LAB – 3

Name: Sridhar

Roll: CB.EN.U4CSE19063

1. IMPLEMENT ALL THE BASIC GATES USING GATE LEVEL ABSTRACTION

Andgate:

Code:

module gateleveland();

reg a;

reg b;

wire y;

and(y,a,b);

initial

begin

$monitor("a=%b, b=%b,y=%b",a,b,y);

a = 0; b = 0; #10;

a = 0; b = 1; #10;

a = 1; b = 0; #10;

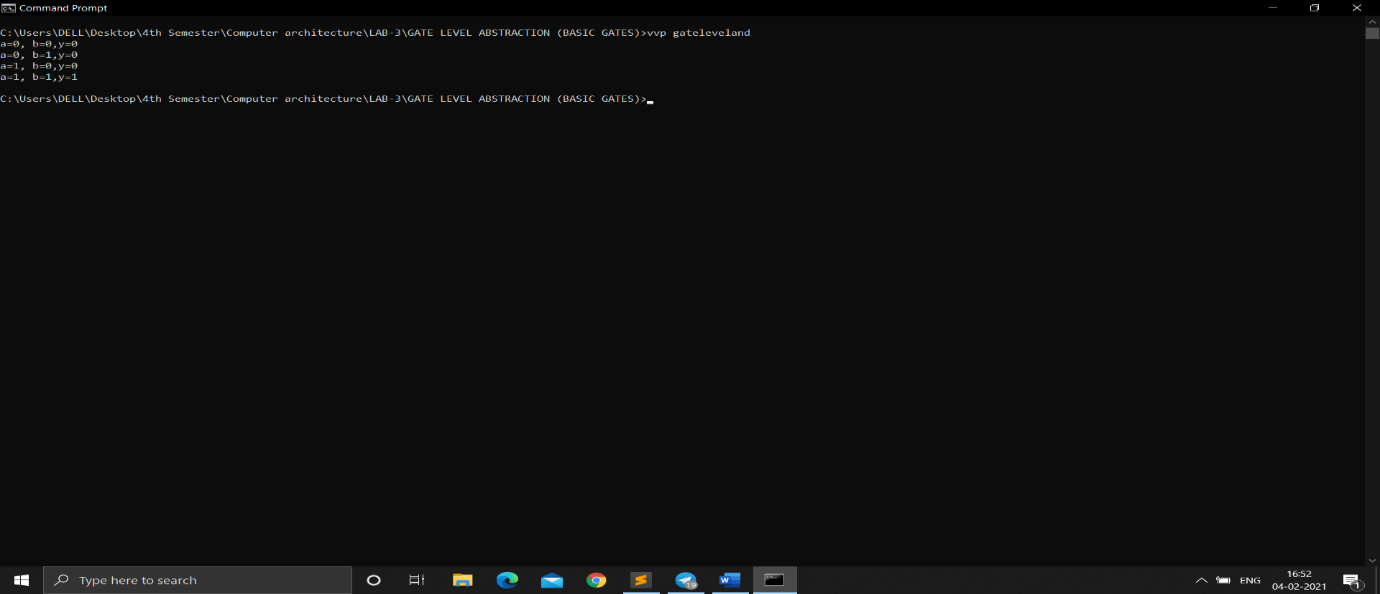
a = 1; b = 1; #10;

$finish;

end

endmodule

screenshot:



orgate:

code:

module gatelevelor();

reg a;

reg b;

wire y;

or(y,a,b);

initial

begin

$monitor("a=%b, b=%b,y=%b",a,b,y);

a = 0; b = 0; #10;

a = 0; b = 1; #10;

a = 1; b = 0; #10;

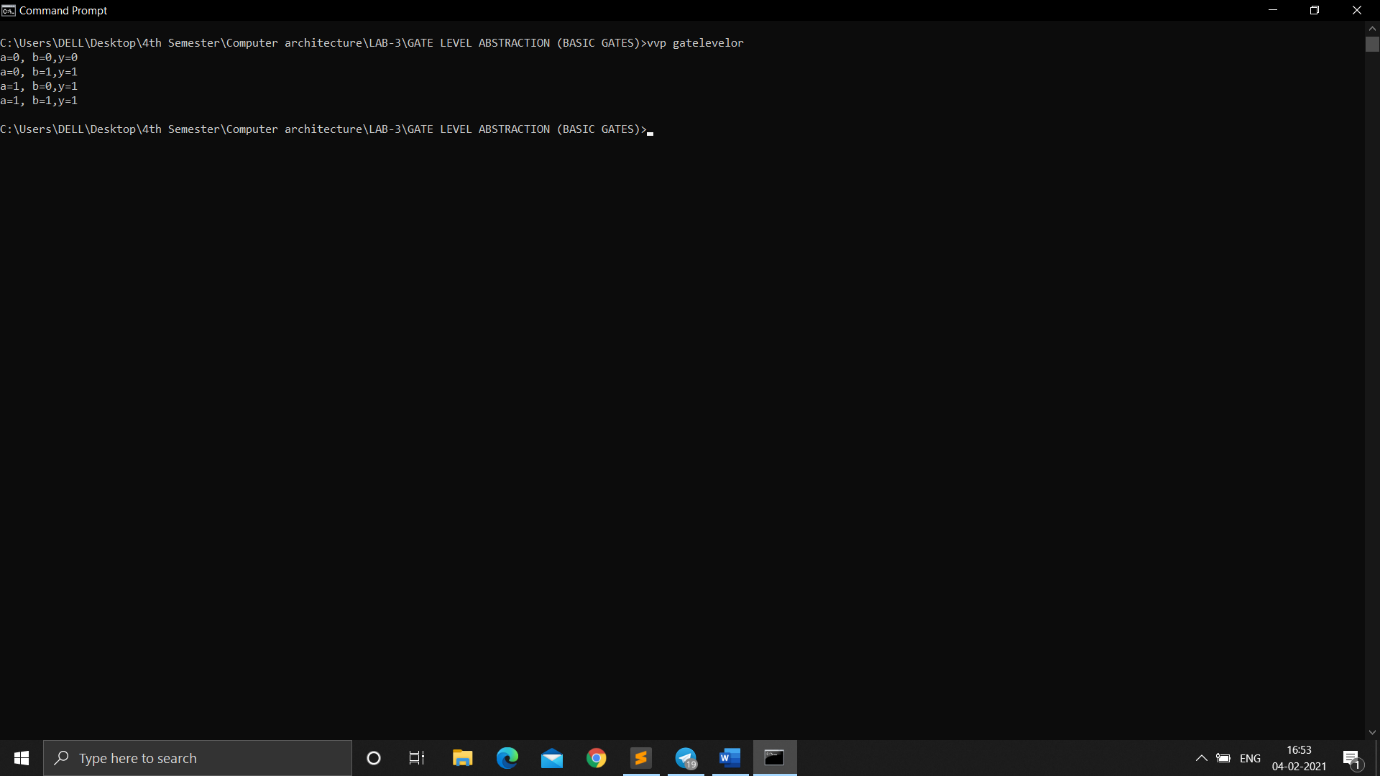
a = 1; b = 1; #10;

