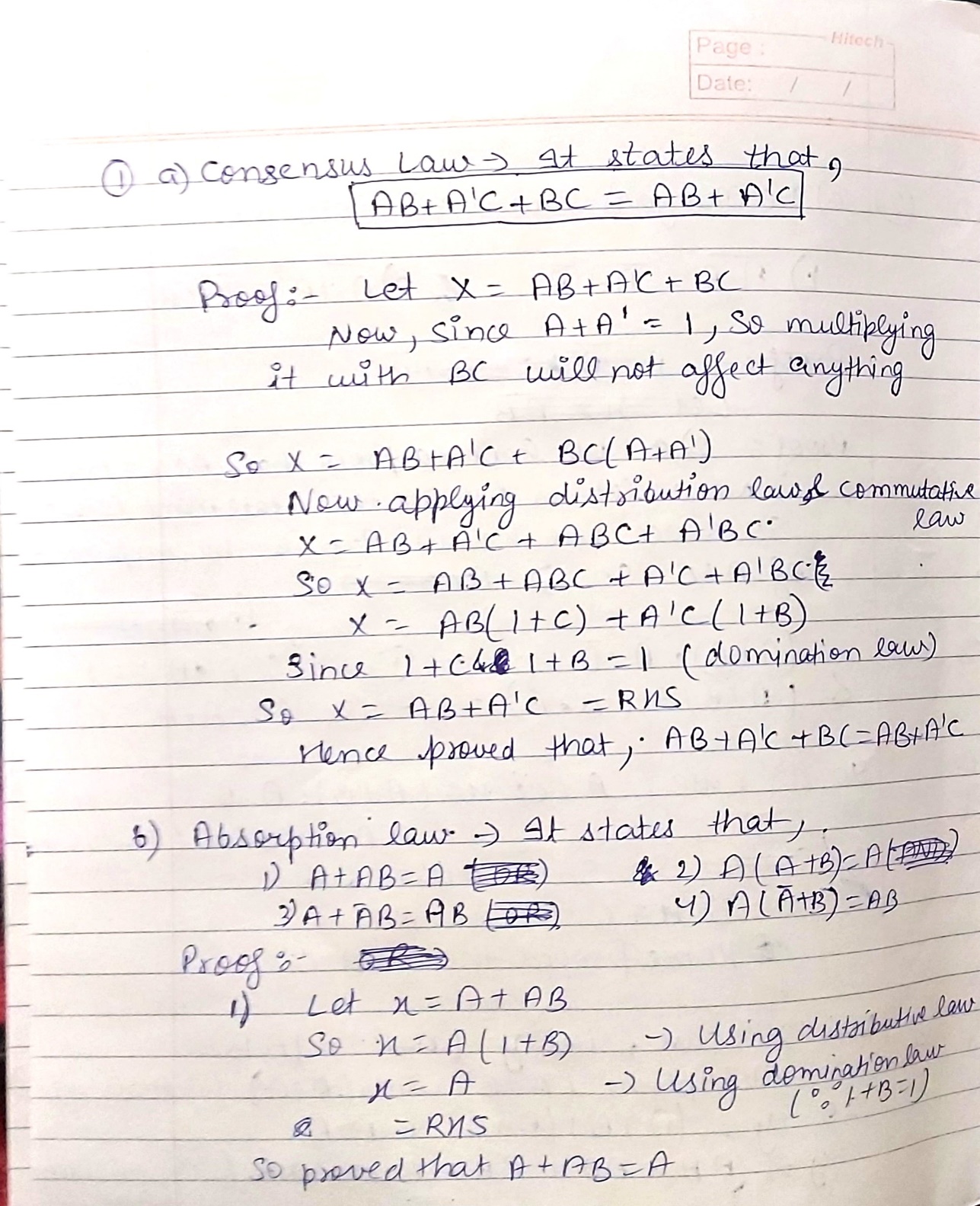
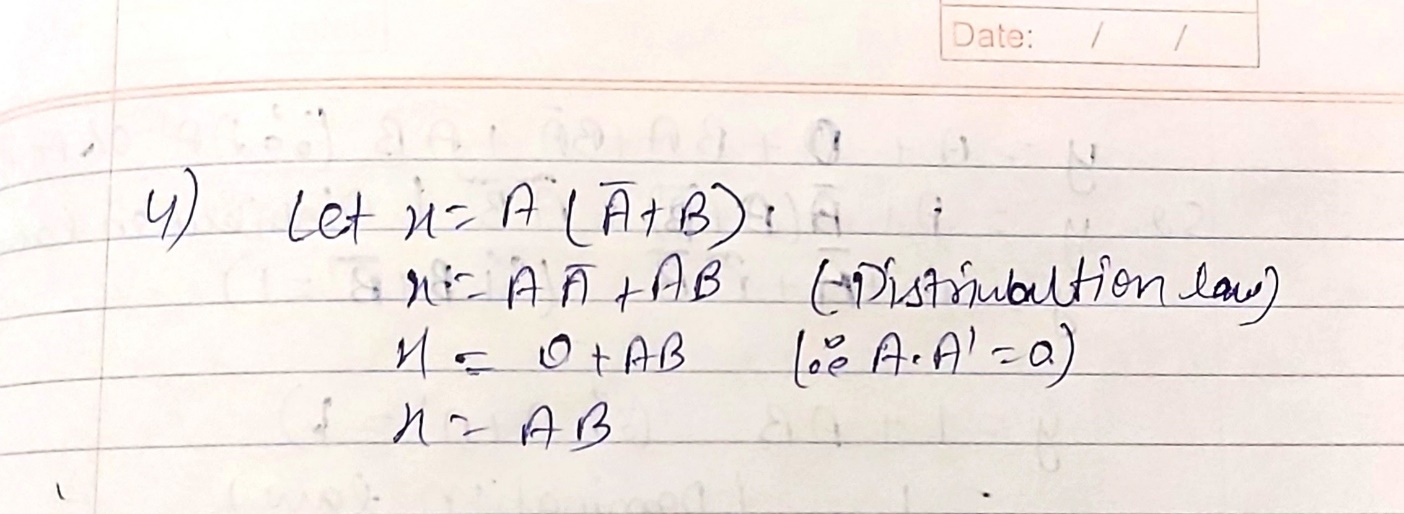
