

|  |  |
| --- | --- |
| Student Name | Afzal Ali Ahmed Shah |
| Roll no | **2024-SE-04** |
| Semester | **2nd** |
| Instructor Name | **Engr. Sidra Rafique** |
| Subject | **CA&LD** |
| Course Code | **CS-1206** |
| Credit Hours | **3+1** |
| Lab | **02** |
| Date | **15-May-2025** |
| Department of Software Engineering | |

**Objective**

The primary objective of this lab is to understand and verify the behavior of basic logic gates using a virtual simulation environment. By conducting this lab, students will be able to:

* Analyze the logical operations of various gates (AND, OR, NOT, NAND, NOR, XOR).
* Understand the theoretical foundation behind digital decision-making circuits.
* Learn how to implement and test circuits virtually using EWB (Electronics Workbench).

**Apparatus Used**

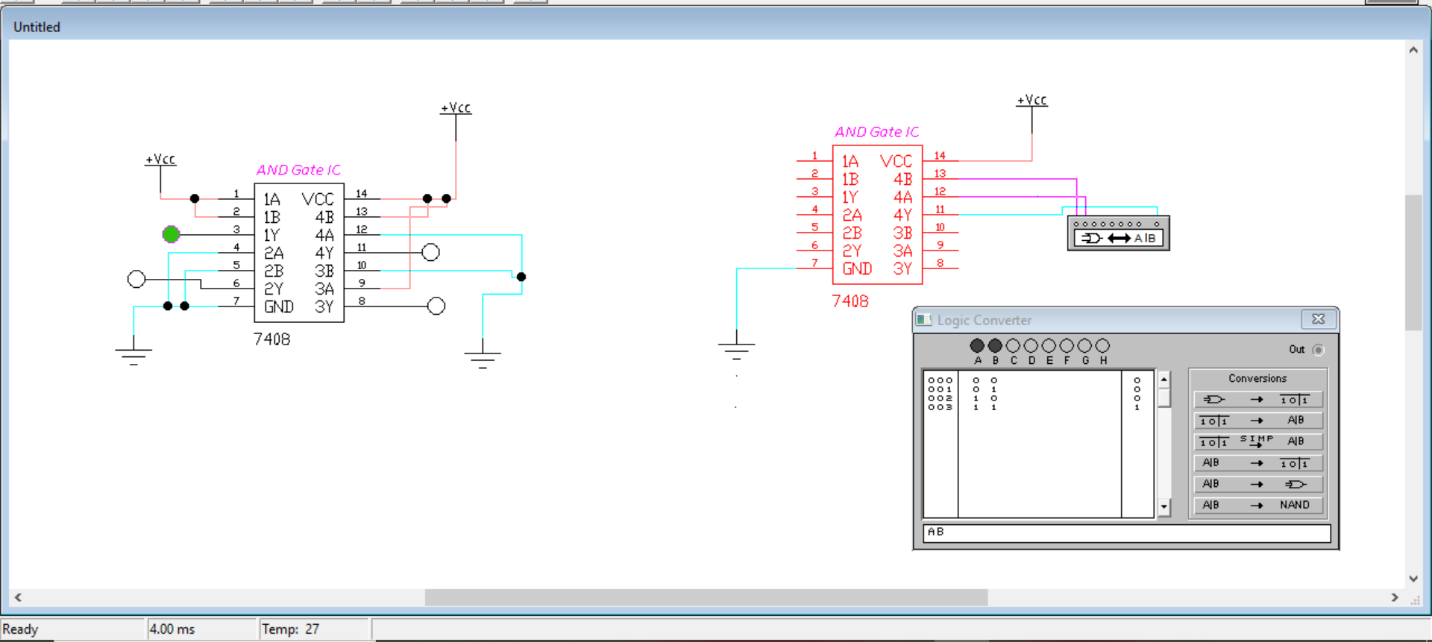
* Power Supply (virtual)
* Breadboard (virtual)
* Connecting Wires (virtual)
* Logic Gates (ICs: 7400, 7402, 7404, 7408, 7432, 7486)
* Simulation Software: **Electronics Workbench (EWB)**

**Theory**

Logic gates are fundamental components used in digital electronics. These gates perform logical operations on one or more binary inputs to produce a single binary output. Each logic gate implements a specific Boolean function.

