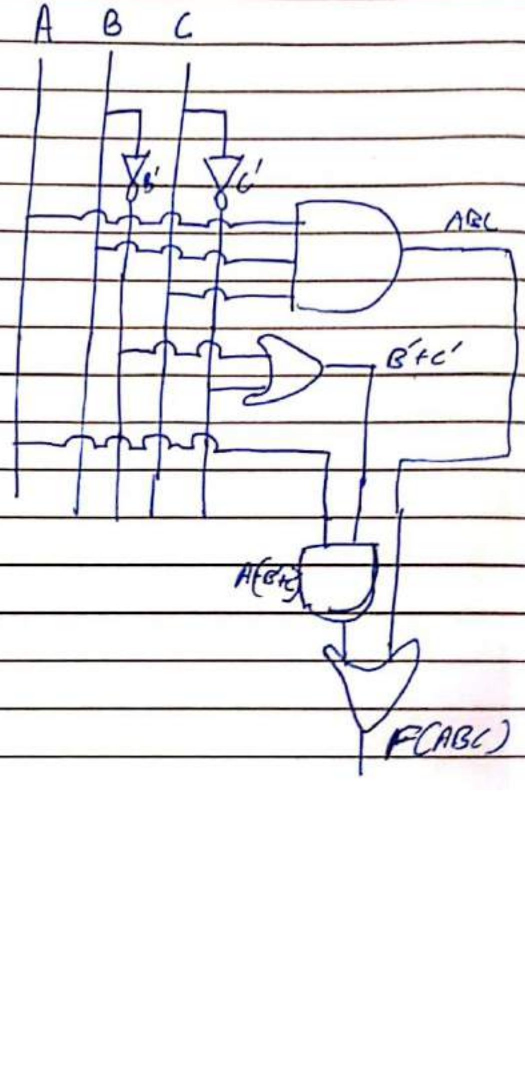
Lab Assignment 3

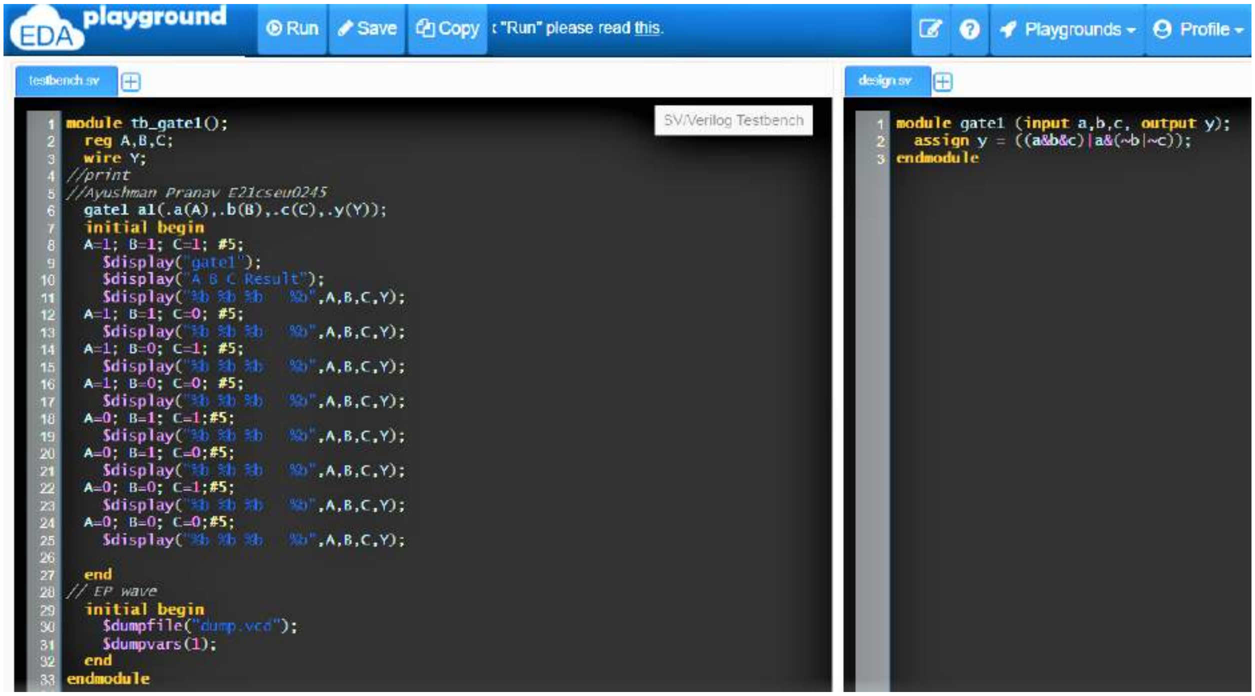
1. Perform the following operation on a given Boolean expression:
   F(ABC) = ABC + A (B’ + C’)
   a. Write a truth table for each Boolean expression.
   b. Draw a schematic diagram for each Boolean expression.
   c. Write a Verilog code for each Boolean expression and then test using wave form and compare with truth table whether your circuit produced same output or not?

$$F(ABC) = \underset{\substack{\downarrow \\ \text{intersection}}}{ABC} + A(\underset{\text{or}}{B'+C'})$$

A	B	C	ABC	B'	C'	B'+C'	A(B'+C')
1	1	1	1	0	0	0	0
1	1	0	0	0	1	1	1
1	0	1	0	1	0	1	1
1	0	0	0	1	1	1	1
0	1	1	0	0	0	0	0
0	1	0	0	0	1	1	0
0	0	1	0	1	0	1	0
0	0	0	0	1	1	1	0

