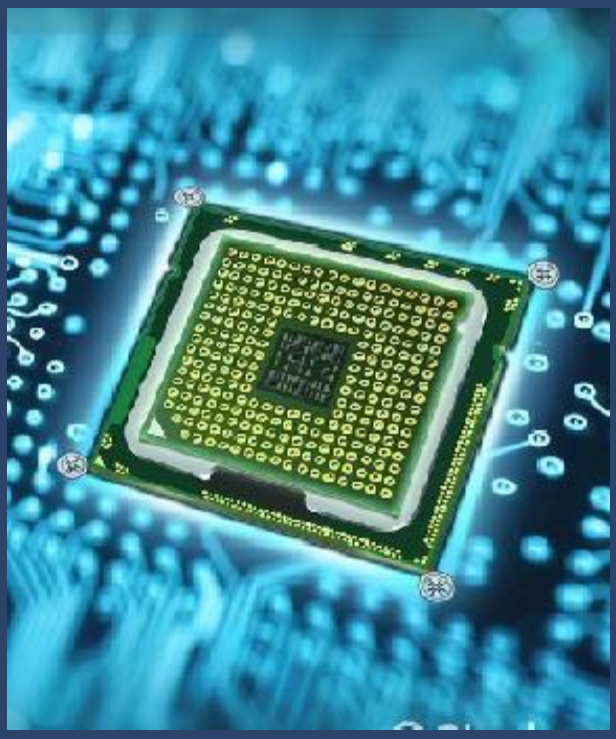
DIGITAL PRINCIPLES AND COMPUTER ORGANIZATION LAB MANUAL



REGULATION 2021

ISBN 978-93-6128-892-0
9 789361 288920

G.SANGEETHA M.E

DIGITAL PRINCIPLES AND COMPUTER ORGANIZATION LAB MANUAL

(COMPUTER SCIENCE AND ENGINEERING)
&
(ARTIFICIAL INTELLIGENCE AND DATA SCIENCE)

REGULATION – 2021

AUTHOR:

Mrs.G.SANGEETHA.,M.E.,

GENERAL GUIDELINES AND SAFETY INSTRUCTIONS

1. Sign in the log register as soon as you enter the lab and strictly observe your lab timings.
2. Strictly follow the written and verbal instructions given by the teacher / Lab Instructor. If you do not understand the instructions, the handouts and the procedures, ask the instructor or teacher.
3. **Never work alone!** You should be accompanied by your laboratory partner and/or the Instructors/teaching assistants all the time.
4. It is mandatory to come to lab in a formal dress and wear your ID cards.
5. Do not wear loose-fitting clothing or jewels in the lab. Rings and necklaces are usual excellent conductors of electricity.
6. Mobile phones should be switched off in the lab. Keep bags in the bag rack.
7. Keep the labs clean at all times, no food and drinks allowed inside the lab.
8. Intentional misconduct will lead to expulsion from the lab.
9. Do not handle any equipment without reading the safety instructions. Read the handout and procedures in the Lab Manual before starting the experiments.
10. Do your wiring, setup, and a careful circuit check out before applying power. Do not make circuit changes or perform any wiring when power is on.
11. Avoid contact with energized electrical circuits.
12. Do not insert connectors forcefully into the sockets.
13. **NEVER** try to experiment with the power from the wall plug.
14. Immediately report dangerous or exceptional conditions to the Lab instructor/ teacher: Equipment that is not working as expected, wires or connectors are broken, the equipment that smells or “smokes”. If you are not sure what the problem is or what's going on, switch off the Emergency shutdown.
15. Never use damaged instruments, wires or connectors. Hand over these parts to the Lab instructor/Teacher.
16. Be sure of location of fire extinguishers and first aid kits in the laboratory.
17. After completion of Experiment, return the bread board, trainer kits, wires, CRO probes and other components to lab staff. Do not take any item from the lab without permission.
18. Observation book and lab record should be carried to each lab. Readings of current lab experiment are to be entered in Observation book and previous lab experiment should be written in Lab record book. Both the books should be corrected by the faculty in each lab.
19. Special Precautions during Soldering practice
    a. Hold the soldering iron away from your body. Don't point the iron towards you.
    b. Don't use a spread solder on the board as it may cause short circuit.
    c. Do not over heat the components as excess heat may damage the components/board.
    d. In case of burn or injury seek first aid available in the labor at the college dispensary.

COMPUTER SCIENCE AND ENGINEERING
&
DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

REGULATION–2021

CS3351-DIGITAL PRINCIPLES AND COMPUTER ORGANIZATION LABORATORY

Mrs.G.SANGEETHA,M.E.,

Assistant Professor/Electronics and Communication Engineering
Annai Mira College of Engineering and Technology
Ranipet-632 517

PREFACE

This book on "DIGITAL PRINCIPLES AND COMPUTER ORGANIZATION LABORATORY **(Computer Science And Engineering& Artificial Intelligence and Data Science)**" covers the complete syllabus prescribed by the Anna University, Chennai for the fourth semester **B.E/ B.Tech**. Degree course under **Outcome Based Education Credit System with the new regulation 2021**.

This book covers logic gates, Boolean functions, adders, subtractor, encoder, decoder, multiplexer, demultiplexer, synchronous counter and shift register for designing digital circuits.

We hope that this book will be useful to both teachers and students. Finally we would request the readers to kindly send their valuable comments and suggestions towards the improvement of the manual and the same will be gratefully acknowledge.

Any suggestion from the reader for the betterment of this book can be dropped into sangeedec@gmail.com.

Mrs.G.SANGEETHA.,M.E.,

ACKNOWLEDGEMENT

We are thankful to and fortunate enough to get constant encouragement, support and guideline from Chairman **Thiru.S.Ramadoss Ayya**, Secretary & Treasurer **Mr.G.Thamotharan** for his blessings to complete the book successfully.

We would not forget to remember our Principal **Dr.T.K.Gopinathan** for his constant assistance in preparing this book.

ANNAI MIRA

COLLEGE OF ENGINEERING AND TECHNOLOGY

ARAPAKKAM, RANIPET DT – 632517.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RECORD NOTEBOOK

SUBJECT CODE/NAME: CS3351, DIGITAL PRINCIPLES AND COMPUTER ORGANIZATION LABORATORY

YEAR/SEM: II/III **BATCH**: 2022 - 2026

AS PER ANNA UNIVERSITY, CHENNAI REGULATION 2021

Name : .....

Register Number : .....

Year & Branch : .....

Semester : .....

Academic Year : .....

ANNAI MIRA COLLEGE OF ENGINEERING AND TECHNOLOGY

NH-46, Chennai-Bengaluru National Highways, Arappakkam,
Ranipet-632517, Tamil Nadu, India
Telephone: 04172-292925 Fax: 04172-292926
Email:amcet.rtet@gmail.com/info@amcet.in Web: www.amcet.in

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

CERTIFICATE

This is to certify that the bonafide record of the practical work done by ..... Register Number ..... of II year B.Tech(Artificial Intelligence and Data Science) submitted for the B.Tech Degree Practical examination (III Semester) in CS3351 – DIGITAL PRINCIPLES AND COMPUTER ORGANIZATION LABORATORY during the academic year 2023 -2024.

Submitted for the practical examination held on ----------

Staff in –Charge **Head of the Department**

Internal Examiner **External Examiner**

LIST OF EXPERIMENTS

1. Verification of Boolean theorems using logic gates.
2. Design and implement of combinational circuits using gates for arbitrary functions.
3. Implement of 4-bit binary adder/subtractor circuits.
4. Implement of code converters.
5. Implement of BCD adder, encoder and decoder circuits.
6. Implement of functions using Multiplexers.
7. Implement of the synchronous counters.
8. Implement of a Universal Shift register.
9. Simulator based study of Computer Architecture.

TABLE OF CONTENT

Ex.No.-1a STUDY OF LOGIC GATES

AIM:

APPARATUS REQUIRED:

SL.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.	NAND GATE 3 I/P	IC 7410	1
8.	IC TRAINER KIT	-	1
9.	PATCH CORD	-	14

THEORY:

Circuit that takes the logical decision and the process are called logic gates. Each gate has one or more input and only one output.

OR, AND and NOT are basic gates. NAND, NOR and X-OR are known as universal gates. Basic gates form these gates.

AND GATE:

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

OR GATE:

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.

NOT GATE:

The NOT gate is called an inverter. The output is high when the input is low. The output is low when the input is high.

AND GATE:

The NAND gate is a contraction of AND-NOT. The output is high when both inputs are low and any one of the input is low .The output is low level when both inputs are high.

NOR GATE:

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.

X- OR GATE:

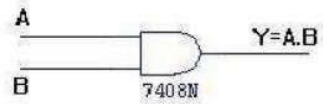
The output is high when any one of the inputs is high. The output is low when both the inputs are low and both the inputs are high.

PROCEDURE:

(i) Connections are given as per circuit diagram.
(ii) Logical inputs are given as per circuit diagram.
(iii) Observe the output and verify the truth table.

AND GATE

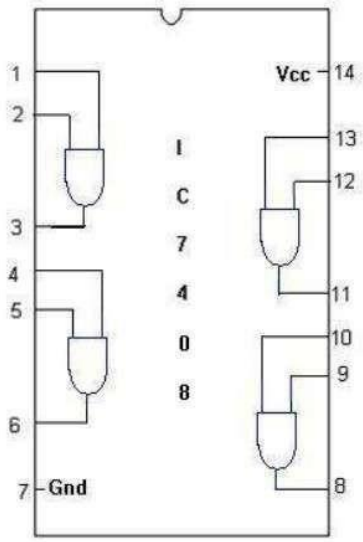
SYMBOL



TRUTH TABLE

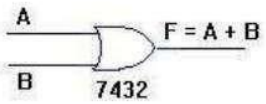
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

PIN DIAGRAM

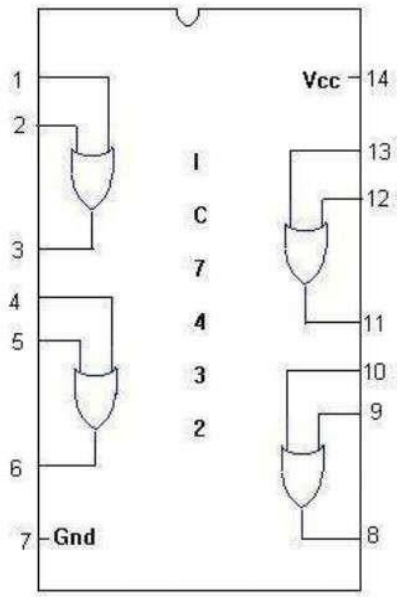


OR GATE

SYMBOL :



PIN DIAGRAM :

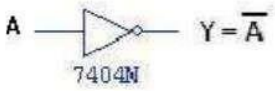


TRUTH TABLE

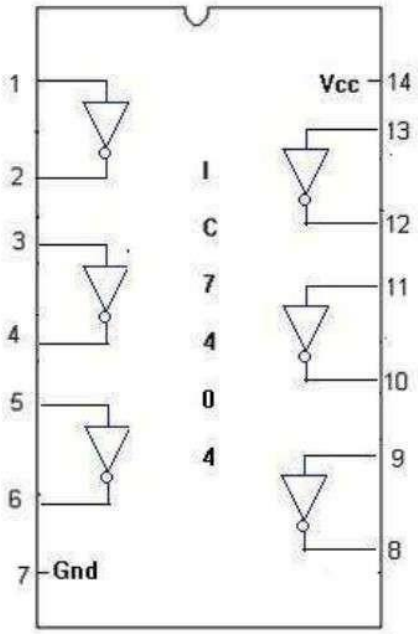
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT GATE

SYMBOL



PIN DIAGRAM

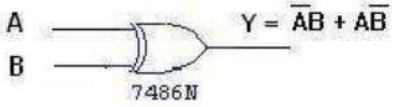


TRUTH TABLE :

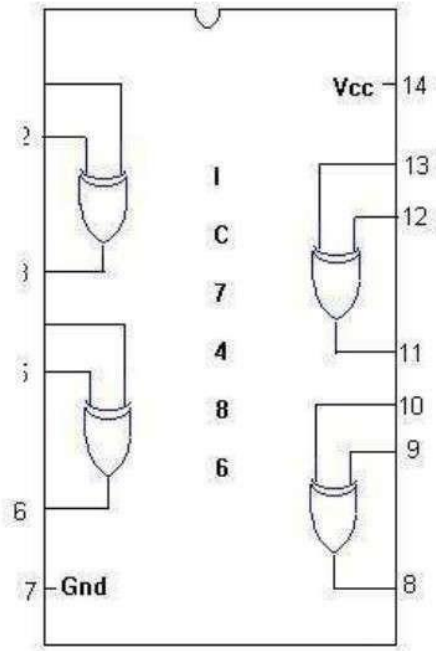
A	$\overline{A}$
0	1
1	0

EX-OR GATE

SYMBOL



PIN DIAGRAM

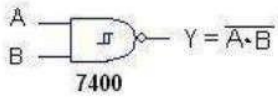


TRUTH TABLE :

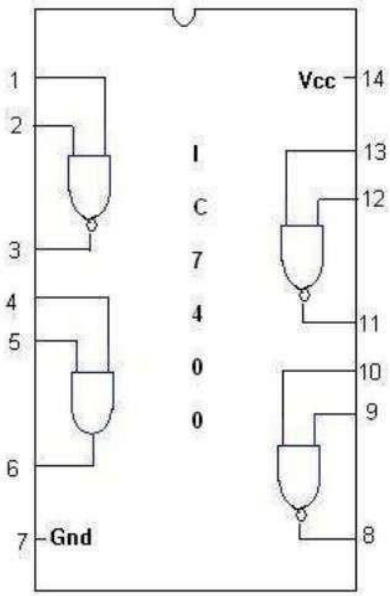
A	B	$\overline{A}B + A\overline{B}$
0	0	0
0	1	1
1	0	1
1	1	0

2-INPUT NAND GATE

SYMBOL



PIN DIAGRAM

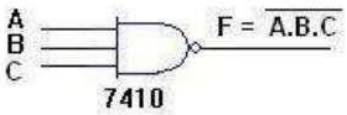


TRUTH TABLE

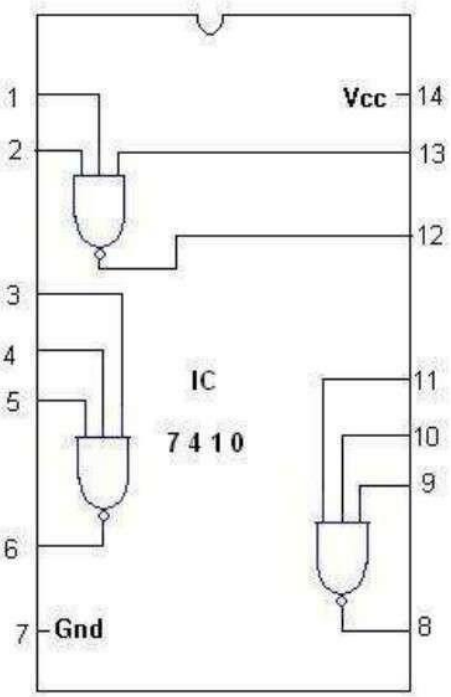
A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

3-INPUT NAND GATE

SYMBOL :



PIN DIAGRAM :

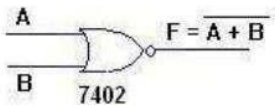


TRUTH TABLE

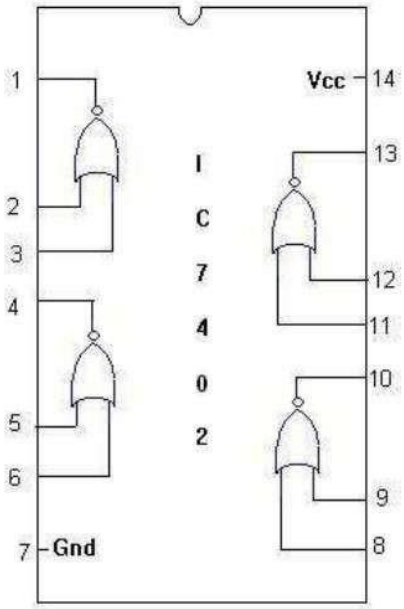
A	B	C	$\overline{A.B.C}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

NOR GATE

SYMBOL :



PIN DIAGRAM :



TRUTH TABLE

A	B	$\overline{A+B}$
0	0	1
0	1	1
1	0	1
1	1	0

RESULT:

Ex.No.-1 VERIFICATION OF BOOLEAN THEOREMS USING DIGITAL LOGIC GATES

AIM:

APPARATUS REQUIRED:

SL. 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.	CONNECTING WIRES	-	As per required

THEORY:

BASIC BOOLEAN LAWS

1. Commutative Law

The binary operator OR, AND is said to be commutative if,

1. A+B = B+A
2. A.B=B.A

2. Associative Law

The binary operator OR, AND is said to be associative if,

1. A+(B+C) = (A+B)+C
2. A.(B.C) = (A.B).C

3. Distributive Law

The binary operator OR, AND is said to be distributive if,

1. A+(B.C) = (A+B).(A+C)
2. A.(B+C) = (A.B)+(A.C)

4. Absorption Law

1. A+AB = A
2. A+AB = A+B

5. Involution (or) Double complement Law

1. A = A

6. Idempotent Law

1. A+A = A
2. A.A = A

7. Complementary Law

1. A+A' = 1
2. A.A' = 0

8. De Morgan's Theorem

1. The complement of the sum is equal to the sum of the product of the individual complements.

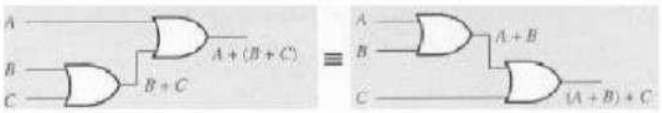
A+B = A.B

2. The complement of the product is equal to the sum of the individual complements.

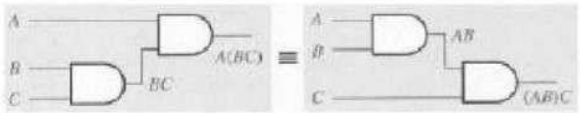
A.B = A+B

Associative Laws of Boolean Algebra

A + (B + C) = (A + B) + C



A • (B • C) = (A • B) • C



Proof of the Associative Property for the OR operation: (A+B)+C = A+(B+C)

A	B	C	(A+B)	(B+C)	A+(B+C)	(A+B)+C
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	1	0	1	1
1	0	1	1	1	1	1
1	1	0	1	1	1	1
1	1	1	1	1	1	1

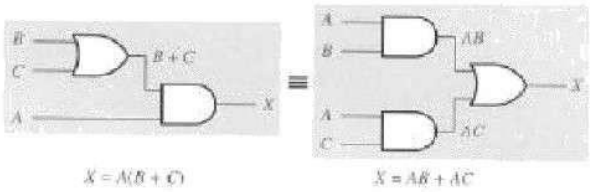
Proof of the Associative Property for the AND operation: (A·B)·C = A·(B·C)

A	B	C	(A·B)	(B·C)	A· (B·C)	(A·B)·C
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	1	0	0	0
1	1	1	1	1	1	1

Distributive Laws of Boolean Algebra

$$A \bullet (B + C) = A \bullet B + A \bullet C$$

$$A\,(B + C) = A\,B + A\,C$$



Proof of Distributive Rule

A	B	C	A·B	A·C	(A·B)+ (A·C)	(B+C)	A·(B+C)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	0	1	0
0	1	1	0	0	0	1	0
1	0	0	0	0	0	0	0
1	0	1	0	1	1	1	1
1	1	0	1	0	1	1	1
1	1	1	1	1	1	1	1

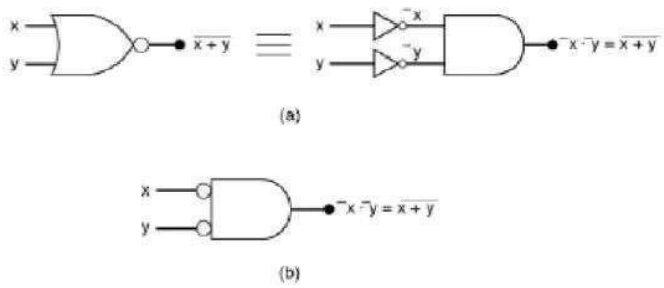
Proof of Distributive Rule

A	B	C	A+B	A+C	(A+B)· (A+C)	(B·C)	A+(B·C)
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1

Demorgan's Theorem

a) Proof of equation (1):

Construct the two circuits corresponding to the functions A'. B'and (A+B)' respectively. Show that for all combinations of A and B, the two circuits give identical results. Connect these circuits and verify their operations.

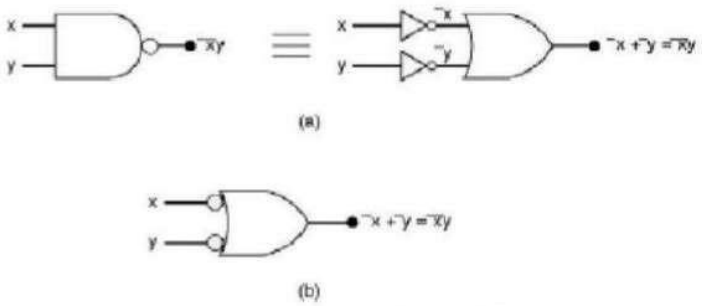


Proof (via Truth Table) of DeMorgan's Theorem $\overline{A \cdot B} = \overline{A} + \overline{B}$

A	B	A·B	$\overline{A \cdot B}$	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

b) Proof of equation (2)

Construct two circuits corresponding to the functions A'+B'and (A.B)' A.B, respectively. Show that, for all combinations of A and B, the two circuits give identical results. Connect these circuits and verify their operations.

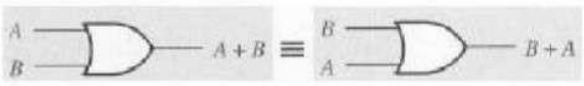


Proof (via Truth Table) of DeMorgan's Theorem $\overline{A + B} = \overline{A} \cdot \overline{B}$

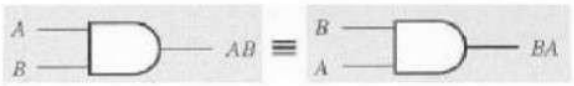
A	B	A+B	$\overline{A + B}$	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Commutative Laws of Boolean Algebra

A + B = B + A



A • B = B • A



We will also use the following set of postulates:

P1: Boolean algebra is closed under the AND, OR, and NOT operations.

P2: The identity element with respect to • is one and + is zero. There is no identity element with respect to logical NOT.

P3: The • and + operators are commutative.

P4: • and + are distributive with respect to one another. That is,
A • (B + C) = (A • B) + (A • C) and A + (B • C) = (A + B) • (A + C).

P5: For every value A there exists a value A' such that A•A' = 0 and A+A' = 1. This value is the logical complement (or NOT) of A.

P6: • and + are both associative. That is, (A•B)•C = A•(B•C) and (A+B)+C = A+(B+C).

You can prove all other theorems in boolean algebra using these postulates.

PROCEDURE:

1. Obtain the required IC along with the Digital trainer kit.
2. Connect zero volts to GND pin and +5 volts to Vcc .
3. Apply the inputs to the respective input pins.
4. Verify the output with the truth table.

RESULT:

Ex.No.-2 DESIGN AND IMPLEMENTATION OF COMBINATIONAL CIRCUITS USING GATES FOR ARBITRARY FUNCTIONS

AIM:

APPARATUS REQUIRED:

SL.NO.	COMPONENT	SPECIFICATION	QTY.
1.	NAND GATE	IC 7400	2
2.	AND GATE	IC 7408	1
3.	NOT GATE	IC 7404	1
4.	OR GATE	IC 7432	1
5.	NOR GATE	IC 7402	2
6.	BREAD BOARD	-	1

THEORY:

Canonical Forms (Normal Forms): Any Boolean function can be written in disjunctivenormal form (sum of min-terms) or conjunctive normal form (product of maxterms).

A Boolean function can be represented by a Karnaugh map in which each cell corresponds to a minterm. The cells are arranged in such a way that any two immediately adjacent cells correspond to two minterms of distance 1. There is more than one way to construct a map with this property.

Karnaugh Maps

For a function of two variables, say, f(x, y),

	x'	x
y'	f(0,0)	f(1,0)
y	f(0,1)	f(1,1)

For a function of three variables, say, f(x, y, z)

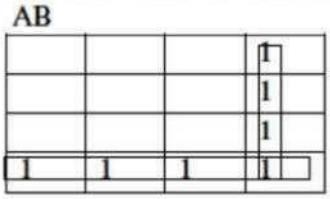
	x'y'	x'y	xy	xy'
z'	f(0,0,0)	f(0,1,0)	f(1,1,0)	f(1,0,0)
z	f(0,0,1)	f(0,1,1)	f(1,1,1)	f(1,0,1)

For a function of four variables: f(w, x, y, z)

	w'x'	w'x	wx	wx'
y'z'	0	4	12	8
y'z	1	5	13	9
yz	3	7	15	11
yz'	2	6	14	10

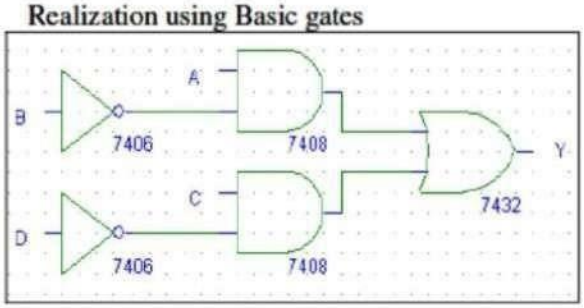
Realization of Boolean expression:

1) $Y = \bar{A}\bar{B}C\bar{D} + \bar{A}BC\bar{D} + ABC\bar{D} + A\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}CD$

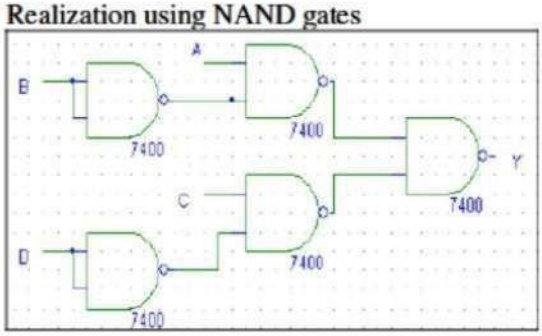


After simplifying using K-Map method we get $Y = A\bar{B} + C\bar{D}$

Realization using Basic gates



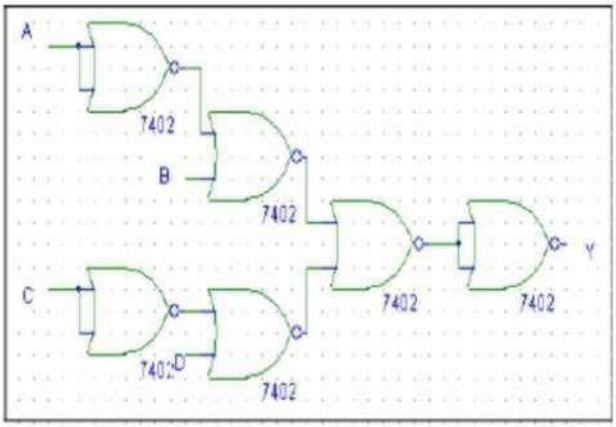
Realization using NAND gates



TRUTH TABLE

INPUTS				OUTPUT
A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Realization using NOR gates



2) For the given Truth Table, realize a logical circuit using basic gates and NAND gates

Inputs				Output
A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

PROCEDURE:

(i) Connections are given as per circuit diagram.

(ii) Logical inputs are given as per circuit diagram.

(iii) Observe the output and verify the truth table.

RESULT:

Ex.No.-3 4-BIT ADDER AND SUBTRACTOR

AIM:

APPARATUS REQUIRED:

SL.NO.	COMPONENT	SPECIFICATION	QTY.
1.	IC	IC 7483	1
2.	EX-OR GATE	IC 7486	1
3.	NOT GATE	IC 7404	1
3.	IC TRAINER KIT	-	1
4.	PATCH CORDS	-	40

THEORY:

4 BIT BINARY ADDER:

A binary adder is a digital circuit that produces the arithmetic sum of two binary numbers. It can be constructed with full adders connected in cascade, with the output carry from each full adder connected to the input carry of next full adder in chain. The augends bits of 'A' and the addend bits of 'B' are designated by subscript numbers from right to left, with subscript 0 denoting the least significant bits. The carries are connected in chain through the full adder. The input carry to the adder is C_0 and it ripples through the full adder to the output carry C_4.

4 BIT BINARY SUBTRACTOR:

The circuit for subtracting A-B consists of an adder with inverters, placed between each data input 'B' and the corresponding input of full adder. The input carry C_0 must be equal to 1 when performing subtraction.

4 BIT BINARY ADDER/SUBTRACTOR:

The addition and subtraction operation can be combined into one circuit with one common binary adder. The mode input M controls the operation. When M=0, the circuit is adder circuit. When M=1, it becomes subtractor.

4 BIT BCD ADDER:

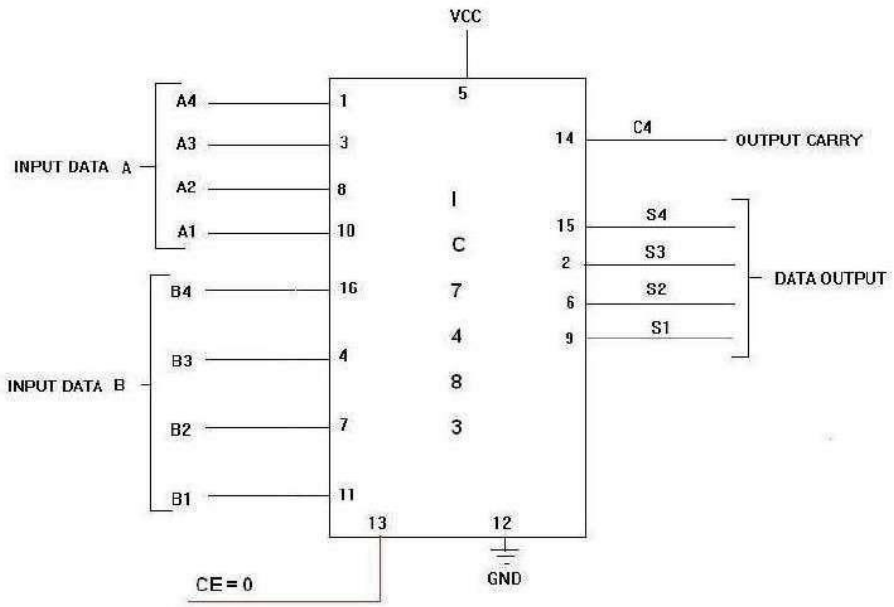
Consider the arithmetic addition of two decimal digits in BCD, together with an input carry from a previous stage. Since each input digit does not exceed 9, the output sum cannot be greater than 19, the 1 in the sum being an input carry. The output of two decimaldigits must be represented in BCD and should appear in the form listed in the columns.

ABCD adder that adds 2 BCD digits and produce a sum digit in BCD. The 2 decimal digits, together with the input carry, are first added in the top 4 bit adder to produce the binary sum.

PIN DIAGRAM FOR IC 7483:

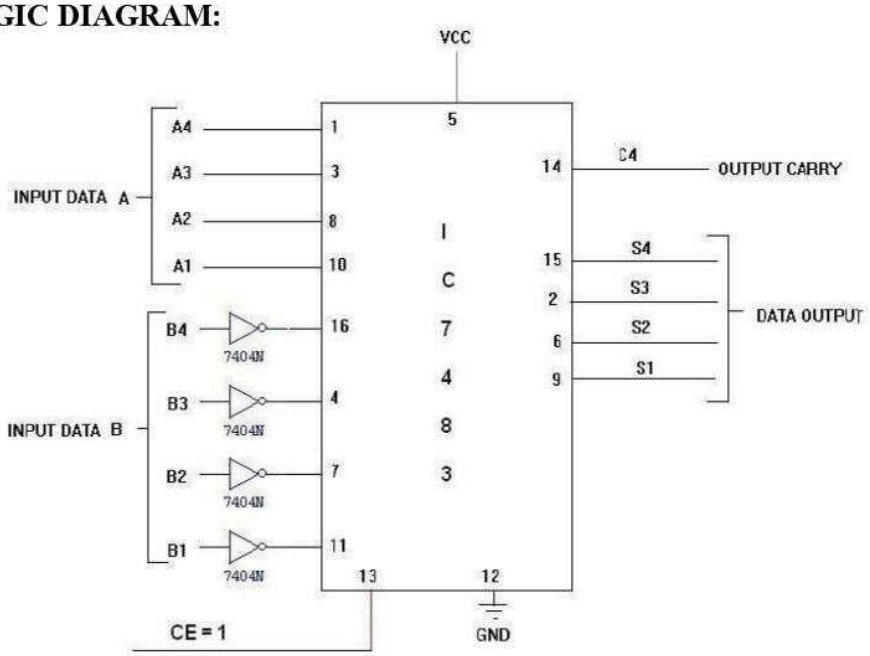
4-BIT BINARY ADDER

LOGIC DIAGRAM:



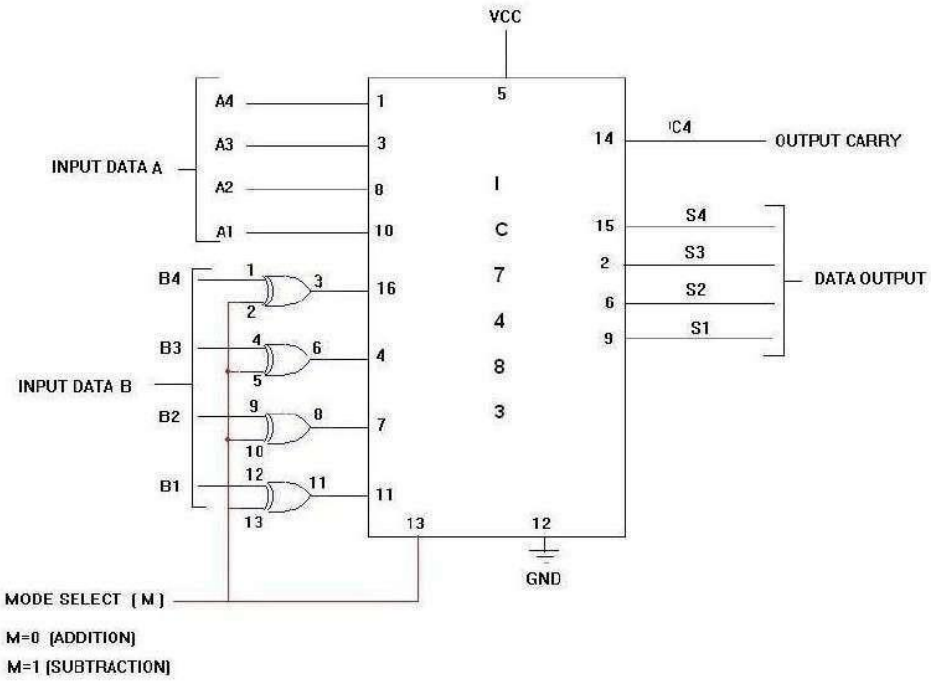
4-BIT BINARY SUBTRACTOR

LOGIC DIAGRAM:



4-BIT BINARY ADDER/SUBTRACTOR

LOGIC DIAGRAM:



TRUTH TABLE:

Input Data A				Input Data B					Addition					Subtraction			
A4	A3	A2	A1	B4	B3	B2	B1	C	S4	S3	S2	S1	B	D4	D3	D2	D1
1	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	1	0
1	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	1	0	1	0	0	1	0	1	0
0	0	0	1	0	1	1	1	0	1	0	0	0	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	0	1	0	0	1	1	1	1
1	1	1	0	1	1	1	1	1	1	0	1	0	0	1	1	1	1
1	0	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	1

PROCEDURE:

(i) Connections were given as per circuit diagram.
(ii) Logical inputs were given as per truth table
(iii) Observe the logical output and verify with the truth tables.

RESULT:

Ex.No.-4 CODE CONVERTOR

AIM:

APPARATUS REQUIRED:

SL.NO.	COMPONENT	SPECIFICATION	QTY.
1.	X-OR GATE	IC 7486	1
2.	AND GATE	IC 7408	1
3.	OR GATE	IC 7432	1
4.	NOT GATE	IC 7404	1
5.	IC TRAINER KIT	-	1
6.	PATCH CORDS	-	35

THEORY:

The availability of large variety of codes for the same discrete elements of information results in the use of different codes by different systems. A conversion circuit must be inserted between the two systems if each uses different codes for same information. Thus, code converter is a circuit that makes the two systems compatible even though each uses different binary code.

The bit combination assigned to binary code to gray code. Since each code uses four bits to represent a decimal digit. There are four inputs and four outputs. Gray code is a non-weighted code.

The input variable are designated as B3, B2, B1, B0 and the output variables are designated as C3, C2, C1, Co. from the truth table, combinational circuit is designed. The Boolean functions are obtained from K-Map for each output variable.

A code converter is a circuit that makes the two systems compatible even though each uses a different binary code. To convert from binary code to Excess-3 code, the input lines must supply the bit combination of elements as specified by code and the output lines generate the corresponding bit combination of code. Each one of the four maps represents one of the four outputs of the circuit as a function of the four input variables.

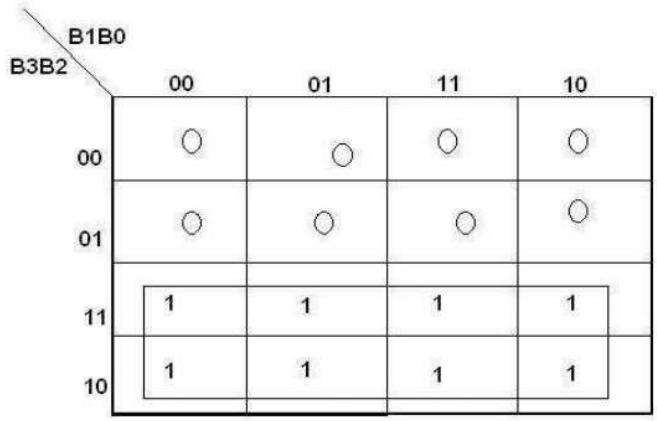
A two-level logic diagram may be obtained directly from the Boolean expressions derived by the maps. These are various other possibilities for a logic diagram that implements this circuit. Now the OR gate whose output is C+D has been used to implement partially each of three outputs.

