

<b>Course Name:</b>	<b>Digital Design Laboratory</b>	<b>Semester:</b>	<b>III</b>
<b>Date of Performance:</b>	<b>19 / 07 / 2024</b>	<b>Batch No:</b>	<b>A1</b>
<b>Faculty Name:</b>		<b>Roll No:</b>	<b>16010123012</b>
<b>Faculty Sign &amp; Date:</b>		<b>Grade/Marks:</b>	<b>___/25</b>

**Experiment No: 1**  
**Title: Study of Basic Gates and Universal Gates**

<b>Aim and Objective of the Experiment:</b>
Understand Basic Logic Gates and Universal Gates

<b>COs to be achieved:</b>
<b>CO1:</b> Recall basic gates & logic families and binary, octal & hexadecimal calculations and conversions.

<b>Tools used:</b>
Trainer kits

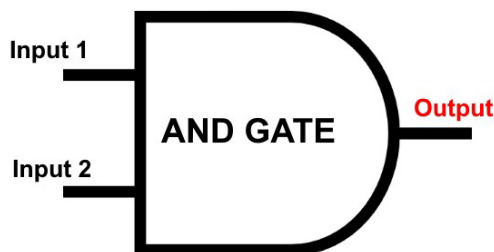
<b>Theory:</b>
<p>Logic gates are electronic circuits that perform logical operations on one or more input signals to produce an output signal based on a set of logical rules. Logic gates can be classified into the following categories:</p> <ol style="list-style-type: none"> <li>Basic Gates: <ol style="list-style-type: none"> <li>AND Gate: The AND gate produces a high output (1) only when all of its inputs are high (1).</li> <li>OR Gate: The OR gate produces a high output (1) if any of its inputs is high (1).</li> <li>NOT Gate (Inverter): The NOT gate produces the logical complement of its input. It takes a single input and produces the opposite value as the output.</li> </ol> </li> <li>Derived Gates: <ol style="list-style-type: none"> <li>NAND Gate: The NAND gate is a combination of an AND gate followed by a NOT gate. It produces the inverse of the AND gate's output. It outputs a low (0) only when all of its inputs are high (1).</li> <li>NOR Gate: The NOR gate is a combination of an OR gate followed by a NOT gate. It produces the inverse of the OR gate's output. It outputs a high (1) only when all of its inputs are low (0).</li> <li>XOR Gate (Exclusive OR): The XOR gate produces a high output (1) when the number of high inputs is odd. It outputs a low (0) when the number of high inputs is even.</li> <li>XNOR Gate (Exclusive NOR): The XNOR gate produces a high output (1) when the number of high inputs is even. It outputs a low (0) when the number of high inputs is odd.</li> </ol> </li> <li>Universal Gates: <p>NAND and NOR gates are considered universal gates because any logic function can be implemented</p> </li> </ol>

using only NAND gates or only NOR gates. This means that with a sufficient number of NAND or NOR gates, you can create circuits that can perform any logical operation.

