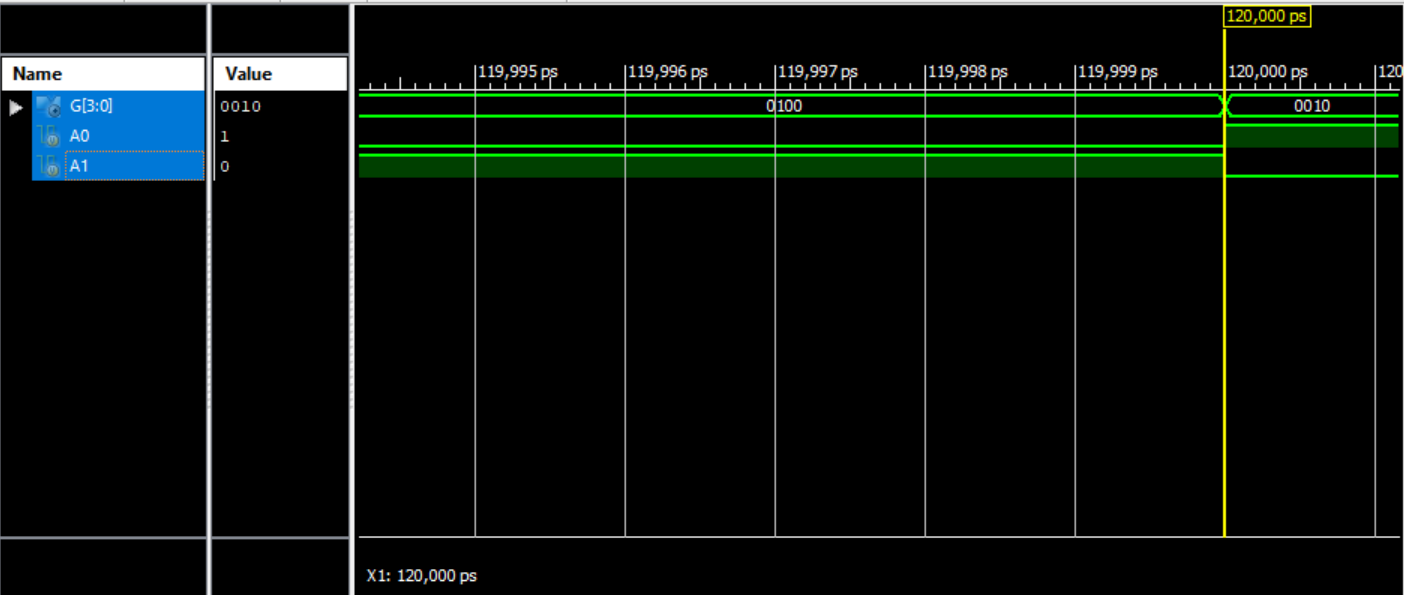
// Add stimulus here

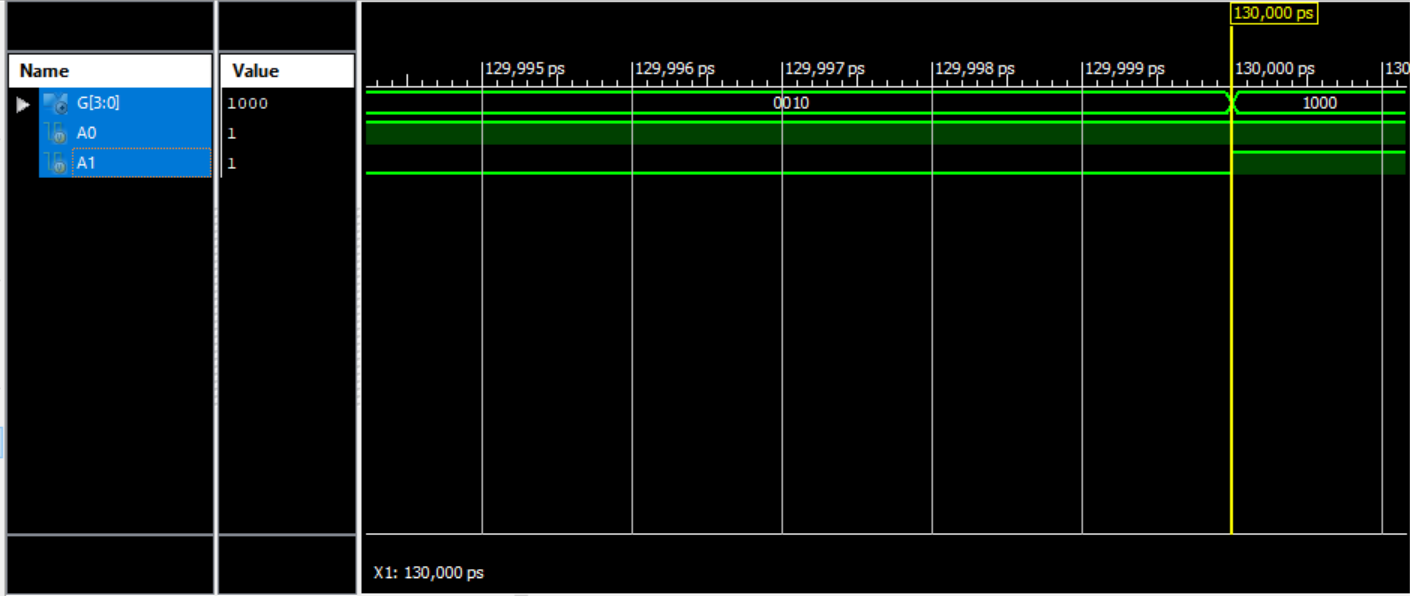
end

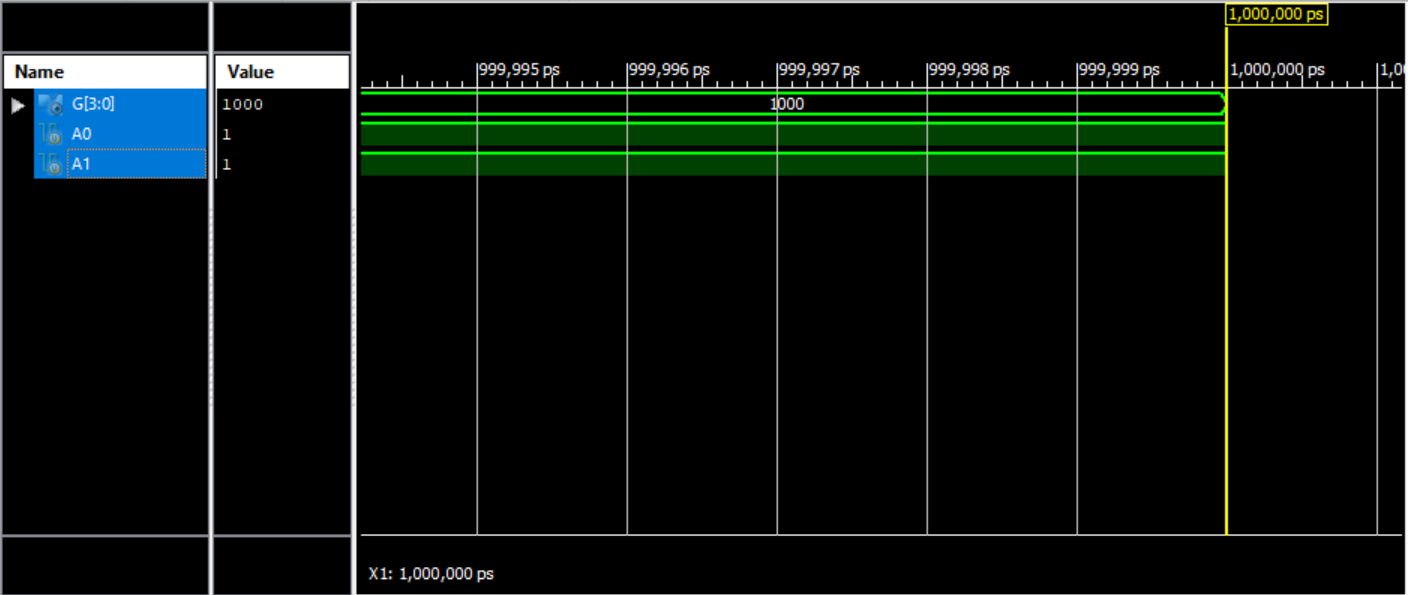
endmodule

**Simulations :**

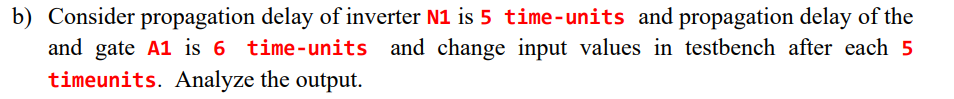




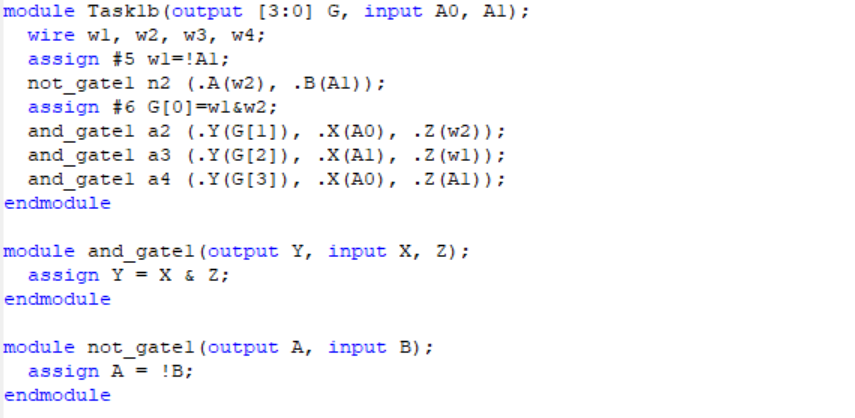




**Part B:**



**Code:**



module Task1b(output [3:0] G, input A0, A1);

wire w1, w2, w3, w4;

assign #5 w1=!A1;

not\_gate1 n2 (.A(w2), .B(A1));

assign #6 G[0]=w1&w2;

and\_gate1 a2 (.Y(G[1]), .X(A0), .Z(w2));

and\_gate1 a3 (.Y(G[2]), .X(A1), .Z(w1));

and\_gate1 a4 (.Y(G[3]), .X(A0), .Z(A1));

endmodule