$finish;

end

endmodule

screenshot:



Notgate:

code:

module gatelevelnot();

reg a;

reg b;

wire y;

not(y,a);

initial

begin

$monitor("a=%b, y=%b",a,y);

a = 1; #10;

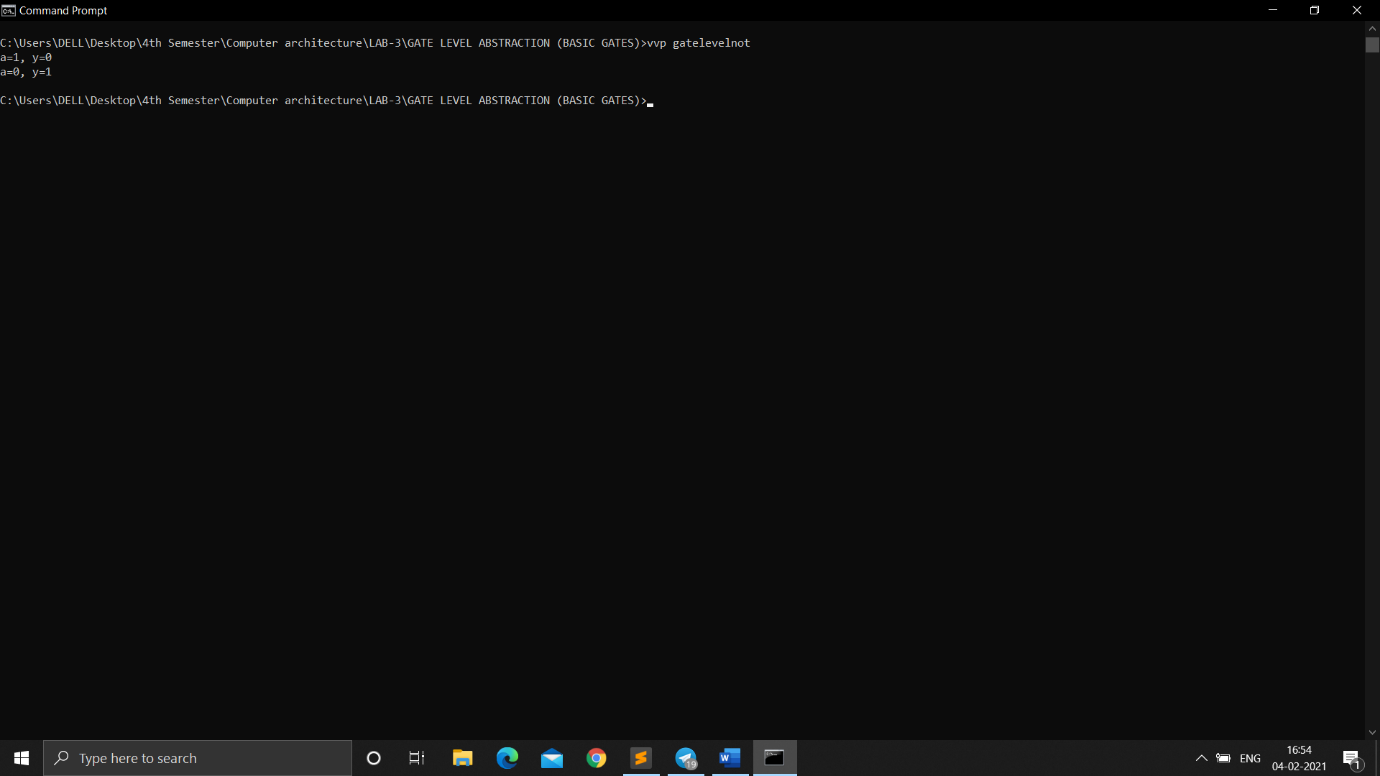
a = 0; #10;

$finish;

end

endmodule

screenshot:



Norgate:

code:

module gatelevelnor();

reg a;

reg b;

wire y;

nor(y,a, b);

initial

begin

$monitor("a=%b, b=%b, y=%b", a, b, y);

a = 0; b = 0; #10;

a = 0; b = 1; #10;

