|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Digital Design Laboratory** | **Semester:** | **III** |
| **Date of Performance:** | **31 / 7 / 2023** | **Batch No:** | **C2** |
| **Faculty Name:** |  | **Roll No:** | **16010122267** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** | **\_\_\_/25** |

**Experiment No: 1**

**Title: Study of Basic Gates and Universal Gates**

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| Understand Basic Logic Gates and Universal Gates |

|  |
| --- |
| **COs to be achieved:** |
| **CO1**: Recall basic gates & logic families and binary, octal & hexadecimal calculations and conversions. |

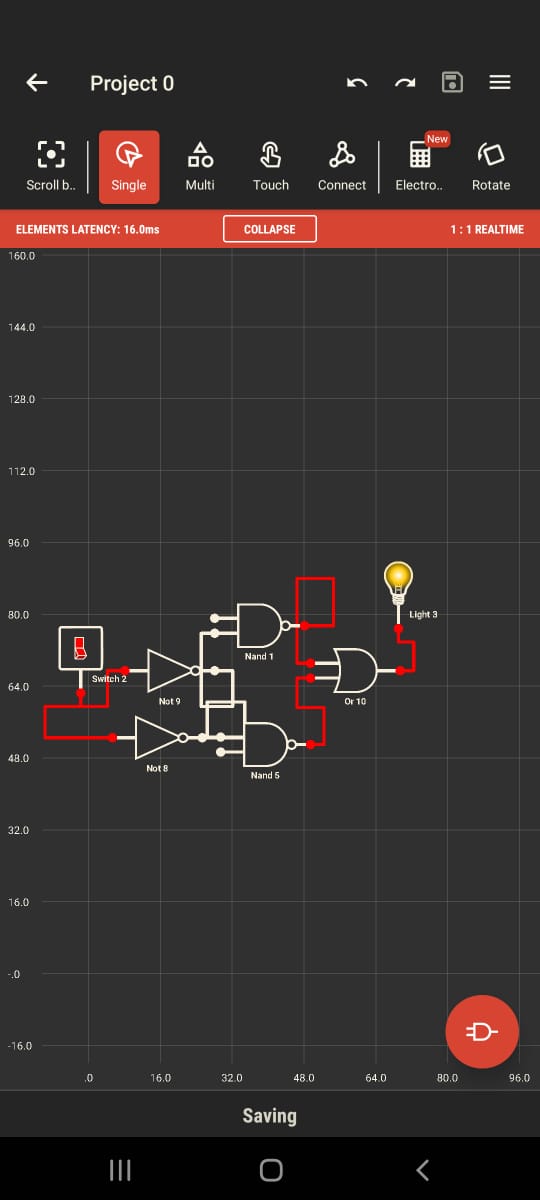
|  |
| --- |
| **Tools used:** |
| Trainer kits |

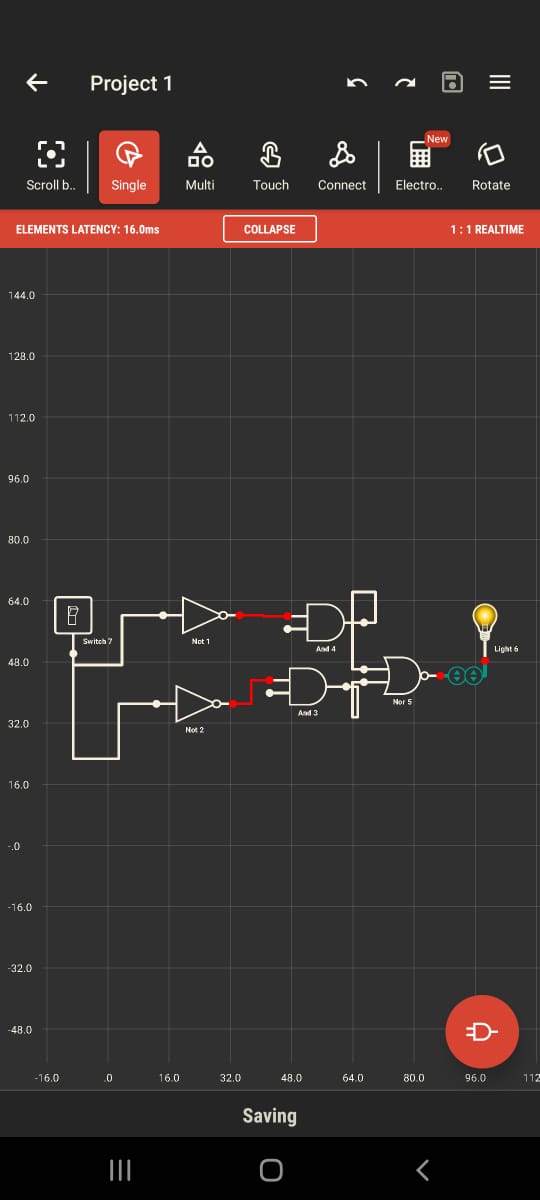
|  |
| --- |
| **Theory:** |
| Logic gates are electronic circuits that perform logical operations on one or more input signals to produce an output signal based on a set of logical rules. Logic gates can be classified into the following categories:   1. Basic Gates:    1. AND Gate: The AND gate produces a high output (1) only when all of its inputs are high (1).    2. OR Gate: The OR gate produces a high output (1) if any of its inputs is high (1).    3. NOT Gate (Inverter): The NOT gate produces the logical complement of its input. It takes a single input and produces the opposite value as the output. 2. Derived Gates:    1. NAND Gate: The NAND gate is a combination of an AND gate followed by a NOT gate. It produces the inverse of the AND gate's output. It outputs a low (0) only when all of its inputs are high (1).    2. NOR Gate: The NOR gate is a combination of an OR gate followed by a NOT gate. It produces the inverse of the OR gate's output. It outputs a high (1) only when all of its inputs are low (0).    3. XOR Gate (Exclusive OR): The XOR gate produces a high output (1) when the number of high inputs is odd. It outputs a low (0) when the number of high inputs is even.    4. XNOR Gate (Exclusive NOR): The XNOR gate produces a high output (1) when the number of high inputs is even. It outputs a low (0) when the number of high inputs is odd. 3. Universal Gates:   NAND and NOR gates are considered universal gates because any logic function can be implemented using only NAND gates or only NOR gates. This means that with a sufficient number of NAND or NOR gates, you can create circuits that can perform any logical operation. |

|  |
| --- |
| **Implementation Details** |
| 1. AND Gate: Y =   Symbol.    Pin Diagram    Truth Table:     1. OR Gate: Y =   Symbol    Pin Diagram    Truth Table:     1. NOT Gate: Y =   Symbol    Pin Diagram    Truth Table:     1. NAND Gate: Y =   Symbol    Pin Diagram    Truth Table:     1. NOR Gate: Y =   Symbol    Pin Diagram    Truth Table:     1. XOR Gate: Y =   Symbol    Pin Diagram    Truth Table:     1. XNOR Gate: Y =   Symbol    Pin Diagram  XNOR Gate Circuit Diagram & Working Explanation  Truth Table:    **Implementation Using NAND Gate**  **NOT GATE**    **AND GATE**    **OR GATE**    **Implementation Using NOR Gate**  **NOT GATE**      **AND GATE**    **OR GATE** |

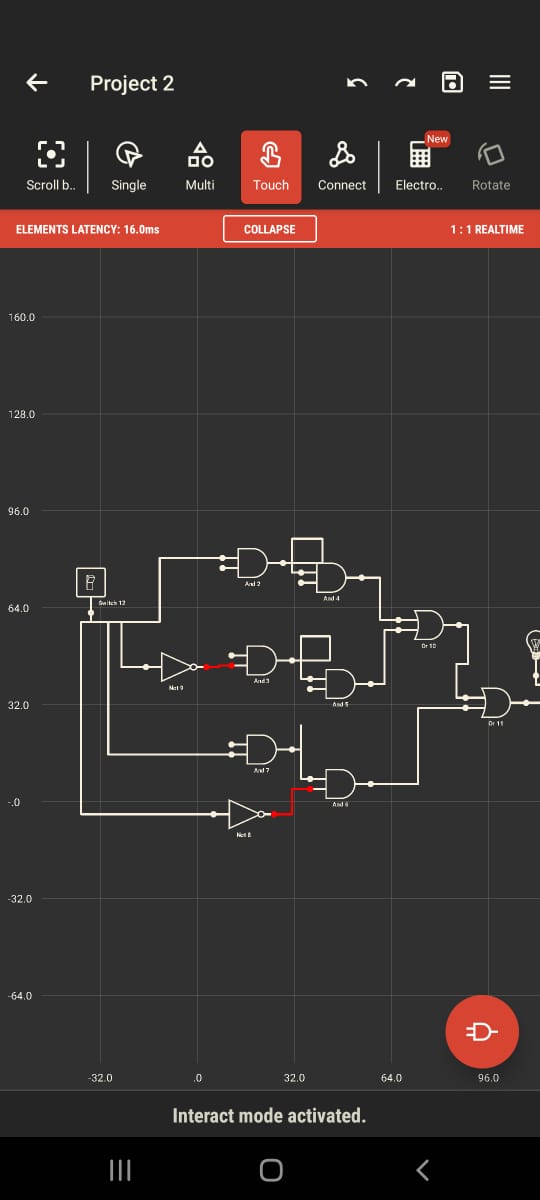
|  |
| --- |
| **Post Lab Subjective/Objective type Questions:** |
| 1. Implement the Boolean function using NAND gates and NOR gates F=A’B + AB’ 2. Implement using combination of gates F = ABC + AB’C + ABC’ |

**Ans 1:**

****

****

**Ans 2:**

****

|  |
| --- |
| **Conclusion:** |
| *Thus, we implemented the logic of numerous logic gates utilising their corresponding integrated circuits.*  *We also learned how to create logic gates using two basic gates: NAND and NOR. We understood the logic behind the gates' truth tables.* |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |