

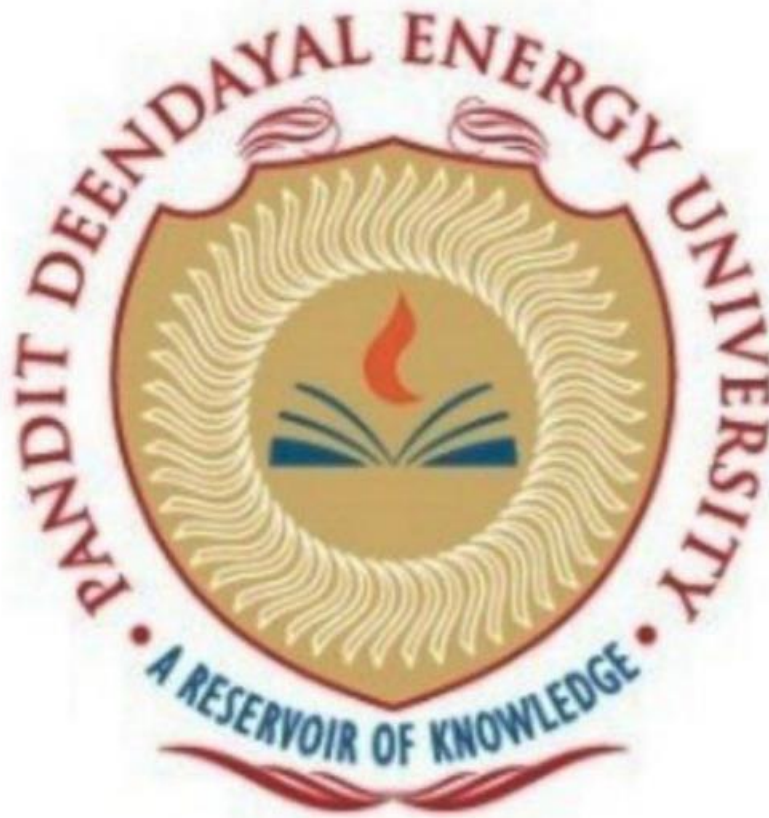
Name - Akshat Shah

Roll No - 21BCP322

Sem – III

Digital Electronics and Computer Organization Lab

Course Code – 20CP203P



Department of Computer Science Engineering  
School of Technology,  
Pandit Deendayal Energy University

Date: 17-08-2022

## Assignment-2

Regarding how basic gates work up on interfacing with a hardware description language (HDL) Verilog.

### Table of Contents

1. AND gate.....	1
2. OR gate .....	2
3. XOR gate .....	3
4. NOR gate.....	4
5. NAND gate .....	5
6. XNOR gate.....	6

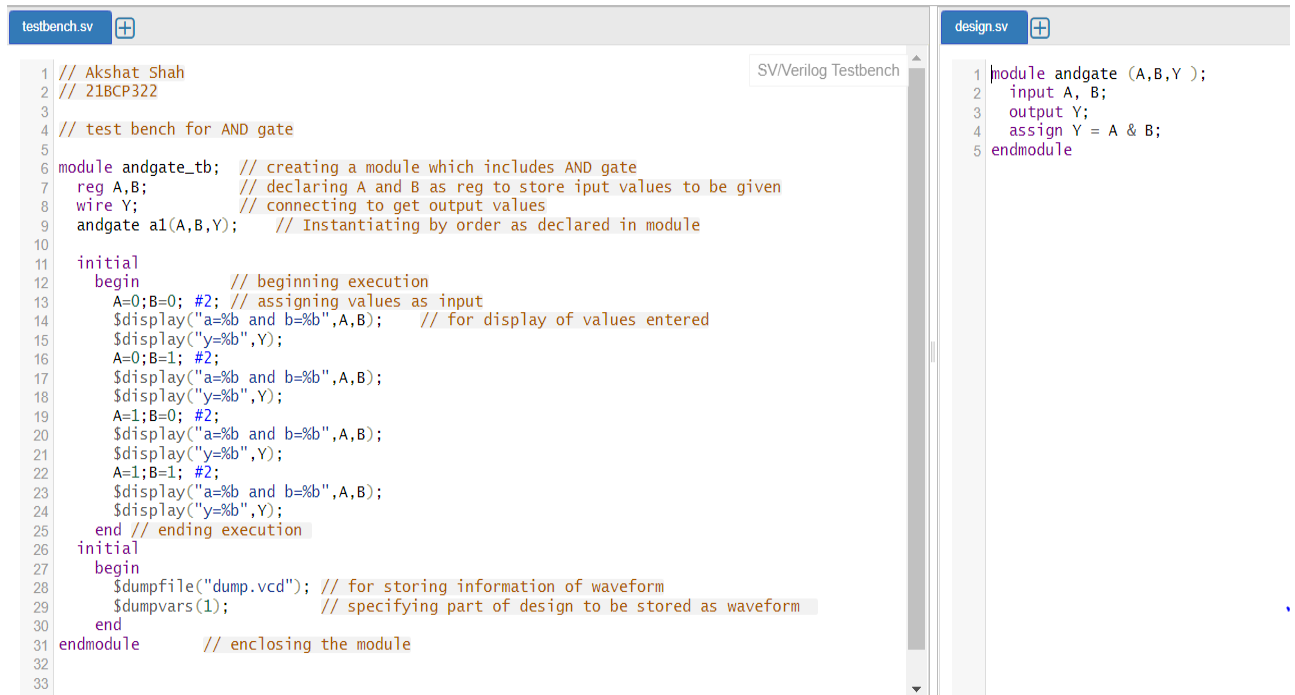
## List of Figures

Fig 1.1 - Testbench and design code for AND gate.....	1
Fig 1.2 - Output as truth table of AND.....	1
Fig 1.3 - Waveform of AND gate output.....	1
Fig 2.1 - Testbench and design code for OR gate.....	2
Fig 2.2 - Output as truth table of OR gate.....	2
Fig 2.3 - Waveform of OR gate output.....	2
Fig 3.1 - Testbench and design code for XOR gate.....	3
Fig 3.2 - Output as truth table of XOR gate.....	3
Fig 3.3 - Waveform of XOR gate output.....	3
Fig 4.1 - Testbench and design code for NOR gate.....	4
Fig 4.2 - Output as truth table of NOR gate.....	4
Fig 4.3 - Waveform of NOR gate output.....	4
Fig 5.1 - Test bench and design code for NAND gate.....	5
Fig 5.2 - Output as truth table of NAND gate.....	5
Fig 5.3 - Waveform of NAND gate output.....	5
Fig 6.1 - Testbench and design code for XNOR gate.....	6
Fig 6.2 - Output as truth table of XNOR gate.....	6
Fig 6.3 - Waveform of XNOR gate output .....	6

## 1. AND gate

Question: Design an AND gate with Logisim. Use the input and output labels according to given truth table. Write a Verilog code to implement AND gate. Write the corresponding Testbench code for the verification of your Verilog code.

Code:



The screenshot shows the Logisim IDE with two tabs: 'testbench sv' and 'design sv'. The 'testbench sv' tab is active, displaying the following Verilog code:

```
1 // Akshat Shah
2 // 218CP322
3
4 // test bench for AND gate
5
6 module andgate_tb; // creating a module which includes AND gate
7   reg A,B; // declaring A and B as reg to store input values to be given
8   wire Y; // connecting to get output values
9   andgate a1(A,B,Y); // Instantiating by order as declared in module
10
11   initial
12   begin // beginning execution
13     A=0;B=0; #2; // assigning values as input
14     $display("a=%b and b=%b",A,B); // for display of values entered
15     $display("y=%b",Y);
16     A=0;B=1; #2;
17     $display("a=%b and b=%b",A,B);
18     $display("y=%b",Y);
19     A=1;B=0; #2;
20     $display("a=%b and b=%b",A,B);
21     $display("y=%b",Y);
22     A=1;B=1; #2;
23     $display("a=%b and b=%b",A,B);
24     $display("y=%b",Y);
25   end // ending execution
26   initial
27   begin
28     $dumpfile("dump.vcd"); // for storing information of waveform
29     $dumpvars(1); // specifying part of design to be stored as waveform
30   end
31 endmodule // enclosing the module
32
33
```

The 'design sv' tab is also visible, showing the following Verilog code for the AND gate:

```
1 module andgate (A,B,Y);
2   input A, B;
3   output Y;
4   assign Y = A & B;
5 endmodule
```

Fig 1.1 - Testbench and design code for AND gate

Output:

```
[2022-08-20 12:33:44 EDT] iverilog '-wall' design.sv testbench.sv && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
a=0 and b=0
y=0
a=0 and b=1
y=0
a=1 and b=0
y=0
a=1 and b=1
y=1
Done
```

Fig 1.2 - Output as truth table of AND

Waveform:



Fig 1.3 - Waveform of AND gate output

## 2. OR gate

Question: Design an OR gate with Logisim. Use the input and output labels according to given truth table. Write a Verilog code to implement OR gate. Write the corresponding Testbench code for the verification of your Verilog code.

Code:

```
testbench.sv
1 // Akshat Shah
2 // 21BCP322
3
4 // test bench for OR gate
5
6 module orgate_tb; // creating a module which includes the gate
7 reg A,B; // declaring A and B as reg to store input values to be given
8 wire Y; // connecting to get output values
9 orgate a1(A,B,Y); // Instantiating by order as declared in module
10
11 initial
12 begin // beginning execution
13 A=0;B=0; #2; // assigning values as input
14 $display("a=%b and b=%b",A,B); // for display of values entered
15 $display("y=%b",Y);
16 A=0;B=1; #2;
17 $display("a=%b and b=%b",A,B);
18 $display("y=%b",Y);
19 A=1;B=0; #2;
20 $display("a=%b and b=%b",A,B);
21 $display("y=%b",Y);
22 A=1;B=1; #2;
23 $display("a=%b and b=%b",A,B);
24 $display("y=%b",Y);
25 end // ending execution
26 initial
27 begin
28 $dumpfile("dump.vcd"); // for storing information of waveform
29 $dumpvars(1); // specifying part of design to be stored as waveform
30 end
31 endmodule // enclosing the module
32
33
design.sv
1 module orgate (A,B,Y );
2 input A, B;
3 output Y;
4 assign Y = A || B;
5 endmodule
```

Fig 2.1 - Testbench and design code for OR gate

Output:

```
[2022-08-20 12:57:55 EDT] iverilog '-wall' design.sv testbench.sv && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
a=0 and b=0
y=0
a=0 and b=1
y=1
a=1 and b=0
y=1
a=1 and b=1
y=1
Finding VCD file...
./dump.vcd
[2022-08-20 12:57:55 EDT] Opening EPWave...
Done
```

Fig 2.2 - Output as truth table of OR gate

Waveform:

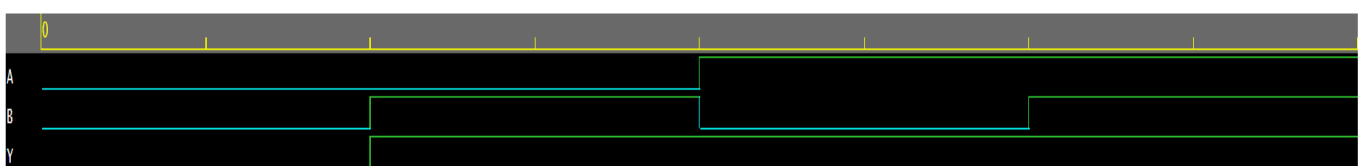
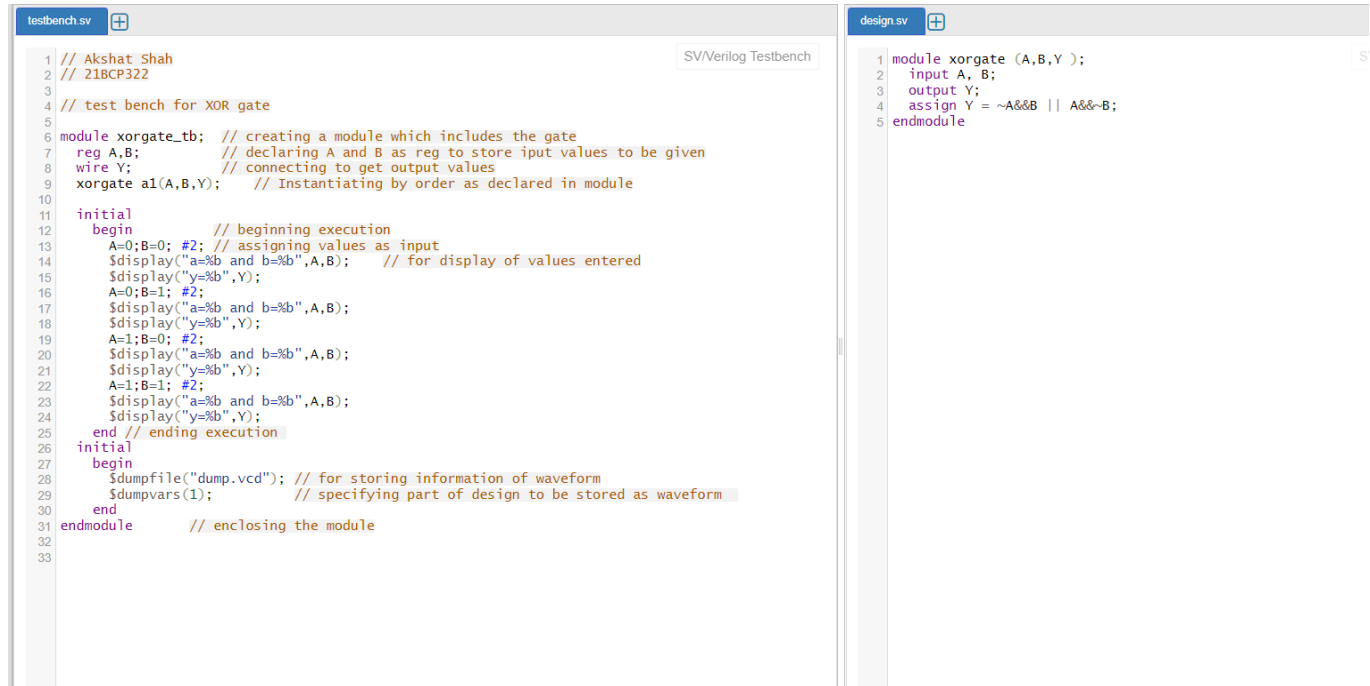


Fig 2.3 - Waveform of OR gate output

### 3. XOR gate

**Question:** Design an XOR gate with Logisim. Use the input and output labels according to given truth table. Write a Verilog code to implement XOR gate. Write the corresponding Testbench code for the verification of your Verilog code.

**Code:**



The screenshot shows the Logisim IDE with two panels. The left panel, titled 'testbench sv', contains the following Verilog code:

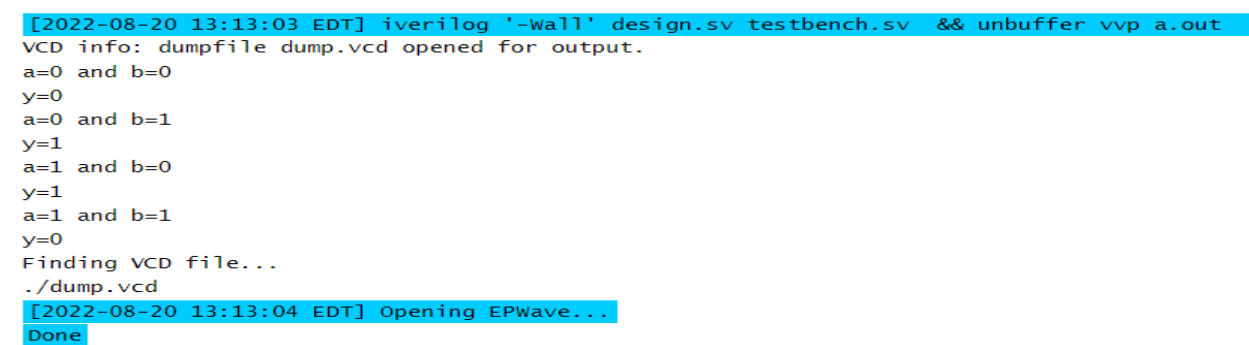
```
1 // Akshat Shah
2 // 21BCP322
3
4 // test bench for XOR gate
5
6 module xorgate_tb; // creating a module which includes the gate
7 reg A,B; // declaring A and B as reg to store input values to be given
8 wire Y; // connecting to get output values
9 xorgate a1(A,B,Y); // Instantiating by order as declared in module
10
11 initial
12 begin // beginning execution
13 A=0;B=0; #2; // assigning values as input
14 $display("a=%b and b=%b",A,B); // for display of values entered
15 $display("y=%b",Y);
16 A=0;B=1; #2;
17 $display("a=%b and b=%b",A,B);
18 $display("y=%b",Y);
19 A=1;B=0; #2;
20 $display("a=%b and b=%b",A,B);
21 $display("y=%b",Y);
22 A=1;B=1; #2;
23 $display("a=%b and b=%b",A,B);
24 $display("y=%b",Y);
25 end // ending execution
26 initial
27 begin
28 $dumpfile("dump.vcd"); // for storing information of waveform
29 $dumpvars(1); // specifying part of design to be stored as waveform
30 end
31 endmodule // enclosing the module
32
33
```

The right panel, titled 'design sv', contains the following Verilog code:

```
1 module xorgate (A,B,Y );
2 input A, B;
3 output Y;
4 assign Y = ~A&&B || A&&~B;
5 endmodule
```

Fig 3.1 - Testbench and design code for XOR gate

**Output:**



The screenshot shows the terminal output of the Verilog testbench execution. The output is as follows:

```
[2022-08-20 13:13:03 EDT] iverilog '-wall' design.sv testbench.sv && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
a=0 and b=0
y=0
a=0 and b=1
y=1
a=1 and b=0
y=1
a=1 and b=1
y=0
Finding VCD file...
./dump.vcd
[2022-08-20 13:13:04 EDT] Opening EPWave...
Done
```

Fig 3.2 - Output as truth table of XOR gate

**Waveform:**

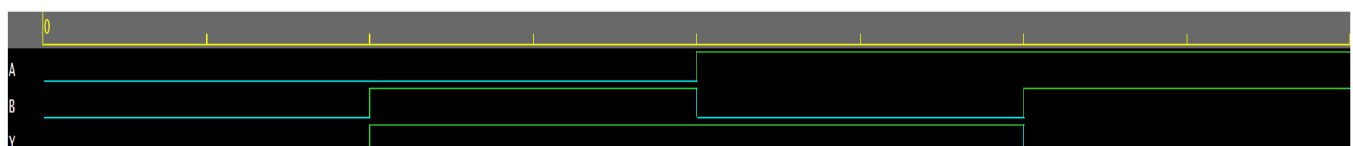
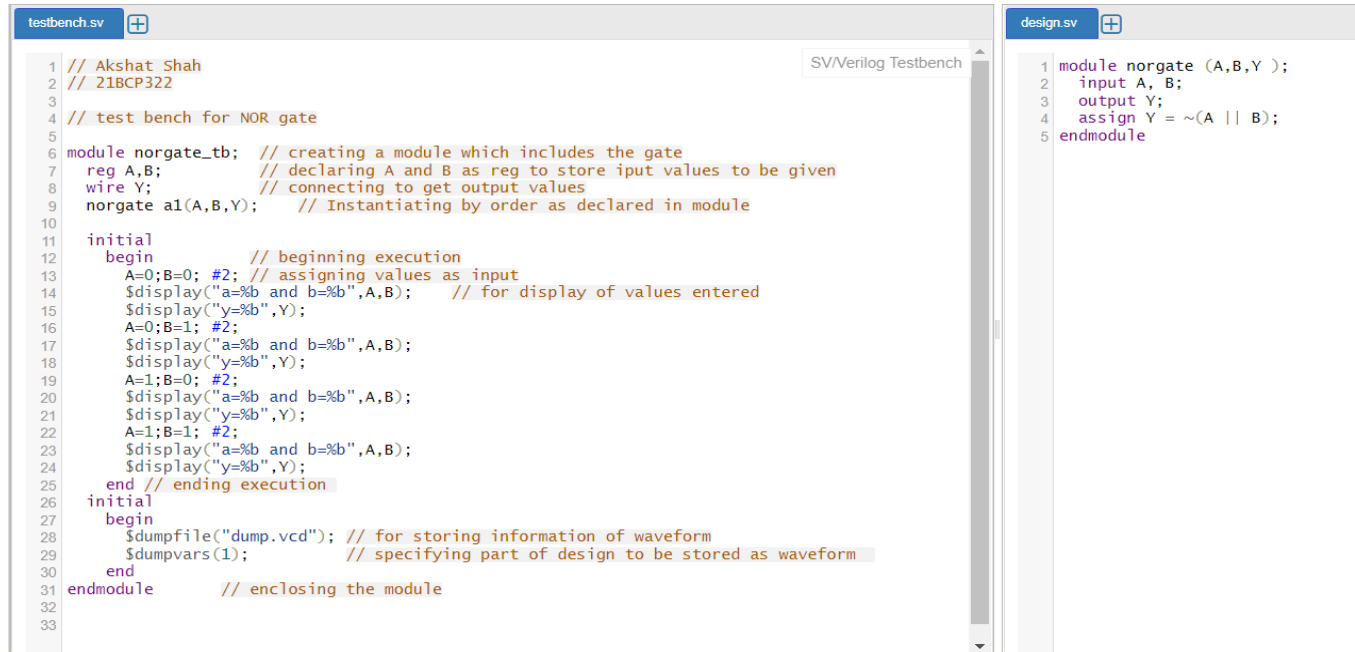


Fig 3.3- Waveform of XOR gate output

## 4. NOR gate

**Question:** Design a NOR gate with Logisim. Use the input and output labels according to given truth table. Write a Verilog code to implement NOR gate. Write the corresponding Testbench code for the verification of your Verilog code.

Code:



```
testbench.vv
1 // Akshat Shah
2 // 21BCP322
3
4 // test bench for NOR gate
5
6 module norgate_tb; // creating a module which includes the gate
7   reg A,B; // declaring A and B as reg to store input values to be given
8   wire Y; // connecting to get output values
9   norgate a1(A,B,Y); // Instantiating by order as declared in module
10
11   initial
12     begin // beginning execution
13       A=0;B=0; #2; // assigning values as input
14       $display("a=%b and b=%b",A,B); // for display of values entered
15       $display("y=%b",Y);
16       A=0;B=1; #2;
17       $display("a=%b and b=%b",A,B);
18       $display("y=%b",Y);
19       A=1;B=0; #2;
20       $display("a=%b and b=%b",A,B);
21       $display("y=%b",Y);
22       A=1;B=1; #2;
23       $display("a=%b and b=%b",A,B);
24       $display("y=%b",Y);
25     end // ending execution
26   initial
27     begin
28       $dumpfile("dump.vcd"); // for storing information of waveform
29       $dumpvars(1); // specifying part of design to be stored as waveform
30     end
31 endmodule // enclosing the module
32
33
design.vv
1 module norgate (A,B,Y );
2   input A, B;
3   output Y;
4   assign Y = ~(A || B);
5 endmodule
```

Fig 4.1 - Testbench and design code for NOR gate

Output:

```
[2022-08-20 13:21:54 EDT] iverilog '-wall' design.vv testbench.vv && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
a=0 and b=0
y=1
a=0 and b=1
y=0
a=1 and b=0
y=0
a=1 and b=1
y=0
Finding VCD file...
./dump.vcd
[2022-08-20 13:21:54 EDT] Opening EPWave...
Done
```

Fig 4.2 - Output as truth table of NOR gate

Waveform:

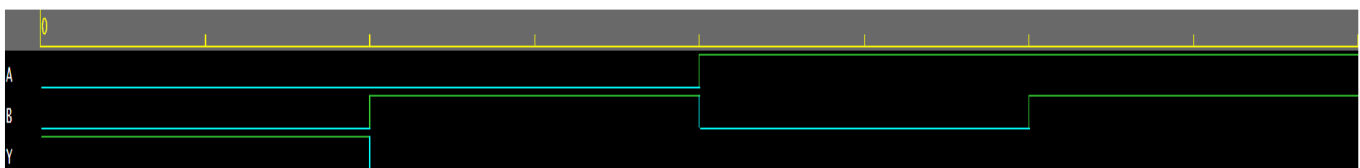


Fig 4.3 - Waveform of NOR gate output

## 5. NAND gate

**Question:** Design a NAND gate with Logisim. Use the input and output labels according to given truth table. Write a Verilog code to implement NAND gate. Write the corresponding Testbench code for the verification of your Verilog code.

Code:

```
testbench.sv
1 // Akshat Shah
2 // 21BCP322
3
4 // test bench for NAND gate
5
6 module nandgate_tb; // creating a module which includes the gate
7   reg A,B;          // declaring A and B as reg to store input values to be given
8   wire Y;           // connecting to get output values
9   nandgate a1(A,B,Y); // Instantiating by order as declared in module
10
11   initial
12     begin           // beginning execution
13       A=0;B=0; #2; // assigning values as input
14       $display("a=%b and b=%b",A,B); // for display of values entered
15       $display("y=%b",Y);
16       A=0;B=1; #2;
17       $display("a=%b and b=%b",A,B);
18       $display("y=%b",Y);
19       A=1;B=0; #2;
20       $display("a=%b and b=%b",A,B);
21       $display("y=%b",Y);
22       A=1;B=1; #2;
23       $display("a=%b and b=%b",A,B);
24       $display("y=%b",Y);
25     end // ending execution
26   initial
27     begin
28       $dumpfile("dump.vcd"); // for storing information of waveform
29       $dumpvars(1);          // specifying part of design to be stored as waveform
30     end
31 endmodule // enclosing the module
32
33
design.sv
1 module nandgate (A,B,Y );
2   input A, B;
3   output Y;
4   assign Y = ~(A & B);
5 endmodule
```

Fig 5.1 - Test bench and design code for NAND gate

Output:

```
[2022-08-20 13:26:35 EDT] iverilog '-wall' design.sv testbench.sv && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
a=0 and b=0
y=1
a=0 and b=1
y=1
a=1 and b=0
y=1
a=1 and b=1
y=0
Finding VCD file...
./dump.vcd
[2022-08-20 13:26:35 EDT] Opening EPWave...
Done
```

Fig 5.2 - Output as truth table of NAND gate

Waveform:

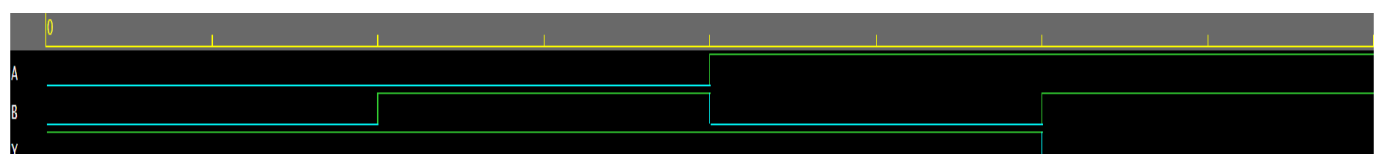


Fig 5.3- Waveform of NAND gate output



## 6. XNOR gate

**Question:** Design an XNOR gate with Logisim. Use the input and output labels according to given truth table. Write a Verilog code to implement XNOR gate. Write the corresponding Testbench code for the verification of your Verilog code.

**Code:**

```
testbench sv  design sv
1 // Akshat Shah
2 // 21BCP322
3
4 // test bench for XNOR gate
5
6 module xnogate_tb; // creating a module which includes the gate
7 reg A,B; // declaring A and B as reg to store input values to be given
8 wire Y; // connecting to get output values
9 xnogate a1(A,B,Y); // Instantiating by order as declared in module
10
11 initial
12 begin // beginning execution
13 A=0;B=0; #2; // assigning values as input
14 $display("a=%b and b=%b",A,B); // for display of values entered
15 $display("y=%b",Y);
16 A=0;B=1; #2;
17 $display("a=%b and b=%b",A,B);
18 $display("y=%b",Y);
19 A=1;B=0; #2;
20 $display("a=%b and b=%b",A,B);
21 $display("y=%b",Y);
22 A=1;B=1; #2;
23 $display("a=%b and b=%b",A,B);
24 $display("y=%b",Y);
25 end // ending execution
26 initial
27 begin
28 $dumpfile("dump.vcd"); // for storing information of waveform
29 $dumpvars(1); // specifying part of design to be stored as waveform
30 end
31 endmodule // enclosing the module
32
```

```
1 module xnogate (A,B,Y);
2 input A, B;
3 output Y;
4 assign Y = ~(A&B || A&~B);
5 endmodule
```

Fig 6.1 - Testbench and design code for XNOR gate

**Output:**

```
[2022-08-20 13:33:57 EDT] iverilog '-wall' design.sv testbench.sv && unbuffer vvp a.out
VCD info: dumpfile dump.vcd opened for output.
a=0 and b=0
y=1
a=0 and b=1
y=0
a=1 and b=0
y=0
a=1 and b=1
y=1
Finding VCD file...
./dump.vcd
[2022-08-20 13:33:57 EDT] Opening EPWave...
Done
```

Fig 6.2 - Output as truth table of XNOR gate

**Waveform:**

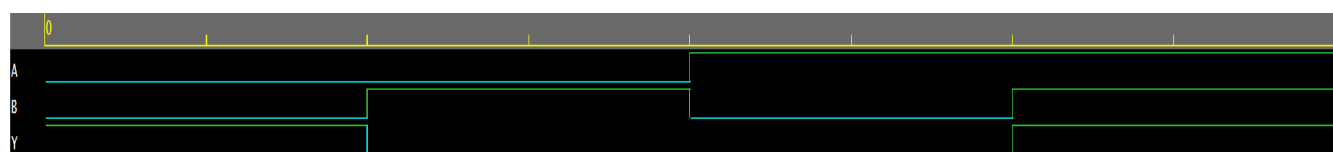


Fig 6.3 - Waveform of XNOR gate output