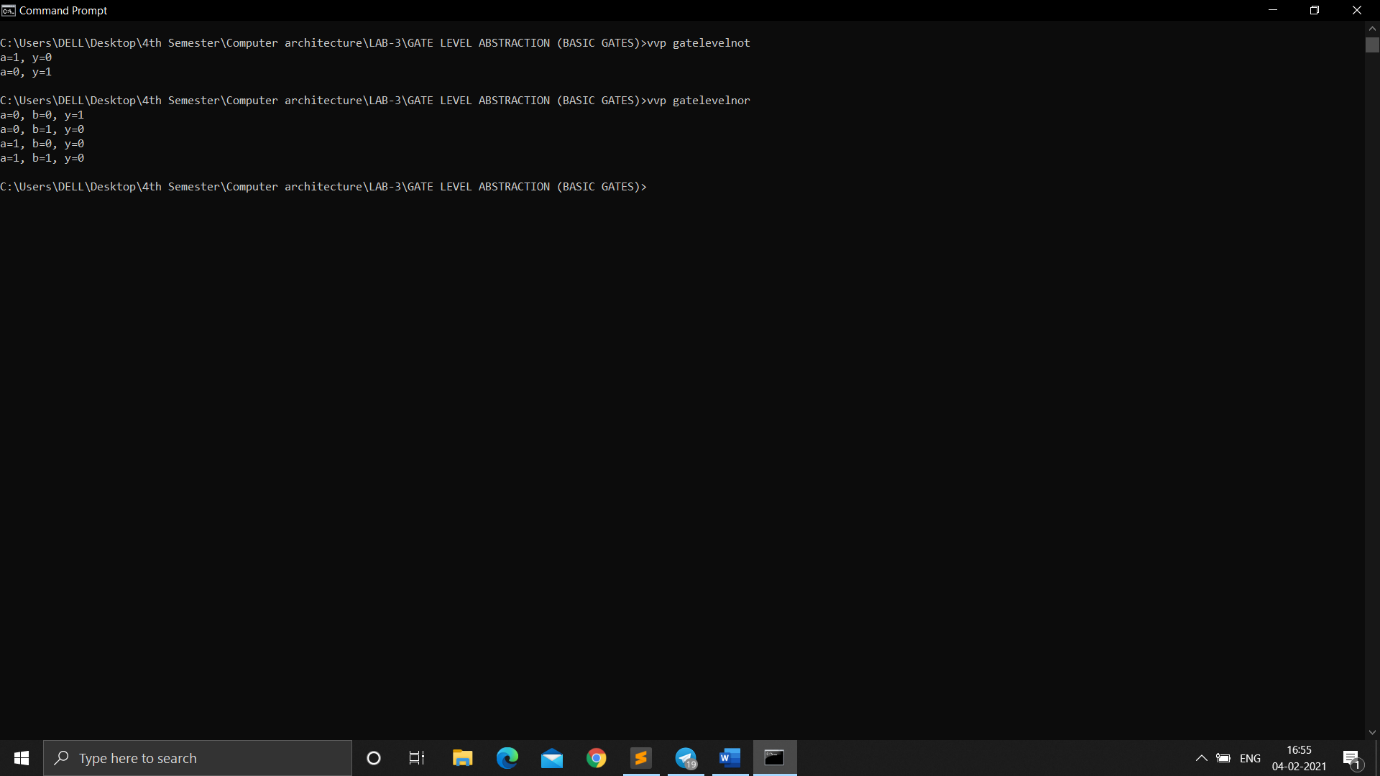
a = 1; b = 0; #10;

a = 1; b = 1; #10;

$finish;

end

endmodule

screenshot:

nand gate:

module gatelevelnand();

reg a;

reg b;

wire y;

nand(y,a, b);

initial

begin

$monitor("a=%b, b=%b, y=%b", a, b, y);

a = 0; b = 0; #10;

a = 0; b = 1; #10;

a = 1; b = 0; #10;

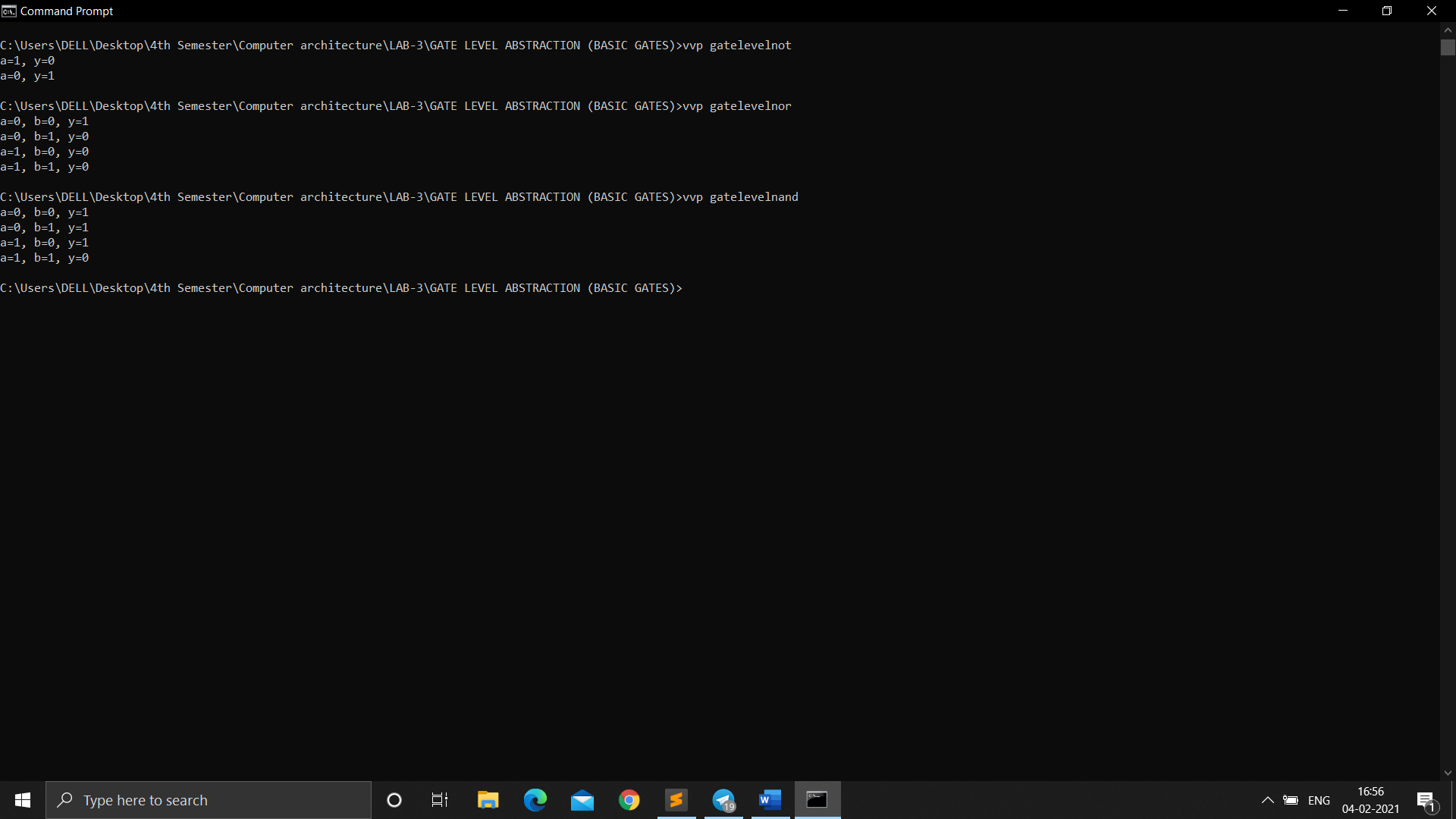
a = 1; b = 1; #10;

$finish;

end

endmodule

screenshot:



xor gate

module gatelevelor();

reg a;

reg b;

wire y;

xor(y,a,b);

initial

begin

$monitor("a=%b, b=%b,y=%b",a,b,y);

a = 0; b = 0; #10;

a = 0; b = 1; #10;

a = 1; b = 0; #10;

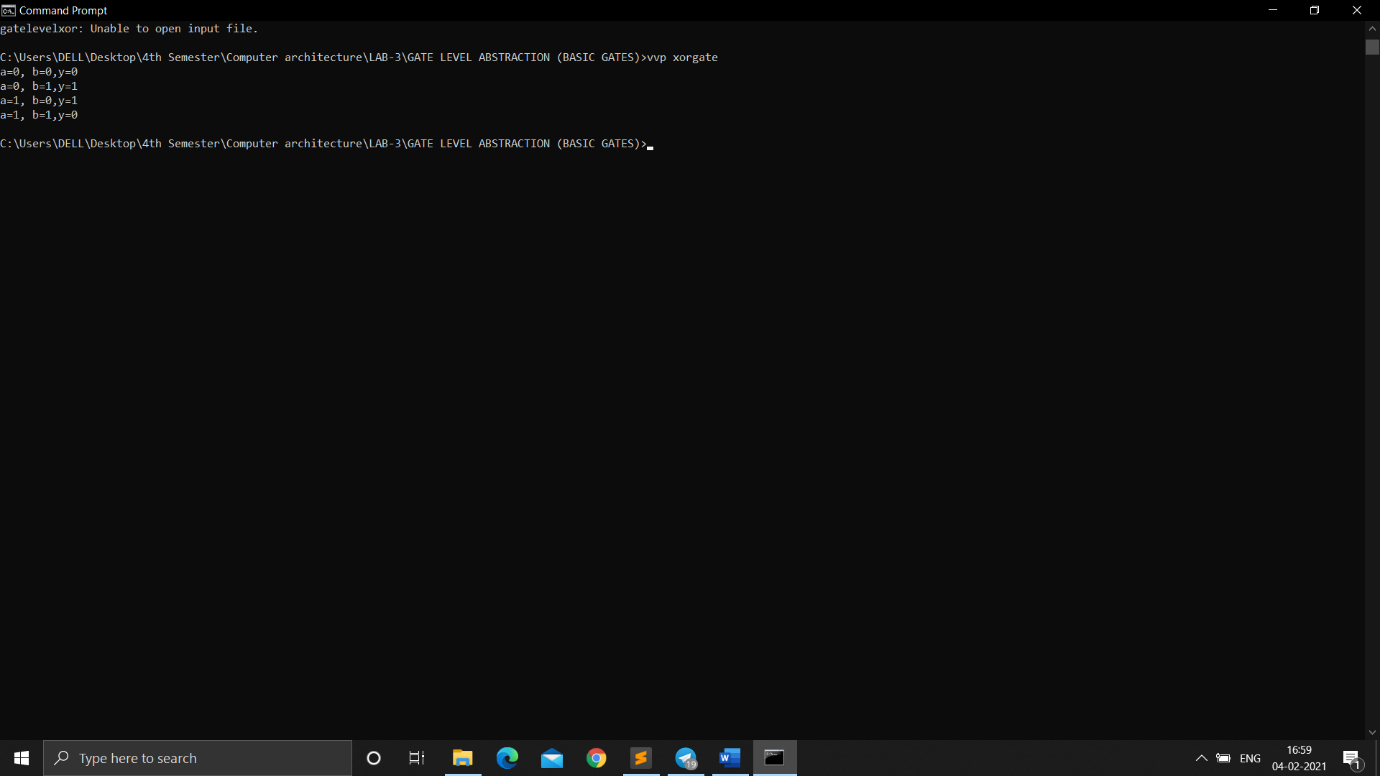
a = 1; b = 1; #10;

$finish;

end

endmodule

screenshot:



Xnor gate:

Code:

module gateleveland();

reg a;

reg b;

wire y;

xnor(y,a,b);

initial

begin

$monitor("a=%b, b=%b,y=%b",a,b,y);

a = 0; b = 0; #10;

a = 0; b = 1; #10;

a = 1; b = 0; #10;

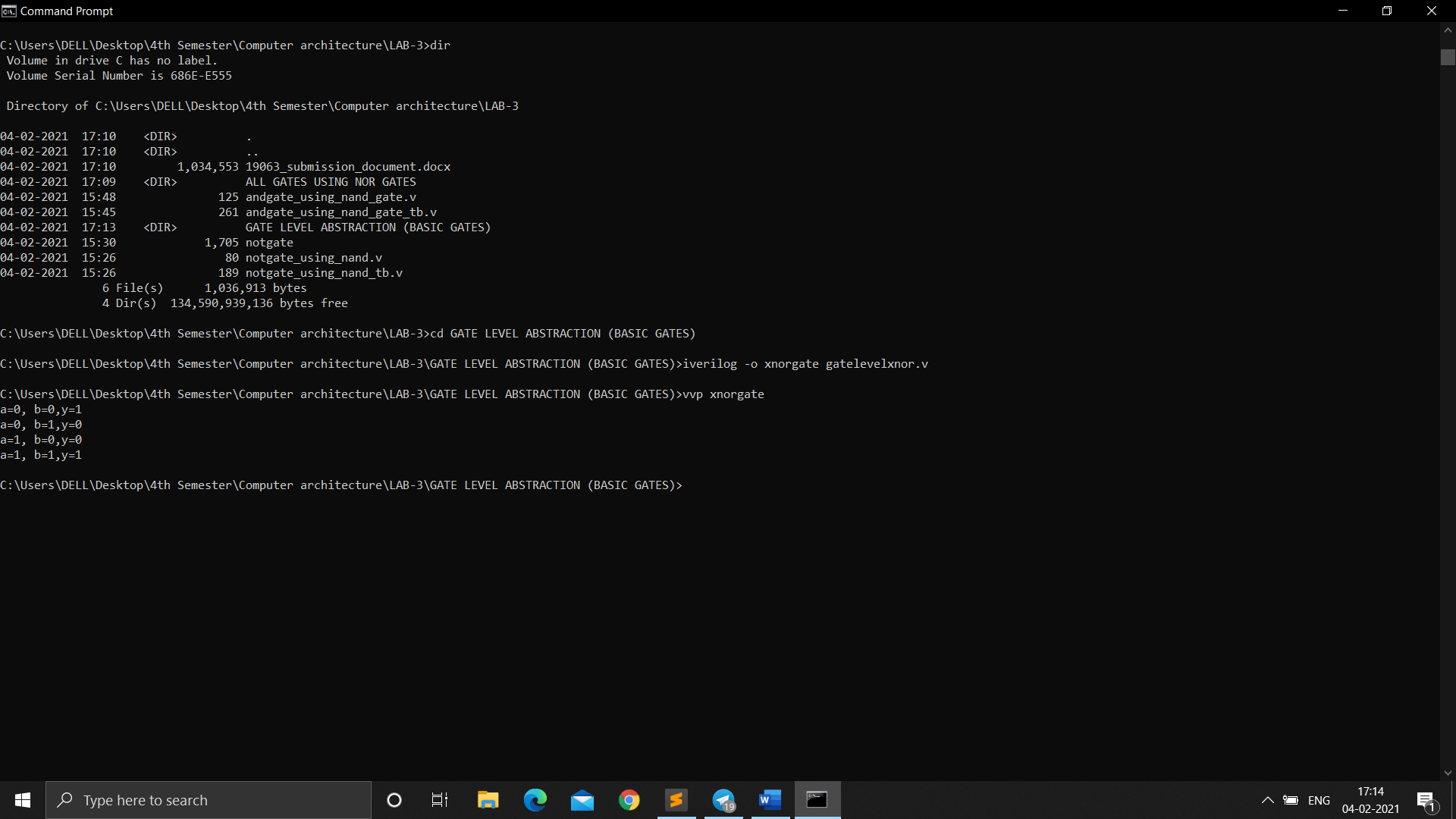
a = 1; b = 1; #10;

$finish;

end

endmodule

screenshot:



3

And gate using nor:

Code:

module gateleveland();

reg a;

reg b;

wire c;

wire d;

wire y;

nor(c,a,a);

nor(d,b,b);

nor(y,c,d);

initial

begin

$monitor("a=%b, b=%b,y=%b",a,b,y);

a = 0; b = 0; #10;

a = 0; b = 1; #10;

a = 1; b = 0; #10;

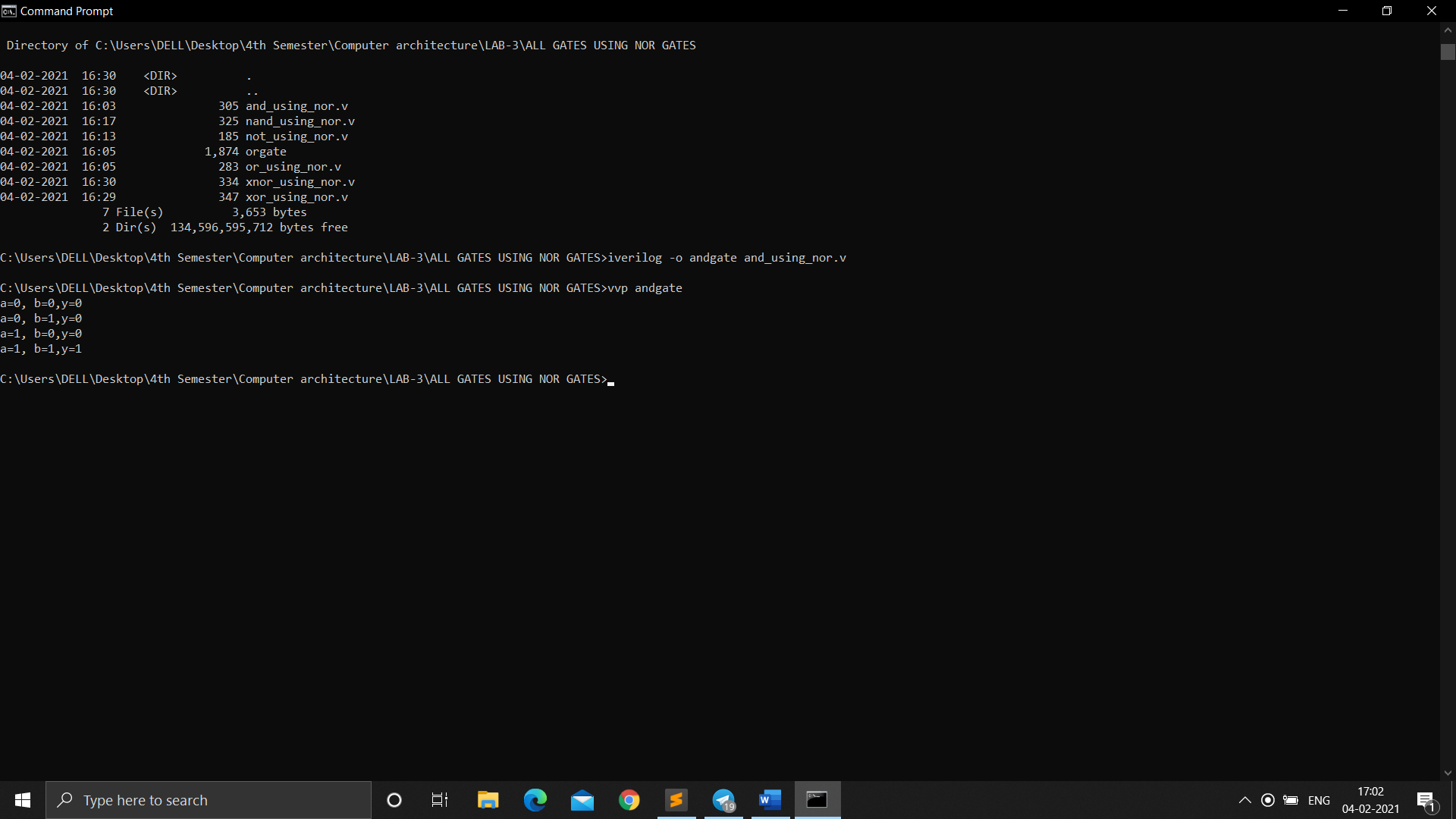
a = 1; b = 1; #10;

$finish;

end

endmodule

screenshot:



or gate using nor:

code:

module gateleveland();

reg a;

reg b;

wire c;

wire y;

nor(c,a,b);

nor(y,c,c);

initial

begin

$monitor("a=%b, b=%b,y=%b",a,b,y);

a = 0; b = 0; #10;

a = 0; b = 1; #10;

a = 1; b = 0; #10;

a = 1; b = 1; #10;

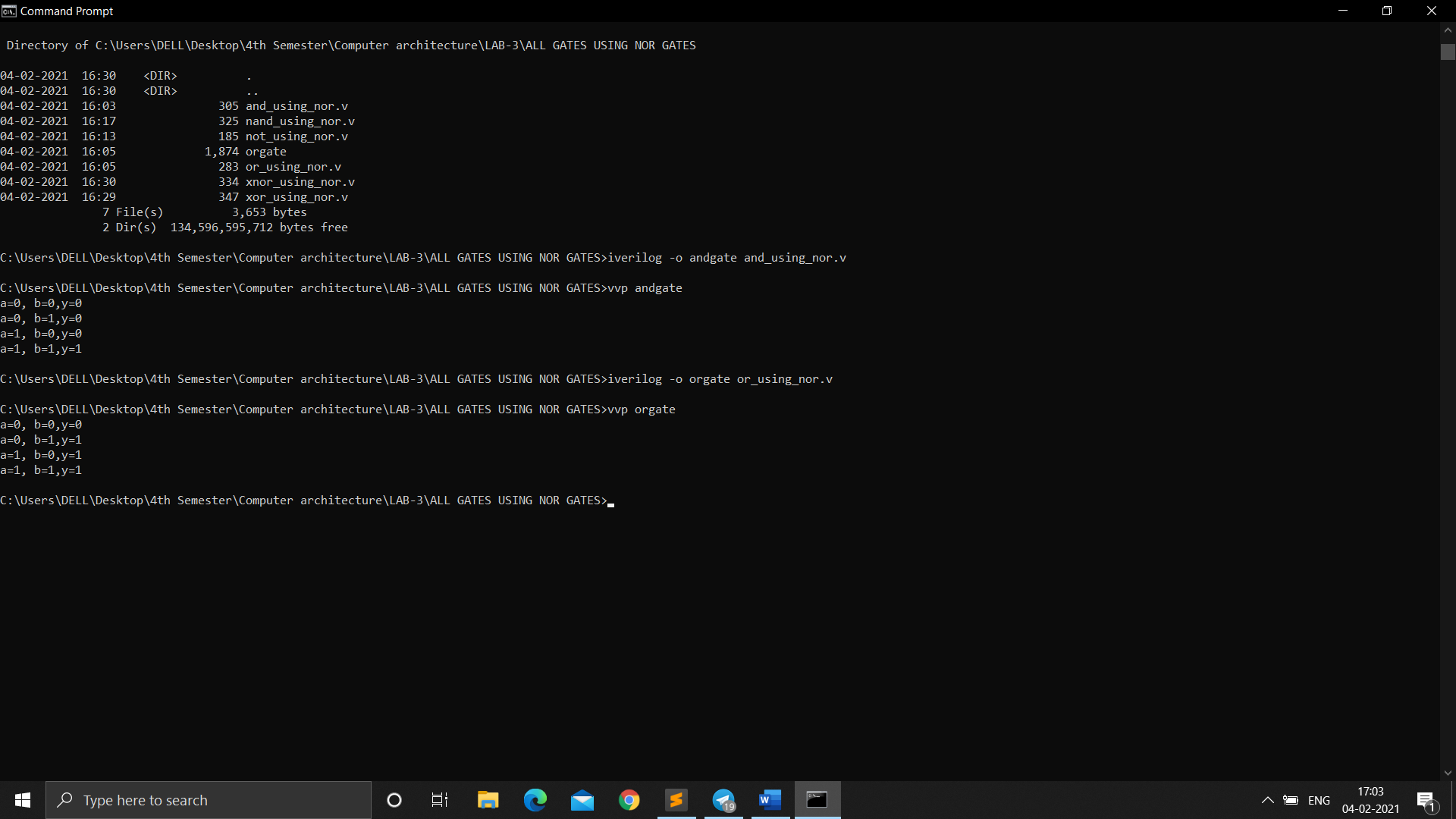
$finish;

end

endmodule

screenshot:

next page



not gate using nor:

code:

module gateleveland();

reg a;

wire y;

nor(y,a,a);

initial

begin

$monitor("a=%b, b=%b,y=%b",a,b,y);

a = 0; #10;

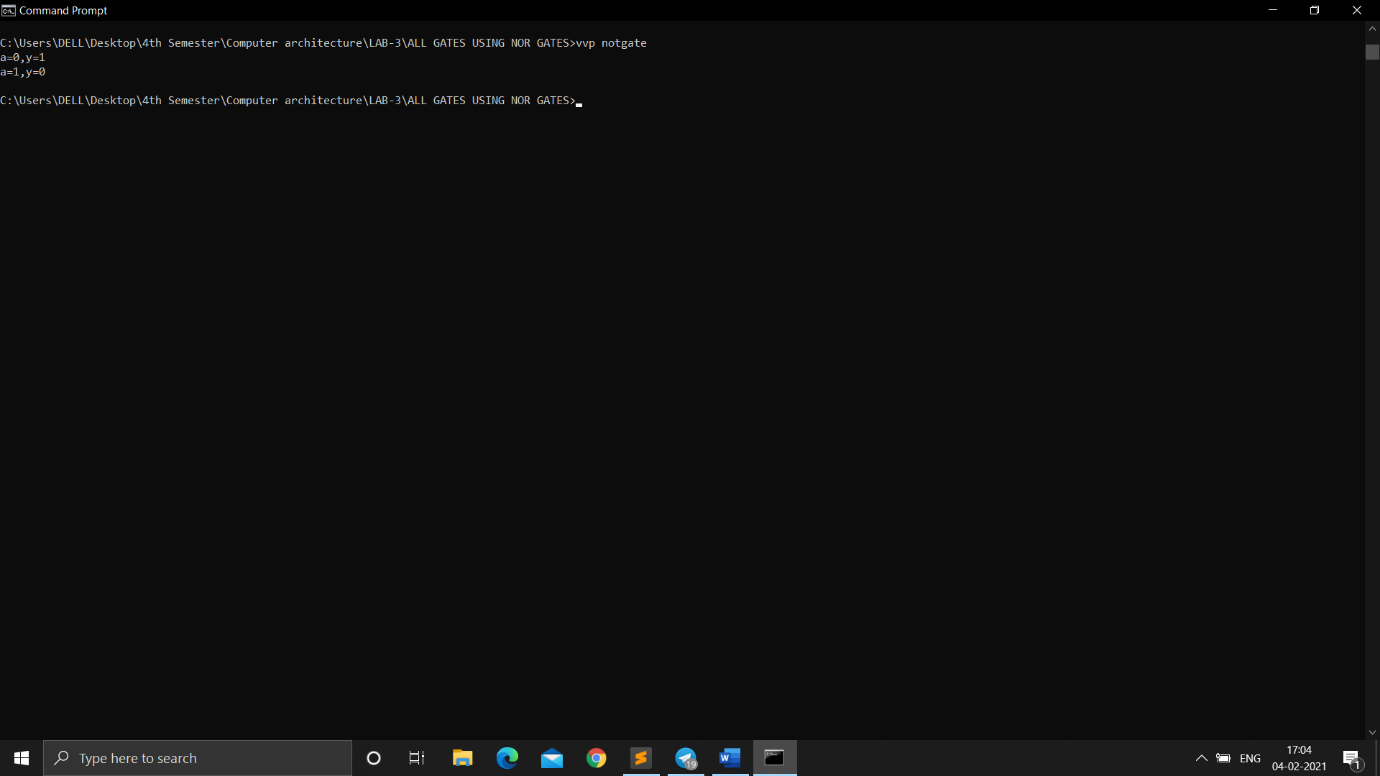
a = 1; #10;

$finish;

end

endmodule

screenshot:



nand using nor:

code:

module gateleveland();

reg a;

reg b;

wire c;

wire d;

wire e;

wire y;

nor(c,a,a);

nor(d,b,b);

nor(e,c,d);

nor(y,e,e);

initial

begin

$monitor("a=%b, b=%b,y=%b",a,b,y);

a = 0; b = 0; #10;

a = 0; b = 1; #10;

a = 1; b = 0; #10;

