DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Lab Manual
Digital Electronics Laboratory
(EC-309)

BACHELOR OF TECHNOLOGY

Subject Code: EC 309

Subject Name: Digital Electronics Laboratory

Teaching Department: Electronics & Communication Engineering

Minimum No. of Experiments to be carried out: 12

List of Experiments:

1. Verification of Logic gates
2. Half Adder & Full Adder
3. Half Subtractor & Full Subtractor
4. Binary to Excess-3 Code & Excess-3 to Binary
5. 1 Bit Comparator
6. Parity Checker & Parity Generator
7. 4:1 Multiplexer
8. 1:4 Demultiplexer
9. BCD to 7 Segment Display
10. Binary to Gray Code & Gray to Binary
11. Conversion of flip flop.
12. Verification of flip flop.
13. Encoder and Decoder Using IC

Introduction

There are 2 hours 30 minutes to a laboratory session in Digital Electronics. It is a necessary part of the course at which attendance is compulsory.

Here are some guidelines to help you perform the experiments and to submit the reports:

1. Read all instructions carefully and carry them all out.
2. Ask a demonstrator if you are unsure of anything.
3. Record actual results (comment on them if they are unexpected!)
4. Write up full and suitable conclusions for each experiment.
5. If you have any doubt about the safety of any procedure, contact the demonstrator beforehand.

The Breadboard

The breadboard consists of two terminal strips and two bus strips (often broken in the centre). Each bus strip has two rows of contacts. Each of the two rows of contacts are a node. That is, each contact along a row on a bus strip is connected together (inside the breadboard). Bus strips are used primarily for power supply connections, but are also used for any node requiring a large number of connections. Each terminal strip has 60 rows and 5 columns of contacts on each side of the centre gap. Each row of 5 contacts is a node.

You will build your circuits on the terminal strips by inserting the leads of circuit components into the contact receptacles and making connections with 22-26 gauge wire. There are wire cutter/strippers and a spool of wire in the lab. It is a good practice to wire +5V and 0V power supply connections to separate bus strips.

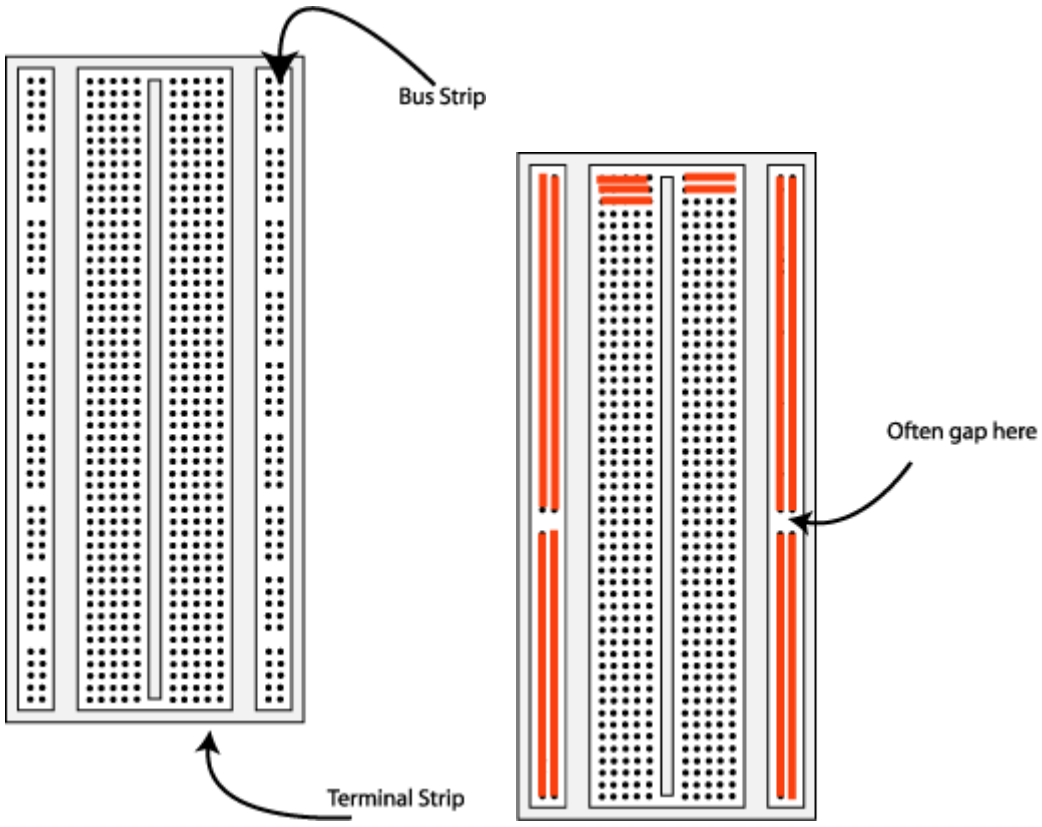


Fig 1. The breadboard. The lines indicate connected holes.

The 5V supply **MUST NOT BE EXCEEDED** since this will damage the ICs (Integrated circuits) used during the experiments. Incorrect connection of power to the ICs could result in them exploding or becoming very hot - with the **possible serious injury occurring to the people working on the experiment! Ensure that the power supply polarity and all components and connections are correct before switching on power .**

Building the Circuit

Throughout these experiments we will use TTL chips to build circuits. The steps for wiring a circuit should be completed in the order described below:

1. Turn the power (Trainer Kit) off before you build anything!
2. Make sure the power is off before you build anything!
3. Connect the +5V and ground (GND) leads of the power supply to the power and ground bus strips on your breadboard.
4. Plug the chips you will be using into the breadboard. Point all the chips in the same direction with pin 1 at the upper-left corner. (Pin 1 is often identified by a dot or a notch next to it on the chip package)
5. Connect +5V and GND pins of each chip to the power and ground bus strips on the breadboard.
6. Select a connection on your schematic and place a piece of hook-up wire between corresponding pins of the chips on your breadboard. It is better to make the short connections before the longer ones. Mark each connection on your schematic as you go, so as not to try to make the same connection again at a later stage.
7. Get one of your group members to check the connections, **before you turn the power on**.
8. If an error is made and is not spotted before you turn the power on. Turn the power off immediately before you begin to rewire the circuit.
9. At the end of the laboratory session, collect you hook-up wires, chips and all equipment and return them to the demonstrator.
10. Tidy the area that you were working in and leave it in the same condition as it was before you started.

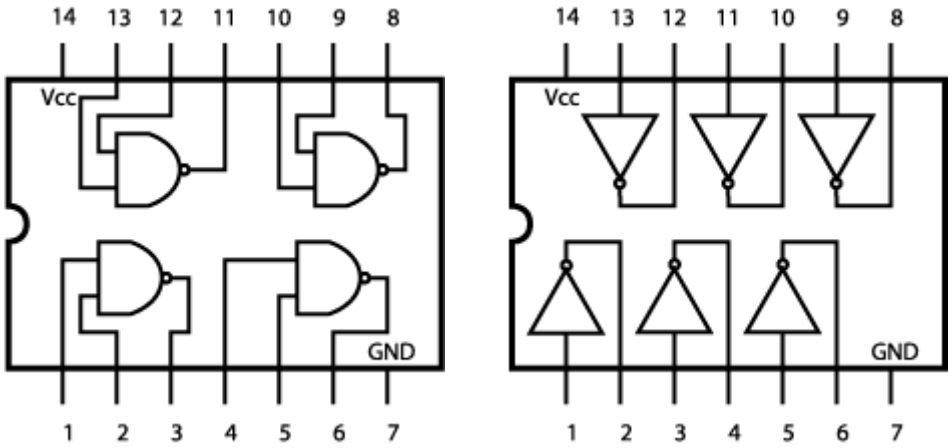
Common Causes of Problems

1. Not connecting the ground and/or power pins for all chips.
2. Not turning on the power supply before checking the operation of the circuit.
3. Leaving out wires.
4. Plugging wires into the wrong holes.
5. Driving a single gate input with the outputs of two or more gates
6. Modifying the circuit with the power on.

In all experiments, you will be expected to obtain all instruments, leads, components at the start of the experiment and return them to their proper place after you have finished the experiment. Please inform the demonstrator or technician if you locate faulty equipment. If you damage a chip, inform a demonstrator, don't put it back in the box of chips for somebody else to use.

Example Implementation of a Logic Circuit

Build a circuit to implement the Boolean function F = /(/A./B), please note that the notation **/A** refers to $\overline{A}$. You should use that notation during the write-up of your laboratory experiments.



Quad 2 Input 7400 Hex 7404 Inverter

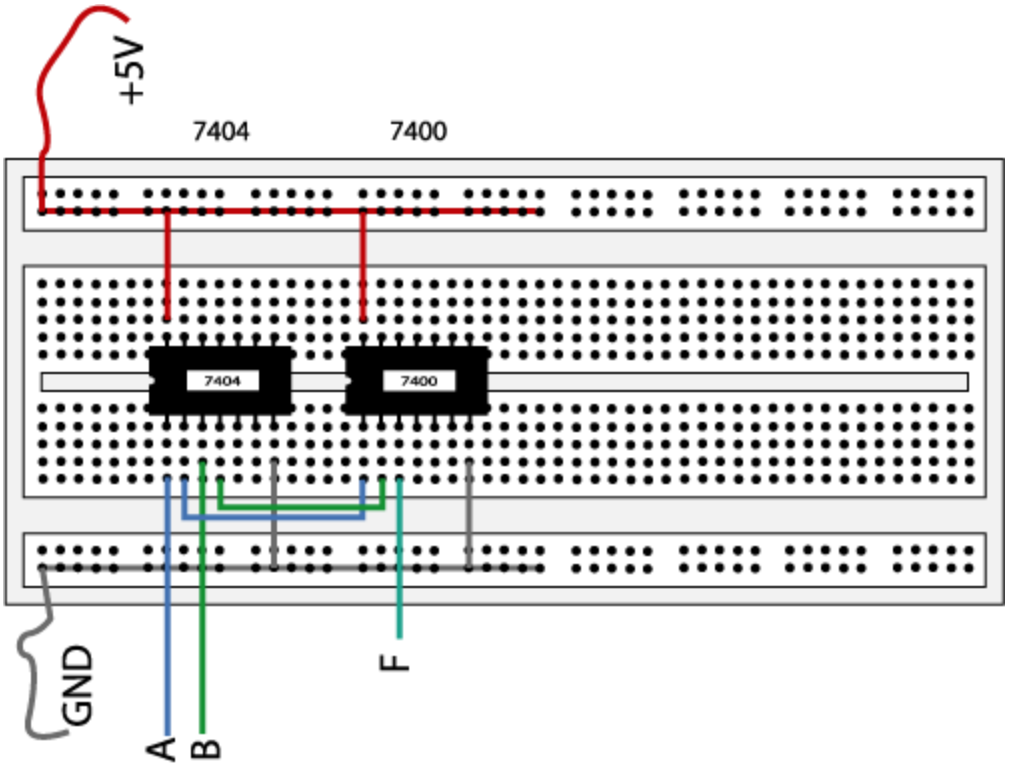


Fig 2. The complete designed and connected circuit

Sometimes the chip manufacturer may denote the first pin by a small indented circle above the first pin of the chip. Place your chips in the same direction, to save confusion at a later stage. Remember that you must connect power to the chips to get them to work.

EXPERIMENT NUMBER: 0

EXPERIMENT NAME: VERIFICATION OF LOGIC GATES

AIM: To study about logic gates and verify their truth tables.

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	IC 7400	1
5.	NOR GATE	IC 7402	1
6.	X-OR GATE	IC 7486	1
8.	IC TRAINER KIT	-	1
9.	WIRES	-	AS REQUIRED

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.

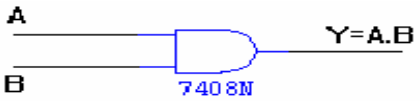
NAND GATE:
The NAND gate is a contraction of AND-NOT. The output is high when both inputs are low and any one of the input is low .The output is low level when both inputs are high.

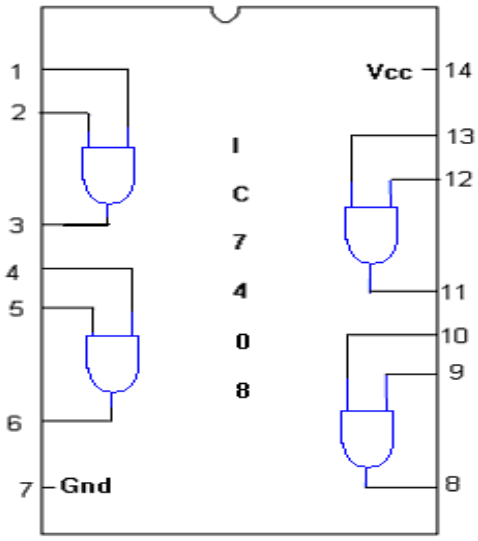
NOR GATE:
The NOR gate is a contraction of OR-NOT. The output is high when both inputs are low. The output is low when one or both inputs are high.

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.

