Lab Report 4 Date:2081/06/11

Title: Realization of logic gates using java.

Objective: The objective of this lab is to implement the basic logic gates (AND, OR, NOT) using C, C++, and Java programming languages.

1. Theory:

1.1 AND Gate:

Definition: An AND gate is a basic digital logic gate that implements logical conjunction – a fundamental operation in Boolean algebra. It operates on two or more binary inputs and produces a true (1) output only when all of its inputs are true (1). Otherwise, the output is false (0).

Truth Table:

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Output (A AND B)** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

1.2 OR Gate:

Definition: An OR gate is a fundamental digital logic gate that implements logical disjunction (OR operation) in Boolean algebra. It operates on two or more binary inputs and produces a true (1) output if at least one of its inputs is true (1). If all inputs are false (0), the output is false (0).

Truth Table:

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Output (A OR B)** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

1.3 NOT Gate:

Definition: A NOT gate is a fundamental digital logic gate that implements logical negation or inversion. It operates on a single binary input and produces an output that is the opposite (inverted) of the input. If the input is true (1), the output will be false (0), and if the input is false (0), the output will be true (1).

Truth Table:

|  |  |
| --- | --- |
| A | NOT A |
| 0 | 1 |
| 1 | 0 |

1. Program in Java:

*public class Main {*

*public static boolean AND(boolean A, boolean B) {*

*return A && B;*

*}*

*public static boolean OR(boolean A, boolean B) {*

*return A || B;*

*}*

*public static boolean NOT(boolean A) {*

*return !A;*

*}*

*public static void main(String[] args) {*

*boolean A = true, B = false;*

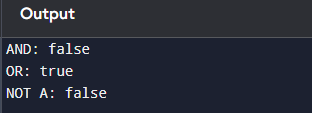
*System.out.println("AND: " + AND(A, B));*

*System.out.println("OR: " + OR(A, B));*

*System.out.println("NOT A: " + NOT(A));*

*}}*

Output:



1. Conclusion:

In this lab, we explored the programing implementation and functioning of fundamental logic gates (AND, OR, NOT, etc.) through hands-on exercises. By coding and simulating these gates, we gained a deeper understanding of how they operate and how they are used in the design of more complex digital circuits.