Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Aim: Write the Verilog code for

AIM - (a) Design of Basic logic gates (AND,OR,NOT,NOR,NAND,EX-OR,EX NOR)

Theory:

Logic Gates:

A logic gate is a simple switching circuit that determines whether an input pulse can pass through to the output in digital circuits. The building blocks of a digital circuit are logic gates, which execute numerous logical operations that are required by any digital circuit. These can take two or more inputs but only produce one output. The mix of inputs applied across a logic gate determines its output. Logic gates use Boolean algebra to execute logical processes. Logic gates are found in nearly every digital gadget we use on a regular basis. Logic gates are used in the architecture of our telephones, laptops, tablets, and memory devices.

Types of Logic Gates:

A logic gate is a digital gate that allows data to be transferred. Logic gates, use logic to determine whether or not to pass a signal. Logic gates, on the other hand, govern the flow of information based on a set of rules. The following types of logic gates are commonly used:

AND OR NOT NOR NAND XOR XNOR

AND Gate:

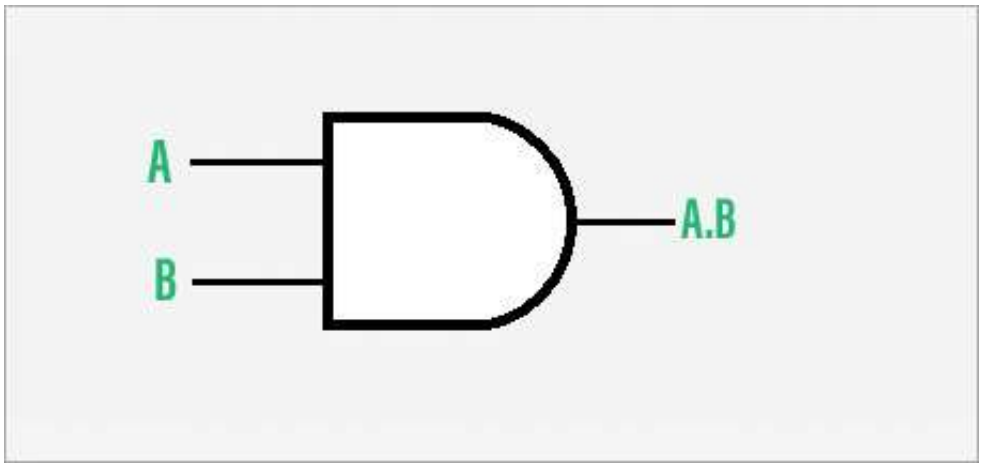
An AND gate has a single output and two or more inputs.

1. When all of the inputs are 1, the output of this gate is 1.
2. The AND gate's Boolean logic is Y=A.B if there are two inputs A and B.

An AND gate's symbol and truth table are as follows:

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Input		Output
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1



Verilog Code:

```
// AND Gate

module and_gate (in1,in2,out);
  input in1;
  input in2;
  output out;
  and(out,in1,in2);
endmodule
```

Test-bench:

```
module and_gate_tb;
  reg in1;
  reg in2;
  wire out;

  and_gate uut(in1,in2,out);

  initial begin
```

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

```
  in1 = 0;
  in2 = 0;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 0;
  in2 = 1;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 1;
  in2 = 0;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 1;
  in2 = 1;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  $finish();
  end
endmodule
```

Results:

```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog and_gate.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
0 - 0 -> 0
0 - 1 -> 0
1 - 0 -> 0
1 - 1 -> 1
and_gate.v:40: $finish called at 40 (1s)
```

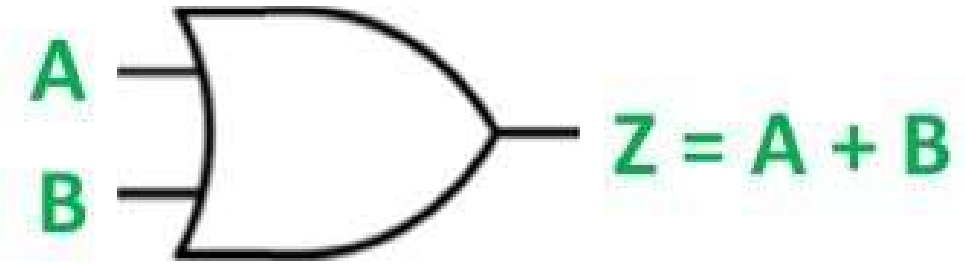
Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

OR Gate:

Two or more inputs and one output can be used in an OR gate.

1. The logic of this gate is that if at least one of the inputs is 1, the output will be 1.
2. The OR gate's output will be given by the following mathematical procedure if there are two inputs A and B: Y=A+B

Input		Output
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1



Verilog Code:

```
// OR Gate

module or_gate (in1,in2,out);
  input in1;
  input in2;
  output out;
  or(out,in1,in2);
endmodule
```

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Test-bench:

```
module or_gate_tb;
 reg in1;
 reg in2;
 wire out;

 or_gate uut(in1,in2,out);

 initial begin
 in1 = 0;
 in2 = 0;
 #10  // Delay of 10 nanoseconds
 $display("%b",in1," - ","%b",in2," -> ","%b",out);

 in1 = 0;
 in2 = 1;
 #10  // Delay of 10 nanoseconds
 $display("%b",in1," - ","%b",in2," -> ","%b",out);

 in1 = 1;
 in2 = 0;
 #10  // Delay of 10 nanoseconds
 $display("%b",in1," - ","%b",in2," -> ","%b",out);

 in1 = 1;
 in2 = 1;
 #10  // Delay of 10 nanoseconds
 $display("%b",in1," - ","%b",in2," -> ","%b",out);

 $finish();
```

```
  end
endmodule
```

Results:

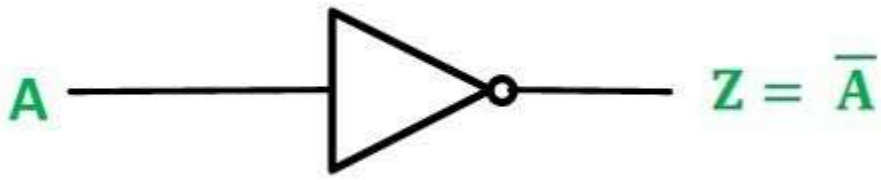
```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog or_gate.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
0 - 0 -> 0
0 - 1 -> 1
1 - 0 -> 1
1 - 1 -> 1
or_gate.v:40: $finish called at 40 (1s)
```

NOT Gate:

The NOT gate is a basic one-input, one-output gate.

1. When the input is 1, the output is 0, and vice versa. A NOT gate is sometimes called an inverter because of its feature.
2. If there is only one input A, the output may be calculated using the Boolean equation Y=A'.

Input	Output
A	Not A
0	1
1	0



Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Verilog Code:

```
// NOT Gate

module not_gate (in,out);
  input in;
  output out;
  not(out,in);
endmodule
```

Test-bench:

```
module not_gate_tb;
  reg in;
  wire out;

  not_gate uut(in,out);

  initial begin
  in = 0;
  #10   // Delay of 10 nanoseconds
  $display("%b",in," -> ","%b",out);

  in = 1;
  #10   // Delay of 10 nanoseconds
  $display("%b",in," -> ","%b",out);
  $finish();
  end
endmodule
```

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Results:

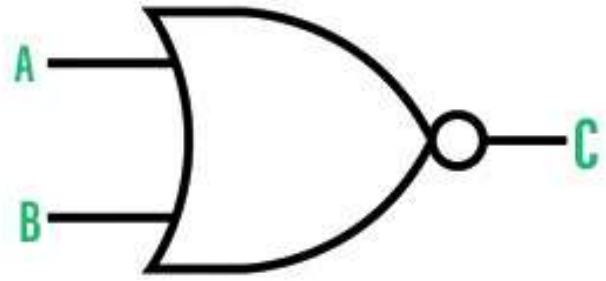
```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog not_gate.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
0 -> 1
1 -> 0
not_gate.v:25: $finish called at 20 (1s)
```

NOR Gate:

A NOR gate, sometimes known as a "NOT-OR" gate, consists of an OR gate followed by a NOT gate.

1. This gate's output is 1 only when all of its inputs are 0. Alternatively, when all of the inputs are low, the output is high.
2. The Boolean statement for the NOR gate is Y=(A+B)' if there are two inputs A and B.

Input		Output
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0



By comparing the truth tables, we can observe that the outputs of the NOR gate are the polar opposite of those of an OR gate. The NOR gate is sometimes known as a universal gate since it may be used to implement the OR, AND, and NOT gates.

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Verilog Code:

```
// NOR Gate

module nor_gate (in1,in2,out);
  input in1;
  input in2;
  output out;
  nor(out,in1,in2);
endmodule
```

Test-bench:

```
module nor_gate_tb;
  reg in1;
  reg in2;
  wire out;

  nor_gate uut(in1,in2,out);

  initial begin
  in1 = 0;
  in2 = 0;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 0;
  in2 = 1;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);
```

```
  in1 = 1;
  in2 = 0;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 1;
  in2 = 1;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  $finish();
  end
endmodule
```

Results:

```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog nor_gate.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
0 - 0 -> 1
0 - 1 -> 0
1 - 0 -> 0
1 - 1 -> 0
nor_gate.v:40: $finish called at 40 (1s)
```

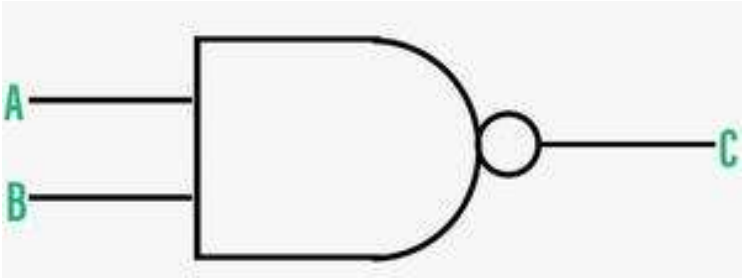
NAND Gate:

A NAND gate, sometimes known as a 'NOT-AND' gate, is essentially a Not gate followed by an AND gate.

1. This gate's output is 0 only if none of the inputs is 0. Alternatively, when all of the inputs are not high and at least one is low, the output is high.
2. If there are two inputs A and B, the Boolean expression for the NAND gate is Y=(A.B)'

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Input		Output
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0



By comparing their truth tables, we can observe that their outputs are the polar opposite of an AND gate. The NAND gate is known as a universal gate because it may be used to implement the AND, OR, and NOT gates.

Verilog Code:

```
// NAND Gate

module nand_gate (in1,in2,out);
  input in1;
  input in2;
  output out;
  nand(out,in1,in2);
endmodule
```

Test-bench:

```
module nand_gate_tb;
  reg in1;
  reg in2;
  wire out;
```

```
  nand_gate uut(in1,in2,out);

  initial begin
  in1 = 0;
  in2 = 0;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 0;
  in2 = 1;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 1;
  in2 = 0;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 1;
  in2 = 1;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  $finish();
  end
endmodule
```

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Results:

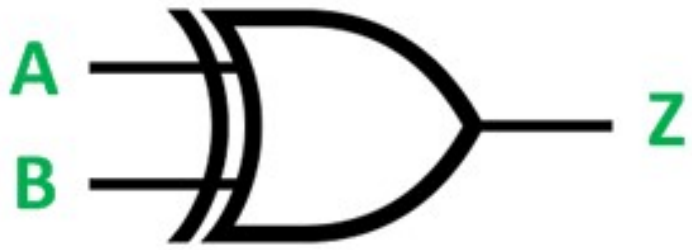
```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog nand_gate.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
0 - 0 -> 1
0 - 1 -> 1
1 - 0 -> 1
1 - 1 -> 0
nand_gate.v:40: $finish called at 40 (1s)
```

XOR Gate:

The Exclusive-OR or 'Ex-OR' gate is a digital logic gate that accepts more than two inputs but only outputs one value.

1. If any of the inputs is 'High,' the output of the XOR Gate is 'High.' If both inputs are 'High,' the output is 'Low.' If both inputs are 'Low,' the output is 'Low.'
2. The Boolean equation for the XOR gate is Y=A'.B+A.B' if there are two inputs A and B.

Input		Output
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



Its outputs are based on OR gate logic, as we can see from the truth table.

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Verilog Code:

```
// XOR Gate

module xor_gate (in1,in2,out);
  input in1;
  input in2;
  output out;
  xor(out,in1,in2);
endmodule
```

Test-bench:

```
module xor_gate_tb;
  reg in1;
  reg in2;
  wire out;

  xor_gate uut(in1,in2,out);

  initial begin
  in1 = 0;
  in2 = 0;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 0;
  in2 = 1;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);
```

```
  in1 = 1;
  in2 = 0;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 1;
  in2 = 1;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  $finish();
  end
endmodule
```

Results:

```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog xor_gate.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
0 - 0 -> 0
0 - 1 -> 1
1 - 0 -> 1
1 - 1 -> 0
xor_gate.v:40: $finish called at 40 (1s)
```

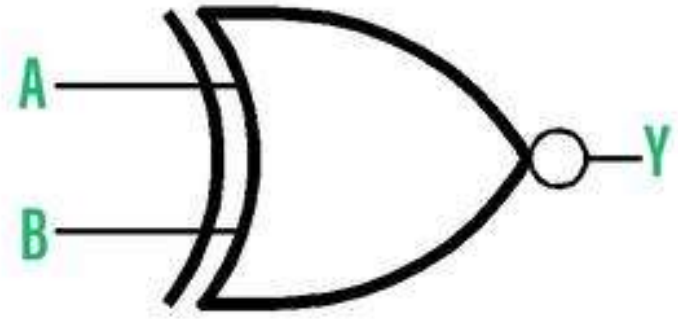
XNOR Gate:

The Exclusive-NOR or 'EX-NOR' gate is a digital logic gate that accepts more than two inputs but only outputs one.

1. If both inputs are 'High,' the output of the XNOR Gate is 'High.' If both inputs are 'Low,' the output is 'High.' If one of the inputs is 'Low,' the output is 'Low.'
2. If there are two inputs A and B, then the XNOR gate's Boolean equation is: Y=A.B+A'B'.

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Input		Output
A	B	A XNOR B
0	0	1
0	1	0
1	0	0
1	1	1



Verilog Code:

```
// XNOR Gate

module xnor_gate (in1,in2,out);
  input in1;
  input in2;
  output out;
  xnor(out,in1,in2);
endmodule
```

Test-bench:

```
module xnor_gate_tb;
  reg in1;
  reg in2;
  wire out;

  xnor_gate uut(in1,in2,out);
```

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

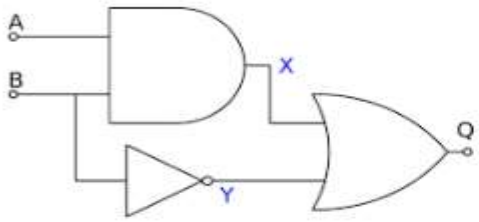
```
  initial begin
  in1 = 0;
  in2 = 0;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 0;
  in2 = 1;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 1;
  in2 = 0;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  in1 = 1;
  in2 = 1;
  #10   // Delay of 10 nanoseconds
  $display("%b",in1," - ","%b",in2," -> ","%b",out);

  $finish();
  end
endmodule
```

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Results:

```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog xnor_gate.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
0 - 0 -> 1
0 - 1 -> 0
1 - 0 -> 0
1 - 1 -> 1
xnor_gate.v:40: $finish called at 40 (1s)
```

AIM - (b) Designing of various logic circuits.



A	B	X	Y	Q
0	0	0	1	1
0	1	0	0	0
1	0	0	1	1
1	1	1	0	1

LOGIC CIRCUIT 1

Verilog Code:

```
// Lab Experiment 2 - Logic Circuit 1

module exp2_lc1(inA, inB, out);
   input inA, inB;
   wire w1, w2;
   output out;

   and(w1, inA, inB);
   not(w2, inB);
   or(out, w1, w2);
endmodule
```

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

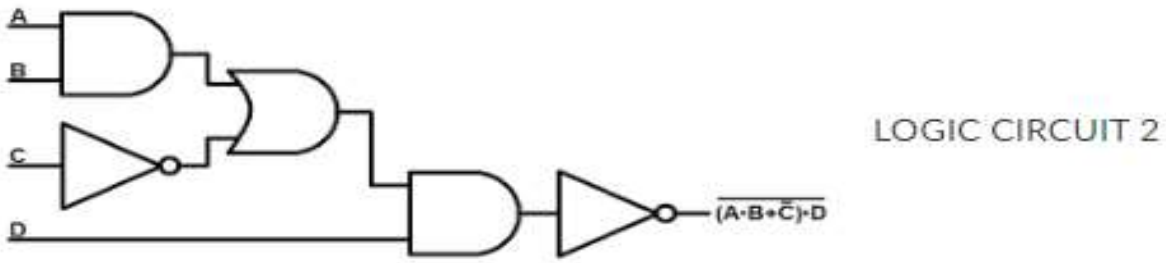
Test-bench:

```
module exp2_lc1_tb;
  reg x1=0, x2=0;
  wire out;
  reg [3:0] inputs; // 3-bit binary counter

  exp2_lc1 a(x2, x1, out);

  initial begin
    for (inputs = 0; inputs < 4; inputs = inputs + 1) begin
       x1 = inputs[0]; // 0th bit for x1
       x2 = inputs[1]; // 1st bit for x2
       #10000; // Delay of 1 second

       // Display inputs and corresponding outputs
       $display("x2 = %b | x1 = %b -> Output : %b", x2, x1, out);
    end
    $finish;
  end
endmodule
```

Results:

```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog exp_2_lc1.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
x2 = 0 | x1 = 0 -> Output : 1
x2 = 0 | x1 = 1 -> Output : 0
x2 = 1 | x1 = 0 -> Output : 1
x2 = 1 | x1 = 1 -> Output : 1
exp_2_lc1.v:32: $finish called at 40000 (1s)
```

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**



LOGIC CIRCUIT 2

Verilog Code:

```
// Lab Experiment 2 - Logic Circuit 2

module exp2_lc2(inA, inB, inC, inD, out);
   input inA, inB, inC, inD;
   wire w1, w2, w3, w4;
   output out;

   and(w1, inA, inB);
   not(w2, inC);
   or(w3, w2, w1);
   and(w4, w3, inD);
   not(out, w4);

endmodule
```

Test-bench:

```
module exp2_lc2_tb;
  reg x1=0, x2=0, x3=0, x4=0;
  wire out;
  reg [4:0] inputs; // 3-bit binary counter
```

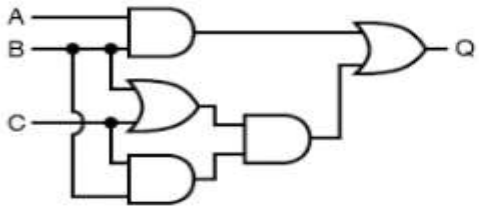
Marwadi University	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

```
  exp2_lc2 a(x1, x2, x3, x4, out);

  initial begin
    for (inputs = 0; inputs < 16; inputs = inputs + 1) begin
      x1 = inputs[0]; // 0th bit for x1
      x2 = inputs[1]; // 1st bit for x2
      x3 = inputs[2]; // 2nd bit for x3
      x4 = inputs[3]; // 3rd bit for x4
      #10000; // Delay of 1 second

      // Display inputs and corresponding outputs
      $display("x4 = %b | x3 = %b | x2 = %b | x1 = %b -> Output : %b", x4, x3, x2, x1, out);
    end
    $finish;
  end
endmodule
```

Results:

```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog exp_2_lc2.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
x4 = 0 | x3 = 0 | x2 = 0 | x1 = 0 -> Output : 1
x4 = 0 | x3 = 0 | x2 = 0 | x1 = 1 -> Output : 1
x4 = 0 | x3 = 0 | x2 = 1 | x1 = 0 -> Output : 1
x4 = 0 | x3 = 0 | x2 = 1 | x1 = 1 -> Output : 1
x4 = 0 | x3 = 1 | x2 = 0 | x1 = 0 -> Output : 1
x4 = 0 | x3 = 1 | x2 = 0 | x1 = 1 -> Output : 1
x4 = 0 | x3 = 1 | x2 = 1 | x1 = 0 -> Output : 1
x4 = 0 | x3 = 1 | x2 = 1 | x1 = 1 -> Output : 1
x4 = 1 | x3 = 0 | x2 = 0 | x1 = 0 -> Output : 0
x4 = 1 | x3 = 0 | x2 = 0 | x1 = 1 -> Output : 0
x4 = 1 | x3 = 0 | x2 = 1 | x1 = 0 -> Output : 0
x4 = 1 | x3 = 0 | x2 = 1 | x1 = 1 -> Output : 0
x4 = 1 | x3 = 1 | x2 = 0 | x1 = 0 -> Output : 1
x4 = 1 | x3 = 1 | x2 = 0 | x1 = 1 -> Output : 1
x4 = 1 | x3 = 1 | x2 = 1 | x1 = 0 -> Output : 1
x4 = 1 | x3 = 1 | x2 = 1 | x1 = 1 -> Output : 0
exp_2_lc2.v:37: $finish called at 160000 (1s)
```

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**



LOGIC CIRCUIT 3

Verilog Code:

```
// Lab Experiment 2 - Logic Circuit 3

module exp2_lc3(inA, inB, inC, out);
   input inA, inB, inC;
   wire w1, w2, w3, w4;
   output out;

   and(w1, inA, inB);
   or(w2, inB, inC);
   and(w3, inB, inC);
   and(w4, w3, w2);
   or(out, w4, w1);

endmodule
```

Test-bench:

```
module exp2_lc3_tb;
  reg x1=0, x2=0, x3=0;
  wire out;
  reg [3:0] inputs; // 3-bit binary counter

  exp2_lc3 a(x1, x2, x3, out);
```

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

```
 initial begin
  for (inputs = 0; inputs < 8; inputs = inputs + 1) begin
     x1 = inputs[0]; // 0th bit for x1
     x2 = inputs[1]; // 1st bit for x2
     x3 = inputs[2]; // 2nd bit for x3
     #10000; // Delay of 1 second

     // Display inputs and corresponding outputs
     $display("x3 = %b | x2 = %b | x1 = %b -> Output : %b", x3, x2, x1, out);
  end
  $finish;
 end
endmodule
```

Results:

```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog exp_2_lc3.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
x3 = 0 | x2 = 0 | x1 = 0 -> Output : 0
x3 = 0 | x2 = 0 | x1 = 1 -> Output : 0
x3 = 0 | x2 = 1 | x1 = 0 -> Output : 0
x3 = 0 | x2 = 1 | x1 = 1 -> Output : 1
x3 = 1 | x2 = 0 | x1 = 0 -> Output : 0
x3 = 1 | x2 = 0 | x1 = 1 -> Output : 0
x3 = 1 | x2 = 1 | x1 = 0 -> Output : 1
x3 = 1 | x2 = 1 | x1 = 1 -> Output : 1
exp_2_lc3.v:36: $finish called at 80000 (1s)
```

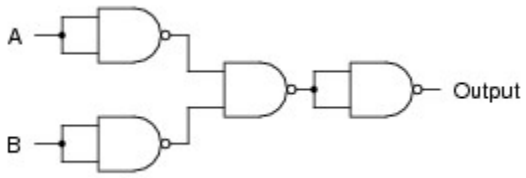
Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Pre Lab-Exercise:

a. What is the difference between Combinational and Sequential circuits?

b. How can you implement and/or/not gate using NAND/NOR?

c. Simplify this logic gate circuit, which uses nothing but NAND gates to accomplish a certain logic function:



Also, Mention Logic function and Truth table for the same.

Verilog Code:

```
// Lab Experiment 2 - Pre Lab Circuit

module exp2_prelab(inA, inB, out);
  input inA, inB;
  wire w1, w2, w3;
```

```
    output out;

    nand(w1, inA, inA);
    nand(w2, inB, inB);
    nand(w3, w2, w1);
    nand(out, w3, w3);

endmodule
```

Test-bench:

```
module exp2_prelab_tb;
  reg x1=0, x2=0;
  wire out;
  reg [2:0] inputs; // 3-bit binary counter

  exp2_prelab a(x1, x2, out);

  initial begin
    for (inputs = 0; inputs < 4; inputs = inputs + 1) begin
       x1 = inputs[0]; // 0th bit for x1
       x2 = inputs[1]; // 1st bit for x2
       #10000; // Delay of 1 second

       // Display inputs and corresponding outputs
       $display("x2 = %b | x1 = %b -> Output : %b", x2, x1, out);
    end
    $finish;
  end
endmodule
```

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Results:

```
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> iverilog exp_2_prelab.v
PS D:\Mirror\ICT\3rd YEAR\SEM 6\DDV_Codes> vvp a.out
x2 = 0 | x1 = 0 -> Output : 1
x2 = 0 | x1 = 1 -> Output : 0
x2 = 1 | x1 = 0 -> Output : 0
x2 = 1 | x1 = 1 -> Output : 0
exp_2_prelab.v:34: $finish called at 40000 (1s)
```

Observation and Result Analysis:

Write the final observation and Analysis

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

Post Lab Exercise:

a. An automobile manufacturer needs a logic circuit to perform a specific task in its new line of cars. These cars will be equipped with a "headlight left on" alarm that sounds any time these two conditions are met: headlights on and ignition switch off. Draw the schematic diagram of a logic gate circuit that will implement this alarm, constructed entirely out of NAND gates. Also write Verilog with data flow modeling style for the same.

Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Design using Verilog (01CT0619)	**Aim:** Write the Verilog code for, (a) Design of Basic logic gates(AND,OR,NOT,NOR,NAND,EX-OR,EX NOR) (b) Designing of various logic circuits.	
Experiment No: 02	**Date:**	**Enrollment No: 92100133020**

b. Write the Boolean expression for this logic gate circuit, then reduce that expression to its simplest form using any applicable Boolean laws and theorems. Finally, draw a new gate circuit diagram based on the simplified Boolean expression that performs the exact same logic function. Also, Write Verilog code in behavioral modeling style for both the logic circuit,
Note: Output must be same for both.