Bahria University,

Karachi Campus

LAB EXPERIMENT NO.

08

LIST OF TASKS

TASK NO	OBJECTIVE
1.	Implementation of Circuit and Truth Table
2.	How do these gate treat when inputs are unconnected (Low or High)? Briefly explain why?
3	Do swapping A and B make any difference?
4	What is difference between an Inverter and a NOT gate?
5	How can you make an inverter using a NOR gate? And using NAND gate?
6	Implement XOR gate function using basic gates?
7	Implement XNOR gate function using basic gates?

Submitted On:

Date: 17/12/2025

Task No. 01:

"Implementation of circuit and Truth Table"

Solution:

A	B	(A.B)'=A'+B'	(A+B)'=A'.B'	A.B	A+B	A⊕B	(A⊕B)'
0	0	1	1	0	0	0	1
0	1	1	0	0	1	1	0
1	0	1	0	0	1	1	0
1	1	0	0	1	1	0	1
0	X	1	0	0	1	1	0
1	X	1	0	0	1	1	0
X	0	1	0	0	1	1	0
X	1	1	0	0	1	1	0
X	X	1	0	0	1	0	1

Task No. 02:

"How do these gate treat when inputs are unconnected (Low or High)? Briefly explain why?"

Solution:

When an input pin on a 74LS TTL integrated circuit is left unconnected (floating), it is reliably interpreted by the gate as a Logic HIGH (1) input. This behavior is by design and is a crucial characteristic of TTL technology.

Why It Happens:

- **Internal "Pull-Up":** Every TTL input pin is connected internally to a little circuit that tries to pull its voltage up towards the main power supply vcc. This is done using an internal biasing network that includes a resistor.
- **Where the Current Goes:**
  - To get a low input, you have to connect the pin to ground. This gives the internal current a path to flow out of the pin and into the ground, pulling the input low.
  - To get a high input, you connect the pin to vcc, or you simply leave it floating.
- **The Floating Result:** When the pin is floating, there is no path for the internal current to flow out. Because this current can't escape, the internal biasing circuit successfully pulls the input voltage all the way up to the HIGH logic level.
- **Undefined**: This means the input is NOT left hanging or undefined (like it would be in a basic theoretical model or a CMOS chip). The TTL design forces the floating pin into a stable high state.

Task No. 03:

"Do swapping A and B make any difference?"

Solution:

Logic gates such as **'AND, OR, NAND, NOR, XOR, XNOR'** are commutative, meaning the order of inputs doesn't matter. So if we swap A and B, the outputs will remain the same.

Task No. 04:

"What is difference between an Inverter and a NOT gate?"

Solution:

There is no difference in function between an inverter and a NOT gate. Hence, a NOT gate is also called an inverter because it inverts the state.

Task No. 05:

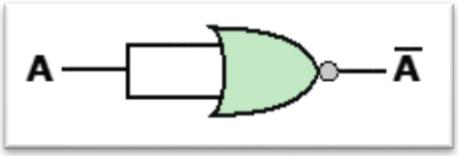
"How can you make an inverter using a NOR gate? And using NAND gate?"

Solution:

We can make an invertor using a NOR gate as well as a NAND gate…

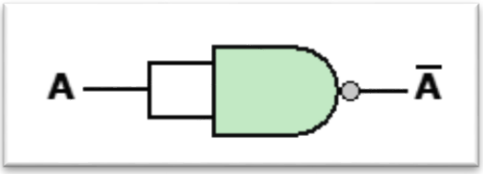
1. **Using a NOR gate:**
   We have a NOR gate logic that is $Y=(A+B)'$ If you connect both inputs together, it becomes

   $$Y = (A+A)'=A'$$



2. **Using a NAND gate:**
   We have a NAND gate logic that is $Y=(A.B)'$ If you connect both inputs together, it becomes

   $$Y = (A.A)'=A'$$

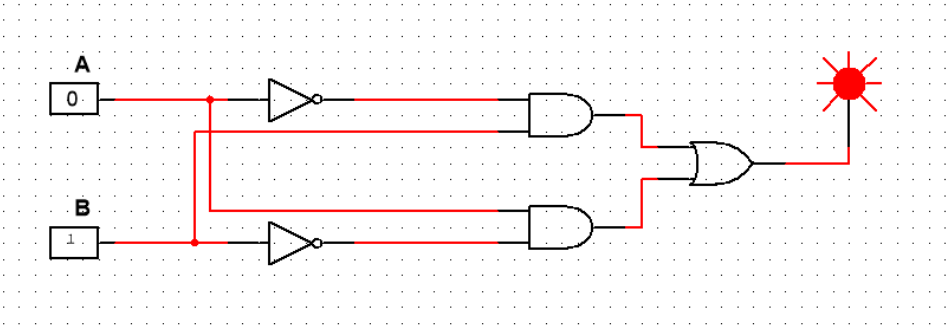


Task No. 06:

"Implement XOR gate function using basic gates?"

Solution:

$$A \oplus B = (A' \cdot B) + (A \cdot B')$$



Task No. 07:

"Implement XNOR gate function using basic gates?"

Solution:

$$A \odot B = (A \cdot B) + (A' \cdot B')$$