DLD LAB WIN_2021

LAB-1

Aim: To simulate different types of logic gates (NOT, NAND, NOR, XOR, XNOR) in different ways of modelling (DATAFLOW, BEHAVIORAL, STRUCTURAL)

Tools used: vivado software

Truth tables:

Truth tables :-

Not gate: Y = ~A

A	Y(A complement)
0	1
1	0

Nand gate :- Y = ~(A&B)

A	B	Y(A and B)
0	0	1
0	1	1
1	0	1
1	1	0

NOR gate :- Y = ~(A+B)

A	B	Y(A or B)
0	0	1
0	1	0
1	0	0
1	1	0

XOR gate :- Y = A^B

A	B	Y(A xor B)
0	0	0
0	1	1
1	0	1
1	1	0

XNOR gate:
Y = ~(A^B)

A	B	Y(A XNOR B)
0	0	1
0	1	0
1	0	0
1	1	1

Boolean expressions:

Boolean expressions for the following gates :-

Not gate :-

$Y = \sim A$

NOR gate :-

$Y = \sim(A+B)$

XNOR gate :-

$Y = \sim(A \wedge B)$

NAND gate :-

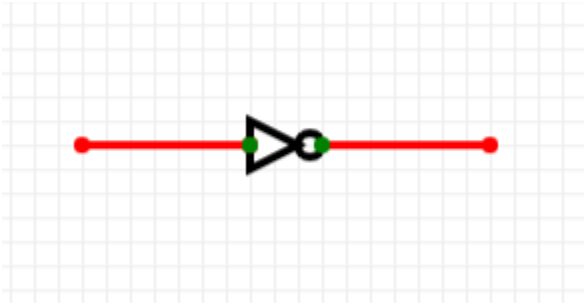
$Y = \sim(A \& B)$

XOR gate :-

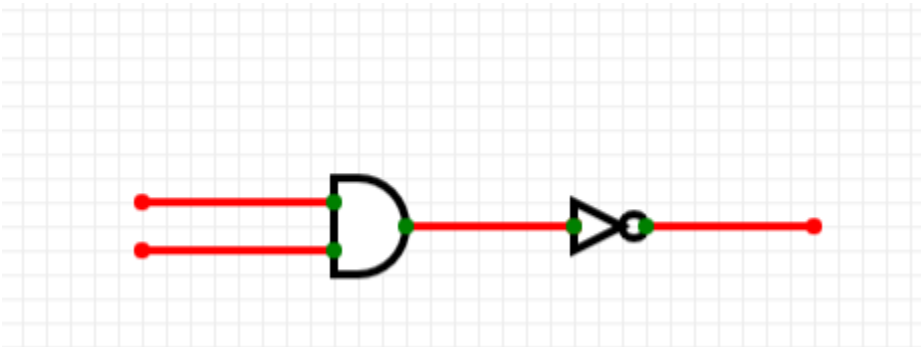
$Y = A \wedge B$

Circuit diagrams:

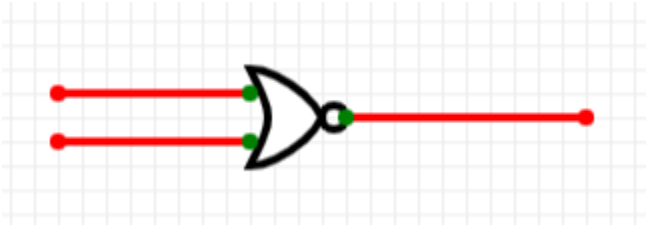
Not gate:



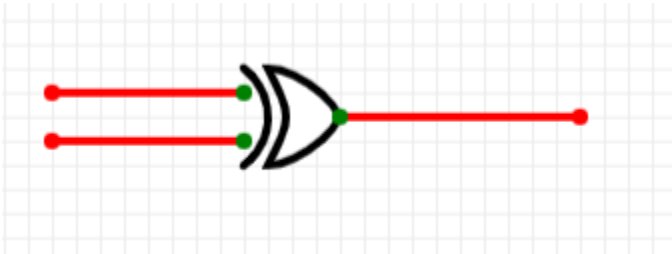
Nand gate:



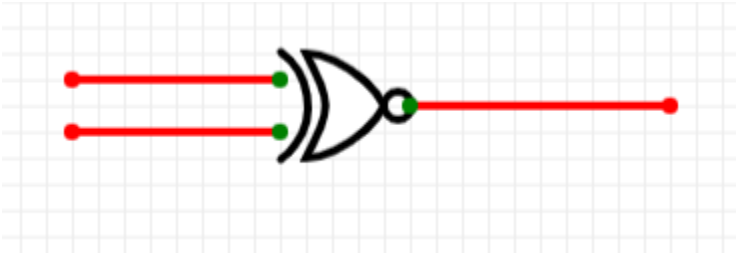
Nor gate:



Xor gate:

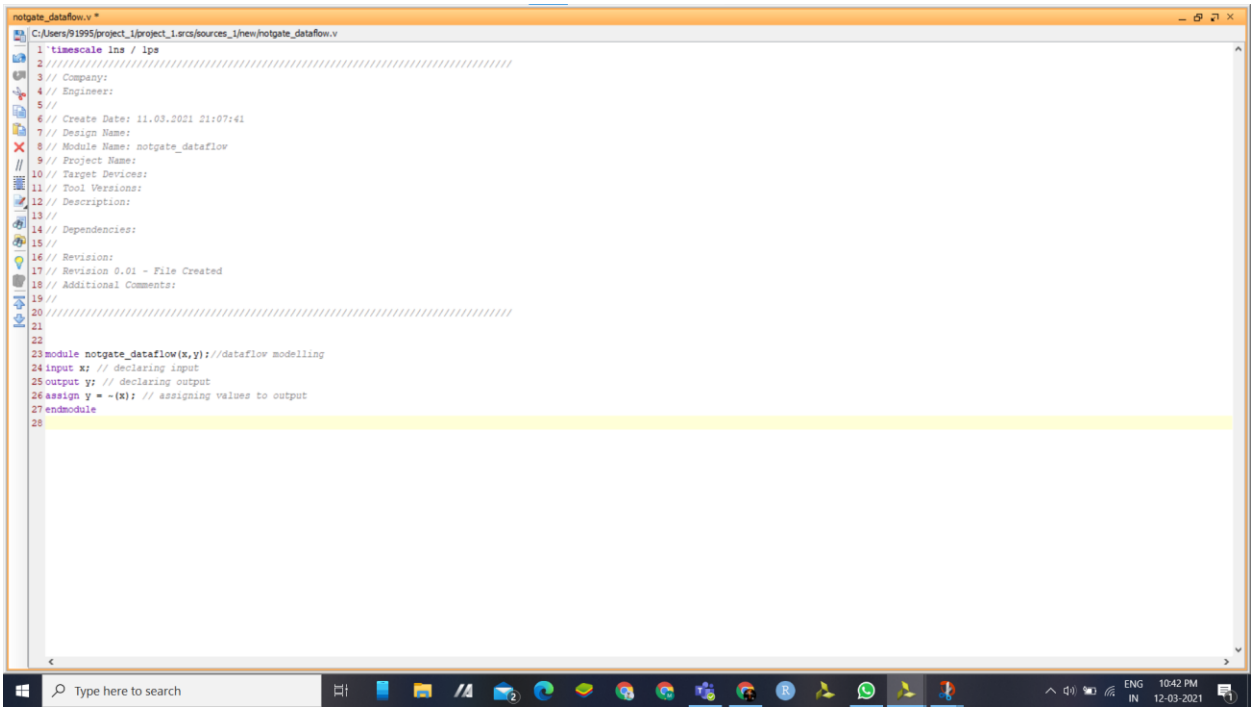


Xnor gate:



Codes:

Not gate:



module notgate_dataflow(x,y);//dataflow modelling

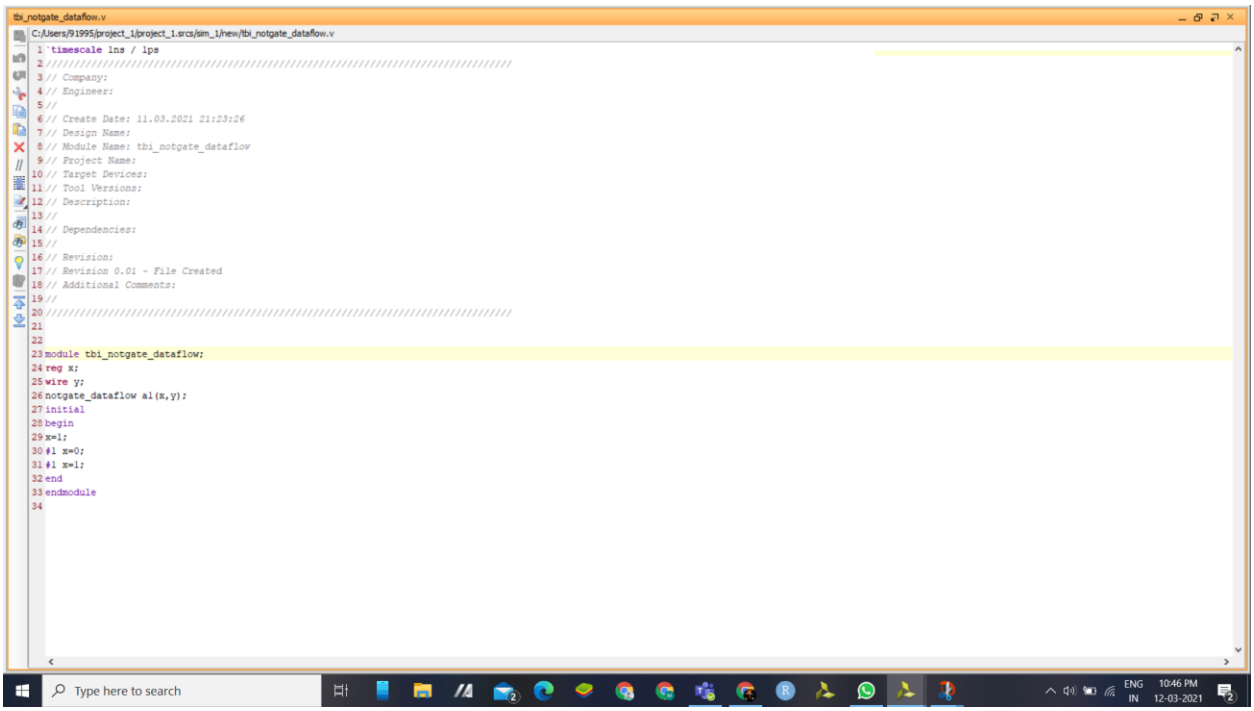
input x; // declaring input

output y; // declaring output

assign y = ~(x); // assigning values to output

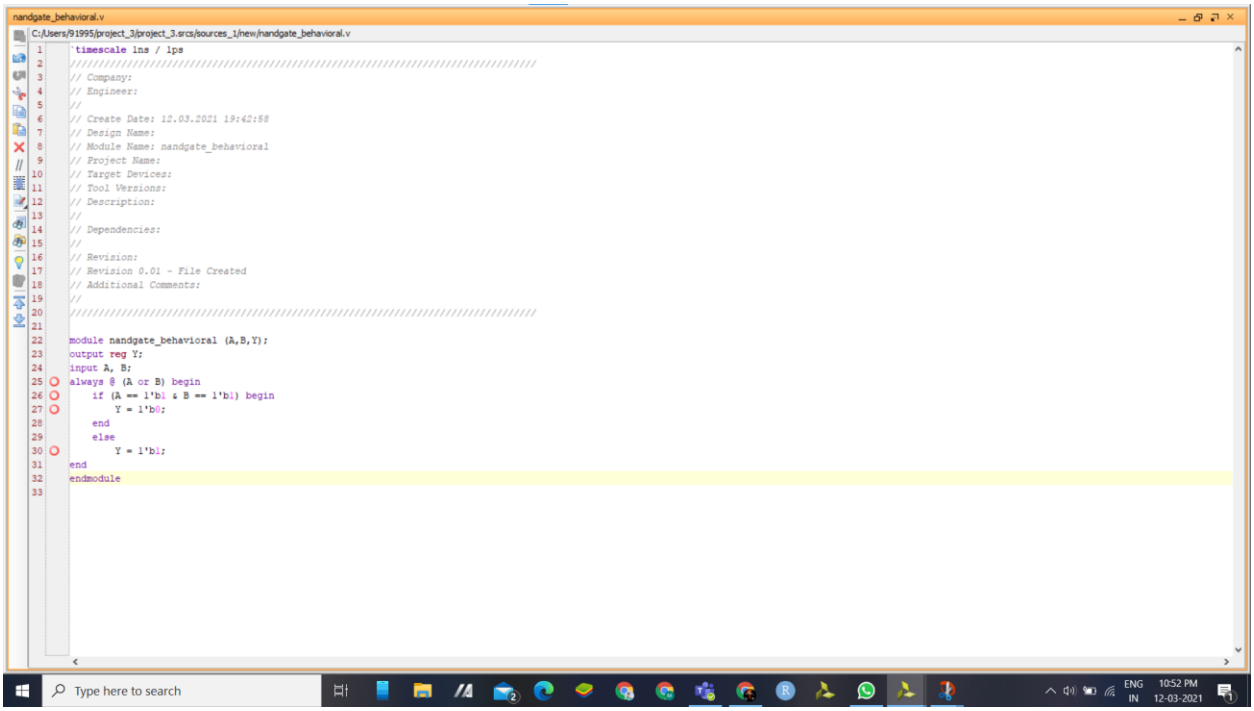
Endmodule

Test bench code:



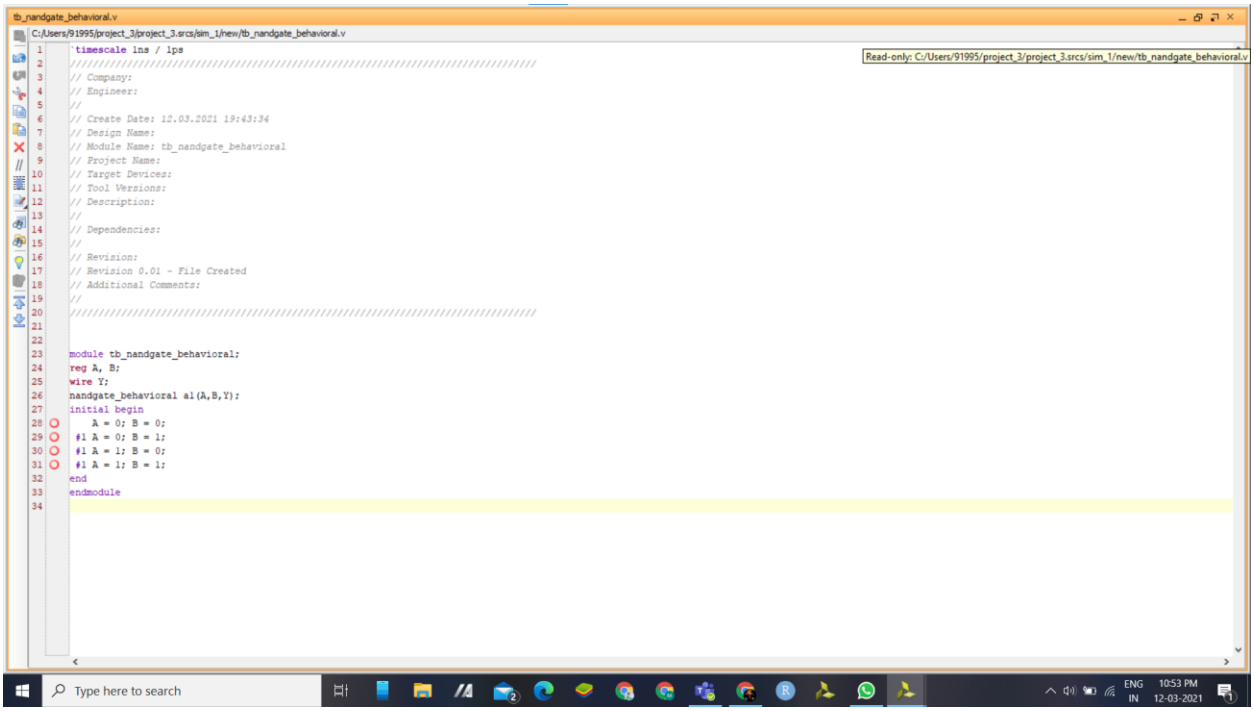

```
module tbi_notgate_dataflow;

reg x;

wire y;

notgate_dataflow a1(x,y);

initial

begin

x=1;

#1 x=0;

#1 x=1;

end

Endmodule
```

Nand gate:




```
module nandgate_behavioral (A,B,Y);
output reg Y;
input A, B;
always @ (A or B) begin
    if (A == 1'b1 & B == 1'b1) begin
        Y = 1'b0;
    end
    else
        Y = 1'b1;
end
endmodule
```

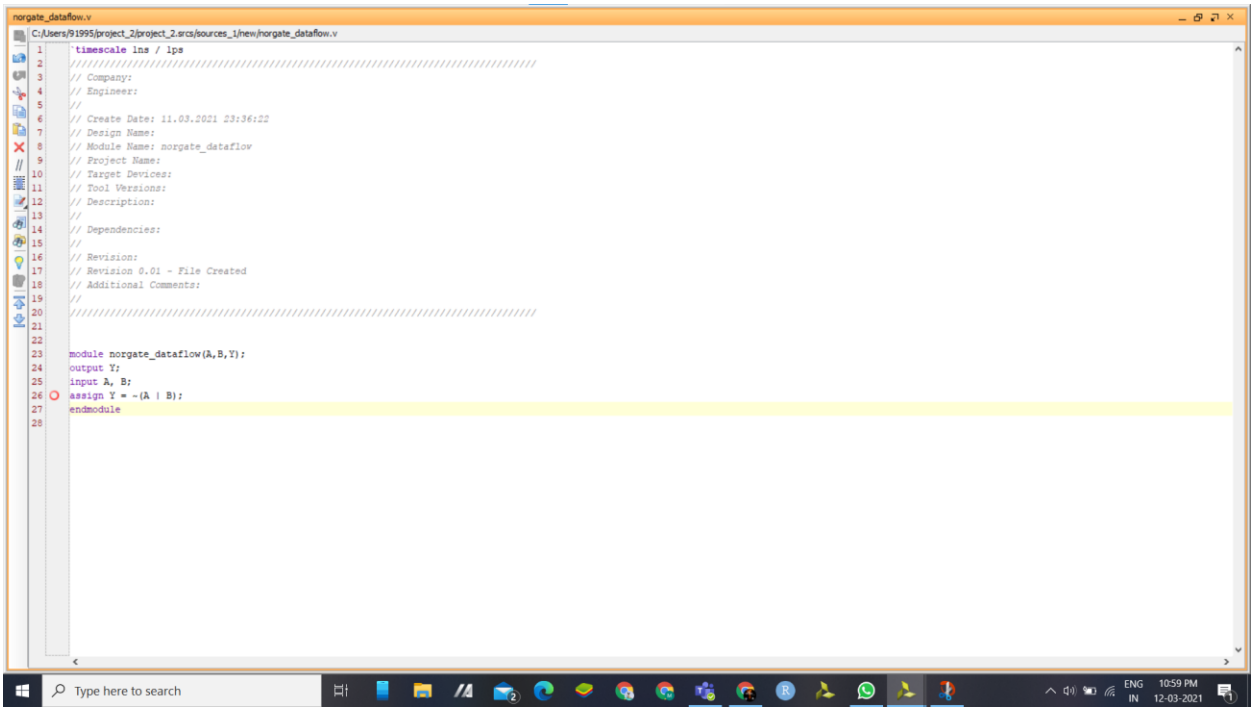
Test bench code:




```
module tb_nandgate_behavioral;
reg A, B;
wire Y;
nandgate_behavioral a1(A,B,Y);
initial begin
    A = 0; B = 0;
#1 A = 0; B = 1;
#1 A = 1; B = 0;
#1 A = 1; B = 1;
end
```

endmodule

Nor gate:



module norgate_dataflow(A,B,Y);

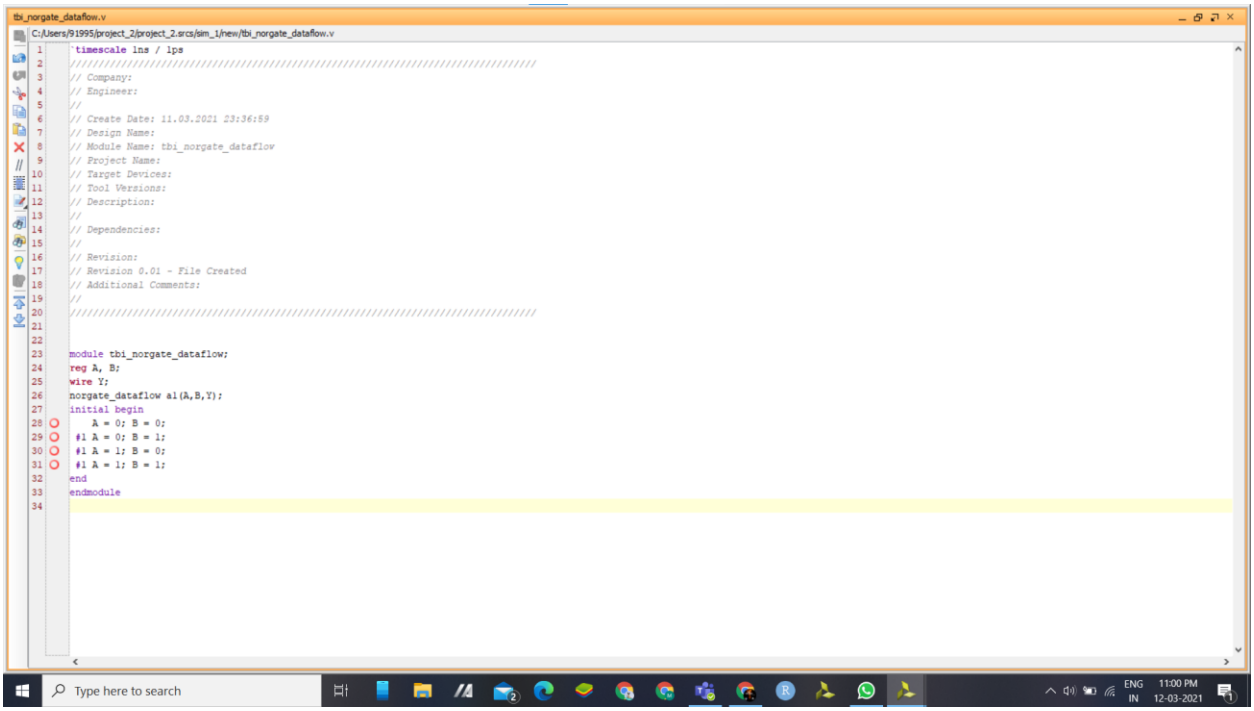
output Y;

