NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY
GREATER NOIDA
(An Autonomous Institute)
(NAAC ACCREDITED)
Approved by AICTE and Affiliated to Dr. A.P.J. Abdul Kalam Technical University Uttar Pradesh, Lucknow

LABORATORY MANUAL

Digital Electronics and IoT Systems Lab (CEC0151/CEC0251)

COURSE: B.TECH. SEMESTER: I/II

School of Electronics Engineering

(Department of Electronics and Communication Engineering)

(NBA ACCREDITED)

VISION OF THE INSTITUTE

To be an institute of academic excellence in digital arena with global outreach delivering socially responsible professionals to become a university and an entrepreneurial hub.

MISSION OF THE INSTITUTE

- To impart quality education and hone students' skills and competencies making them future ready.
- To foster an ecosystem for research, product development, innovation, incubation and entrepreneurship.
- To instill values and ethics to produce socially responsible technocrats addressing global problems.
- To develop an environment for sharing and exchange of resources globally for lifelong learning.

VISION OF THE DEPARTMENT

To be a renowned Centre of Excellence in Electronics and Communication Engineering, developing globally competent ethical resources to serve society.

MISSION OF THE DEPARTMENT

- To impart a robust teaching and learning process thriving on qualified, trained resources and state-of-the-art infrastructure.
- To promote innovation and research culture by providing students with hands-on experience for solving real-time problems and developing sustainable products and solutions.
- To imbibe ethical values, entrepreneurial zeal, and lifelong learning ability to develop future-ready professionals.

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development.

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required.

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions.

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems.

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment.

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws.

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences.

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change.

COURSE OUTCOMES (COs)

Course outcome: After completion of the course, the student will be able to	
CO1	Verify truth table of various types of Logic Gates.
CO2	Design, implement and verify combinational logic circuits.
CO3	Implement and verify truth table of various types of Flip- Flops.
CO4	Design and analyse different types of sequential logic circuits.
CO5	Implement programming in IoT development boards with IO sensors.

Course articulation Matrix (Mapping of COs with POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**3**	**2**	**2**
CO2	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**3**	**2**	**2**
CO3	**3**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**3**	**2**	**2**
CO4	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**3**	**2**	**2**
CO5	**3**	**3**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**2**	**2**
Average	**3**	**2.6**	**3**	**-**	**-**	**-**	**-**	**-**	**3**	**2**	**2**

Enter correlation levels 1, 2 or 3 as defined below:

1: Slight (Low) **2:** Moderate (Medium) **3:** Substantial (High)

If there is no correlation, put "-"

INSTRUCTIONS TO STUDENTS FOR WRITING THE RECORD

In the record, the index page should be filled properly by writing the corresponding experiment number, experiment name, date on which it was done and the page number.

On the **right side** page of the record following has to be written:

1. **Title:** The title of the experiment should be written in the page in capital letters.
2. In the left top margin, experiment number and date should be written.
3. **Aim:** The purpose of the experiment should be written clearly.
4. **Apparatus/Tools/Equipments/Components used:**

   A list of the Apparatus/Tools/Equipments/Components used for doing the experiment should be entered.
5. **Theory:** Simple working of the circuit/experimental set up/algorithm should be written.
6. **Procedure:** Steps for doing the experiment and recording the readings should be briefly described (flow chart/ Circuit Diagrams / programs in the case of computer/processor related experiments).
7. **Results:** The results of the experiment must be summarized in writing and should be fulfilling the aim.

On the **Left side** page of the record following has to be recorded:

1. **Circuit/Program:** Neatly drawn circuit diagrams for the experimental set up.
2. **Design:** The design of the circuit components for the experimental set up for selecting the components should be clearly shown if necessary.
3. **Observations:**
   - Data should be clearly recorded using Tabular Columns.
   - Unit of the observed data should be clearly mentioned.
   - Relevant calculations should be shown. If repetitive calculations are needed, only show a sample calculation and summarize the others in a table.

GENERAL SAFETY GUIDELINES

Following safety guidelines must be followed while performing lab experiments:

1. Student entry in the lab is ensured strictly as per the allocated time slots or seeking prior proper permission from the lab faculty or instructor.
2. Students are expected to conduct themselves in a responsible manner while working in the laboratory.
3. They should keep their bags on the shelf provided outside the lab and carry only essential items such as lab record, manual, pen-pencil, copy and calculator etc. inside the lab.
4. Students are not allowed to carry food items (not even chewing gum), beverages and water bottles while working in the laboratory.
5. They are expected to observe good housekeeping practices and ensure equipment, sitting stools and components to be handled carefully and kept at proper place after finishing the work to keep the lab clean and tidy.
6. While working in the lab
   - Avoid stretching cables and connectors while using the equipment.
   - Rig the circuit and get it verified from the lab instructor before connecting it to power source.
   - Pay proper attention towards earthing of equipment. Ensure proper ventilation in the lab while working.
   - Ensure use of wire clippers, insulating tape, plug-pins to prevent any shocking hazards.
   - In case of any short circuit, sensing burning smell or observing any smoke switch off power supply and immediately report to the faculty/lab instructor available in the lab.
7. In case of any minor injury please contact the lab instructor or lab faculty. The first aid Box is available in the department.
8. In case of any fire emergency, contact the faculty or lab instructor. For your information, the fire safety equipment is available on each floor near notice board.

DO's and DON'Ts in Laboratory

1. Come fully prepared for the experiment in the laboratory.
2. Make the connections without connecting the leads to the supply.
3. Re-check the connections and show it to the teacher /instructor before switching-on the power supply to the circuit/ trainer kit.
4. Energize the circuit only with the permission of the teacher/instructor.
5. After the experiment, disconnect the connections and put back the connecting wires/leads at appropriate place.
6. Return all the issued apparatus to the lab-staff at the end of lab session.
7. In case of shock, switch-off the power supply immediately.
8. Strictly follow the procedure given with the respective experiments.
9. Avoid loose connections.
10. Don't touch the main power supply leads with bare hand and avoid body earth.
11. Don't use the mobile phones during laboratory.
12. Strictly follow the instructions given by the teacher/Lab Instructor.

Lab Safety Guidelines

1. Earthing should be in proper condition.
2. Fuse must have correct rating.
3. Use rubber sole shoe while working.
4. Always use insulated screw driver, line testers etc.
5. First aid box must be provided and maintained.
6. Fire bucket with dry sand should be ready.
7. Insulation of conductor must be proper and good condition.
8. Each installation is periodically inspected.
9. There must be an instructor for the restoration of person suffering from shock (elementary first aid instruction).

B.TECH FIRST YEAR (SEM. I/II)			
Course Code	**CEC0151 / CEC0251**	**L T P**	**Credit**
Course Title	**Digital Electronics and IoT Systems Lab**	**0 0 2**	**1**
Name of Experiments		**CO**	
1. Verification of the truth tables of Basic Logic Gates and Universal Logic Gates using TTL ICs: **a)** AND (7408) **b)** OR (7432) **c)** NOT (7404) **d)** NAND (7400) **e)** NOR (7402)		**CO1**	
2. Implementation of the given Boolean function using TTL Logic Gates (NOT, AND and OR Gates) in SOP for following Boolean expressions: **a)** Y1 = AB' +A'B **b)** Y2 = ABC + A'B'C' + A'C **c)** F(A, B, C, D) = ∑(0, 2, 5, 7, 8, 10, 13, 15)		**CO1**	
3. Implementation of the given Boolean function using TTL Logic Gates (NOT, AND and OR Gates) in POS forms for following Boolean expressions: **a)** Y1 = (A'+B)(A+B') **b)** Y2 = (A+B+C)(A'+B'+C')(A'+C) **c)** F(A, B, C, D) = M(0,2,5,7,8,10,12,15)		**CO1**	
4. Implementation of Half-adder, Full-adder and Full-adder using two Half-adder with TTL Logic Gates (EXOR-7486, AND-7408, OR-7432) and verify its truth table.		**CO2**	
5. Implementation of Half-subtractor, Full-subtractor and Full-subtractor using two Half-subtractor with TTL Logic Gates (EXOR-7486, AND-7408, OR-7432) and verify its truth table.		**CO2**	
6. Implement 2 Bit magnitude comparator using Logic Gates and verify the truth table.		**CO2**	
7. Implement and verify F (A, B, C) = ∑ (3, 5, 6, 7) using: **a)** 8:1 multiplexer **b)** 4:1 multiplexer		**CO2**	
8. Verification of truth table of Flip-Flop using NAND gate (7400) & NOR gates (7402). **a)** RS Flip-Flop **b)** JK Flip-Flop **c)** D Flip-Flop **d)** T Flip-Flop		**CO3**	
9. Implement D Flip-Flop using SR Flip-Flop and verify the truth table.		**CO3**	

10. Design Mod–N Synchronous Up Counter & Down Counter using 7476 JK Flip-Flop.	**CO4**
11. Describing hardware in IoT: Hardware Architecture of Arduino UNO Board, Types of Arduino Board.	**CO5**
12. Fundamentals of Arduino Programming: Installation of Arduino IDE, Working with structures, Variables, Flow control, Digital I/O, f. Analog I/O, Time, Math, Random, Serial.	**CO5**
13. Interfacing Arduino with I/O Devices: Push button, LED, Ultrasonic Sensor.	**CO5**

EXPERIMENT NO: 1

OBJECTIVE: Verification of the truth tables of Basic Logic Gates and Universal Logic Gates using TTL ICs.

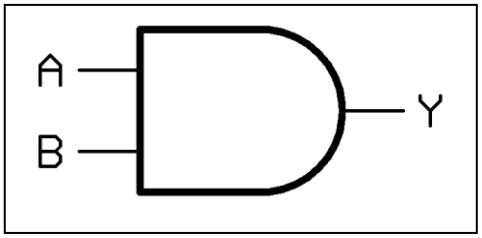
a) AND (7408)
b) OR (7432)
c) NOT (7404)
d) NAND (7400)
e) NOR (7402)

APPARATUS REQUIRED:

S. No.	COMPONENT	SPECIFICATION	QTY
1	AND GATE	IC 7408	1
2	OR GATE	IC 7432	1
3	NOT GATE	IC 7404	1
4	NAND GATE 2 I/P	IC 7400	1
5	NOR GATE	IC 7402	1
6	X-OR GATE	IC 7486	1
7	X-NOR GATE	IC74266	1
8	IC TRAINER KIT	…	1
9	PATCH CORD	…..	1

THEORY: Logic gates are electronic circuits which perform logical functions on one or more inputs to produce one output. There are seven logic gates. When all the input combinations of a logic gate are written in a series and their corresponding outputs written along them, then this input/ output combination is called Truth Table. OR, AND and NOT are basic gates. NAND, NOR are known as universal gates.

AND GATE (IC7408): The AND gate performs a logical multiplication commonly known as AND functions. The output is high when both the inputs are high. The output is low level when any one of the inputs is low.