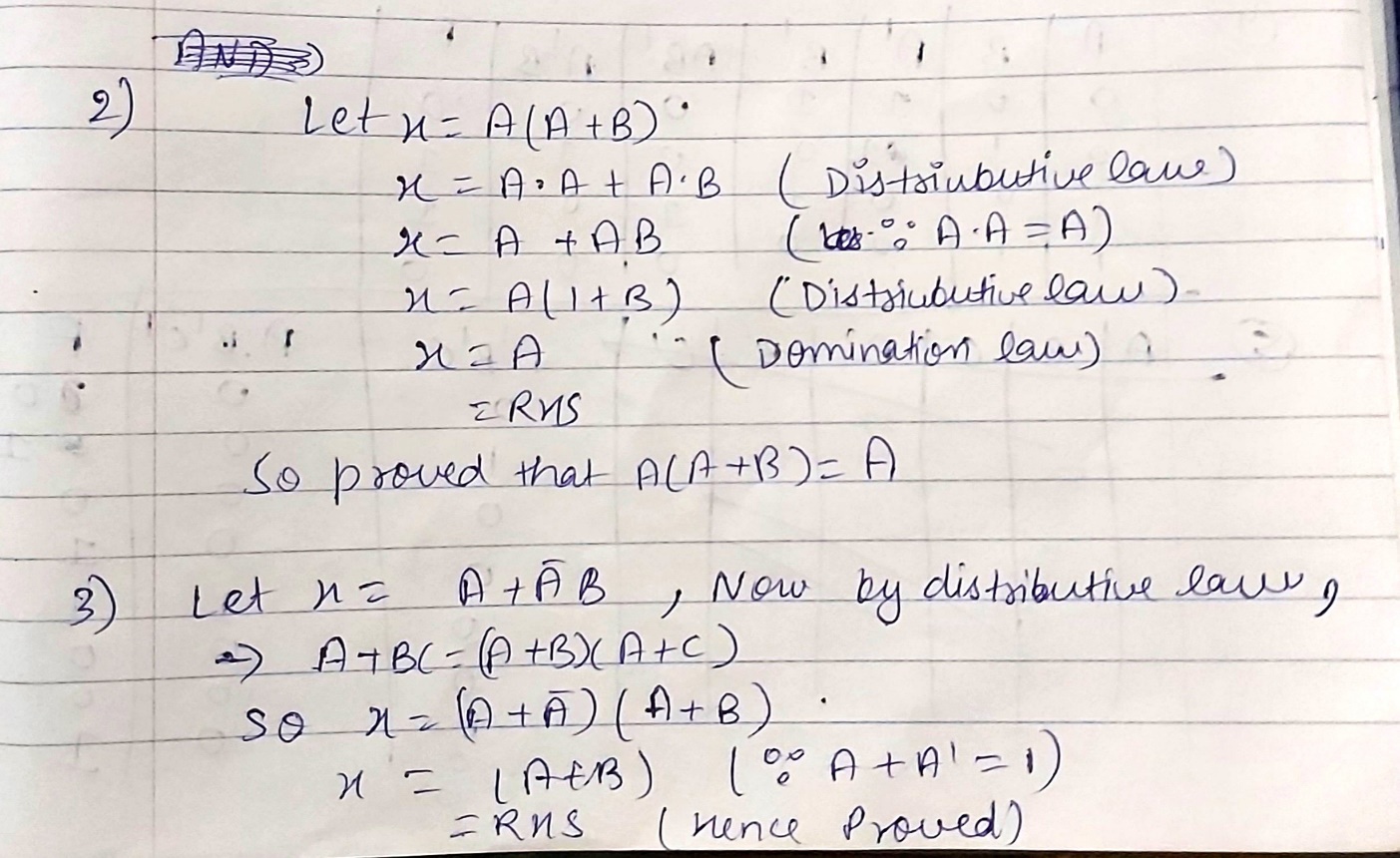
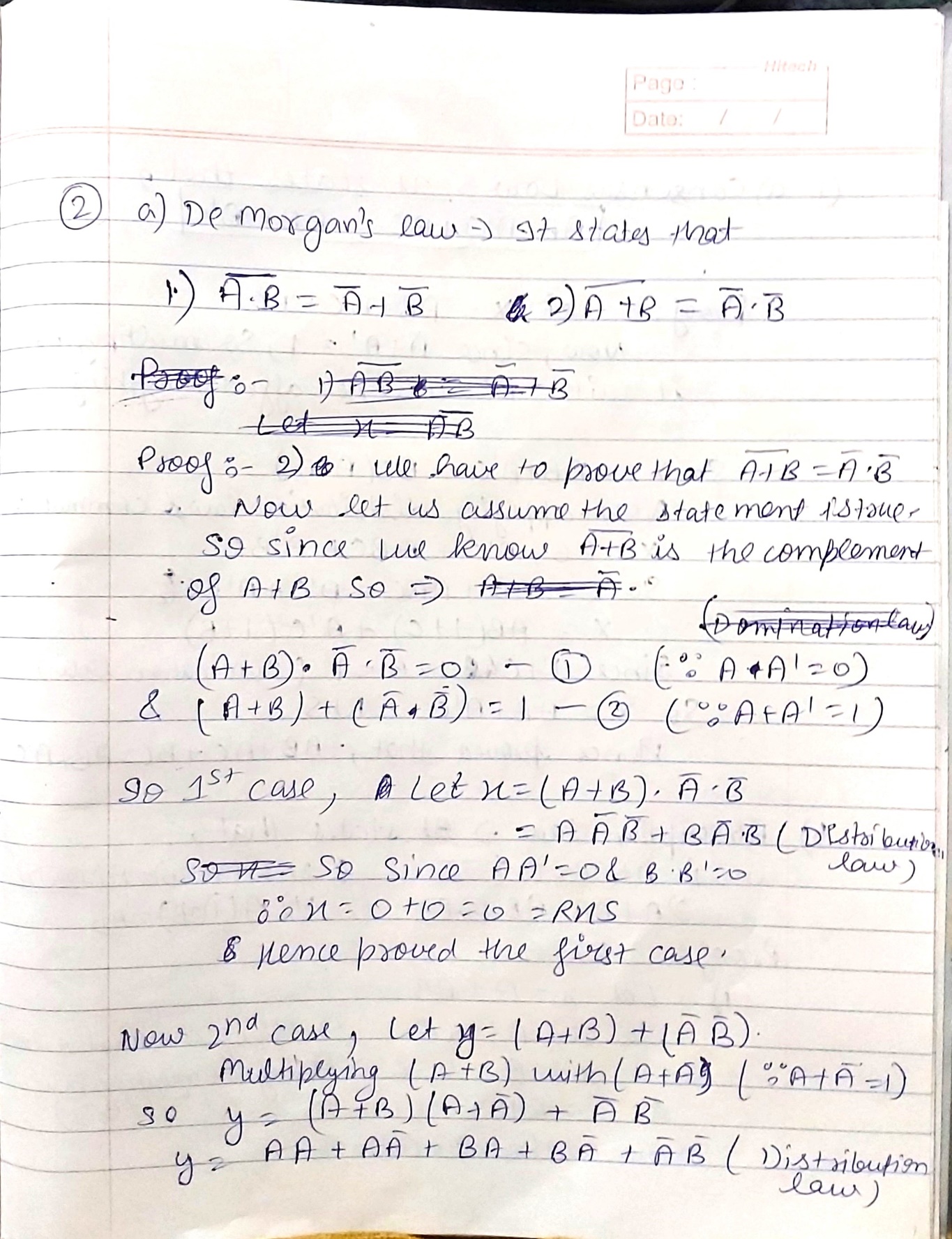
Q1. State and prove following laws of Boolean algebra.