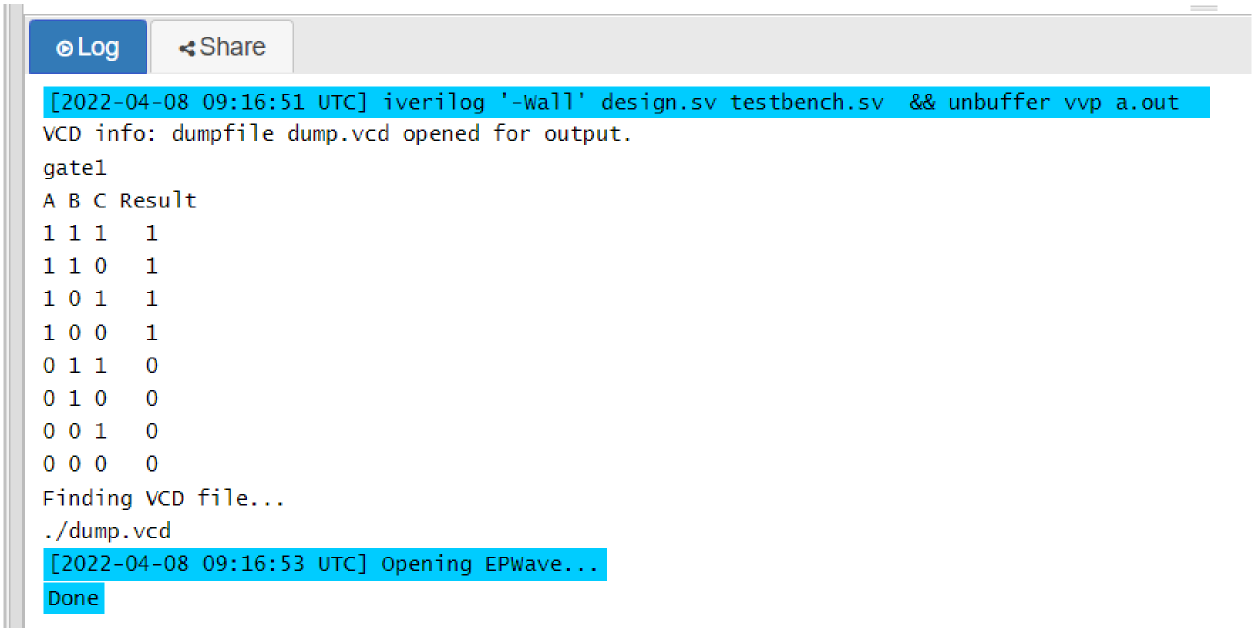
F(ABC)
1
1
1
1
0
0
0
0



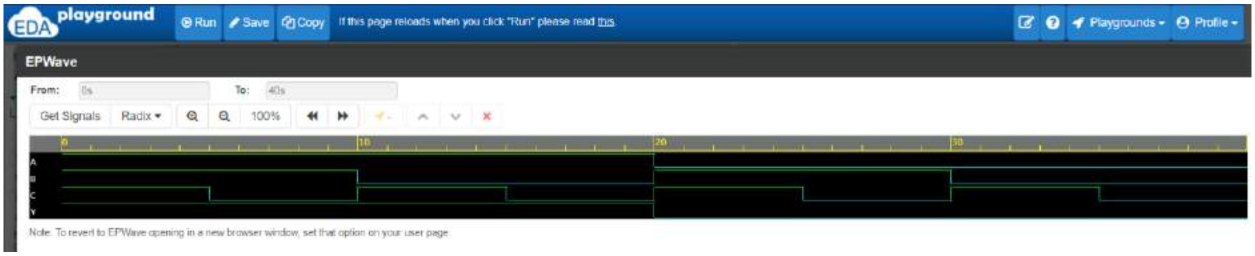


```
module tb_gate1();
  reg A,B,C;
  wire Y;
//print
//Ayushman Pranav E21cseu0245
  gatel al(.a(A),.b(B),.c(C),.y(Y));
  initial begin
  A=1; B=1; C=1; #5;
    $display("gatel");
    $display("A B C Result");
    $display("%b %b %b   %b",A,B,C,Y);
  A=1; B=1; C=0; #5;
    $display("%b %b %b   %b",A,B,C,Y);
  A=1; B=0; C=1; #5;
    $display("%b %b %b   %b",A,B,C,Y);
  A=1; B=0; C=0; #5;
    $display("%b %b %b   %b",A,B,C,Y);
  A=0; B=1; C=1;#5;
    $display("%b %b %b   %b",A,B,C,Y);
  A=0; B=1; C=0;#5;
    $display("%b %b %b   %b",A,B,C,Y);
  A=0; B=0; C=1;#5;
    $display("%b %b %b   %b",A,B,C,Y);
  A=0; B=0; C=0;#5;
    $display("%b %b %b   %b",A,B,C,Y);

  end
// EP wave
  initial begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
  end
endmodule
```

```
module gate1 (input a,b,c, output y);
  assign y = ((a&b&c)|a&(~b|~c));
endmodule
```



```
[2022-04-08 09:16:51 UTC] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
gate1
A B C Result
1 1 1   1
1 1 0   1
1 0 1   1
1 0 0   1
0 1 1   0
0 1 0   0
0 0 1   0
0 0 0   0
Finding VCD file...
./dump.vcd
[2022-04-08 09:16:53 UTC] Opening EPWave...
Done
```



Note: To revert to EPWave opening in a new browser window, set that option on your user page.

Design bench.sv

```
module gate1 (input a,b,c, output y);
  assign y = ((a&b&c)|a&(~b|~c));
endmodule
```