input A, B;

assign Y = ~(A | B);

endmodule

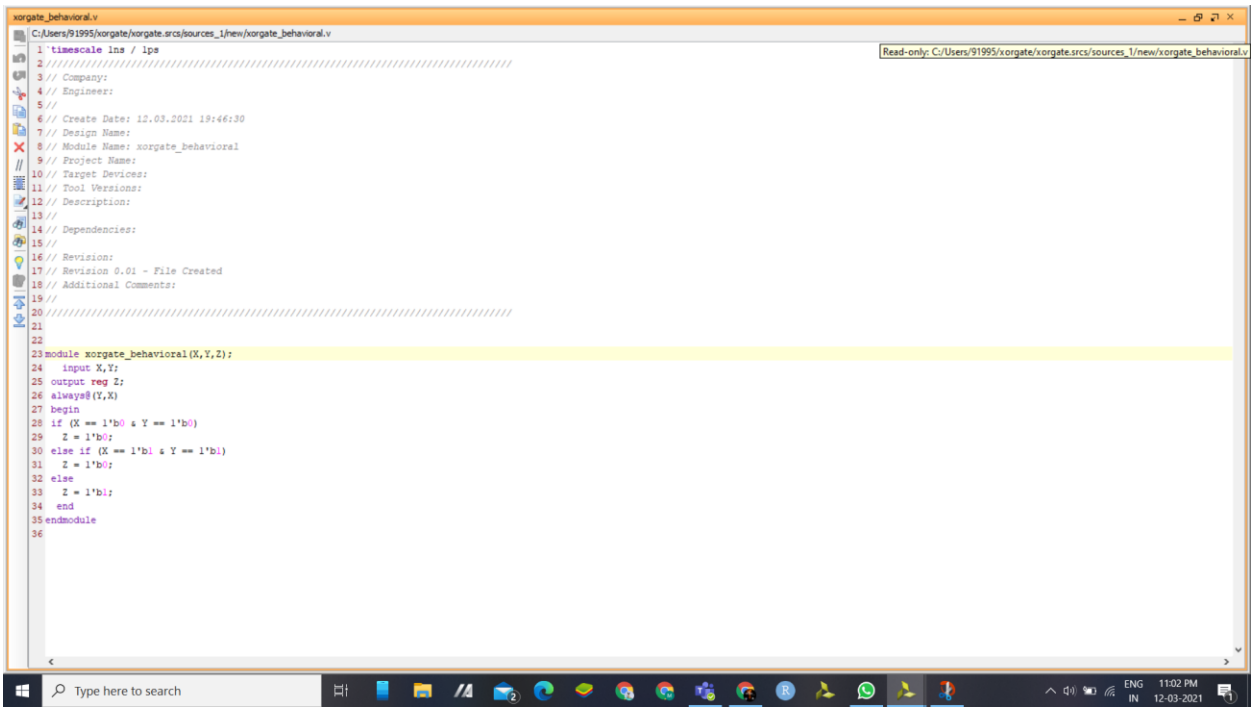
Test bench code:




```
module tbi_norgate_dataflow;

reg A, B;

wire Y;

norgate_dataflow a1(A,B,Y);

initial begin

    A = 0; B = 0;

#1 A = 0; B = 1;

#1 A = 1; B = 0;

#1 A = 1; B = 1;

end
```

endmodule

Xor gate:



module xorgate_behavioral(X,Y,Z);

input X,Y;

output reg Z;

always@(Y,X)

begin

if (X == 1'b0 & Y == 1'b0)

Z = 1'b0;

else if (X == 1'b1 & Y == 1'b1)

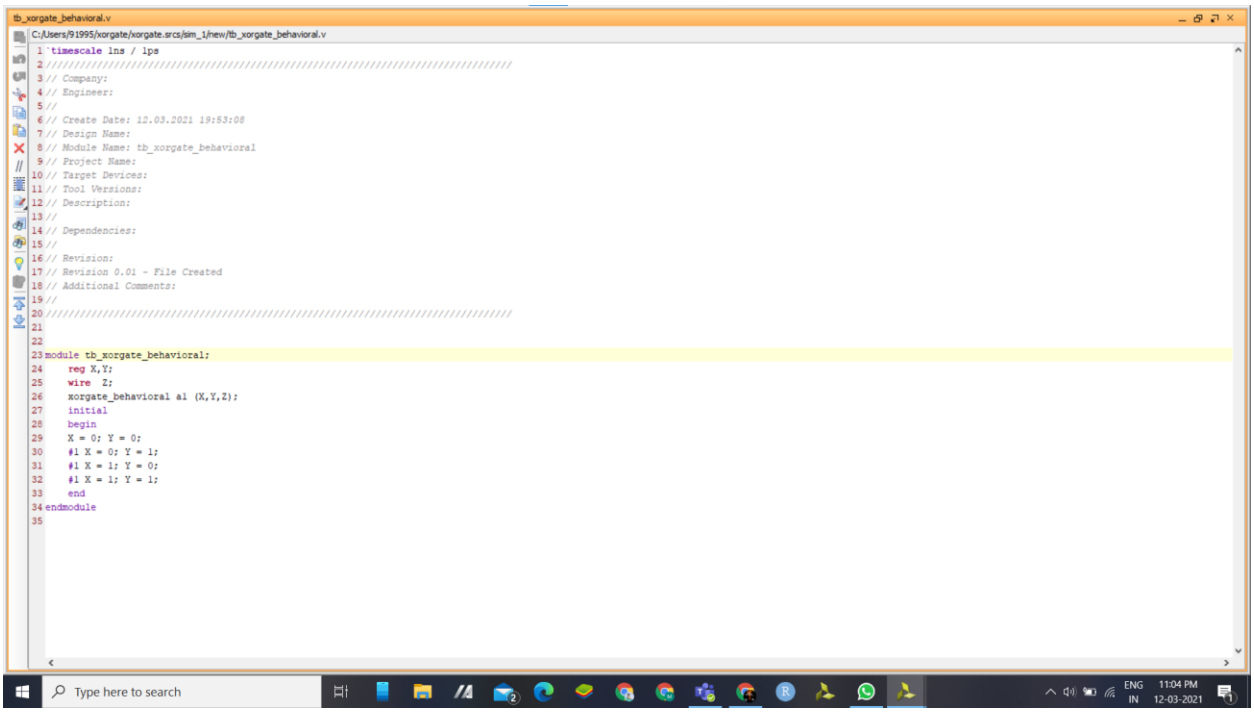
Z = 1'b0;

else

Z = 1'b1;

```
  end
endmodule
```

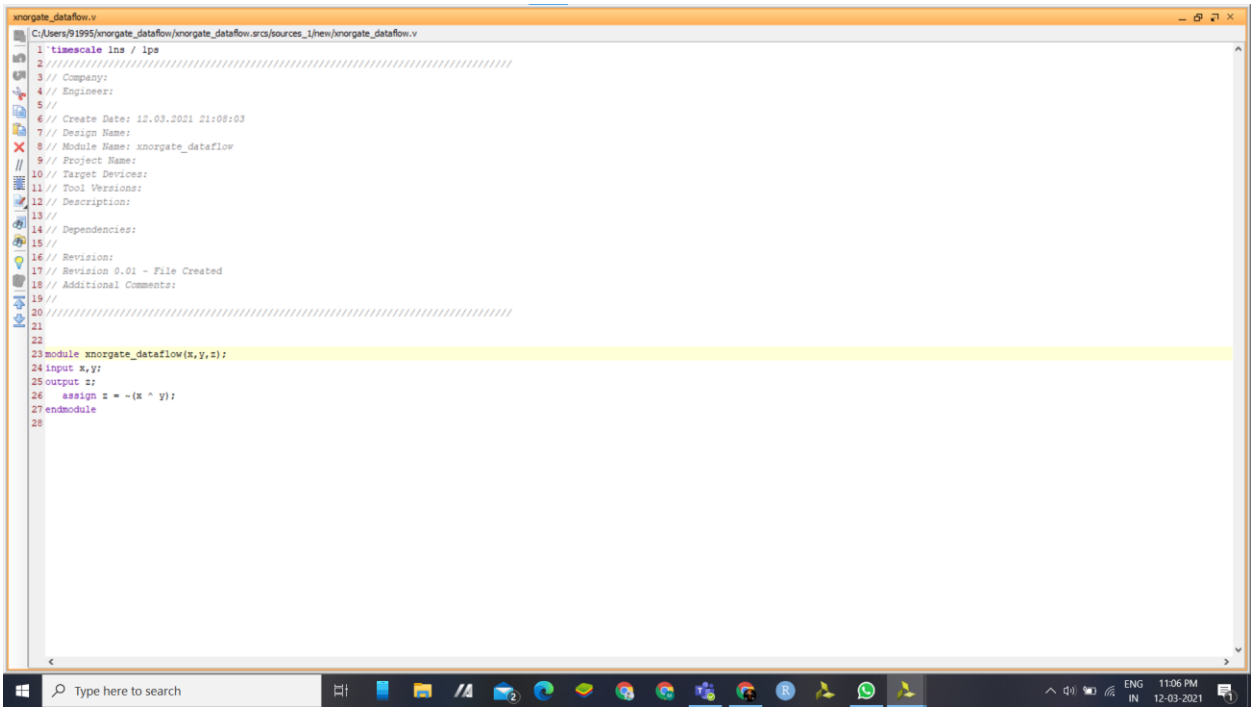
Test bench code:




```
module tb_xorgate_behavioral;
    reg X,Y;
    wire  Z;
    xorgate_behavioral a1 (X,Y,Z);
    initial
    begin
    X = 0; Y = 0;
    #1 X = 0; Y = 1;
    #1 X = 1; Y = 0;
    #1 X = 1; Y = 1;
```

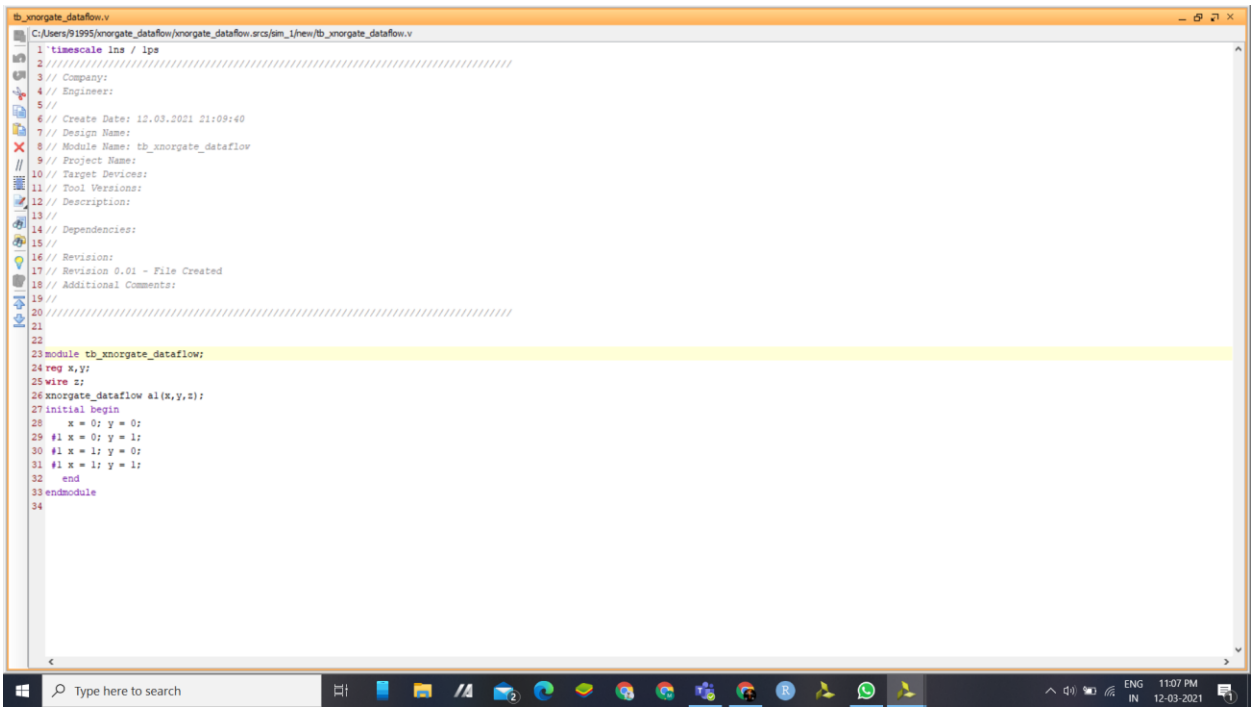
```
   end
endmodule
```

xnor gate:




```
module xnorgate_dataflow(x,y,z);
input x,y;
output z;
   assign z = ~(x ^ y);
endmodule
```

Test bench code:



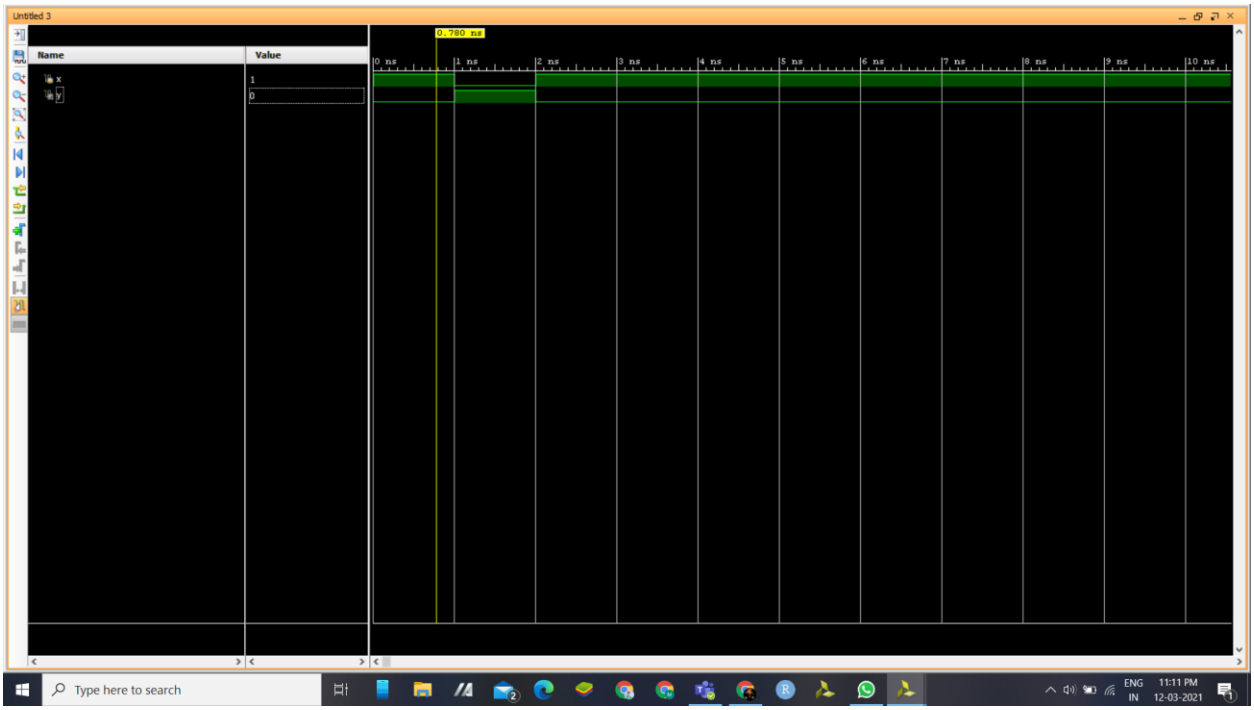

```
module tb_xnorgate_dataflow;
reg x,y;
wire z;
xnorgate_dataflow a1(x,y,z);
initial begin
    x = 0; y = 0;
#1 x = 0; y = 1;
#1 x = 1; y = 0;
#1 x = 1; y = 1;
  end
endmodule
```

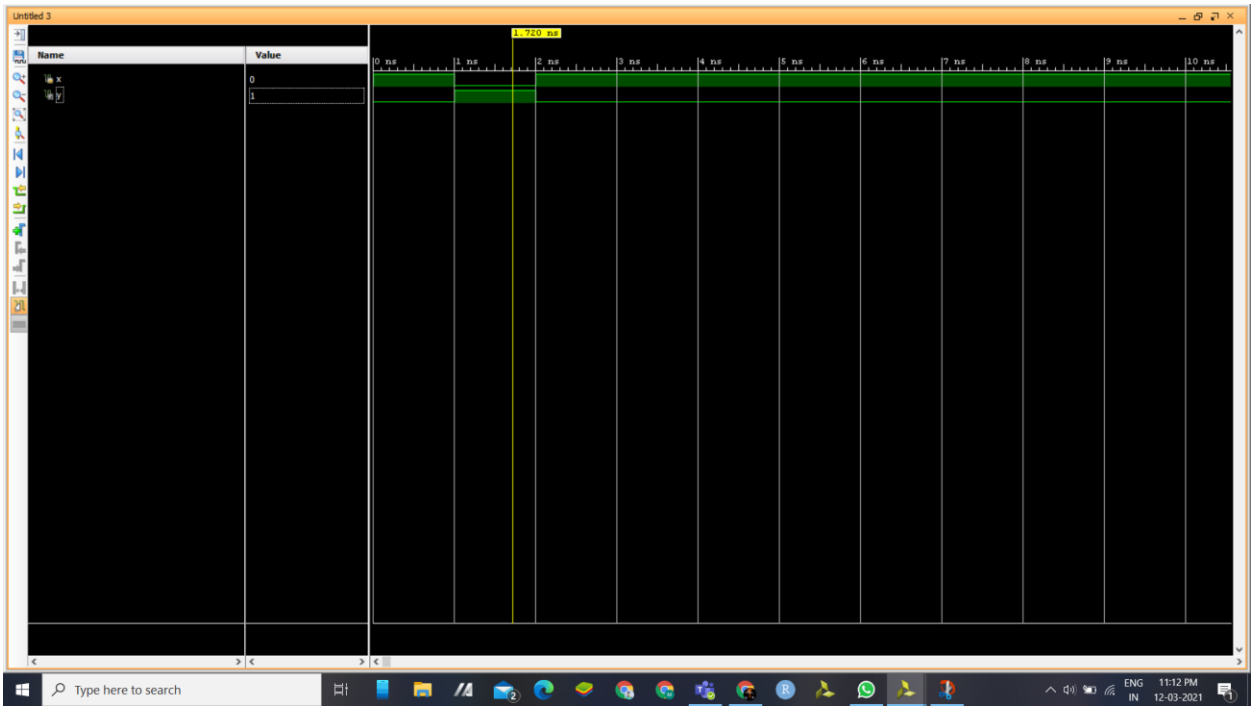
Results of codes:

>>>>><<<<<

DATE:12-03-2021 NAME:MAJJIGA JASWANTH REGNO:20BCD7171

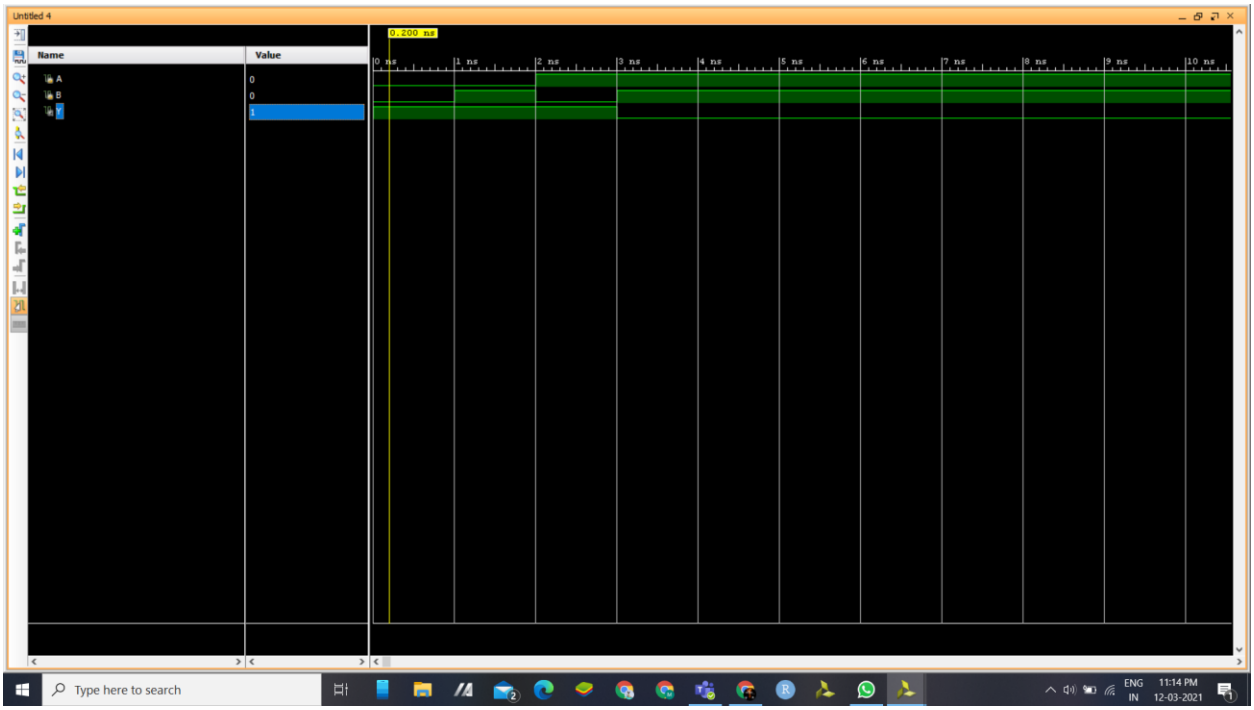
Not gate:

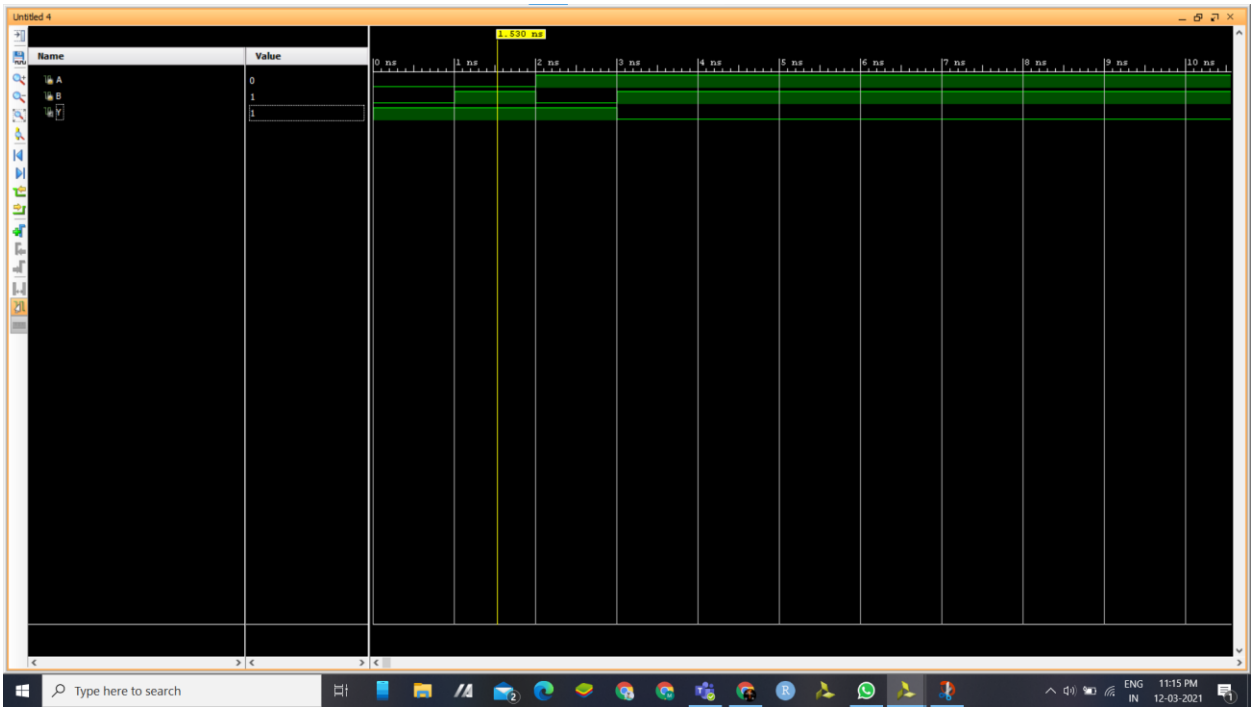




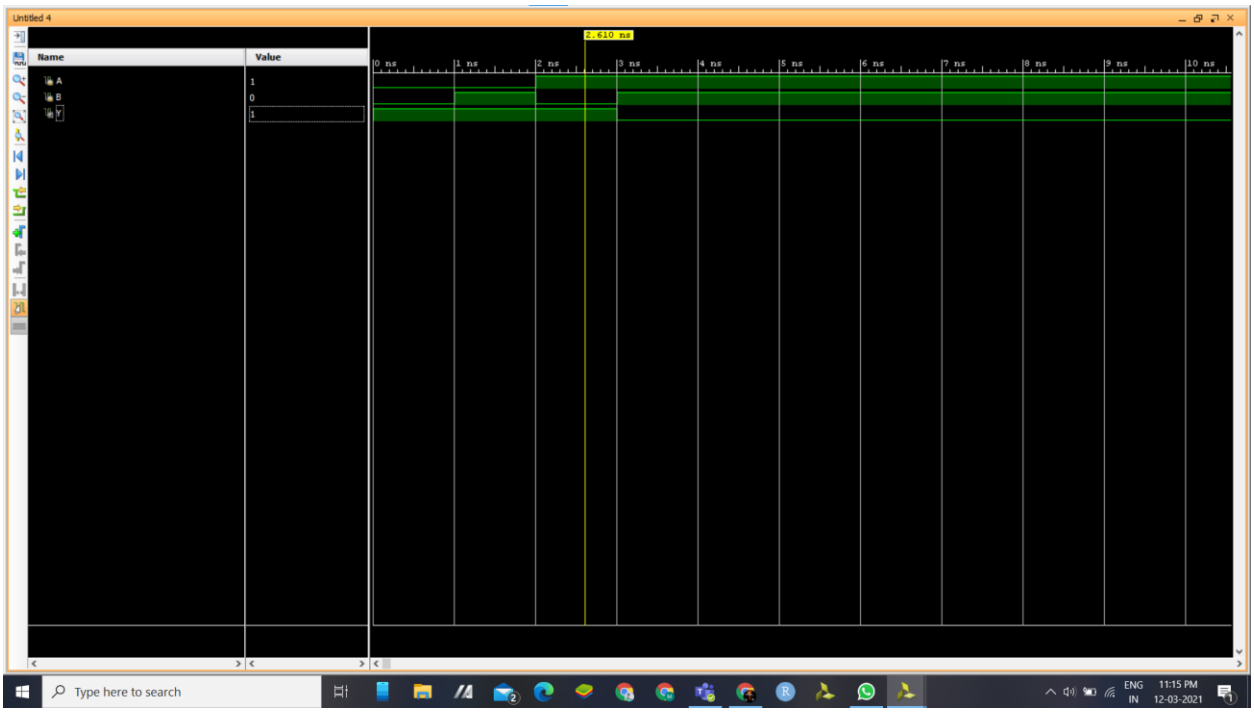
DATE:12-03-2021 NAME:MAJJIGA JASWANTH REGNO:20BCD7171

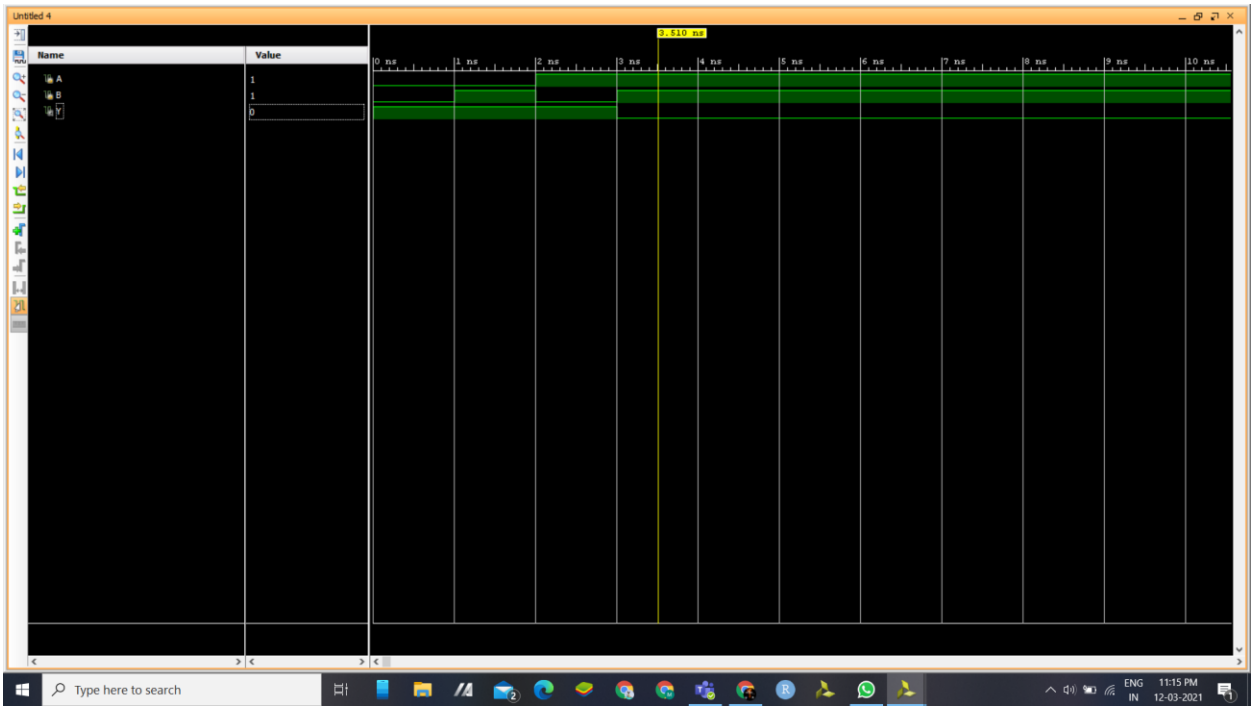
Nand gate:





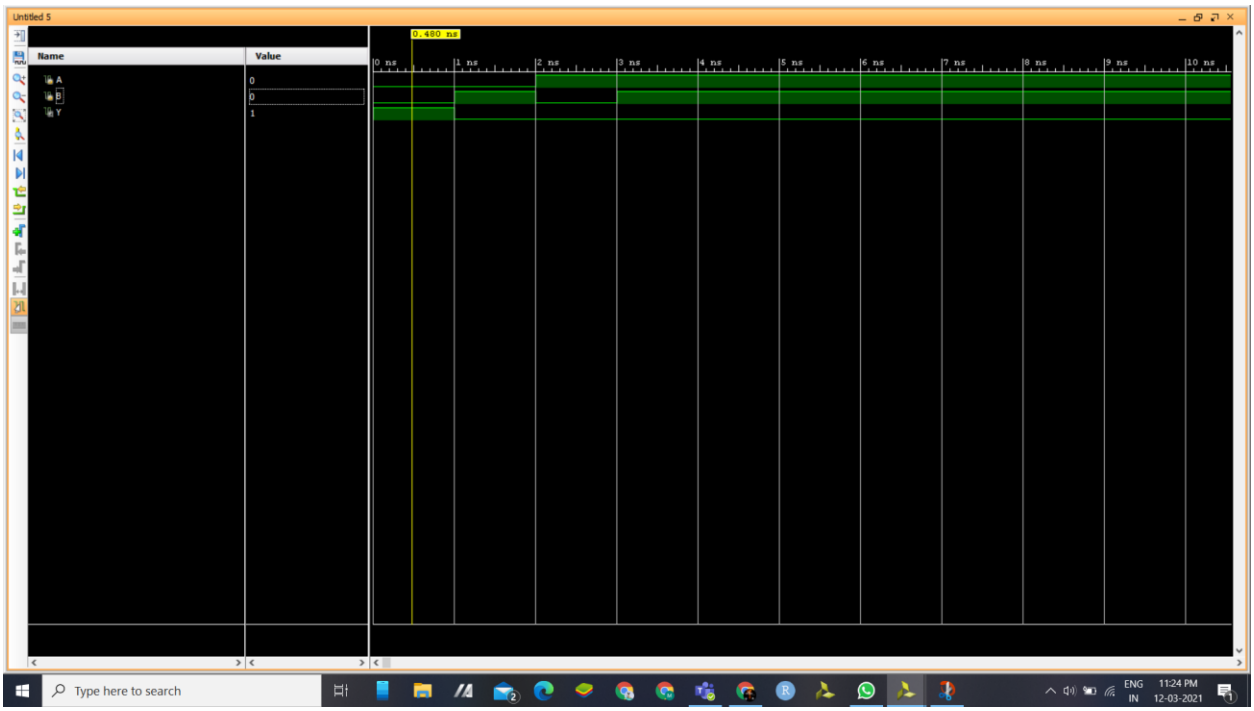
DATE:12-03-2021 NAME:MAJJIGA JASWANTH REGNO:20BCD7171

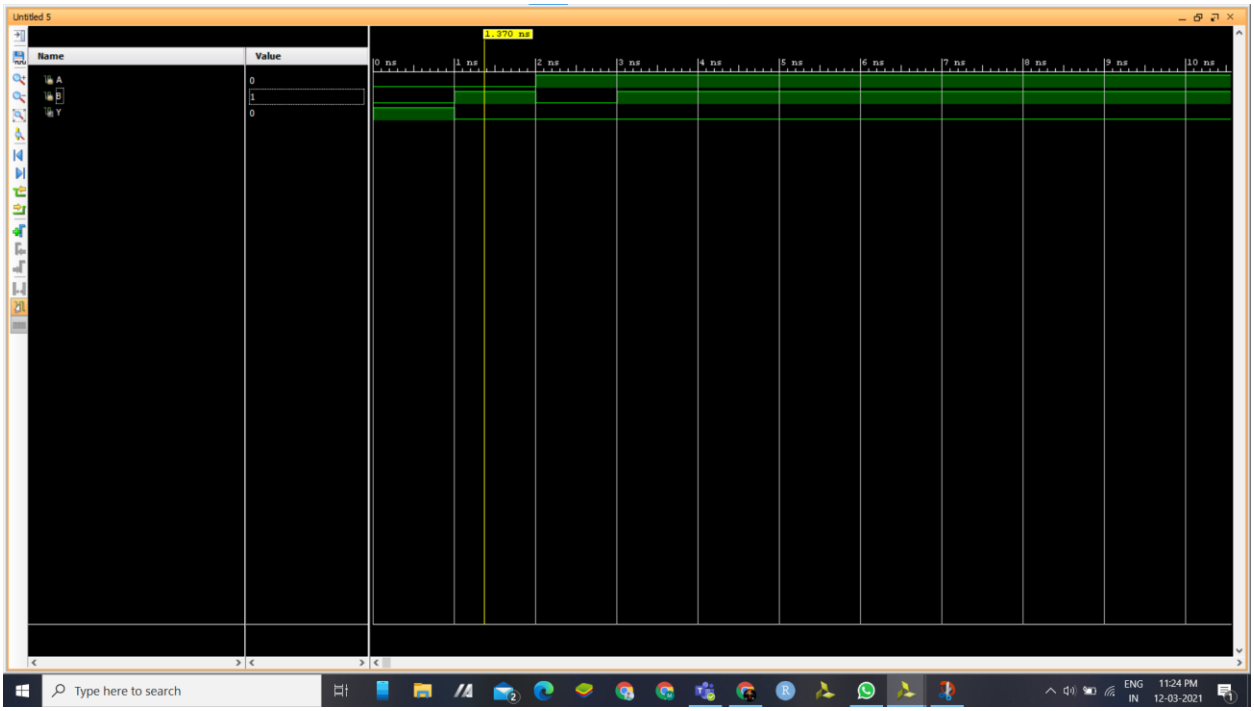




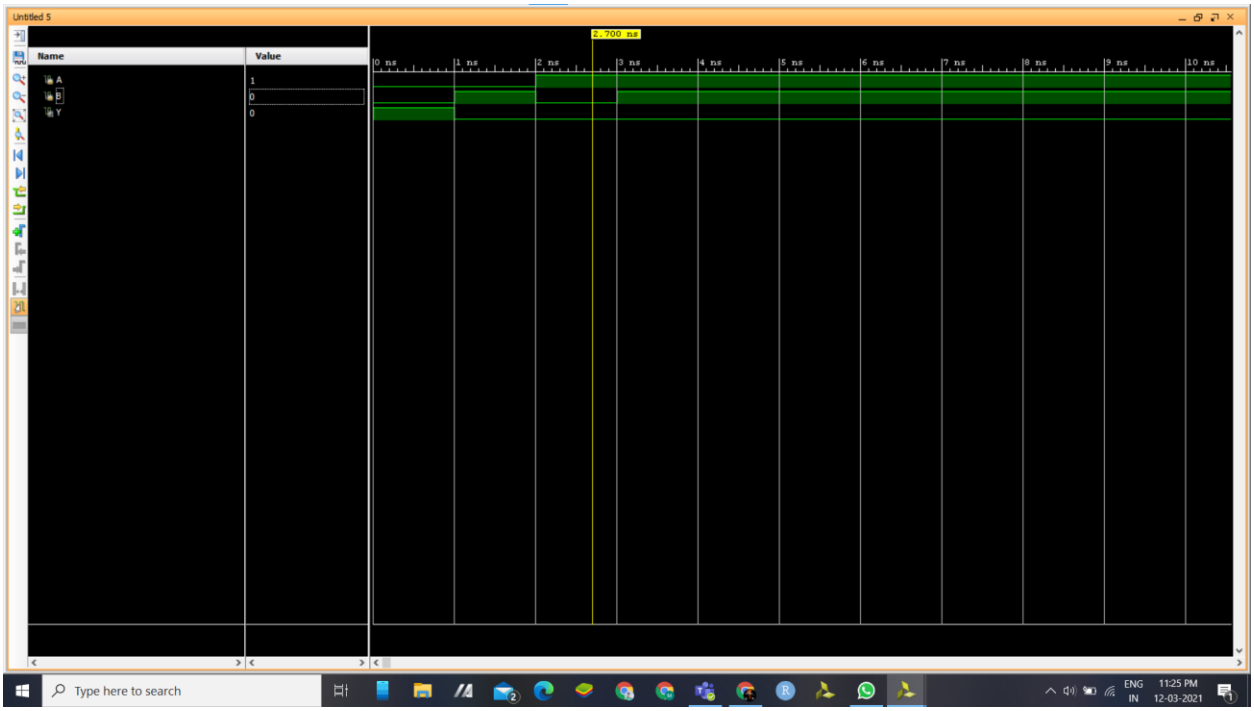
DATE:12-03-2021 NAME:MAJJIGA JASWANTH REGNO:20BCD7171

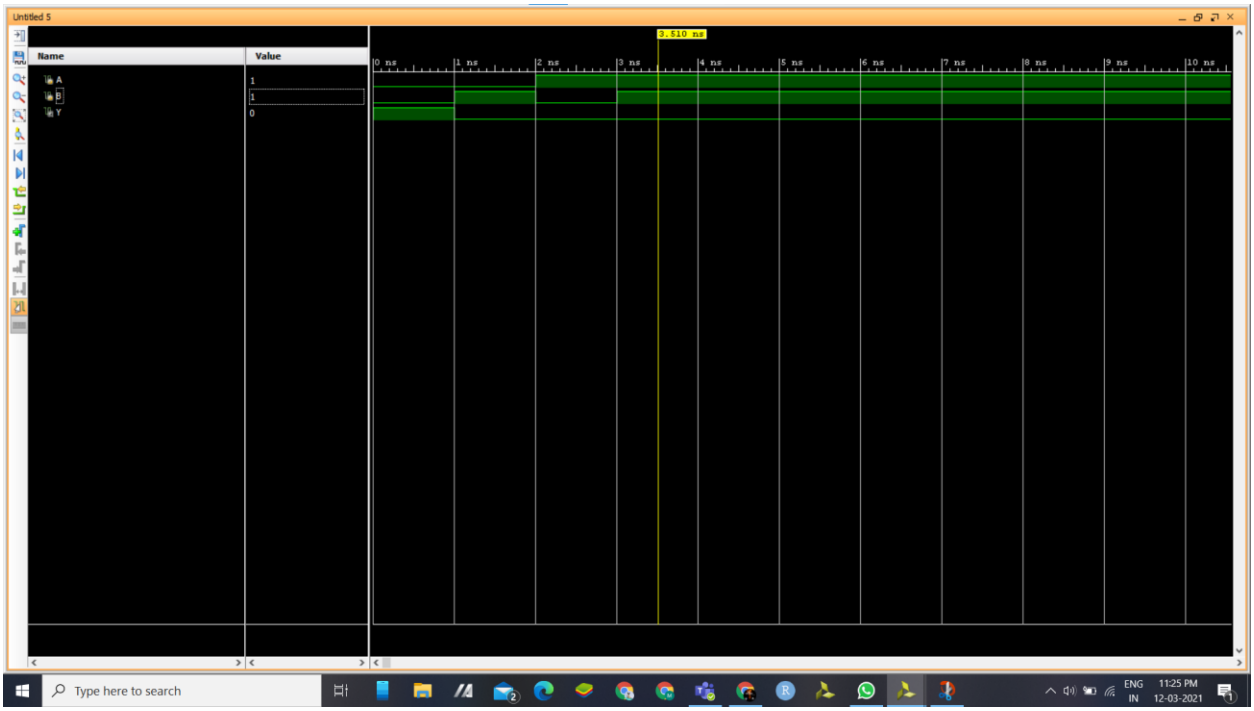
Nor gate:



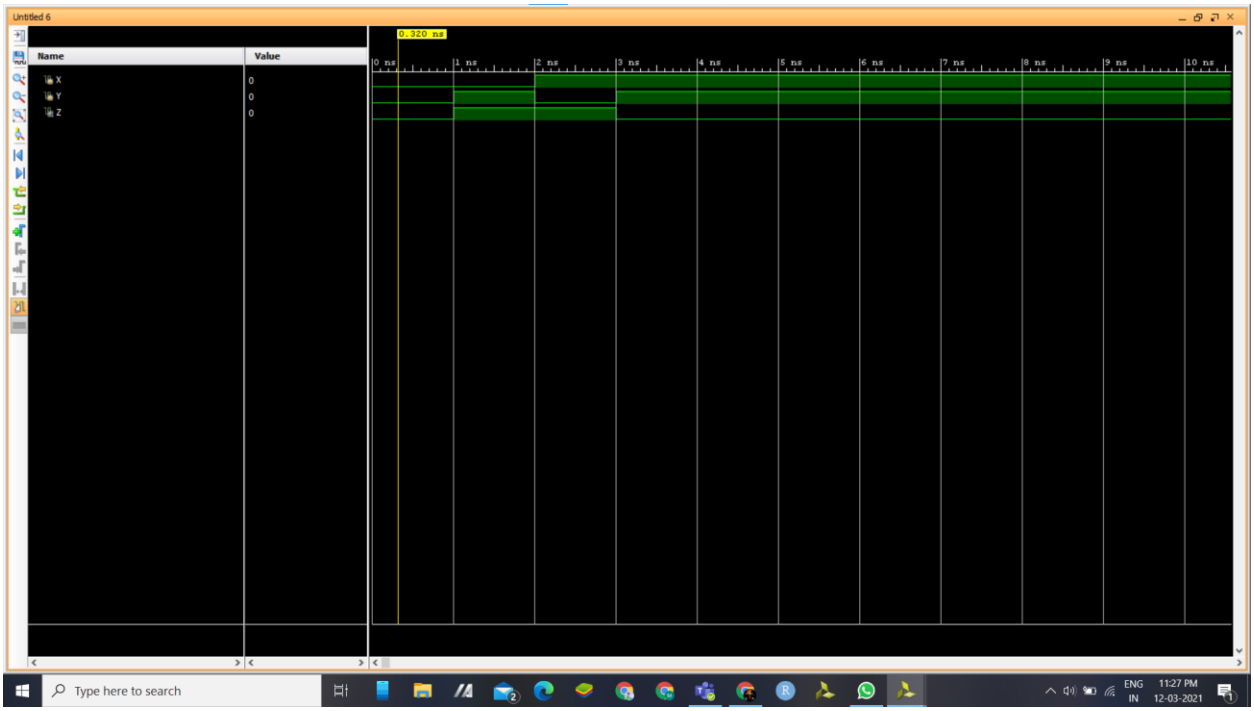


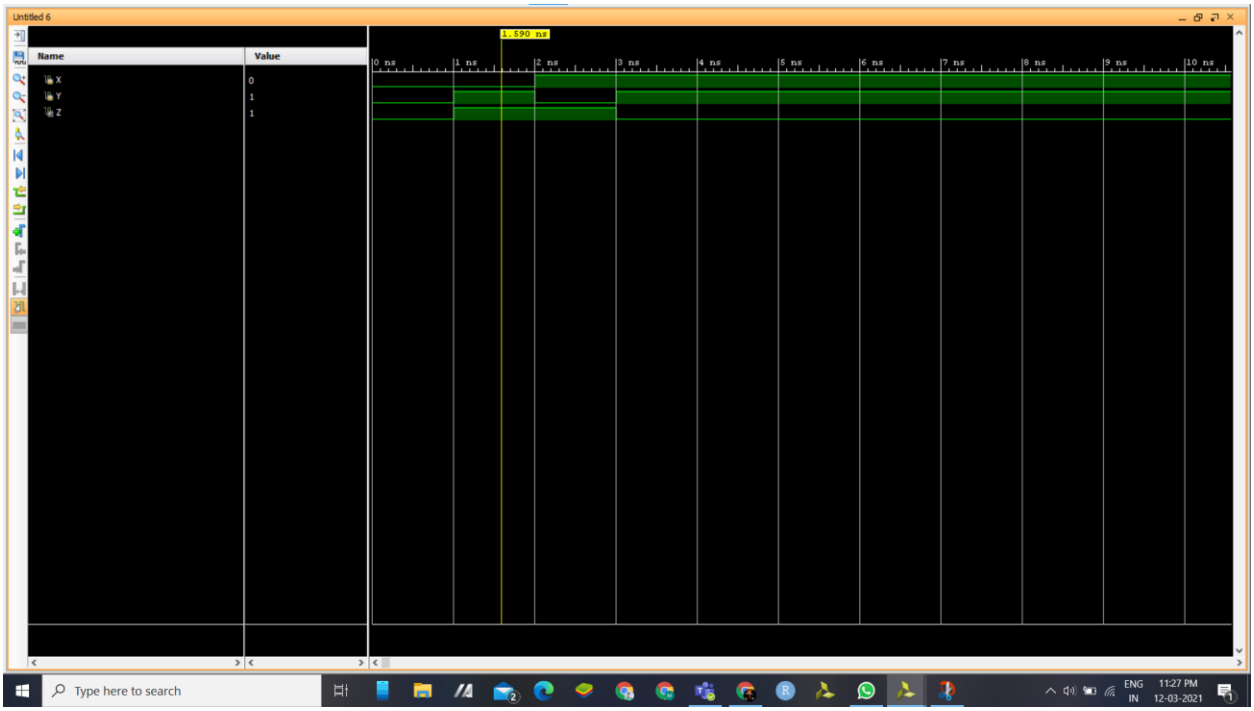
DATE:12-03-2021 NAME:MAJJIGA JASWANTH REGNO:20BCD7171



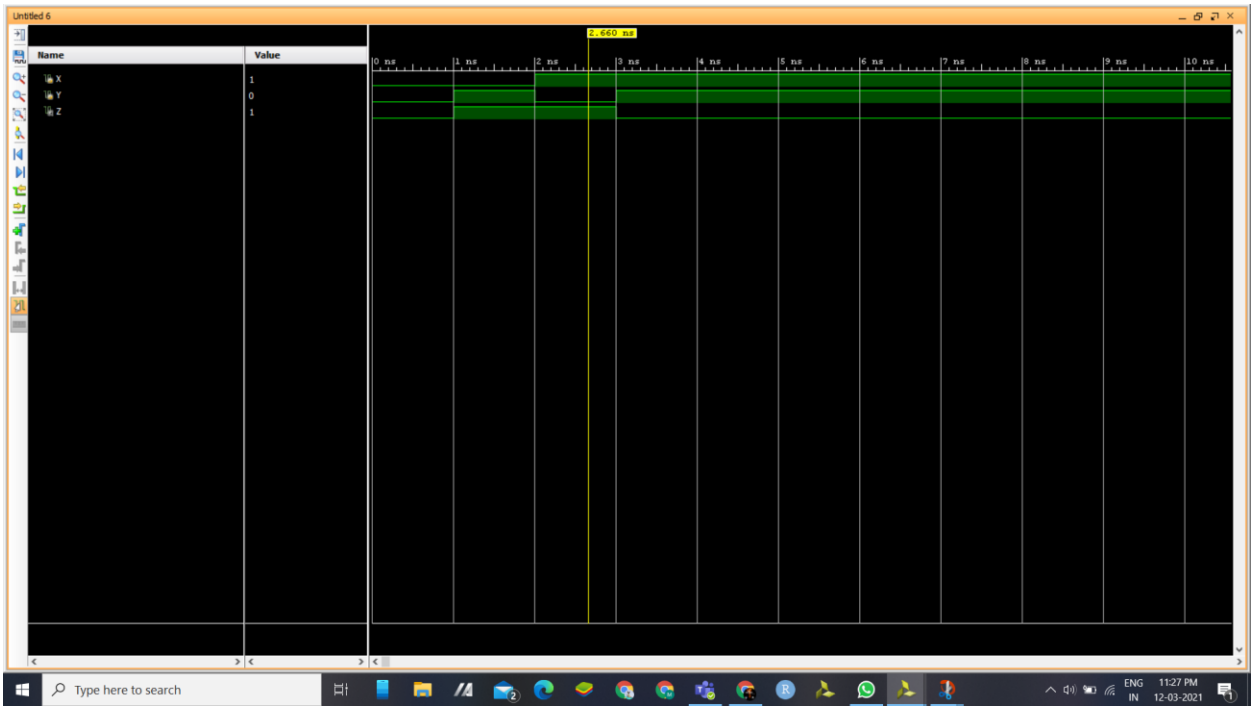


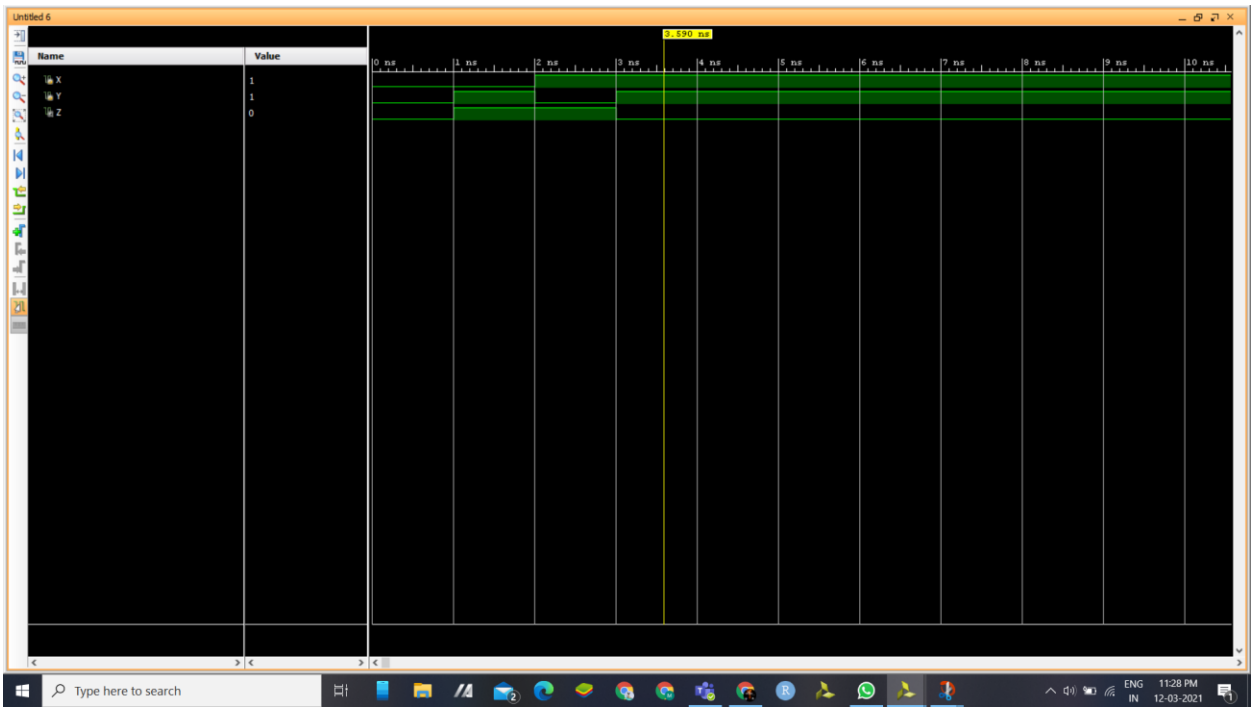
Xor gate:


Untitled 6
0.320 ns
Name
Value
X
Y
Z
0
0
0
0 ns
1 ns
2 ns
3 ns
4 ns
5 ns
6 ns
7 ns
8 ns
9 ns
10 ns
Type here to search
ENG
IN
11:27 PM
12-03-2021

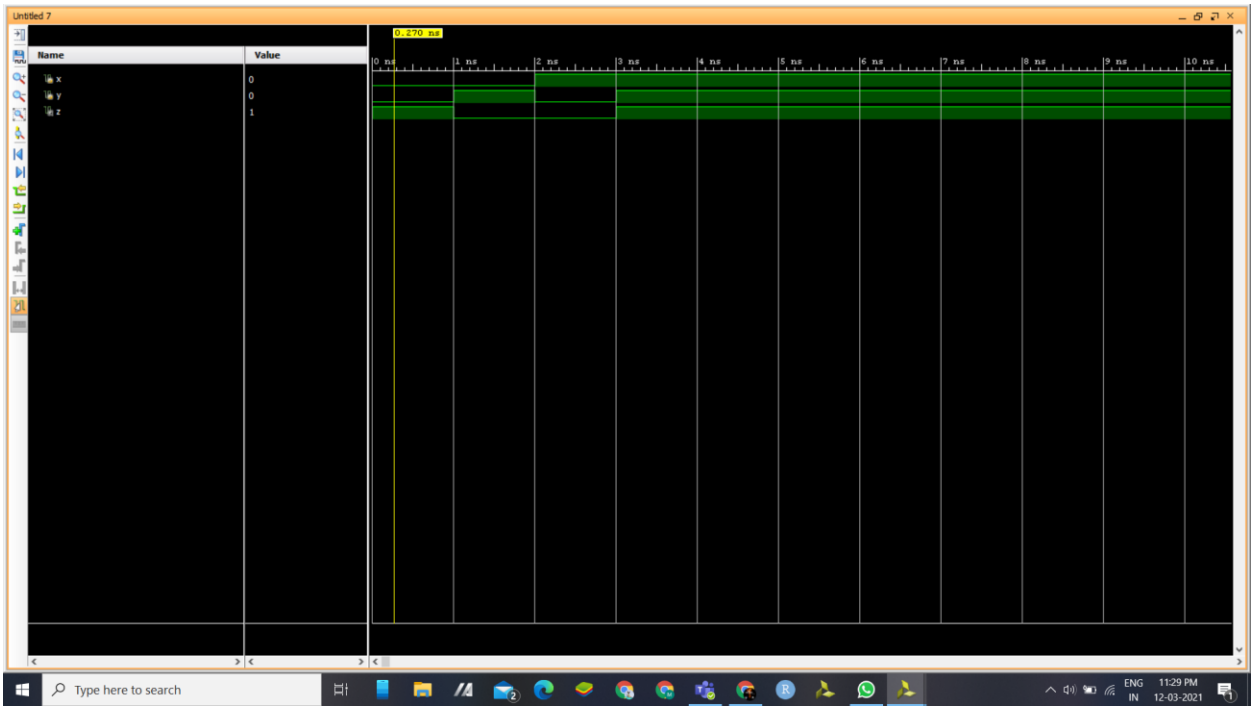

Untitled 6
1.590 ns
Name
Value
X
Y
Z
0
1
1
0 ns
1 ns
2 ns
3 ns
4 ns
5 ns
6 ns
7 ns
8 ns
9 ns
10 ns
Type here to search
ENG
IN
11:27 PM
12-03-2021

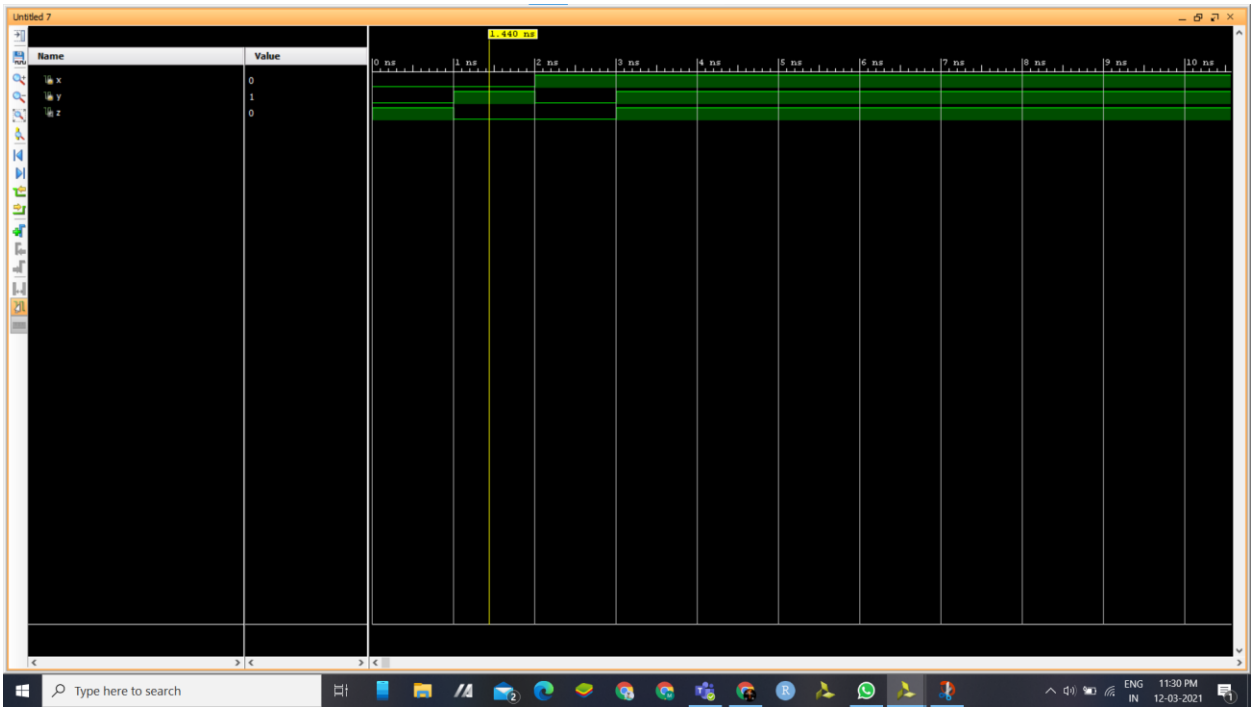
DATE:12-03-2021 NAME:MAJJIGA JASWANTH REGNO:20BCD7171



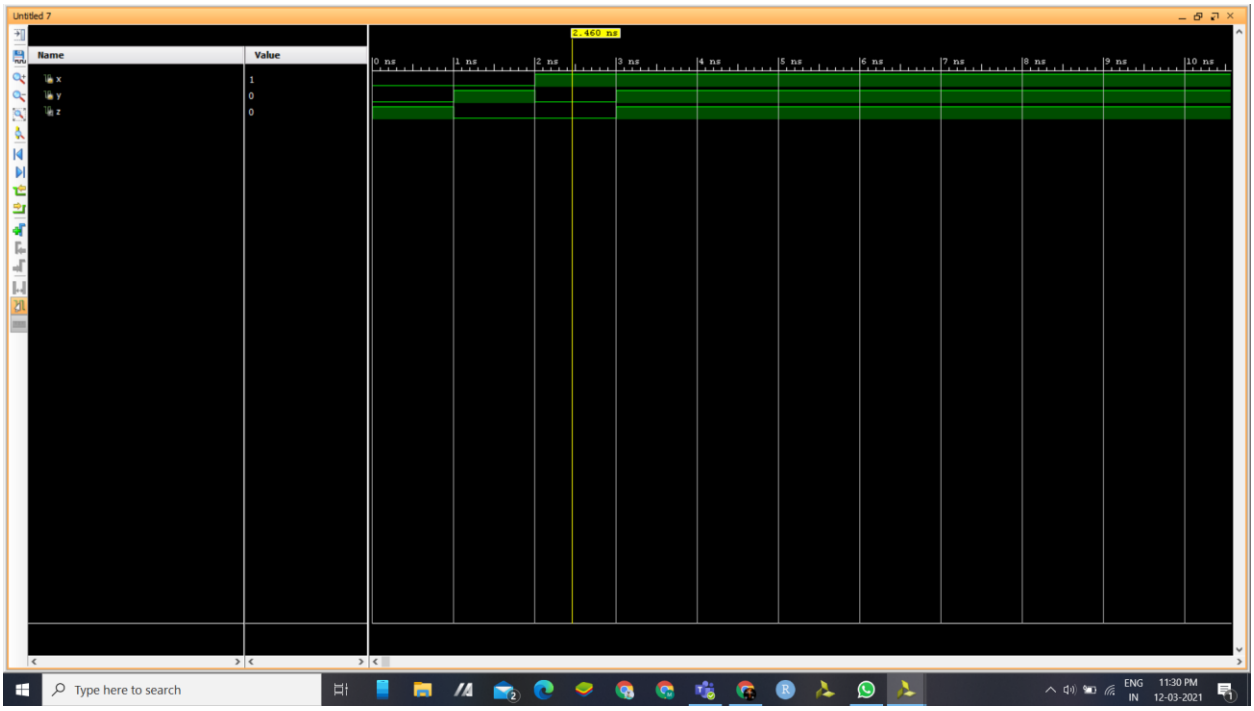


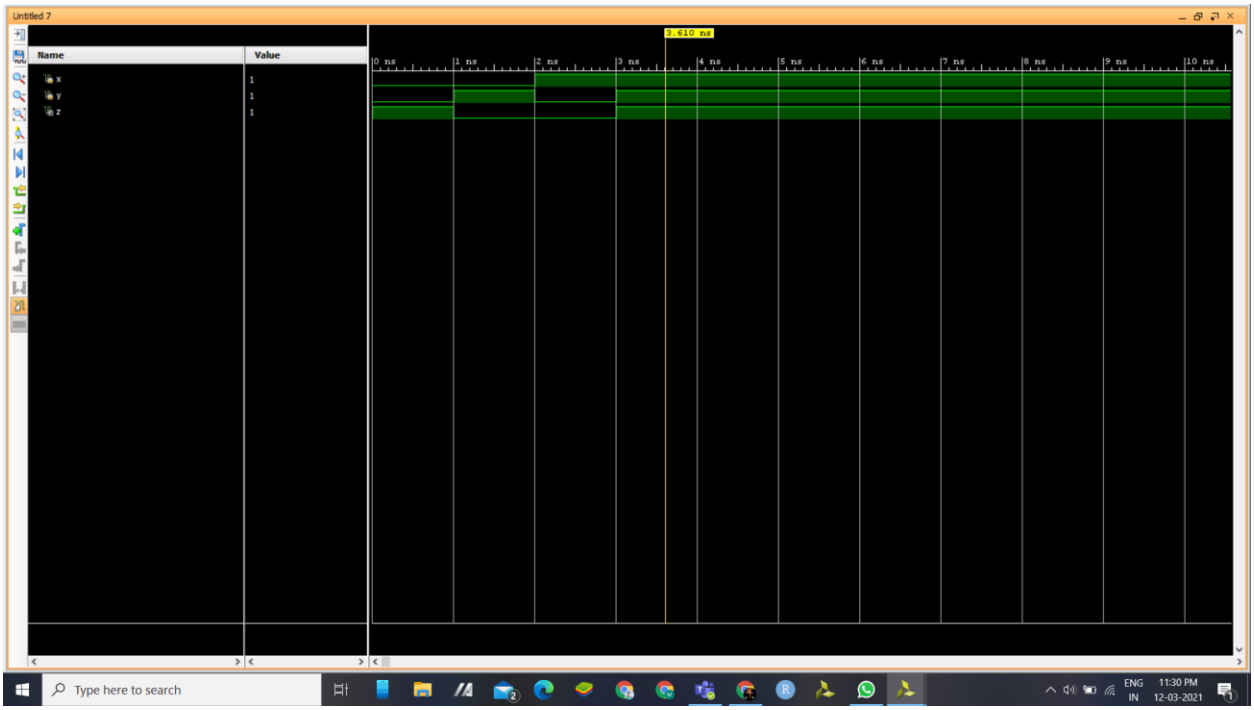
Xnor gate:


Untitled 7
0.270 ns
Name
Value
x
0
y
0
z
1
0 ns
1 ns
2 ns
3 ns
4 ns
5 ns
6 ns
7 ns
8 ns
9 ns
10 ns
Type here to search
ENG
IN
11:29 PM
12-03-2021


Untitled 7
1.440 ns
Name
Value
x
0
y
1
z
0
0 ns
1 ns
2 ns
3 ns
4 ns
5 ns
6 ns
7 ns
8 ns
9 ns
10 ns
Type here to search
ENG
IN
11:30 PM
12-03-2021

DATE:12-03-2021 NAME:MAJJIGA JASWANTH REGNO:20BCD7171





Conclusion:

Now I got full knowledge about the verilog codes of the logic gates and there are three modelling styles also in that in this lab only two I learned(dataflow and structural)

1)dataflow modelling

2)behavioral modelling

3)structural modelling

There are three modelling styles in that two was very well known and I can do any codes for logic gates

Finally in this lab I came to know about

- Truth tables
- Logic gate diagrams
- Boolean expressions for logic gates
- Different types of modelling
- Output analysis

Reference links:

circuitverse