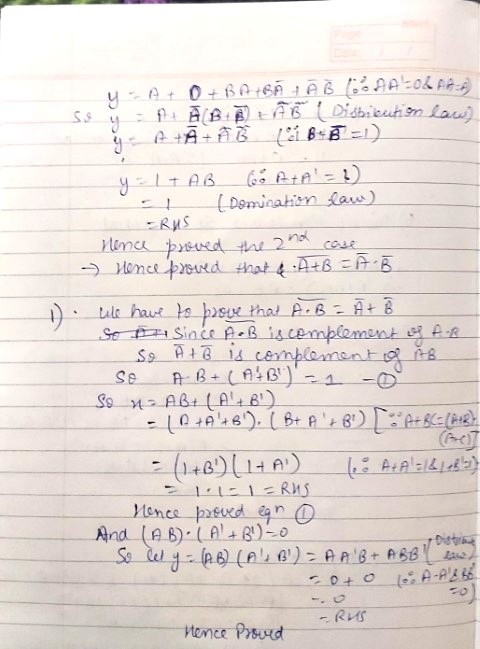
1. Consensus Law (b) Absorption Law

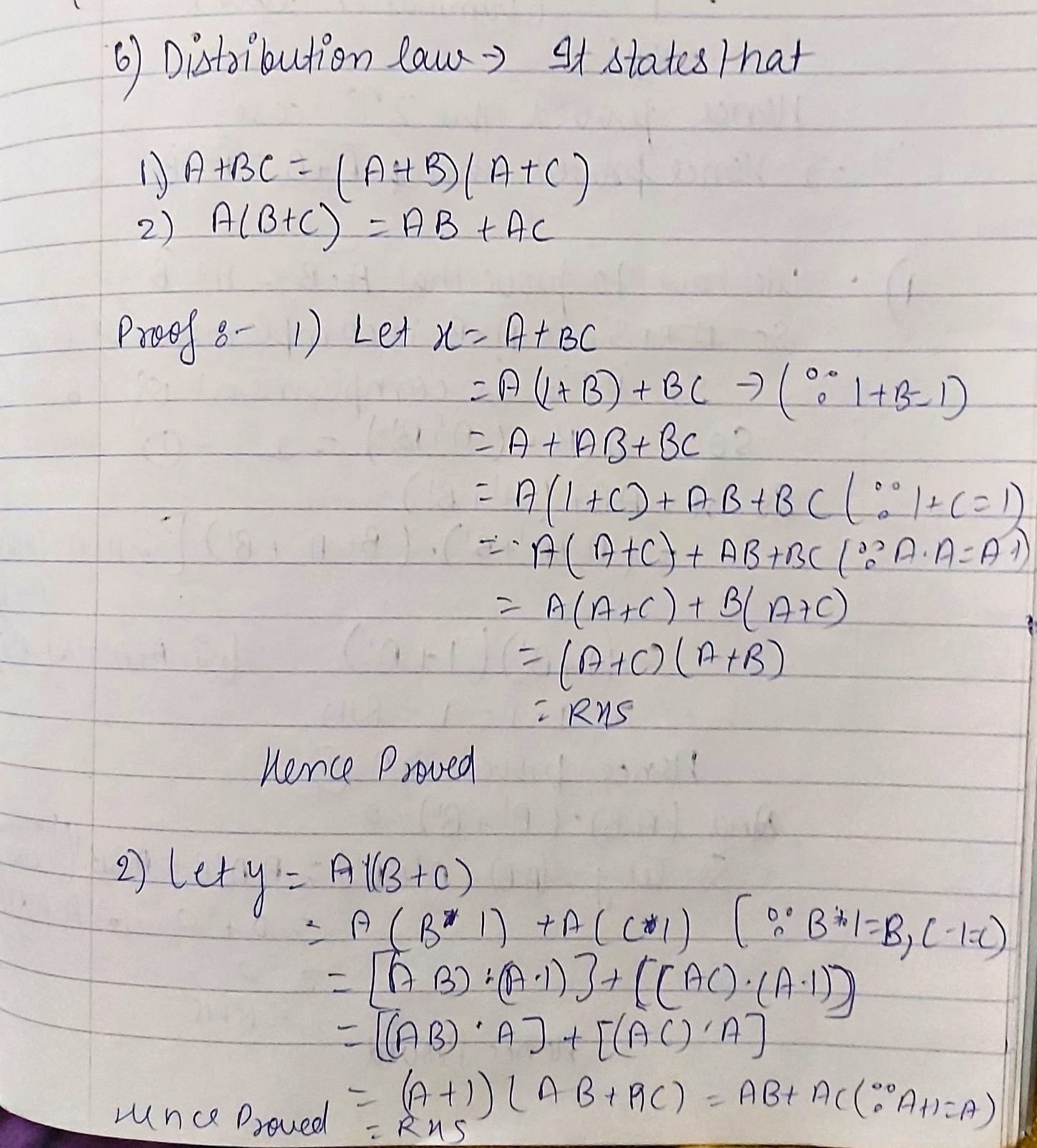




Q2. State and prove following laws of Boolean algebra. (a) De-Morgan’s Law (b) Distributive Law







Q3. Given Boolean expression: F(ABC) = A’BC + A’B’C’ + ABC

