STELLA MARY'S COLLEGE OF ENGINEERING

(Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai&Accredited by NAAC and
NBA(CSE & MECH)) Aruthenganvilai, Kallukatti Junction, AzhikalPost
Kanyakumari District – 629 202, Tamil Nadu

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Name of the Laboratory	**:**	**DIGITAL PRINCIPLES AND COMPUTER ORGANIZATION LABORATORY**
Laboratory Code	**:**	**CS3351**
Name of the Student	**:**	
University Register No.	**:**	
Branch	**:**	
Semester	**:**	**3**
Academic Year	**:**	**2022-2023 ODD**

INSTITUTE VISION

To emerge as a premiere institution, acknowledged as a center for excellence imparting technical education, creating technocrats who can address the needs of the society through exploration and experimentation and uplift mankind.

INSTITUTE MISSION

To provide an education that transforms students, through rigorous course-work and by providing an understanding of the needs of the society and the industry.

DEPARTMENT VISION

To produce Computer Science professionals who can accomplish path-breaking solutions for a better society, through quality technical education, on gaining the required inter-personal, entrepreneurial and computing skills.

DEPARTMENT MISSION

M1-To impart a holistic and experiential learning experience by making use of innovative teaching methodologies.

M2-To provide optimal technology solutions through collaborative and life- long learning for industry and societal needs with universal ethical values.

M3-To nurture leadership skills and facilitate various co-curricular and extra-curricular activities to implant the spirit of entrepreneurship.

M4-To provide industry-institute-interaction opportunities in order to motivate inter-disciplinary research capabilities with an inquiring mind.

Program Educational Objectives (PEOs)

PROGRAMME OUTCOMES (POs):

1.Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and

give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one‘s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Programme Specific Outcomes(PSOs)

At the completion of the programme, the students will be able to:

PSO1:

Use data management techniques and algorithmic thinking for Software Design and Development practices.

PSO2:

Develop reliable IT solutions based on the expertise in Distributed Applications Development, Web Designing and Networking for various societal needs and entrepreneurial practices ethically.

PSO3:

Manage multidisciplinary environments effectively through their interpersonal and analytical skills and be responsible members and leaders of the society.

Program Educational Objectives(PEOs)

PEO1:

Graduates will be competent in creating innovative technologies through inter-disciplinary research and comprehensive skills sets that are suitable for the global computing industry.

PEO2:

Graduates will be capable of managing leading positions with a broad understanding of the application of ethics in evolving computer-based solutions for the societal needs.

PEO3:

Graduates will imbibe entrepreneurial qualities and develop their career by upgrading their, communication, analytical and professional skills constantly.

STELLA MARY'S COLLEGE OF ENGINEERING

Aruthenganvilai, Azhikkal (P.O), Kanyakumari District- 629 202

Bonafide Certificate

This is to certify that this is a bonafide record of the work done by Mr./Ms.________________________________of ______semester B.E ., Department of Computer Science and Engineering, Register No. ____________________for the____ laboratory during the academic year 20 -20 (Odd/Even).

Faculty-in-charge

(with Name,Date,Designation&Dept)

Head of the department

(with Name,Date, &Dept)

Submitted for the Practical Examination held on ______________

Internal Examiner

External Examiner

CS3351 – DIGITAL PRINCIPLES AND COMPUTER ORGANIZATION LABORATORY

COURSE OBJECTIVES

1	To analyze and design combinational circuits.
2	To analyze and design sequential circuits
3	To understand the basic structure and operation of a digital computer.
4	To study the design of data path unit, control unit for processor and to familiarize with the hazards.
5	To understand the concept of various memories and I/O interfacing.

COURSE OUTCOMES

1	Design various combinational digital circuits using logic gates
2	Design and Test the Digital Logic Circuits
3	Design sequential circuits and analyze the design procedures
4	Study the basic structure and operation of computer architecture
5	Implementation of control unit techniques and the concept of pipelining
6	Understanding the hierarchical memory system, cache memories and virtual memory.
7	Understanding the different ways of communication with I/O devices and standard I/O interfaces.

CS3351- DIGITAL PRINCIPLES AND COMPUTER ORGANIZATION LAB

LIST OF DIGITAL EXPERIMENTS

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

INSTRUCTION TO THE CANDIDATES

1	Be on time
2	Students come with proper uniform and must wear shoes at all times
3	Follow safety precautions in the laboratory
4	Girl students should have their hair tucked in uniform
5	Exercise enough care and attention in handling of instrument
6	Ensure that the power is OFF before you start connecting up the circuit and after completion of lab experiments
7	Avoid ornaments during lab hours
8	Avoid direct contacts with energized circuits
9	Do not leave the lab without the permission of the faculty in-charge

CONTENTS

S.No		Name of the Experiment	Marks	Page no	Signature

EX.NO:1 **STUDY OF LOGIC**

GATES DATE :

AIM

To study about logic gates and verify their truth tables.

COMPONENTS REQUIRED

SL. No.	COMPONENT	SPECIFICATION	QUANTITY
1	AND Gate	IC 7408	1
2	OR Gate	IC 7432	1
3	NOT Gate	IC 7404	1
4	NAND Gate	IC 7400	1
5	NOR Gate	IC 7402	1
6	X-OR Gate	IC 7486	1
7	NAND Gate	IC 7410	1
8	AND Gate	IC 7411	1
9	IC Trainer Kit	-	1
10	Connecting Wires	-	Few

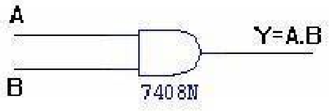
THEORY

A logic gate is an electronic circuit which makes logical decisions. To arrive at these decisions, the most common logic gates used are OR, AND, NOT, NAND and NOR gates. The NAND and NOR gates are called as the Universal gates. The exclusive OR (XOR) gate is another logic gate which can be constructed using basic gates such as AND, OR and NOT gates. Each gate has two or more input and only one output except for the Not gate, which has only one input. The logic gates are the building blocks of hardware which are available in the form of various IC families. Each gate has a distinct logic symbol and its operation can be described by means of an algebraic function. The relationship between input and output variables of each gate can be represented in a tabular form called a truth table.

AND GATE

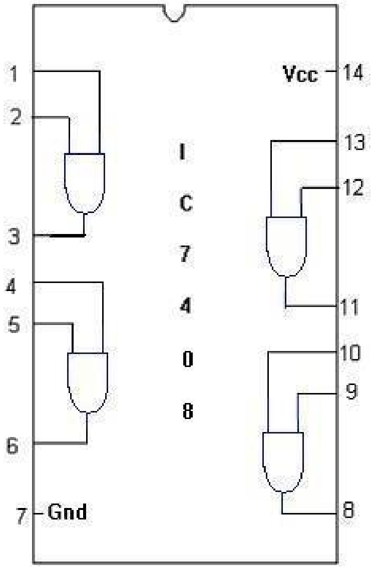
The AND gate performs a logical multiplication commonly known as AND function. The AND gate has two or more inputs and a single output. The output of an AND gate is HIGH (1) only when all the inputs are high. Even if any one of the inputs is low, the output will be LOW (0). If A and B are the input variables of an AND gate and Y is its output, then Y = A.B Where the dot (.) denotes the AND operation.

SYMBOL **PIN DIAGRAM**



TRUTH TABLE

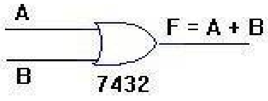
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1



OR GATE

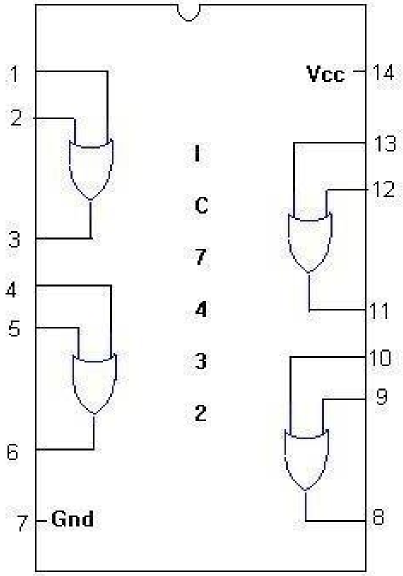
The OR gate performs a logical addition commonly known as OR function. The OR gate has two or more inputs and a single output. The operation of OR gate is such that a HIGH on the output is produced when any of the inputs is high. The output is Low only when all the inputs are low. If A and B are the input variables of an AND gate and Y is its output, then Y = A+B Where the symbol (+) denotes the OR operation.

SYMBOL : PIN DIAGRAM :



TRUTH TABLE

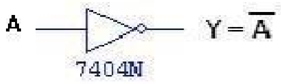
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1



NOT GATE

The NOT gate performs the basic logical function called inversion or complementation.The purpose of this gate is to convert one logic level into the opposite logic level. It has oneinput and one output. When a HIGH level is applied to an inverter, a LOW level appears as itsoutput and vice versa. If A is an input variable of a NOT gate and Y is its output, then Y=A.

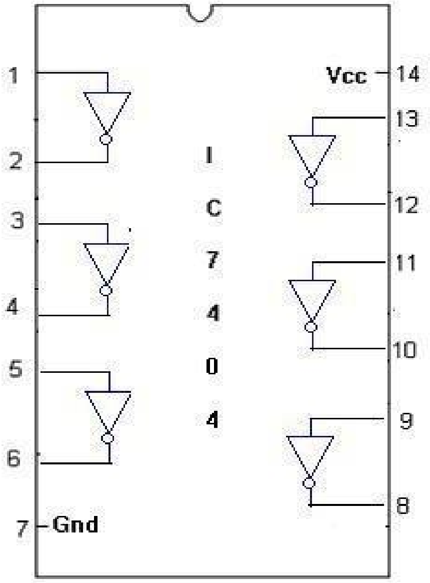
SYMBOL



TRUTH TABLE :

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

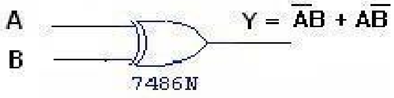
PIN DIAGRAM



EXCLUSIVE-OR (X-OR) GATE

An Exclusive-OR gate is a gate with two or more inputs and one output. The output of XOR gate is high when any one of the inputs is high. The output is low when both the inputs are low and both the in~~pu~~ts are high. If A and B are the input variables of a XOR gate and Y is its output, then Y = AB+AB

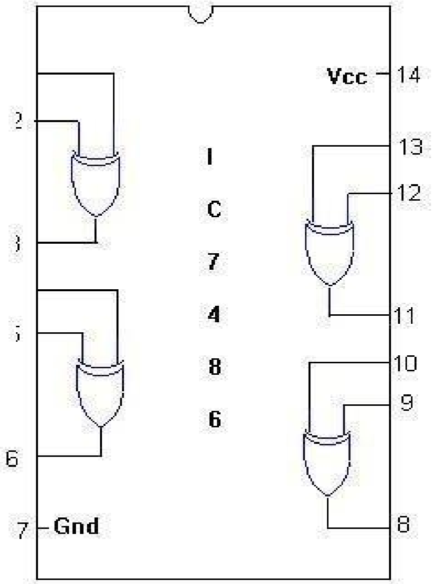
SYMBOL



TRUTH TABLE :

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

PIN DIAGRAM

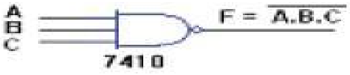


NAND GATE

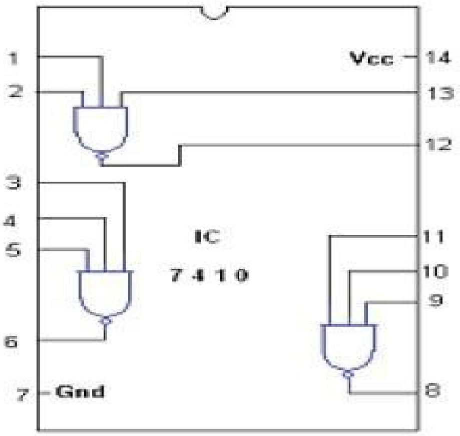
The NAND gate is a contraction of AND-NOT. It has two or more inputs and a single output. When all inputs are high, the output is LOW. If any one or both the inputs are low, then the output is HIGH. If Aand B are the input variables of a NAND gate and Y is its output, then Y = AB.

3-INPUT NAND GATE :

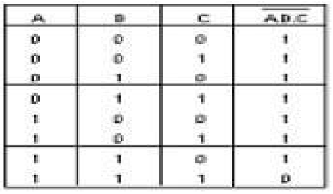
SYMBOL:



PIN DIAGRAM:



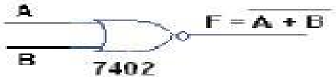
TRUTH TABLE



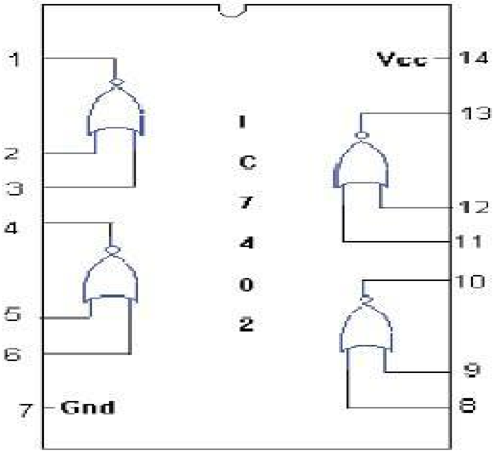
A	B	C	$\overline{A.B.C}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

NOR GATE:

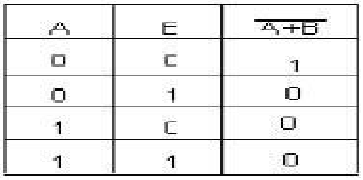
SYMBOL:



PIN DIAGRAM:



TRUTH TABLE

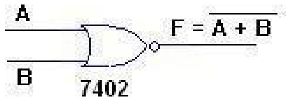


A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

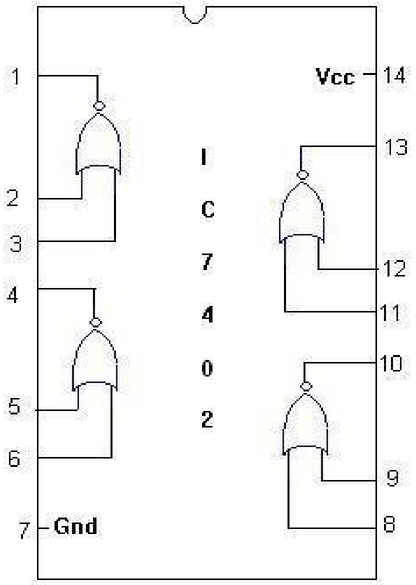
NOR GATE

The NOR gate is a contraction of OR-NOT. It has two or more inputs and only one output. The output is HIGH when both inputs are low. The output is LOW if anyone or both inputs are high. If A and B are the input variables of a NOR gate and Y is its output, then Y = A+B.

SYMBOL :



PIN DIAGRAM :



TRUTH TABLE

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

Thus all logic gates are studied and their truth tables are verified.

EX.NO: 2

DATE :

VERIFICATION OF BOOLEAN THEOREMS USING DIGITAL LOGIC GATES

AIM

To verify the Boolean Theorems using logic gates

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 individualcomplements.

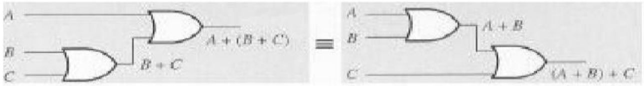
   A+B = A.B

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

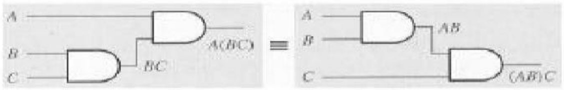
   A.B = A+B

Associative Laws of Boolean Algebra

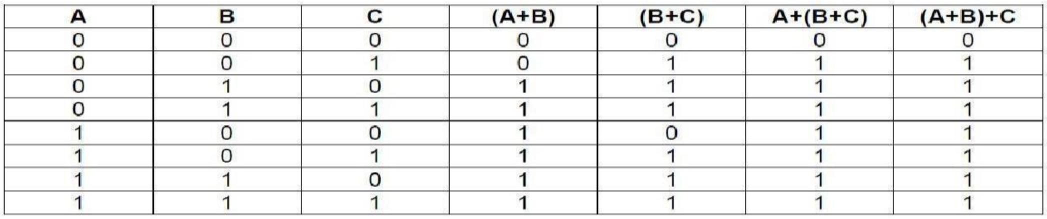
A + (B + C) = (A + B) + C



A • (B • C) = (A • B) • C

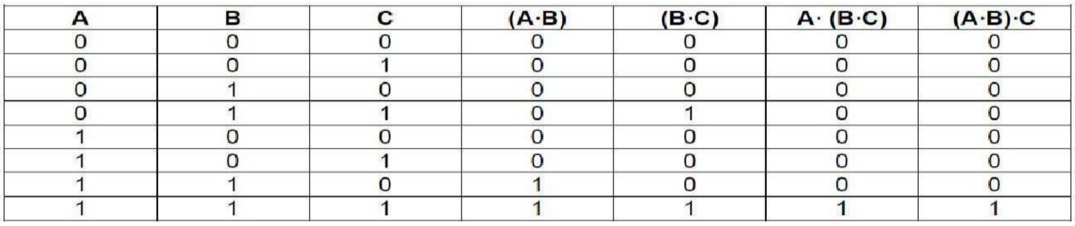


Proof of the Associative Property for the OR operation: (A+B)+C = A+(B+C)



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

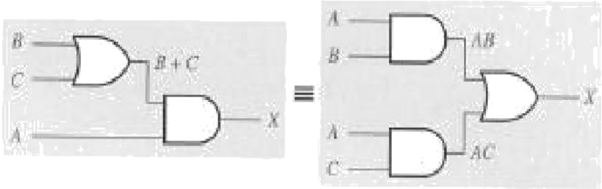
Proof of the Associative Property for the AND operation: (A·B)·C = A·(B·C)



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

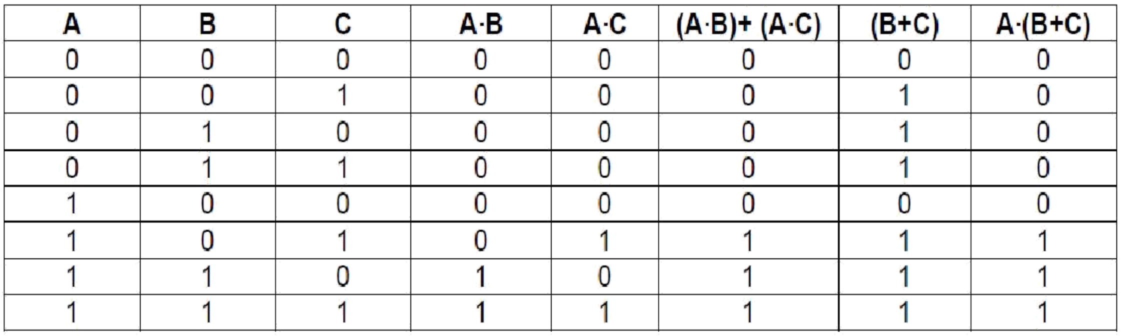
Distributive Laws of Boolean Algebra

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



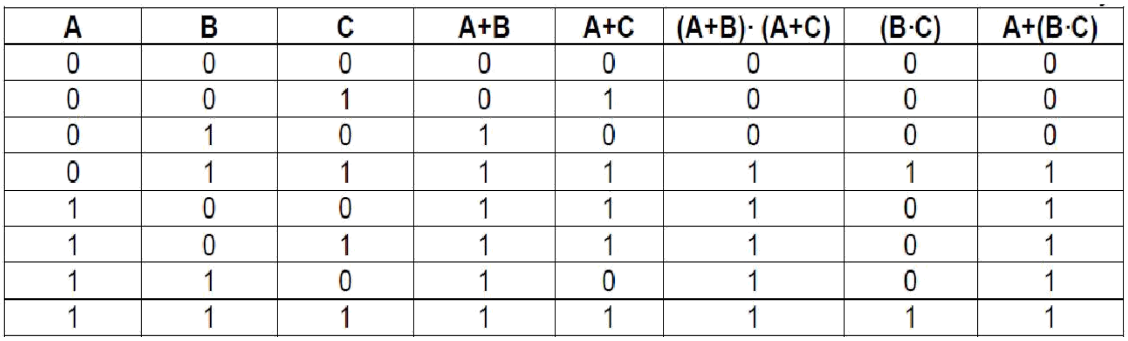
$X = A(B + C)$ $X = AB + AC$

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

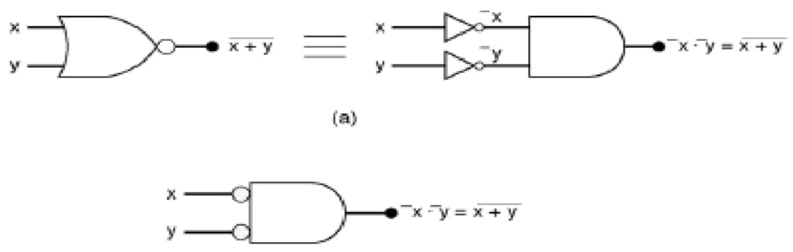


A	B	C	A+B	A+C	(A+B)· (A+C)	(B·C)	A+(B·C)
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	1	1	1	1

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.



(a)

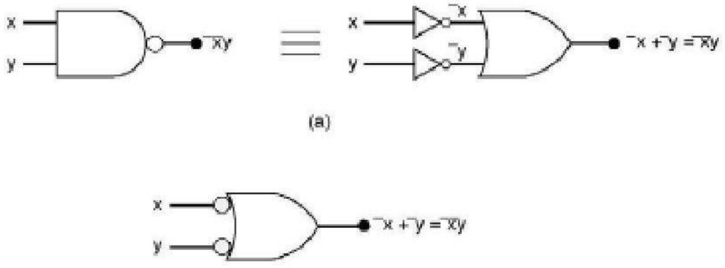
(b)

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

A	B	A·B	$\overline{A \bullet B}$	$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
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



(a)

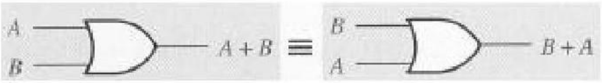
(b)

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

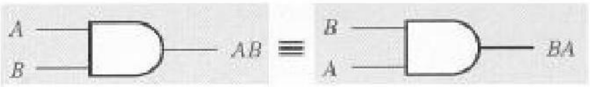
A	B	A+B	$\overline{A + B}$	$\overline{A}$	$\overline{B}$	$\overline{A} \bullet \overline{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

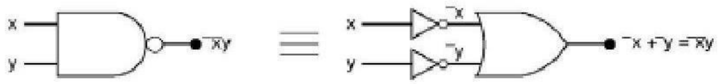
Commutative Laws of Boolean Algebra

A + B = B + A

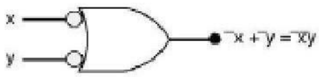


A • B = B • A





(a)



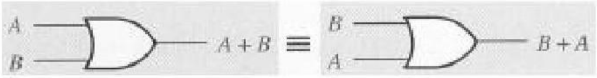
(b)

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

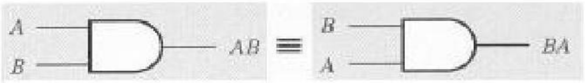
A	B	A+B	$\overline{A+B}$	$\overline{A}$	$\overline{B}$	$\overline{A}\bullet\overline{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

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 identityelement 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:

Thus the above stated Boolean laws are verified.

EX NO: 3 **IMPLEMENTAION OF 4-BIT BINARY ADDER/SUBSTRACTOR CIRCUITS**

DATE :

AIM

To design and implement 4-bit adder and substractor using basic gates and MSI device IC 7483.

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 toleft, 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 C0 and it ripples through the full adder to the output carry C4.

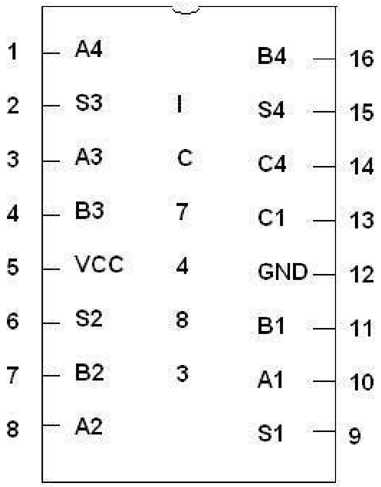
4 -BIT BINARY SUBTRACTOR:

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 C0 must be equal to 1 when performing subtraction.

4-BIT BINARY ADDER/SUBTRACTOR:

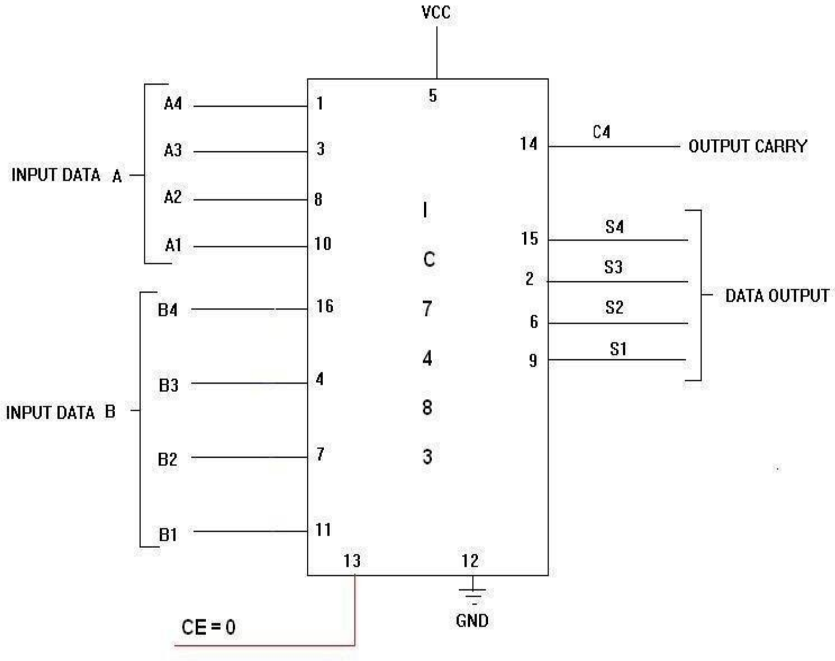
The addition and subtraction operation can be combined into one circuit with one common binary adder. The mode input M controls the operation. When M=0, the circuit is adder circuit. When M=1, it becomes subtractor.

PIN DIAGRAM FOR IC7483



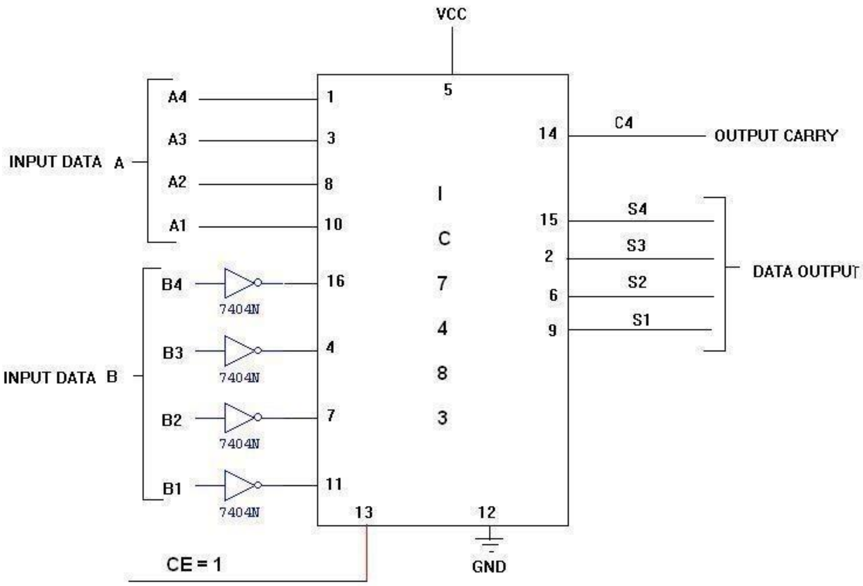
4-BIT BINARY ADDER

LOGIC DIAGRAM



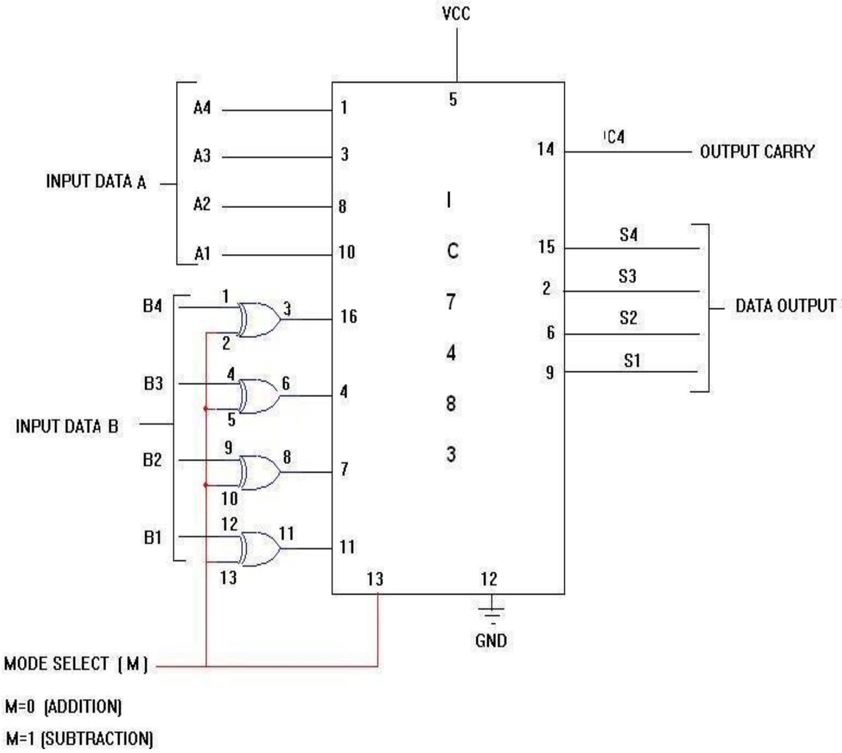
4-BIT BINARY SUBSTRACTOR

LOGIC DIAGRAM



4-BIT BINARY ADDER/SUBSTRACTOR

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:

Thus the 4-bit adder and subtractor using basic gates and MSI device IC 7483 is designed and implemented.

EX.NO: 4 **DESIGN AND IMPLEMENTATION OF CODE CONVERTER**

DATE:

AIM

To design and implement 4-bit

(i) Binary to gray code converter
(ii) Gray to binary code converter
(iii) BCD to excess-3 code converter
(iv) Excess-3 to BCD code converter using logic gates.

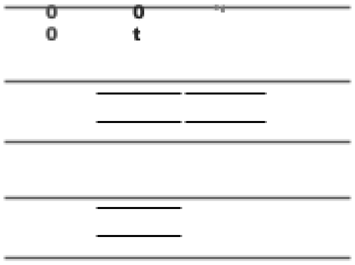
COMPONENTS REQUIRED

SL. No.	COMPONENT	SPECIFICATION	QUANTITY
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	Connecting Wires	-	Few

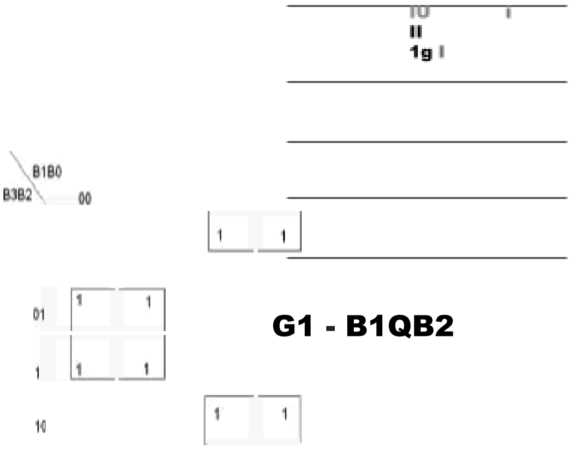
THEORY

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.

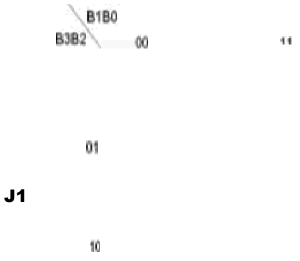
Binary input				Gray code output			
1	0	0	0	1	1	0	0



G2- B3@B2

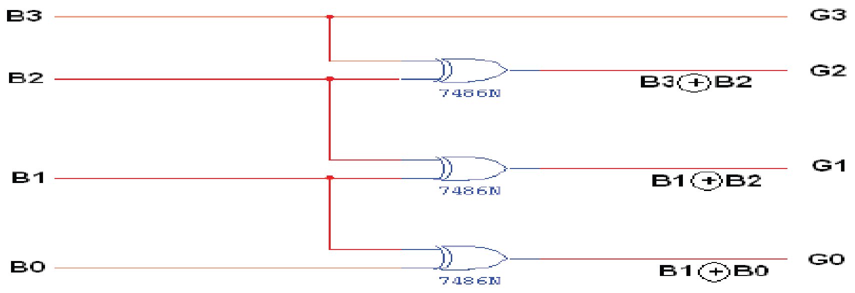


G1 - B1QB2



G0 - B1 @B0

LOGIC DIAGRAM:



GRAY CODE TO BINARY CONVERTER

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 B3:

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

B3 = G3

K-Map for B2:

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

B2=G3'G2 + G3G2'

B2 = G3⊕G2

K-Map for B1:

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

K-Map for BO

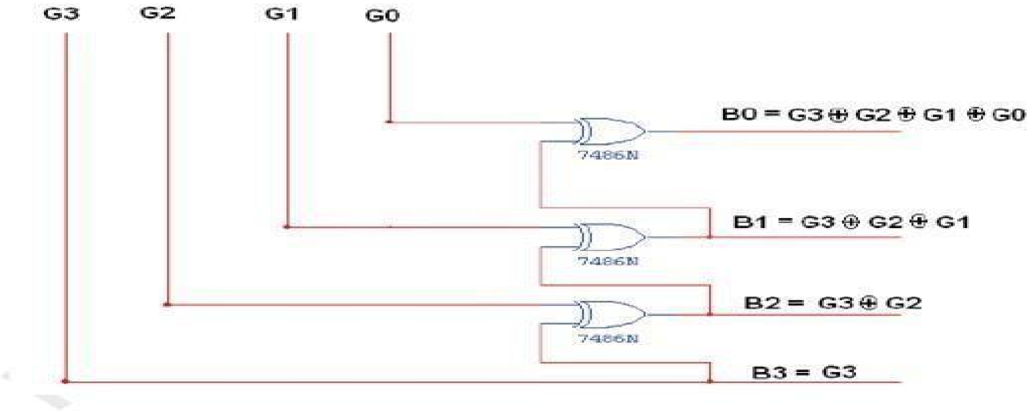
G3G2 \ G1G0	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

B1=G3'G2'G1+G3'G2G1'+G3G2G1+G3G2'G1'
= G1(G3'G2'+G3G2) + G1'(G3'G2+G2G3')
= $G1 \oplus (G2 \oplus G3)$

B0=G3'G2'G1'G0+G3'G2'G1G0'+G3'G2G1'G0'
+G3'G2G1G0+ G3G2 G1'G0+G3G2G1G0'
+G3G2'G1'G0'+G3G2'G1G0

B0 = G3 ⊕ G2 ⊕ G1 ⊕ G0

LOGIC DIAGRAM:

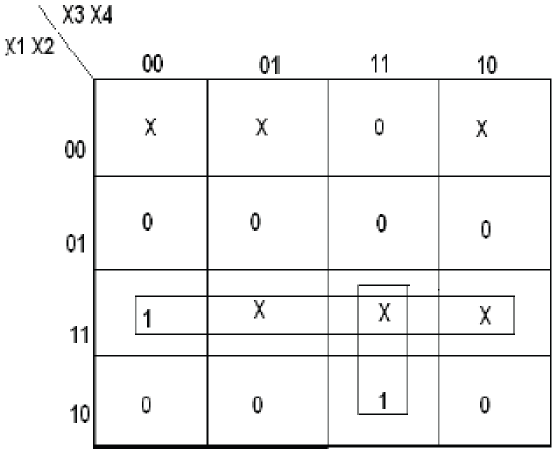


<u>EXCESS-3 TO BCD CONVERTER</u>

TRUTH TABLE:

Excess – 3 Input				BCD Output			
X1	**X2**	**X3**	**X4**	**A**	**B**	**C**	**D**
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

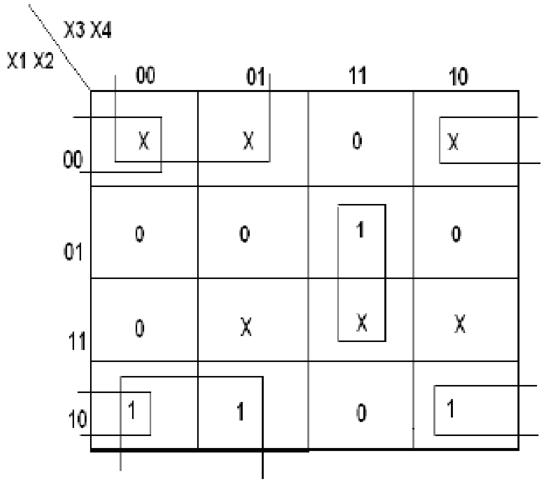
K-Map for A:



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

A = X1 X2 + X3 X4 X1
= X1(X2 + X3 X4)

K-Map for B:

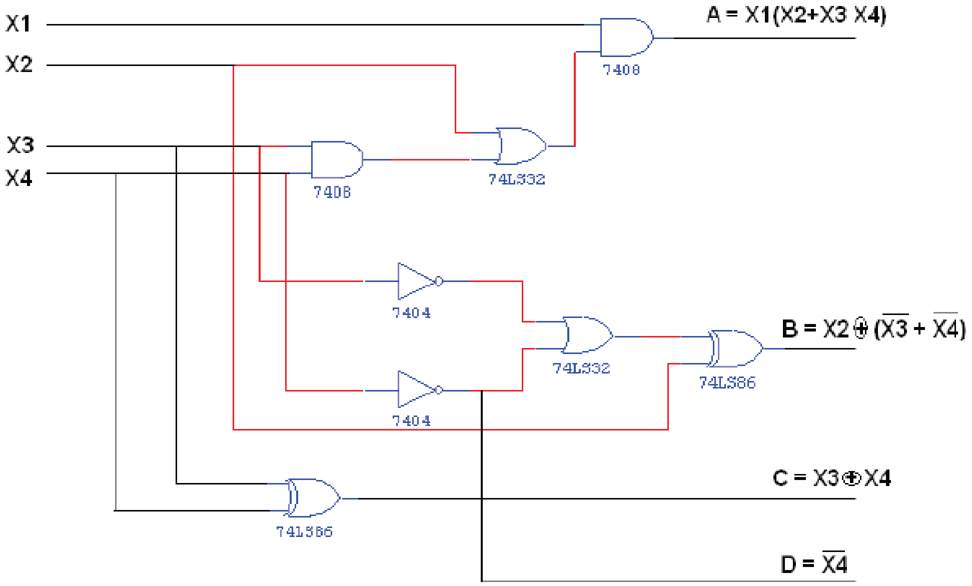


X1 X2 \ X3 X4	00	01	11	10
00	X	X	0	X
01	0	0	1	0
11	0	X	X	X
10	1	1	0	1

B =X2'X3'+ X2'X4' +X2X3X4
=X2'(X3'+X4')+X2X3X4

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

LOGIC DIAGRAM



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

Thus the code converters are designed using logic gates and their truth tables are verif

,

EX.NO :5 IMPLEMENTATION OF BCD ADDER, ENCODER AND DECODER CIRCUITS

DATE :

AIM

To design and implement BCD adder, Encoder and decoder using logic gates.

COMPONENTS 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
4.	IC Trainer Kit	-	1
5.	Connecting Wires	-	Few

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. A BCD adder that adds2 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.

ENCODER

An encoder is a digital circuit that performs inverse operation of a decoder. An encoder has 2n inputlines 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 giventime otherwise the circuit is meaningless. It has an ambiguity that when all inputs are zero the outputs are zero. The zero outputs can also be generated when D0 = 1.

DECODER

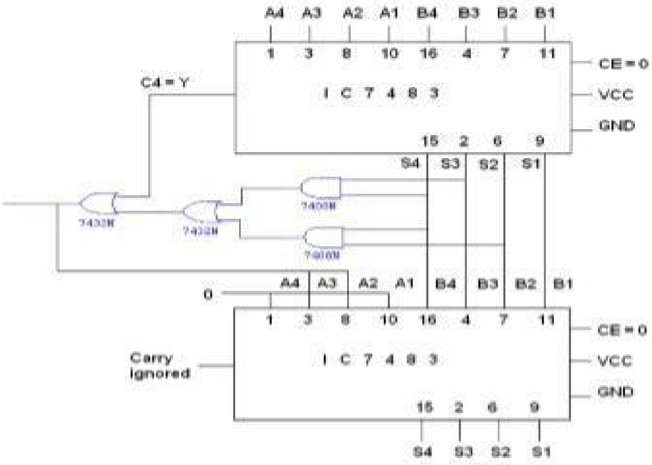
A decoder is a multiple input multiple output logic circuit which converts coded input into coded output where input and output codes are different. The input code generally has fewer bits than the output code. Each

,

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 2n possible outputs. 2n output values are from 0 through out 2n – 1.

LOGIC DIAGRAM:

BCD ADDER



K Map for Y

Y = S4 S3 + S4 S2

S3 S4 \ S1 S2	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

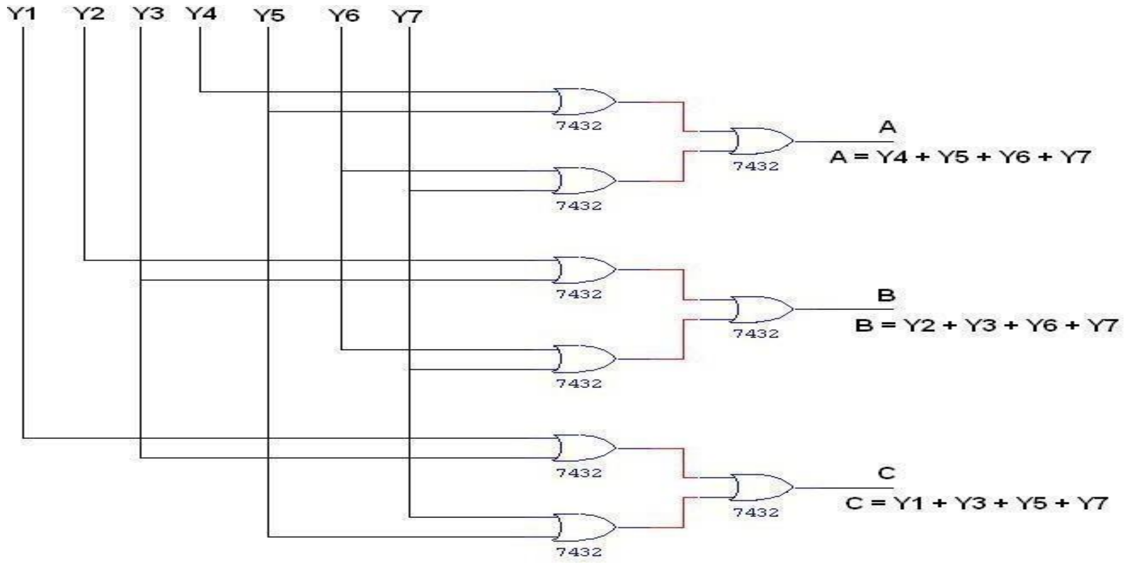
TRUTH TABLE:

BCD SUM				CARRY
S3	S2	S1	S0	Y
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

OUTPUT :

Sl.No	A				B				Carry	Sum			
	A4	A3	A2	A1	B4	B3	B2	B1	C	S4	S3	S2	S1
1													
2													
3													

LOGIC DIAGRAM FOR ENCODER

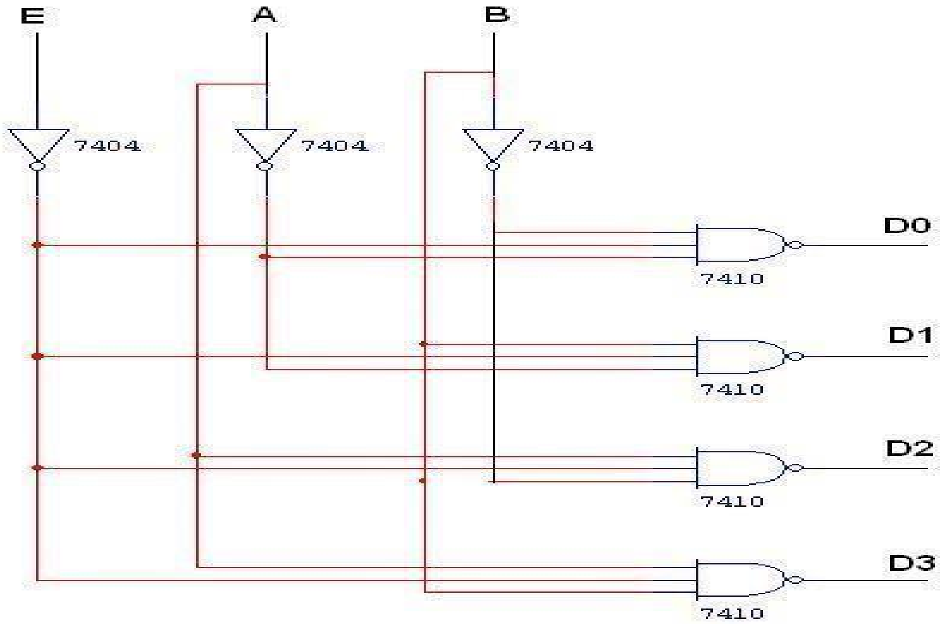


TRUTH TABLE

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

A= Y4 + Y5 + Y6 + Y7
B= Y2 + Y3 + Y6 + Y7
C = Y1 + Y3 + Y5 + Y7

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

D0 = E’A’B’

D1 = E’A’B

D2 = E’AB’

D3 = E’AB

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

Thus the BCD adder, encoder and decoder are designed and implemented using logic gates.