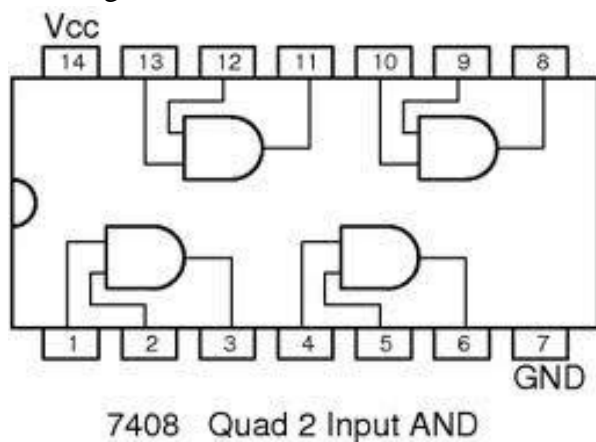
### Implementation Details

1. AND Gate:  $Y = A.B$

Symbol -



Pin Diagram -

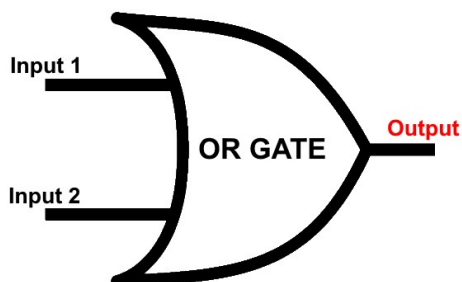


Truth Table -

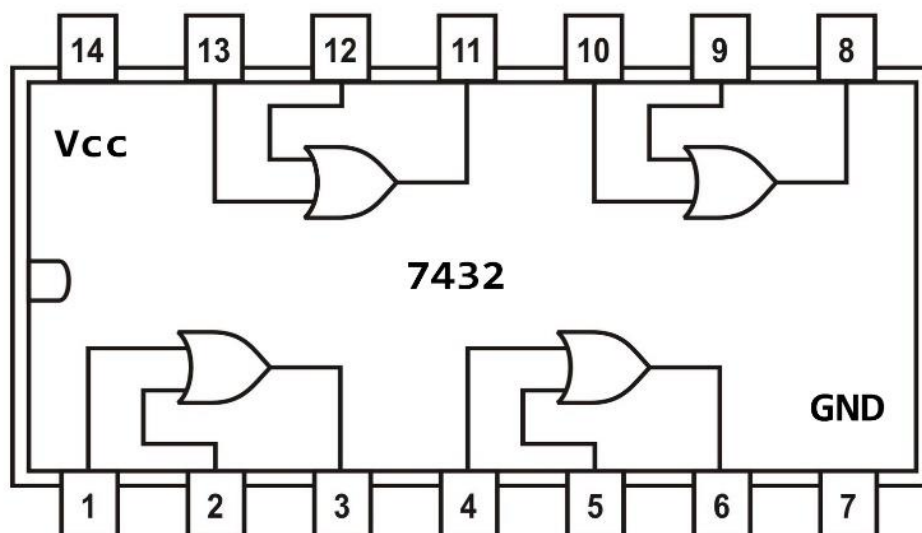
Input		Output
