 **The University of Azad Jammu & Kashmir, Muzaffarabad**

**Computer Architecture & Logic Design**

|  |  |
| --- | --- |
| **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 DE Morgan’s Laws

**De Morgan’s First Law Verification**

**(A · B)' = A' + B'**

**Objective**:

To verify De Morgan’s First Law practically using basic logic gates.

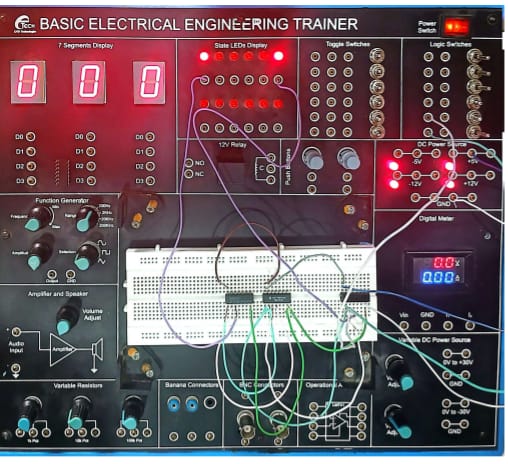
* Used the ICs: **7408 (AND)**, **7404 (NOT)**, and **7432 (OR)**.
* These ICs were inserted on the trainer board.
* VCC and GND connections were made to each IC using the trainer's power rails.
* **Left Side (LHS)**: Connected inputs A and B to the **AND gate**, then connected the output of the AND gate to the **NOT gate**. The NOT gate's output was connected to **LED1** to observe (A · B)'.

Figure 1: De morghan Law -1

* **Right Side (RHS)**: Connected A and B each to individual **NOT gates**, and connected their outputs to an **OR gate**. The OR gate's output was connected to **LED2** to observe A' + B'.
* The input switches were toggled through all 4 combinations: 00, 01, 10, and 11.
* In every case, **LED1 and LED2 gave the same output**, proving the law.

**(A) Gate-Level Circuit**

Circuit built using basic gates (not IC models)

* Opened EWB and created a new project.
* Added two logic switches as inputs **A** and **B**.
* Connected A and B to an **AND gate**, then the output to a **NOT gate**, and connected the output of that to **LED1** — representing (A · B)'.
* Separately, connected A and B to **two NOT gates**, then their outputs to an **OR gate**, and its output to **LED2** — representing A' + B'.
* Toggled input switches through all 4 combinations.
* Both LEDs gave identical outputs, confirming correctness.

**EWB Implementation – De Morgan’s First Law**

**ICs Used in EWB Simulation:**

* **7408** – AND Gate
* **7404** – NOT Gate (2 ICs)
* **7432** – OR Gate

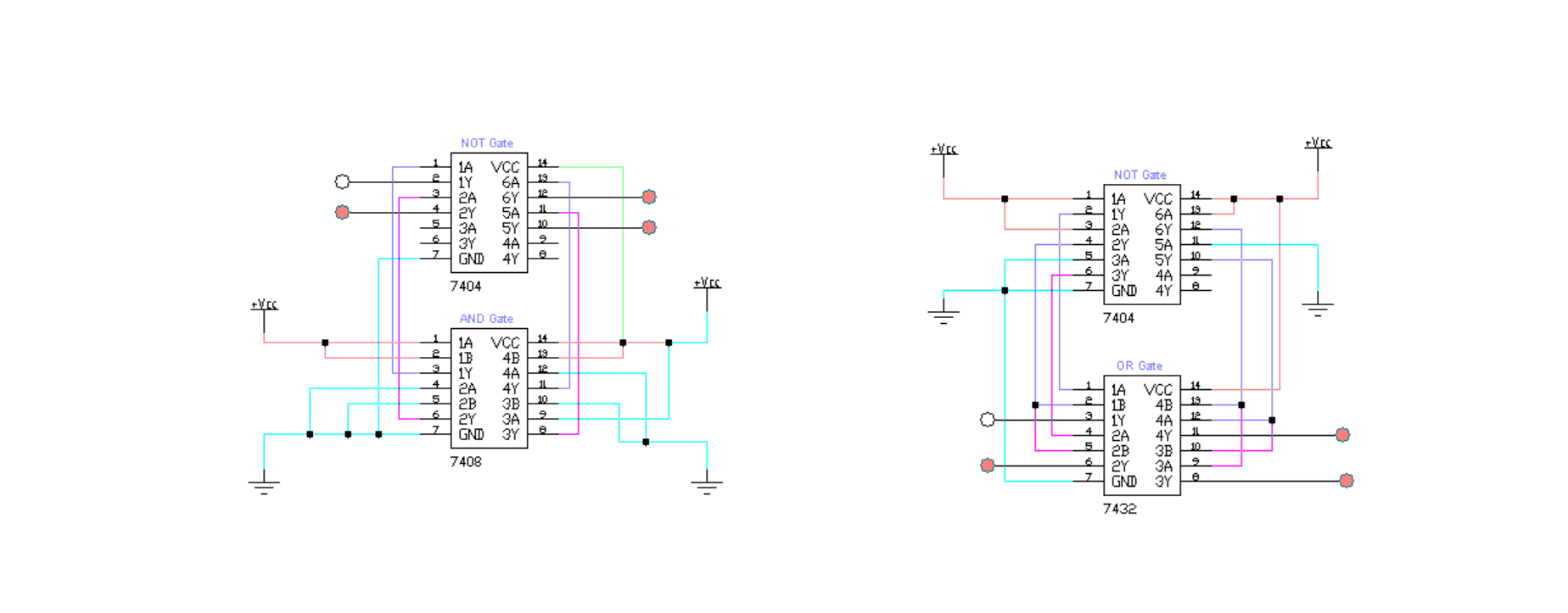
****

Figure 2: DE Morghan First Law ICs Circuit Verification

**Setup for Left-Hand Side (LHS): (A · B)'**

1. Placed **7408 AND Gate IC** on the EWB trainer board.
2. Connected input A and B manually by using input logic controls in EWB (no switches used).
3. Connected the output of the AND gate to a **NOT gate (7404)**.
4. Attached four **virtual LEDs** at the output of the NOT gate for all four combinations:
   * A = 0, B = 0
   * A = 0, B = 1
   * A = 1, B = 0
   * A = 1, B = 1
5. Powered the ICs with virtual **VCC (+5V)** and **GND** terminals.

**Setup for Right-Hand Side (RHS): A' + B'**

1. Connected A and B separately to **two NOT gates (7404)** to get A' and B'.
2. Fed the inverted outputs into the **7432 OR Gate IC**.
3. Attached **four LEDs** to the output of this OR gate, again for the same four combinations.

**Result Verification:**

* The output LEDs of both LHS and RHS circuits were checked side by side.
* **All outputs matched** for each input combination (00, 01, 10, 11).
* This successfully verified **De Morgan’s First Law** virtually using ICs in EWB.