a) Write the truth table for the above Boolean expression.

b) Draw the schematic diagram for the above Boolean expression.

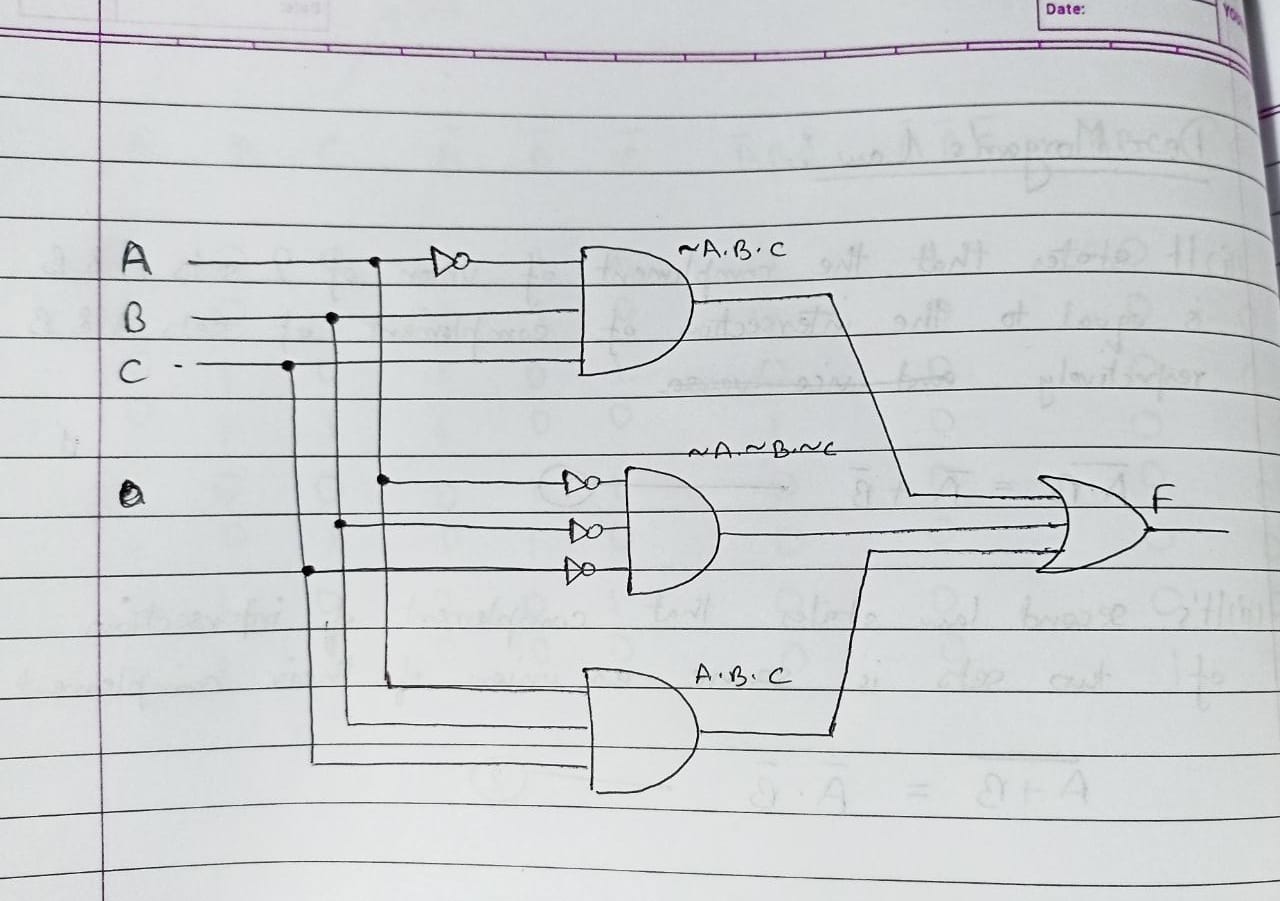
c) Write the Verilog code for the Boolean expression and then compare testbench generated waveform with the truth table to verify your circuit.