A	B	$Y = A.B$
0	0	0
0	1	0
1	0	0
1	1	1

## 2. OR Gate: $Y = A + B$

Symbol –



Pin Diagram –

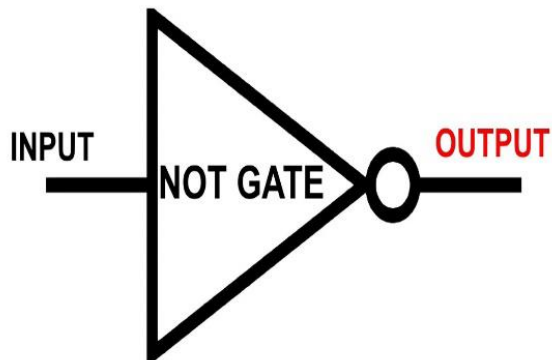


Truth Table –

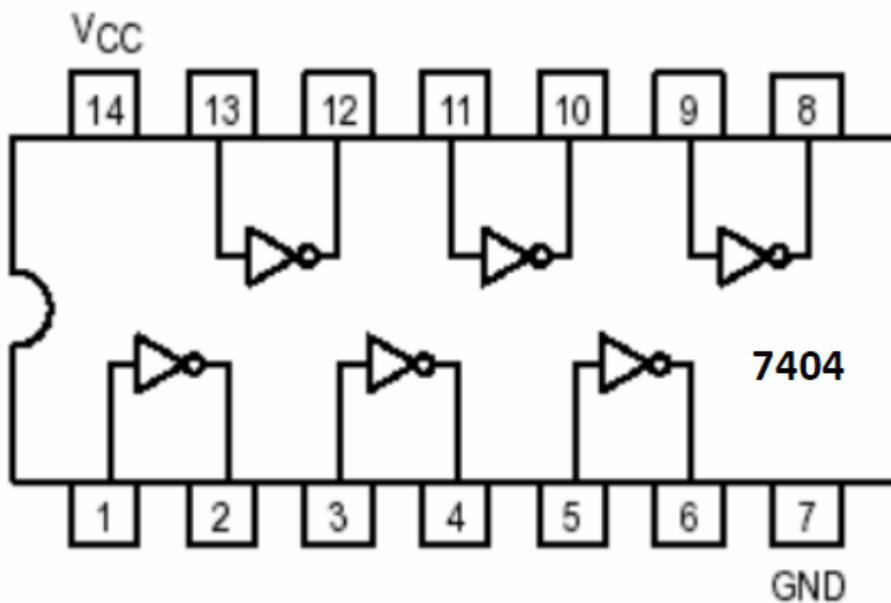
Input		Output
A	B	$Y=A+B$
0	0	0
0	1	1
1	0	1
1	1	1