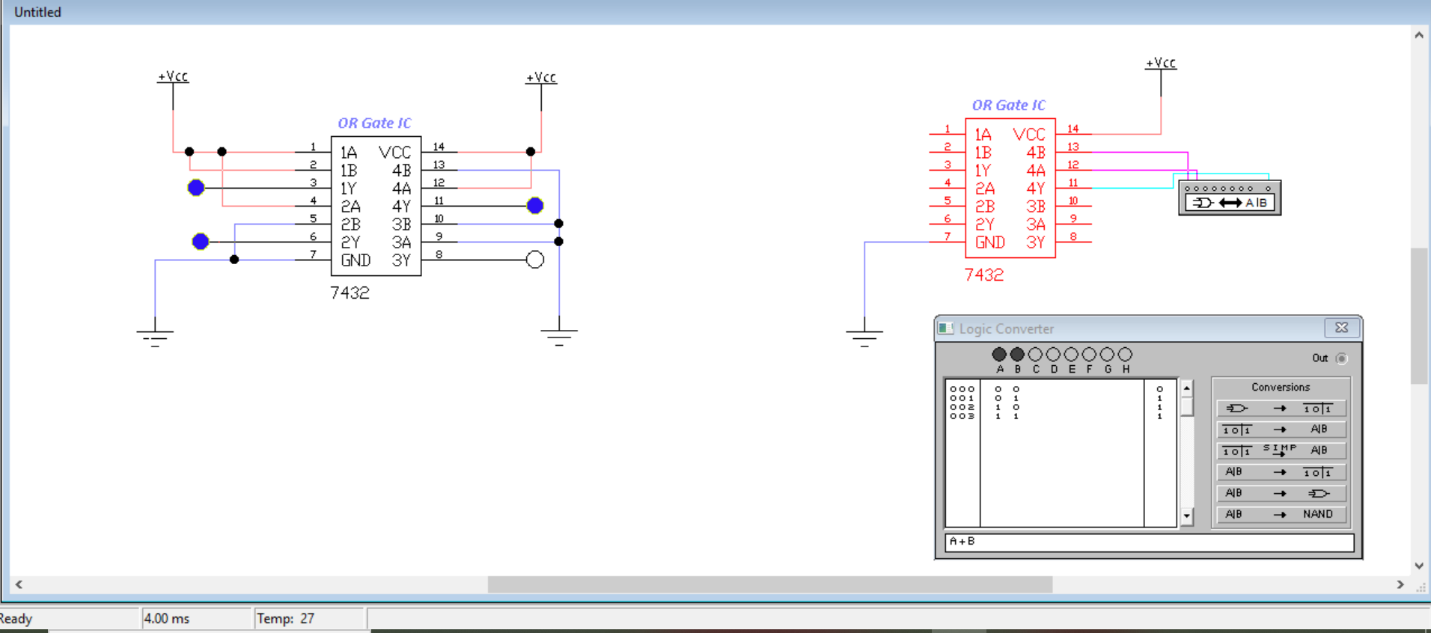
**AND Gate**

Symbol: ·  
Logic Equation: **Y = A · B**  
The AND gate outputs high (1) only when **both inputs** are high.



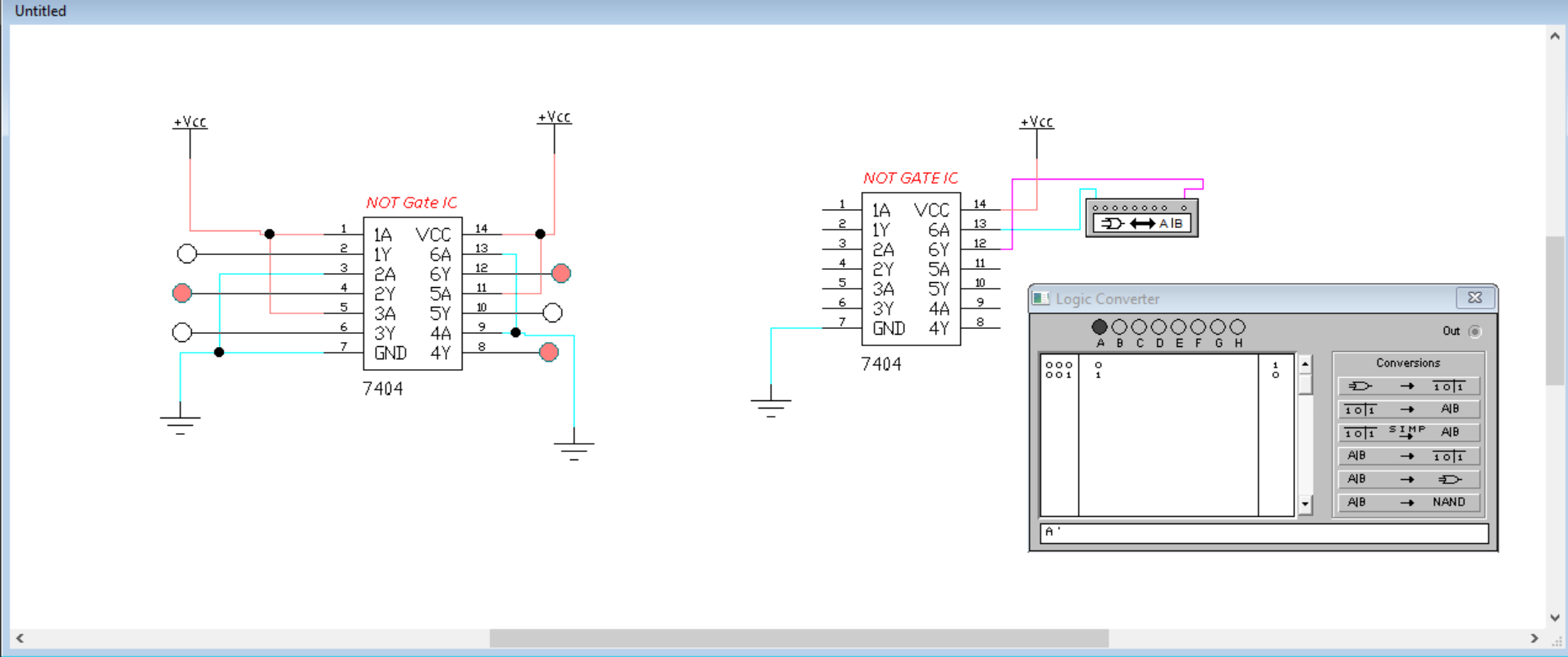
**OR Gate**

Symbol: +  
Logic Equation: **Y = A + B**  
The OR gate outputs high (1) when **at least one input** is high.