ANS:

->a) TRUTH TABLE:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | A’ | B’ | C’ | A’BC | A’B’C’ | ABC | F |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

->b) SCHEMATIC DIAGRAM:



-> c) CODE & WAVEFORM:

#Design code:

module dig\_gate(

input a,b,c,

output y);

endmodule

#Test Bench Code:

module tb\_dig\_gate;

reg A,B,C;

wire Y;

dig\_gate a1 (.a(A),.b(B),.c(C),.y(Y));

initial

begin

A = 0; B =0; C=0; #5;

A = 0; B =0; C=1; #5;

A = 0; B =1; C=0; #5;

A = 0; B =1; C=1; #5;

A = 1; B =0; C=0; #5;

A = 1; B =0; C=1; #5;

A = 1; B =1; C=0; #5;

A = 1; B =1; C=1; #5;

end

initial

begin

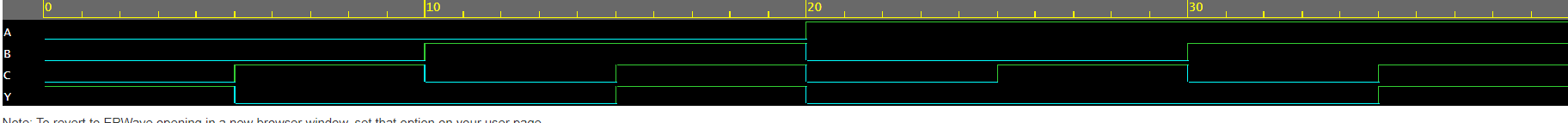
$dumpfile("dump.vcd");

$dumpvars(1);

end

endmodule

#Waveform:



Q4. Given Boolean expression F = AB’ + A’B

a) Write the truth table for the above Boolean expression.

b) Draw the schematic diagram for the above Boolean expression.