### 3. NOT Gate: $Y = \bar{A}$

Symbol -



Pin Diagram -

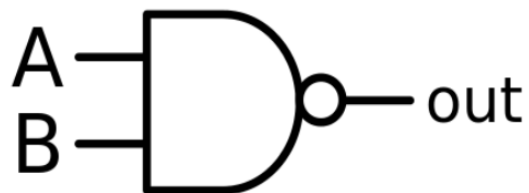


Truth Table -

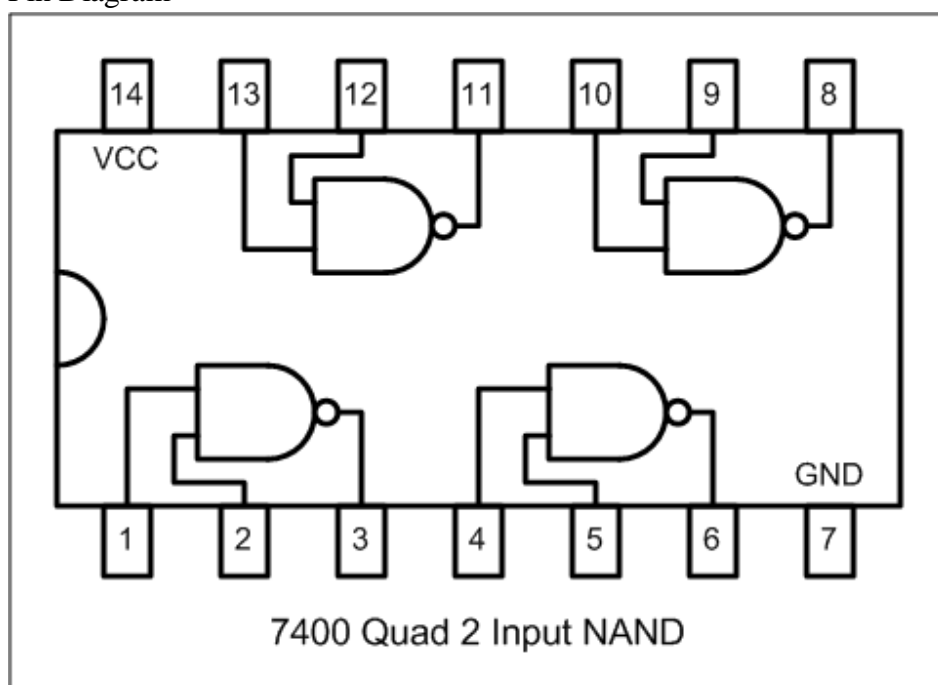
Input	Output
A	Y
0	1
1	0

#### 4. NAND Gate: $Y = A.B$

Symbol –



Pin Diagram –

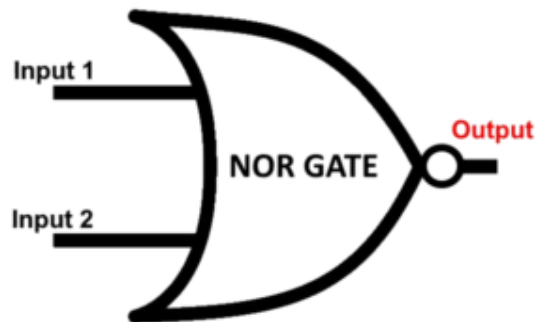


Truth Table-

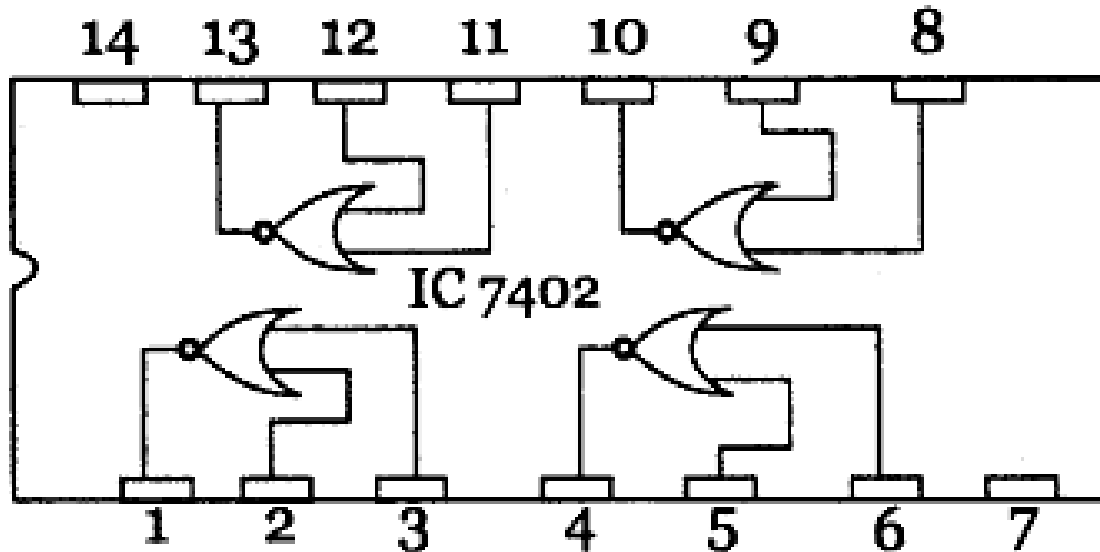
Input		Output
A	B	$Y = \overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0

### 5. NOR Gate: $Y = A + B$

Symbol –



Pin Diagram –



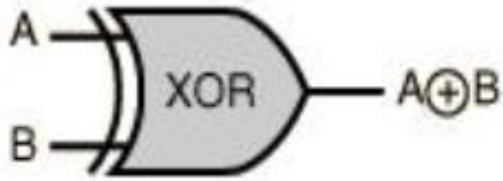
Truth Table –

Input		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

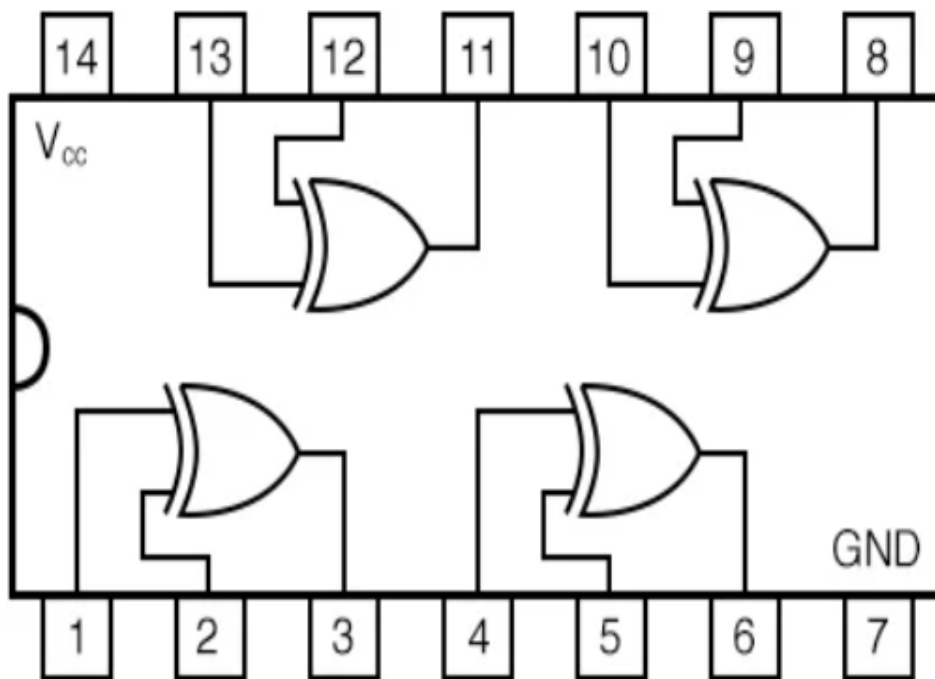
---

6. XOR Gate:  $Y = A \oplus B$

Symbol –



### Pin Diagram –

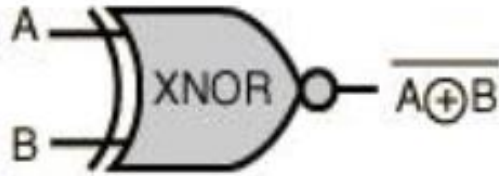


Truth Table –

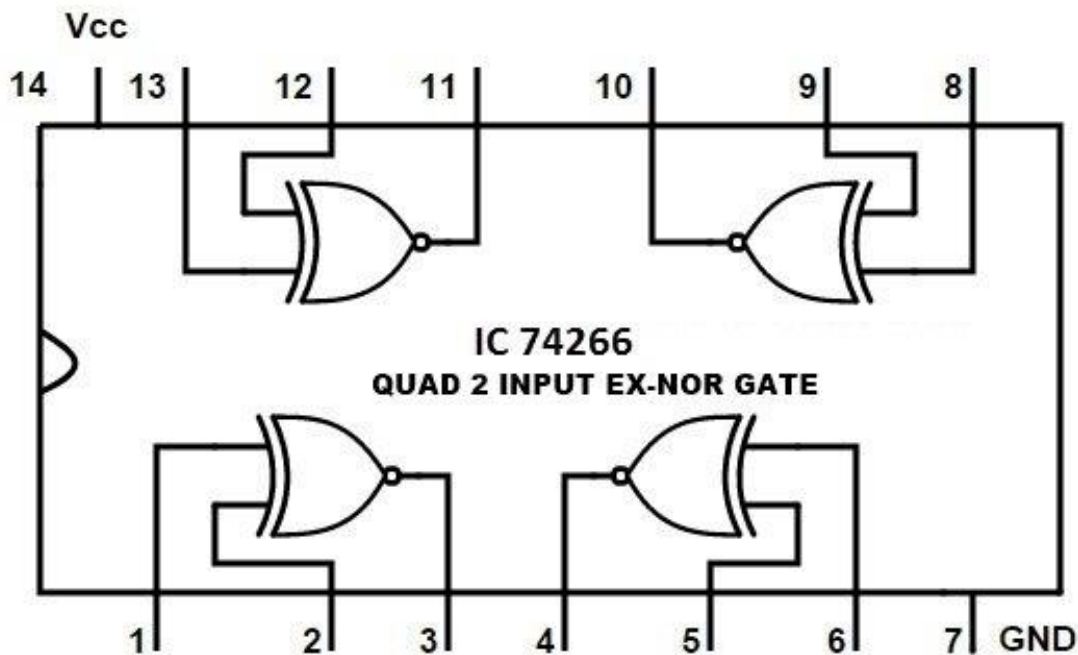
Inputs		Output
A	B	$Y=A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

7. XNOR Gate:  $Y = \overline{A \oplus B}$

Symbol –



Pin Diagram –



Truth Table –

Inputs		Output
A	B	$Y = \overline{A \oplus B}$
0	0	1
0	1	0
1	0	0
1	1	1

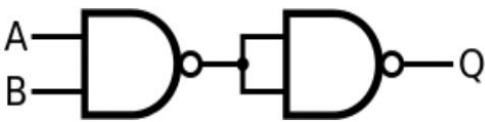


**Implementation Using NAND Gate -  
NOT GATE –**



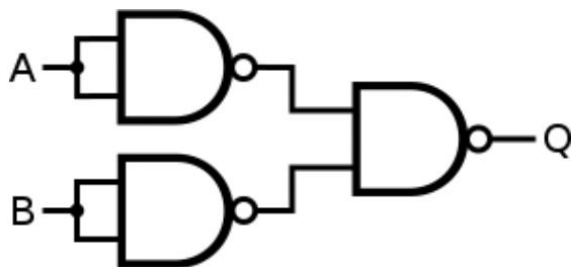
A	Q
0	1
1	0

**AND GATE –**



A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

**OR GATE –**



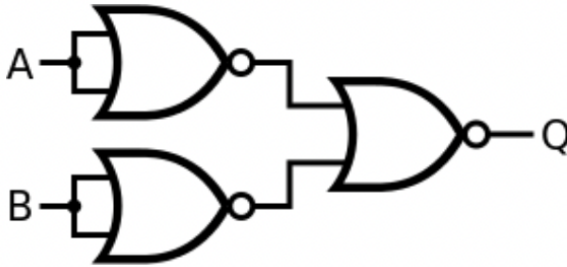
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

**Implementation Using NOR Gate-  
NOT GATE –**



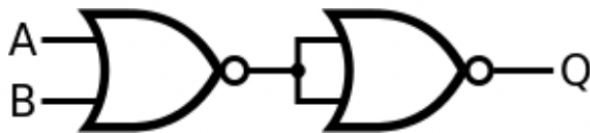
A	Q
0	1
1	0

### AND GATE –



A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

### OR GATE –



A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

### Lab Work:

DDL

CLASSMATE  
Date \_\_\_\_\_  
Page \_\_\_\_\_

- NAND Gate as universal logic element  
NAND Gate as an inverter gate

$x \rightarrow \text{NAND} \rightarrow z = \overline{x}$   
 $x \cdot x = x$

X	Z
0	1
1	0

} Equivalent to inverter.

- NAND Gate as an AND gate

$x, y \rightarrow \text{NAND} \rightarrow \text{Inverter} \rightarrow z = \overline{\overline{x \cdot y}} = x \cdot y$

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

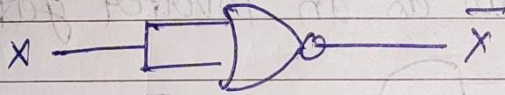
AND Gate

- NAND Gate as an OR gate

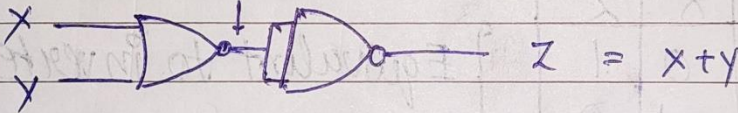
$x, y \rightarrow \text{NAND} \rightarrow \text{NAND} \rightarrow z = \overline{\overline{x} \cdot \overline{y}} = \overline{\overline{x} \cdot \overline{y}} = x + y$

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

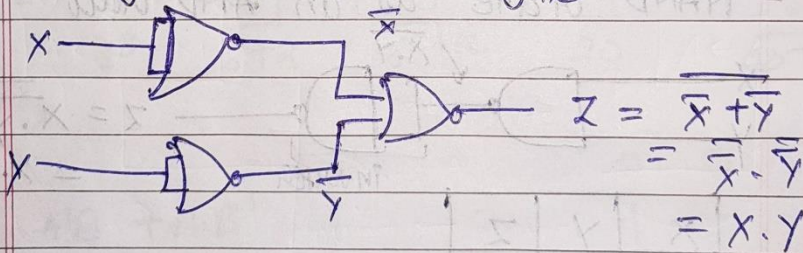
- NOR gate as an Inverter.



- NOR gate as an OR gate



- NOR gate as an AND gate

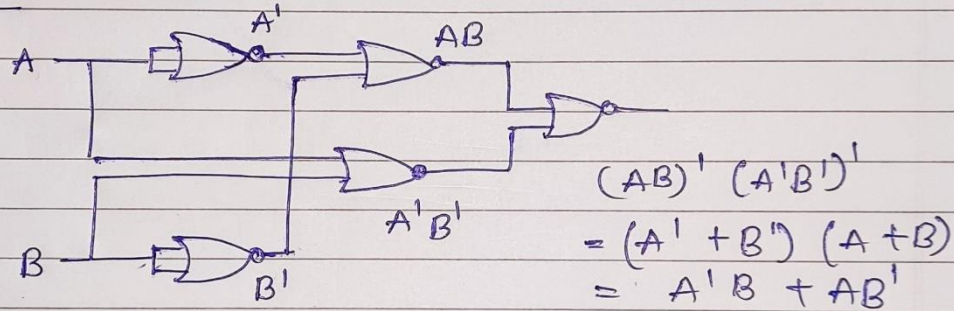


**Post Lab Subjective/Objective type Questions:**

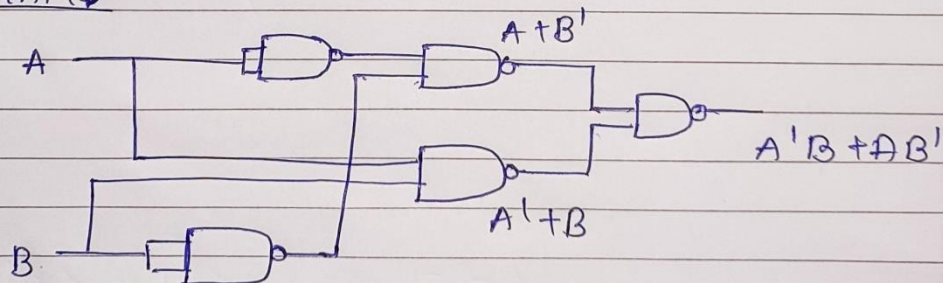
1. Implement the Boolean function using NAND gates and NOR gates  $F = A'B + AB'$

Q1.  $F = A'B + AB'$

→ NOR

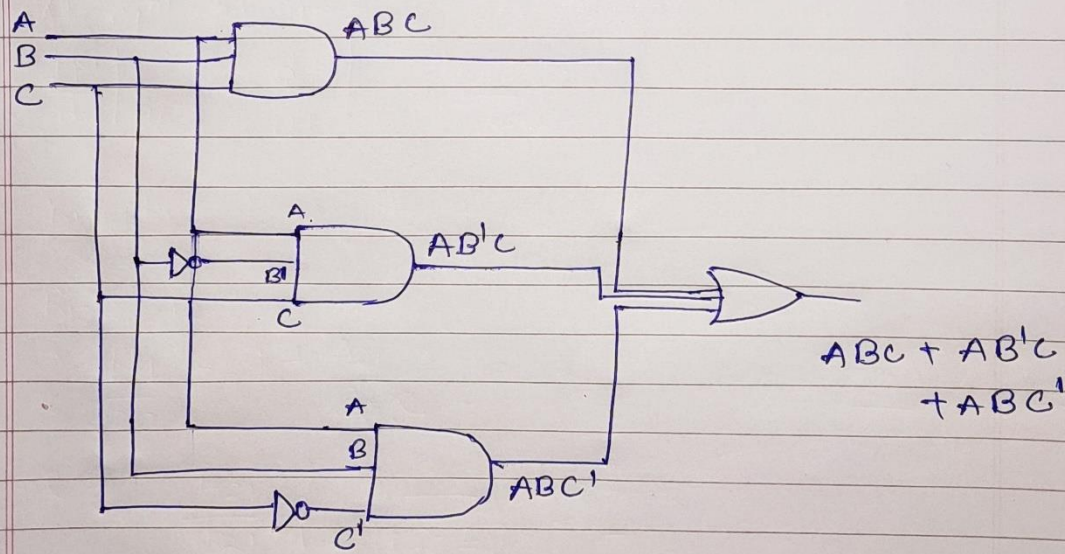


NAND



2. Implement using combination of gates  $F = ABC + AB'C + ABC'$

Q2.  $F = ABC + AB'C + ABC'$







**SOMAIYA**  
VIDYAVIHAR UNIVERSITY

K J Somaiya College of Engineering

**K. J. Somaiya College of Engineering, Mumbai-77**

(A Constituent College of Somaiya Vidyavihar University)

**Department of Computer Engineering**



**Conclusion:**

We studied about logic gates and their combinations, including basic gates and universal gates. We learned how to use these logic gates and implemented them in integrated circuits and learned how to create logic gates using two basic gates, NAND and NOR.

**Signature of faculty in-charge with Date:**