BINARY TO GRAY CODE CONVERTOR

TRUTH TABLE:

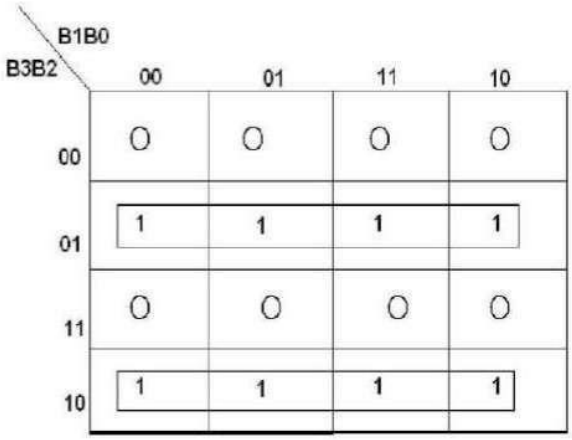
Binary Input				Gray Code Output			
B3	**B2**	**B1**	**B0**	**G3**	**G2**	**G1**	**G0**
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

K-Map for G_3



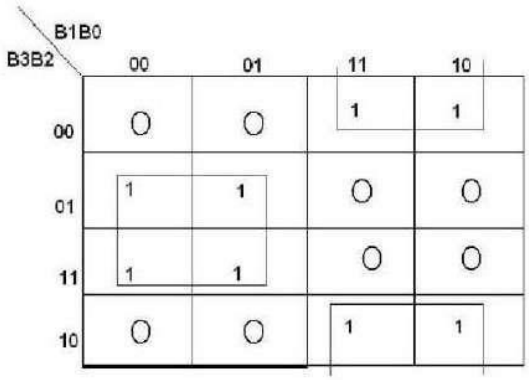
$$G_3 = B_3$$

K-Map for G_2



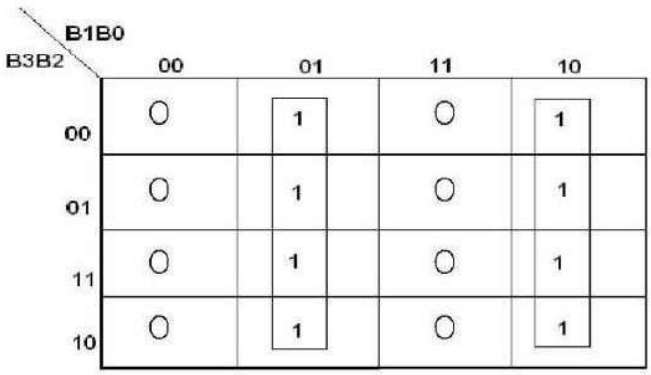
$G2 = B3 \oplus B2$

K-Map for G_1



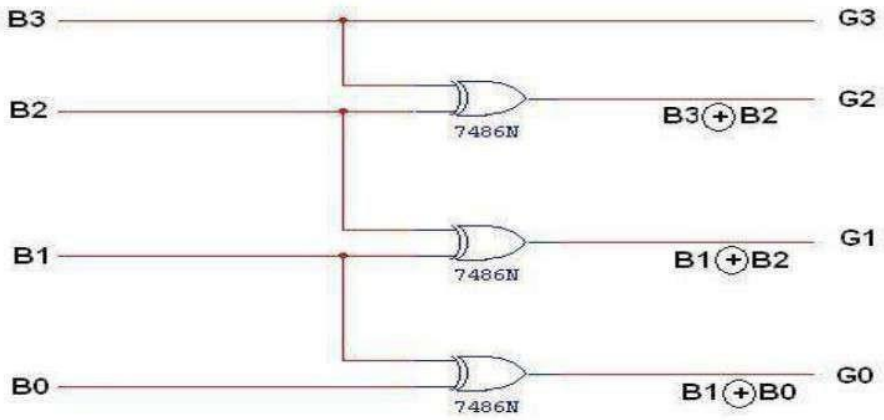
$G1 = B1 \oplus B2$

K-Map for G_0



$G0 = B1 \oplus B0$

LOGIC DIAGRAM:

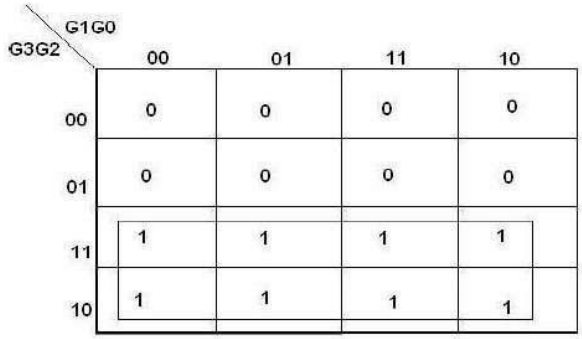


GRAY CODE TO BINARY CONVERTOR

TRUTH TABLE:

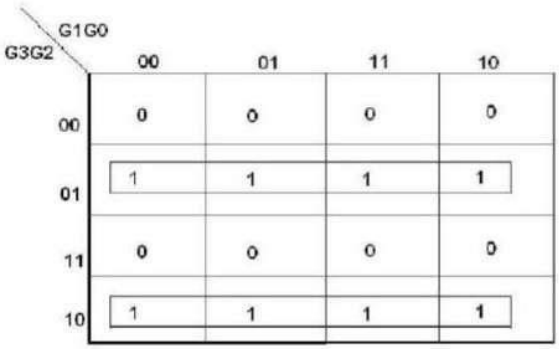
GRAY CODE				BINARY CODE			
G3	**G2**	**G1**	**G0**	**B3**	**B2**	**B1**	**B0**
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

K-Map for B_3:



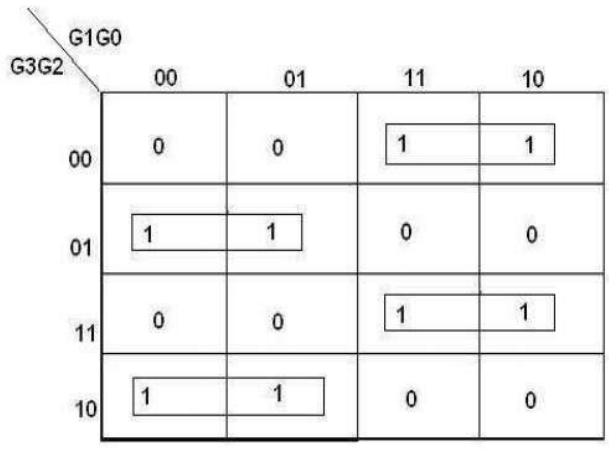
B3 = G3

K-Map for B_2:



B2 = G3⊕G2

K-Map for B1:



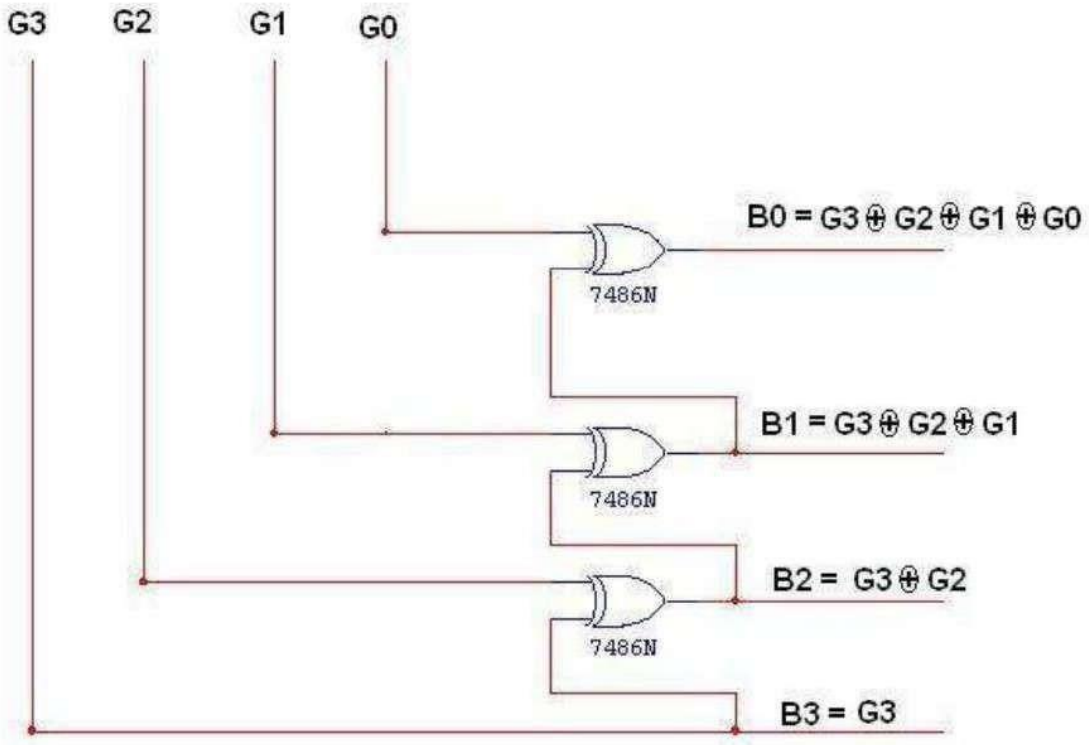
B1 = G3⊕G2⊕G1

K-Map for B0:

G3G2 \ G1G0	00	01	11	10
00	0	①	0	①
01	①	0	①	0
11	0	①	0	①
10	①	0	①	0

$$B0 = G3 \oplus G2 \oplus G1 \oplus G0$$

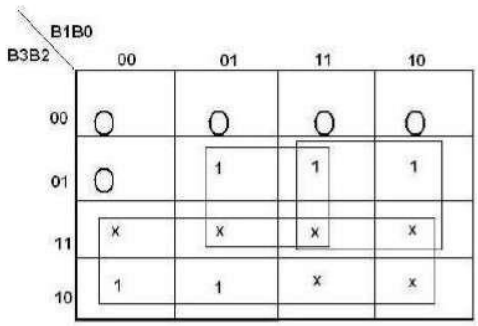
LOGIC DIAGRAM:



TRUTH TABLE: **BCD TO EXCESS-3 CONVERTOR**

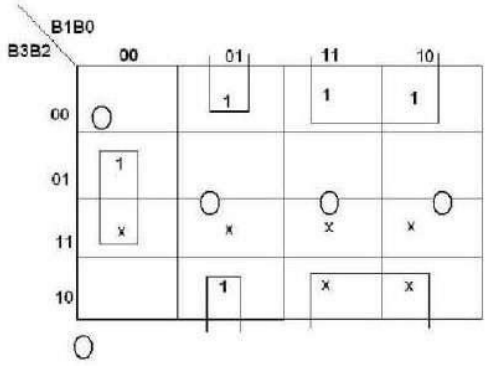
BCD input				**Excess – 3 output**			
B3	**B2**	**B1**	**B0**	**G3**	**G2**	**G1**	**G0**
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	X

K-Map for E_3:



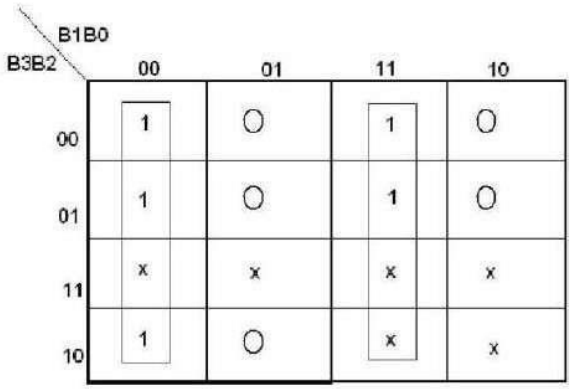
E3 = B3 + B2 (B0 + B1)

K-Map for E_2:



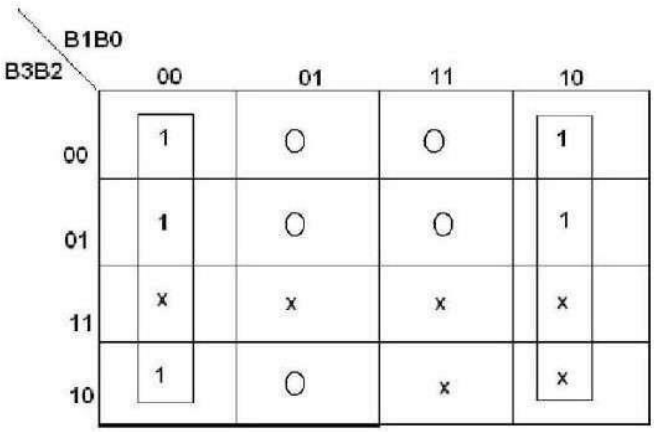
$$E2 = B2 \oplus (B1 + B0)$$

K-Map for E_1:



$$E1 = B1 \bar{\oplus} B0$$

K-Map for E_0:



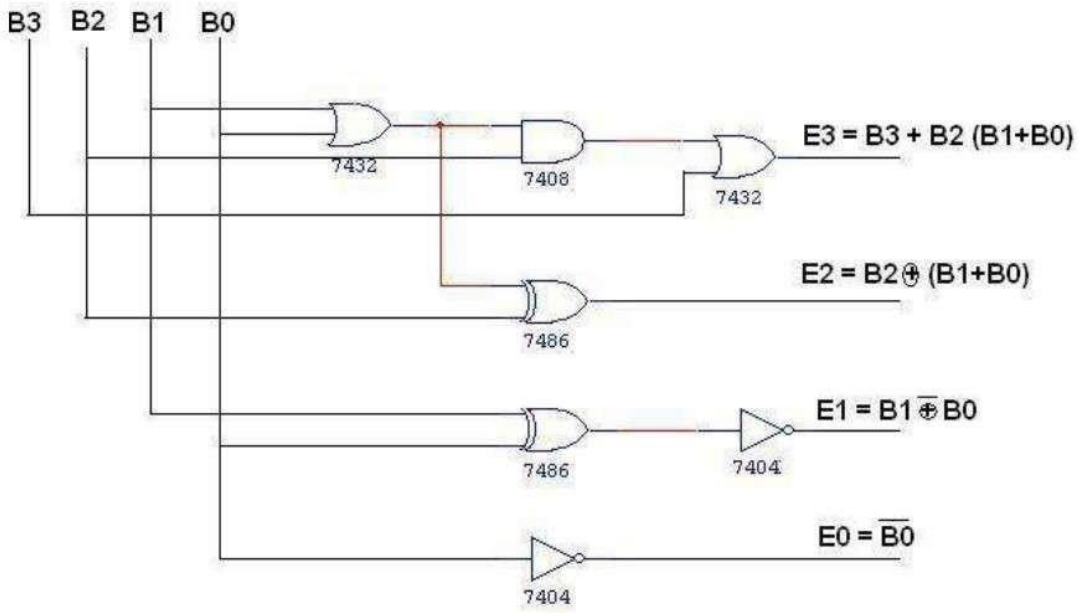
$$E0 = \overline{B0}$$

EXCESS-3 TO BCD CONVERTOR

TRUTH TABLE:

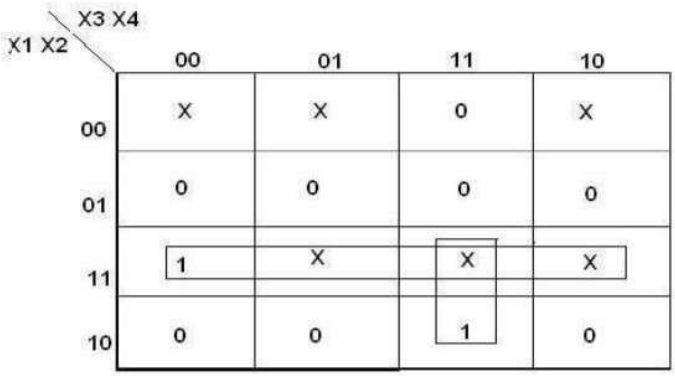
Excess – 3 Input				BCD Output			
B3	**B2**	**B1**	**B0**	**G3**	**G2**	**G1**	**G0**
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	1
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	1
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	1	0	0	1

LOGIC DIAGRAM:



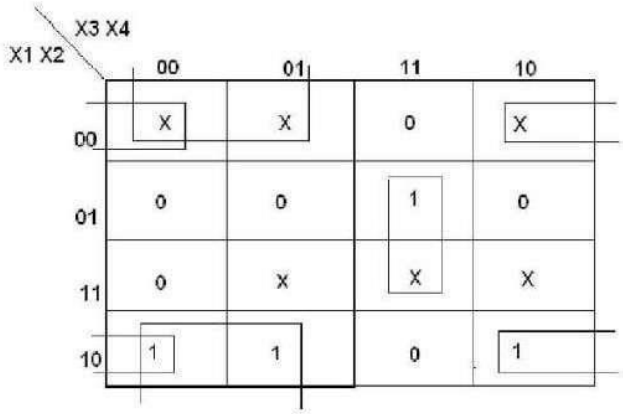
EXCESS-3 TO BCD CONVERTOR

K-Map for A:



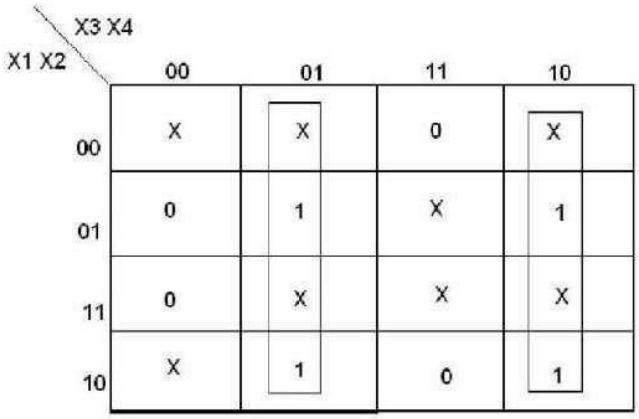
A = X1 X2 + X3 X4 X1

K-Map for B:



$$B = X2 \oplus (\overline{X3} + \overline{X4})$$

K-Map for C:



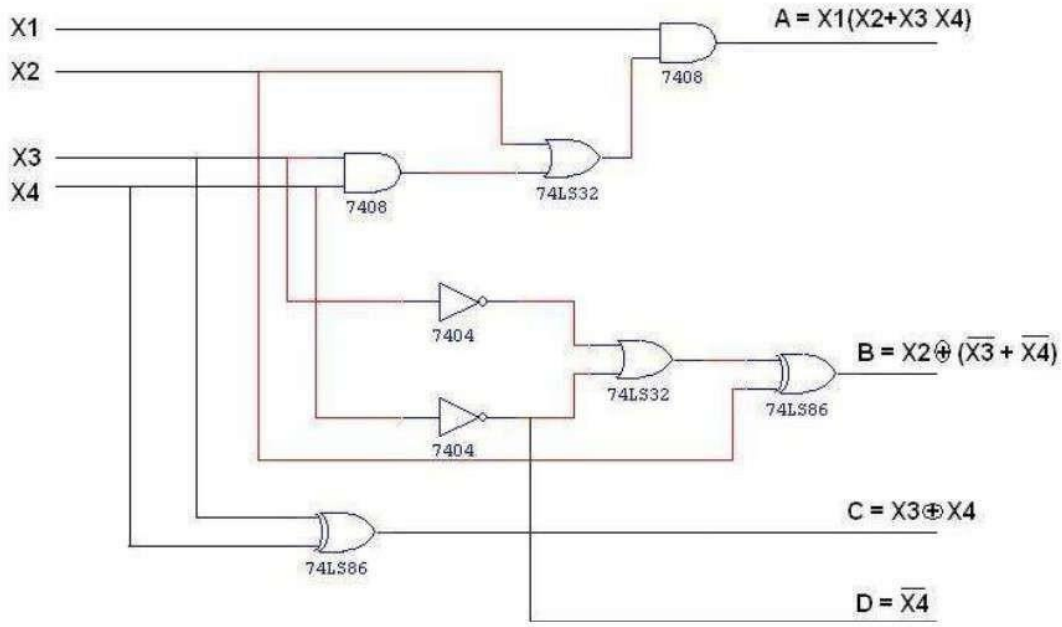
$$C = X3 \oplus X4$$

K-Map for D:

X1 X2 \ X3 X4	00	01	11	10
00	X	X	0	X
01	1	0	0	1
11	1	X	X	X
10	1	0	0	1

$D = \overline{X4}$

EXCESS-3 TO BCD CONVERTOR



PROCEDURE:

(i) Connections were given as per circuit diagram.
(ii) Logical inputs were given as per truth table
(iii) Observe the logical output and verify with the truth tables.

RESULT:

EX. NO:5a

DESIGN AND IMPLEMENTATION OF BCD ADDER

AIM:

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	IC	IC 7483	2
2.	AND GATE	IC 7408	1
3.	OR GATE	IC 7432	1
3.	IC TRAINER KIT	-	1
4.	PATCH CORDS	-	As required

THEORY:

4 BIT BCD ADDER:

Consider the arithmetic addition of two decimal digits in BCD, together with an input carry from a previous stage. Since each input digit does not exceed 9, the output sum cannot be greater than 19, the 1 in the sum being an input carry. The output of two decimal digits must be represented in BCD and should appear in the form listed in the columns.

ABCD adder that adds 2 BCD digits and produce a sum digit in BCD. The 2 decimal digits, together with the input carry, are first added in the top 4 bit adder to produce the binary sum.

PROCEDURE:

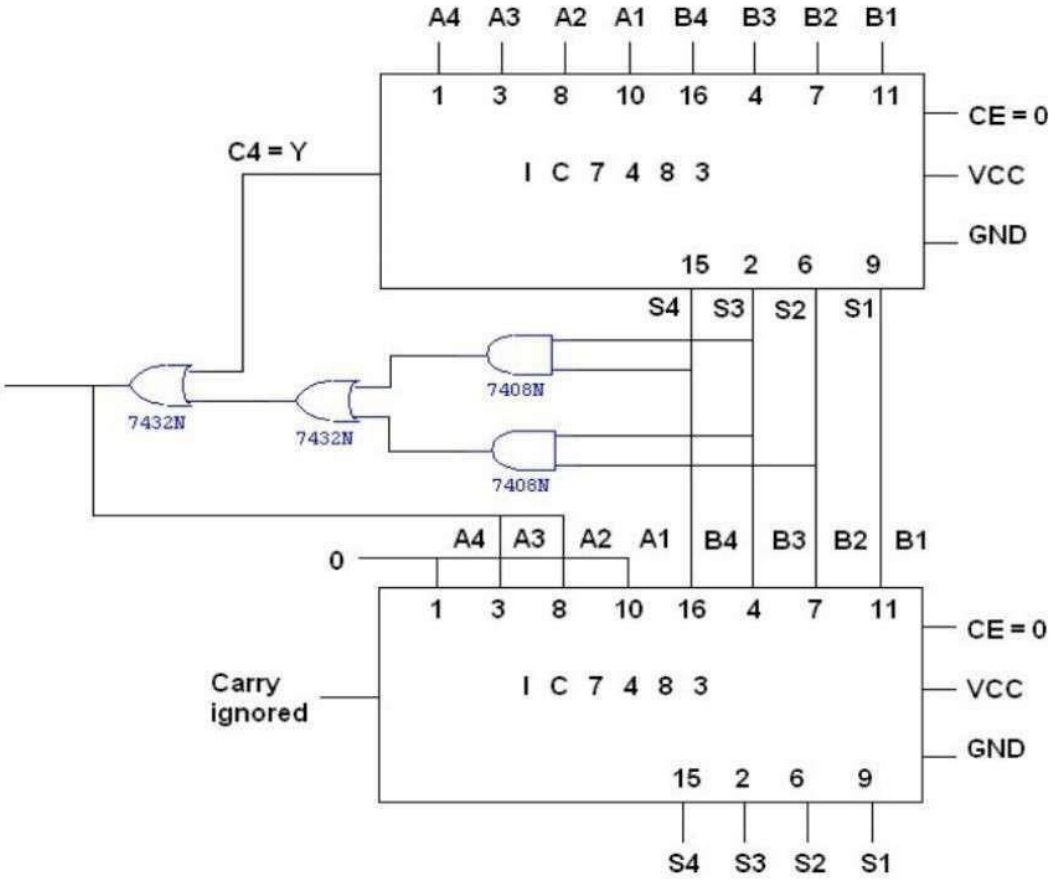
(i) Logical inputs were given as per truth table

(ii) Observe the logical output and verify with the truth tables.

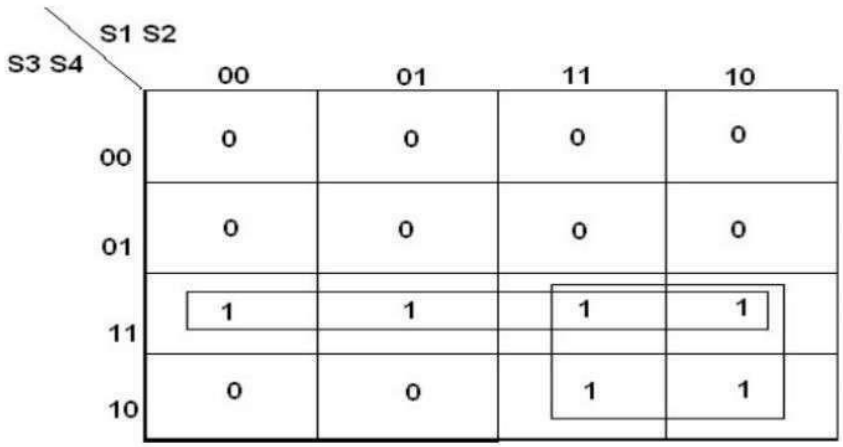
(iii) Connections were given as per circuit diagram.