Inputs		Outputs
A	**B**	**Y**
0	0	0
0	1	0
1	0	0
1	1	1

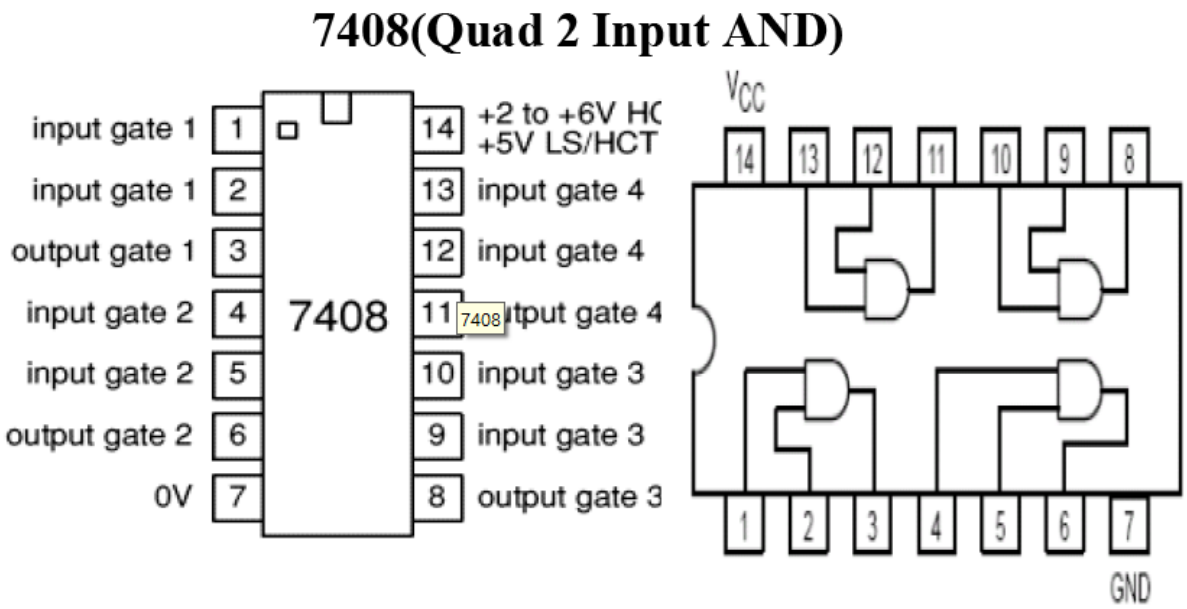
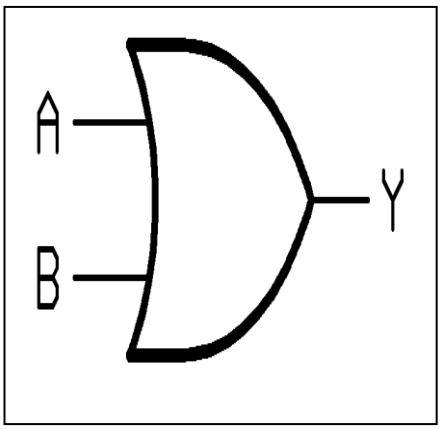


Figure 1.1 Pin diagram of 2 input AND (IC7408) Gate

OR GATE (IC7432):

The OR gate performs a logical addition commonly known as OR function. The output is high when any one of the inputs is high. The output is low level when both the inputs are low.



Inputs		Outputs
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

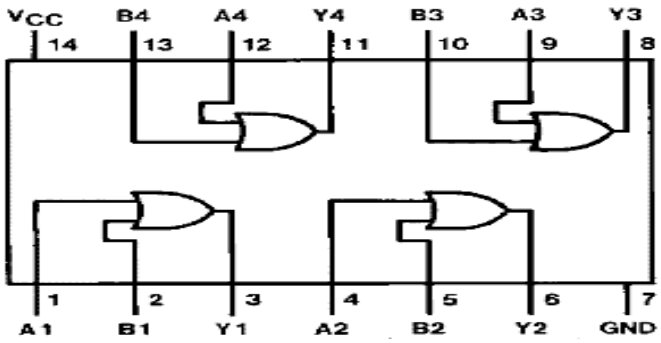
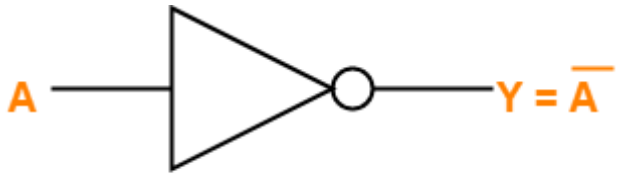


Figure 1.2 Pin diagram of 2 input OR (IC 7432) Gate

NOT GATE (IC 7404): The NOT gate called as an inverter. The output is high when input is low. The output is low when input is high.



Inputs	Outputs
A	Y
1	0
0	1

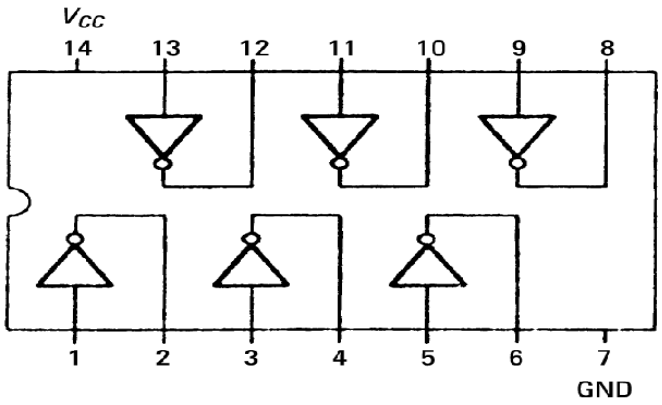
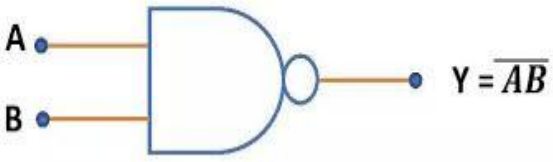


Figure 1.3 Pin diagram of 2 input NOT (IC 7404) Gate

NAND GATE (IC 7400):

NAND gate is a combination of AND & NOT gate. The output is high when both the inputs are low and any of the input is low. The output is low when both the inputs are high.



Inputs		Outputs
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

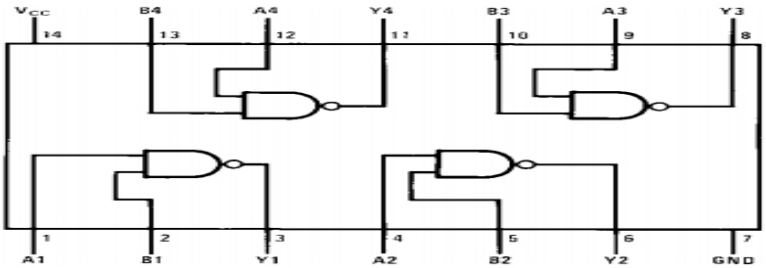
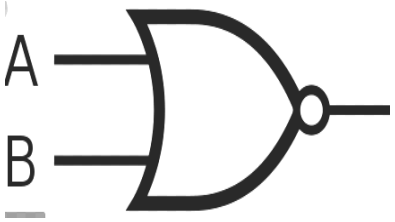


Figure 1.4 Pin diagram of 2 input NAND Gate (IC 7400)

NOR GATE (IC7402):

The NOR gate is a contraction of OR-NOT. The output is high when both inputs are low. The output is low when one or both inputs are high.



INPUTS		OUTPUT
A	B	X=(A+B)'
0	0	1
0	1	0
1	0	0
1	1	0

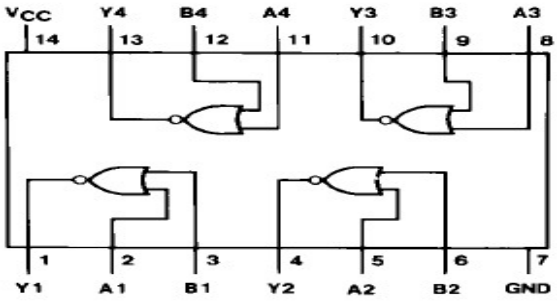


Figure 1.5 Pin diagram of 2 input NOR (IC7402) Gate

X-OR GATE (IC 7486):

The output Q is true if either input A is true or input B is true. But not when both of them are true. The expression for EX-OR gate is:

X = A'B + AB'

INPUTS		OUTPUT
A	B	X=A'B+AB'
0	0	0
0	1	1
1	0	1
1	1	0

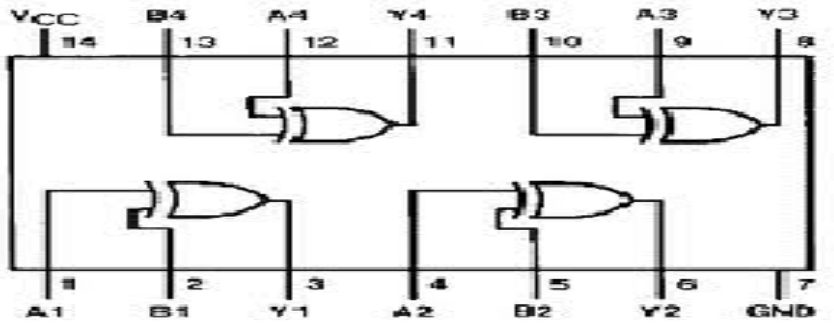


Figure 1.6 Pin diagram of 2 input XOR (IC 7486) Gate

X-NOR GATE (IC 74266):

EX-NOR gate (sometimes spelled "Ex-nor" or "Ex-nor "and rarely written NXOR) is a digital logic gate whose function is the inverse of the exclusive OR (XOR) gate. A HIGH output (1) results if both of the inputs to the gate are the same. If one but not both inputs are HIGH (1), a LOW output (0) results.

B1	B2	Y1
0	0	1
0	1	0
1	0	0
1	1	1

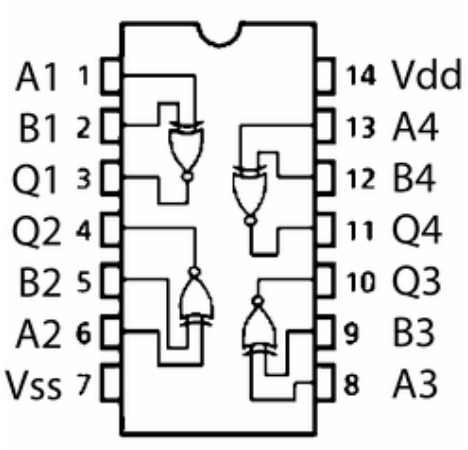


Figure 1.7 Pin diagram of 2 input XNOR (IC 74266) Gate

RESULT: All the Logic Gates have been successfully studied and verified.

PRECAUTIONS:

1) Make the connections according to the IC pin diagram.
2) The connections should be tight.
3) The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer:

Q1. What is Digital Gate?

Ans: Digital gates are basically electronic components which are used for switching and manipulating binary data.

Q2. What is the use of logic gates?

Ans: In electronics, a logic gate is an idealized or physical device implementing a Boolean function; that is, it performs a logical operation on one or more binary inputs and produces a single binary output.

Q3: What is truth table?

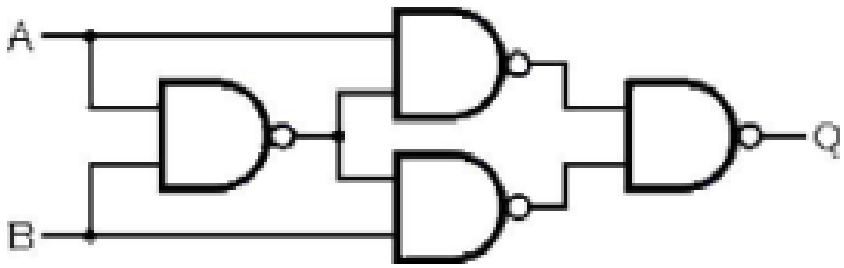
Ans: Truth table is a table from which we can get o/p of different gates

Q4. What is universal gate?

Ans: A universal gate is a gate which can implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates. In practice, this is advantageous since NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families.

Q5. Draw the EX-OR gate by using the NAND gate?

Ans:



EXPERIMENT NO: 2

OBJECTIVE: Implementation of the given Boolean function using TTL Logic Gates (NOT, AND and OR Gates) in SOP for following Boolean expressions:

a) Y1 = AB' + A'B for SOP

b) Y2 = ABC + A'B'C' + A'C

c) F(A, B, C, D)=∑(0, 2, 5, 7, 8, 10, 13, 15)

APPARATUS REQUIRED:

S. No.	COMPONENT	SPECIFICATION	QTY
1	AND GATE	IC 7408	2
2	OR GATE	IC 7432	2
3	NOT GATE	IC 7404	2
4	IC TRAINER KIT	…	1
5	BREAD BOARD		1
6	POWER SUPPLY	0-30 V DC	1
7	PATCH CORD, CONNECTING WIRES	…..	

THEORY:

A Boolean function is an algebraic form of Boolean expression. A Boolean function of n-variables is represented by $f(x_1, x_2, x_3 \ldots x_n)$. By using Boolean laws and theorems, we can simplify the Boolean functions of digital circuits. A different ways of representing a Boolean function is given below.

- Sum-of-Products (SOP) Form
- Product-of-sums (POS) form
- Canonical forms

There are two types of canonical forms:

- Sum-of-min terms or Canonical SOP
- Product-of- max terms or Canonical POS

Boolean functions can be represented by using NAND gates and also by using K-map (Karnaugh map) method. We can standardize the Boolean expressions by using by two standard forms.

SOP form – Sum of Products form

POS form – Product of Sums form

SUM OF PRODUCT (SOP) FORM: The sum-of-products (SOP) form is a method (or form) of simplifying the Boolean expressions of logic gates. In this SOP form of Boolean function representation, the variables are operated by AND (product) to form a product term and all these product terms are ORed (summed or added) together to get the final function.

A sum-of-products form can be formed by adding (or summing) two or more product terms using a Boolean addition operation. Here the product terms are defined by using the AND operation and the sum term is defined by using OR operation.

The sum-of-products form is also called as Disjunctive Normal Form as the product terms are ORed together and Disjunction operation is logical OR. Sum-of-products form is also called as Standard SOP.

Implementation of Y_1= AB' +A'B

TRUTH TABLE:

A	B	A'	B'	AB'	A'B	Y_1=AB'+A'B
0	0	1	1	0	0	**0**
0	1	1	0	0	1	**1**
1	0	0	1	1	0	**1**
1	1	0	0	1	0	**0**

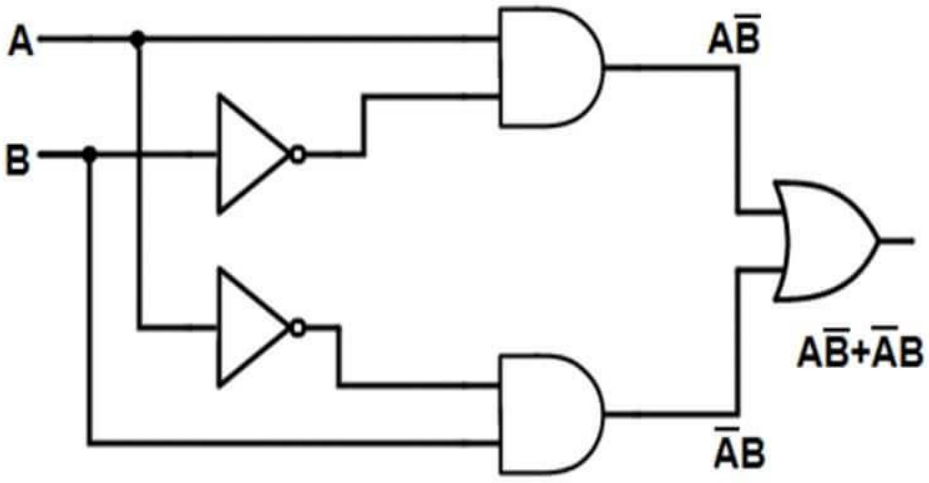


Figure 2.1 Implementation of SOP Form

Implementation of Y_2=ABC + A'B'C' + A'C

A	B	C	A'	B'	C'	ABC	A'B'C'	A'C	Y_2=ABC + A'B'C' + A'C
0	0	0	1	1	1	0	1	0	**1**
0	0	1	1	1	0	0	0	1	**1**
0	1	0	1	0	1	0	0	0	**0**
0	1	1	1	0	0	0	0	1	**1**
1	0	0	0	1	1	0	0	0	**0**
1	0	1	0	1	0	0	0	0	**0**
1	1	0	0	0	1	0	0	0	**0**
1	1	1	0	0	0	1	0	0	**1**

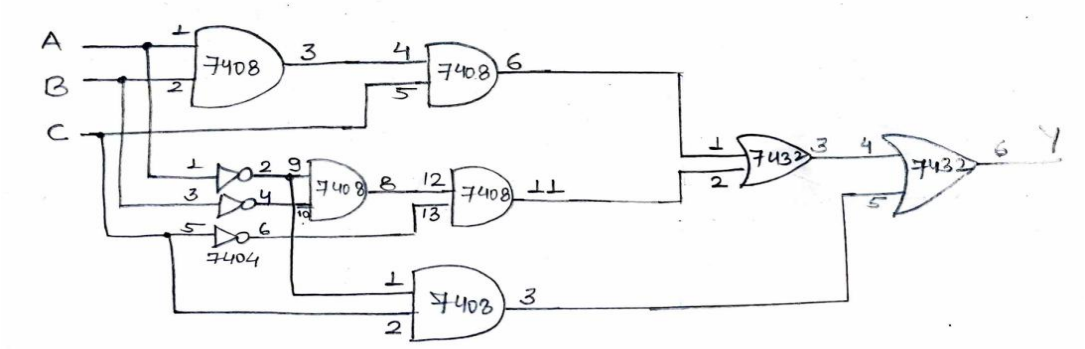
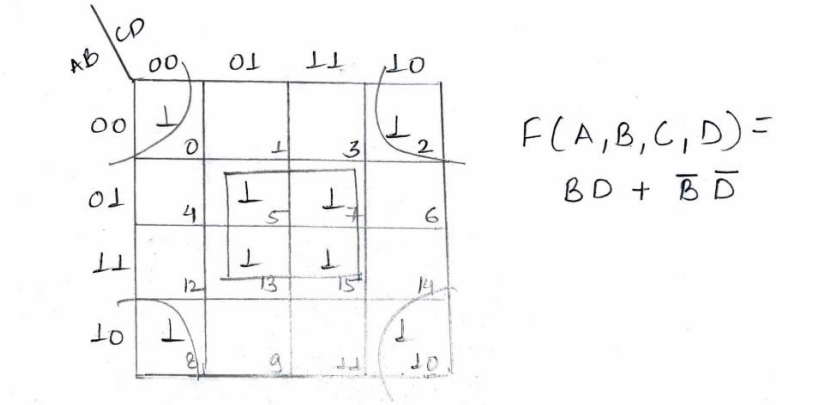


Figure 2.2 Implementation of SOP Form

Implementation of F(A,B,C,D)=∑(0,2,5,7,8,10,13,15)



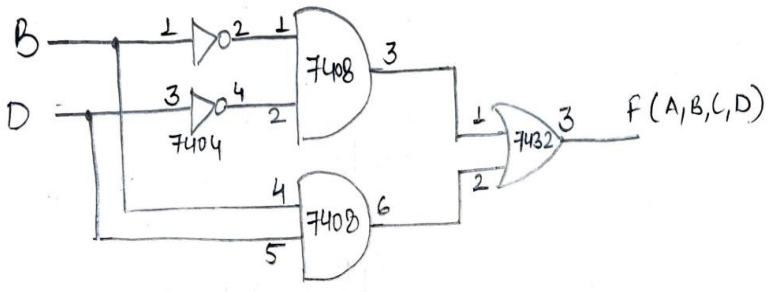


Figure 2.3 Implementation of SOP Form

RESULT: Implementation of various SOP forms has been studied and verified.

PRECAUTIONS:

1) Make the connections according to the IC pin diagram.
2) The connections should be tight.
3) The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer:

Q1. What is SOP and POS form?

Ans: The term "Sum of Products" or "SOP" is widely used for the canonical form that is a disjunction (OR) of minterms. Its De Morgan dual is a "Product of Sums" or "POS" for the canonical form that is a conjunction (AND) of maxterms.

Q2: What is difference between SOP and POS?

Ans: The main difference between SOP and POS is that the SOP is a way of representing a Boolean expression using min terms or product terms while the POS is a way of representing a Boolean expression using max terms or sum terms

Q3. What is SOP K map?

Ans: A minterm is a Boolean expression resulting in 1 for the output of a single cell, and 0's for all other cells in a Karnaugh map, or truth table.

Q4. Convert the following SOP expression to an equivalent POS expression.

Ans: $A\,B\,C + A\,\overline{B}\,\overline{C} + A\,\overline{B}\,C + A\,B\,\overline{C} + \overline{A}\,\overline{B}\,C$

$(A + B + C)(A + \overline{B} + C)(A + \overline{B} + \overline{C})$

Q5. What is canonical SOP?

Ans: Canonical SOP form means Canonical Sum of Products form. In this form, each product term contains all literals. So, these product terms are nothing but the min terms. Hence, canonical SOP form is also called as sum of min terms form.

EXPERIMENT NO: 3

OBJECTIVE: Implementation of the given Boolean function using TTL Logic Gates (NOT, AND and OR Gates) in POS forms for following Boolean expressions:

a) Y1 = (A'+B)(A+B')

b) Y2 = (A+B+C)(A'+B'+C')(A'+C)

c) F(A, B, C, D) = M(0, 2, 5, 7, 8, 10, 12, 15)

APPARATUS REQUIRED:

S. No.	COMPONENT	SPECIFICATION	QTY
1	AND GATE	IC 7408	1
2	OR GATE	IC 7432	1
3	NOT GATE	IC 7404	1
4	IC TRAINER KIT	…	1
5	BREAD BOARD		1
6	POWER SUPPLY	0-30 V DC	1
7	CONNECTING WIRES	…..	

THEORY:

A Boolean function is an algebraic form of Boolean expression. A Boolean function of n-variables is represented by f(x1, x2, x3….xn). By using Boolean laws and theorems, we can simplify the Boolean functions of digital circuits. A different ways of representing a Boolean function is given below.

- Sum-of-Products (SOP) Form
- Product-of-sums (POS) form
- Canonical forms

There are two types of canonical forms:

- Sum-of-min terms or Canonical SOP
- Product-of- max terms or Canonical POS

Boolean functions can be represented by using NAND gates and also by using K-map (Karnaugh map) method. We can standardize the Boolean expressions by using by two standard forms.

SOP form – **S**um **O**f **P**roducts form

POS form – **P**roduct **O**f **S**ums form

PRODUCT OF SUM (POS) FORM:

The product of sums form is a method (or form) of simplifying the Boolean expressions of logic gates. In this POS form, all the variables are ORed, i.e. written as sums to form sum terms. All these sum terms are ANDed (multiplied) together to get the product-of-sum form. This form is exactly opposite to the SOP form. So this can also be said as "Dual of SOP form".

Here the sum terms are defined by using the OR operation and the product term is defined by using AND operation. When two or more sum terms are multiplied by a Boolean OR operation, the resultant output expression will be in the form of product-of-sums form or POS form.

The product-of-sums form is also called as Conjunctive Normal Form as the sum terms are ANDed together and Conjunction operation is logical AND. Product-of-sums form is also called as Standard POS.

Implementation of Y_1= (A'+B) (A+B')

A	B	A'	B'	A'+B	A+B'	Y_1= (A'+B) (A+B')
0	0	1	1	1	1	**1**
0	1	1	0	1	0	**0**
1	0	0	1	0	1	**0**
1	1	0	0	1	1	**1**

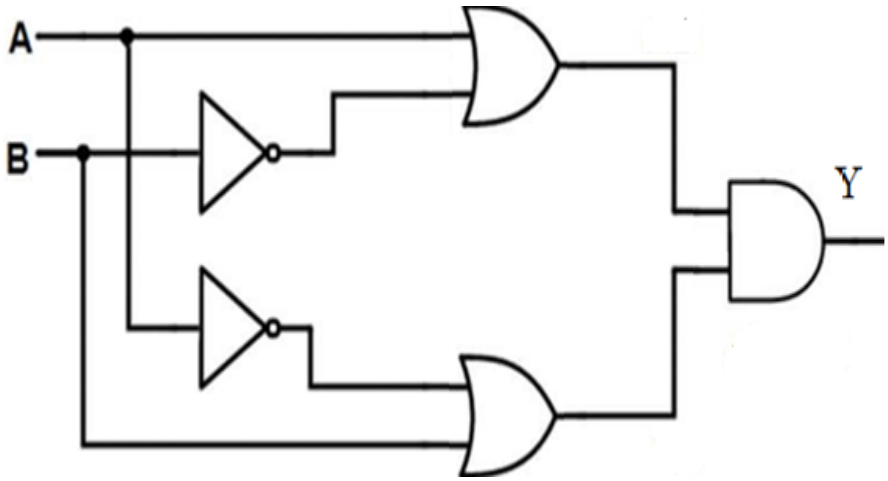


Figure 3.1 Implementation of POS Form

Implementation of Y2= (A+B+C)(A'+B'+C')(A'+C)

A	B	C	A'	B'	C'	A+B+C	A'+B'+C'	A'+C	Y2 = (A+B+C) (A'+B'+C') (A'+C)
0	0	0	1	1	1	0	1	1	**0**
0	0	1	1	1	0	1	1	1	**1**
0	1	0	1	0	1	1	1	1	**1**
0	1	1	1	0	0	1	1	1	**1**
1	0	0	0	1	1	1	1	0	**0**
1	0	1	0	1	0	1	1	1	**1**
1	1	0	0	0	1	1	1	0	**0**
1	1	1	0	0	0	1	0	1	**0**

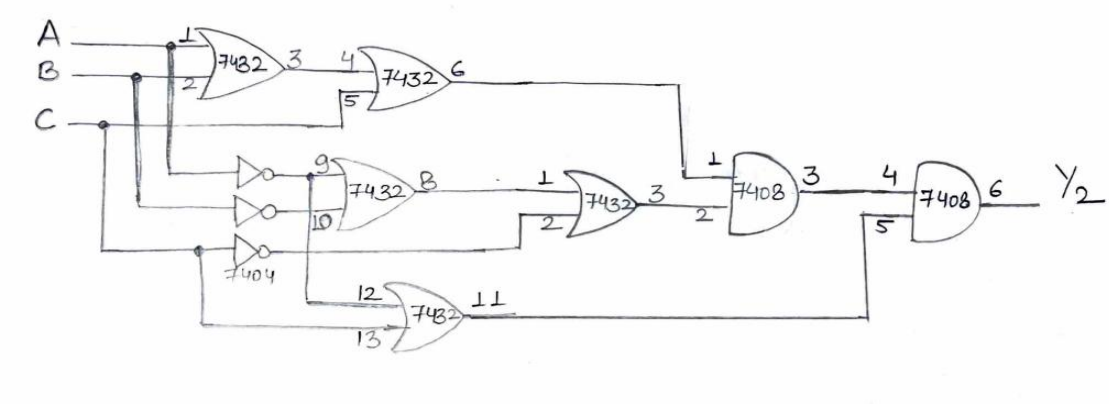
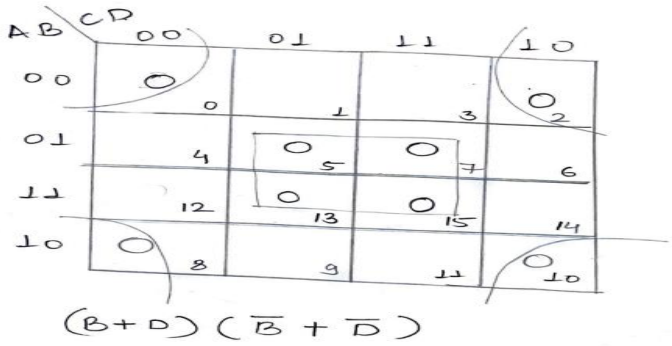


Figure 3.2 Implementation of POS Form

Implementation of *F(A,B,C,D)= M(0,2,5,7,8,10,12,15)*



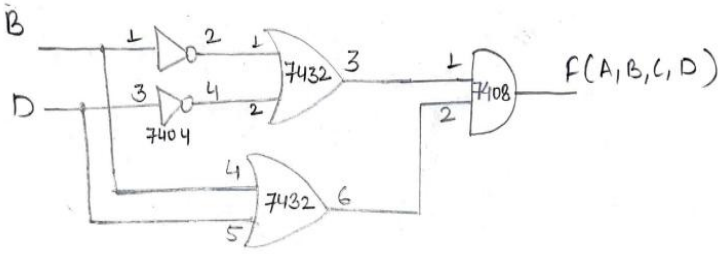


Figure 3.3 Implementation of POS Form

RESULT: : Implementation of various POS forms has been studied and verified.

PRECAUTIONS:

1) Make the connections according to the IC pin diagram.
2) The connections should be tight.
3) The V_{cc} and ground should be applied carefully at the specified pin only.

Quiz Questions with answer:

Q1. What is SOP and POS form?

Ans: The term "Sum of Products" or "SOP" is widely used for the canonical form that is a disjunction (OR) of minterms. Its De Morgan dual is a "Product of Sums" or "POS" for the canonical **form** that is a conjunction (AND) of maxterms.

Q2. What is difference between SOP and POS?

Ans: The main difference between SOP and POS is that the SOP is a way of representing a Boolean expression using min terms or product terms while the POS is a way of representing a Boolean expression using max terms or sum terms

Q3. What is SOP K map?

Ans: A minterm is a Boolean expression resulting in 1 for the output of a single cell, and 0's for all other cells in a Karnaugh map, or truth table.

Q4. Convert the following SOP expression to an equivalent POS expression.

Ans: $A\,B\,C + A\,\overline{B}\,\overline{C} + A\,\overline{B}\,C + A\,B\,\overline{C} + \overline{A}\,\overline{B}\,C$

$(A + B + C)(A + \overline{B} + C)(A + \overline{B} + \overline{C})$

Q5. What is canonical SOP?

Ans: Canonical SOP form means Canonical Sum of Products form. In this form, each product term contains all literals. So, these product terms are nothing but the min terms. Hence, canonical SOP form is also called as sum of min terms form.

EXPERIMENT NO: 4

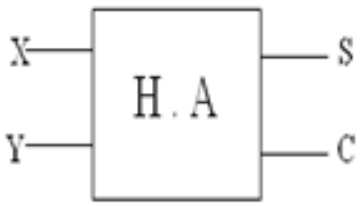
OBJECTIVE: Implementation of Half-adder, Full-adder and Full-adder using two Half-adder with TTL Logic Gates (EXOR-7486, AND-7408, OR-7432) and verify its truth table.

APPARATUS REQUIRED:

S. No.	COMPONENT	SPECIFICATION	QTY
1	AND GATE	IC 7408	2
2	OR GATE	IC 7432	2
3	X-OR GATE	IC 7486	2
4	IC TRAINER KIT	…	1
5	BREAD BOARD		1
6	POWER SUPPLY	0-30V DC	1
7	PATCH CORD, CONNECTING WIRES	…..	

THEORY: Half adder is a combinational arithmetic circuit that adds two numbers and produces a sum bit(S) and carry bit (C) as the output. If A and B are the input bits, then sum bit(S) is the X-OR of A and B and the carry bit (C) will be the AND of A and B.

Half Adder: The block diagram of half adder is:



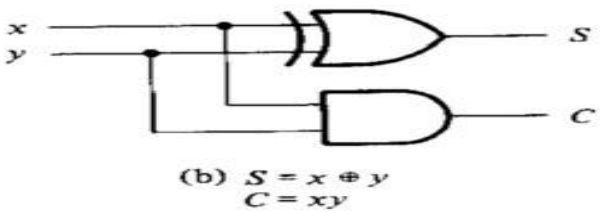
Inputs		Outputs	
X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Figure 4.1 Block diagram and Truth Table of Half adder

Boolean equations for sum and carry are:

$$S = \bar{X}Y + X\bar{Y}$$

$$C = XY$$



(b) $S = x \oplus y$
$C = xy$

Figure 4.2 Logic diagram of half adder

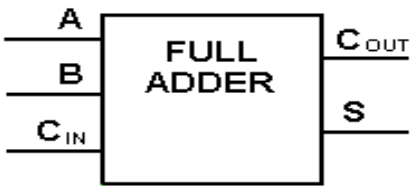
Full Adder:

Full adder is developed to overcome the drawbacks of Half Adder circuit. It can add two one-bit numbers A and B, sum S and carry C. The full adder is a three input and two output combinational circuit.

Truth Table for full adder

Inputs			Outputs	
A	**B**	**C_{in}**	**Sum**	**Carry**
0	0	0	**0**	**0**
0	0	1	**1**	**0**
0	1	0	**1**	**0**
0	1	1	**0**	**1**
1	0	0	**1**	**0**
1	0	1	**0**	**1**
1	1	0	**0**	**1**
1	1	1	**1**	**1**

The block diagram of full adder is:



Boolean equations for Sum and Carry are:

$$\textbf{Sum} = A \text{ XOR } B \text{ XOR } C_{in}$$

$$\textbf{Carry} = AB + BC_{in} + C_{in}A$$

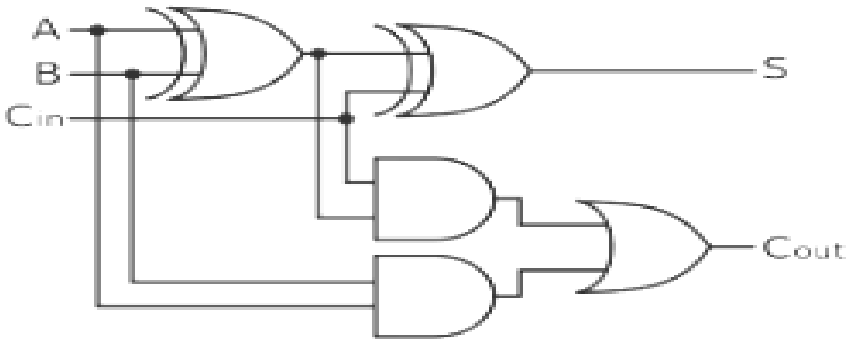


Figure 4.3 Logic diagram of full adder

Full Adder (By two half adder):

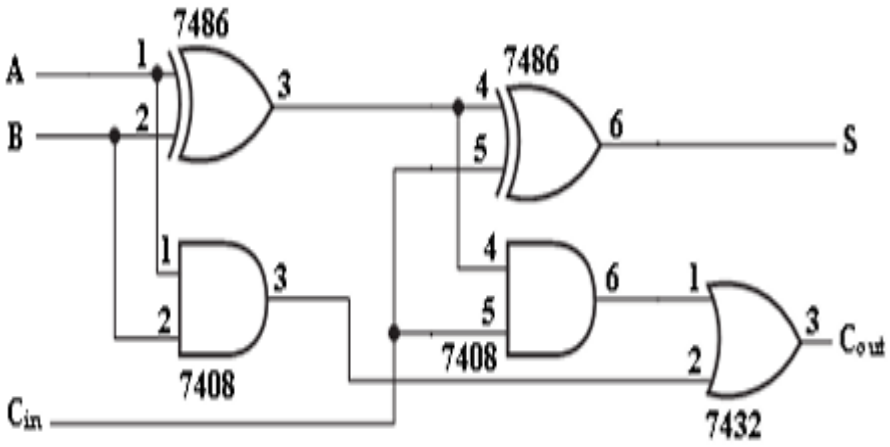


Figure 4.4 Logic diagram of full adder by two half adder

Truth Table for full adder is:

Inputs			**Outputs**	
A	**B**	**C_{in}**	**Sum**	**Carry**
0	0	0	**0**	**0**
0	0	1	**1**	**0**
0	1	0	**1**	**0**
0	1	1	**0**	**1**
1	0	0	**1**	**0**
1	0	1	**0**	**1**
1	1	0	**0**	**1**
1	1	1	**1**	**1**

RESULT: Half adder and Full adder has been studied and verified.

PRECAUTIONS:

1) Make the connections according to the IC pin diagram.
2) The connections should be tight.
3) The V_{cc} and ground should be applied carefully at the specified pin only.

EXPERIMENT NO: 5

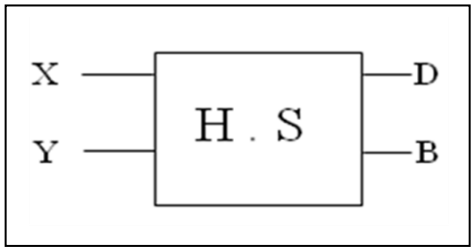
OBJECTIVE: Implementation of Half-subtractor, Full-subtractor and Full-subtractor using two Half-subtractor with TTL Logic Gates (EXOR-7486, AND-7408, OR-7432) and verify its truth table.

APPARATUS REQUIRED:

S. No.	COMPONENT	SPECIFICATION	QTY
1	AND GATE	IC 7408	2
2	OR GATE	IC 7432	2
3	X-OR GATE	IC 7486	2
4	NOR GATE	IC 7402	1
5	IC TRAINER KIT	…	1
6	BREAD BOARD		1
7	POWER SUPPLY	0-30V DC	1
8	PATCH CORD, CONNECTING WIRES	…..	

THEORY: Half Subtractor is an arithmetic circuit that subtracts two binary numbers from each other, for example X-Y to find the resulting difference between the two numbers. Half subtractor produces a difference (D) by using a borrow bit (B) from the previous column.

The block diagram of Half Subtractor is:



Inputs		Outputs	
X	Y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Figure 5.1 Block diagram and Truth Table of Half Subtractor

Boolean equations for Difference and Borrow are:

$$D = \bar{X}Y + X\bar{Y} = X \oplus Y$$

$$B = \bar{X}Y$$

Logic Diagram of Half Subtractor:

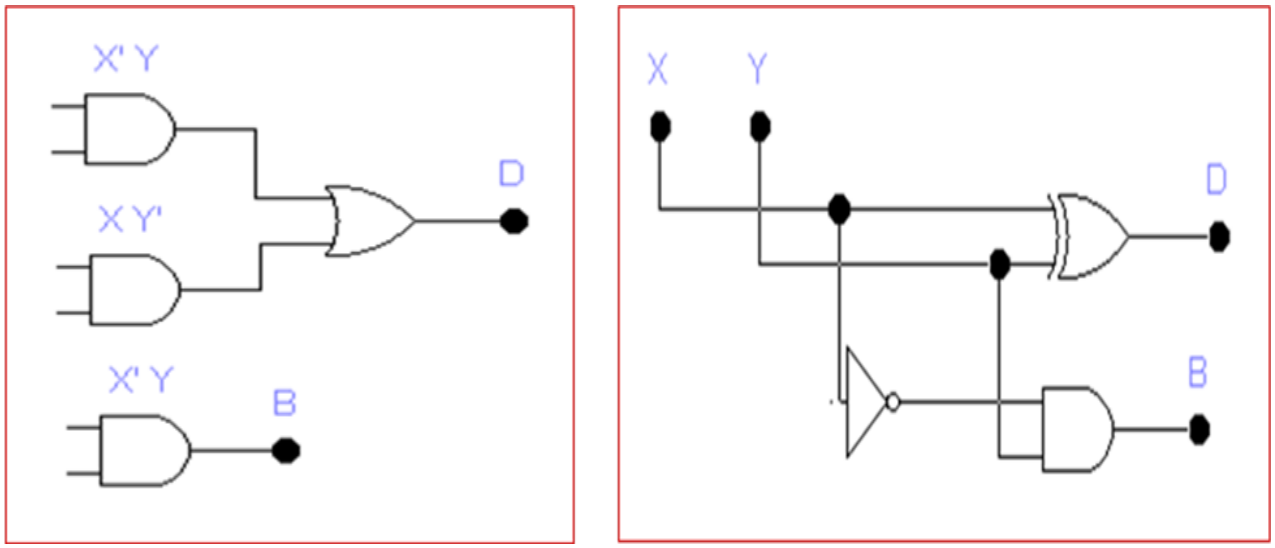


Figure 5.2 Logic diagram of Half Subtractor

Full Subtractor: A full subtractor is a combinational circuit that performs subtraction of two bits, one is minuend and other is subtrahend, taking into account borrow of the previous adjacent lower minuend bit. This circuit has three inputs and two outputs. The three inputs A, B and Bin, denote the minuend, subtrahend, and previous borrow, respectively. The two outputs, D and Bout represent the difference and output borrows.

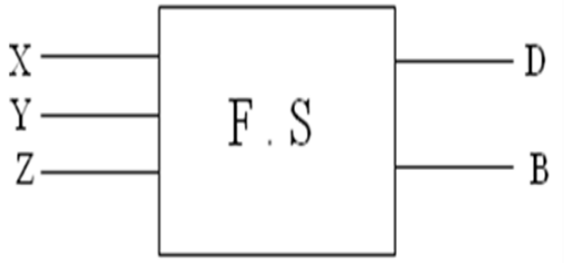


Figure 5.3 Block diagram of Half Subtractor

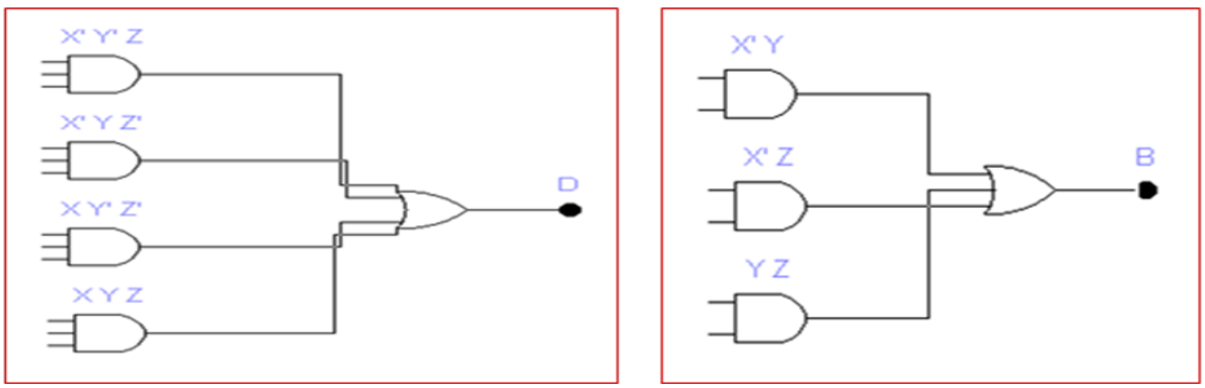


Figure 5.4 Block diagram of Full Subtractor

Truth Table: Note that, Here X, Y, Z is similar to A, B, B_{in}.

X = A	Y = B	Z = B_{in}	D	B_{out}
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

From the above truth table, we can find the Boolean expression

$$D = A \oplus B \oplus Bin$$

$$Bout = A' Bin + A' B + B Bin$$

FULL SUBTRACTOR WITH TWO HALF SUBTRACTORS

A full subtractor can be realized using two half subtractors. It will take two half-subtractors and one OR gate. The logic circuit diagram of the full subtractor using two half subtractors is shown in Figure

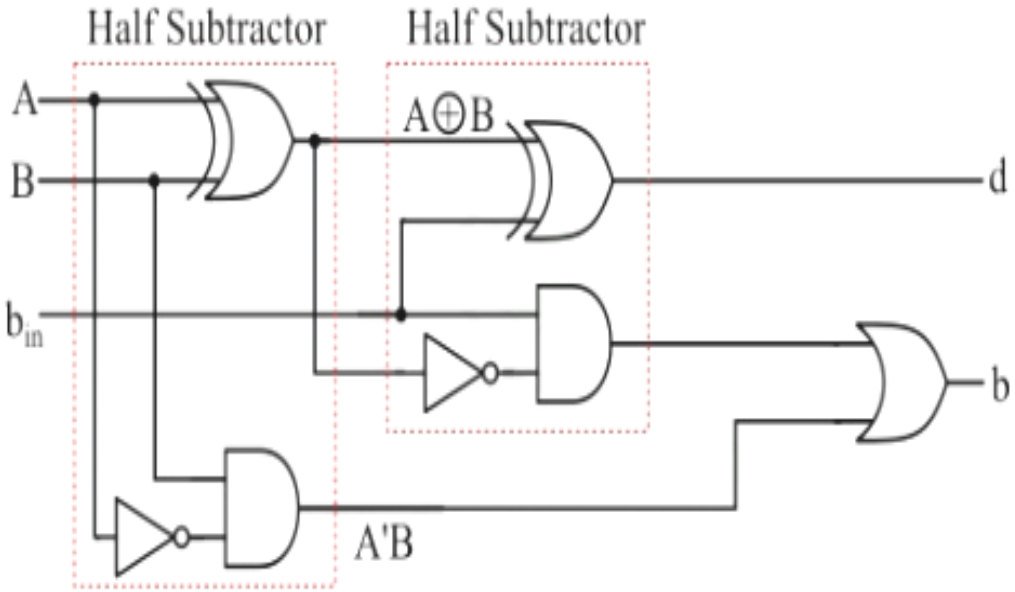


Figure 5.5 Block diagram of Full Subtractor with two Half Subtractor

$$D = (A \text{ XOR } B) \text{ XOR } Bin$$

$$Bout = Bin\ (A \text{ XOR } B)' + A'B$$

TRUTH TABLE FOR FULL SUBTRACTOR

Inputs			Outputs	
A	**B**	**B_{in}**	**Difference(D)**	**Borrow(B_{out})**
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

RESULT: Half subtractor and Full subtractor have been studied and verified.

PRECAUTIONS:

1) Make the connections according to the IC pin diagram.
2) The connections should be tight.
3) The V_{cc} and ground should be applied carefully at the specified pin only.

EXPERIMENT NO: 6

OBJECTIVE: Implement 2 Bit magnitude comparator using Logic gates and verify the truth table.

APPARATUS REQUIRED:

S. No.	COMPONENTS	SPECIFICATION	QTY
1	IC TRAINER KIT		1
2	EX-OR GATE	IC7486	1
3	NOT GATE	IC7404	1
4	OR GATE	IC7432	1
5	AND GATE	IC7408	1
6	COMPARATOR	IC7485	1
7	POWER SUPPLY	0-30 V DC	1
8	BREAD BOARD		1
9	PATCH CORD, CONNECTING WIRES		

THEORY:

A **magnitude comparator** is a combinational circuit that compares two given numbers (A and B) and determines whether one is equal to, less than or greater than the other. The output is in the form of three binary variables representing the conditions A = B, A>B and A<B, if A and B are the two numbers being compared. The two binary numbers A and B with two digits each, written in descending order as,

$$\mathbf{A = A_1 A_0 \quad and \quad B = B_1 B_0}$$

Each subscripted letter represents one of the digits in the number. It is observed from the bit contents of two numbers that A = B, when $A_1 = B_1$ and $A_0 = B_0$. When the numbers are binary they possess the value of either 1 or 0, the equality relation of each pair can be expressed logically by the equivalence function as,

$$Xi = AiBi + Ai'Bi' \qquad \text{for } i = 1, 2, 3, 4.$$

$$\text{Or,} \quad Xi = (A \oplus B)' \qquad \text{or, } Xi' = A \oplus B$$

$$\text{Or,} \quad Xi = (AiBi' + Ai'Bi)'$$

Where, $X_i = 1$ only if the pair of bits in position *i* are equal.

LOGIC DIAGRAM:

2-bit Magnitude Comparator:

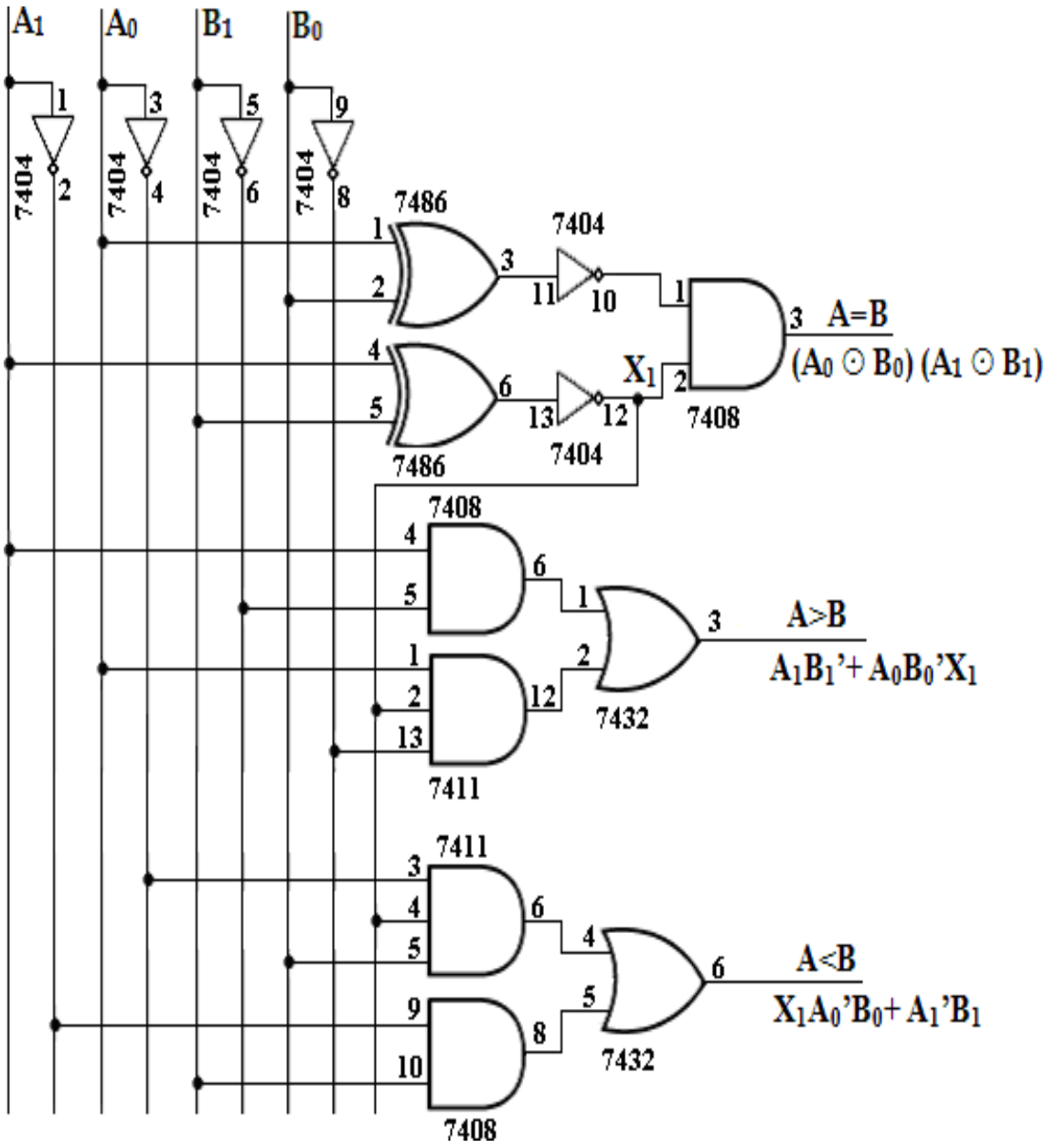


Figure 6.1 2- Bit Magnitude Comparator

- To satisfy the equality condition of two numbers A and B, it is necessary that all X*i* must be equal to logic 1. This indicates the AND operation of all X*i* variables. In other words, we can write the Boolean expression for two equal 2-bit numbers.

$$(A = B) = X_1\ X_0$$

The binary variable (A=B) is equal to 1 only if all pairs of digits of the two numbers are equal.

- To determine if A is greater than or less than B, we inspect the relative magnitudes of pairs of significant bits starting from the most significant bit. If the two digits of the most significant position are equal, the next significant pair of digits is compared. The comparison process is continued until a pair of unequal digits is found. It may be concluded that;

i. A>B, if the corresponding digit of A is 1 and B is 0.

ii. A<B, if the corresponding digit of A is 0 and B is 1.

Therefore, we can derive the logical expression of such sequential comparison by the following two Boolean functions,

$$(A>B) = A_1B_1' + X_1A_0B_0'$$

$$(A<B) = A_1'B_1 + X_1A_0'B_0$$

The symbols (A>B) and (A<B) are binary output variables that are equal to 1 when A>B or A<B, respectively.

Truth Table:

A	B	A>B	A=B	A<B
0 0 0 0	**0 0 0 0**	**0**	**1**	**0**
0 0 0 1	**0 0 0 1**	**1**	**0**	**0**
0 0 0 0	**0 0 0 1**	**0**	**0**	**1**

PROCEDURE:

1) Connections are given as per the logic diagram.
2) Logic inputs are given as per the truth table.
3) Observe the logic output and verify with the truth tables.

RESULT: Thus the 2-bit magnitude comparator was designed and implemented using logic gates and IC7485.

PRECAUTIONS:

1) Make the connections according to the IC pin diagram.
2) The connections should be tight.
3) The V_{cc} and ground should be applied carefully at the specified pin only.

EXPERIMENT NO: 7

OBJECTIVE: Implement and verify F (A, B, C) = ∑ (3, 5, 6, 7) using:

a) 8:1 multiplexer

b) 4:1 multiplexer

APPARATUS REQUIRED:

S. No.	COMPONENTS	SPECIFICATION	QTY
1	IC TRAINER KIT		1
2	NOT GATE	IC7404	1
3	OR GATE	IC7432	1
4	AND GATE	IC7408	1
5	POWER SUPPLY	0-30 V DC	1
6	BREAD BOARD		1
7	CONNECTING WIRES		

THEORY:

(a) Using 8x1 multiplexer: A1, A2, A3 will be select lines.

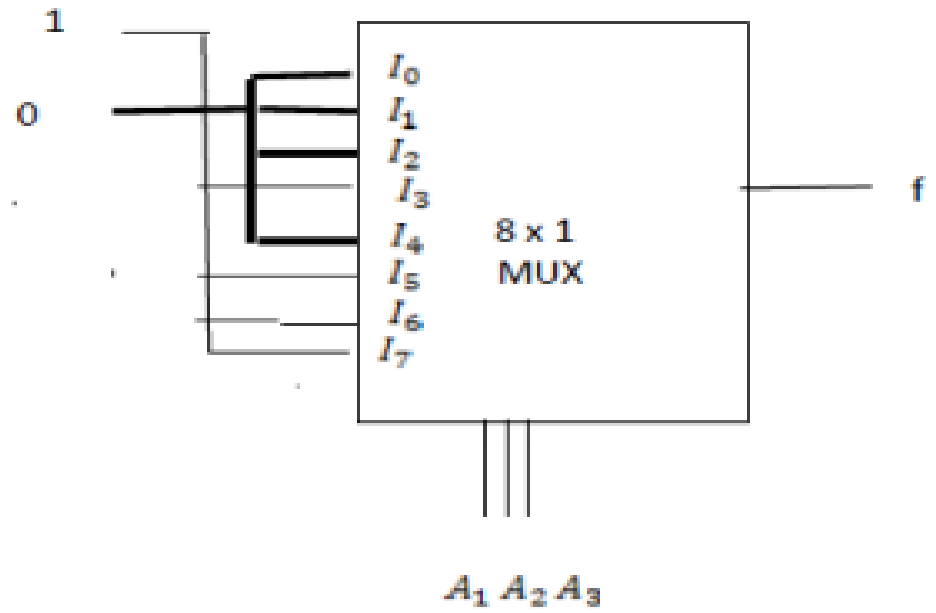
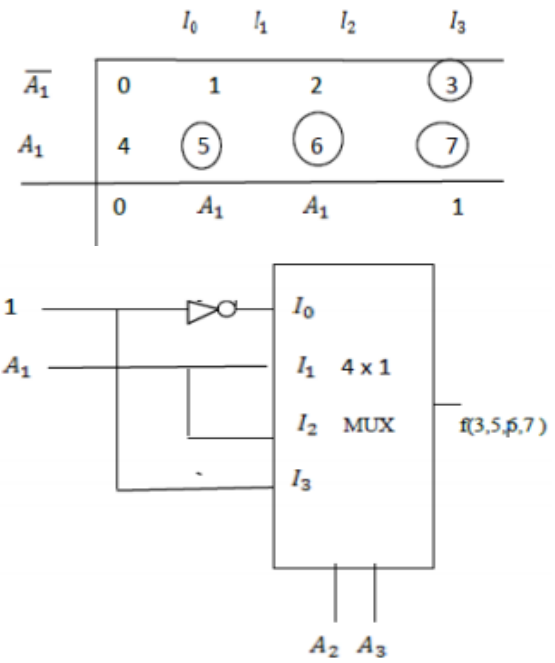


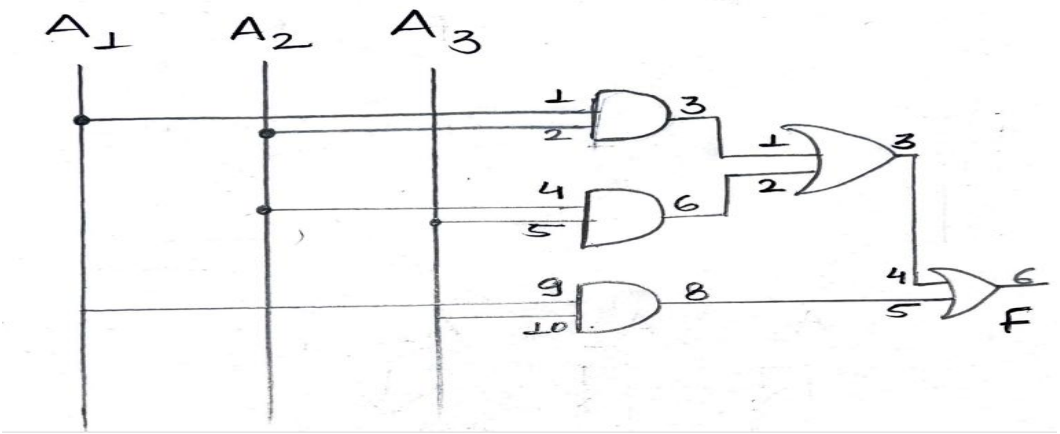
Figure 7.1 8:1 Multiplexor Implementation

(b) Implement F (A1, A2, A3) = ∑ (3, 5, 6, 7) using 4x1 multiplexer:

If MSB i.e. A1 is used as input variable and A2, A3 as select lines.

Minterms	A_1	A_2	A_3	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1





F(A,B,C) = A_1A_2 + A_2A_3 + A_1A_3

Figure 7.2 4:1 Multiplexor Implementation

RESULT: Implementation of F (A, B, C) = ∑ (3, 5, 6, 7) using 8:1 multiplexer and 4:1 multiplexer has successfully studied and expression verified.

PRECAUTIONS:

1) All connections should be made neat and tight.
2) Power supply and ICs should be handled with carefully.
3) While making connections main supply should be kept switched off.
4) Never touch live and naked wires.

EXPERIMENT NO: 8

OBJECTIVE: Verification of truth table of Flip Flop using NAND gate (7400) & NOR gates (7402).

a) RS Flip-Flop

b) JK Flip-Flop

c) D Flip-Flop

d) T Flip-Flop

APPARATUS REQUIRED:

S. No.	COMPONENTS	SPECIFICATION	QTY
1	IC TRAINER KIT		1
2	NAND GATE	IC7400	1
3	NOR GATE	IC7402	1
4	NOT GATE	IC7404	1
5	OR GATE	IC7432	1
6	AND GATE	IC7408	1
7	POWER SUPPLY	0-30 V DC	1
8	BREAD BOARD		1
9	CONNECTING WIRES		

THEORY:

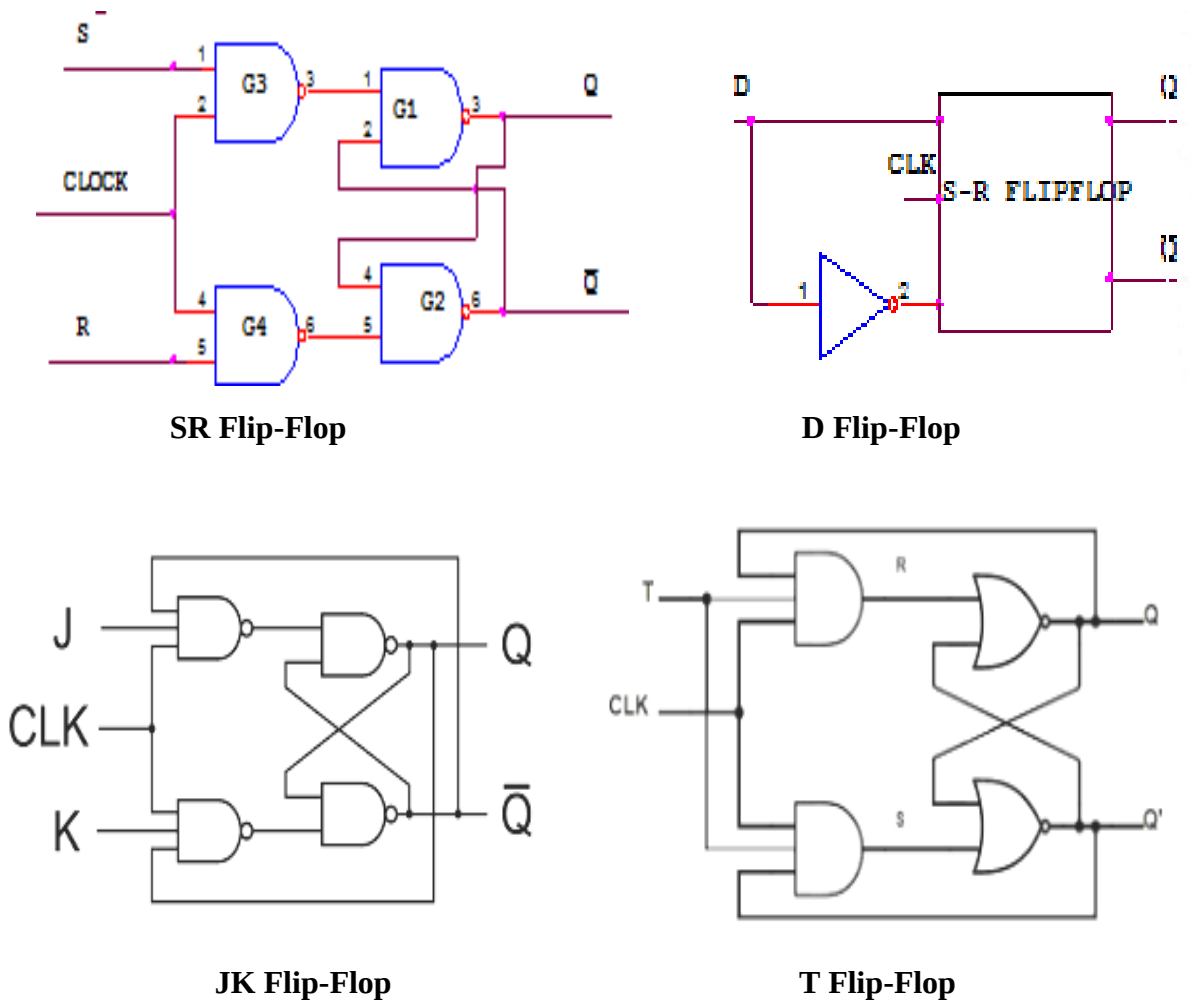
RS FLIP-FLOP: There are two inputs to the Flip-Flop defined as R and S. When I/Ps R=0 and S=0 then O/P remains unchanged. When I/Ps R=0 and S=1 the Flip- Flop is switches to the stable state where O/P is 1 i.e. SET. The I/P condition is R=1 and S=0 the Flip- Flop is switched to the stable state where O/P is 0 i.e. RESET. The I/P condition is R=1 and S=1 the Flip- Flop is switched to the stable state where O/P is forbidden.

JK FLIP-FLOP: For purpose of counting, the JK Flip- Flop is the ideal element to use. The variable J and K are called control I/Ps because they determine what the Flip- Flop does when a positive edge arrives. When J and K are both 0s, both AND gates are disabled and Q retains its last value.

D FLIP-FLOP: This kind of flip flop prevents the value of D from reaching the Q output until clock pulses occur. When the clock is low, both AND gates are disabled D can change value without affecting the value of Q. On the other hand, when the clock is high, In this case, Q is forced to equal the value of D. When the clock again goes low, Q retains or stores the last value of D. a D flip flop is a bi-stable circuit whose D input is transferred to the output after a clock pulse is received.

T FLIP-FLOP: The T or "toggle" Flip- Flop changes its output on each clock edge, giving an output which is half the frequency of the signal to the T input. It is useful for constructing binary counters, frequency dividers, and general binary addition devices. It can be made from a J-K Flip-Flop by tying both of its inputs high.

CIRCUIT DIAGRAM:



SR Flip-Flop **D Flip-Flop**

JK Flip-Flop **T Flip-Flop**

Figure 8.1 Logic diagram of various type of Flip-Flops

PROCEDURE:

1. Connect the circuit as shown in figure.
2. Apply Vcc& ground signal to every IC.
3. Observe the input & output according to the truth table.

TRUTH TABLE:

SR FLIP-FLOP:

CLOCK	S	R	Q_{n+1}
1	0	0	NO CHANGE
1	0	1	0
1	1	0	1
1	1	1	?

D FLIP-FLOP:

D	Q_{n+1}
0	0
1	1

JK FLIP-FLOP:

CLOCK	J	K	Q_{n+1}
1	0	0	NO CHANGE
1	0	1	0
1	1	0	1
1	1	1	Qn'

T FLIP-FLOP:

CLOCK	T	Q_{n+1}
1	0	NO CHANGE
1	1	Qn'

RESULT: Truth tables have been verified for all the Flip-Flops.

PRECAUTIONS:

1) Make the connections according to the IC pin diagram.
2) The connections should be tight.
3) The V_{cc} and ground should be applied carefully at the specified pin only.

EXPERIMENT NO: 9

OBJECTIVE: Implement D flip flop using SR flip flop and verify the truth table.

APPARATUS REQUIRED:

S. No.	COMPONENTS	SPECIFICATION	QTY
1	IC TRAINER KIT		1
2	NAND GATE	IC7400	1
4	NOT GATE	IC7404	1
5	POWER SUPPLY	0-30 V DC	1
6	BREAD BOARD		1
7	CONNECTING WIRES		

THEORY:

1. S-R Flip- Flop:

S-R Flip- Flop is similar to S-R latch expect clock signal and two AND gates. The circuit responds to the positive edge of clock pulse to the inputs S and R.

2. D Flip- Flop:

D Flip- Flop is a modified SR Flip- Flop which has an additional inverter. It prevents the inputs from becoming the same value.

Conversion of S-R Flip-Flop into D Flip-Flop:

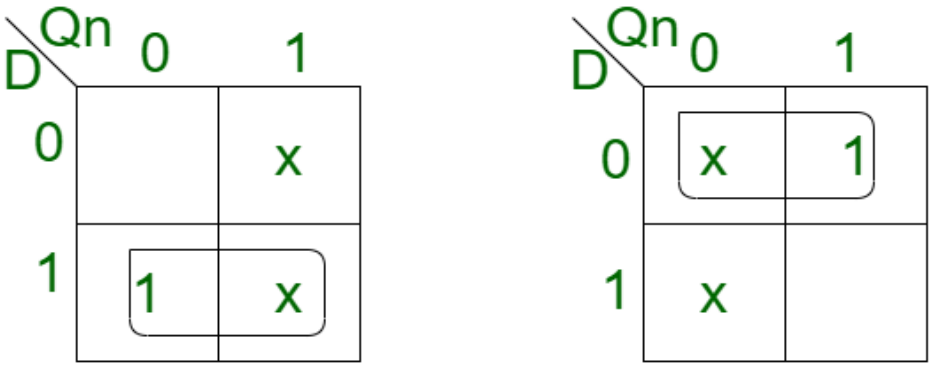
Step-1:

Construct the characteristic table of D Flip- Flop and excitation table of S-R Flip- Flop.

D	Qn	Qn+1	S	R
0	0	0	0	x
0	1	0	x	1
1	0	1	1	x
1	1	1	x	0

Step-2:

Using the K-map we find the Boolean expression of S and R in terms of D.



S=D

R=D'

Step-3:

We construct the circuit diagram of the conversion of S-R Flip- Flop into D Flip- Flop.

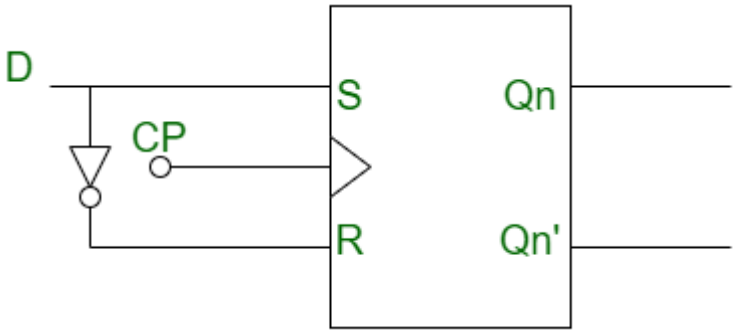


Figure 9.1 Conversion of S-R Flip- Flop into D Flip- Flop

RESULT: Implementation of D Flip-Flop using SR Flip-Flop has been verified.

PRECAUTIONS:

1) Make the connections according to the IC pin diagram.
2) The connections should be tight.
3) The V_{cc} and ground should be applied carefully at the specified pin only.

EXPERIMENT NO: 10

OBJECTIVE: Design Mod – N Synchronous Up Counter & Down Counter using 7476 JK Flip- Flop.

APPARATUS REQUIRED:

S. No.	COMPONENTS	SPECIFICATION	QTY
1	TRAINER KIT		1
2	DUAL, POSITIVE-EDGE TRIGGERED J-K FLIP-FLOP	IC7476	1
4	NOT GATE	IC7404	1
5	OR GATE	IC7432	1
6	AND GATE	IC7408	1
7	POWER SUPPLY	0-30 V DC	1
8	BREAD BOARD		1
9	CONNECTING WIRES		

THEORY:

Synchronous Counters can be made from Toggle or D-type Flip- Flops. Synchronous counters are easier to design than asynchronous counters. are all clocked together at the same time with the same clock signal. Due to this common clock pulse all output states switch or change simultaneously.

Mod – N Synchronous Up Counter & Down Counter using 7476 JK Flip- Flop

CLK	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0

J	K	Q_{n+1}	Comment
0	0	Q_n	NC
0	1	0	Reset
1	0	1	Set
1	1	Q_n'	Toggle

Note: Qn is Present State
Qn+1 is Next State

Figure 10.1 Count Table and Truth Table of J-K Flip Flop

Q_n	Q_{n+1}	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Present State			Next State			Excitation Table					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_1	K_1	J_2	K_2	J_3	K_3
0	0	0	0	1	0	0	x	1	x	0	x
0	1	0	0	1	1	1	x	x	0	0	x
0	1	1	1	1	0	x	1	x	0	1	x
1	1	0	1	0	1	1	x	x	1	x	0
1	0	1	0	0	1	x	0	0	x	x	1
0	0	1	0	0	0	x	1	0	x	0	x

Figure 10.2 State Table and Excitation Table of J-K Flip Flop

K – Maps:

J1

Q3 \ Q2 Q1	00	01	11	10
0	0	x	x	1
1	x	x	x	1

$J_1 = Q_2$

K1

Q3 \ Q2 Q1	00	01	11	10
0	x	1	1	x
1	x	0	x	x

$K_1 = \overline{Q_3}$

J2

Q3 \ Q2 Q1	00	01	11	10
0	1	0	x	x
1	x	0	x	x

$J_2 = \overline{Q_1}$

K2

Q3 \ Q2 Q1	00	01	11	10
0	X	X	0	0
1	X	X	X	1

$K_2 = Q_3$

J3

Q3 \ Q2 Q1	00	01	11	10
0	0	0	1	0
1	X	X	X	X

$J_3 = Q_2 * Q_1$

K3

Q3 \ Q2 Q1	00	01	11	10
0	X	X	X	X
1	X	1	X	0

$K_3 = Q_2$

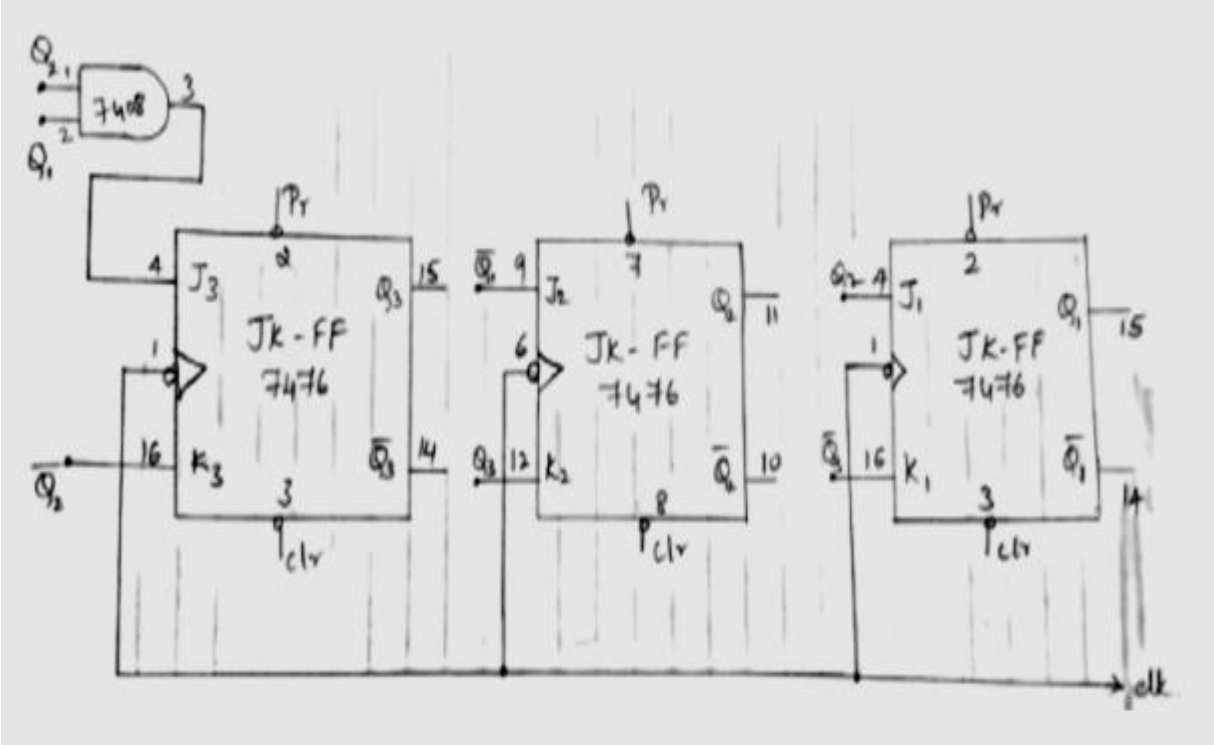


Figure 10.3 Circuit diagram of Mod 3 Synchronous Counter

RESULT: Mod -3 Synchronous Up Counter & Down Counter using 7476 JK Flip- Flop, has been realized successfully.

PRECAUTIONS:

1) All connections should be made neat and tight.
2) Power supply and ICs should be handled with carefully.
3) While making connections main supply should be kept switched off.
4) Never touch live and naked wires.

EXPERIMENT NO: 11

OBJECTIVE: Describing hardware in IoT: Hardware Architecture of Arduino UNO Board, Types of Arduino Board.

APPARATUS REQUIRED:

- Arduino UNO Board
- Different Arduino boards (Arduino Nano, Mega, Due, etc.)
- Datasheets/documentation
- Computer/Laptop

THEORY:

Arduino is an open-source electronics platform based on simple, easy-to-use hardware and software. The Arduino UNO board is the most widely used microcontroller board. It is based on the **ATmega328P microcontroller.**

Key Components of Arduino UNO:

Microcontroller: ATmega328P

Digital I/O Pins: 14 (6 PWM)

Analog Input Pins: 6

Flash Memory: 32 KB

Clock Speed: 16 MHz

USB Interface: For programming and power supply

Types of Arduino Boards:

Arduino UNO – Beginner-friendly, based on ATmega328P.

Arduino Nano – Compact, breadboard-friendly version.

Arduino Mega – More I/O pins, based on ATmega2560, for larger projects.

Arduino Due– ARM Cortex-M3 based, 32-bit microcontroller.

Arduino Lily pad – Wearable electronics applications.

OBSERVATION TABLE:

Arduino Board	Microcontroller	Digital I/O Pins	Analog Pins	Memory (Flash)	Clock Speed	Special Features
UNO	ATmega328P	14	6	32 KB	16 MHz	Most common, USB powered
Nano	ATmega328P	14	8	32 KB	16 MHz	Small size, breadboard use
Mega	ATmega2560	54	16	256 KB	16 MHz	Large projects, more I/O
Due	ARM Cortex-M3	54	12	512 KB	84 MHz	32-bit processing
Lily pad	ATmega328P	14	6	16 KB	8 MHz	Wearable, textile projects

RESULT: The hardware architecture of Arduino UNO and various types of Arduino boards were studied and understood.

PRECAUTIONS:

1) Handle Arduino boards carefully to avoid static discharge damage.
2) Ensure correct power supply to the board.
3) Avoid short circuits while connecting external devices.

EXPERIMENT NO: 12

OBJECTIVE: Fundamentals of Arduino Programming: Installation of Arduino IDE, Working with structures, Variables, Flow control, Digital I/O f. Analog I/O, Time, Math, Random, Serial.

APPARATUS REQUIRED:

- Computer/Laptop
- Arduino UNO board
- USB cable
- LEDs, resistors, push buttons, sensors

THEORY:

The Arduino IDE is used to write, compile, and upload programs (called **sketches**) to Arduino boards.

Variables & Structures: Used to store and organize data.

Flow Control: if, else, loops (for, while) to control execution.

Digital I/O: Used to read/write HIGH (1) / LOW (0) signals.

Analog I/O: Reads analog input (0–1023) or outputs PWM signals.

Time Functions: `delay( )`, `millis( )`.

Math/Random: Built-in functions for calculations and random numbers.

Serial Communication: `Serial.begin( )`, `Serial.print( )` for PC-Arduino communication.

Program Example (Blink LED)

```
// Blink LED Program
int led Pin = 13;
void setup () {
  pinMode (ledPin, OUTPUT);      // set pin as output
}
void loop() {
  digitalWrite (ledPin, HIGH);      // turn LED on
  delay(1000);                      // wait 1 second
  digitalWrite (ledPin, LOW);       // turn LED off
  delay(1000);                      // wait 1 second
}
```

OBSERVERTION TABLE:

Code Component	Functionality
pin Mode ( )	Serial.print ( )
digital Write ( )	Sets HIGH / LOW signal
digital Read ( )	Reads digital input
analog Read ( )	Reads analog values
analog Write ( )	Outputs PWM
delay (ms)	Pauses program execution
Serial.print ( )	Sends data to Serial Monitor

RESULT: Basic Arduino programs were successfully written and executed using Arduino IDE.

PRECAUTIONS:

1) Always select the correct board and COM port in Arduino IDE.
2) Verify the program before uploading.
3) Avoid wrong connections of input/output devices.

EXPERIMENT NO: 13

OBJECTIVE: Interfacing Arduino with I/O Devices: Push button, LED, Ultrasonic Sensor.

APPARATUS REQUIRED:

- Arduino UNO board
- Push button
- LED and resistor (220Ω)
- Ultrasonic sensor (HC-SR04)
- Jumper wires
- Breadboard

THEORY:

Push Button: Acts as a digital input (pressed = HIGH, released = LOW).

LED: Output device to show ON/OFF state.

Ultrasonic Sensor (HC-SR04): Measures distance using sound waves. It sends an ultrasonic pulse and calculates distance based on the time taken for the echo.

Distance formula

$$\text{Distance (cm)} = (\text{Time} * 0.034) / 2$$

Program Example (Ultrasonic Sensor + LED Indicator):

```
// Ultrasonic Sensor with LED
#define trigPin 9
#define echoPin 10
#define ledPin 13
void setup ( ) {
  pinMode (trigPin, OUTPUT);
  pinMode (echoPin, INPUT);
  pinMode (ledPin, OUTPUT);
 Serial.begin(9600);
}
```

```
void  loop( ) {
digitalWrite (trig Pin, LOW);
delayMicroseconds (2);
digitalWrite (trigPin, HIGH);
delayMicroseconds (10);
digitalWrite (trigPin, LOW);

long duration = pulseIn(echoPin, HIGH);
int distance = duration * 0.034 / 2;

Serial.print ("Distance: ");
Serial.println(distance);

 if (distance < 10) {
    digitalWrite (ledPin, HIGH);          // LED ON if object is close
  } else {
 digitalWrite (ledPin, LOW);
  }
  delay (500);
}
```

OBSERVATION TABLE:

Input Device	Output Device	Condition	Observation
Push Button	LED	Pressed	LED ON
Push Button	LED	Released	LED OFF
Ultrasonic	LED	< 10 cm	LED ON
Ultrasonic	LED	≥ 10 cm	LED OFF

RESULT:

Arduino was successfully interfaced with push button, LED, and ultrasonic sensor. The LED responded correctly to input signals and sensor readings.

PRECAUTIONS:

1) Check connections before powering the circuit.
2) Use current-limiting resistor with LED.
3) Place ultrasonic sensor in open space for accurate measurements.
4) Avoid touching live wires while operating.