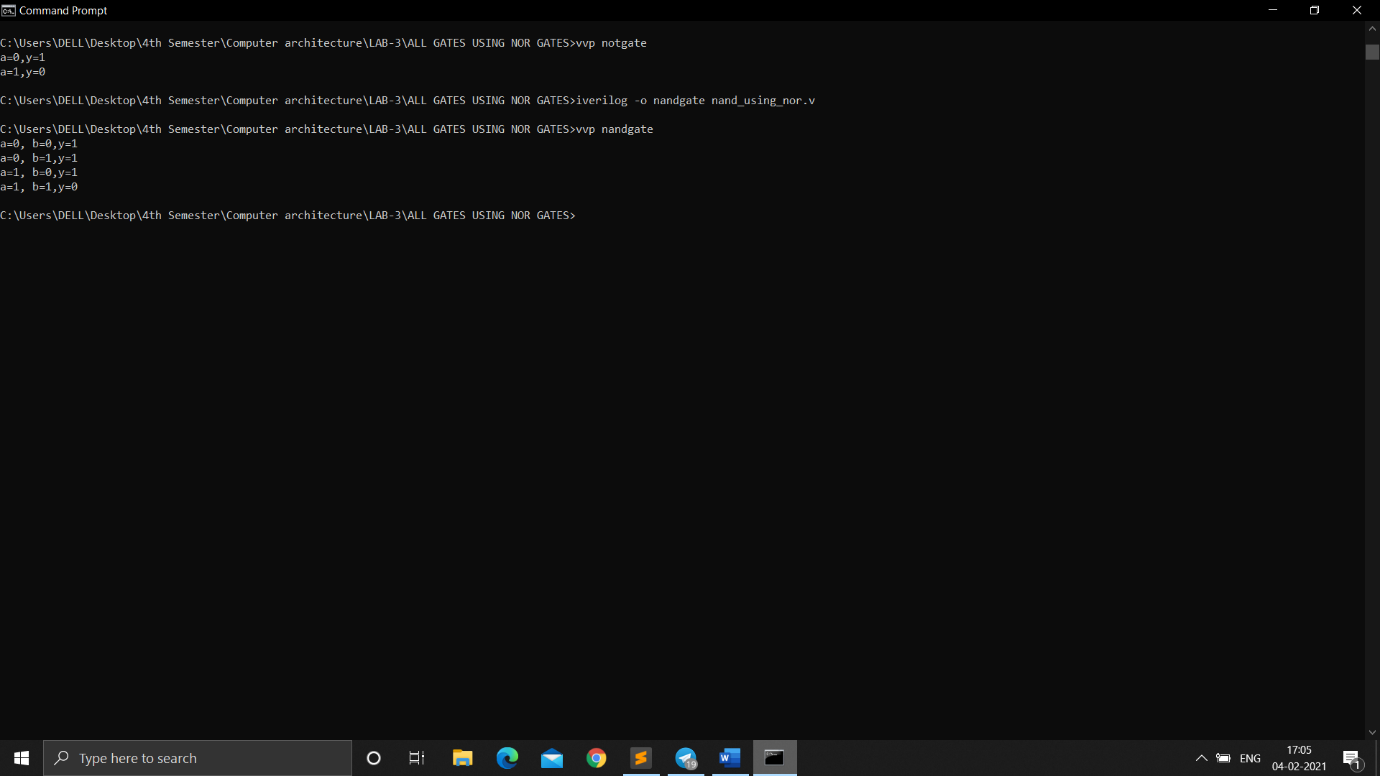
a = 1; b = 1; #10;

$finish;

end

endmodule

screenshot:



XOR GATE USING NOR:

module gateleveland();

reg a;

reg b;

wire c;

wire d;

wire e;

wire f;

wire y;

nor(c,a,b);

nor(d,c,a);

nor(e,c,b);

nor(f,e,d);

nor(y,f,f);

initial

begin

$monitor("a=%b, b=%b,y=%b",a,b,y);

a = 0; b = 0; #10;

a = 0; b = 1; #10;

a = 1; b = 0; #10;

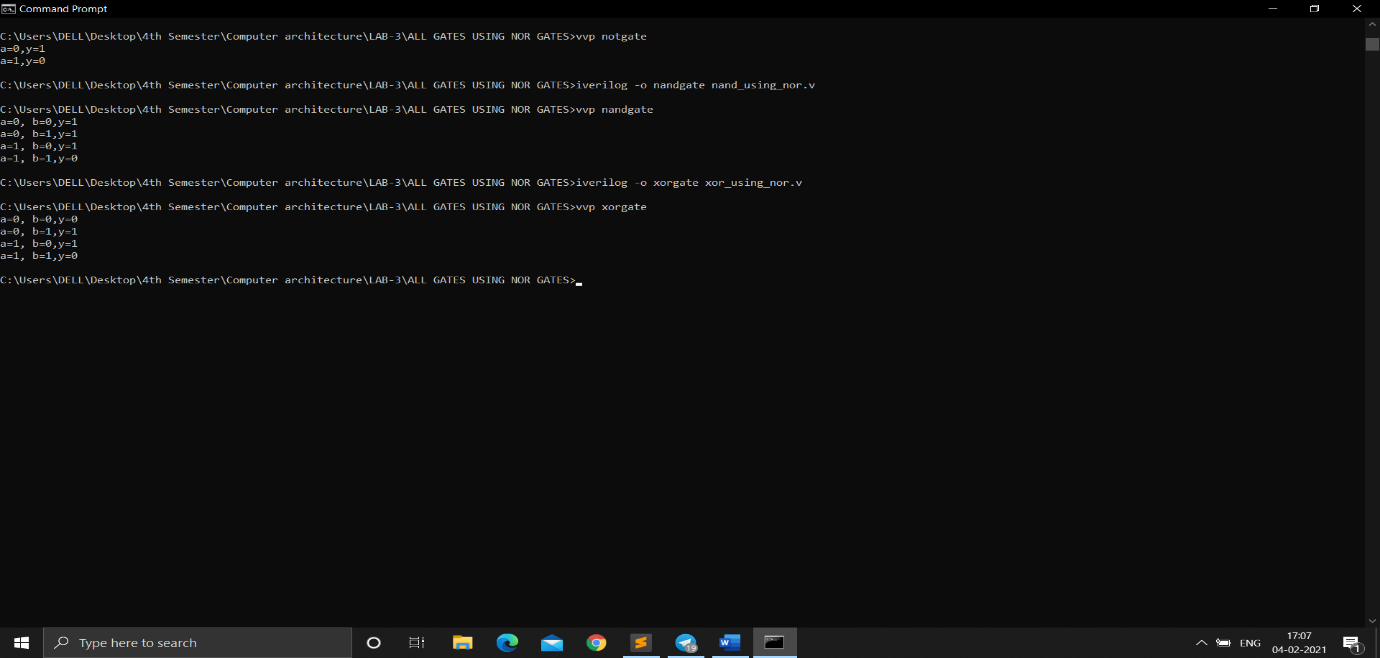
a = 1; b = 1; #10;

$finish;

end

endmodule

screenshot:



XNOR GATE USING NOR:

Code:

module gateleveland();

reg a;

reg b;

wire c;

wire d;

wire e;

wire f;

wire y;

nor(c,a,b);

nor(d,c,a);

nor(e,c,b);

nor(y,e,d);

initial

begin

$monitor("a=%b, b=%b,y=%b",a,b,y);

a = 0; b = 0; #10;

a = 0; b = 1; #10;

a = 1; b = 0; #10;

a = 1; b = 1; #10;

$finish;

end

endmodule

screenshot:

