

**Vidyavardhini’s**

**College of Engineering & Technology**

Vasai Road (W)

**Department of Computer Engineering**

**Laboratory Manual**

**[Student Copy]**

|  |  |  |  |
| --- | --- | --- | --- |
| **Semester** | **III** | **Class** | **S.E** |
| **Course Code** | **CSL302** | | |
| **Course Name** | **Digital Logic & Computer Organization Architecture Lab** | | |



**Vidyavardhini’s College of Engineering & Technology**

**Vision**

To be a premier institution of technical education; always aiming at becoming a valuable resource for industry and society.

**Mission**

* To provide technologically inspiring environment for learning.
* To promote creativity, innovation and professional activities.
* To inculcate ethical and moral values.
* To cater personal, professional and societal needs through quality education.

**Department Vision:**

To evolve as a center of excellence in the field of Computer Engineering to cater to industrial and societal needs.

**Department Mission:**

* To provide quality technical education with the aid of modern resources.
* Inculcate creative thinking through innovative ideas and project development.
* To encourage life-long learning, leadership skills, entrepreneurship skills with ethical & moral values.

**Program Education Objectives (PEOs):**

PEO1: To facilitate learners with a sound foundation in the mathematical, scientific and engineering fundamentals to accomplish professional excellence and succeed in higher studies in Computer Engineering domain

PEO2: To enable learners to use modern tools effectively to solve real-life problems in the field of Computer Engineering.

PEO3: To equip learners with extensive education necessary to understand the impact of computer technology in a global and social context.

PEO4: To inculcate professional and ethical attitude, leadership qualities, commitment to societal responsibilities and prepare the learners for life-long learning to build up a successful career in Computer Engineering.

**Program Specific Outcomes (PSOs):**

PSO1: Analyze problems and design applications of database, networking, security, web technology, cloud computing, machine learning using mathematical skills, and computational tools.

PSO2: Develop computer-based systems to provide solutions for organizational, societal problems by working in multidisciplinary teams and pursue a career in the IT industry.

**Program Outcomes (POs):**

Engineering Graduates will be able to:

* **PO1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
* **PO2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
* **PO3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
* **PO4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
* **PO5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
* **PO6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
* **PO7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
* **PO8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
* **PO9. Individual and teamwork:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
* **PO10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
* **PO11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
* **PO12. Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**Course Objectives**

|  |  |
| --- | --- |
| 1 | Acquiring design experience in using a truth table to implement a logic circuit |
| 2 | To implement operations of the arithmetic unit using algorithms. |
| 3 | Design and simulate different digital circuits. |
| 4 | To design memory subsystem including cache memory. |
| 5 | To demonstrate CPU and ALU design. |

**Course Outcomes**

|  |  |  |  |
| --- | --- | --- | --- |
| At the end of the course student will be able to: | | **Action verb** | **Bloom Level** |
| **CSL302.1** | Verify truth table of logic, universal gates, and realize logic circuits using hardware. | Verify,  Realize | Apply (level - 3) |
| **CSL302.2** | Implement combinational circuits design using hardware. | Implement | Apply (level - 3) |
| **CSL302.3** | Implement sequential & code conversion circuits design using hardware. | Implement | Apply (level - 3) |
| **CSL302.4** | Write Booth's, Restoring, and Non-Restoring algorithms for arithmetic operations using C-Programming language. | Write | Apply (level - 3) |
| **CSL302.5** | Implement ripple carry adder, carry look-ahead adder, ALU design using virtual lab. | Implement | Apply (level - 3) |
| **CSL302.6** | Implement CPU, memory and Cache memory designs using virtual lab. | Implement | Apply (level - 3) |

**Mapping of Experiments with Course Outcomes**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Experiment | Course Outcomes | | | | | |
| **CSL302.1** | **CSL302.2** | **CSL302.3** | **CSL302.4** | **CSL302.5** | **CSL302.6** |
| Verify truth table of various logic gates using ICs. | 3 |  |  |  |  |  |
| Realize the gates using universal gates | 3 |  |  |  |  |  |
| Implement the given Boolean function using logic gates in SOP and POS forms. | 3 |  |  |  |  |  |
| Realize half subtractor and full subtractor |  | 3 |  |  |  |  |
| Code conversion. |  |  | 3 |  |  |  |
| Implement Booth’s algorithm using c programming. |  |  |  | 3 |  |  |
| Implement restoring division algorithm using c programming.. |  |  |  | 3 |  |  |
| Implement ripple carry adder. |  |  |  |  | 3 |  |
| Implement carry look ahead adder. |  |  |  |  | 3 |  |
| Implement ALU design. |  |  |  |  | 3 |  |

**INDEX**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Name of Experiment** | **D.O.P.** | **D.O.C.** | **Page No.** | **Remark** |
| 1 | To verify the truth table of various logic gates using ICs. | 26/07/23 | 02/08/23 |  |  |
| 2 | To realize the gates using universal gates. | 02/08/23 | 09/08/23 |  |  |
| 3 | To implement the given Boolean function using logic gates in SOP and POS forms. | 09/08/23 | 16/08/23 |  |  |
| 4 | To realize half subtractor and full subtractor | 16/09/23 | 23/08/23 |  |  |
| 5 | To implement 4-bit Binary to Gray and Gray to binary code converter. | 23/08/23 | 06/09/23 |  |  |
| 6 | To implement Booth’s algorithm. | 06/09/23 | 13/09/23 |  |  |
| 7 | To implement restoring division algorithm. | 13/09/23 | 27/09/23 |  |  |
| 8 | To implement ripple carry adder. | 27/09/23 | 04/10/23 |  |  |
| 9 | To implement carry look ahead adder. | 04/10/23 | 11/10/23 |  |  |
| 10 | To implement ALU design. | 11/10/23 | 11/10/23 |  |  |

**General guidelines for performing hardware based experiments**

* Connecting wires should be rubbed with sand papers so that there is no rust.
* Never touch live and naked wires.
* Make sure that the apparatus is switched off while placing ICs and connecting of wires.
* Always connect ground first and then connect Vcc.
* The connections should be made neat and tight.
* ICs are placed in a proper way in the breadboard. There is no short of current in same inputs.
* After completed the experiments switch off the supply of the apparatus

**General guidelines to use the simulator for performing the experiment**

* Start the simulator as directed. For more detail please refer to the manual for using the simulator
* The simulator supports 5-valued logic
* To add the logic components to the editor or canvas (where you build the circuit) select any component and click on the position of the canvas where you want to add the component
* The pin configuration is shown when you select the component and press the 'show pinconfig' button in the left toolbar or whenever the mouse is hovered on any canned component of palette
* To connect any two components select the connection tool of palette, and then click on the source terminal and then click on the the target terminal
* To move any component select the component using the selection tool and drag the component to the desired position
* To give a toggle input to the circuit, use 'Bit Switch' which will toggle its value with a double click
* Use 'Bit Display' component to see any single bit value. 'Digital Display' will show the output in digital format
* undo/redo, delete, zoom in/zoom out, and other functionalities have been given in the top toolbar for ease of circuit building
* Use start/stop clock pulse to start or stop the clock input of the circuit. Clock period can be set from the given 'set clock' button in the left toolbar
* Use 'plot graph' button to see input-output wave forms
* Users can save their circuits with .logic extension and reuse them
* After building the circuit press the simulate button in the top toolbar to get the output
* If the circuit contains a clock pulse input, then the 'start clock' button will start the simulation of the whole circuit. Then there is no need to again press the 'simulate' button

**Name:Gaurav Kishor Patil**

**Roll No:54 Batch:C**

**Div:2**

|  |
| --- |
| Experiment No. 1 |
| Truth table of various logic gates using ICs. |
| Date of Performance:26/07/23 |
| Date of Submission:02/08/23 |

**Aim -** To verify the truth table of various logic gates using ICs.

**Objective -**

1. Understand how to use the breadboard to patch up, test your logic design and debug it.
2. The principal objective of this experiment is to fully understand the function and use of logic gates.
3. Understand how to implement simple circuits based on a schematic diagram using logic gates.

**Components required -**

1. IC’s 7408, 7432, 7404

2. Bread Board.

3. Connecting wires.

**Theory -**

In digital electronics, a gate is logic circuits with one output and one or more inputs. Logic gates are available as integrated circuits.

**AND gate** :

AND gate performs logical multiplication, more commonly known as AND operation. The AND gate output will be in high state only when all the inputs are in high state.7408 is a Quad 2 input AND gate.

**OR gate:**

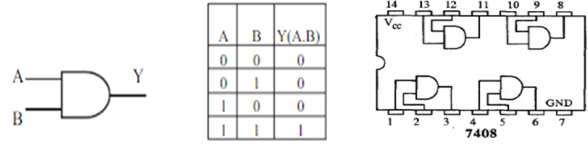
It performs logical addition. Its output become high if any of the inputs is in logic high. 7432 is a Quad 2 input OR gate.

**NOT gate:**

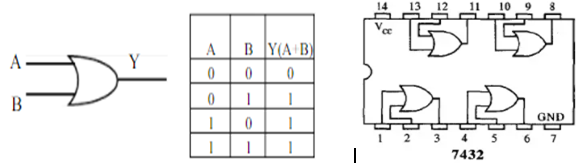
It performs basic logic function for inversion or complementation. The purpose of the inverter is to change one logic level to the opposite level. IC 7404 is a Hex inverter.

**Circuit Diagram, Truth Table -**

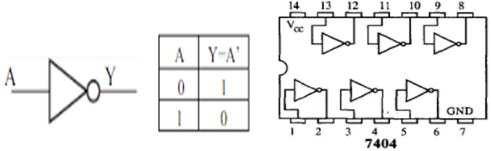
**AND Gate -**



**OR Gate -**



**NOT Gate -**



**Procedure:**

1.Test all the components in the Ic packages using a digital IC tester. Also assure whether all the connecting wires are in good condition by testing for the continuity using a Multimeter or a trainer kit.

2.Verify the dual in line package (DIP) inout of the IC before feeding the inputs.

3.Set up the circuits and observe the outputs.

**Conclusion -**

The Universal Logic Gates are the ones that can produce any other logic or Boolean expression. The AND, OR, and NOT are the basic gates and can produce any logic or Boolean expression. The logic gate(s) which can produce any other logic are grouped as a “Set”.