

### 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?

$$F(ABC) = ABC + A(B' + C')$$

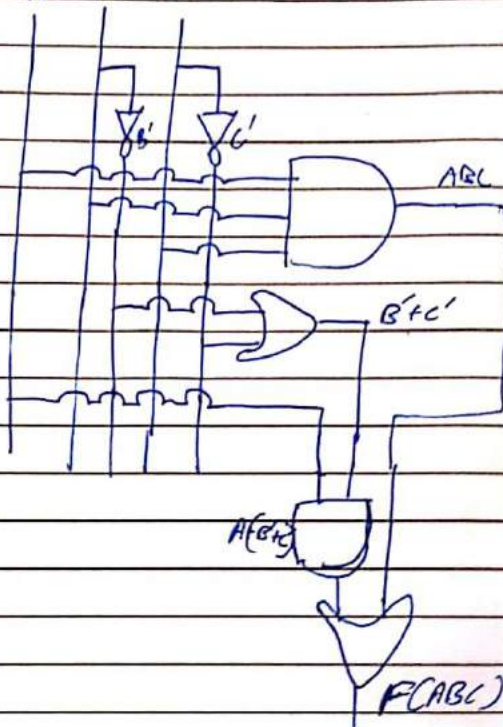
Intersection

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  
0  
0  
0  
0

A B C



EDA playground

Run

Save

Copy

Run please read this.

Playgrounds

Profile

testbench.sv

```

1 module tb_gate1();
2   reg A,B,C;
3   wire Y;
4   //print
5   //Ayushman Pranav E21cseu0245
6   gate1 al(a(A),b(B),c(C),y(Y));
7   initial begin
8     A=1; B=1; C=1; #5;
9     $display("gate1");
10    $display("A B C Result");
11    $display("%b %b %b %b",A,B,C,Y);
12    A=1; B=1; C=0; #5;
13    $display("%b %b %b %b",A,B,C,Y);
14    A=1; B=0; C=1; #5;
15    $display("%b %b %b %b",A,B,C,Y);
16    A=1; B=0; C=0; #5;
17    $display("%b %b %b %b",A,B,C,Y);
18    A=0; B=1; C=1; #5;
19    $display("%b %b %b %b",A,B,C,Y);
20    A=0; B=1; C=0; #5;
21    $display("%b %b %b %b",A,B,C,Y);
22    A=0; B=0; C=1; #5;
23    $display("%b %b %b %b",A,B,C,Y);
24    A=0; B=0; C=0; #5;
25    $display("%b %b %b %b",A,B,C,Y);
26  end
27  // EP wave
28  initial begin
29    $dumpfile("dump.vcd");
30    $dumpvars(1);
31  end
32 end
33 endmodule

```

SV/Verilog Testbench

design.sv

```

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

```

Log

Share

[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

EDA playground

Run

Save

Copy

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

Playgrounds

Profile

EPWave

From: 0s To: 40s

Get Signals Radix 100%

10

20

30

A

B

C

Y

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
```

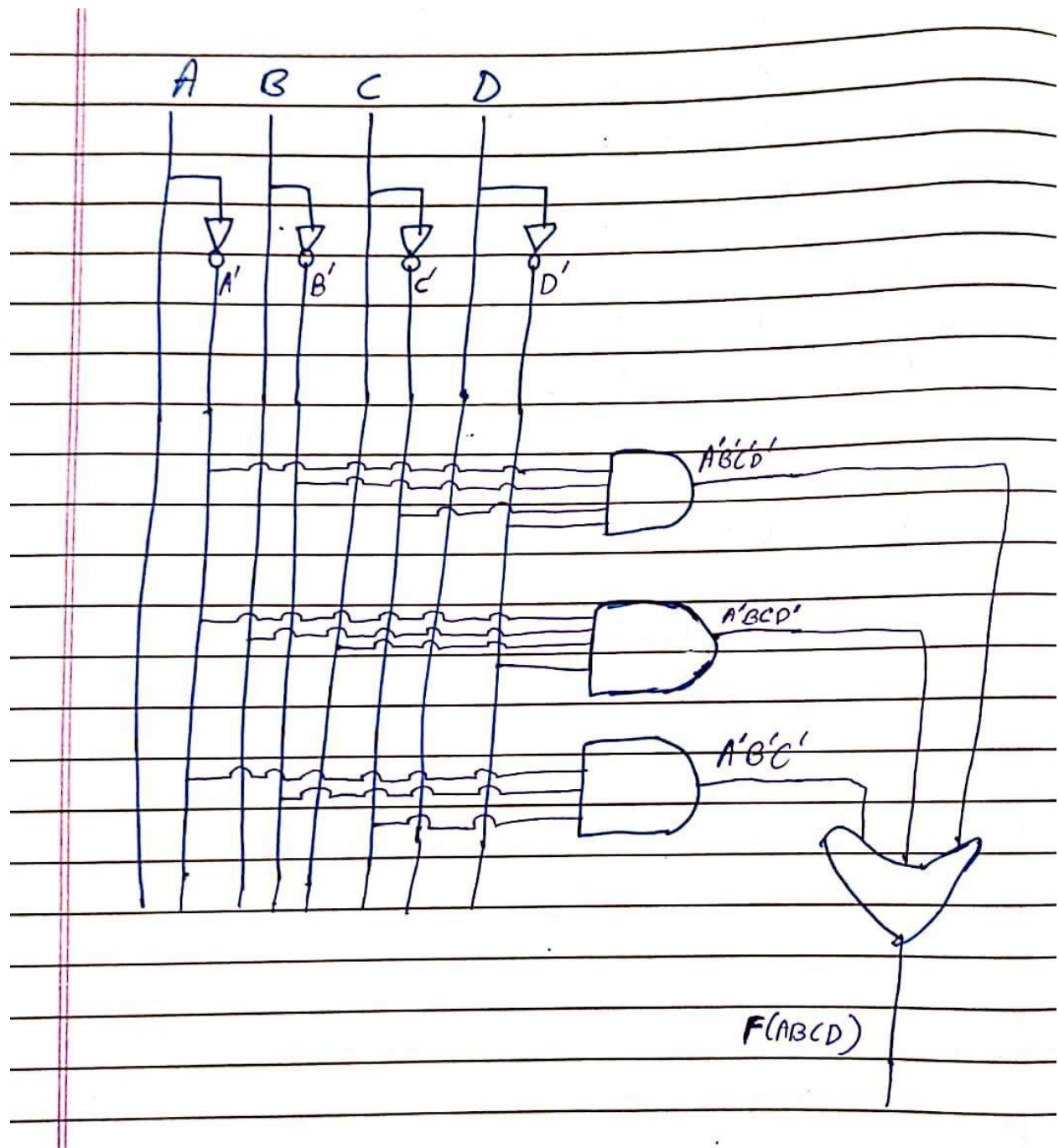
2. Perform the following operation on a given Boolean expression:

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

- 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?

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

				X				X	X	X	X
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





testbench.sv

SV/Venilog Testbench

```

1 module tb_gate();
2   reg A,B,C,D;
3   wire Y;
4   //print
5   //Ayushman Pranav E21cseu0245
6   gate1 a1(.a(A),.b(B),.c(C),.d(D),.y(Y));
7   initial begin
8     A=1;B=1;C=1;D=1;#5;
9     $display("A B C D Result");
10    $display("%b %b %b %b",A,B,C,D,Y);
11    A=1;B=1;C=1;D=0;#5;
12    $display("%b %b %b %b",A,B,C,D,Y);
13    A=1;B=1;C=0;D=1;#5;
14    $display("%b %b %b %b",A,B,C,D,Y);
15    A=1;B=1;C=0;D=0;#5;
16    $display("%b %b %b %b",A,B,C,D,Y);
17    A=1;B=0;C=1;D=1;#5;
18    $display("%b %b %b %b",A,B,C,D,Y);
19    A=1;B=0;C=1;D=0;#5;
20    $display("%b %b %b %b",A,B,C,D,Y);
21    A=1;B=0;C=0;D=1;#5;
22    $display("%b %b %b %b",A,B,C,D,Y);
23    A=1;B=0;C=0;D=0;#5;
24    $display("%b %b %b %b",A,B,C,D,Y);
25    A=0;B=1;C=1;D=1;#5;
26    $display("%b %b %b %b",A,B,C,D,Y);
27    A=0;B=1;C=1;D=0;#5;
28    $display("%b %b %b %b",A,B,C,D,Y);
29    A=0;B=1;C=0;D=1;#5;
30    $display("%b %b %b %b",A,B,C,D,Y);
31    A=0;B=1;C=0;D=0;#5;
32    $display("%b %b %b %b",A,B,C,D,Y);
33    A=0;B=0;C=1;D=1;#5;
34    $display("%b %b %b %b",A,B,C,D,Y);
35    A=0;B=0;C=1;D=0;#5;
36    $display("%b %b %b %b",A,B,C,D,Y);
37    A=0;B=0;C=0;D=1;#5;
38    $display("%b %b %b %b",A,B,C,D,Y);
39    A=0;B=0;C=0;D=0;#5;
40    $display("%b %b %b %b",A,B,C,D,Y);
41  end
42  // EP wave
43  initial begin
44    $dumpfile("dump.vcd");
45    $dumpvars(1);
46  end
47 endmodule

```

design.sv

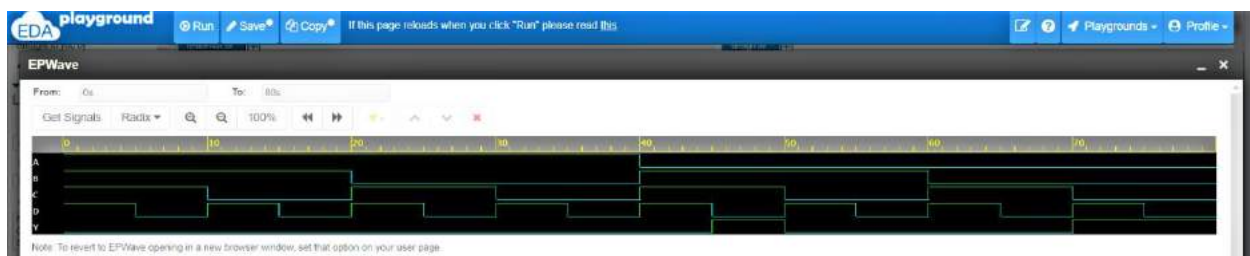
```

1 module gate1 (input a,b,c,d, output y);
2   assign y = ((~a&b&c&d)|(~a&b&c&~d)|(~a&b&~c&d));
3 endmodule
4
5

```



```
Log Share
[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
```



## 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);

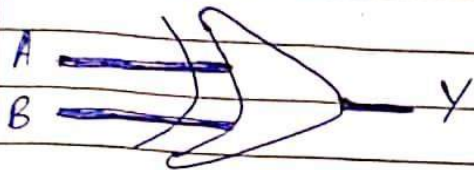
end

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

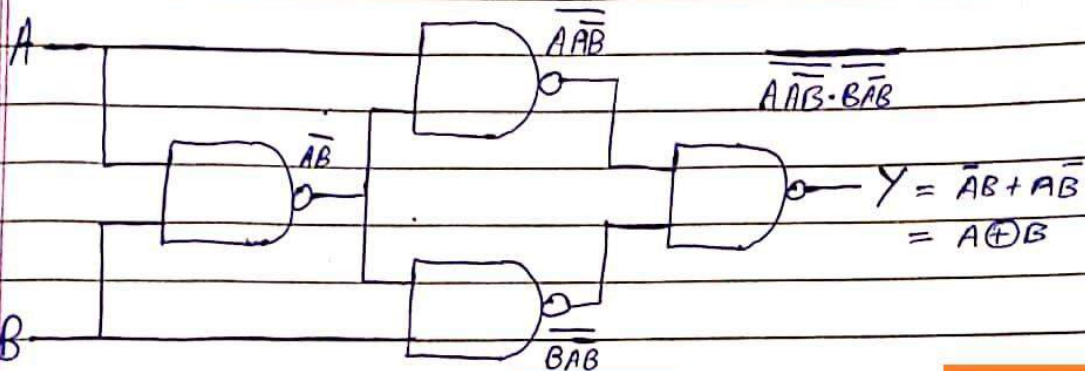


$$Y = A \oplus B$$

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

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

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

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

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

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

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

EDA playground

Run

Save

Copy

< "Run" please read this

Playgrounds

Profile

testbench.sv

```

1 module tb_gate();
2     reg A,B;
3     wire Y;
4     //print
5     //Ayushman Pranav E21cseu0245
6     gate1 a1(.a(A),.b(B),.y(Y));
7     initial begin
8         A=1;B=1;#5;
9         $display("A,B,Result");
10        $display("%b %b %b",A,B,Y);
11        A=1;B=0;#5;
12        $display("%b %b %b",A,B,Y);
13        A=0;B=1;#5;
14        $display("%b %b %b",A,B,Y);
15        A=0;B=0;#5;
16        $display("%b %b %b",A,B,Y);
17    end
18    end
19    // EP wave
20    initial begin
21        $dumpfile("dump.vcd");
22        $dumpvars(1);
23    end
24 endmodule
25

```

SV/Verilog Testbench

design.sv

```

1 module
2     gate1 (input a,b, output y);
3     assign y = ~(~(a&(~(a&b)))&~(b&~(a&b)));
4 endmodule
5
6

```

LogShare

[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

EDA playgroundRunSaveCopyIf this page reloads when you click "Run" please read this.

EPWave

From: 0sTo: 20s

Get SignalsRadix100%



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

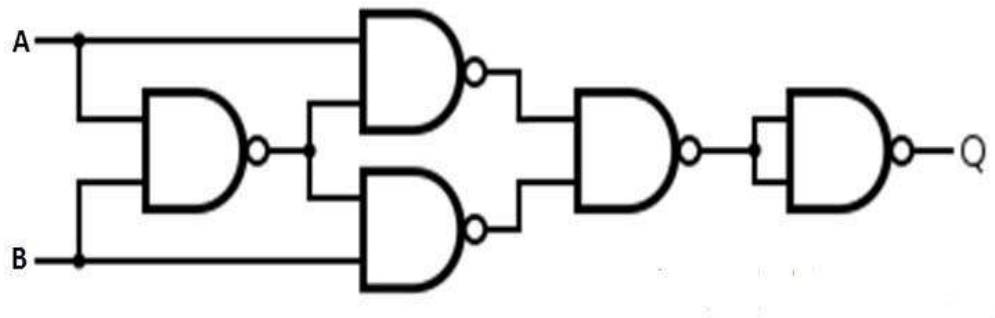
// EP wave

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

endmodule

```

4. For the given circuit diagram do the following:
- Derive the Boolean expression.
  - Write the truth table for the above expression
  - 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?





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

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

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

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

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

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

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

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

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

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

$$= A \odot B$$

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

Logically equivalent

EDA playground

Run

Save

Copy

?

Playgrounds

Profile

testbench.sv

```

1 module tb_gate;
2   reg A,B;
3   wire Y;
4   //print
5   //Ayushman Pranav E21CSEU0245
6   gate1 a1(.a(A),.b(B),.y(Y));
7   initial
8   begin
9     A=1; B=1; #5;
10    $display("%b %b %b",A,B,Y);
11    A=1; B=0; #5;
12    $display("%b %b %b",A,B,Y);
13    A=0; B=1; #5;
14    $display("%b %b %b",A,B,Y);
15    A=0; B=0; #5;
16    $display("%b %b %b",A,B,Y);
17  end
18
19  initial begin
20    $dumpfile("dump.vcd");
21    $dumpvars(1);
22  end
23
24 endmodule

```

design.sv

```

1 module
2   gate1 (input a,b, output y);
3   assign y = ~(~(a&(~(a&b)))&~(b&~(a&b)));
4 endmodule

```

Log

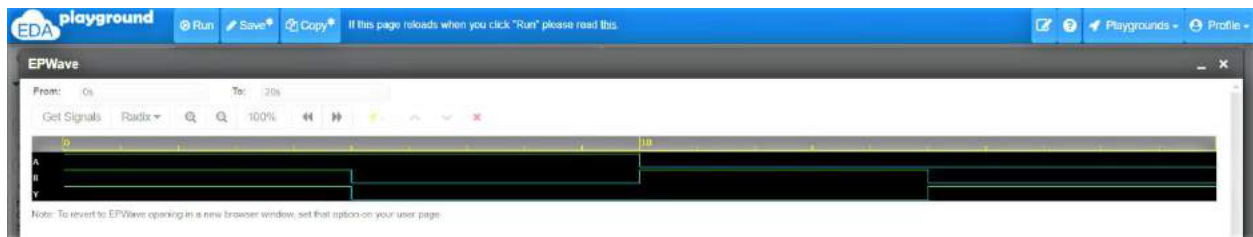
Share

[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



## 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
```