LOGIC DIAGRAM:

BCD ADDER



K MAP



Y = S4 (S3 + S2)

TRUTH TABLE:

BCD SUM				CARRY
S4	S3	S2	S1	C
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

RESULT:

EX. NO:5b

DESIGN AND IMPLEMENTATION OF ENCODER AND DECODER

AIM:

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	3 I/P NAND GATE	IC 7410	2
2.	OR GATE	IC 7432	3
3.	NOT GATE	IC 7404	1
2.	IC TRAINER KIT	-	1
3.	PATCH CORDS	-	27

THEORY:

ENCODER:

An encoder is a digital circuit that performs inverse operation of a decoder. An encoder has 2^n input lines and n output lines. In encoder the output lines generates the binary code corresponding to the input value. In octal to binary encoder it has eight inputs, one for each octal digit and three output that generate the corresponding binary code. In encoder it is assumed that only one input has a value of one at any given time otherwise the circuit is meaningless. It has an ambiguila that when all inputs are zero the outputs are zero. The zero outputs can also be generated when D0 = 1.

DECODER:

A decoder is a multiple input multiple output logic circuit which converts coded input into coded output where input and output codes are different. The input code generally has fewer bits than the output code. Each input code word produces a different output code word i.e there is one to one mapping can be expressed in truth table. In the block diagram of decoder circuit the encoded information is present as n input producing 2^n possible outputs. 2^n output values are from 0 through out $2^n - 1$.

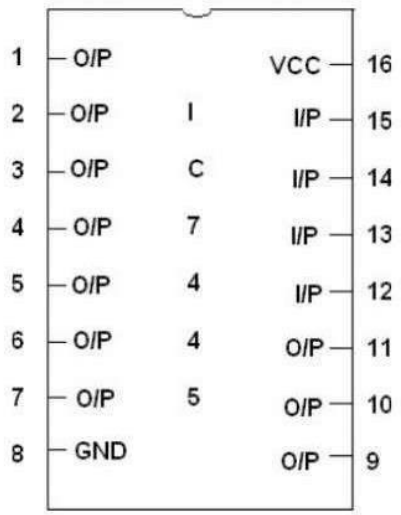
PROCEDURE:

(i) Connections are given as per circuit diagram.

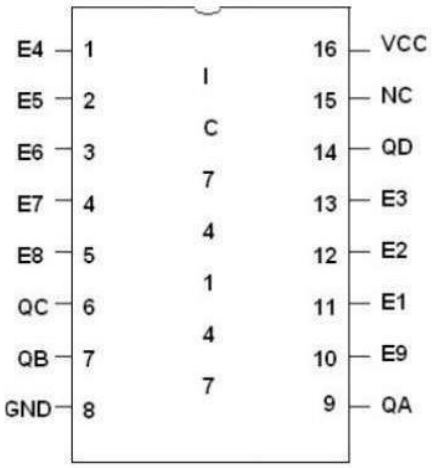
(ii) Logical inputs are given as per circuit diagram.

(iii) Observe the output and verify the truth table

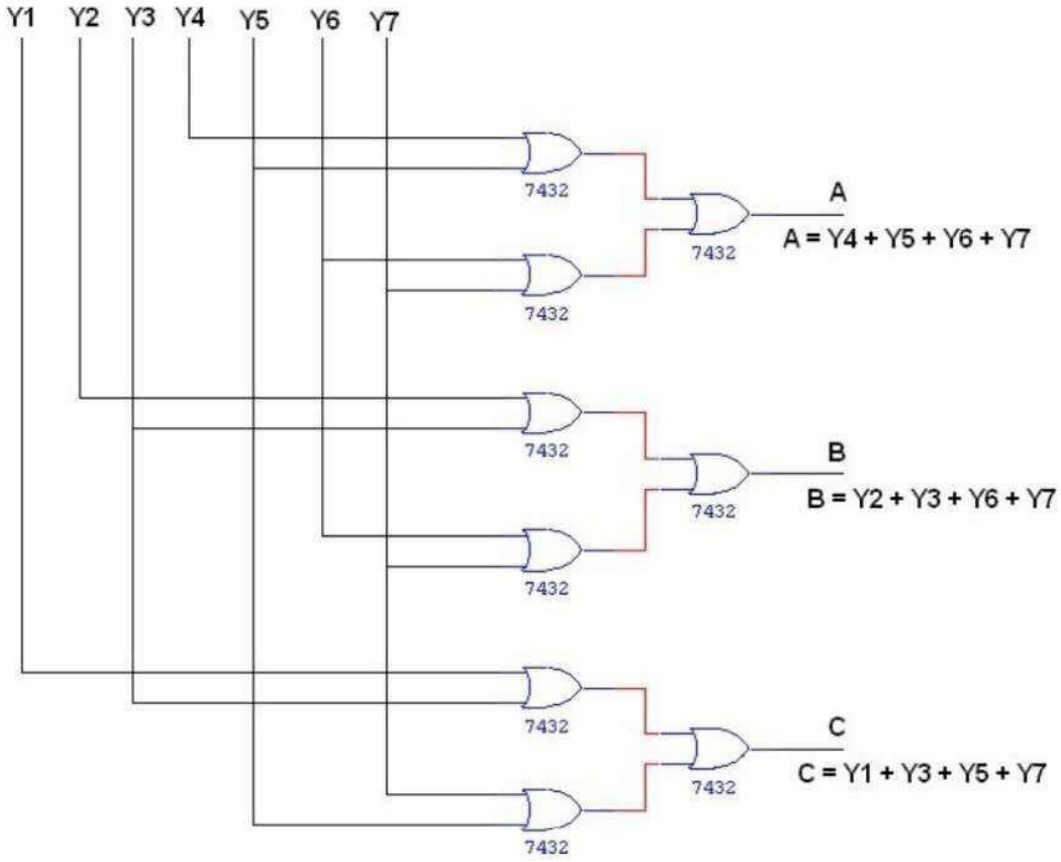
PIN DIAGRAM FOR IC 7445:
BCD TO DECIMAL DECODER:



PIN DIAGRAM FOR IC 74147:



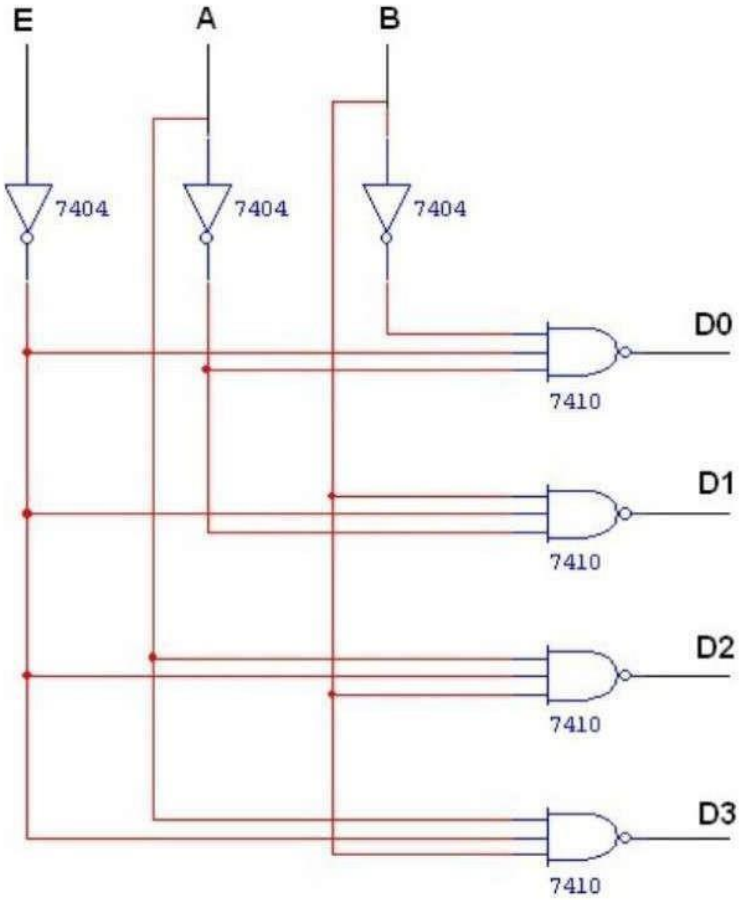
LOGIC DIAGRAM FOR ENCODER:



TRUTH TABLE:

INPUT							OUTPUT		
Y1	Y2	Y3	Y4	Y5	Y6	Y7	A	B	C
1	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	1
0	0	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	1	1	1	1

LOGIC DIAGRAM FOR DECODER:



TRUTH TABLE:

INPUT			OUTPUT			
E	**A**	**B**	**D0**	**D1**	**D2**	**D3**
1	**0**	**0**	**1**	**1**	**1**	**1**
0	**0**	**0**	**0**	**1**	**1**	**1**
0	**0**	**1**	**1**	**0**	**1**	**1**
0	**1**	**0**	**1**	**1**	**0**	**1**
0	**1**	**1**	**1**	**1**	**1**	**0**

RESULT:

Ex.No.-6 MULTIPLEXERS

AIM:

APPARATUS REQUIRED:

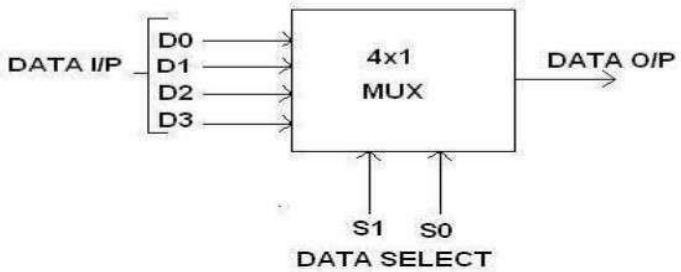
SL.NO.	COMPONENT	SPECIFICATION	QTY.
1.	3 I/P AND GATE	IC 7411	2
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
2.	IC TRAINER KIT	-	1
3.	PATCH CORDS	-	32

THEORY:

MULTIPLEXER:

Multiplexer means transmitting a large number of information units over a smaller number of channels or lines. A digital multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally there are 2^n input line and n selection lines whose bit combination determine which input is selected.

BLOCK DIAGRAM FOR MULTIPLEXER:



FUNCTION TABLE:

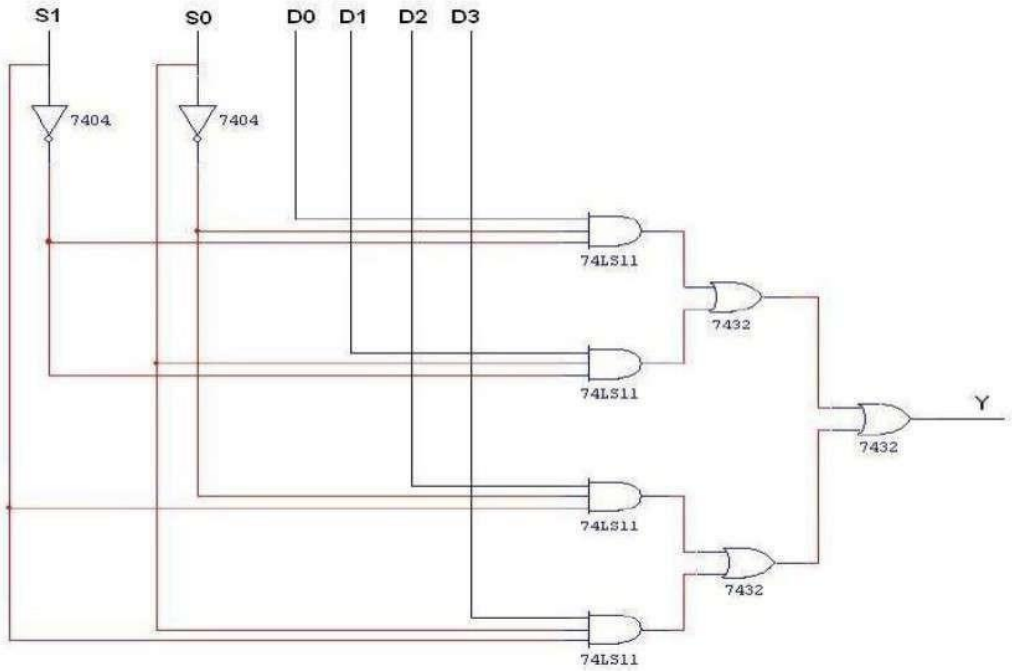
S1	S0	INPUTS Y
0	0	D0 → D0 S1' S0'
0	1	D1 → D1 S1' S0
1	0	D2 → D2 S1 S0'
1	1	D3 → D3 S1 S0

Y = D0 S1' S0' + D1 S1' S0 + D2 S1 S0' + D3 S1 S0

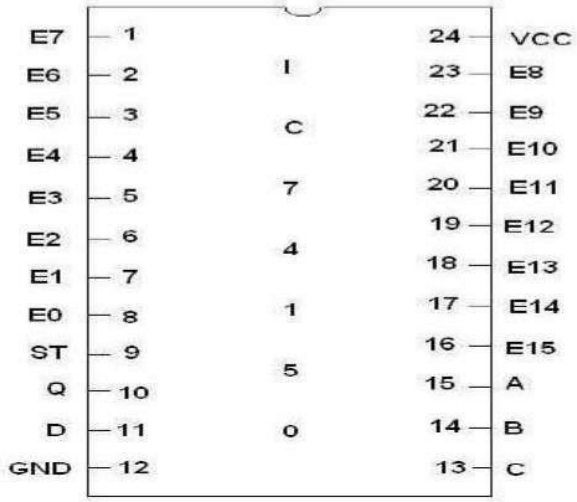
TRUTH TABLE:

S1	S0	Y = OUTPUT
0	0	D0
0	1	D1
1	0	D2
1	1	D3

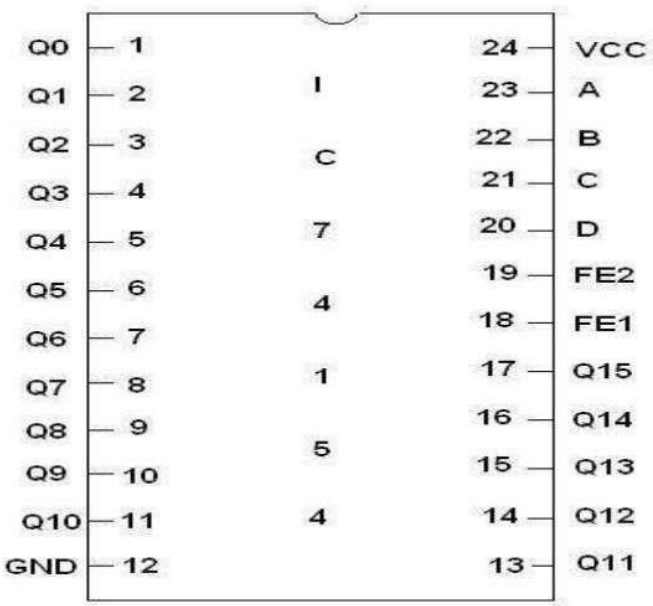
CIRCUIT DIAGRAM FOR MULTIPLEXER:



PIN DIAGRAM FOR IC 74150:



PIN DIAGRAM FOR IC 74154:



PROCEDURE:

(i) Connections are given as per circuit diagram.
(ii) Logical inputs are given as per circuit diagram.
(iii) Observe the output and verify the truth table.

RESULT:

Ex.No.-7 SYNCHRONOUS COUNTERS

AIM:

APPARATUS REQUIRED:

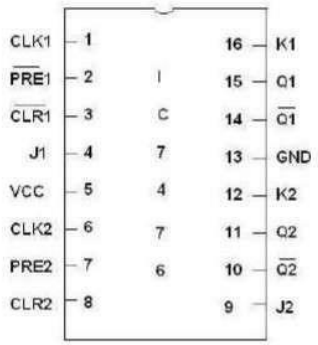
S.NO.	NAME OF THE APPARATUS	RANGE	QUANTITY
1.	Digital IC trainer kit		1
2.	JK Flip Flop	IC 7473	2
3.	D Flip Flop	IC 7473	1
4.	NAND gate	IC 7400	1
5.	Connecting wires		As required

THEORY:

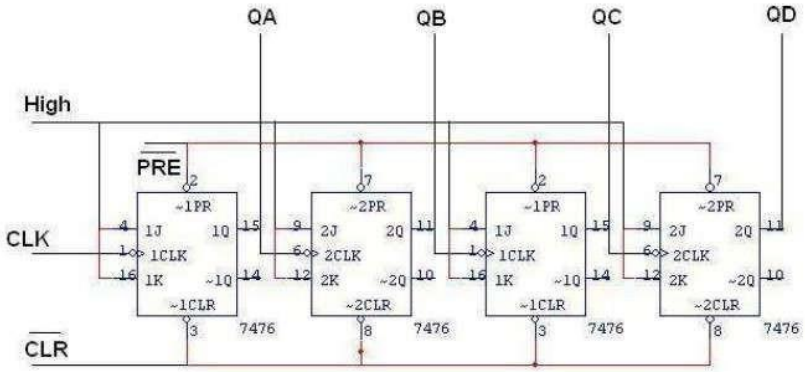
A synchronous decade counter is also called as ripple counter. In a ripple counter the flip flop output transition serves as a source for triggering other flip flops. In other words the clock pulse inputs of all the flip flops are triggered not by the incoming pulses but rather by the transition that occurs in other flip flops. The term asynchronous refers to the events that do not occur at the same time. With respect to the counter operation, asynchronous means that the flip flop within the counter are not made to change states at exactly the same time, they do not because the clock pulses are not connected directly to the clock input of each flip flop in the counter.

A counter is a register capable of counting number of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. A specified sequence of states appears as counter output. This is the main difference between a register and a counter. There are two types of counter, synchronous and asynchronous. In synchronous common clock is given to all flip flop and in asynchronous first flip flop is clocked by external pulse and then each successive flip flop is clocked by Q or Q output of previous stage. A soon the clock of second stage is triggered by output of first stage. Because of inherent propagation delay time all flip flops are not activated at same time which results in asynchronous operation.

PIN DIAGRAM FOR IC 7476:



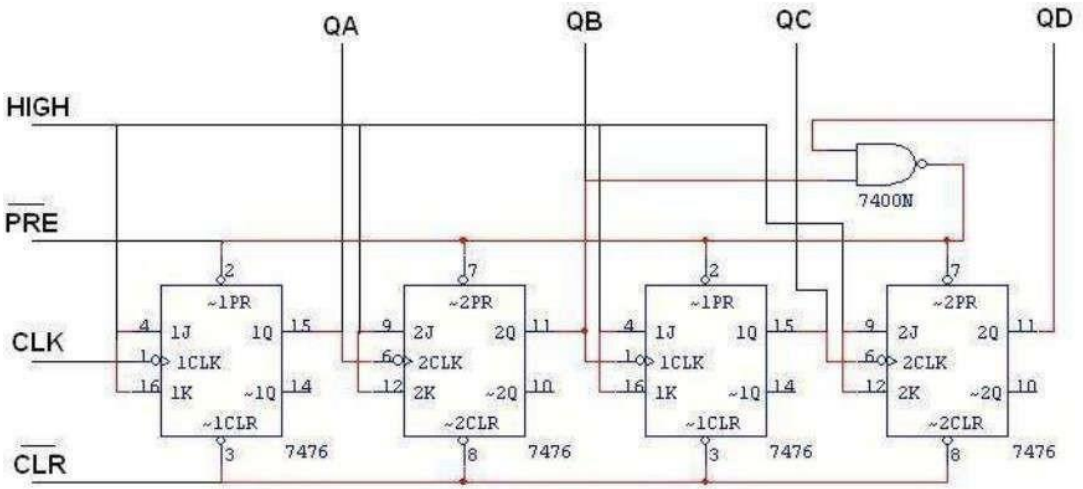
CIRCUIT DIAGRAM:



TRUTH TABLE:

CLK	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1
10	0	1	0	1
11	1	1	0	1
12	0	0	1	1
13	1	0	1	1
14	0	1	1	1
15	1	1	1	1

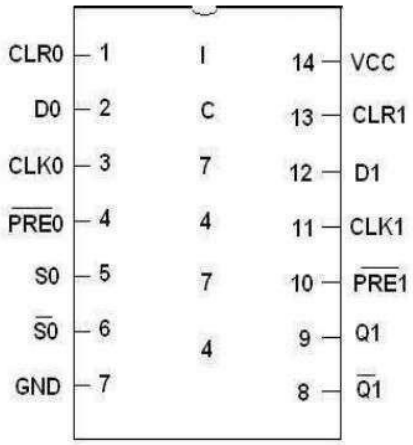
LOGIC DIAGRAM FOR MOD - 10 RIPPLE COUNTER:



TRUTH TABLE:

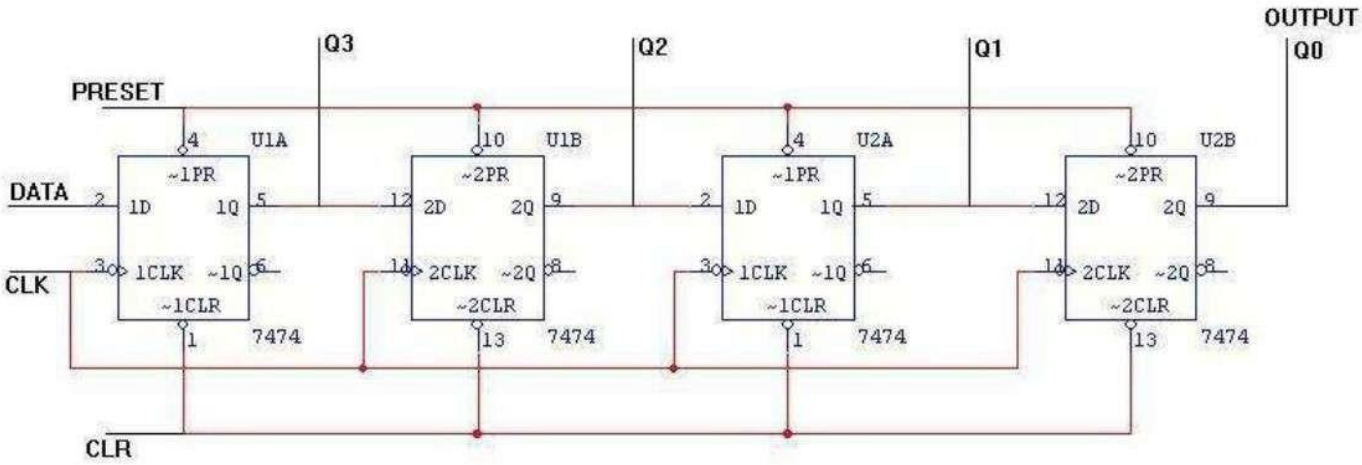
CLK	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1
10	0	0	0	0

PIN DIAGRAM:



SYNCHRONOUS COUNTER

LOGIC DIAGRAM:



TRUTH TABLE:

CLK	DATA	OUTPUT			
		Q_A	Q_B	Q_C	Q_D
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	0	0	1

PROCEDURE:

(i) Connections are given as per circuit diagram.
(ii) Logical inputs are given as per circuit diagram.
(iii) Observe the output and verify the truth table.

RESULT:

Ex.No.-8 SHIFT REGISTER

AIM:

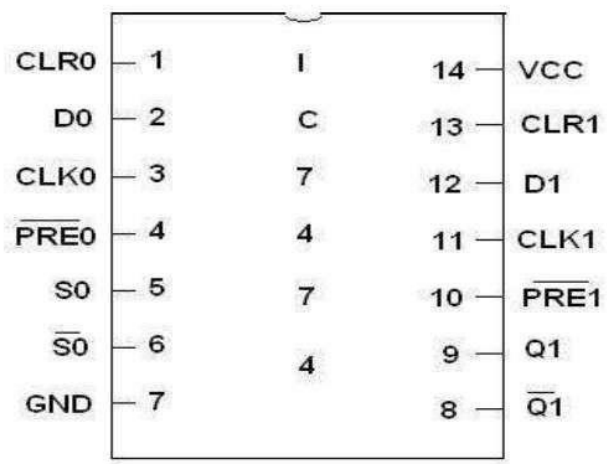
APPARATUS REQUIRED:

SL.NO.	COMPONENT	SPECIFICATION	QTY.
1.	D FLIP FLOP	IC 7474	2
2.	OR GATE	IC 7432	1
3.	IC TRAINER KIT	-	1
4.	PATCH CORDS	-	35

THEORY:

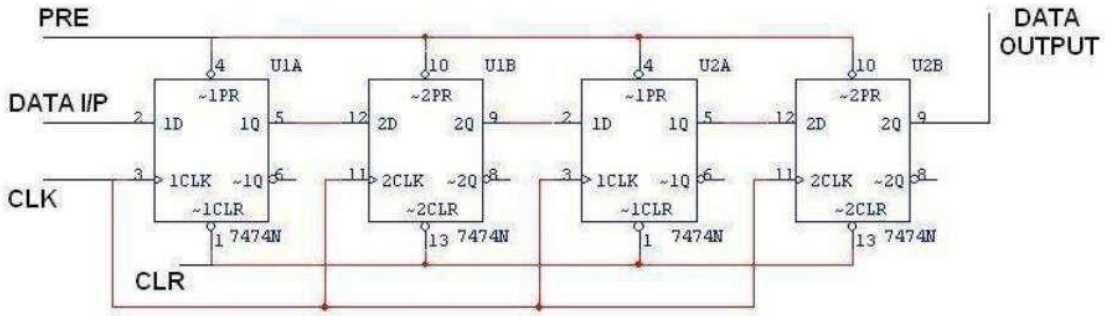
A register is capable of shifting its binary information in one or both directions is known as shift register. The logical configuration of shift register consist of a D-Flip flop cascaded with output of one flip flop connected to input of next flip flop. All flip flops receive common clock pulses which causes the shift in the output of the flip flop. The simplest possible shift register is one that uses only flip flop. The output of a given flip flop is connected to the input of next flip flop of the register. Each clock pulse shifts the content of register one bit position to right.