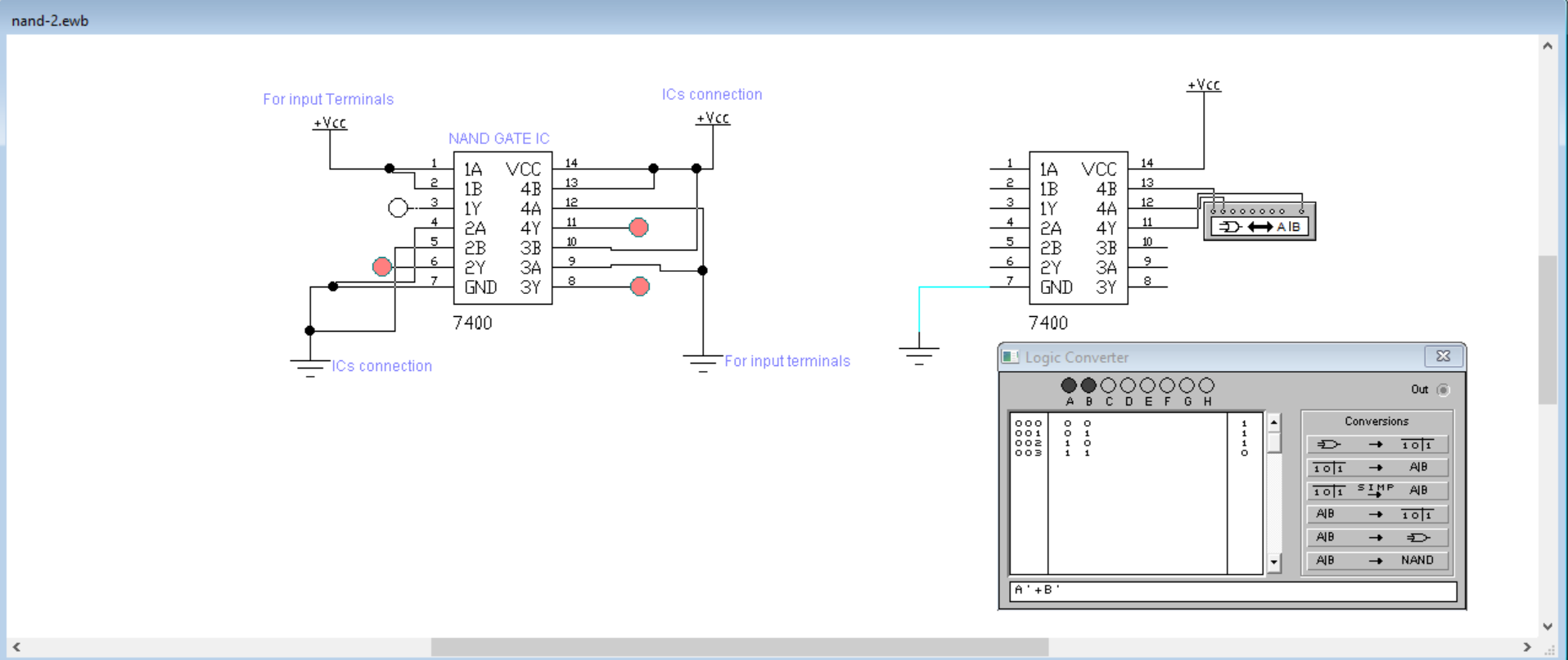
**NOT Gate**

Symbol: – or Overline  
Logic Equation: **Y = Ā**  
The NOT gate outputs the **inverse** of the input. If input is 1, output is 0 and vice versa.