EX.NO:6 IMPLEMENTATION OF FUNCTIONS USING MULIPLEXERS

Date:

AIM

To design and implement multiplexer using logic gates.

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
4.	IC Trainer Kit	-	1
5.	Connecting Wires	-	Few

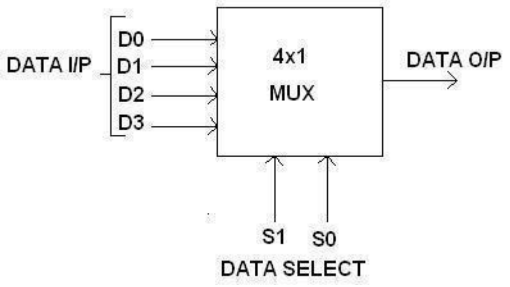
THEORY

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 2n input line and n selection lines whose bit combination determine which input is selected.

FUNCTION TABLE

BLOCK DIAGRAM OF MULIPLEXERS

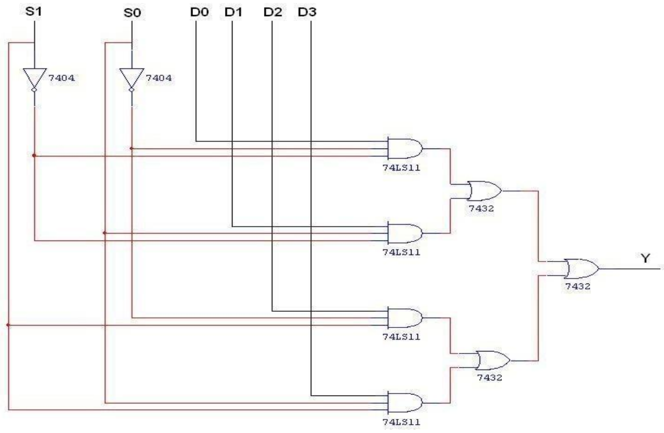


TRUTH TABLE:

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

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

CIRCUIT DIAGRAM FOR MULTIPLEXER



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

Thus the Multiplexer are designed and implemented using logic gates.

Ex. No. :7 **DESIGN AND IMPLEMENTATION OF 3-BIT**

Date **SYNCHRONOUS UP/DOWN COUNTER**

AIM

To design and implement 3 bit synchronous up/down counter.

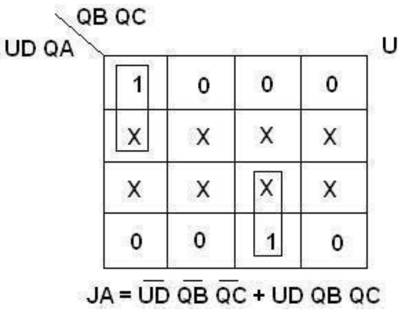
COMPONENTS REQUIRED

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	JK Flip Flop	IC 7476	2
2.	3 I/P AND Gate	IC 7411	1
3.	OR Gate	IC 7432	1
4.	XOR Gate	IC 7486	1
5.	NOT Gate	IC 7404	1
6.	IC Trainer Kit	-	1
7.	Connecting Wires	-	Few

THEORY

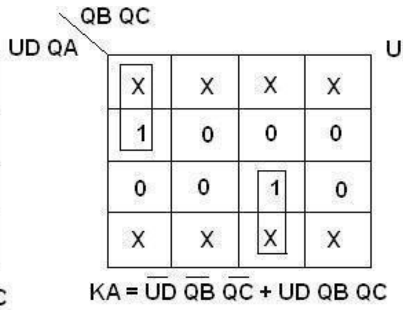
A counter is a register capable of counting number of clock pulse arriving at its clock input. Counter represents the number of clock pulses arrived. An up/down counter is one that is capable of progressing in increasing order or decreasing order through a certain sequence. An up/down counter is also called bidirectional counter. Usually up/down operation of the counter is controlled by up/down signal. When this signal is high counter goes through up sequence and when up/down signal is low counter follows reverse sequence.

K MAP



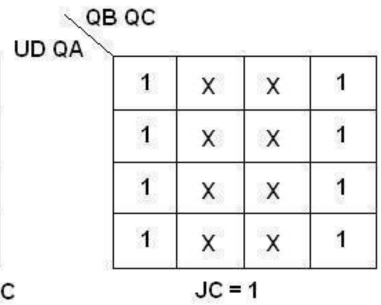
1	0	0	0
X	X	X	X
X	X	X	X
0	0	1	0

$JA = \overline{UD}\,\overline{QB}\,\overline{QC} + UD\ QB\ QC$



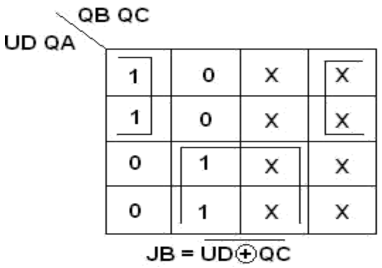
X	X	X	X
1	0	0	0
0	0	1	0
X	X	X	X

$KA = \overline{UD}\,\overline{QB}\,\overline{QC} + UD\ QB\ QC$



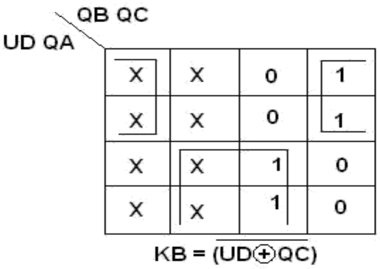
1	X	X	1
1	X	X	1
1	X	X	1
1	X	X	1

JC = 1



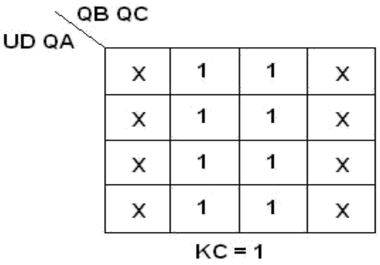
1	0	X	X
1	0	X	X
0	1	X	X
0	1	X	X

$JB = \overline{UD \oplus QC}$



X	X	0	1
X	X	0	1
X	X	1	0
X	X	1	0

$KB = (\overline{UD \oplus QC})$



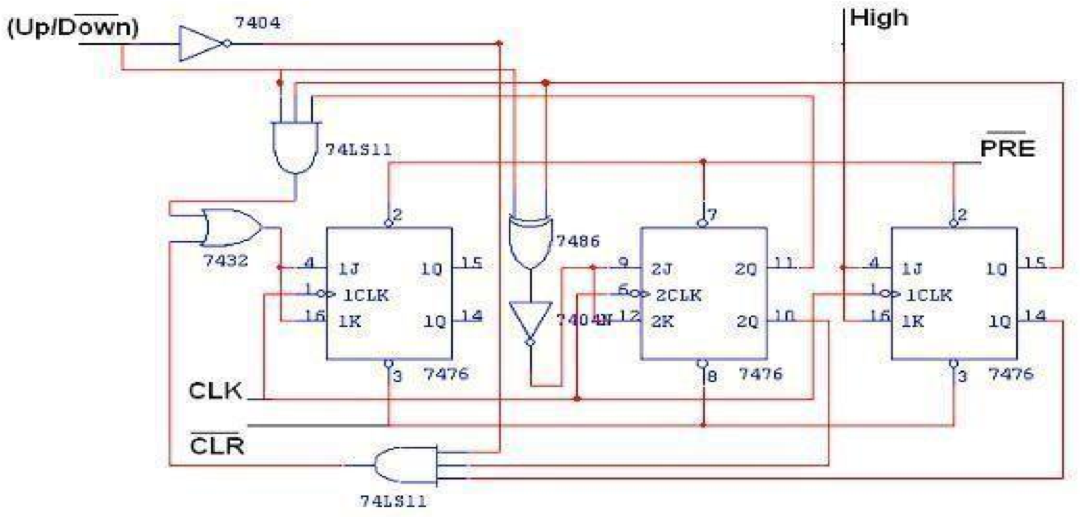
X	1	1	X
X	1	1	X
X	1	1	X
X	1	1	X

KC = 1

CHARACTERISTICS TABLE

Q	Qt+1	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

LOGIC DIAGRAM



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

Thus the 3-bit synchronous UP/DOWN counter is designed and its truth table is verified.

EX.NO:8 **DESIGN AND IMPLEMENTATION OF SHIFT REGISTERS**

DATE :

AIM

To design and implement

(i) Serial in serial out
(ii) Serial in parallel out
(iii) Parallel in serial out
(iv) Parallel in parallel out shift registers using Flip Flops.

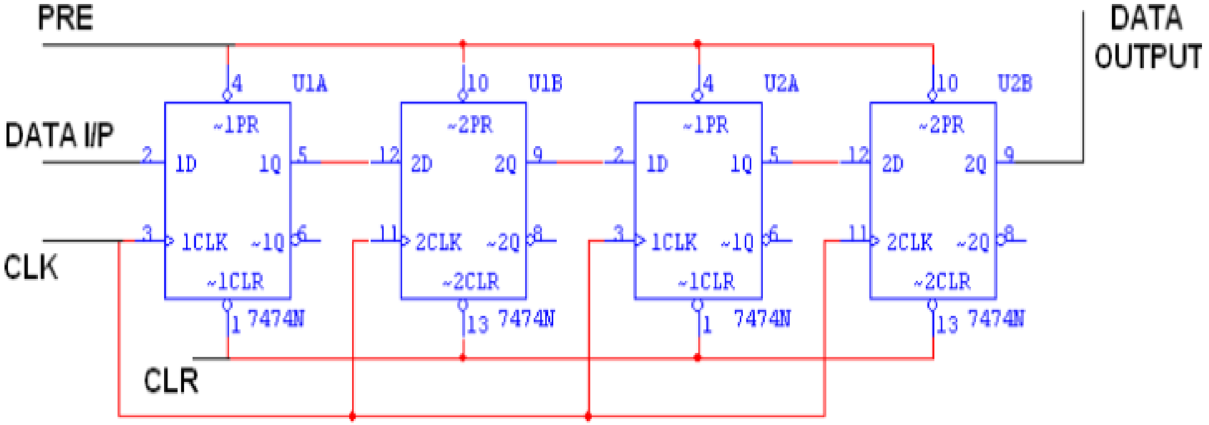
COMPONENTS REQUIRED

SL. No.	COMPONENT	SPECIFICATION	QUANTITY
1	D Flip Flop	IC 7474	2
2	OR Gate	IC 7432	1
3	IC Trainer Kit	-	1
4	Connecting Wires	-	Few

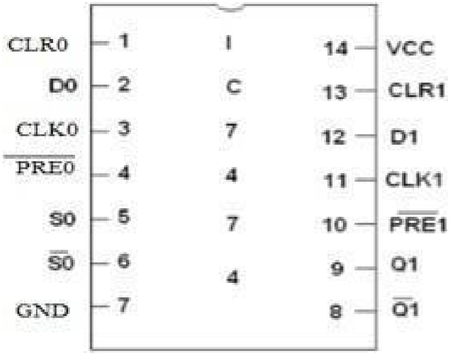
THEORY

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.

SERIAL IN SERIAL OUT



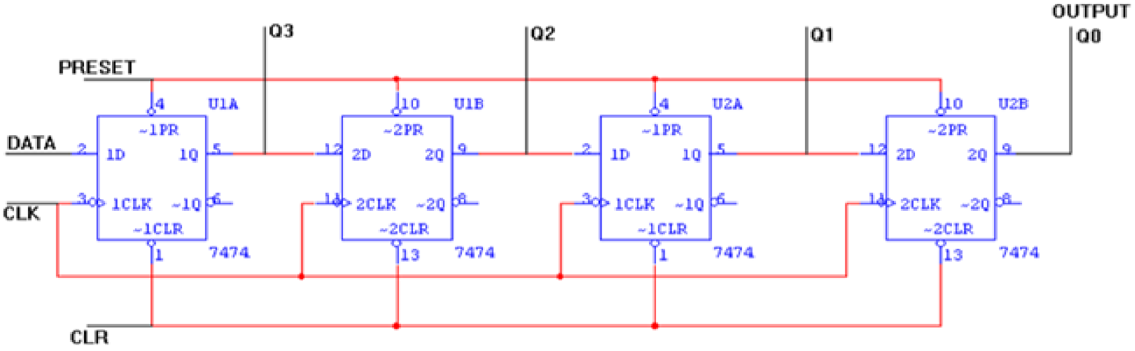
PIN DIAGRAM



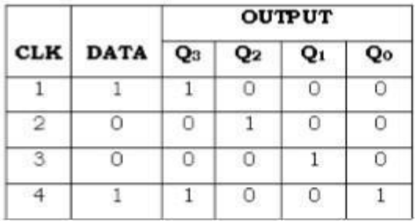
TRUTH TABLE

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

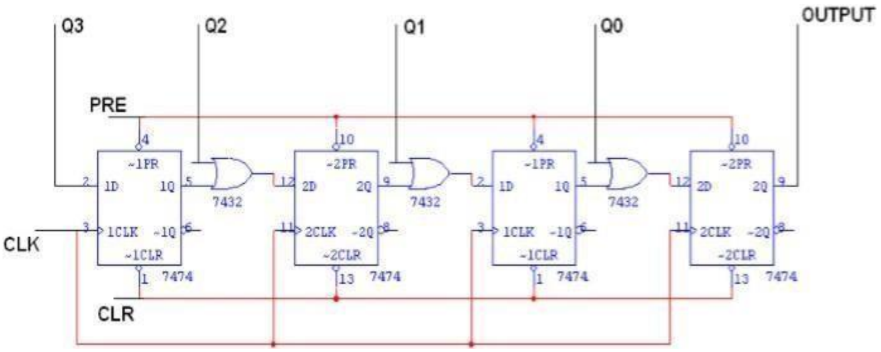


TRUTH TABLE



CLK	DATA	OUTPUT			
		Q_3	Q_2	Q_1	Q_0
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0
4	1	1	0	0	1

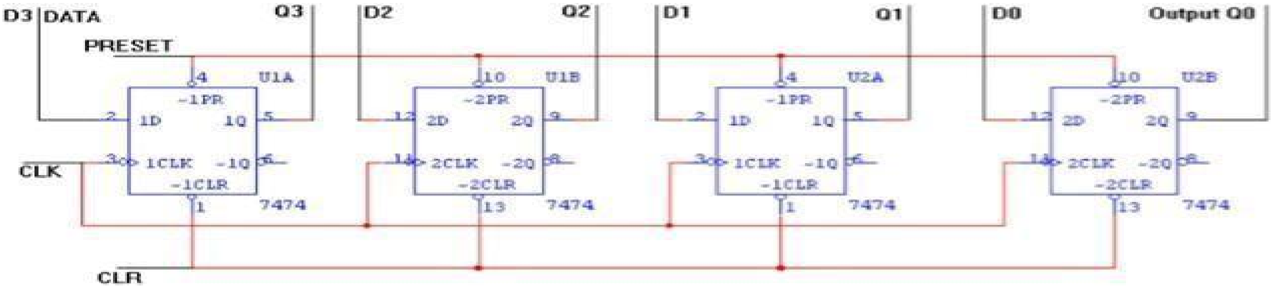
PARALLEL IN SERIAL OUT



TRUTH TABLE

CLK	Q3	Q2	Q1	Q0	OUPUT
1	1	0	0	1	1
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	1

PARALLEL IN PARALLEL OUT



TRUTH TABLE

CLK	DATA INPUT				OUTPUT			
	D3	D2	D1	D0	Q3	Q2	Q1	Q0
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

Thus the Serial in serial out, Serial in parallel out, Parallel in serial out and Parallel in parallel out Shift registers are constructed.