**Truth Tables: (A · B)' = A' + B'**

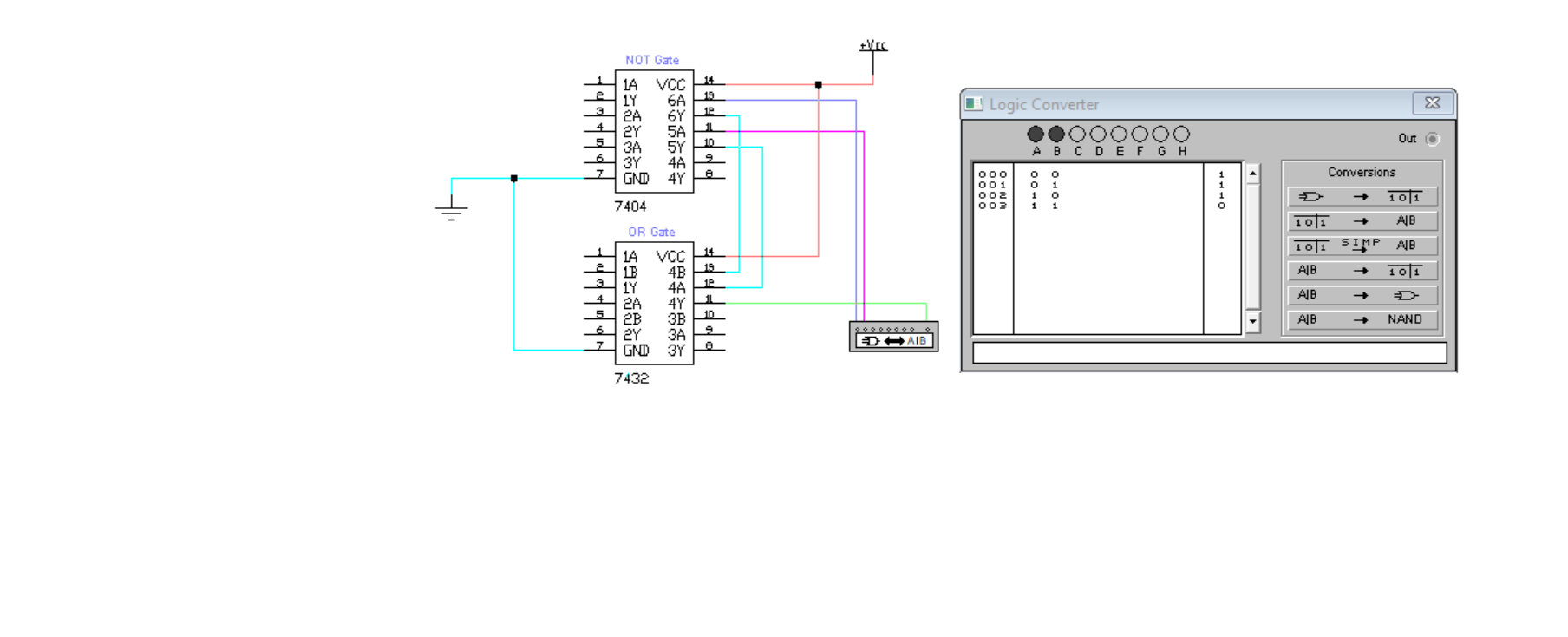
****

Figure 3: Truth Table through Logic Convertor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | A · B | (A · B)' | A' | B' | A' + B' |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |

**De Morgan’s Second Law**

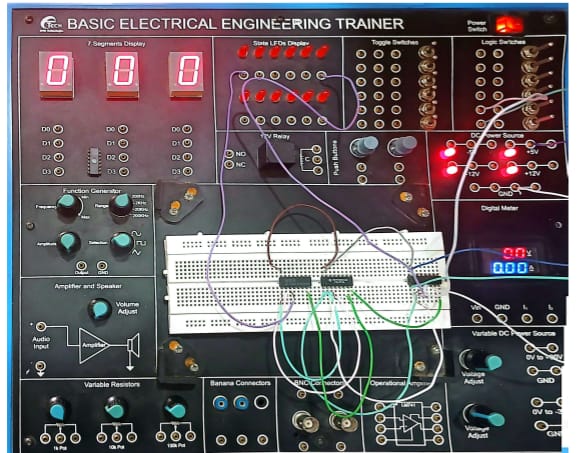
**(A + B)' = A' · B'**

**Lab Implementation (Digital Trainer Board – Practical Setup)**

**Objective:**

To verify De Morgan’s Second Law by practically implementing both sides of the expression on the digital trainer board using ICs, power supply, and LEDs.

**Left-Hand Side (LHS): (A + B)'**

1. Placed the **7432 OR Gate IC** on the digital trainer board.
2. Connected input A and B using wired input lines from the trainer board input section.
3. Output of the OR gate was connected to the **7404 NOT Gate IC** to invert (A + B).
4. ****Four different combinations (00, 01, 10, 11) were tested manually.
5. Output was connected to **LED** to verify the result visually for each case.

**Right-Hand Side (RHS): A' · B'**

1. Inputs A and B were separately connected to **two NOT gates** (using 7404 IC) to get A' and B'.
2. The outputs of those NOT gates were connected to a **7408 AND Gate IC**.
3. Four combinations were again applied.
4. Output of the AND gate was connected to an **LED**.
5. Outputs from both sides were compared — **both were same for all inputs**, proving the law.

**EWB Implementation**

The simulation was done in **two ways**:

1. Using GATE-based circuit
2. Using **IC-based circuit on virtual trainer board** in EWB  
   (No switches used, inputs controlled using built-in logic tools of EWB)

**Gate-Level Circuit Implementation in EWB**

**LHS: (A + B)'**

1. Placed **OR Gate** from the EWB component library.
2. Connected inputs A and B using logic toggles.
3. Output of OR gate was passed through a **NOT gate** to invert the result.
4. LED was attached to output to display the final result.

**RHS: A' · B'**

1. Connected A and B separately to **two NOT gates**.
2. Connected their outputs to an **AND gate**.
3. Final output from AND was connected to another LED.
4. Switched between combinations using toggles and verified both outputs matched.

**IC-Level Circuit Implementation in EWB**

**ICs Used:**

* **7432** – OR Gate
* **7404** – NOT Gate (used twice)
* **7408** – AND Gate

**Procedure for EWB IC Setup:**

**Setup 1 (LHS): (A + B)'**

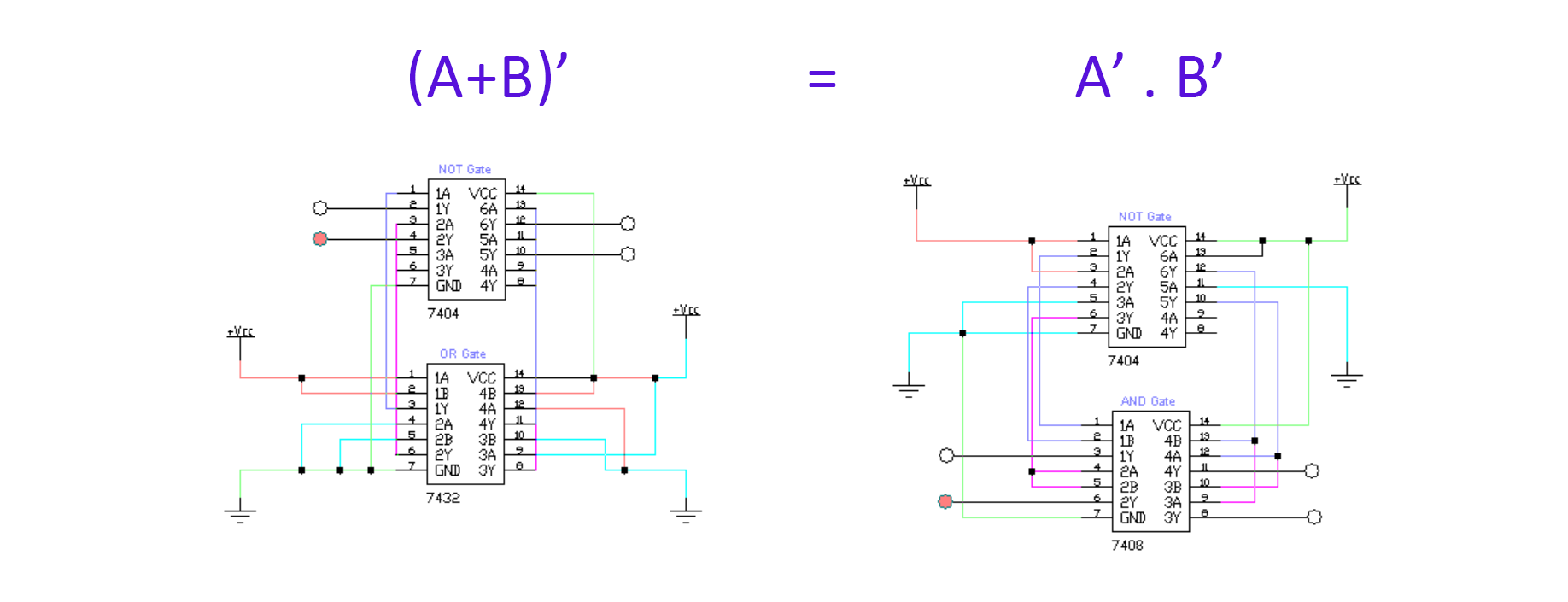
1. Placed **7432 OR Gate IC** on the trainer board.
2. Connected A and B to two inputs of the OR gate.
3. Output was routed to a **NOT gate** using 7404 IC.
4. Final output connected to an LED.

Figure 4: Proof Through IC Circuit

**Setup 2 (RHS): A' · B'**

1. Input A sent through one NOT gate, input B through another.
2. Outputs from both NOTs connected to **7408 AND Gate IC**.
3. Final AND output connected to an **LED**.
4. Four LEDs used for testing four combinations (00, 01, 10, 11).

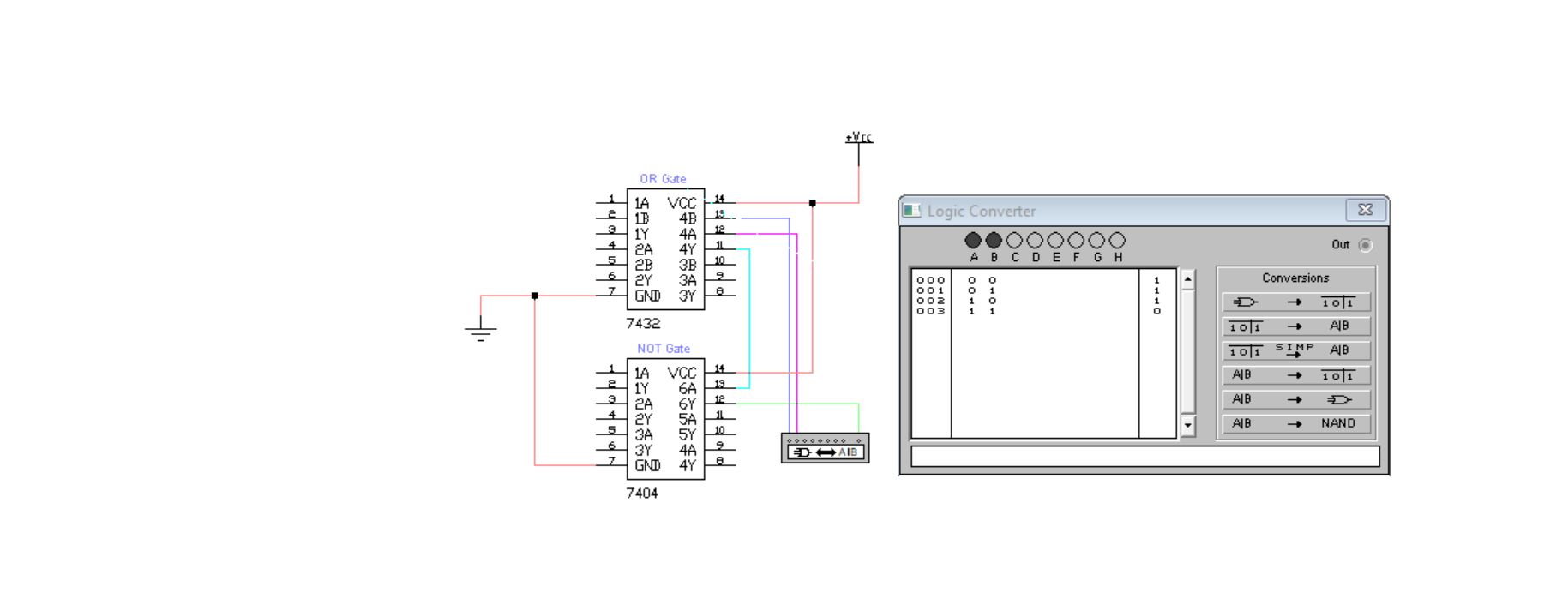
**TRUTH TABLE: (A+B)’ = A’ . B’**

Figure 5: Truth Table through Logic Convertor

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | A + B | (A + B)' | A' | B' | A' · B' |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |

**Result:**

* For all combinations of A and B:
  + The output of LHS and RHS circuits **were identical**.

**De Morgan’s Second Law was verified successfully** both practically and in simulation.