AND GATE:

SYMBOL:



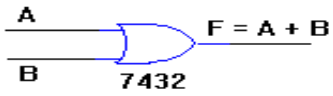


TRUTH TABLE

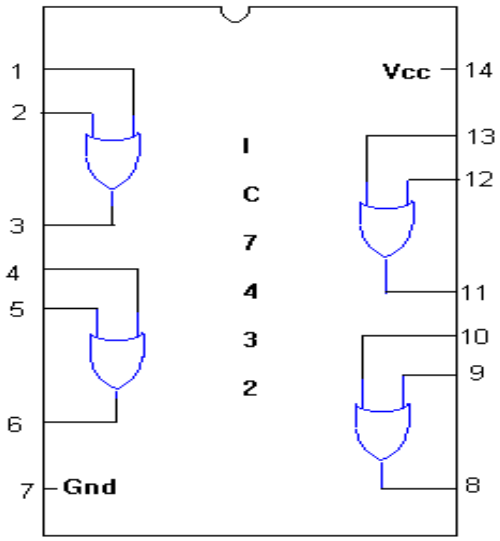
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

OR GATE:

SYMBOL :



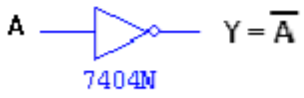
PIN DIAGRAM :

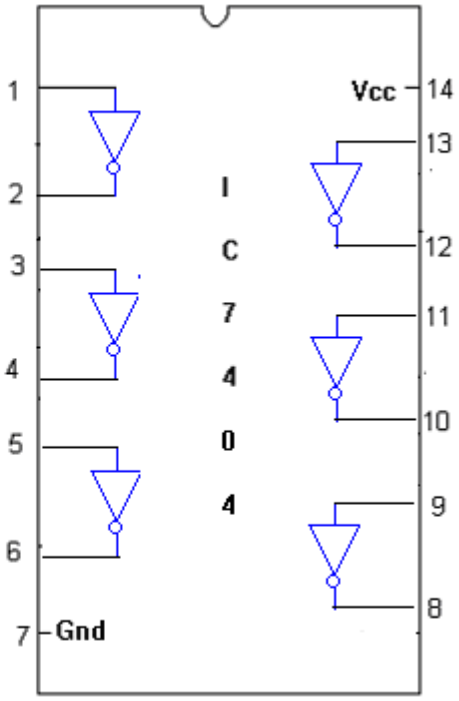


TRUTH TABLE

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

NOT GATE:

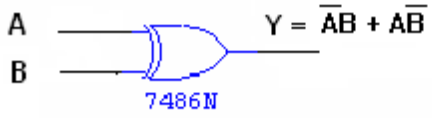




TRUTH TABLE :

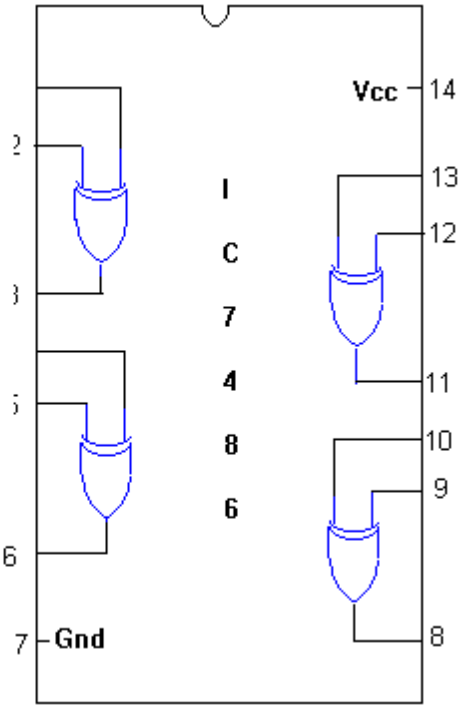
A	$\overline{A}$
0	1
1	0

X-OR GATE :



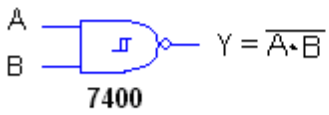
TRUTH TABLE :

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

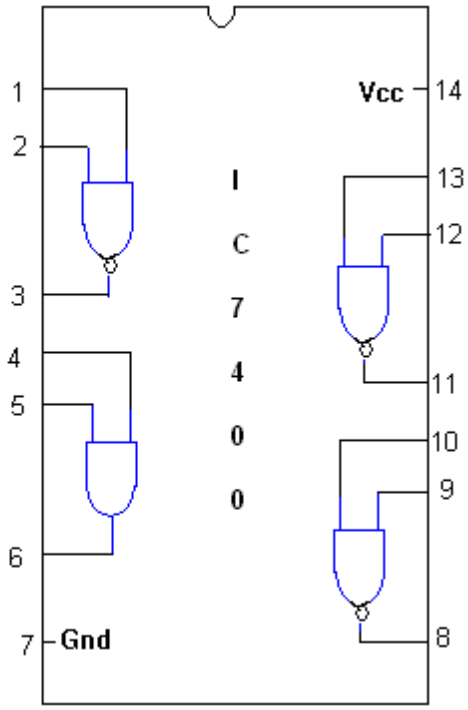


2-INPUT NAND GATE:

SYMBOL:



PIN DIAGRAM:

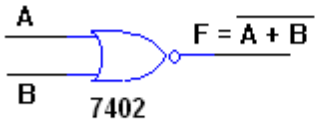


TRUTH TABLE

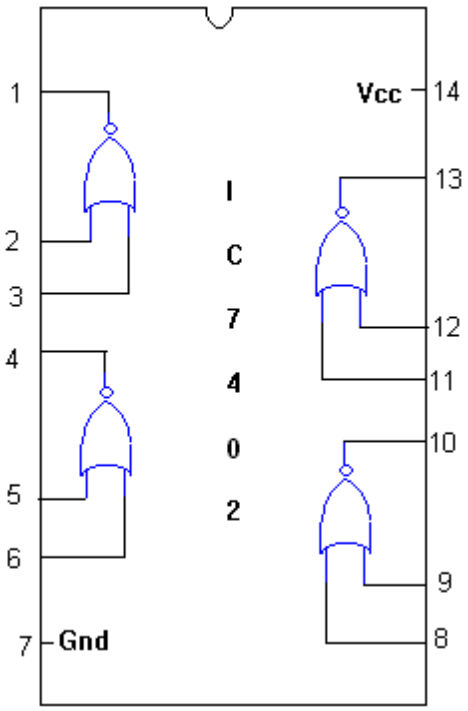
A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

NOR GATE:

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 the logic gates are studied and their truth tables were verified.

EXPERIMENT NUMBER: 1

EXPERIMENT NAME: Design a (i) Half adder and (ii) Full adder

AIM: To design a Half Adder and Full Adder circuit using logic gates and verify it using truth table.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	1
2.	X-OR GATE	IC 7486	1
3.	NOT GATE	IC 7404	1
4.	OR GATE	IC 7432	1
5.	IC TRAINER KIT	-	1
6.	CONNECTING WIRES	-	AS REQUIRED

THEORY:

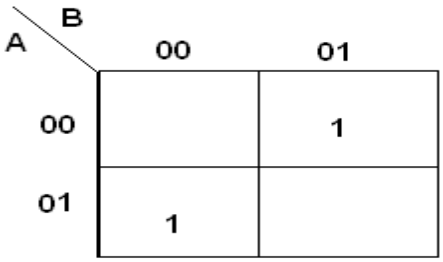
HALF ADDER:
A half adder has two inputs for the two bits to be added and two outputs one from the sum ' S' and other from the carry ' c' into the higher adder position. The circuit is called as a carry signal from the addition of the less significant bits sum from the X-OR Gate the carry out from the AND gate.

FULL ADDER:
A full adder is a combinational circuit that forms the arithmetic sum of input; it consists of three inputs and two outputs. A full adder is useful to add three bits at a time but a half adder cannot do so. In full adder sum output will be taken from X-OR Gate, carry output will be taken from OR Gate.

TRUTH TABLE OF HALF ADDER:

A	B	CARRY	SUM
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

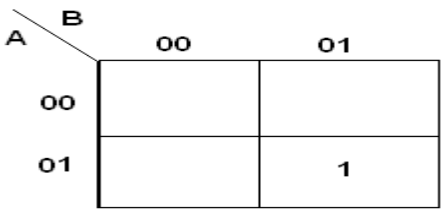
K-Map for SUM:



SUM = A'B + AB'

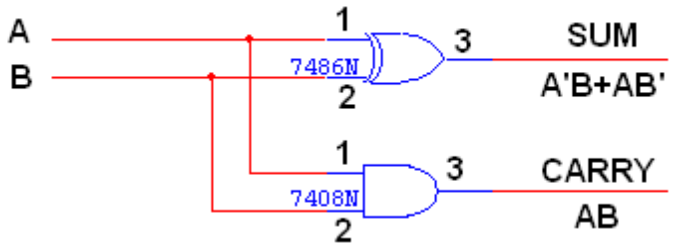
= A B ⊕

K-Map for CARRY:



CARRY = AB

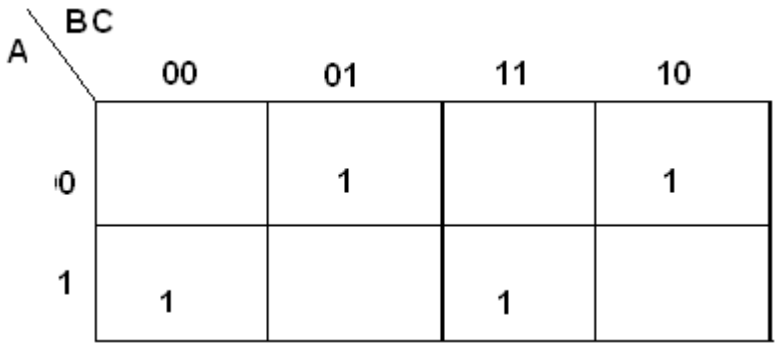
CIRCUIT DIAGRAM:



TRUTH TABLE OF FULL ADDER:

A	B	C	CARRY	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

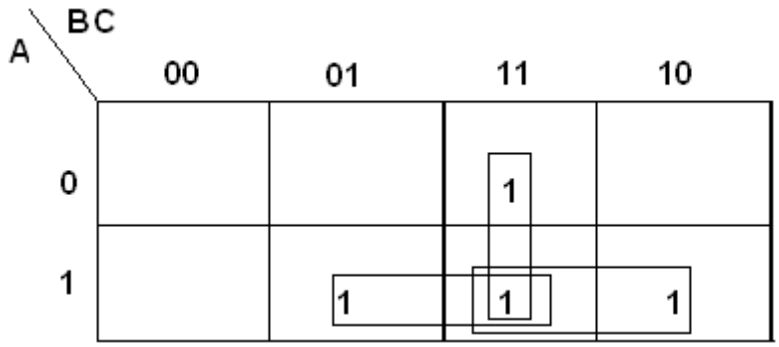
K-Map for SUM:



SUM = A'B'C + A'BC' + ABC' + ABC

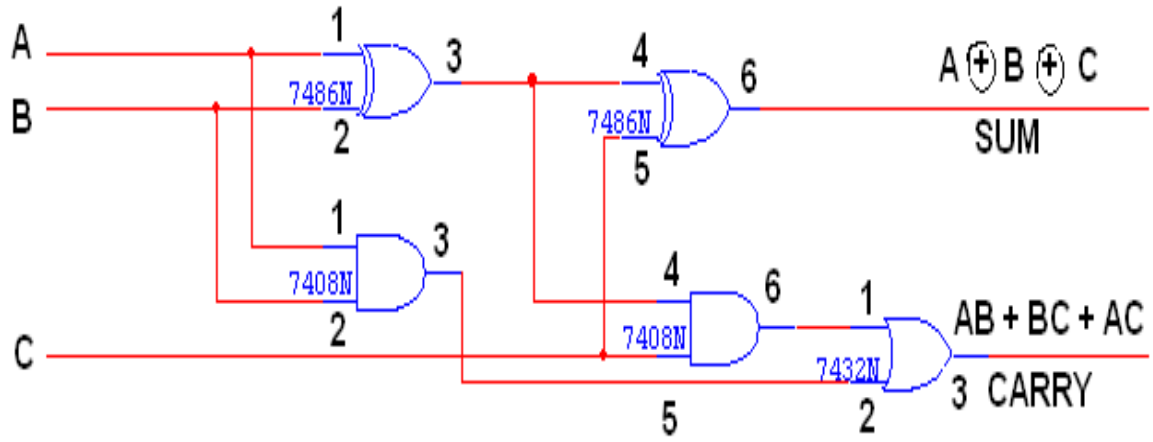
= A ⊕ B ⊕ C

K-Map for CARRY:



CARRY = AB + BC + AC

CIRCUIT 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 half adder, full adder was designed and their truth table is verified.

EXPERIMENT NUMBER: 2

EXPERIMENT NAME: Design a (i) Half subtractor and (ii) Full subtractor

AIM: To design a Half subtractor and Full subtractor circuit using logic gates and verify it using truth table.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	1
2.	X-OR GATE	IC 7486	1
3.	NOT GATE	IC 7404	1
4.	OR GATE	IC 7432	1
5.	IC TRAINER KIT	-	1
6.	CONNECTING WIRES	-	AS REQUIRED

THEORY:

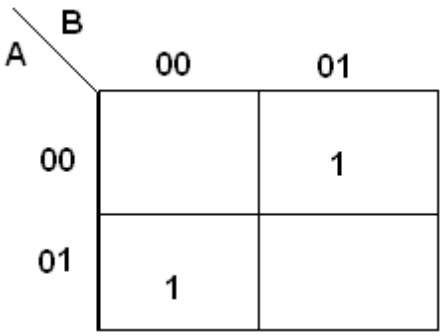
HALF SUBTRACTOR:
The half subtractor is constructed using X-OR and AND Gate. The half subtractor has two input and two outputs. The outputs are difference and borrow. The difference can be applied using X-OR Gate, borrow output can be implemented using an AND Gate and an inverter.

FULL SUBTRACTOR:
The full subtractor is a combination of X-OR, AND, OR, NOT Gates. In a full subtractor the logic circuit should have three inputs and two outputs. The two half subtractor put together gives a full subtractor .The first half subtractor will be C and A B. The output will be difference output of full subtractor. The expression AB assembles the borrow output of the half subtractor and the second term is the inverted difference output of first X-OR.

TRUTH TABLE OF HALF SUBTRACTOR:

A	B	BORROW	DIFFERENCE
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

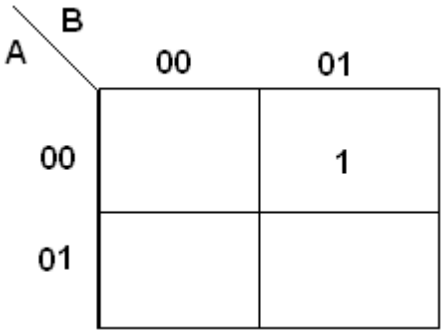
K-Map for DIFFERENCE:



DIFFERENCE = A'B + AB'

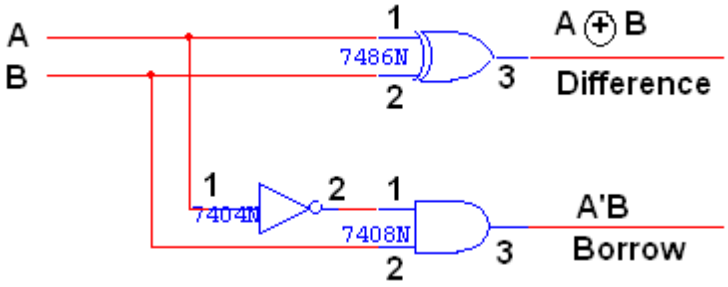
= A ⊕ B

K-Map for BORROW:



BORROW = A'B

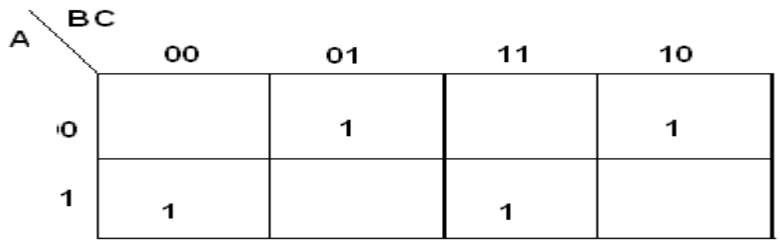
CIRCUIT DIAGRAM:



TRUTH TABLE OF FULL SUBTRACTOR:

A	B	C	BORROW	DIFFERENCE
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

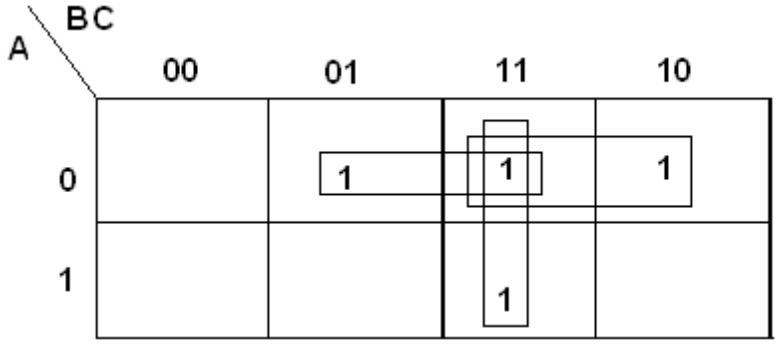
K-Map for Difference:



Difference = A'B'C + A'BC' + AB'C' + ABC

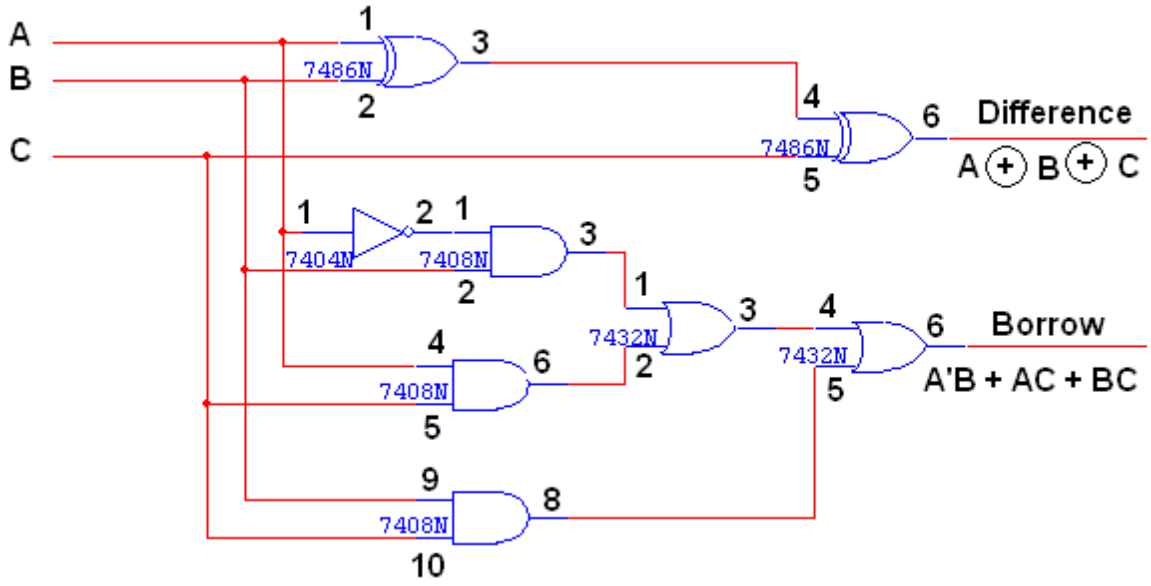
= A ⊕ B ⊕

K-Map for Borrow:



Borrow = A'B + BC + A'C

CIRCUIT 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 half subtractor and full subtractor was designed and their truth table is verified.

EXPERIMENT NUMBER: 3

EXPERIMENT NAME: Design a 1-bit Comparator

AIM: To design a 1-bit Comparator using logic gates.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	1
2.	X-OR GATE	IC 7486	1
3.	NOT GATE	IC 7404	1
4.	IC TRAINER KIT	-	1
5.	CONNECTING WIRES	-	AS REQUIRED

THEORY:
1-Bit Comparator is a logical circuit, which compares two signals A and B and generates three logical outputs, whether A > B, A < B or A = B.

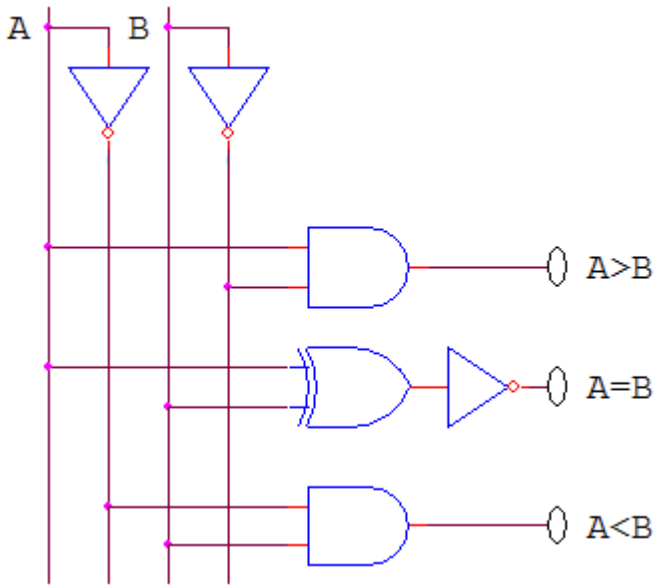
TRUTH TABLE:

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

From the K- Maps, we get:-

A>B = AB’
A=B = A’B’+AB
A<B = A’B

CIRCUIT 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 1-bit Comparator was designed and their truth table is verified.

EXPERIMENT NUMBER: 4

EXPERIMENT NAME: Design a 3-BIT Odd Parity Generator.

AIM: To design a 3-bit Odd Parity Generator using logic gates.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	X-OR GATE	IC 7486	1
2.	NOT GATE	IC 7404	1
3.	IC TRAINER KIT	-	1
4.	CONNECTING WIRES	-	AS REQUIRED

THEORY:

A parity bit is used for the purpose of detecting errors during transmission of binary information. A parity bit is an extra bit included with a binary message to make the number of 1's either odd or even. The message including the parity bit is transmitted and then checked at the receiving end for errors.

An error is detected if the checked parity does not correspond with the one transmitted. The circuit that generates the parity bit in the transmitter is called a parity generator and the circuit that checks the parity in the receiver is called a parity checker. In even parity the added parity bit will make the total number of 1's an even amount and in odd parity the added parity bit will make the total number of 1's an odd amount. In a three bit odd parity generator the three bits in the message together with the parity bit are transmitted to their destination, where they are applied to the parity checker circuit. The parity checker circuit checks for possible errors in the transmission. Since the information was transmitted with odd parity the four bits received must have an odd number of 1's. An error occurs during the transmission if the four bits received have an even number of 1's, indicating that one bit has changed during transmission. The output of the parity checker is denoted by PEC (parity error check) and it will be equal to 1 if an error occurs, i.e., if the four bits received has an even number of 1's.

ODD PARITY GENERATOR

TRUTH TABLE:

A	B	C	P
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

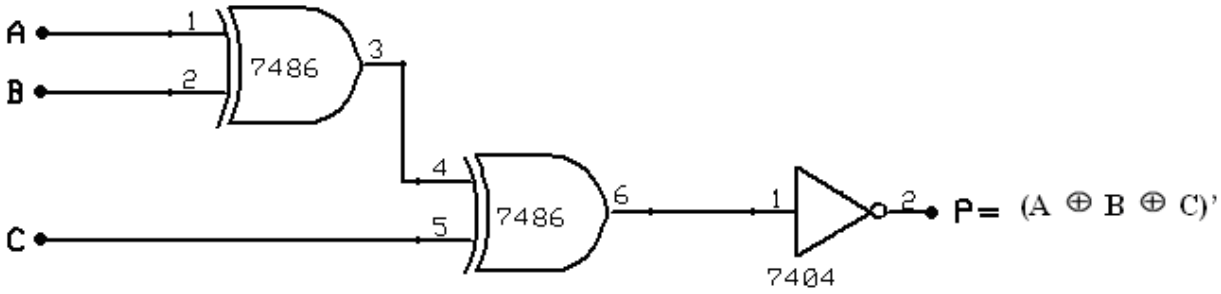
From the truth table the expression for the output parity bit is,

P(A, B, C) = Σ (0, 3, 5, 6)

Also written as,

P = A'B'C' + A'BC + AB'C + ABC' = (A $\oplus$ B $\oplus$ C)'

CIRCUIT 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 Odd Parity Generator was designed and their truth table is verified.

EXPERIMENT NUMBER: 5

EXPERIMENT NAME: Design a 3-BIT Odd Parity Checker.

AIM: To design a 3-bit Odd Parity Checker using logic gates.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	X-OR GATE	IC 7486	1
2.	NOT GATE	IC 7404	1
3.	IC TRAINER KIT	-	1
4.	CONNECTING WIRES	-	AS REQUIRED

THEORY:

A parity bit is used for the purpose of detecting errors during transmission of binary information. A parity bit is an extra bit included with a binary message to make the number of 1's either odd or even. The message including the parity bit is transmitted and then checked at the receiving end for errors.

An error is detected if the checked parity does not correspond with the one transmitted. The circuit that generates the parity bit in the transmitter is called a parity generator and the circuit that checks the parity in the receiver is called a parity checker. In even parity the added parity bit will make the total number of 1's an even amount and in odd parity the added parity bit will make the total number of 1's an odd amount. In a three bit odd parity generator the three bits in the message together with the parity bit are transmitted to their destination, where they are applied to the parity checker circuit. The parity checker circuit checks for possible errors in the transmission. Since the information was transmitted with odd parity the four bits received must have an odd number of 1's. An error occurs during the transmission if the four bits received have an even number of 1's, indicating that one bit has changed during transmission. The output of the parity checker is denoted by PEC (parity error check) and it will be equal to 1 if an error occurs, i.e., if the four bits received has an even number of 1's.

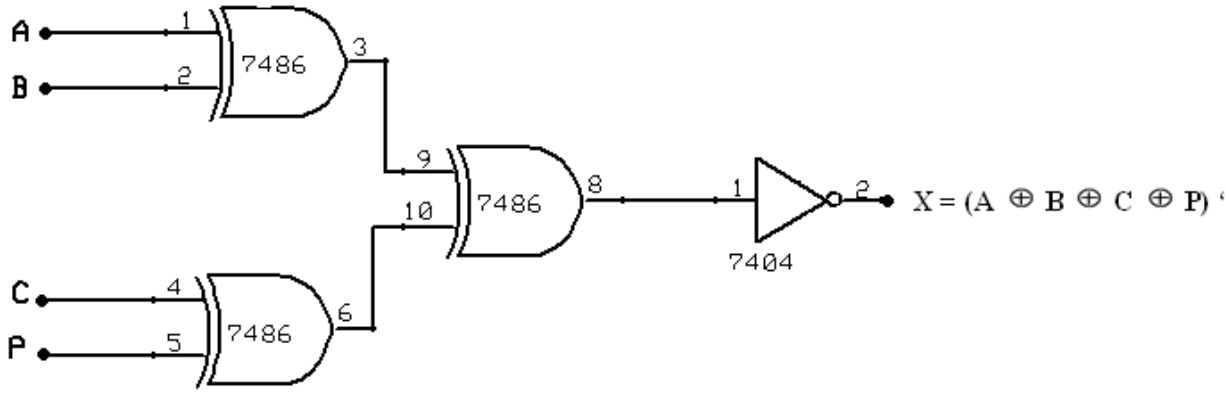
TRUTH TABLE:

A	B	C	P	X
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

From the truth table the expression for the output parity checker bit is,
X (A, B, C, P) = Σ (0, 3, 5, 6, 9, 10, 12, 15)

The above expression is reduced as,
X = (A ⊕ B ⊕ C ⊕ P)

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 Odd Parity Checker was designed and their truth table is verified.

EXPERIMENT NUMBER: 6

EXPERIMENT NAME: Design a 4-Bit Binary to Gray Code Converter.

AIM: To design a 4-Bit Binary to Gray Code converter using logic gates.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	X-OR GATE	IC 7486	1
2.	IC TRAINER KIT	-	1
3.	CONNECTING WIRES	-	AS REQUIRED

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.

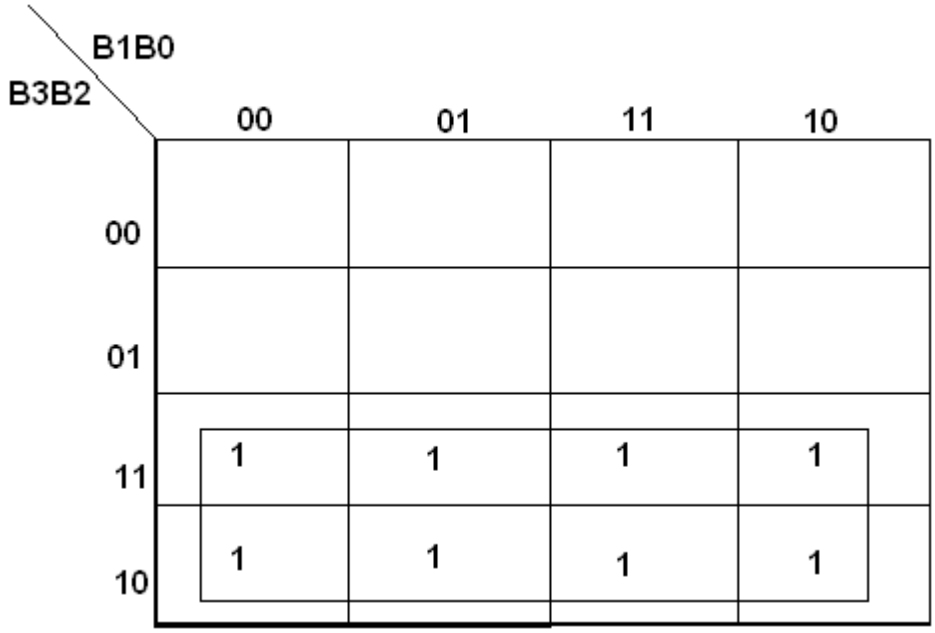
The input variable are designated as B3, B2, B1, B0 and the output variables are designated as C3, C2, C1, Co. from the truth table, combinational circuit is designed. The Boolean functions are obtained from K-Map for each output variable.

TRUTH TABLE:

BINARY INPUT **GRAY OUTPUT**

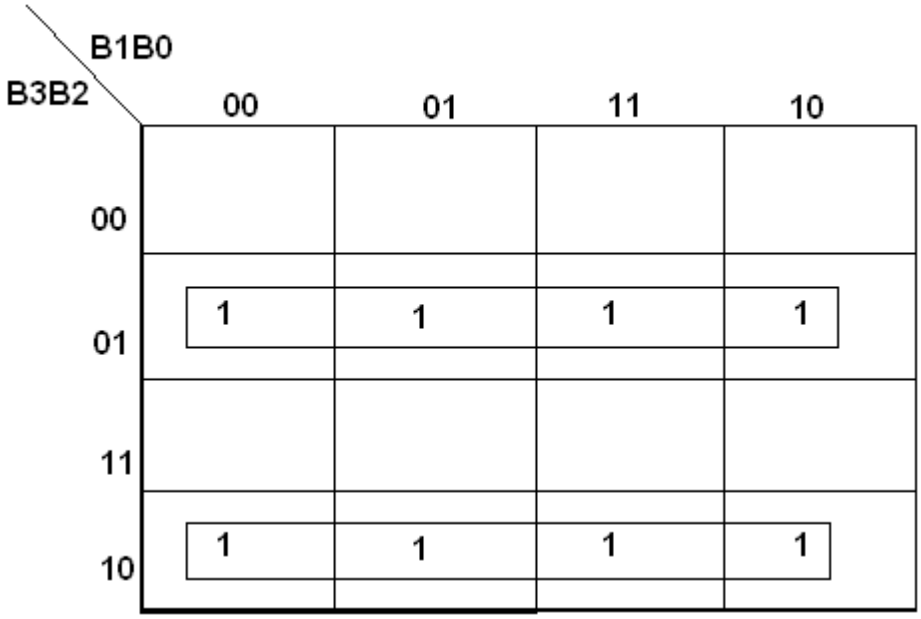
B3	B2	B1	B0	G3	G2	G1	G0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

K-Map for G_3:



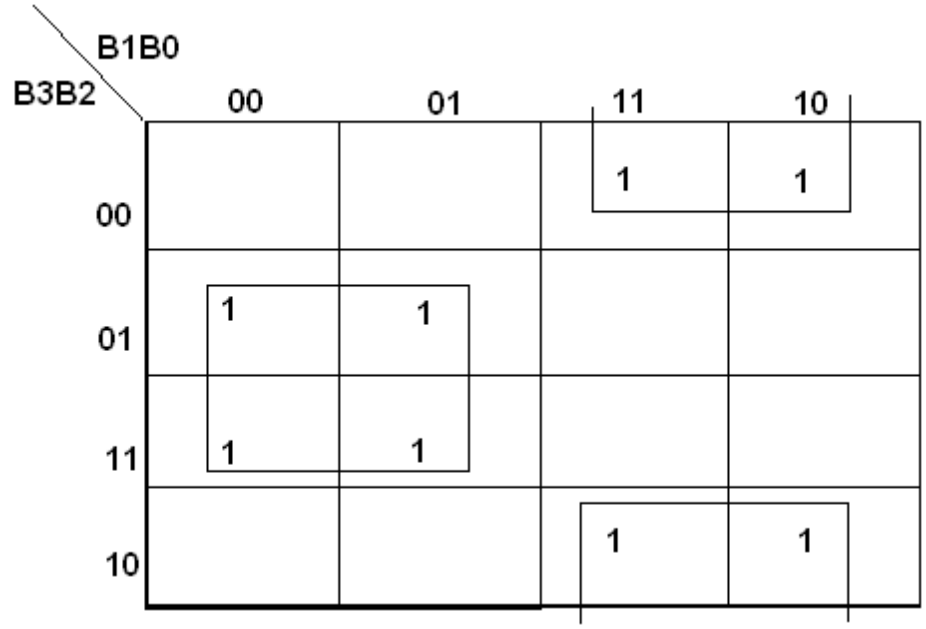
$G_3 = B_3$

K-Map for G_2:



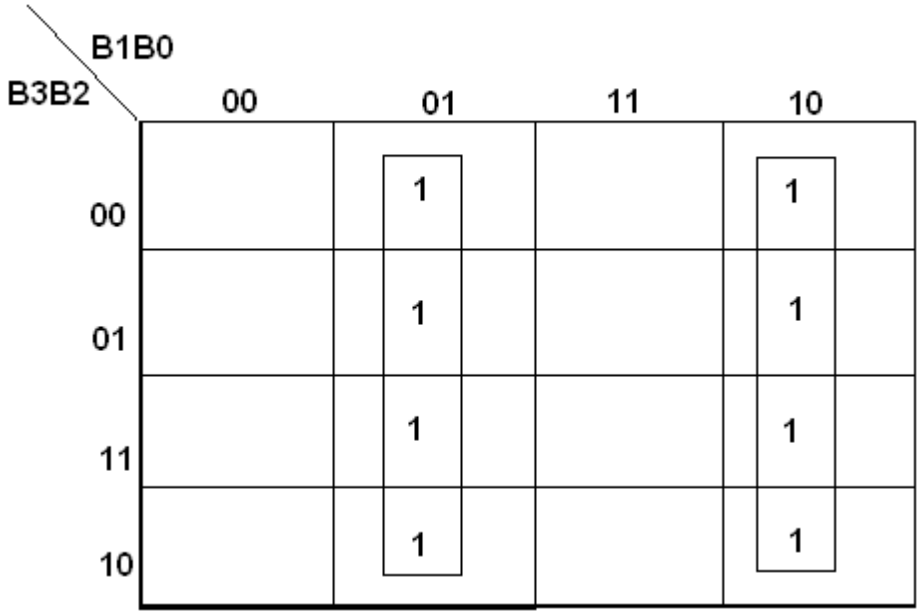
$G2 = B3 \oplus B2$

K-Map for G_1:



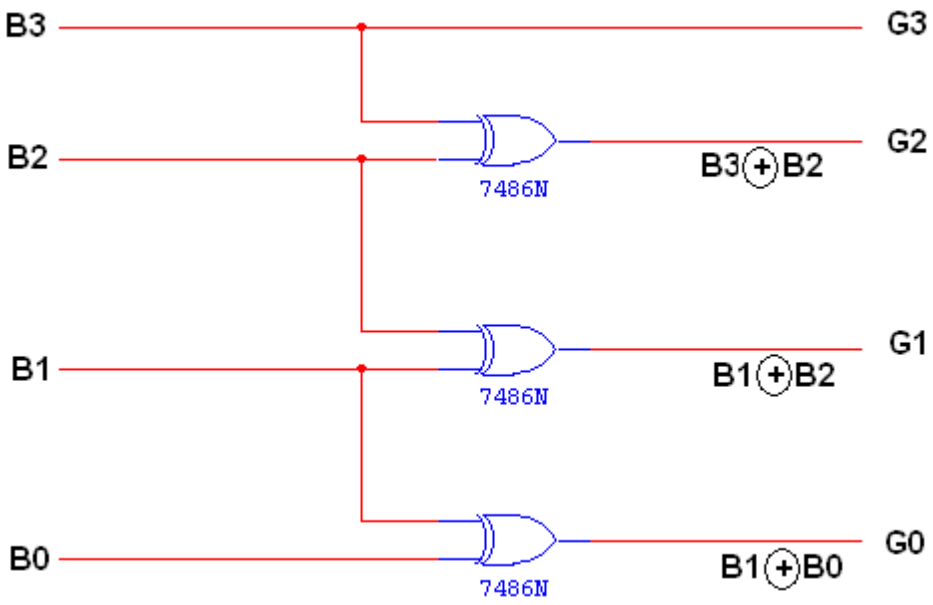
$G1 = B1 \oplus B2$

K-Map for G_0:



$G0 = B1 \oplus B0$

CIRCUIT 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 4-Bit Binary to Gray Code Converter was designed and their truth table is verified.

EXPERIMENT NUMBER: 6

EXPERIMENT NAME: Design a 4-Bit Gray to Binary Code Converter.

AIM: To design a 4-Bit Binary to Gray Code converter using logic gates.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	X-OR GATE	IC 7486	1
2.	IC TRAINER KIT	-	1
3.	CONNECTING WIRES	-	AS REQUIRED

THEORY:

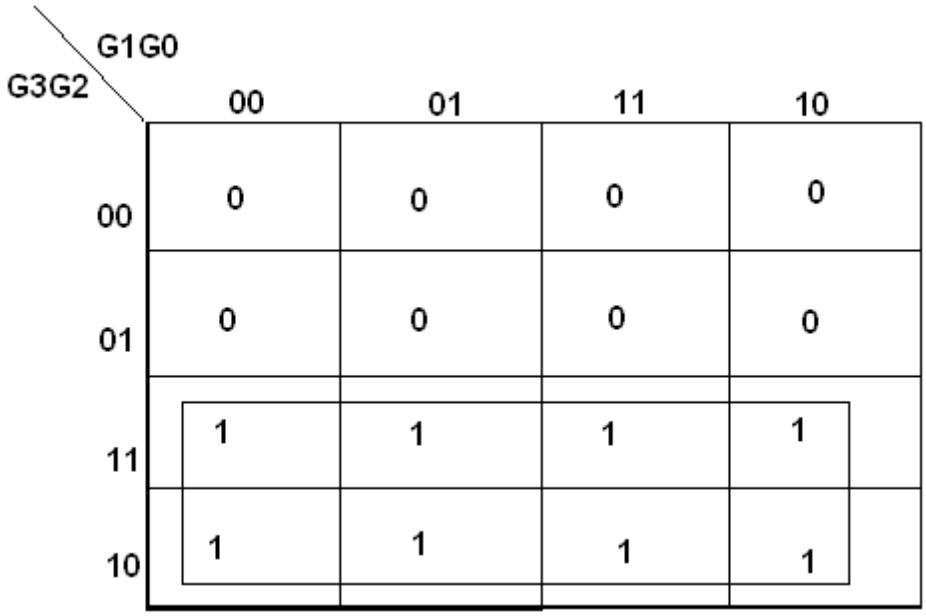
The availability of large variety of codes for the same discrete elements of information results in the use of different codes by different systems. A conversion circuit must be inserted between the two systems if each uses different codes for same information. Thus, code converter is a circuit that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

TRUTH TABLE:

GRAY INPUT **BINARY OUTPUT**

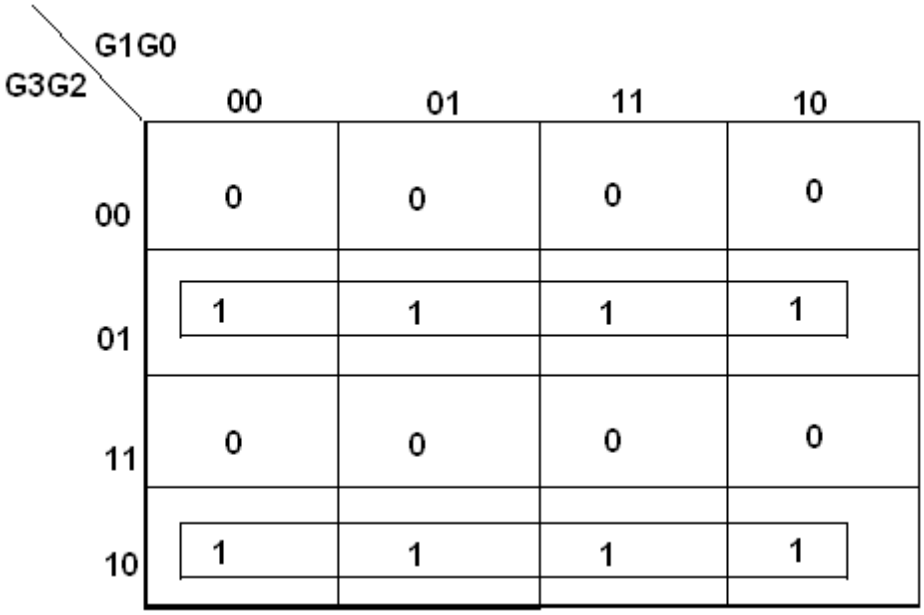
G3	G2	G1	G0	B3	B2	B1	B0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	0	1	0	0	0	1	1
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0
0	1	0	0	0	1	1	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	1	1	0	1	0
1	1	1	0	1	0	1	1
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	0	0	1	1	1	1	0
1	0	0	0	1	1	1	1

K-Map for B_3:



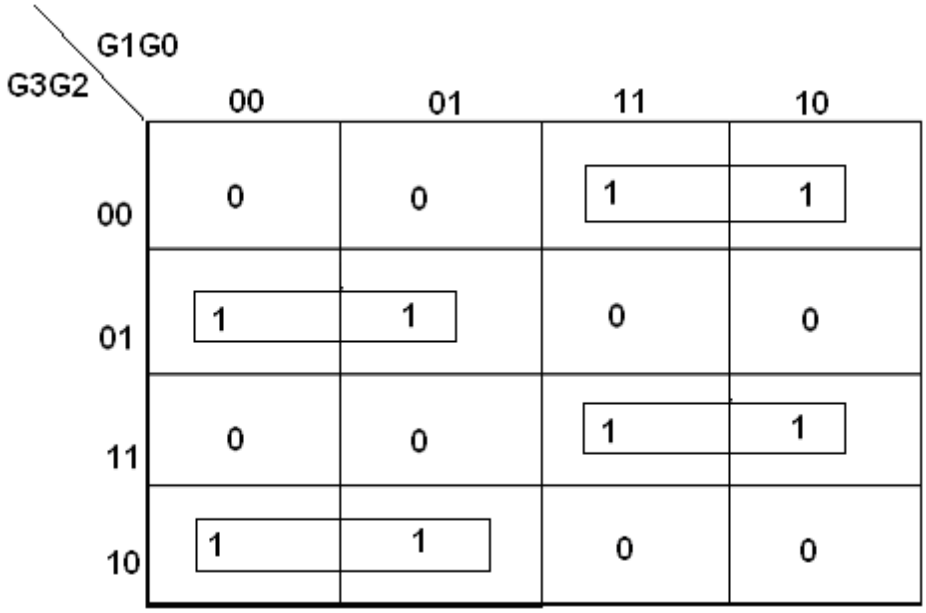
B3 = G3

K-Map for B_2:



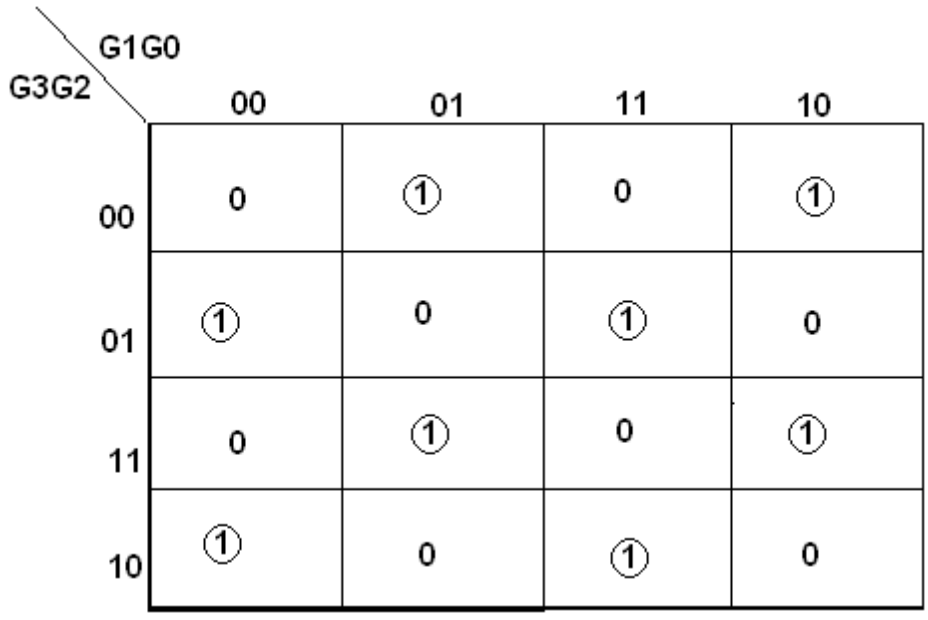
$B2 = G3 \oplus G2$

K-Map for B_1:



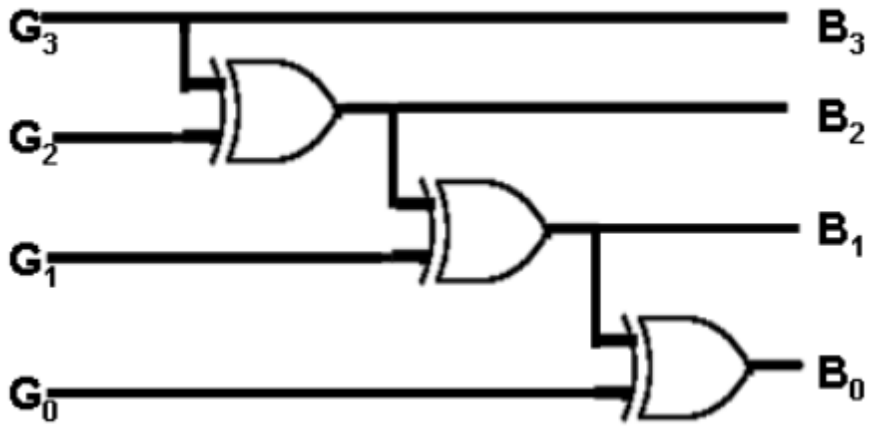
$B1 = G3 \oplus G2 \oplus G1$

K-Map for B_0:



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

CIRCUIT 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 4-Bit Gray to Binary Code Converter was designed and their truth table is verified.

EXPERIMENT NUMBER: 7

EXPERIMENT NAME: Design a BCD to Excess-3 Code Converter.

AIM: To design a BCD to Excess-3 Code Converter using logic gates.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	1
2.	X-OR GATE	IC 7486	1
3.	NOT GATE	IC 7404	1
4.	OR GATE	IC 7432	1
5.	IC TRAINER KIT	-	1
6.	CONNECTING WIRES	-	AS REQUIRED

THEORY:

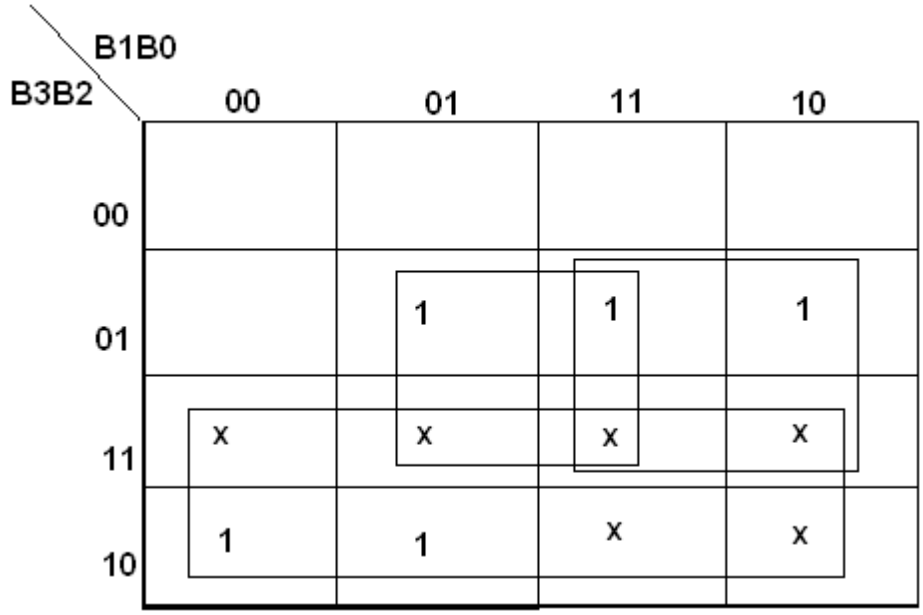
A code converter is a circuit that makes the two systems compatible even though each uses a different binary code. To convert from binary code to Excess-3 code, the input lines must supply the bit combination of elements as specified by code and the output lines generate the corresponding bit combination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

TRUTH TABLE:

BINARY INPUT **EXCESS-3 OUTPUT**

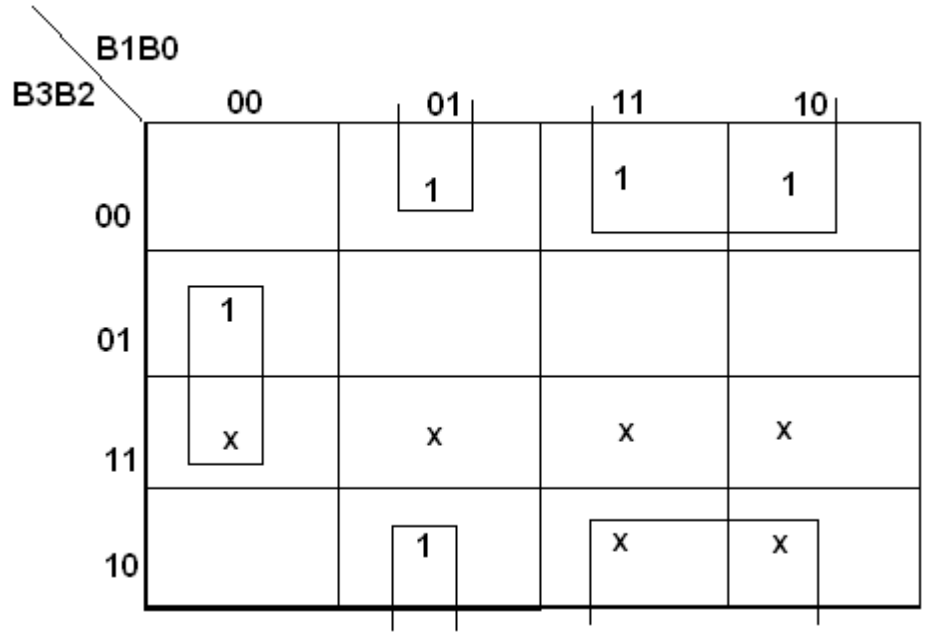
B3	B2	B1	B0	E3	E2	E1	E0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

K-Map for E_3:



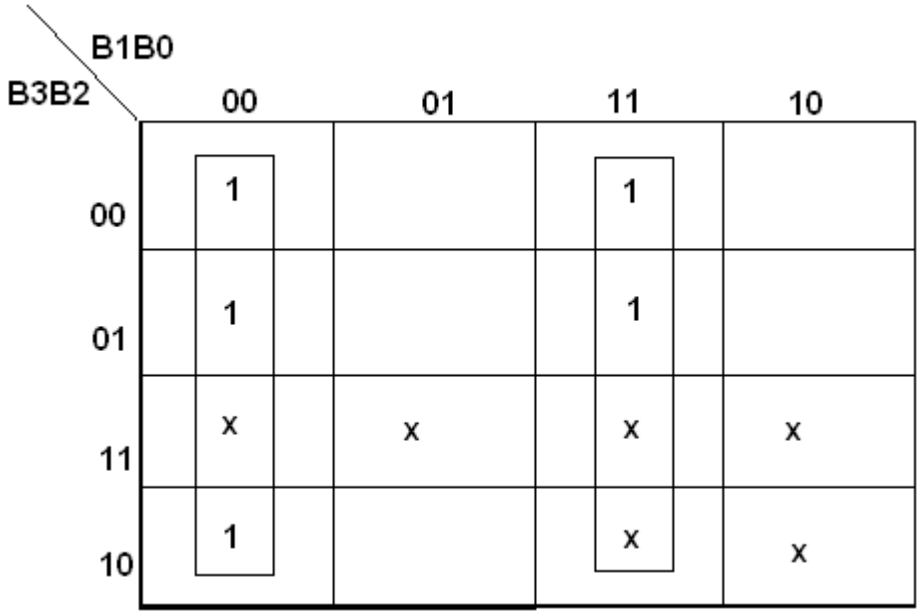
E3 = B3 + B2 (B0 + B1)

K-Map for E_2:



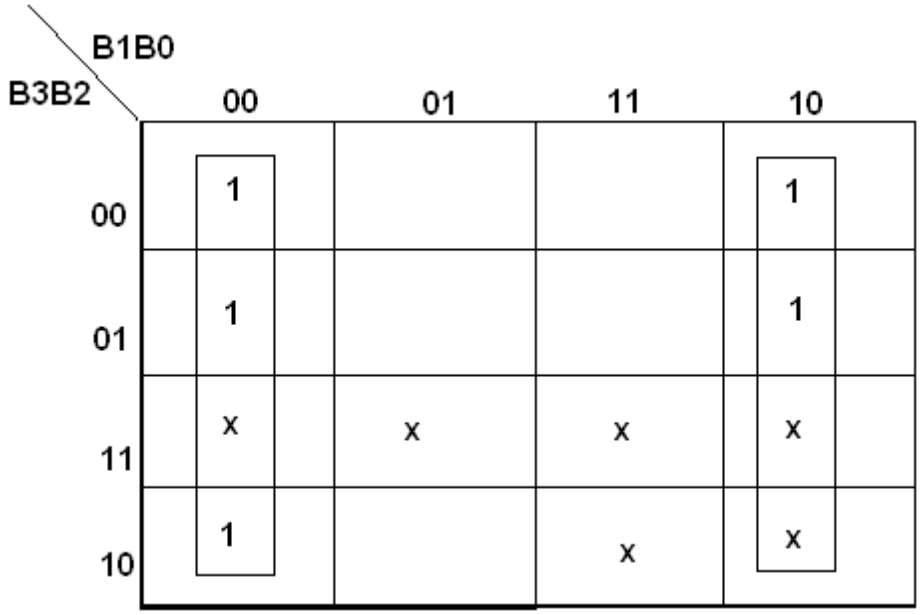
$E2 = B2 \oplus (B1 + B0)$

K-Map for E_1:



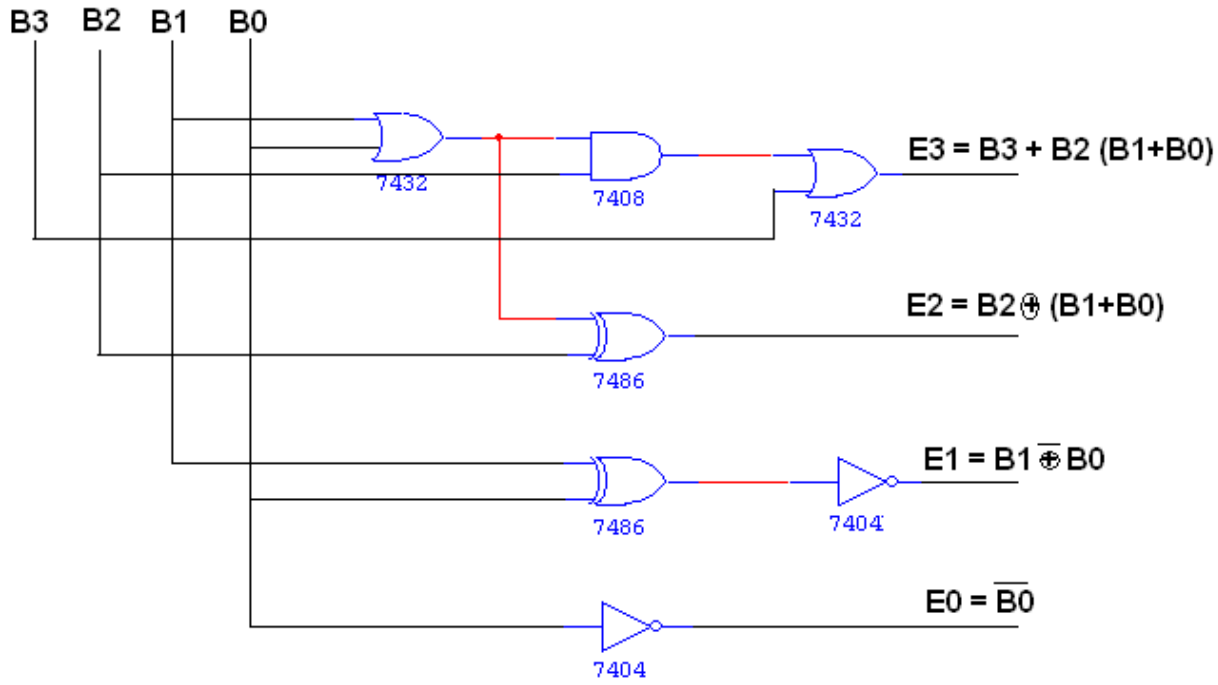
$$E1 = B1 \overline{\oplus} B0$$

K-Map for E_0:



$$E0 = \overline{B0}$$

CIRCUIT 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 BCD to Excess-3 Code Converter was designed and their truth table is verified.

EXPERIMENT NUMBER: 8

EXPERIMENT NAME: Design a Excess-3 to BCD Code Converter.

AIM: To design a Excess-3 to BCD Code Converter using logic gates.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	1
2.	X-OR GATE	IC 7486	1
3.	NOT GATE	IC 7404	1
4.	OR GATE	IC7432	1
5.	IC TRAINER KIT	-	1
6.	CONNECTING WIRES	-	AS REQUIRED

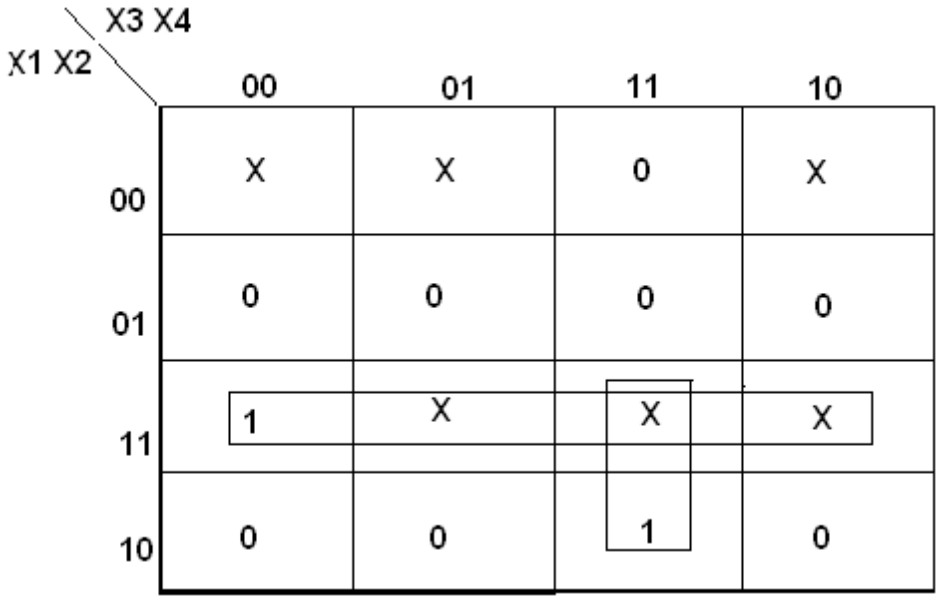
THEORY:

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.

TRUTH TABLE:

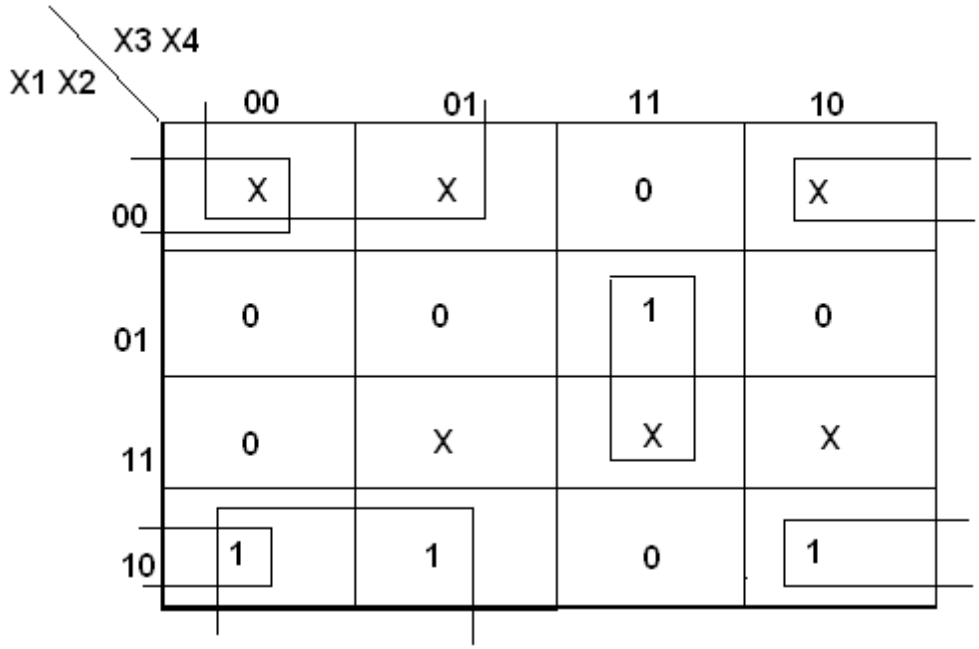
X4	X3	X2	X1	D	C	B	A
0	**0**	**1**	**1**	**0**	**0**	**0**	**0**
0	**1**	**0**	**0**	**0**	**0**	**0**	**1**
0	**1**	**0**	**1**	**0**	**0**	**1**	**0**
0	**1**	**1**	**0**	**0**	**0**	**1**	**1**
0	**1**	**1**	**1**	**0**	**1**	**0**	**0**
1	**0**	**0**	**0**	**0**	**1**	**0**	**1**
1	**0**	**0**	**1**	**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:



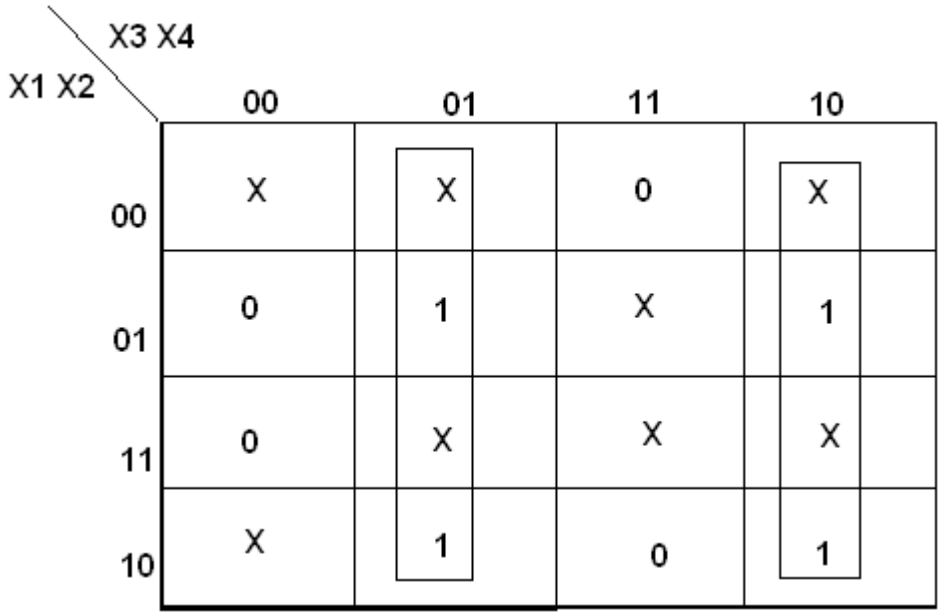
A = X1 X2 + X3 X4 X1

K-Map for B:



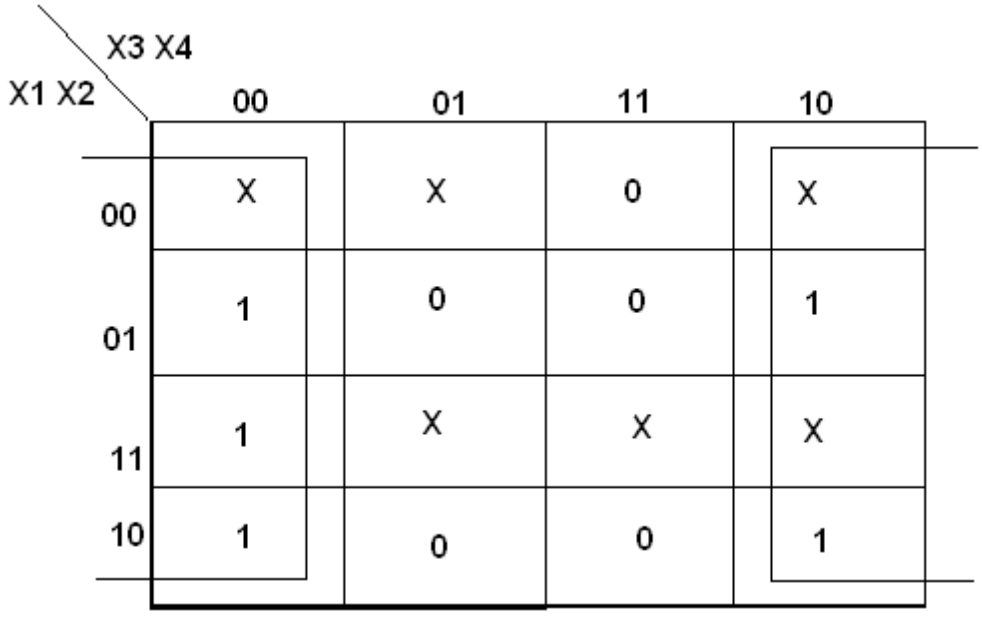
$$B = X2 \oplus (\overline{X3} + \overline{X4})$$

K-Map for C:



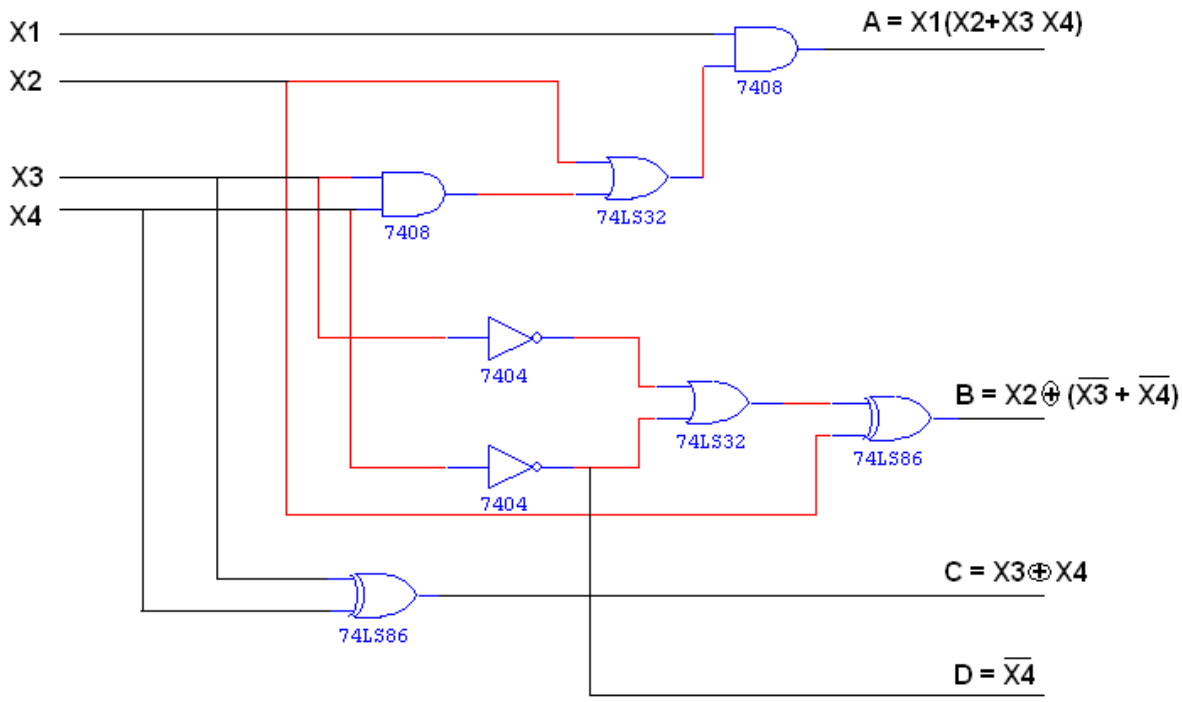
$C = X3 \oplus X4$

K-Map for D:



$D = \overline{X4}$

CIRCUIT 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 Excess-3 to BCD Code Converter was designed and their truth table is verified.

EXPERIMENT NUMBER: 9

EXPERIMENT NAME: Design a 2:4 Line Decoder using logic gates.

AIM: To design a 2:4 Line Decoder using logic gates.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	1
2.	NOT GATE	IC 7404	1
3.	IC TRAINER KIT	-	1
4.	CONNECTING WIRES	-	AS REQUIRED

THEORY:

A decoder is a combinational circuit that connects the binary information from 'n' input lines to a maximum of 2_n unique output lines. Decoder is also called a min-term generator/maxterm generator. A min-term generator is constructed using AND and NOT gates. The appropriate output is indicated by logic 1 (positive logic). Max-term generator is constructed using NAND gates. The appropriate output is indicated by logic 0 (Negative logic).

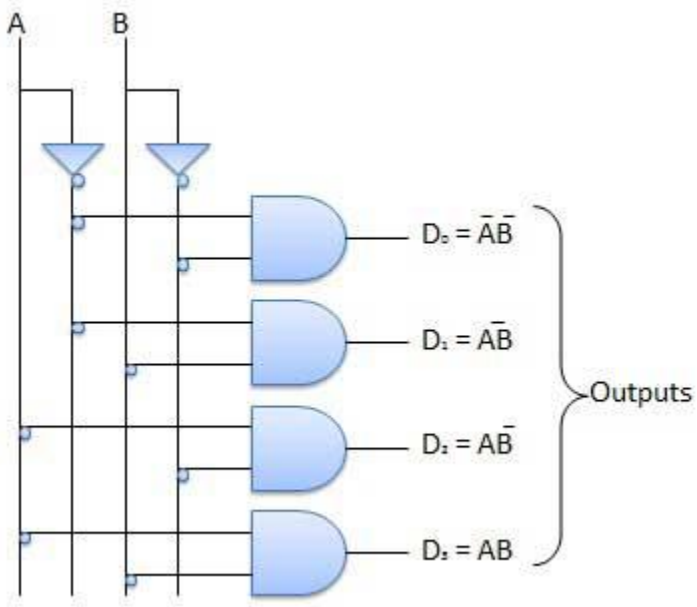
TRUTH TABLE:

A	B	Y0	Y1	Y2	Y3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

After simplification, we get:

Y0 = A'B'
Y1 = A'B
Y2 = AB'
Y3 = AB

CIRCUIT 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 2:4 Line Decoder using logic gates was designed and their truth table is verified.

EXPERIMENT NUMBER: 10

EXPERIMENT NAME: Design a 1:4 Demultiplexer using logic gates.

AIM: To design a 1:4 Demultiplexer using logic gates.

APPARATUS REQUIRED:

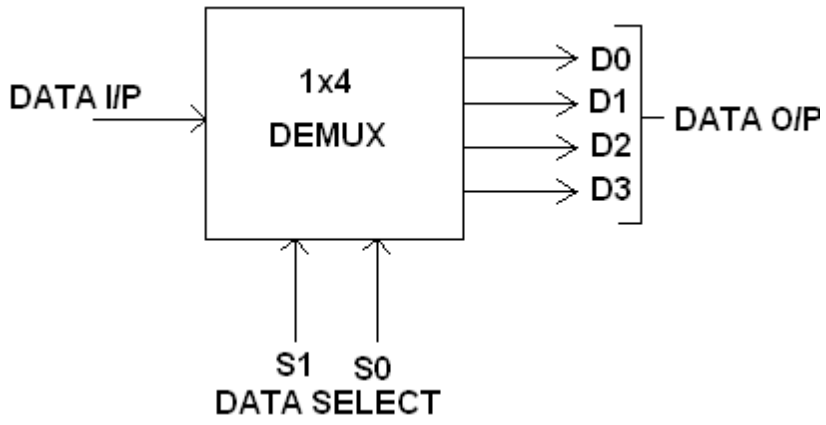
Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	2
2.	NOT GATE	IC 7404	1
3.	IC TRAINER KIT	-	1
4.	CONNECTING WIRES	-	AS REQUIRED

THEORY:

The function of Demultiplexer is in contrast to multiplexer function. It takes information from one line and distributes it to a given number of output lines. For this reason, the demultiplexer is also known as a data distributor. Decoder can also be used as demultiplexer.

In the 1: 4 demultiplexer circuit, the data input line goes to all of the AND gates. The data select lines enable only one gate at a time and the data on the data input line will pass through the selected gate to the associated data output line.

BLOCK DIAGRAM FOR 1:4 DEMULTIPLEXER:

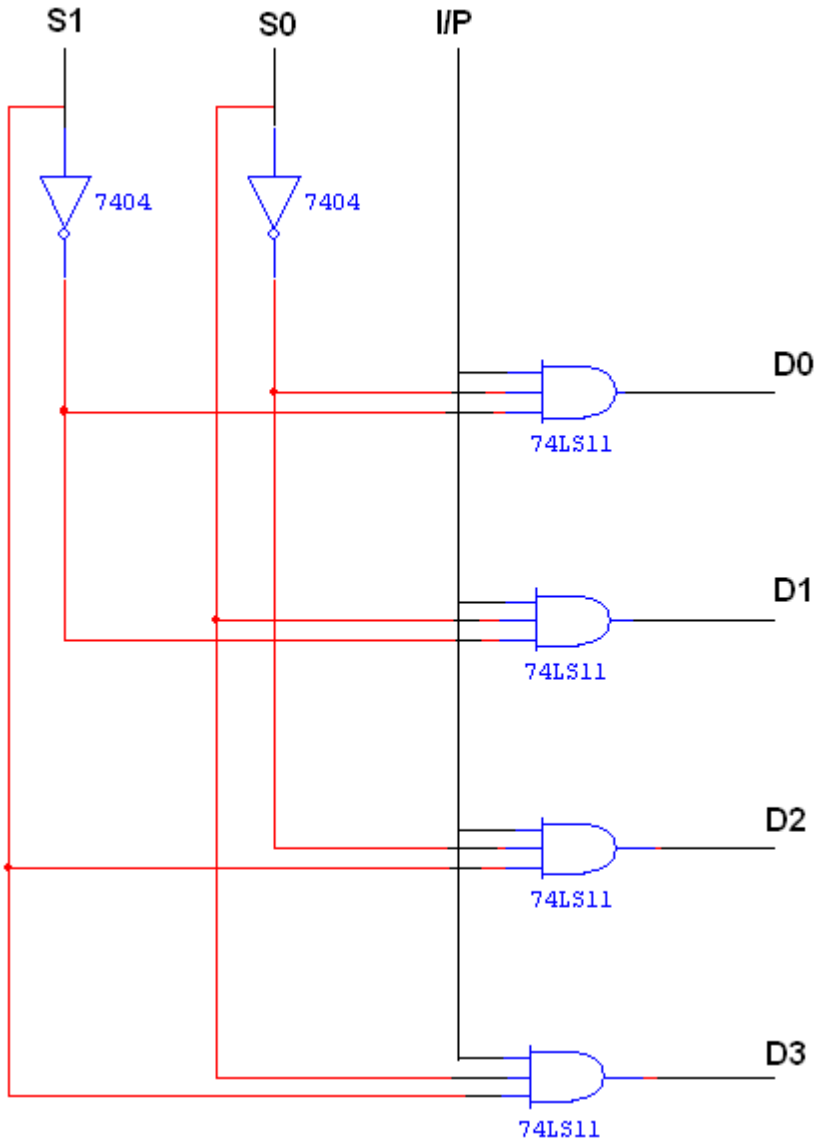


FUNCTION TABLE:

S1	S0	INPUT
0	0	X → D0 = X S1' S0'
0	1	X → D1 = X S1' S0
1	0	X → D2 = X S1 S0'
1	1	X → D3 = X S1 S0

Y = X S1' S0' + X S1' S0 + X S1 S0' + X S1 S0

LOGIC DIAGRAM FOR DEMULTIPLEXER:



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 1:4 Demultiplexer using logic gates was designed and their truth table is verified.

EXPERIMENT NUMBER: 11

EXPERIMENT NAME: Design a 4:1 Multiplexer using logic gates.

AIM: To design a 4:1 Multiplexer using logic gates.

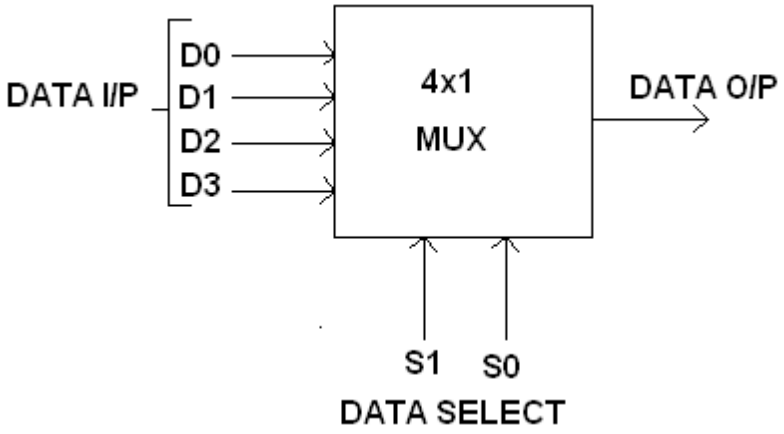
APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	2
2.	NOT GATE	IC 7404	1
3.	OR GATE	IC 7432	1
4.	IC TRAINER KIT	-	1
5.	CONNECTING WIRES	-	AS REQUIRED

THEORY:

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 4:1 MULTIPLEXER:

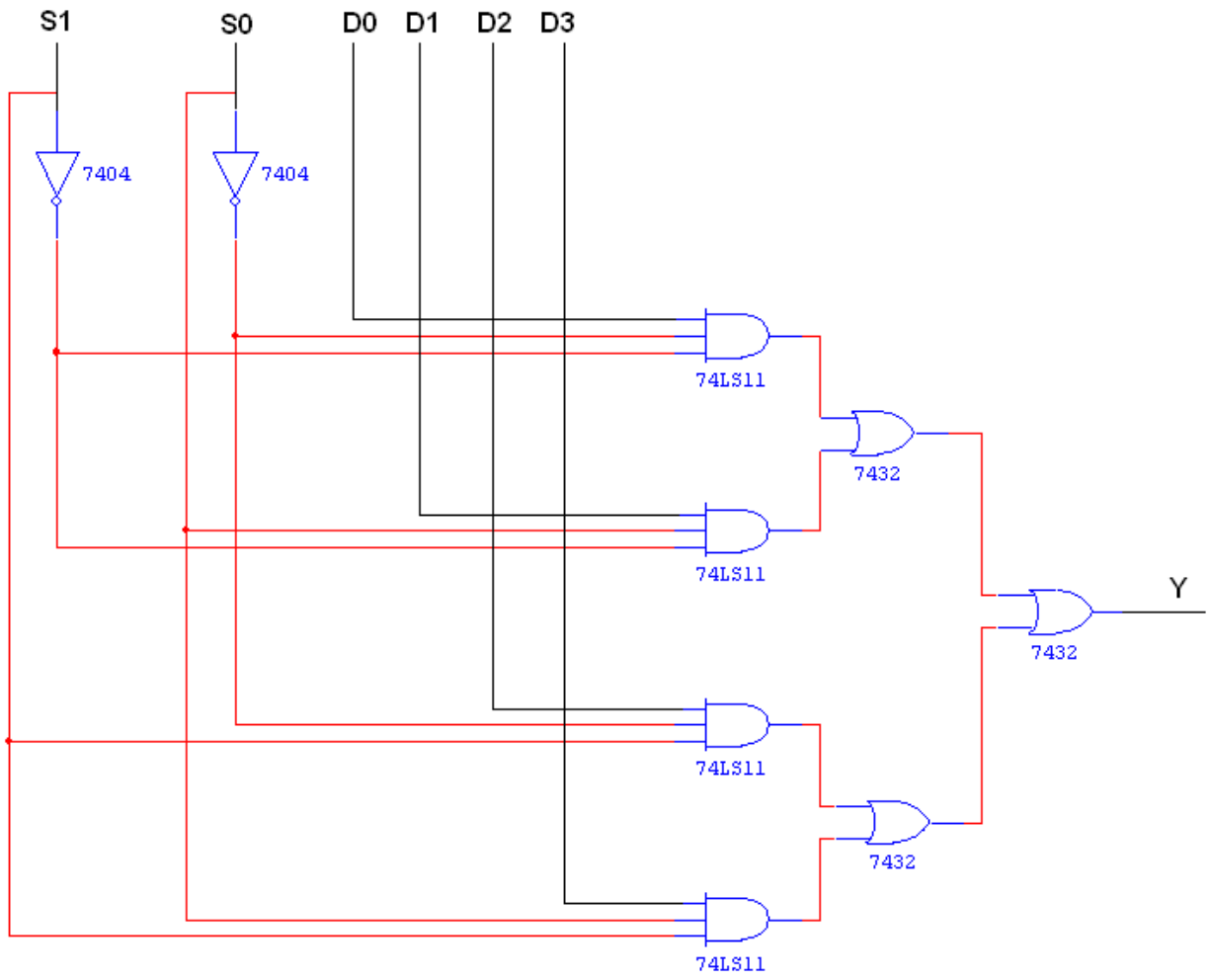


FUNCTION TABLE:

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

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 4:1 Multiplexer using logic gates was designed and their truth table is verified.

EXPERIMENT NUMBER: 12

EXPERIMENT NAME: Design a 8:1 Multiplexer using TWO 4:1 Multiplexers (use multiplexer IC).

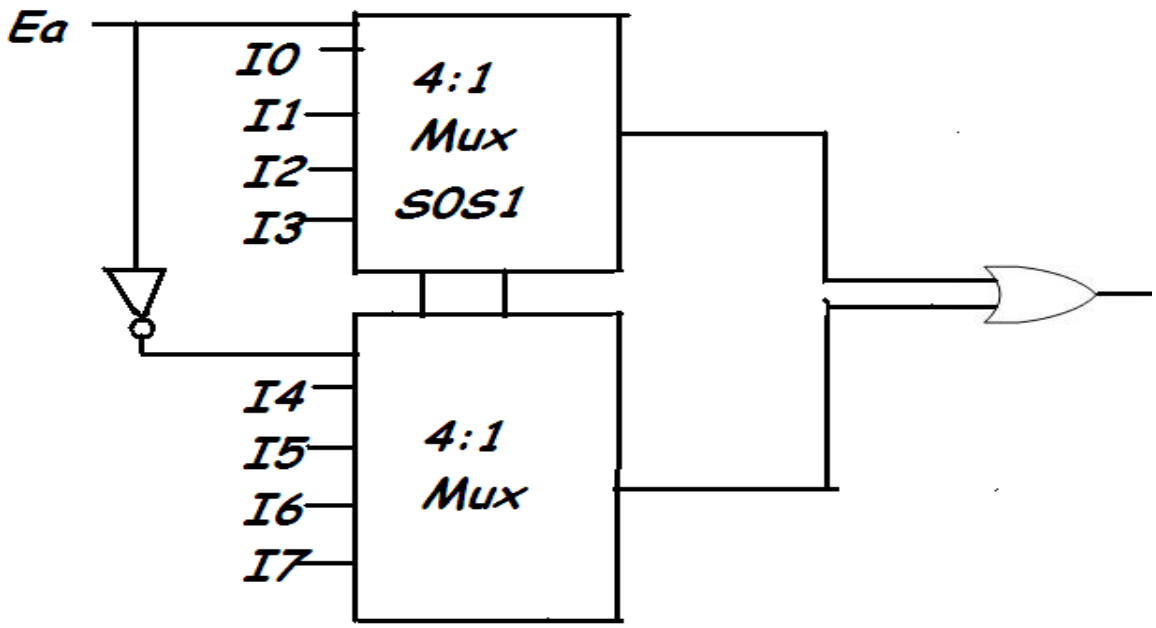
AIM: To design a 8:1 Multiplexer using TWO 4:1 Multiplexers (using IC74153)

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	DUAL 4:1 MUX	IC 74153	2
2.	NOT GATE	IC 7404	1
3.	OR GATE	IC 7432	1
4.	IC TRAINER KIT	-	1
5.	CONNECTING WIRES	-	AS REQUIRED

THEORY:

A data selector, more commonly called a Multiplexer, shortened to "Mux" or "MPX", is combinational logic switching devices that operate like a very fast acting multiple position rotary switches. They connect or control, multiple input lines called "channels" consisting of either 2, 4, 8 or 16 individual inputs, one at a time to an output. Then the job of a multiplexer is to allow multiple signals to share a single common output. A single multiplexer as Ic is 4:1, i.e., it can handle a maximum of 4 inputs.when te number of inputs is more then 4, a multiplexer tree can be used ,also known as multiplexer stack.

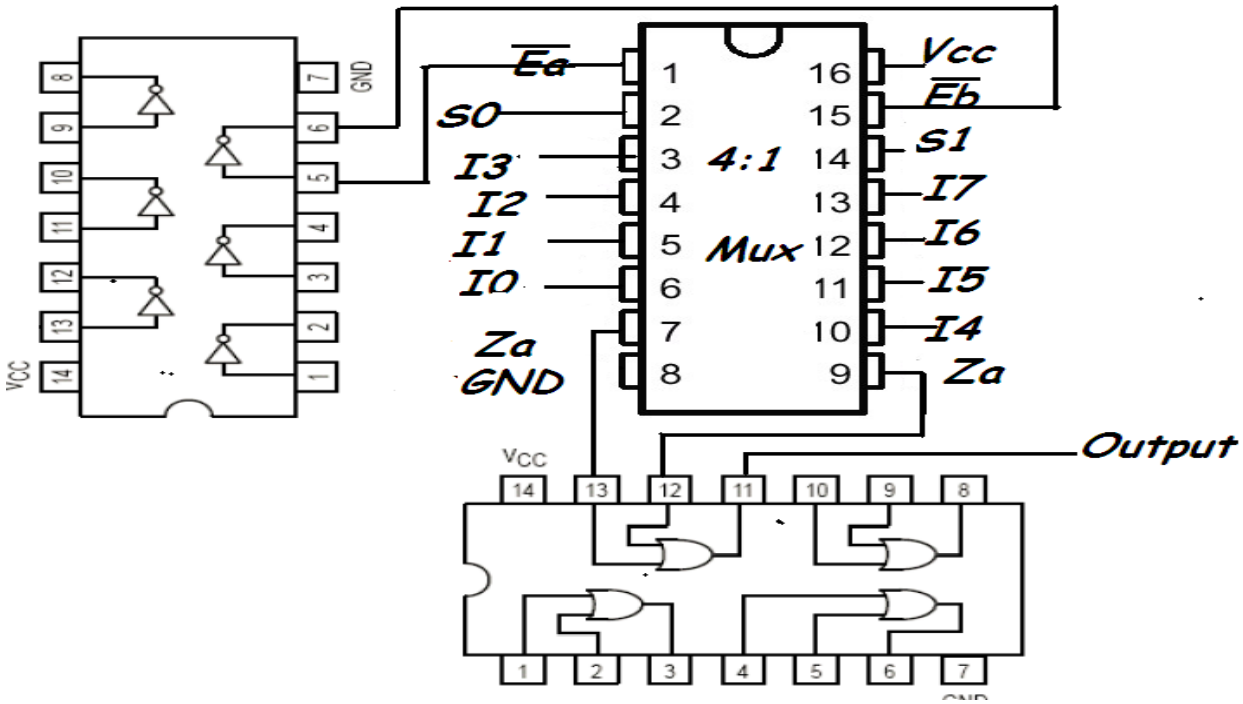
BLOCK DIAGRAM OF 8:1 MULTIPLEXER USING TWO 4:1 MUXs:



Truth Table of 8:1 Mux Using Dual 4:1 Mux

Select Lines			Inputs								Output		
E_a	S_0	S_1	I_o	I_1	I_2	I_3	I_4	I_5	I_6	I_7	Z_a	Z_b	Y
0	0	0	0	×	×	×	×	×	×	×	0	×	0
0	0	0	1	×	×	×	×	×	×	×	1	×	1
0	0	1	×	0	×	×	×	×	×	×	0	×	0
0	0	1	×	1	×	×	×	×	×	×	1	×	1
0	1	0	×	×	0	×	×	×	×	×	0	×	0
0	1	0	×	×	1	×	×	×	×	×	1	×	1
0	1	1	×	×	×	0	×	×	×	×	0	×	0
0	1	1	×	×	×	1	×	×	×	×	1	×	1
1	0	0		×	×	×	0	×	×	×	×	0	0
1	0	0	×	×	×	×	1	×	×	×	×	1	1
1	0	1	×	×	×	×	×	0	×	×	×	0	0
1	0	1	×	×	×	×	×	1	×	×	×	1	1
1	1	0	×	×	×	×	×	×	0	×	×	0	0
1	1	0	×	×	×	×	×	×	1	×	×	1	1
1	1	1	×	×	×	×	×	×	×	0	×	0	0
1	1	1	×	×	×	×	×	×	×	1	×	1	1

CIRCUIT 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 8:1 Multiplexer using TWO 4:1 Multiplexers (use multiplexer IC) was designed and their truth table is verified.

EXPERIMENT NUMBER: 13

EXPERIMENT NAME: Design a Full Adder using Decoder IC.

AIM: To design a Full Adder using IC 74138.

APPARATUS REQUIRED:

Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	3:8 DECODER	IC 74138	1
2.	4 I/P NAND GATE	IC 7420	1
3.	IC TRAINER KIT	-	1
4.	CONNECTING WIRES	-	AS REQUIRED

THEORY:
A full adder is a combinational circuit that forms the arithmetic sum of input; it consists of three inputs and two outputs.
IC 74138 is a 3:8 line decoder IC. The min terms of Sum and Carry are used to find out the expression.

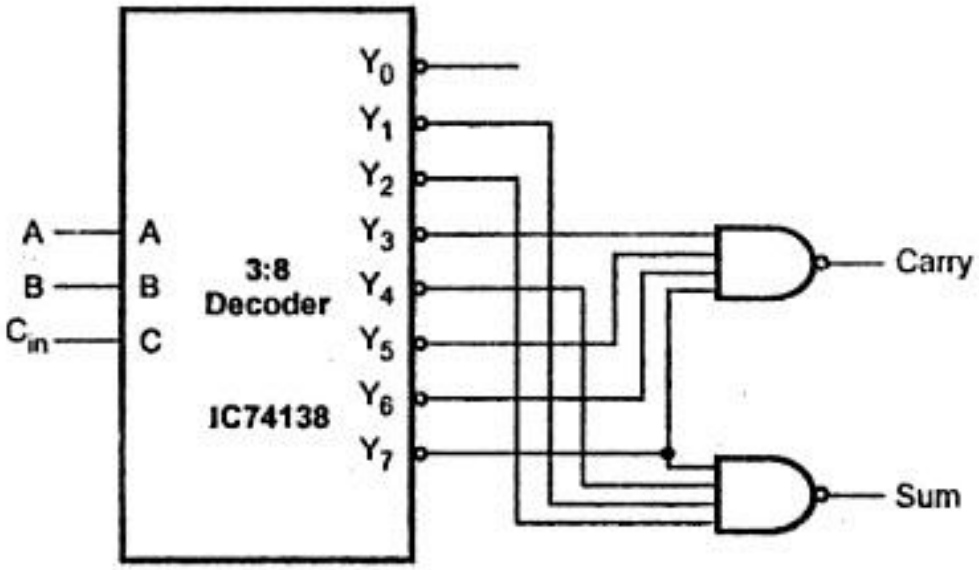
TRUTH TABLE OF FULL ADDER:

A	B	C	CARRY	SUM
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

From the truth table, we get:-

$S = \sum m (1,2,4,7)$
$C = \sum m (3,5,6,7)$

CIRCUIT 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 Full Adder using Decoder IC was designed and their truth table is verified.

EXPERIMENT NUMBER: 14

EXPERIMENT NAME: Design a Half Adder using a 4:1 Multiplexer IC.

AIM: To design a Half Adder using IC 74153.

APPARATUS REQUIRED:

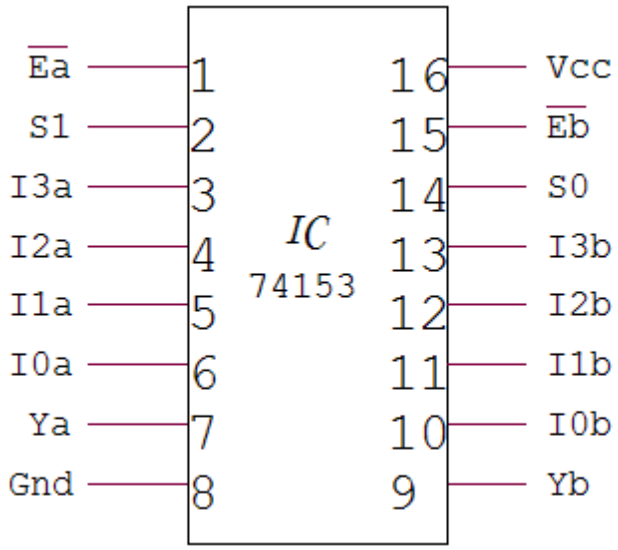
Sl.No.	COMPONENT	SPECIFICATION	QTY.
1.	4:1 MUX	IC 74153	1
2.	IC TRAINER KIT	-	1
3.	CONNECTING WIRES	-	AS REQUIRED

THEORY:
A half adder has two inputs for the two bits to be added and two outputs one from the sum ' S' and other from the carry ' c' into the higher adder position.

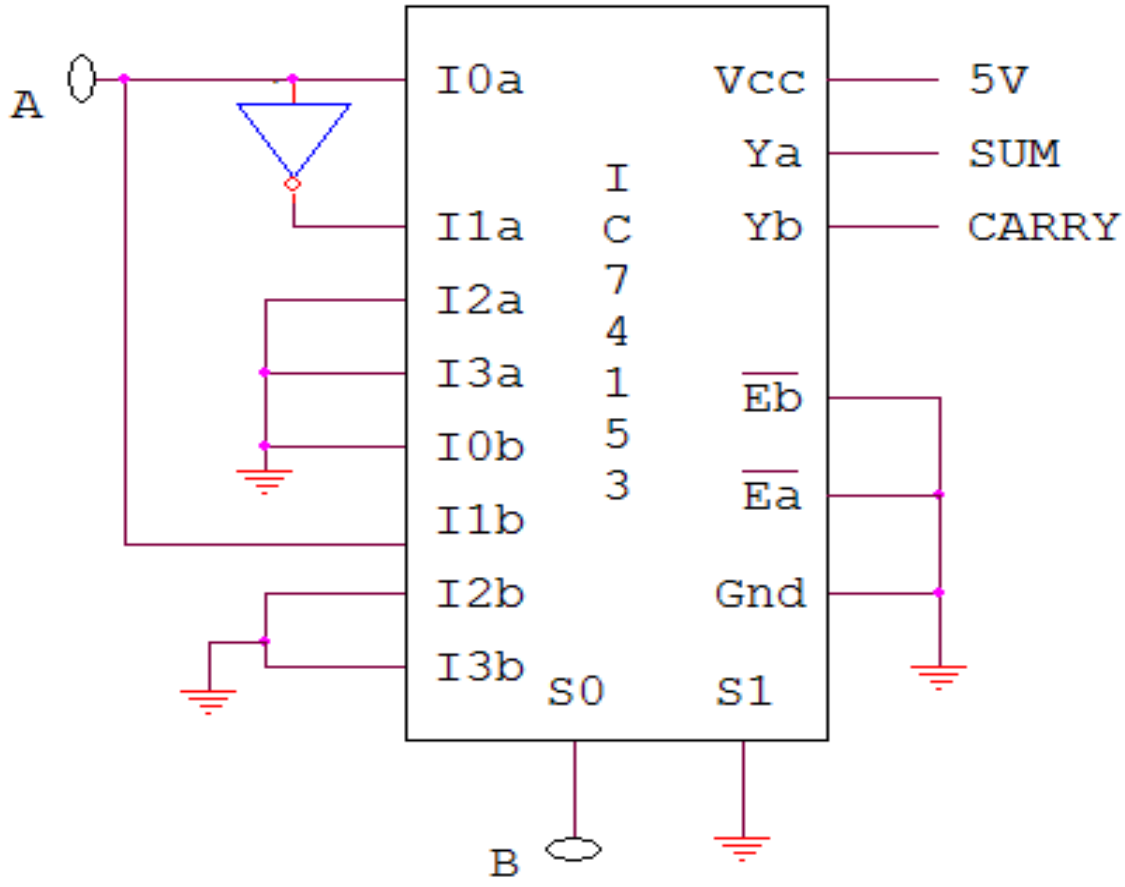
TRUTH TABLE OF HALF ADDER:

A	B	CARRY	SUM
0	**0**	**0**	**0**
0	**1**	**0**	**1**
1	**0**	**0**	**1**
1	**1**	**1**	**0**

PIN DIAGRAM OF 74153:



CIRCUIT 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 Half Adder using a 4:1 Multiplexer IC was designed and their truth table is verified.