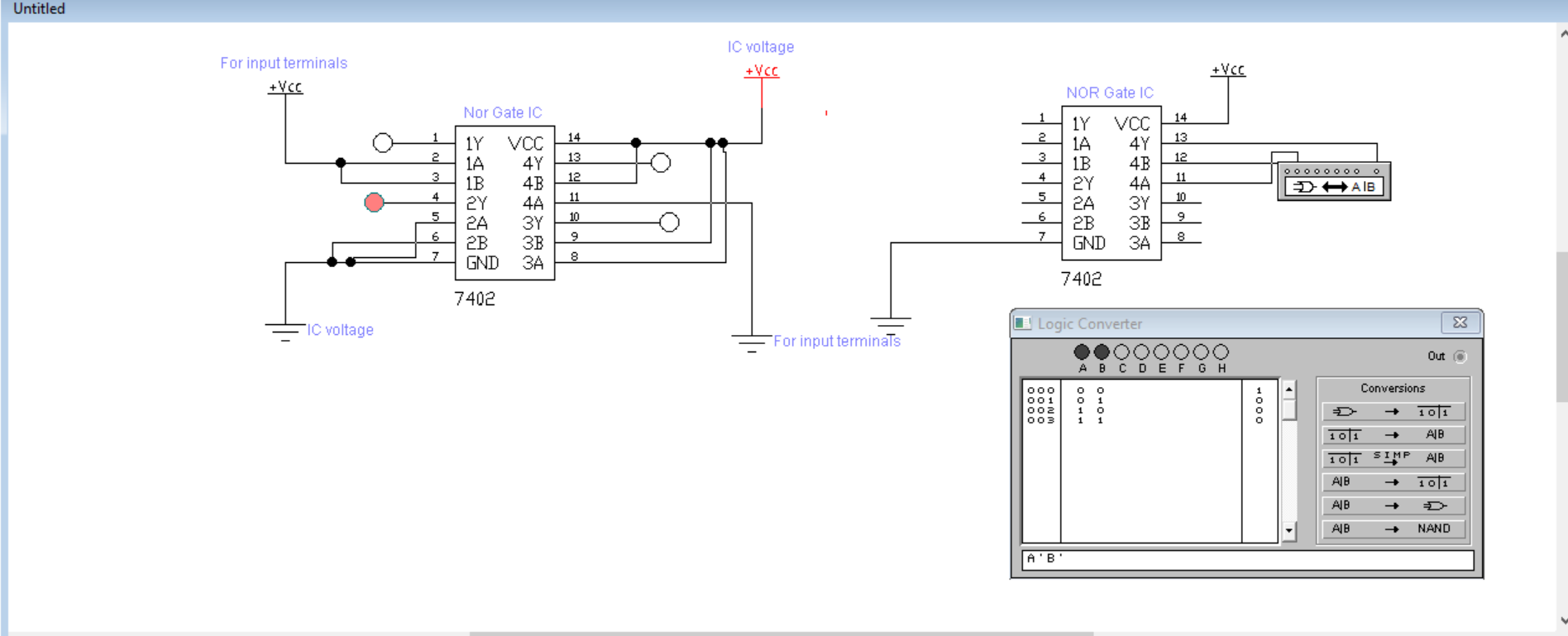
**NAND Gate**

Logic Equation: **Y = (A · B)’**  
It is the complement of the AND gate. The output is low (0) **only when both inputs are high**.



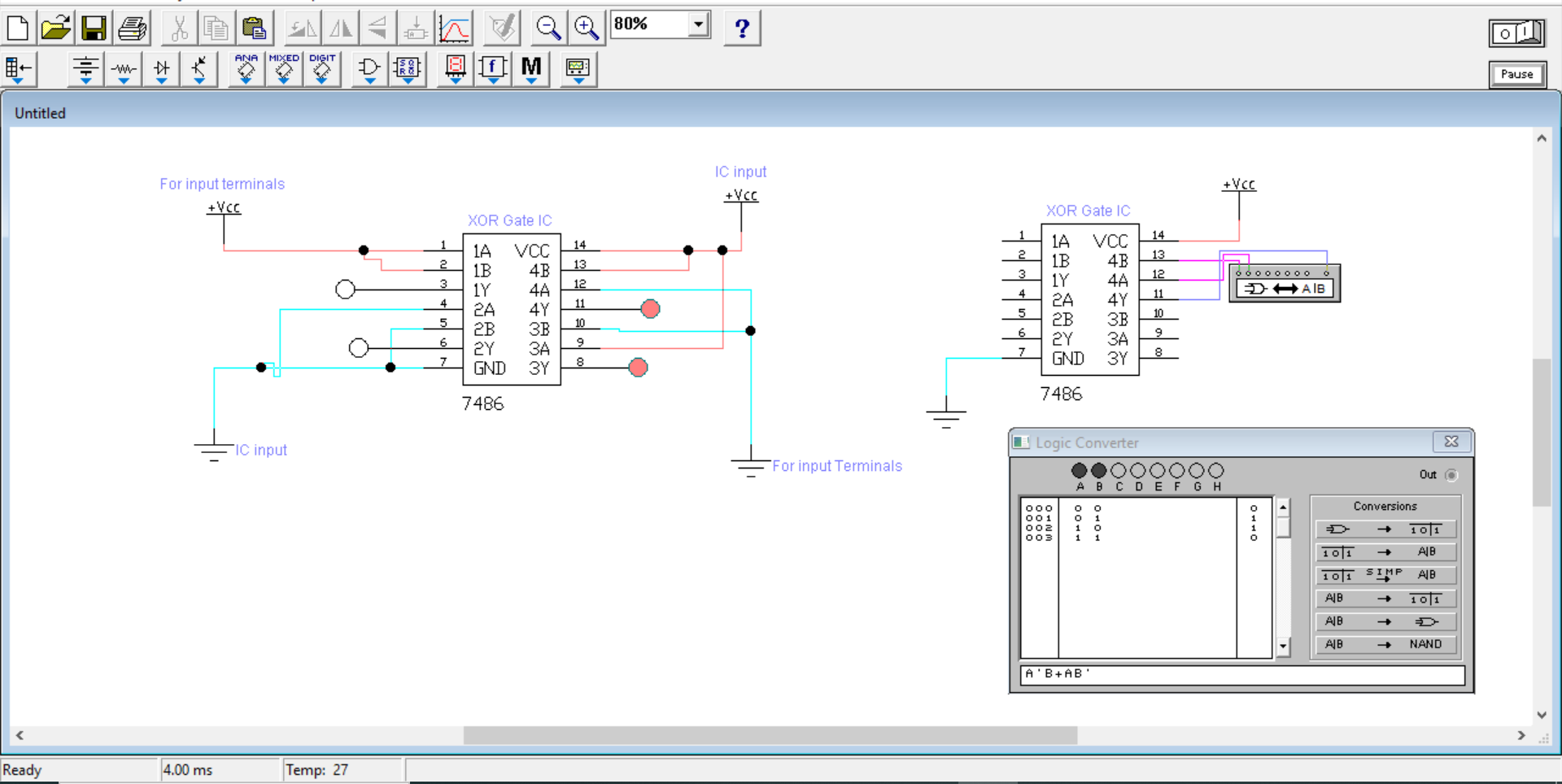
**NOR Gate**

Logic Equation: **Y = (A + B)’**  
It is the complement of the OR gate. The output is high (1) **only when both inputs are low**.



**XOR Gate (Exclusive OR)**

Logic Equation: **Y = A⊕B = AB’ + A’B**  
The XOR gate outputs high (1) when **the inputs are different**, i.e., one is high and the other is low.



These gates are commonly available in integrated circuit (IC) form and are used to build complex logical expressions and digital systems.

**Procedure**

1. **Open Electronics Workbench (EWB):**  
   Launch the simulation software and create a new project.
2. **Select the Required Logic ICs:**  
   Use the component library to choose the ICs representing the desired logic gates (7400 for NAND, 7402 for NOR, etc.).
3. **Place the Components:**  
   Insert the selected ICs on the virtual breadboard workspace.
4. **Connect Power Supply:**
   * Connect pin 14 of each IC to **Vcc (5V)**.
   * Connect pin 7 of each IC to **Ground (0V)**.  
     These are standard power connections for TTL ICs.
5. **Wire the Inputs and Outputs:**
   * Use logic switches to simulate the input signals A and B.
   * Use virtual indicators (e.g., logic probes or LEDs) to observe the output Y.
6. **Test Each Gate Separately:**
   * Apply different combinations of binary inputs (0 and 1).
   * Observe the resulting output for each combination.
   * Ensure that the behavior matches the expected logic equation of each gate.
7. **Repeat for All Gates:**  
   Follow the same procedure for AND, OR, NOT, NAND, NOR, and XOR gates.

**Conclusion**

Through this experiment, we successfully verified the functionality of all basic logic gates using Electronics Workbench. Each gate responded precisely according to its theoretical Boolean expression. This lab strengthened our foundational understanding of digital logic design and demonstrated how these gates are used to perform binary decision-making in electronic circuits.