PIN DIAGRAM OF IC 7474:



SERIAL IN SERIAL OUT

LOGIC DIAGRAM:

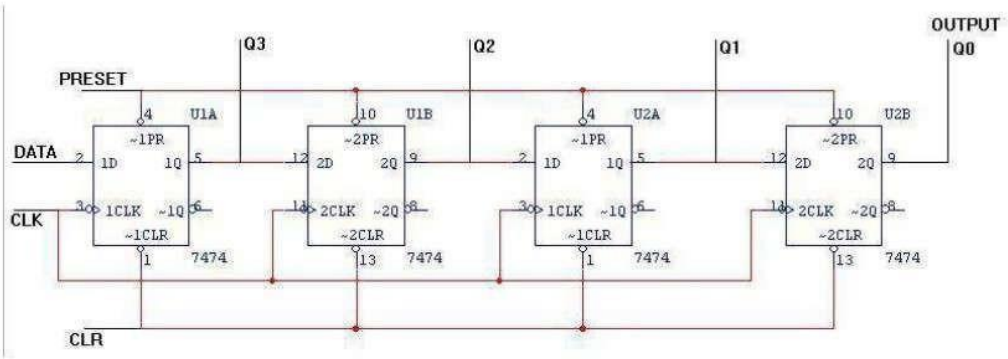


TRUTH TABLE:

CLK	Serial In	Serial Out
1	1	0
2	0	0
3	0	0
4	1	1
5	X	0
6	X	0
7	X	1

SERIAL IN PARALLEL OUT

LOGIC DIAGRAM:

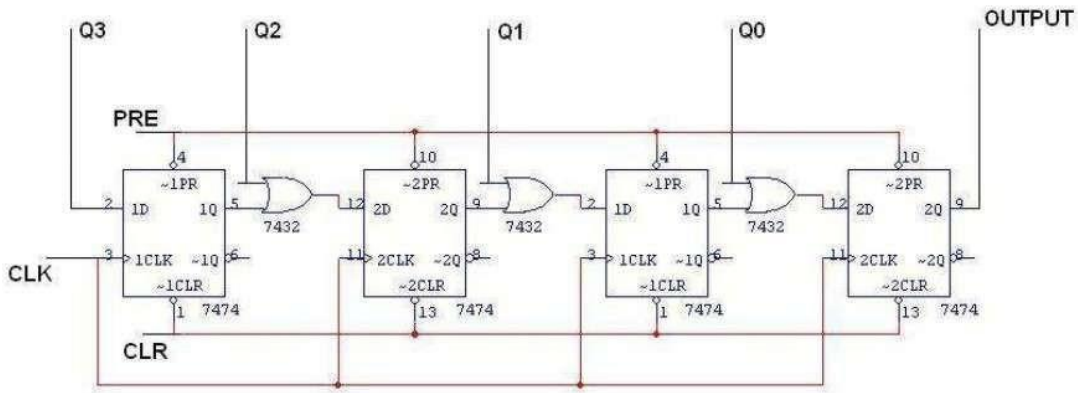


TRUTH TABLE:

CLK	DATA	OUTPUT			
		Q_A	Q_B	Q_C	Q_D
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	0	0	1

PARALLEL IN SERIAL OUT

LOGIC DIAGRAM:

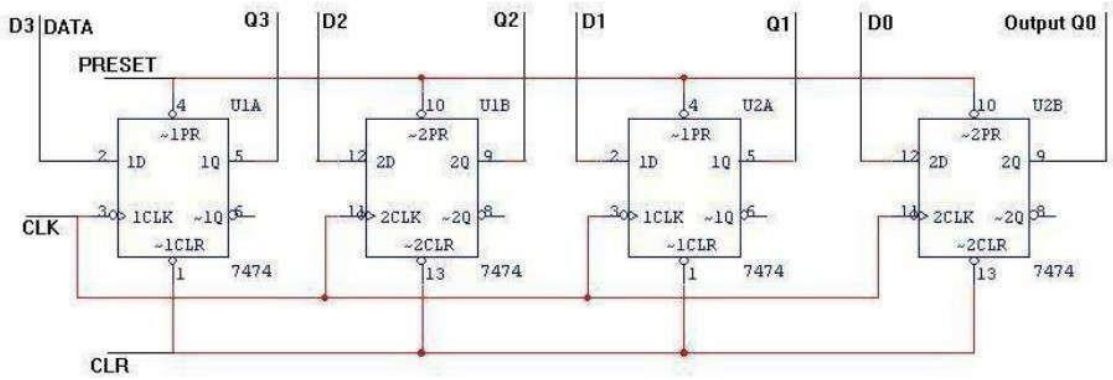


TRUTH TABLE:

CLK	Q3	Q2	Q1	Q0	O/P
0	1	0	0	1	1
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	1

PARALLEL IN PARALLEL OUT

LOGIC DIAGRAM:



TRUTH TABLE:

CLK	DATA INPUT				OUTPUT			
	D_A	D_B	D_C	D_D	Q_A	Q_B	Q_C	Q_D
1	1	0	0	1	1	0	0	1
2	1	0	1	0	1	0	1	0

PROCEDURE:

(i) Connections are given as per circuit diagram.
(ii) Logical inputs are given as per circuit diagram.
(iii) Observe the output and verify the truth table.

RESULT:

EX.NO: 9

SIMULATOR BASED STUDY OF COMPUTER ARCHITECTURE

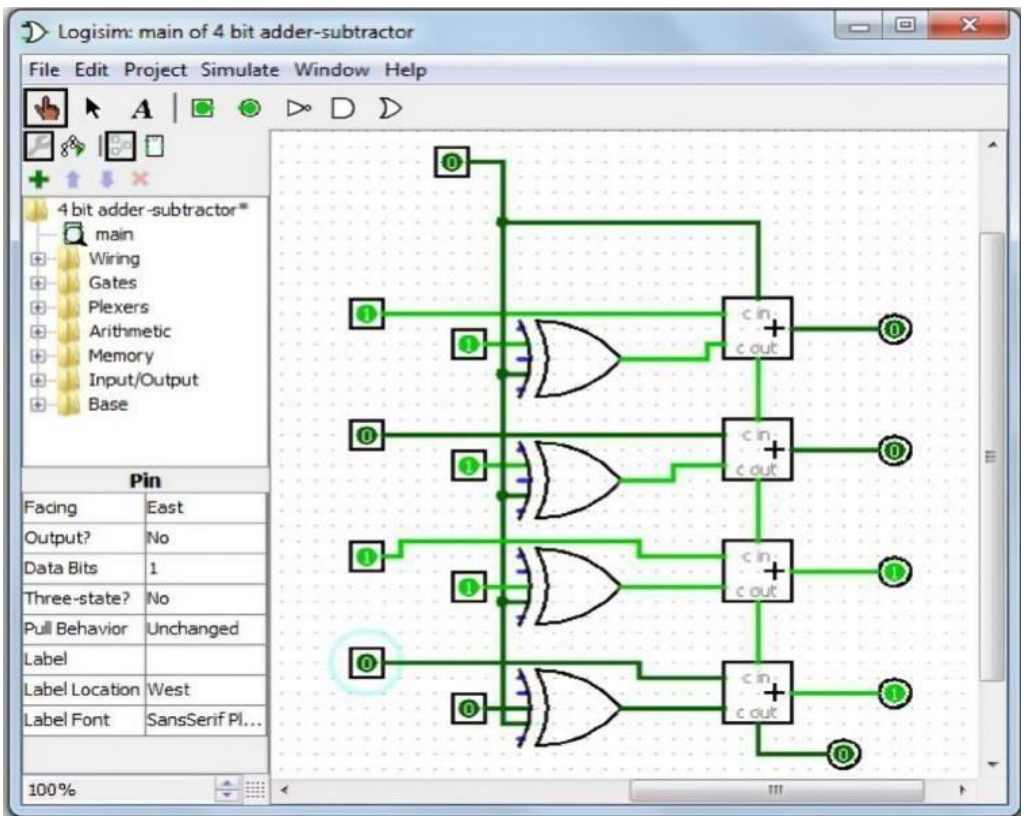
AIM:

THEORY:

Computer architecture is one of the important subjects offered at universities across the world. Teaching in traditional way can be insufficient if the teaching focus is solely on the textbook materials. One of the most critical aspects on teaching this discipline is how to support the theoretical concepts of the subject with appropriate practical experience, usually organized as laboratory experiments. But practically, students are unable to understand the subject. For this reason, may educators have begun using different computer architecture simulators based on hardware and software to solve this problem[5]. There are minly about three simulators: Logism, CEDAR and CPT sim.

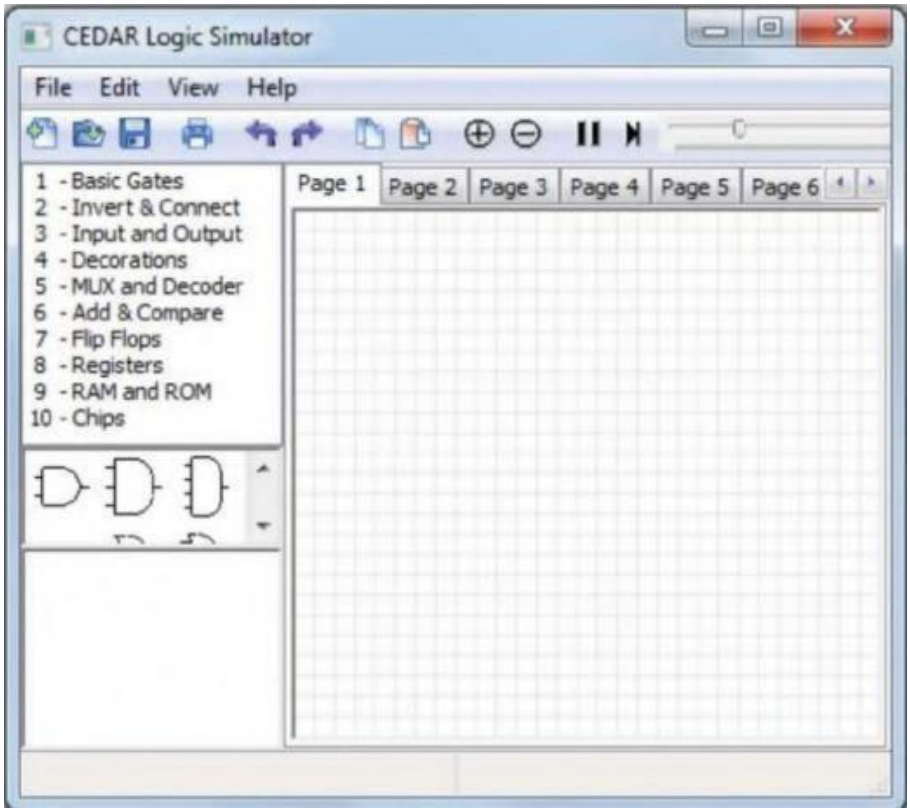
LOGISM:

Logisim is a simple software which can be used for implementing circuits with basic obasic gates. Users of this simulator can draw the circuits using the tool box available. The circuit automatically propagates circuit values through the circuit by selecting the suitable tool and the user can toggle the input conditions to learn how the circuit behaves in other situations. Student themselves were able to understand how to connect basic gates to make simple as wll as complex circuits with the help of Logisim.



CEDAR

CEDAR is a power simulator in which the student can implement a complete computer and will be able to understand the internal details of a computer more clearly. Using CEDAR simulator student can 1) build the entire computer hardware using fundamental logic gates; 2) write an assembler to translate the test program into machine level program; 3) load the program into the memory of the computer; and 4) run the test program on these hardware. After the implementation students can see how a computer executes a program and what are the signals generated during each clock pulse.



CPU Sim

CPU Sim is an interactive simulation tool in which the user can specify the details of the CPU to be simulated, such as register set, of microinstructions, set of machine instructions and set of assembly language instructions. Users of the tool can write their own machine or assembly language program and run on the CPU they have created. It simulates the computer architecture at register transfer level so that the students will get a better understanding about the system. User of the simulator has to specify the hardware units and the microinstructions for the CPU and then create the set of machine instructions. Corresponding to each machine instruction a sequence of microinstructions is to be formed.

RESULT: