The University of Azad Jammu & Kashmir, Muzaffarabad

Computer Architecture & Logic Design

Lab 3: NAND & NOR Gate Construction and Verification using AND, OR & NOT Gates

Student Name:	Shahzad Ahmed Awan
Roll No:	2024-SE-15
Course Title:	Computer Architecture & Logic Design
Course Code:	CS-1205
Instructor:	Engr. Sidra Rafique
Submission Date:	03-June -2025

Department of Software Engineering

Verification of NAND gate through AND & NOT

Objective:

To construct a **NAND gate using an AND gate and a NOT gate**, and verify its output by comparing it to a standard NAND gate IC.

Practical Lab Work Summary:

The lab was performed using a **logic trainer board** (electronic setup with built-in **LEDs**, **power supply**, and **ground terminals**). The setup included:

- **One AND gate IC (Quad 2-input AND - 7408)**
- **One NOT gate IC (Hex Inverter - 7404)**
- **One NAND gate IC (Quad 2-input NAND - 7400)**

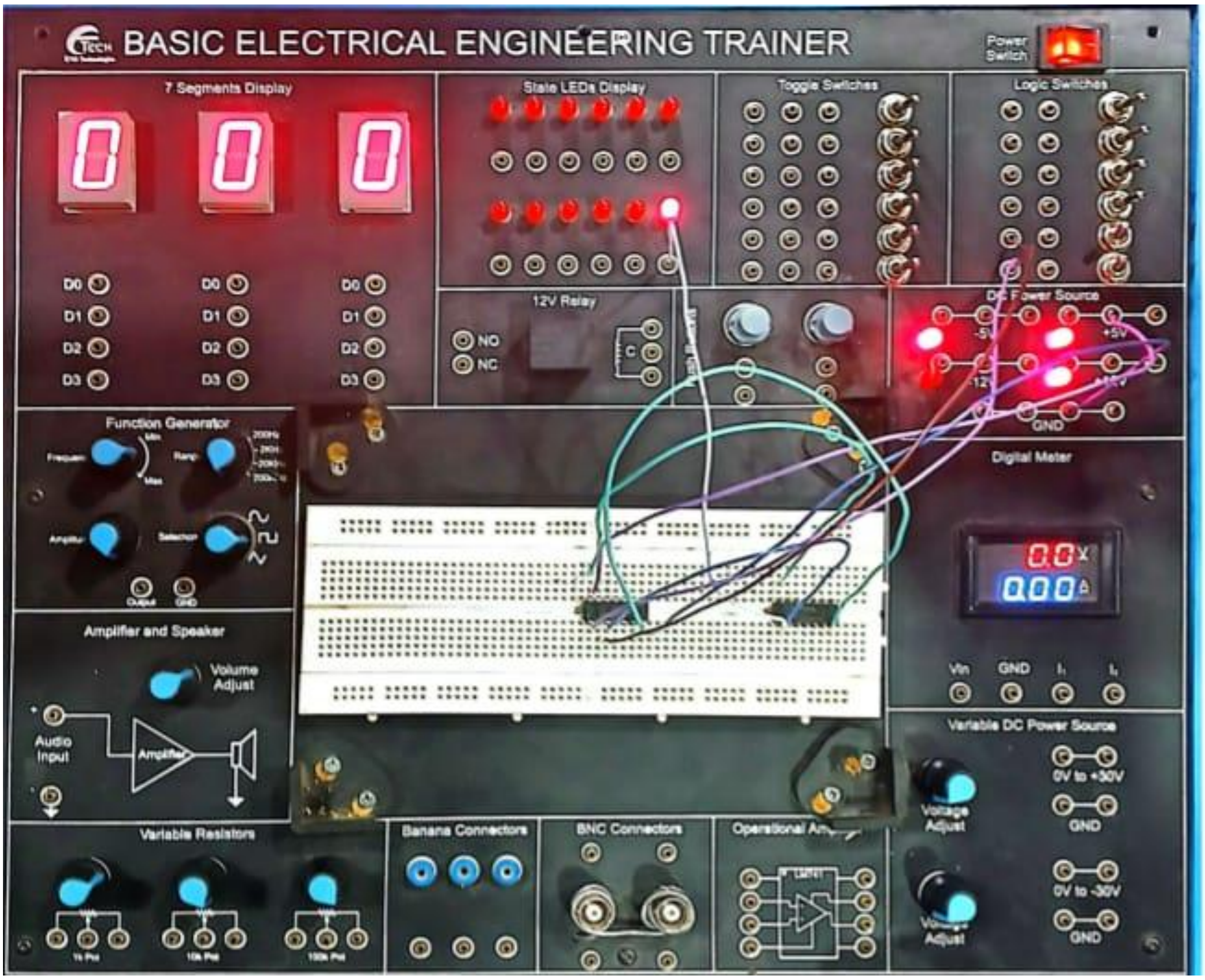


Figure 1: LAb Implementation

The **NAND gate was built** by connecting the output of the AND gate into the input of the NOT gate. Wires and switches were used for input toggling. **Probes or LED indicators** were connected at the NOT gate's output terminal to observe logic changes.

After building the NAND function manually, the same inputs were then connected to the standard NAND gate IC, and the outputs were verified against each other. All **four input combinations (00, 01, 10, 11)** were applied, and the outputs were compared to ensure that the manually built NAND and the actual NAND gate IC behaved identically. A **truth table** was prepared to confirm the logic equivalency.

Lab #3: Procedure – Virtual Implementation in EWB

NAND Using AND + NOT Gate – EWB Procedure:

1. **EWB software was launched**, and a new circuit file was created.
2. From the **component library**, the following elements were added to the workspace:
    - One **2-input AND gate**
    - One **NOT gate**
    - Two **logic switches** (for binary input A and B)
    - One **VCC (power source)**
    - One **Ground (GND)**
    - One **logic indicator** (probe or LED) to monitor output
3. The **input switches** were connected to the two inputs of the AND gate.
4. The **output of the AND gate** was routed directly into the **input of the NOT gate** to

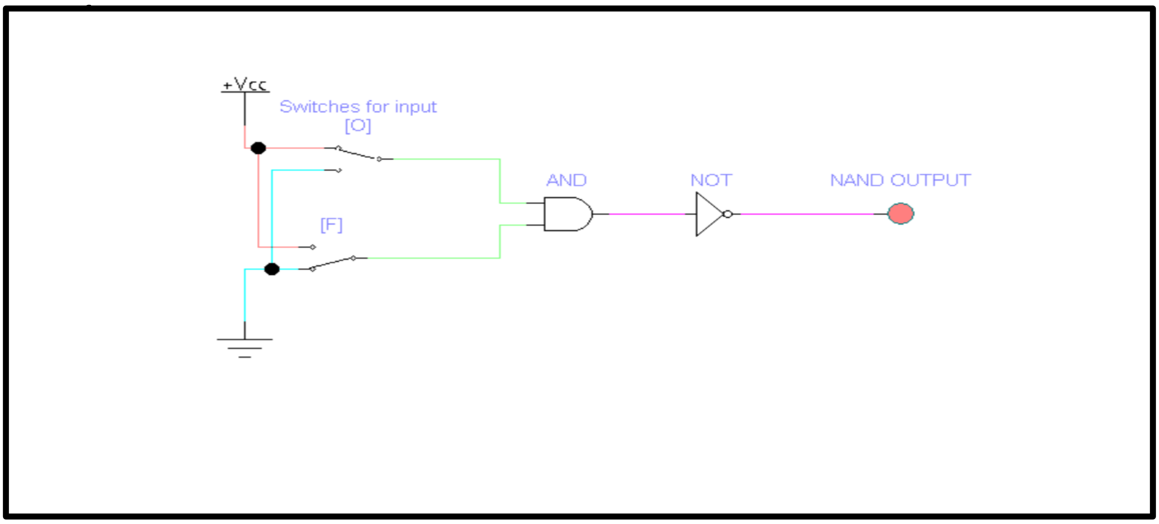


Figure 2: Implementation through Gates

5. The **output of the NOT gate** was connected to the **output probe/LED** to monitor the result of the NAND function.
6. **VCC and GND terminals** were connected appropriately to each logic gate for power supply.
7. The **simulation was started**, and each of the **four input combinations** (00, 01, 10, 11) was applied using the switches.

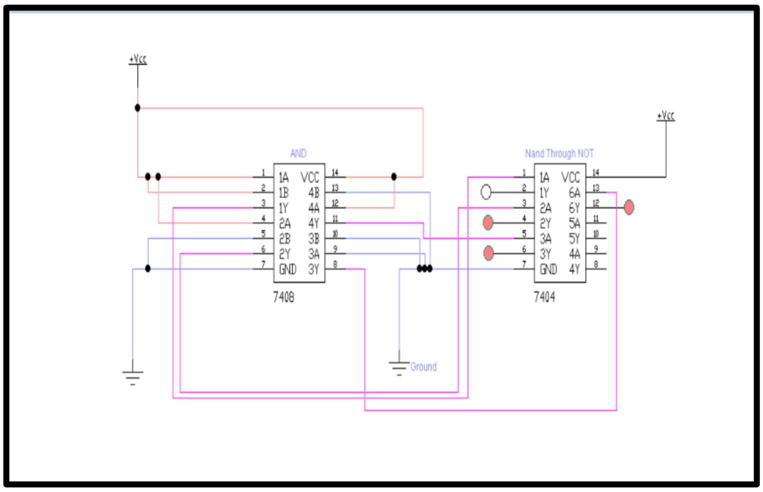
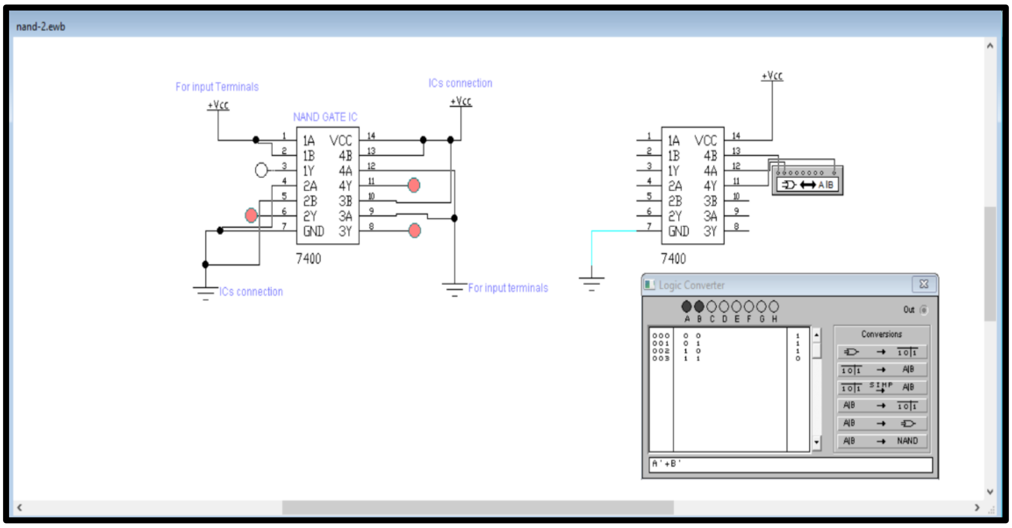


Figure 3: Implementation Through ICs

8. The **output was observed at the NOT gate's output**, confirming that the constructed logic matched a NAND gate's behavior.

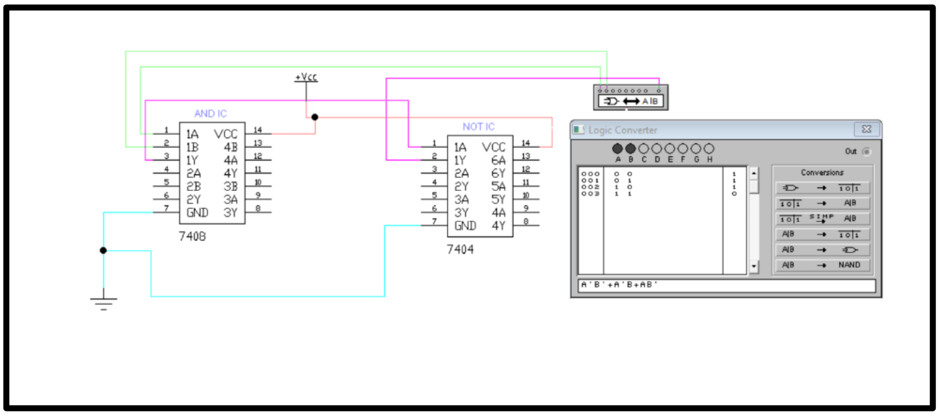


A'+B'

Figure 4: Verification through actual NAND

NAND Gate Truth Table (Verification):

Truth Table in EWB:



```
000  0 0   1
001  0 1   1
002  1 0   1
003  1 1   0
```

```
A'B'+A'B+AB'
```

Input A	Input B	AND Output	NOT Output (NAND)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

NOR Lab Implementation:

The lab was performed using a logic trainer board (electronic setup with built-in LEDs, power supply, and ground terminals). The setup included:

- One OR gate IC (Quad 2-input OR - 7432)
- One NOT gate IC (Hex Inverter - 7404)
- One NOR gate IC (Quad 2-input NOR - 7402)

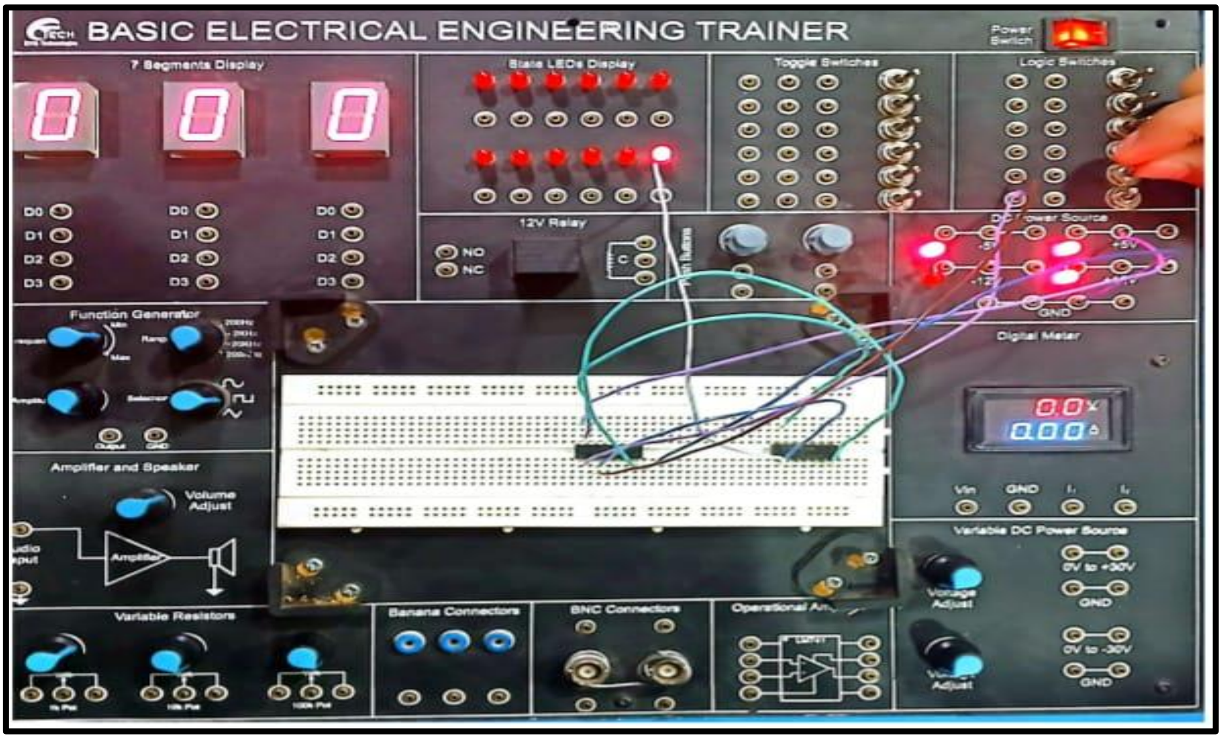


Figure 5: LAb Implementation

The NOR gate was built by connecting the output of the OR gate into the input of the NOT gate. Wires and switches were used for input toggling. Probes or LED indicators were connected at the NOT gate's output terminal to observe logic changes.

After building the NOR function manually, the same inputs were then connected to the standard NOR gate IC, and the outputs were verified against each other. All four input combinations (00, 01, 10, 11) were applied, and the outputs were compared to ensure that the manually built NOR and the actual NOR gate IC behaved identically. A truth table was prepared to confirm the logic equivalency.

Procedure – Virtual Implementation in EWB

NOR Using OR + NOT Gate – EWB Procedure:

- **EWB software** was launched, and a new circuit file was created.

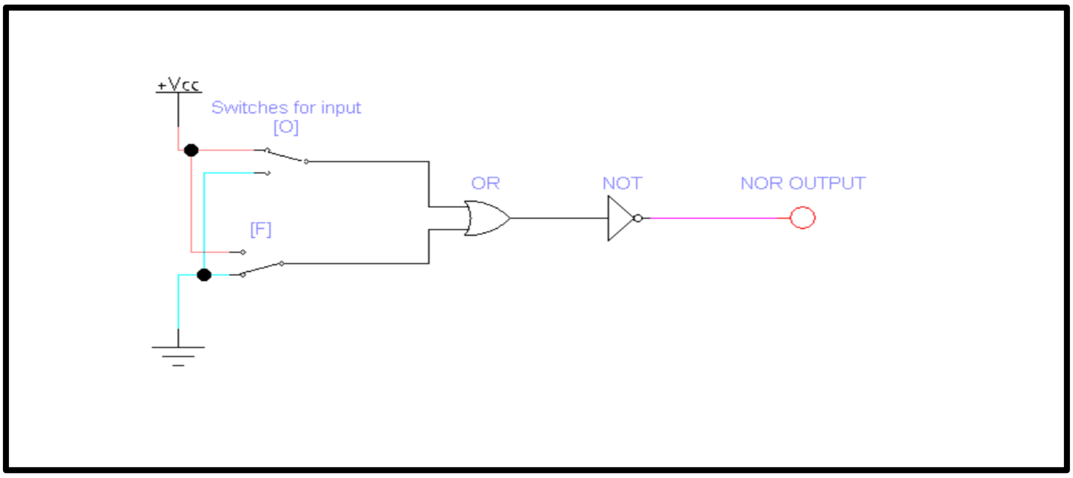


Figure 6: NOR Circuit using OR & NOT

- From the **component library**, the following elements were added to the workspace:

o One 2-input OR gate

o One NOT gate

o Two logic switches (for binary input A and B)

o One VCC (power source)

o One Ground (GND)

o One logic indicator (probe or LED) to monitor outputThe input switches were connected to the two inputs of the OR gate.

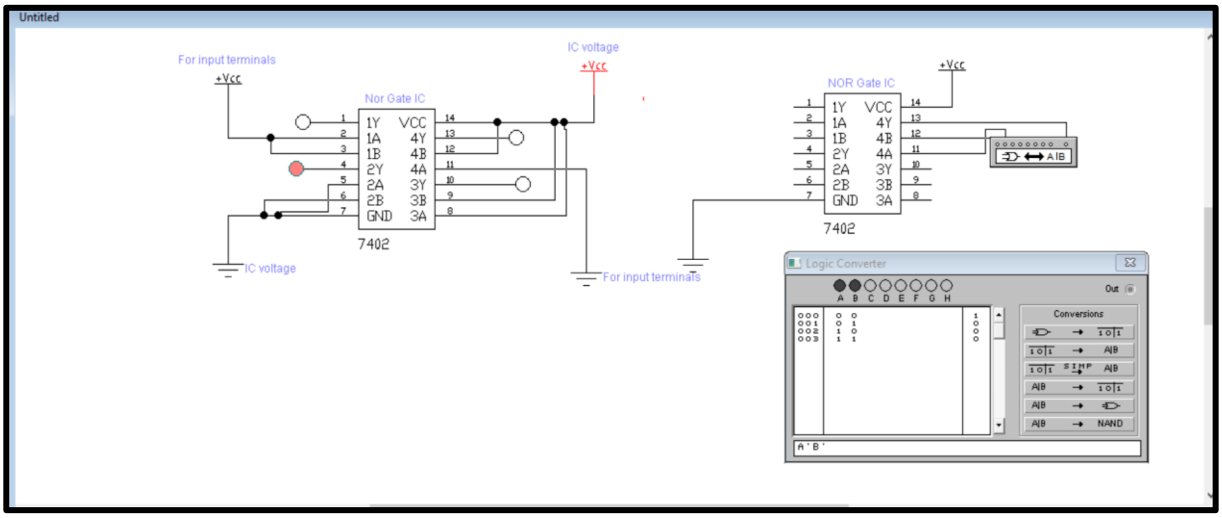


Figure 7: Simple NOR ICs Behavior

- The output of the OR gate was routed directly into the input of the NOT gate to perform inversion.
- The output of the NOT gate was connected to the output probe/LED to monitor the result of the NOR function.
- VCC and GND terminals were connected appropriately to each logic gate for power supply.

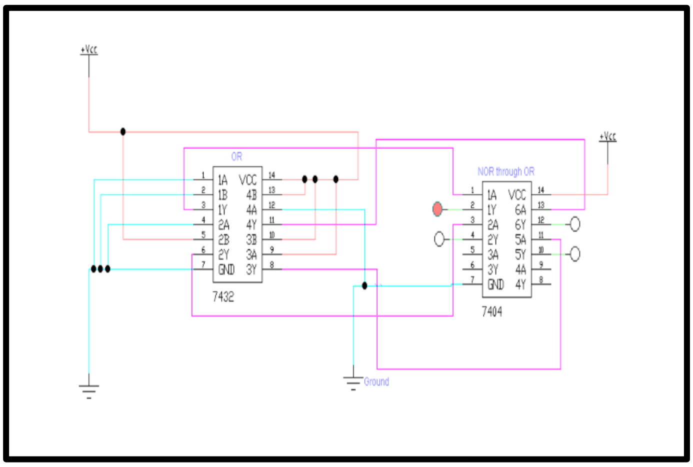
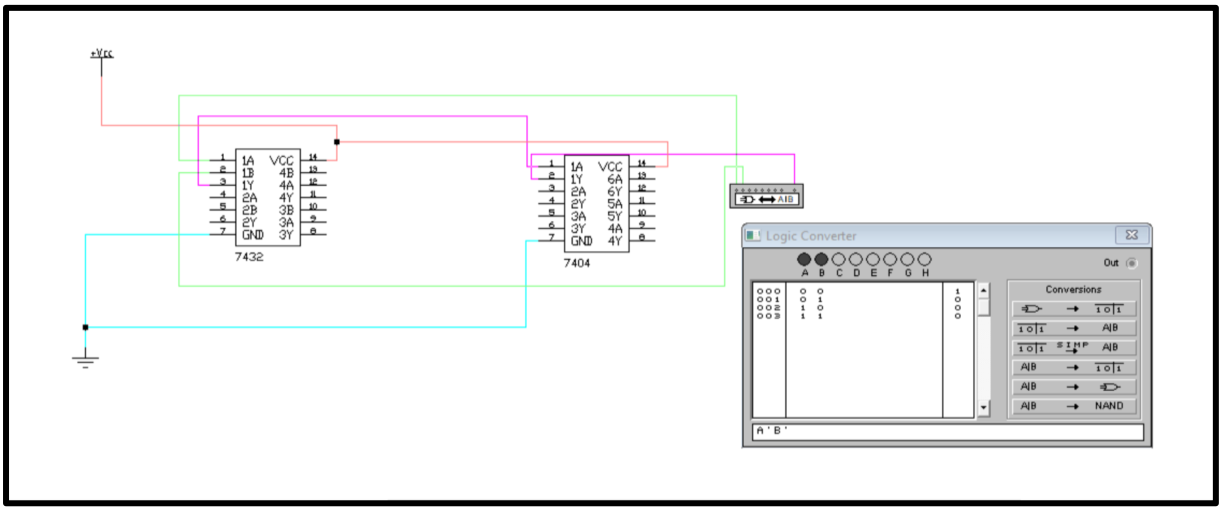


Figure 8: ICs Circuit Implementation

- The simulation was started, and each of the four input combinations (00, 01, 10, 11) was applied using the switches.
- The output was observed at the NOT gate's output, confirming that the constructed logic matched a NOR gate’s behavior.

Truth Table of NOR:



```
000   0  0    1
001   0  1    0
002   1  0    0
003   1  1    0
```

```
A'B'
```

Figure 9: Truth Table Through ICs