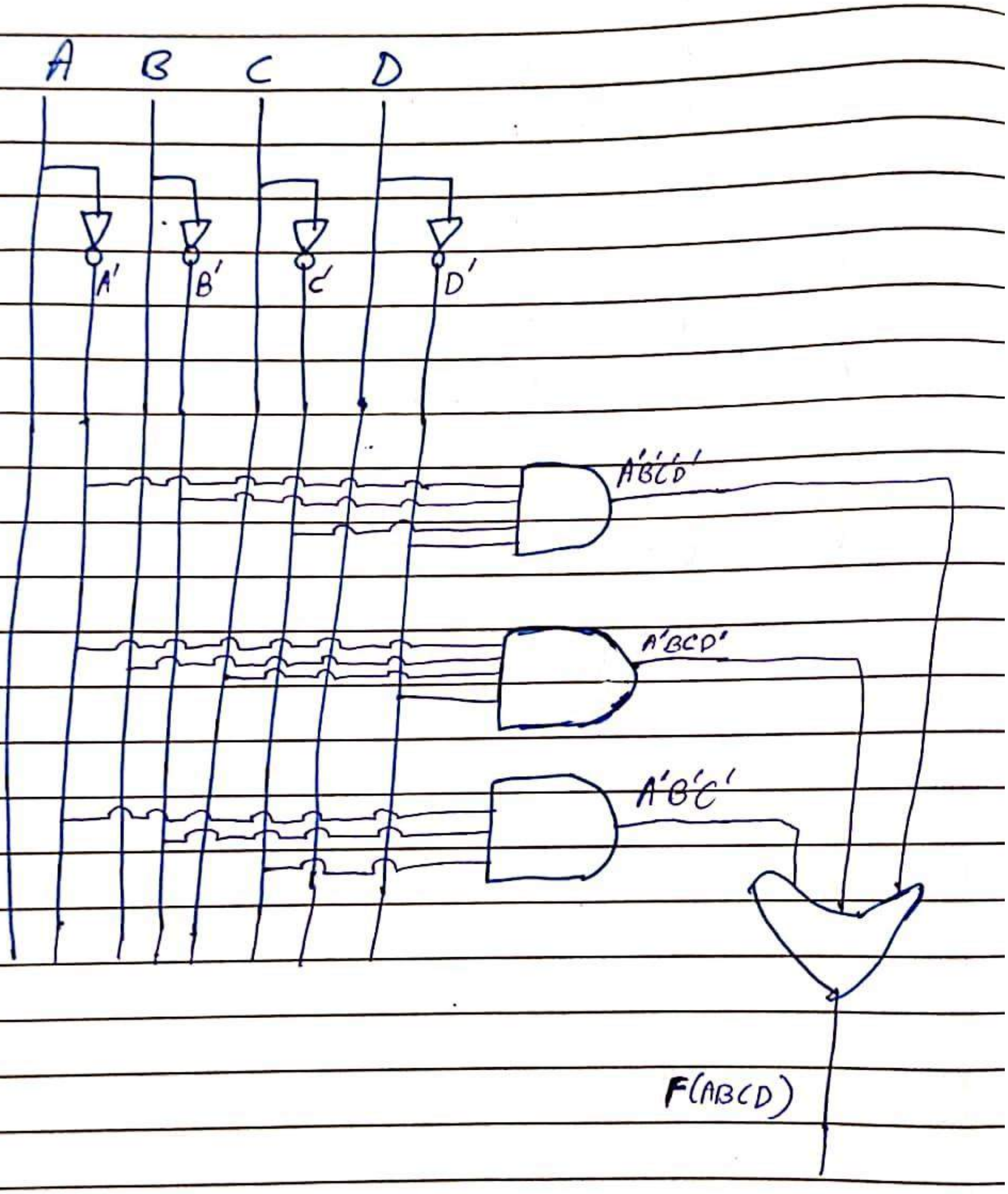
Test bench.sv

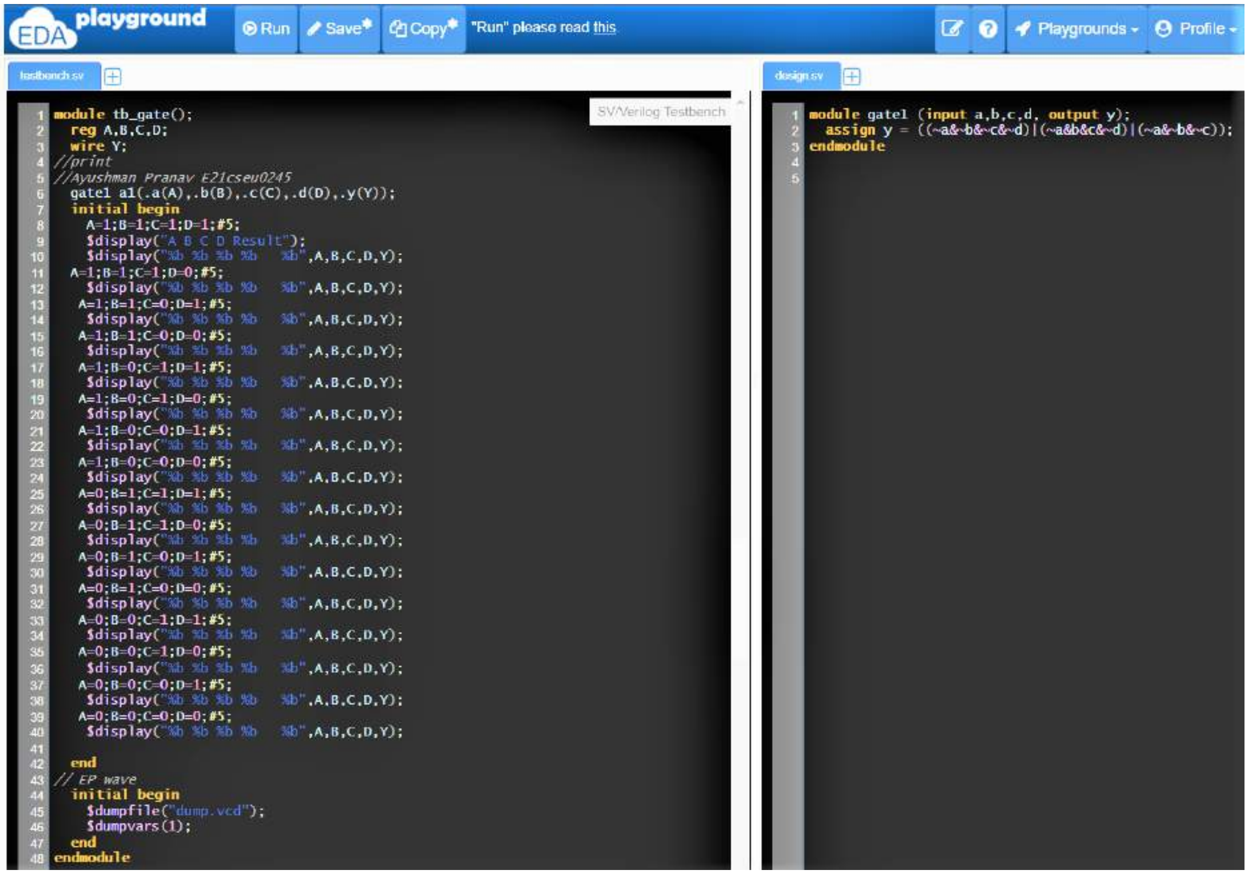
```
module tb_gate1();
  reg A,B,C;
  wire Y;
//print
//Ayushman Pranav E21cseu0245
  gate1 a1(.a(A),.b(B),.c(C),.y(Y));
  initial begin
  A=1; B=1; C=1; #5;
    $display("gate1");
    $display("A B C Result");
    $display("%b %b %b   %b",A,B,C,Y);
  A=1; B=1; C=0; #5;
    $display("%b %b %b   %b",A,B,C,Y);
  A=1; B=0; C=1; #5;
    $display("%b %b %b   %b",A,B,C,Y);
  A=1; B=0; C=0; #5;
    $display("%b %b %b   %b",A,B,C,Y);
  A=0; B=1; C=1;#5;
    $display("%b %b %b   %b",A,B,C,Y);
  A=0; B=1; C=0;#5;
    $display("%b %b %b   %b",A,B,C,Y);
  A=0; B=0; C=1;#5;
    $display("%b %b %b   %b",A,B,C,Y);
  A=0; B=0; C=0;#5;
    $display("%b %b %b   %b",A,B,C,Y);


  end
// EP wave
  initial begin
    $dumpfile("dump.vcd");
            $dumpvars(1);
  end
endmodule
```

2. Perform the following operation on a given Boolean expression:
   F(ABCD) = A’B’C’D’ + A’BCD’ + A’B’C’
   a. Write a truth table for each Boolean expression.
   b. Draw a schematic diagram for each Boolean expression.
   c. Write a Verilog code for each Boolean expression and then test using wave form and compare with truth table whether your circuit produced same output or not?

$F(ABCD) = A'B'C'D' + A'BCD' + A'B'C'$

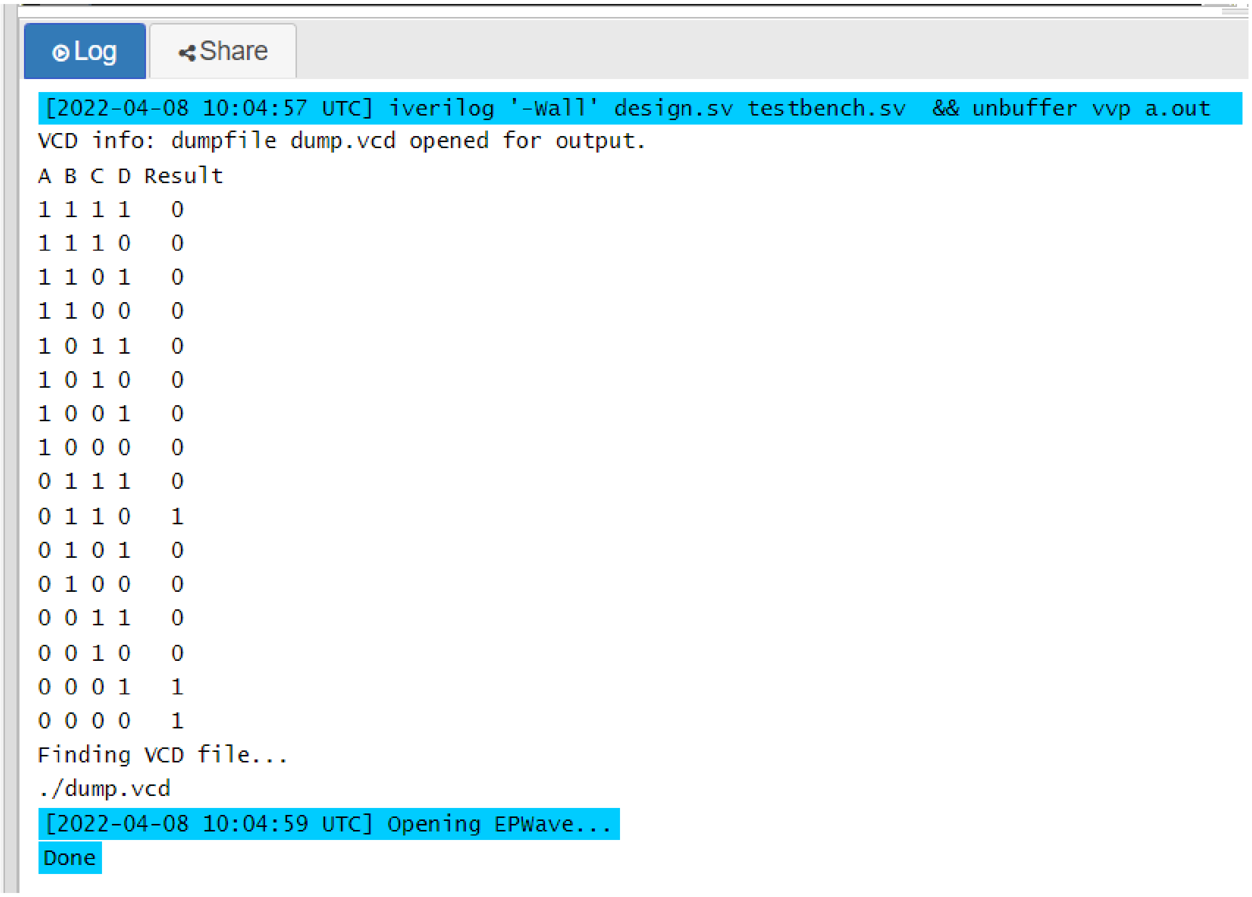
								&	&	&	or
A	B	C	D	A'	B'	C'	D'	A'B'C'D'	A'BCD'	A'B'C'	F(ABCD)
1	1	1	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	0	0	1	1	0	0	0	0
1	0	1	1	0	1	0	0	0	0	0	0
1	0	1	0	0	1	0	1	0	0	0	0
1	0	0	1	0	1	1	0	0	0	0	0
1	0	0	0	0	1	1	1	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0
0	1	1	0	1	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0	0	0	0	0
0	1	0	0	1	0	1	1	0	0	0	0
0	0	1	1	1	1	0	0	0	0	0	0
0	0	1	0	1	1	0	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0	1	1
0	0	0	0	1	1	1	1	1	0	1	1


A
B
C
D
A'
B'
C'
D'
A'B'C'D'
A'BCD'
A'B'C'
F(ABCD)

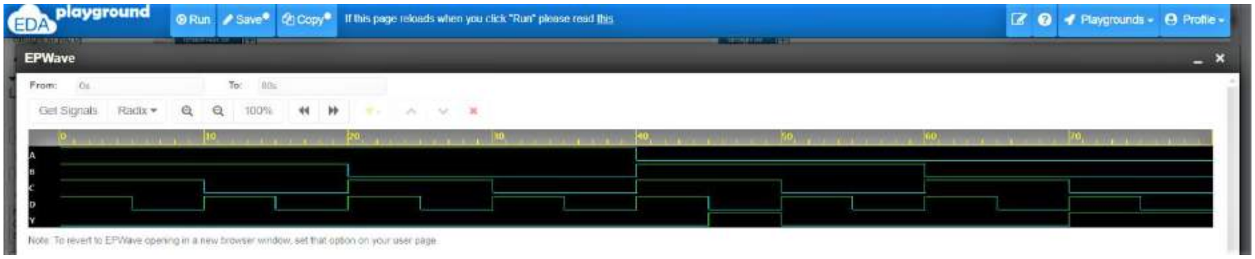


```
module tb_gate();
  reg A,B,C,D;
  wire Y;
//print
//Ayushman Pranav E21cseu0245
  gate1 a1(.a(A),.b(B),.c(C),.d(D),.y(Y));
  initial begin
    A=1;B=1;C=1;D=1;#5;
    $display("A B C D Result");
    $display("%b %b %b %b   %b",A,B,C,D,Y);
  A=1;B=1;C=1;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=1;B=1;C=0;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=1;B=1;C=0;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=1;B=0;C=1;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=1;B=0;C=1;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=1;B=0;C=0;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=1;B=0;C=0;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=1;C=1;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=1;C=1;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=1;C=0;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=1;C=0;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=0;C=1;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=0;C=1;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=0;C=0;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=0;C=0;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);

  end
// EP wave
  initial begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
  end
endmodule
```

```
module gate1 (input a,b,c,d, output y);
  assign y = ((~a&~b&~c&~d)|(~a&b&c&~d)|(~a&~b&~c));
endmodule
```



```
[2022-04-08 10:04:57 UTC] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
A B C D Result
1 1 1 1   0
1 1 1 0   0
1 1 0 1   0
1 1 0 0   0
1 0 1 1   0
1 0 1 0   0
1 0 0 1   0
1 0 0 0   0
0 1 1 1   0
0 1 1 0   1
0 1 0 1   0
0 1 0 0   0
0 0 1 1   0
0 0 1 0   0
0 0 0 1   1
0 0 0 0   1
Finding VCD file...
./dump.vcd
[2022-04-08 10:04:59 UTC] Opening EPWave...
Done
```



Note: To revert to EPWave opening in a new browser window, set that option on your user page.

Design bench.sv

```
module gate1 (input a,b,c,d, output y);
  assign y = ((~a&~b&~c&~d)|(~a&b&c&~d)|(~a&~b&~c));
endmodule
```

Test bench.sv

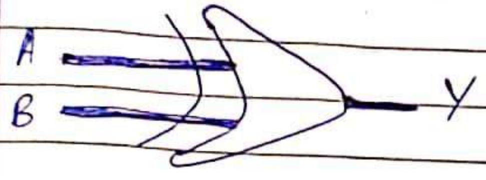
```
module tb_gate();
  reg A,B,C,D;
  wire Y;
```

```
//print
//Ayushman Pranav E21cseu0245
  gate1 a1(.a(A),.b(B),.c(C),.d(D),.y(Y));
  initial begin
    A=1;B=1;C=1;D=1;#5;
    $display("A B C D Result");
    $display("%b %b %b %b   %b",A,B,C,D,Y);
  A=1;B=1;C=1;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=1;B=1;C=0;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=1;B=1;C=0;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=1;B=0;C=1;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=1;B=0;C=1;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=1;B=0;C=0;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=1;B=0;C=0;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=1;C=1;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=1;C=1;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=1;C=0;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=1;C=0;D=0;#5;
```

```
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=0;C=1;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=0;C=1;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=0;C=0;D=1;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);
   A=0;B=0;C=0;D=0;#5;
    $display("%b %b %b %b   %b",A,B,C,D,Y);


  end
// EP wave
  initial begin
    $dumpfile("dump.vcd");
            $dumpvars(1);
  end
endmodule
```

3. Representation of XOR gate using only NAND gates and perform the following operations:
    a. Derive the Boolean expression.
    b. Write the truth table for the above expression
    c. Write a Verilog code for each Boolean expression and then test using wave form and compare with truth table whether your circuit produced same output or not?

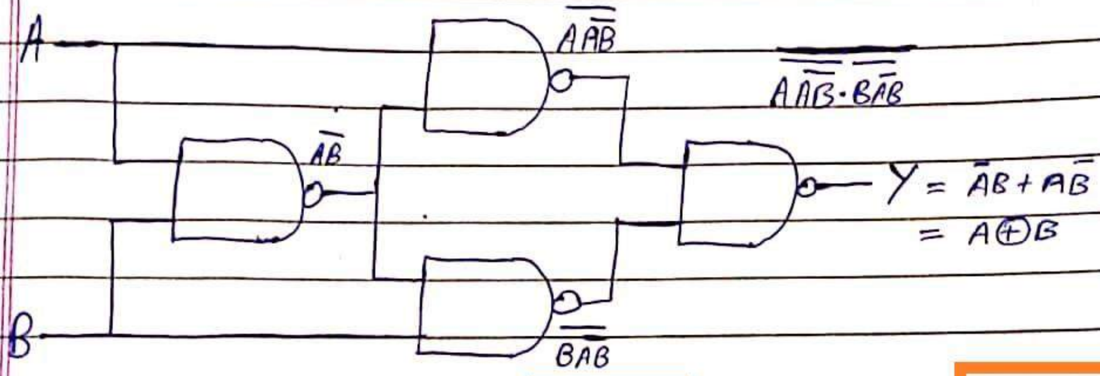
XOR Gate



$Y = A \oplus B$
$= \bar{A}B + A\bar{B}$

XOR Gate

A	B	$Y = \bar{A}B + A\bar{B}$
1	1	0
1	0	1
0	1	1
0	0	0


A
B
$\overline{A\overline{AB}}$
$\overline{AB}$
$\overline{\overline{A\overline{AB}}\cdot\overline{B\overline{AB}}}$
$\overline{B\overline{AB}}$
$Y = \bar{A}B + A\bar{B}$
$= A \oplus B$

A	B	$\bar{A}$	$\bar{B}$	$\bar{A}B$	$A\bar{B}$	$\bar{A}B+A\bar{B}$	$\overline{AB}$	$\overline{A\overline{AB}}$	$\overline{B\overline{AB}}$	$\overline{\overline{A\overline{AB}}\cdot\overline{B\overline{AB}}}$
1	1	0	0	0	0	0	0	1	1	0
1	0	0	1	0	1	1	1	0	1	1
0	1	1	0	1	0	1	1	1	0	1
0	0	1	1	0	0	0	1	1	1	0

Logically equivalent

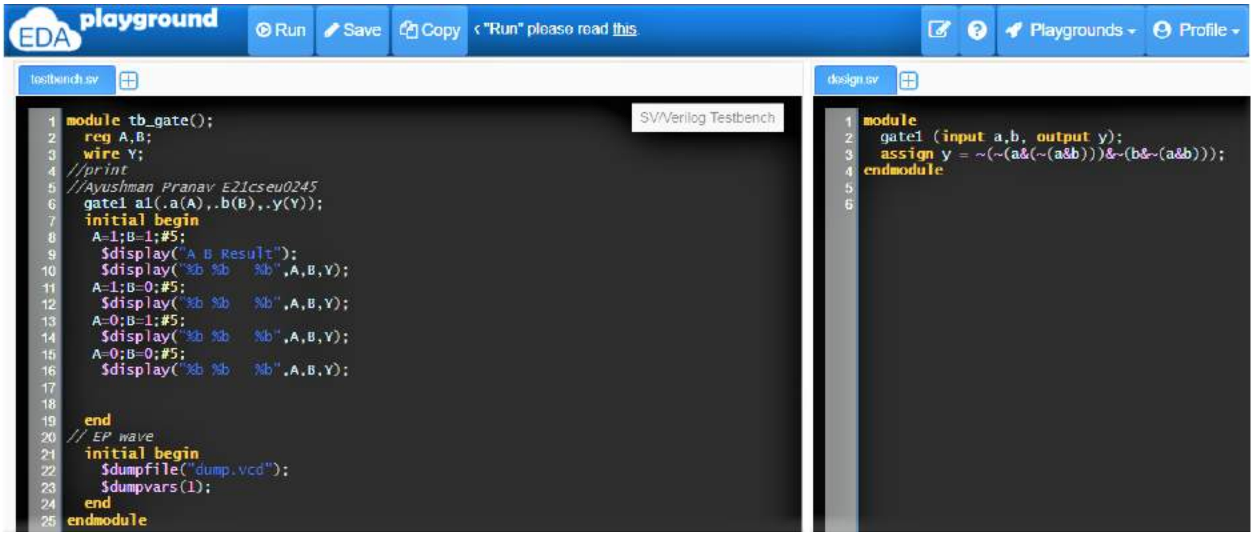
$$\bar{A}B + A\bar{B}$$

$$= A\bar{A} + A\bar{B} + B\bar{A} + B\cdot\bar{B}$$

$$= A(\bar{A}+\bar{B}) + B(\bar{A}+\bar{B})$$

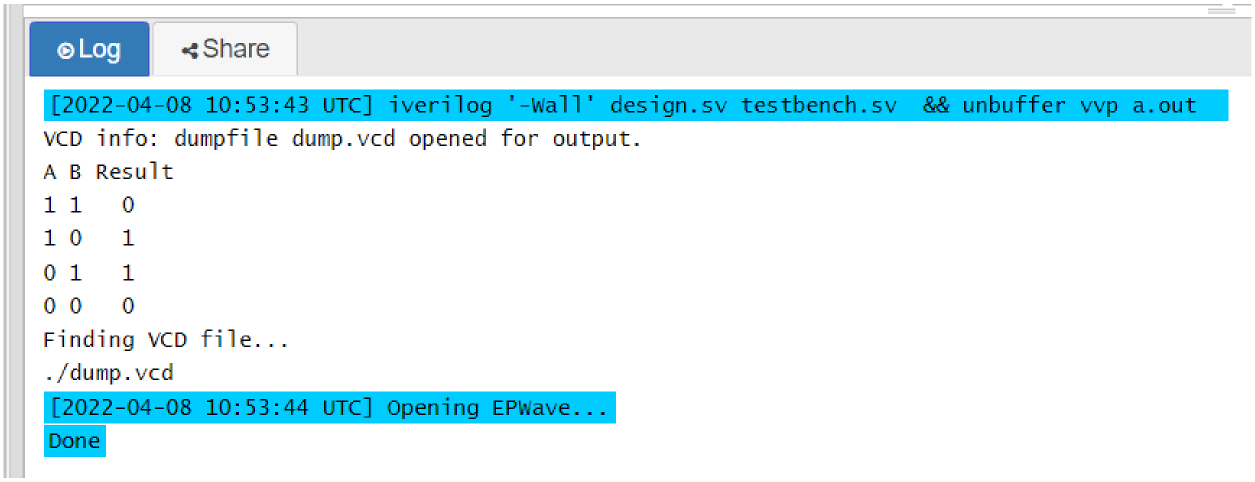
$$= A\cdot\overline{AB} + B\,\overline{AB}$$

$$= \overline{\overline{A\overline{AB}}\;\overline{B\overline{AB}}}$$

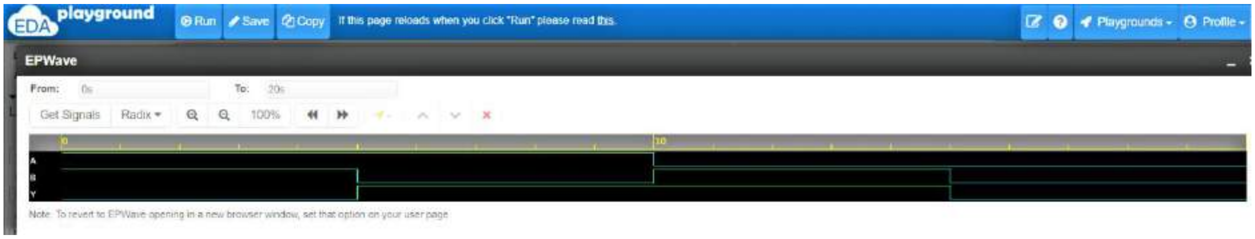


```
module tb_gate();
  reg A,B;
  wire Y;
//print
//Ayushman Pranav E21cseu0245
  gate1 a1(.a(A),.b(B),.y(Y));
  initial begin
   A=1;B=1;#5;
    $display("A B Result");
    $display("%b %b   %b",A,B,Y);
   A=1;B=0;#5;
    $display("%b %b   %b",A,B,Y);
   A=0;B=1;#5;
    $display("%b %b   %b",A,B,Y);
   A=0;B=0;#5;
    $display("%b %b   %b",A,B,Y);


  end
// EP wave
  initial begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
  end
endmodule
```

```
module
  gate1 (input a,b, output y);
  assign y = ~(~(a&(~(a&b)))&~(b&~(a&b)));
endmodule
```



```
[2022-04-08 10:53:43 UTC] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
A B Result
1 1   0
1 0   1
0 1   1
0 0   0
Finding VCD file...
./dump.vcd
[2022-04-08 10:53:44 UTC] Opening EPWave...
Done
```



If this page reloads when you click "Run" please read this.

Note: To revert to EPWave opening in a new browser window, set that option on your user page

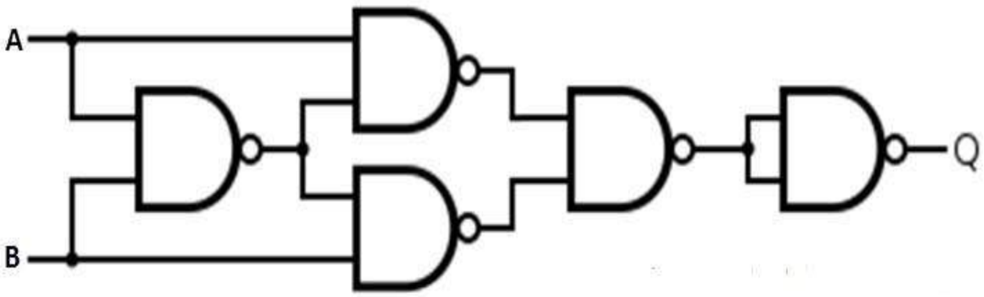
Design bench.sv

```
module
  gate1 (input a,b, output y);
  assign y = ~(~(a&(~(a&b)))&~(b&~(a&b)));
endmodule
```

Test bench.sv

```
module tb_gate();
  reg A,B;
  wire Y;
//print
//Ayushman Pranav E21cseu0245
  gate1 a1(.a(A),.b(B),.y(Y));
  initial begin
   A=1;B=1;#5;
    $display("A B Result");
    $display("%b %b   %b",A,B,Y);
   A=1;B=0;#5;
    $display("%b %b   %b",A,B,Y);
   A=0;B=1;#5;
    $display("%b %b   %b",A,B,Y);
   A=0;B=0;#5;
    $display("%b %b   %b",A,B,Y);



  end
// EP wave
  initial begin
    $dumpfile("dump.vcd");
          $dumpvars(1);
  end
endmodule
```

4. For the given circuit diagram do the following:
   a. Derive the Boolean expression.
   b. Write the truth table for the above expression
   c. Write a Verilog code for each Boolean expression and then test using wave form and compare with truth table whether your circuit produced same output or not?



$\bar{A}B + A\bar{B}$

$= A\bar{A} + A\bar{B} + B\bar{A} + B\cdot\bar{B}$

$= A(\bar{A} + \bar{B}) + B(\bar{A} + \bar{B})$

$= A\cdot\overline{AB} + B\,\overline{AB}$

$= \overline{\overline{A\overline{AB}}\;\overline{B\overline{AB}}}$

$= \overline{\bar{A}B + A\bar{B}}$

$= \overline{\bar{A}B}\,.\;\overline{A\bar{B}}$

$= (A + \bar{B}).(\bar{A} + B) = A\bar{A} + AB + \bar{B}\bar{A} + \bar{B}B$

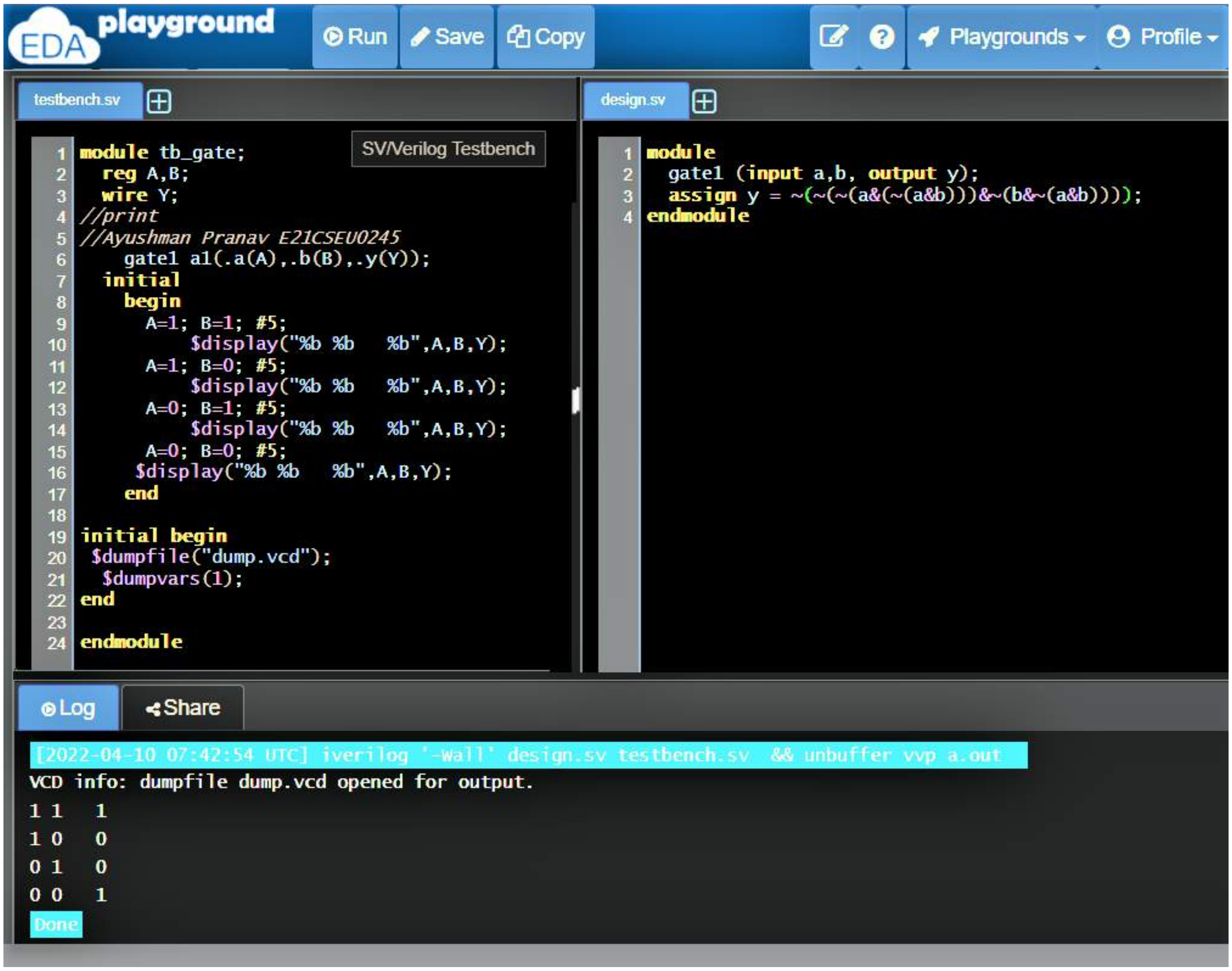
$= F + AB + \bar{B}\bar{A} + F$

$= AB + \bar{A}\bar{B}$

$= A \odot B$

A	B	$\bar{A}$	$\bar{B}$	AB	$\bar{A}\bar{B}$	$AB+\bar{A}\bar{B}$	$\overline{A\oplus B}$
1	1	0	0	1	0	1	1
1	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0
0	0	1	1	0	1	1	1

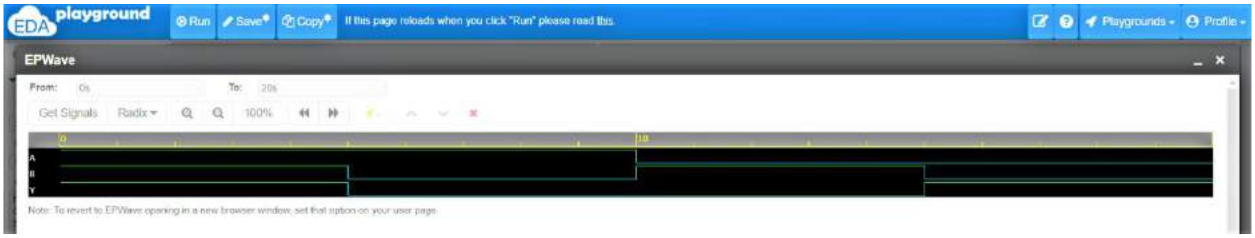
Logically equivalent



```
module tb_gate;
  reg A,B;
  wire Y;
//print
//Ayushman Pranav E21CSEU0245
    gate1 a1(.a(A),.b(B),.y(Y));
  initial
    begin
      A=1; B=1; #5;
          $display("%b %b   %b",A,B,Y);
      A=1; B=0; #5;
          $display("%b %b   %b",A,B,Y);
      A=0; B=1; #5;
          $display("%b %b   %b",A,B,Y);
      A=0; B=0; #5;
     $display("%b %b   %b",A,B,Y);
    end

initial begin
 $dumpfile("dump.vcd");
  $dumpvars(1);
end

endmodule
```

```
module
  gate1 (input a,b, output y);
  assign y = ~(~(~(a&(~(a&b)))&~(b&~(a&b))));
endmodule
```

```
[2022-04-10 07:42:54 UTC] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
1 1   1
1 0   0
0 1   0
0 0   1
Done
```



If this page reloads when you click "Run" please read this.

From: 0s To: 20s

Note: To revert to EPWave opening in a new browser window, set that option on your user page.

Design bench.sv

```
module
  gate1 (input a,b, output y);
  assign y = ~(~(~(a&(~(a&b)))&~(b&~(a&b))));
endmodule
```

Test bench.sv

```
module tb_gate;
  reg A,B;
  wire Y;
//print
//Ayushman Pranav E21CSEU0245
    gate1 a1(.a(A),.b(B),.y(Y));
  initial
    begin
      A=1; B=1; #5;
          $display("%b %b   %b",A,B,Y);
      A=1; B=0; #5;
          $display("%b %b   %b",A,B,Y);
      A=0; B=1; #5;
          $display("%b %b   %b",A,B,Y);
      A=0; B=0; #5;
     $display("%b %b   %b",A,B,Y);
    end
```

```
initial begin
 $dumpfile("dump.vcd");
  $dumpvars(1);
end

endmodule
```