module and\_gate1(output Y, input X, Z);

assign Y = X & Z;

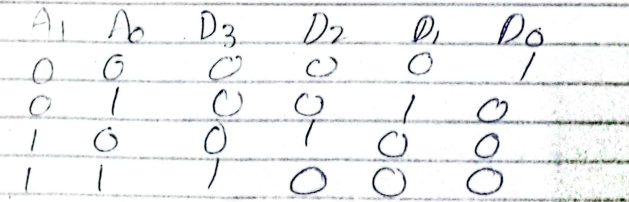
endmodule

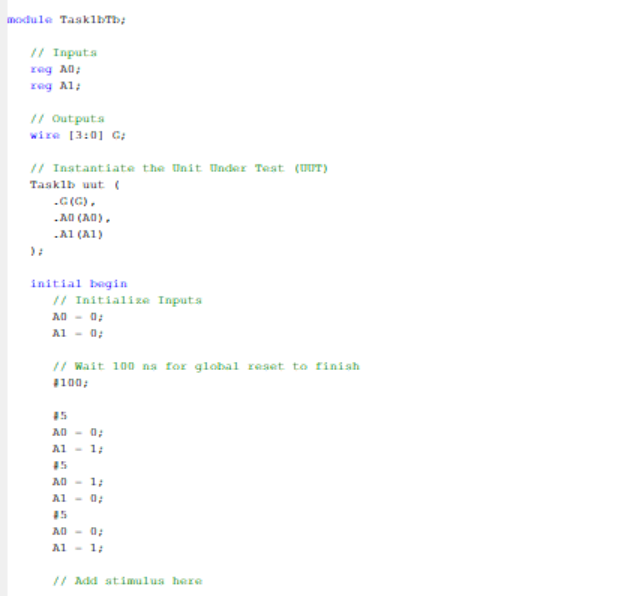
module not\_gate1(output A, input B);

assign A = !B;

endmodule

**TestBench:**





module Task1bTb;

// Inputs

reg A0;

reg A1;

// Outputs

wire [3:0] G;

// Instantiate the Unit Under Test (UUT)

Task1b uut (

.G(G),

.A0(A0),

.A1(A1)

);

initial begin

// Initialize Inputs

A0 = 0;

A1 = 0;

// Wait 100 ns for global reset to finish

#100;

#5

A0 = 0;

A1 = 1;

#5

A0 = 1;

A1 = 0;

#5

A0 = 0;

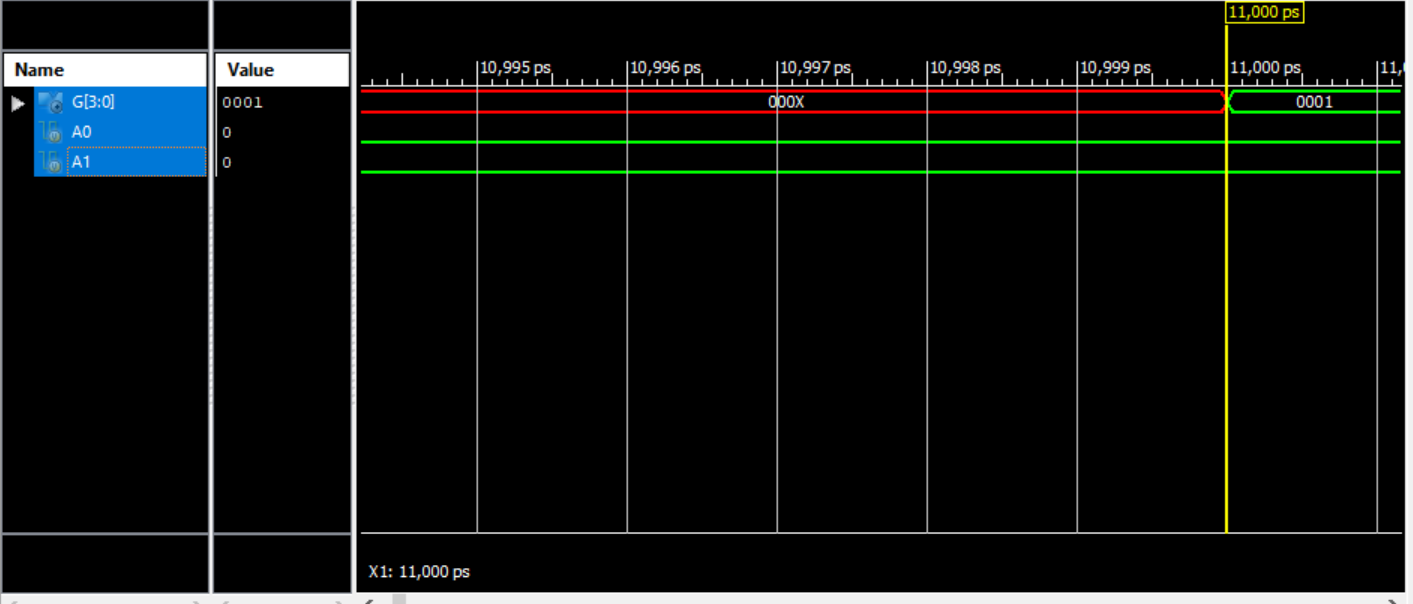
A1 = 1;

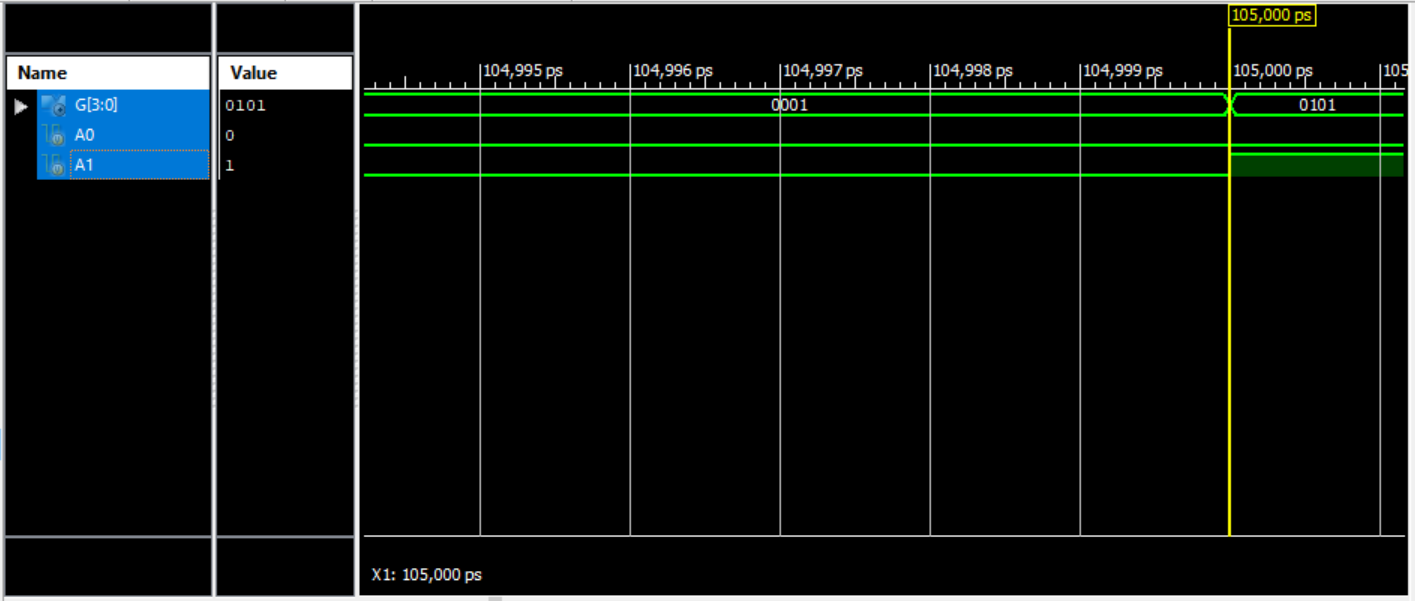
// Add stimulus here

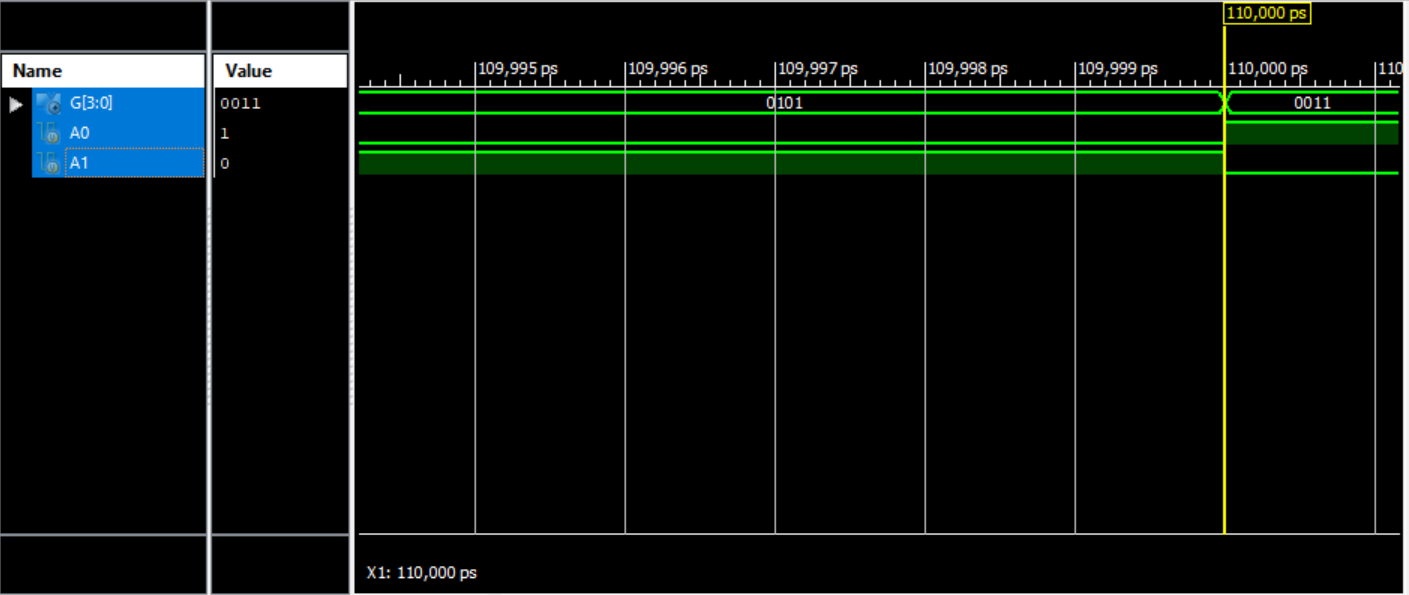
end

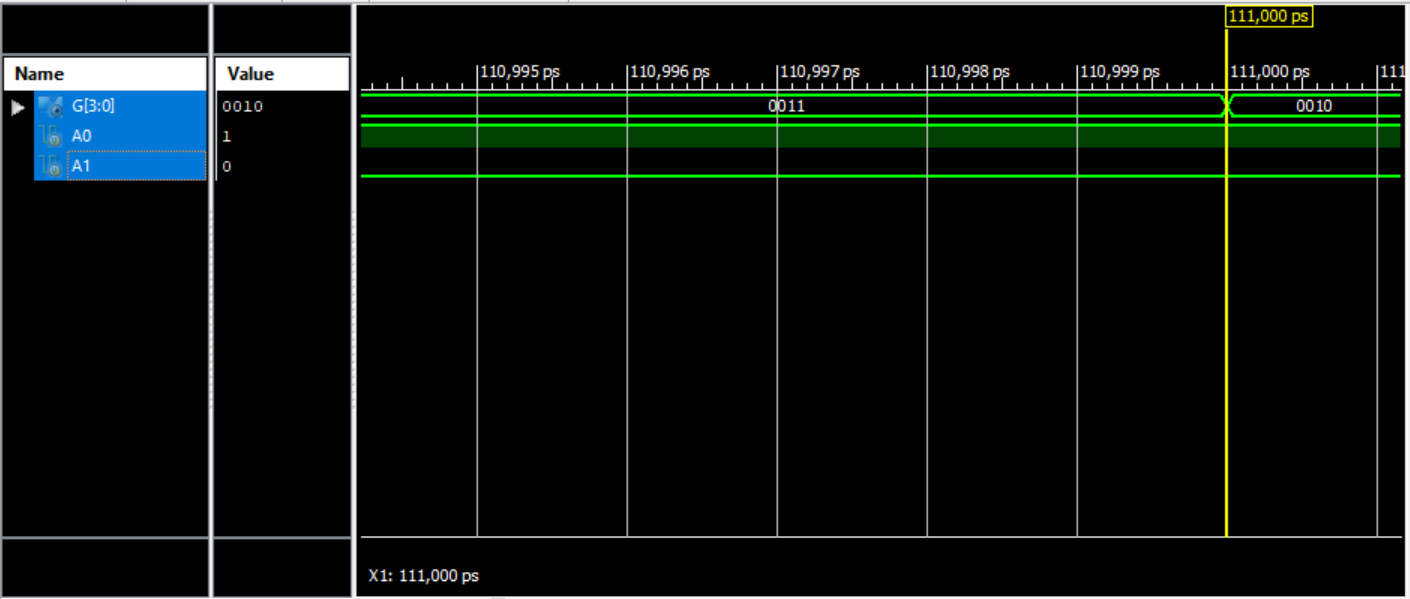
endmodule

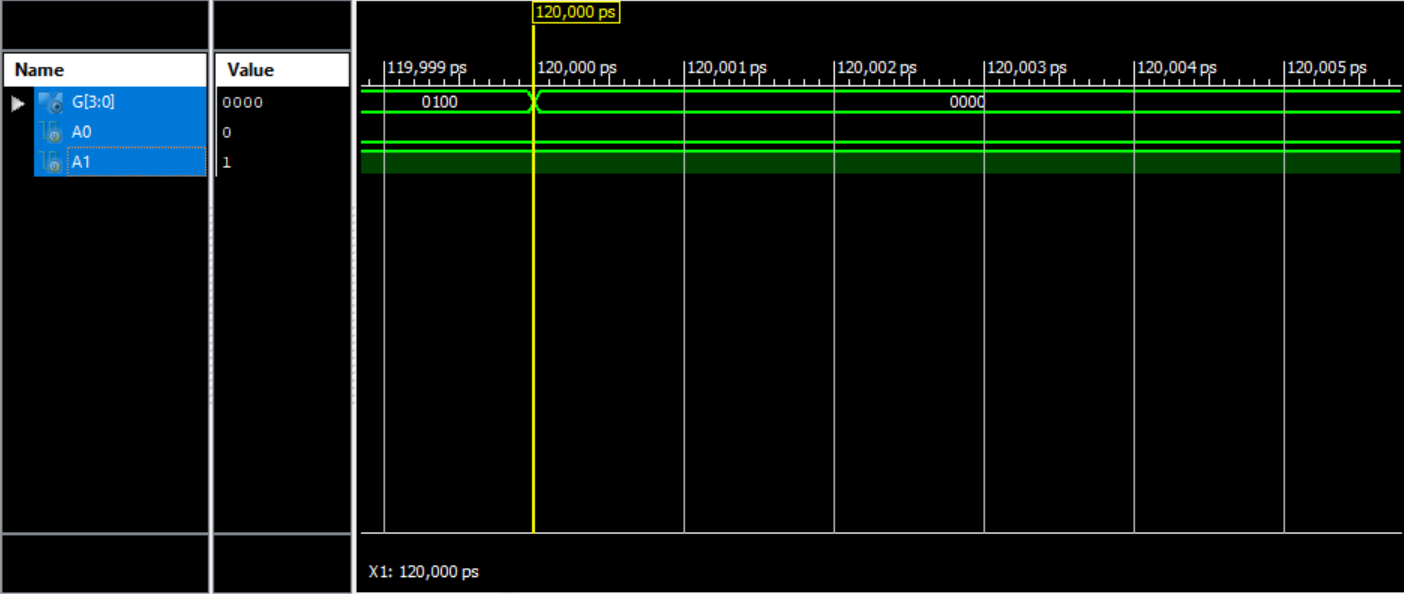
**Simulations :**



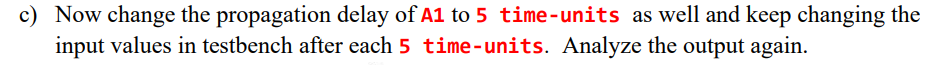




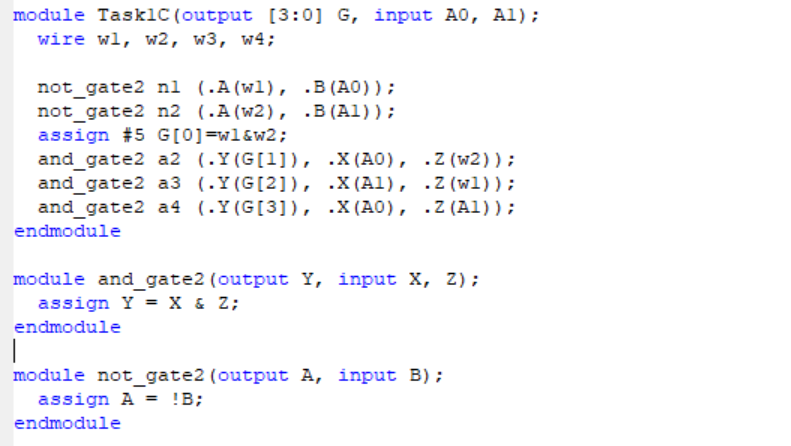




**Part C:**



**Code:**



module Task1C(output [3:0] G, input A0, A1);

wire w1, w2, w3, w4;

not\_gate2 n1 (.A(w1), .B(A0));

not\_gate2 n2 (.A(w2), .B(A1));

assign #5 G[0]=w1&w2;

and\_gate2 a2 (.Y(G[1]), .X(A0), .Z(w2));

and\_gate2 a3 (.Y(G[2]), .X(A1), .Z(w1));

and\_gate2 a4 (.Y(G[3]), .X(A0), .Z(A1));

endmodule

module and\_gate2(output Y, input X, Z);

assign Y = X & Z;

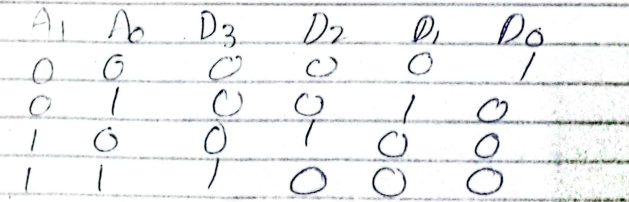
endmodule

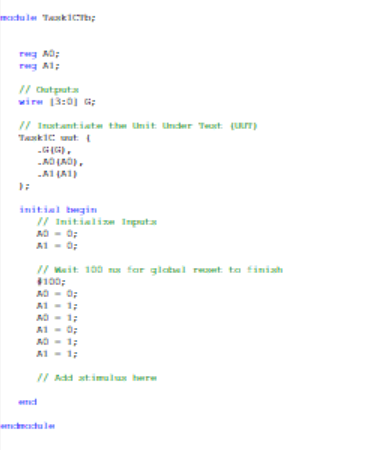
module not\_gate2(output A, input B);

assign A = !B;

endmodule

**TestBench:**





module Task1CTb;

reg A0;

reg A1;

// Outputs

wire [3:0] G;

// Instantiate the Unit Under Test (UUT)

Task1C uut (

.G(G),

.A0(A0),

.A1(A1)

);

initial begin

// Initialize Inputs

A0 = 0;

A1 = 0;

// Wait 100 ns for global reset to finish

#100;

A0 = 0;

A1 = 1;

A0 = 1;

A1 = 0;

A0 = 1;

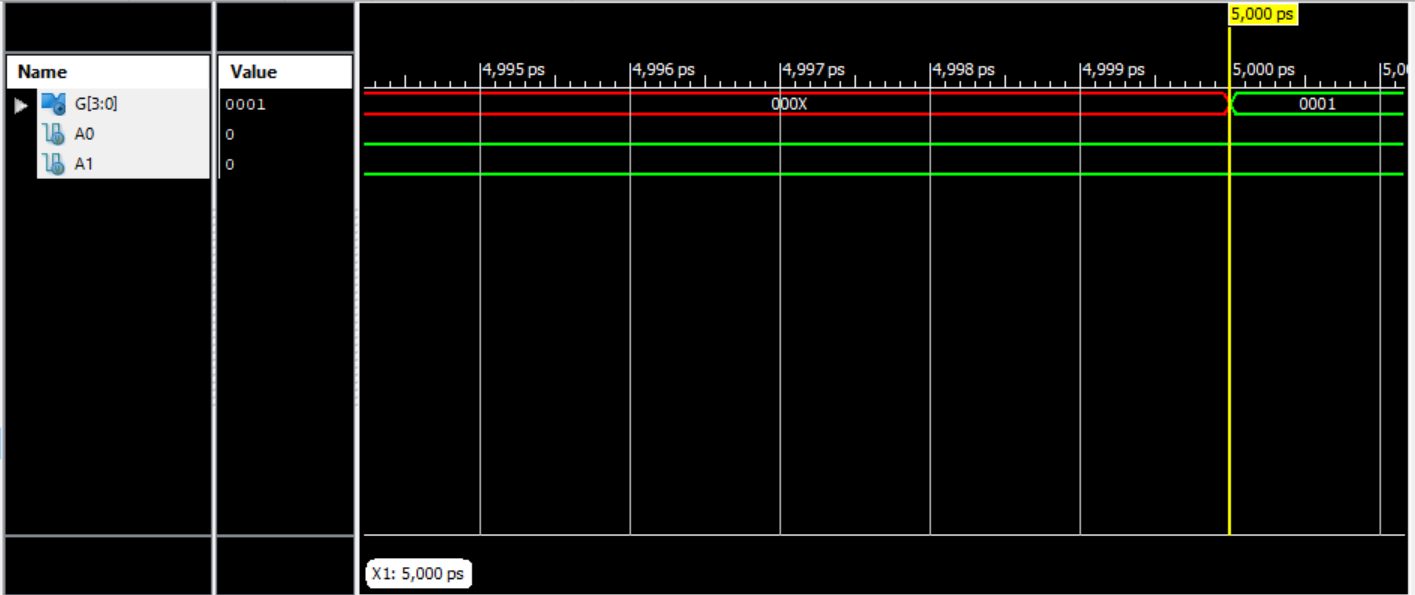
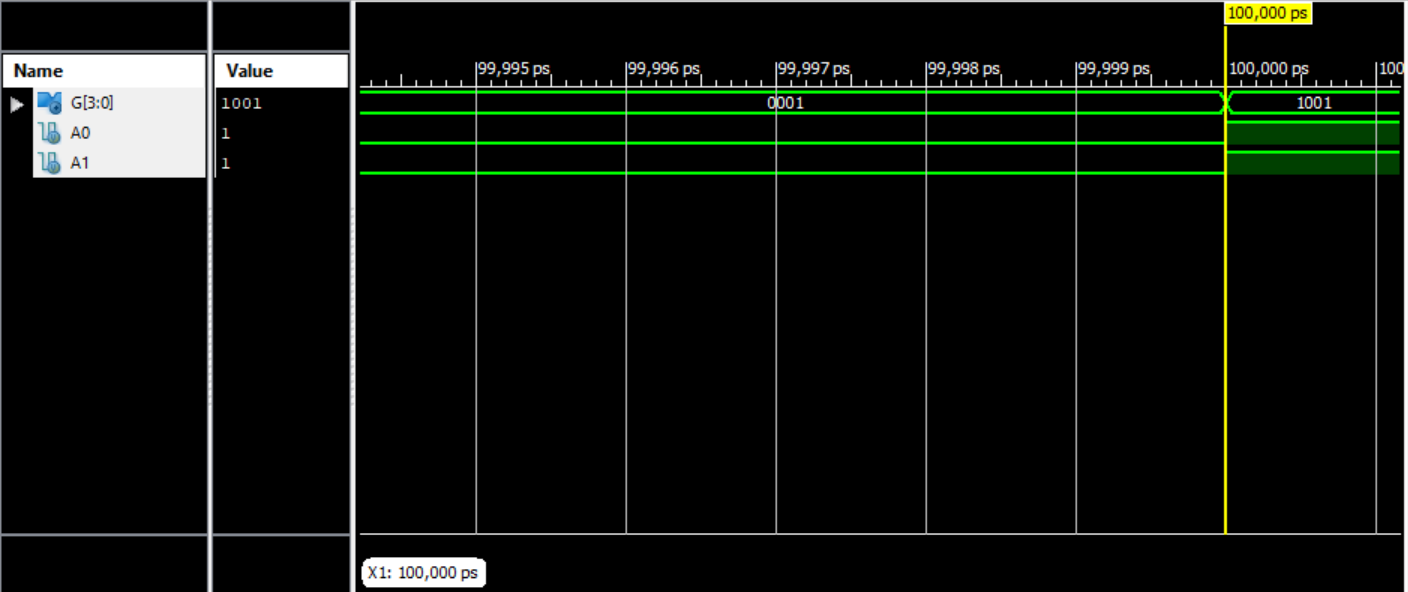
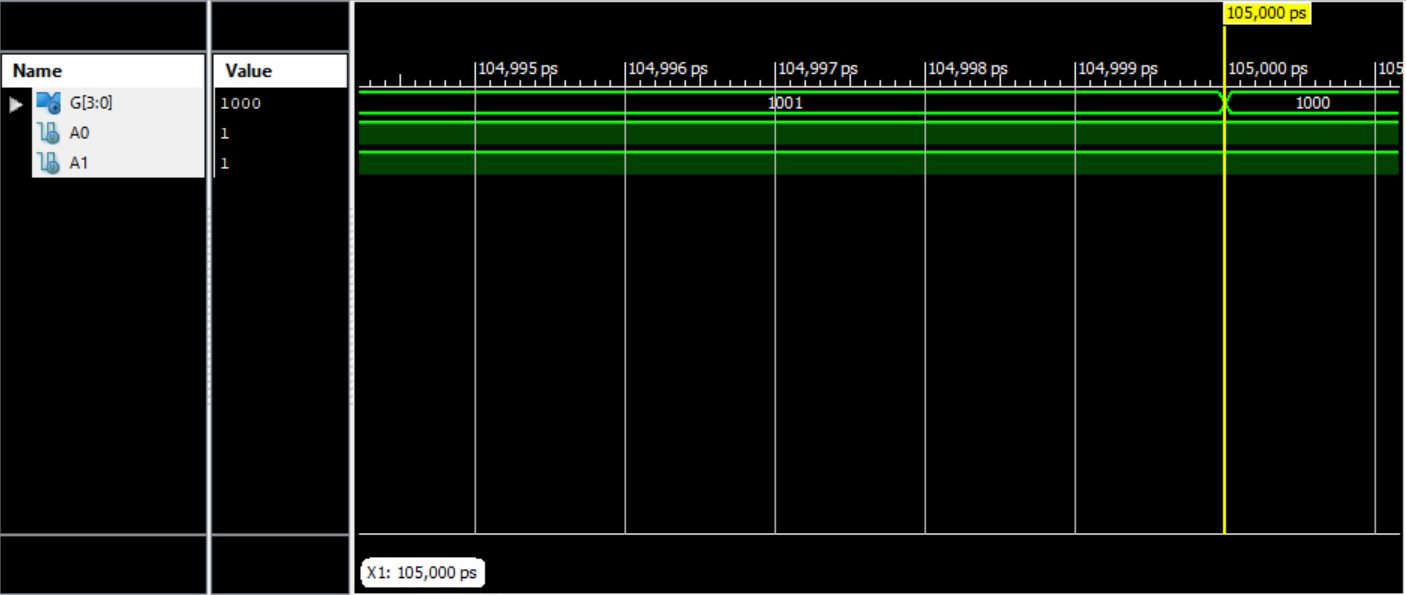
A1 = 1;

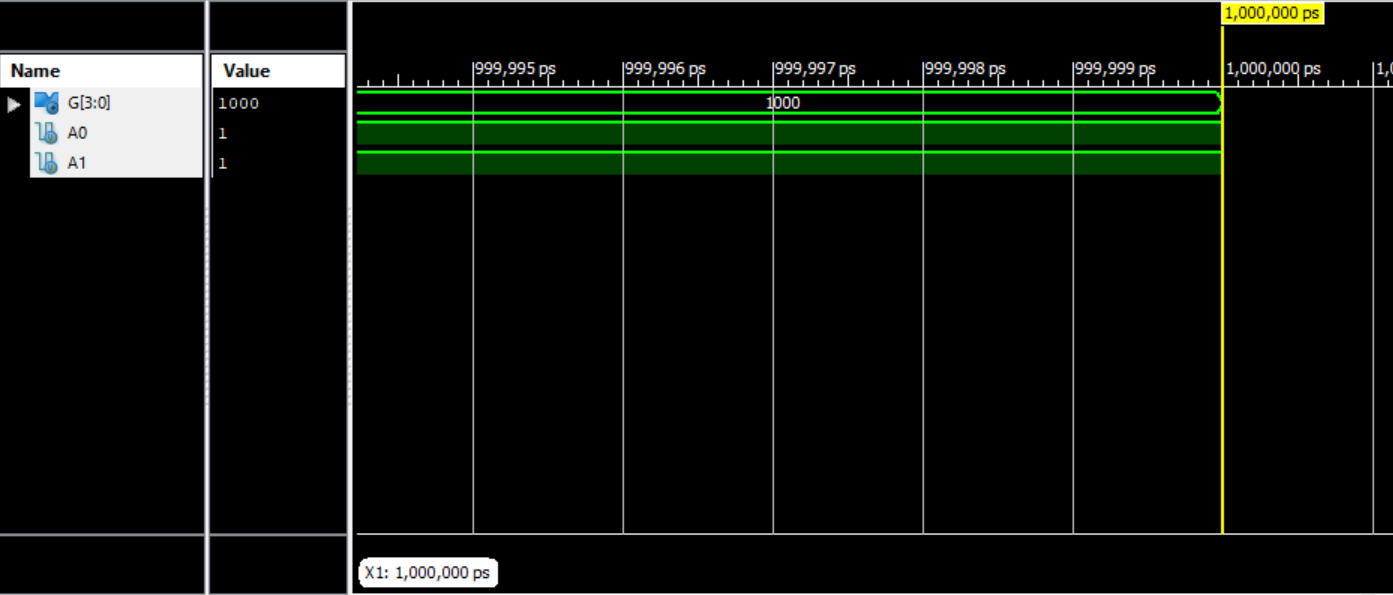
// Add stimulus here

end

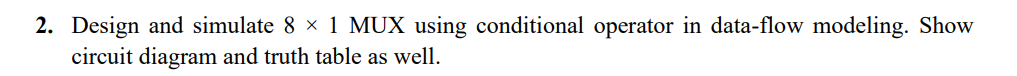
endmodule

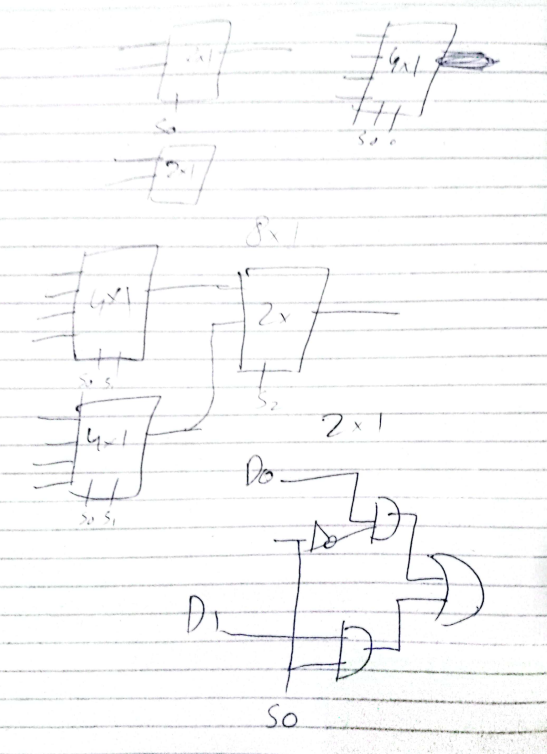
**Simulations :**

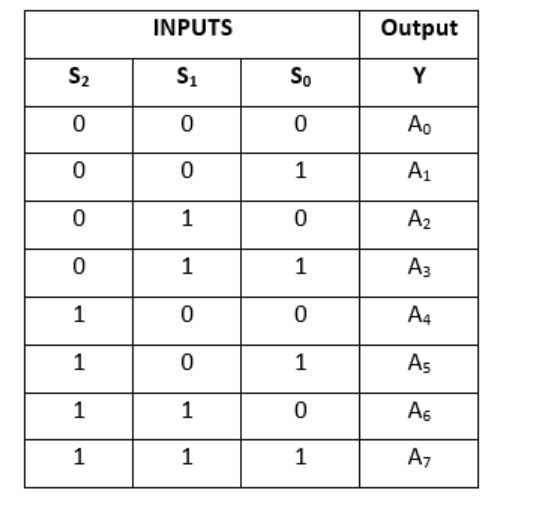
  

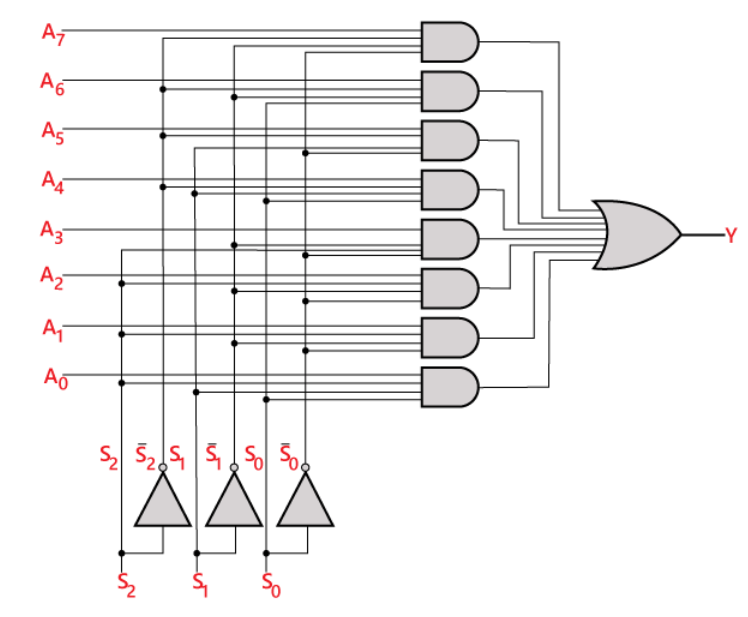


**Question No 2:**

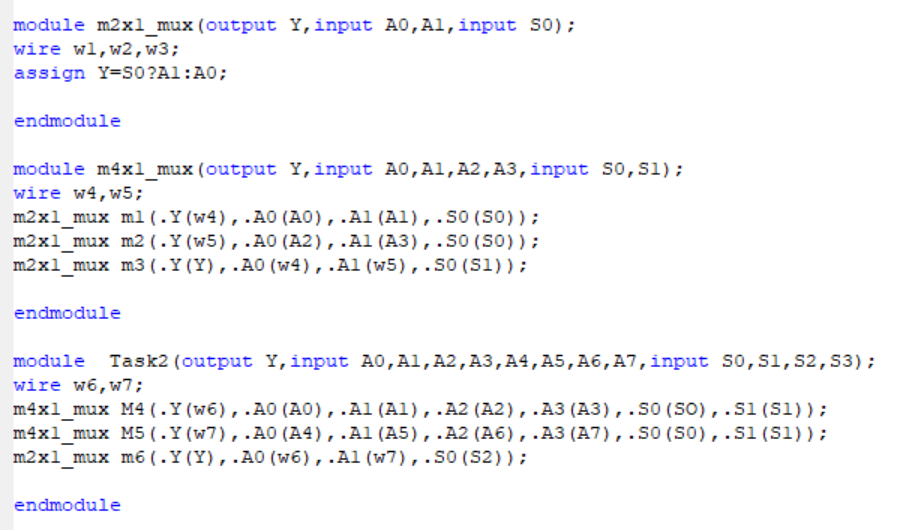








**Code:**



module m2x1\_mux(output Y,input A0,A1,input S0);

wire w1,w2,w3;

assign Y=S0?A1:A0;

endmodule

module m4x1\_mux(output Y,input A0,A1,A2,A3,input S0,S1);

wire w4,w5;

m2x1\_mux m1(.Y(w4),.A0(A0),.A1(A1),.S0(S0));

m2x1\_mux m2(.Y(w5),.A0(A2),.A1(A3),.S0(S0));

m2x1\_mux m3(.Y(Y),.A0(w4),.A1(w5),.S0(S1));

endmodule

module Task2(output Y,input A0,A1,A2,A3,A4,A5,A6,A7,input S0,S1,S2,S3);

wire w6,w7;

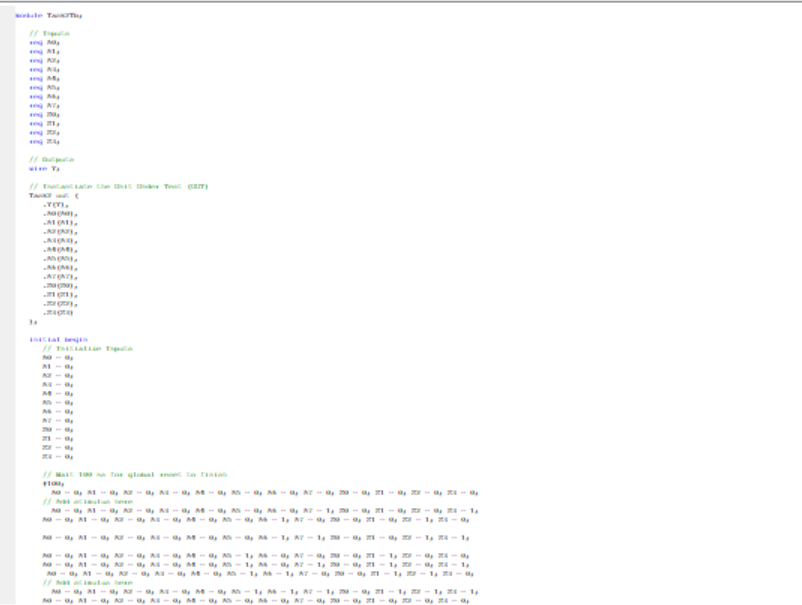
m4x1\_mux M4(.Y(w6),.A0(A0),.A1(A1),.A2(A2),.A3(A3),.S0(SO),.S1(S1));

m4x1\_mux M5(.Y(w7),.A0(A4),.A1(A5),.A2(A6),.A3(A7),.S0(S0),.S1(S1));

m2x1\_mux m6(.Y(Y),.A0(w6),.A1(w7),.S0(S2));

endmodule

**TestBench:**



module Task2Tb;

// Inputs

reg A0;

reg A1;

reg A2;

reg A3;

reg A4;

reg A5;

reg A6;

reg A7;

reg S0;

reg S1;

reg S2;

reg S3;

// Outputs

wire Y;

// Instantiate the Unit Under Test (UUT)

Task2 uut (

.Y(Y),

.A0(A0),

.A1(A1),

.A2(A2),

.A3(A3),

.A4(A4),

.A5(A5),

.A6(A6),

.A7(A7),

.S0(S0),

.S1(S1),

.S2(S2),

.S3(S3)

);

initial begin

// Initialize Inputs

A0 = 0;

A1 = 0;

A2 = 0;

A3 = 0;

A4 = 0;

A5 = 0;

A6 = 0;

A7 = 0;

S0 = 0;

S1 = 0;

S2 = 0;

S3 = 0;

// Wait 100 ns for global reset to finish

#100;

A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 0; A5 = 0; A6 = 0; A7 = 0; S0 = 0; S1 = 0; S2 = 0; S3 = 0;

// Add stimulus here

A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 0; A5 = 0; A6 = 0; A7 = 1; S0 = 0; S1 = 0; S2 = 0; S3 = 1;

A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 0; A5 = 0; A6 = 1; A7 = 0; S0 = 0; S1 = 0; S2 = 1; S3 = 0;

A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 0; A5 = 0; A6 = 1; A7 = 1; S0 = 0; S1 = 0; S2 = 1; S3 = 1;

A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 0; A5 = 1; A6 = 0; A7 = 0; S0 = 0; S1 = 1; S2 = 0; S3 = 0;

A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 0; A5 = 1; A6 = 0; A7 = 1; S0 = 0; S1 = 1; S2 = 0; S3 = 1;

A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 0; A5 = 1; A6 = 1; A7 = 0; S0 = 0; S1 = 1; S2 = 1; S3 = 0;

// Add stimulus here

A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 0; A5 = 1; A6 = 1; A7 = 1; S0 = 0; S1 = 1; S2 = 1; S3 = 1;

A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 0; A5 = 0; A6 = 0; A7 = 0; S0 = 0; S1 = 0; S2 = 0; S3 = 0;

A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 1; A5 = 0; A6 = 0; A7 = 1; S0 = 1; S1 = 0; S2 = 0; S3 = 1;

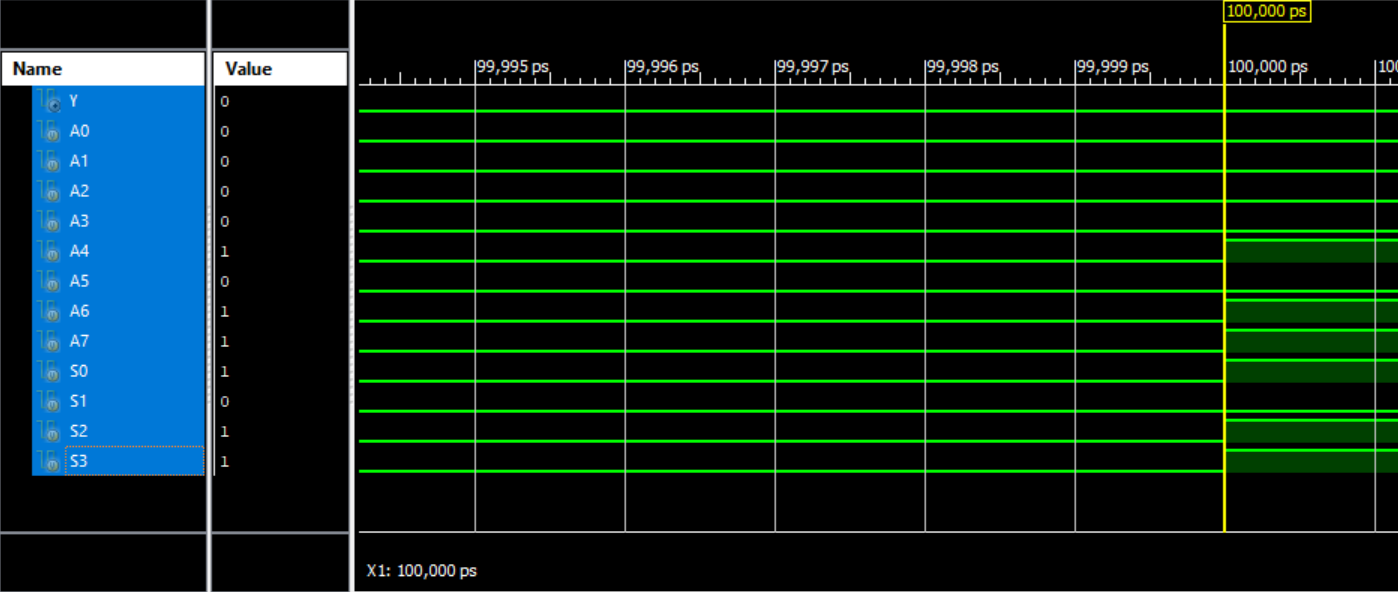
A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 1; A5 = 0; A6 = 1; A7 = 0; S0 = 1; S1 = 0; S2 = 1; S3 = 0;

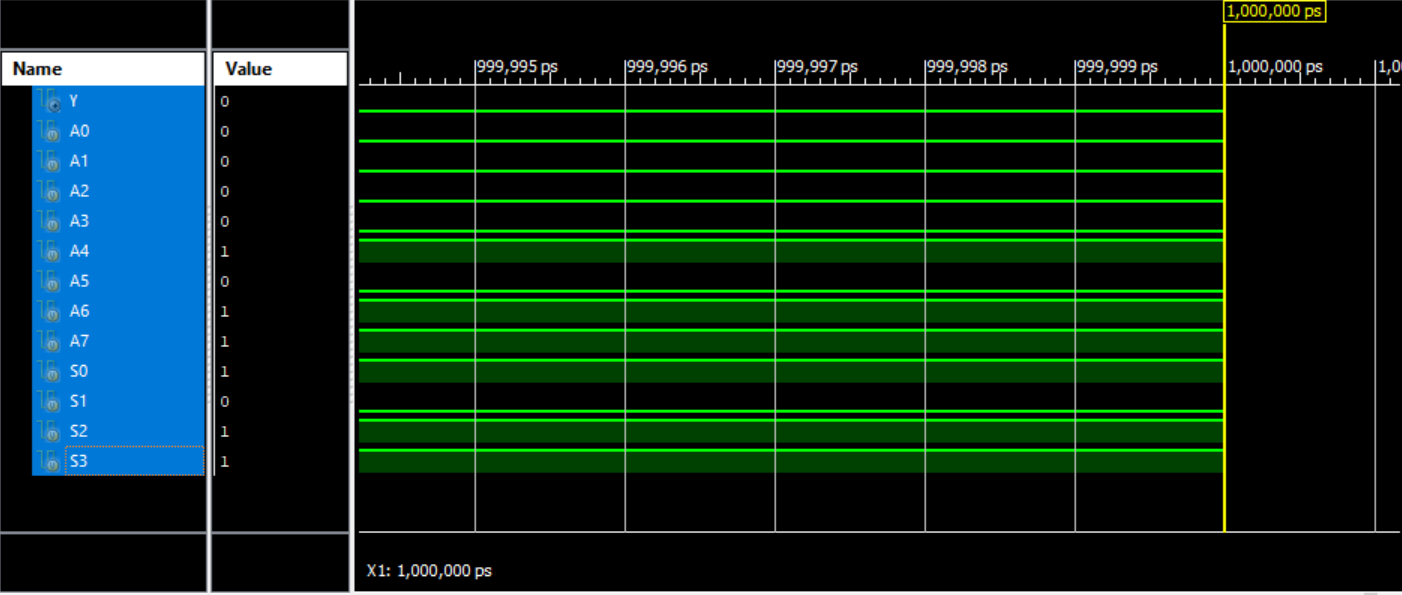
A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 1; A5 = 0; A6 = 1; A7 = 1; S0 = 1; S1 = 0; S2 = 1; S3 = 1;

end

endmodule

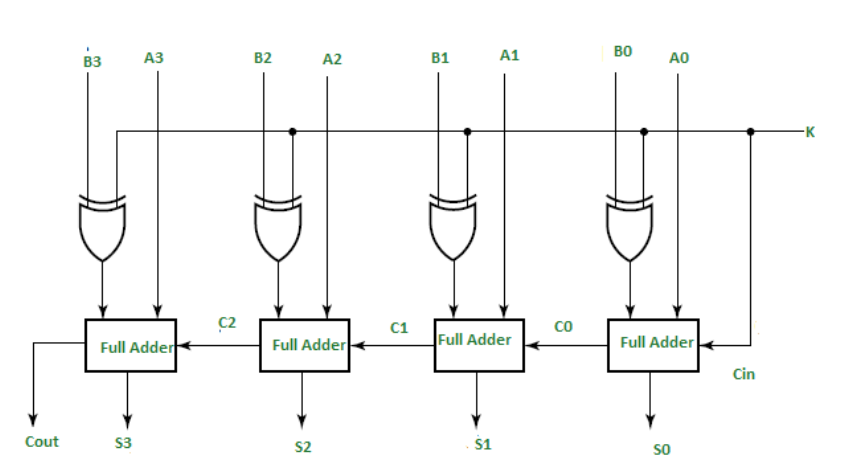
**Simulations :**

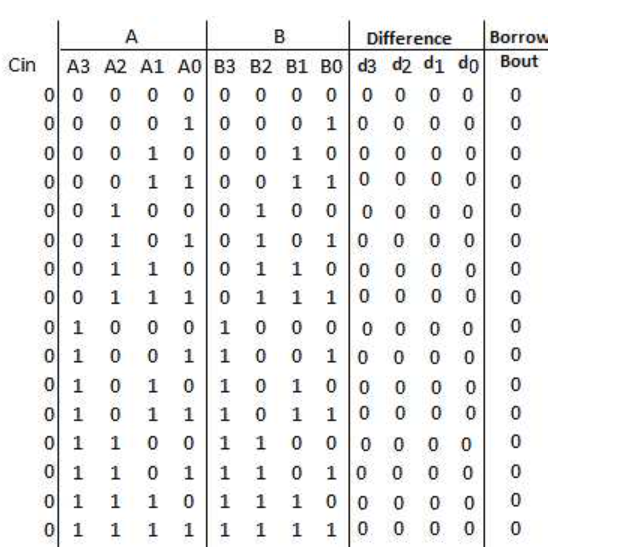




**Question:**







**Code:**

module Task3(

output [3:0] Y

,output Cout

,input[3:0] A

,input [3:0] B,

input Bin );

wire [3:0] W;

wire C0,C1,C2,C3;

wire X0,X1,X2,X3;

//xor

assign X3=B[3]^Bin;

assign X2=B[2]^Bin;

assign X1=B[1]^Bin;

assign X0=B[0]^Bin;

full\_1\_adder fa1(Y[0],C0,A[0],X0,Bin);

full\_1\_adder fa2(Y[1],C1,A[1],X1,C0);

full\_1\_adder fa3(Y[2],C2,A[2],X2,C1);

full\_1\_adder fa4(Y[3],Cout,A[3],X3,C2);

endmodule

module full\_1\_adder(output Y,Cout,input A,B,Cin);

assign w1=A^B;

assign Y=w1^Cin;

assign w2=w1&Cin;

assign w3=A&B;

assign Cout=w2+w3;

endmodule

**TestBench:**

module Task3Tb;

// Inputs

reg [3:0] A;

reg [3:0] B;

reg Bin;

// Outputs

wire [3:0] Y;

wire Cout;

// Instantiate the Unit Under Test (UUT)

Task3 uut (

.Y(Y),

.Cout(Cout),

.A(A),

.B(B),

.Bin(Bin)

);

initial begin

// Initialize In

A = 4'b0101;

B = 4'b0011;

Bin = 0;

#100;

#10 A = 0000;B = 0000;Bin = 1;

#10 A = 00001;B = 0001;Bin = 0;

#10 A = 0010;B = 0010;Bin = 1;

#10 A = 0011;B = 0011;Bin = 0;

#10 A = 0100;B = 0100;Bin = 1;

#10 A = 0101;B = 0101;Bin = 0;

#10 A = 0110;B = 0110;Bin = 1;

#10 A = 0111;B = 0111;Bin = 0;

#10 A = 1000;B = 1000;Bin = 1;

#10 A = 1001;B = 1001;Bin = 0;

#10 A = 1010;B = 1010;Bin = 1;

#10 A = 1011;B = 1011;Bin = 0;

#10 A = 1100;B = 1100;Bin = 1;

#10 A = 1101;B = 1101;Bin = 0;

#10 A = 1110;B = 1110;Bin = 1;

#10 A = 1111;B = 1111;Bin = 0;

// Add stimulus here

end

endmodule

**Simulations :**