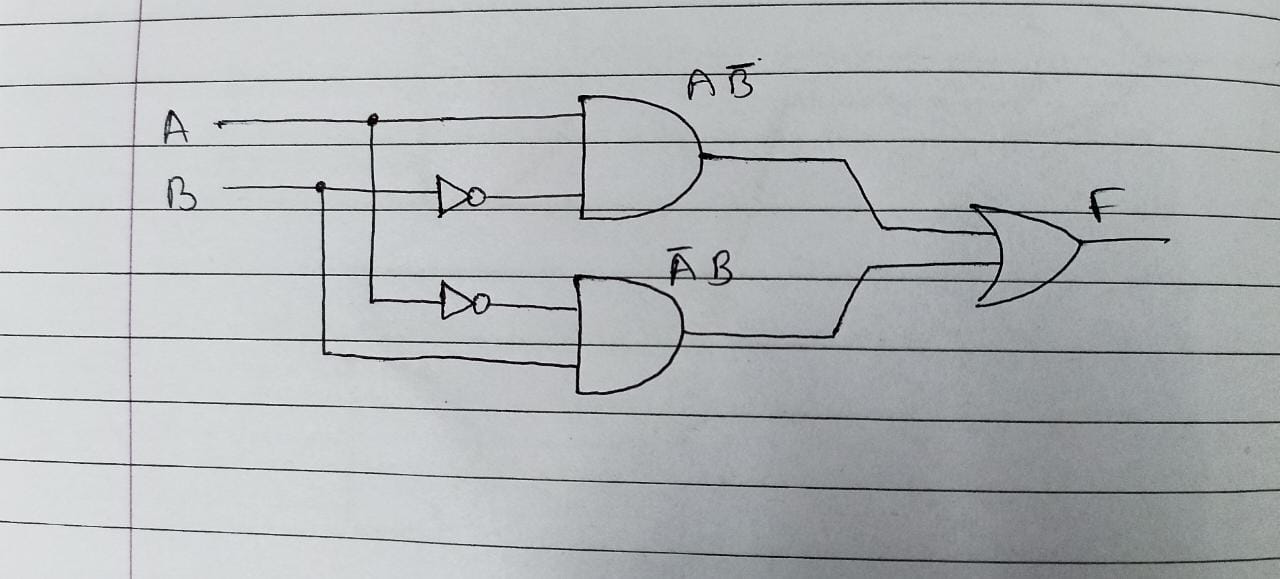
c) Write the Verilog code for the Boolean expression and then compare testbench generated waveform with the truth table to verify your circuit.

ANS:

a) TRUTH TABLE:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | A’ | B’ | AB’ | A’B | F |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |

b) SCHEMATIC DIAGRAM:



c)CODE AND WAVEFORM:

#Design Code:

module q4\_gate(

input a,b,

output y);

assign y = ((~a) && (b))||((a) && (~b));

endmodule

#Test Bench Code:

module tb\_q4\_gate;

reg A,B;

wire Y;

q4\_gate a1 (.a(A) ,.b(B),.y(Y));

initial

begin

A = 0; B =0; #5;

A = 0; B =1; #5;

A = 1; B =0; #5;

A = 1; B =1; #5;

end

initial

begin

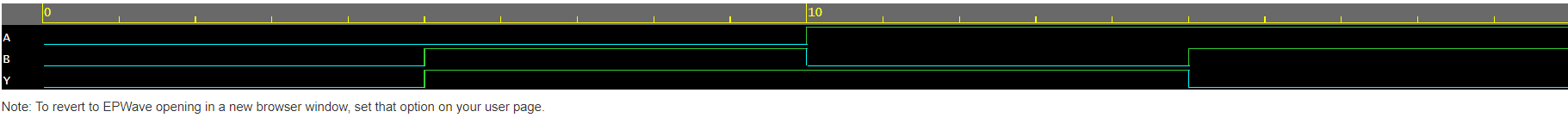
$dumpfile("dump.vcd");

$dumpvars(1);

end

endmodule

#Waveform:



Q5. Given Boolean expression F = A’B’C + A’BC+AB’C’+ABC

a) Write the truth table for the above Boolean expression.

b) Draw the schematic diagram for the above Boolean expression.

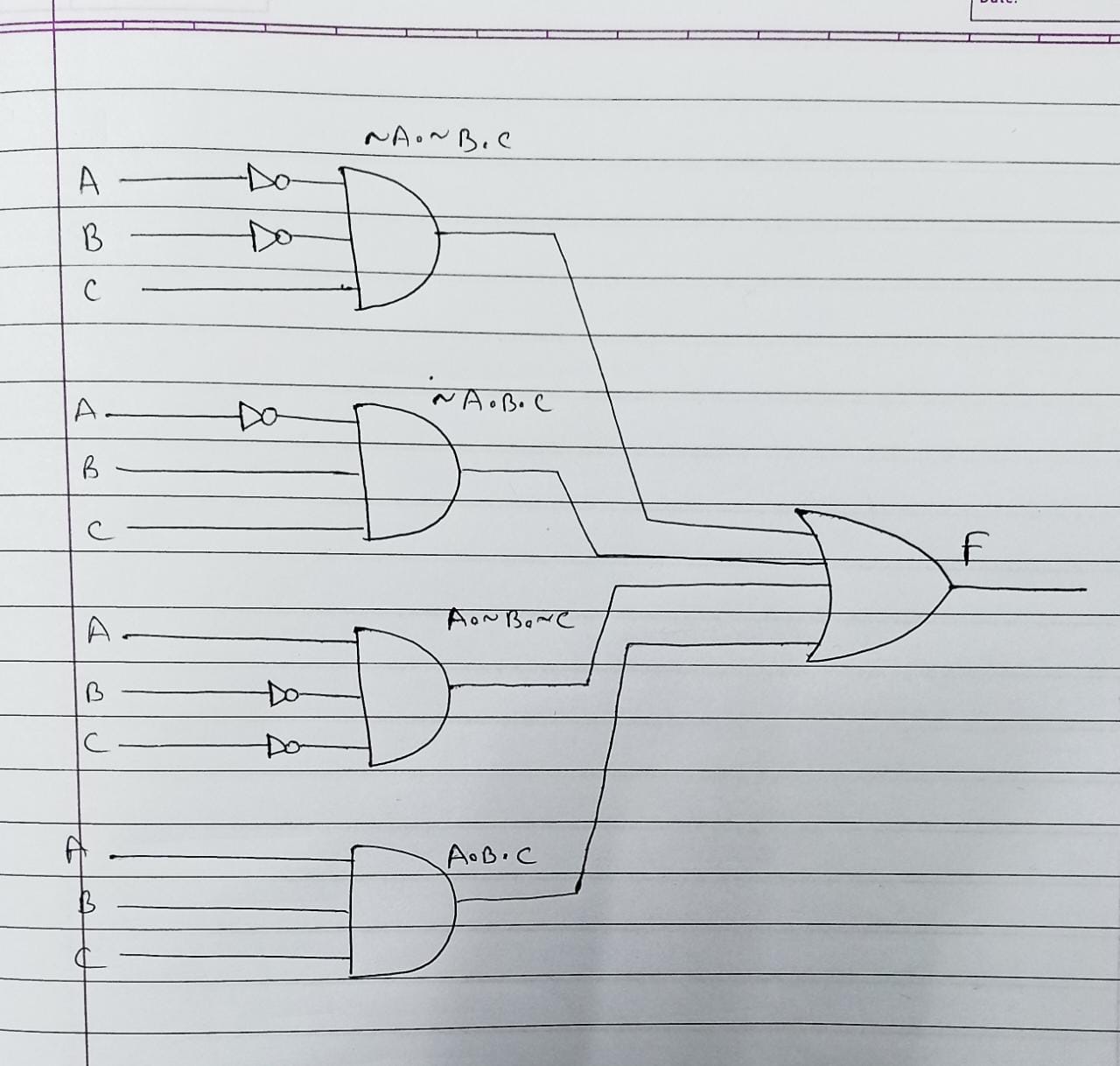
c) Write the Verilog code for the Boolean expression and then compare testbench generated waveform with the truth table to verify your circuit.

ANS:

a) TRUTH TABLE:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | A’ | B’ | C’ | A’B’C | A’BC | AB’C’ | ABC | F |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

b) SCHEMATIC DIAGRAM:



c)CODE AND WAVEFORM:

#Design Code:

module dig\_gate(

input a,b,c,

output y);

endmodule

#Test Bench Code:

module tb\_dig\_gate;

reg A,B,C;

wire Y;

dig\_gate a1 (.a(A),.b(B),.c(C),.y(Y));

initial

begin

A = 0; B =0; C=0; #5;

A = 0; B =0; C=1; #5;

A = 0; B =1; C=0; #5;

A = 0; B =1; C=1; #5;

A = 1; B =0; C=0; #5;

A = 1; B =0; C=1; #5;

A = 1; B =1; C=0; #5;

A = 1; B =1; C=1; #5;

end

initial

begin

$dumpfile("dump.vcd");

$dumpvars(1);

end

endmodule

#Waveform:

