BSc. Electronics 3rd Sem.

Digital Electronics & VHDL Lab

Lab Course Code: PS/ELEC/C- 302L

List of experiments

1. Introduction to digital laboratory equipment and ICs.
2. To verify the truth table of AND, OR, NOT, NAND, NOR, EX-OR, EX-NOR, logic gates.
3. To design and verify AND, OR, NOT, NOR and XOR gates using NAND/NOR gate only.
4. Implementation of half subtractor and Full subtractor using logic gates.
5. To set up and test a 7-segment static display system to display numbers 0 to 9.
6. To design & implement 4-bit Adder & Subtractor using Full Adder IC 7483.
7. To design and implement an encoder and decoder using logic gates.
8. To design and implement Multiplexer and Demultiplexer using logic gates.

Experiment No : 1

AIM : Introduction to Digital Laboratory Equipments & IC"s

The Breadboard

The breadboard consists of two terminal strips and two bus strips (often broken in the center). 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 center 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+5Vand 0Vpower supply connections to separate bus strips.

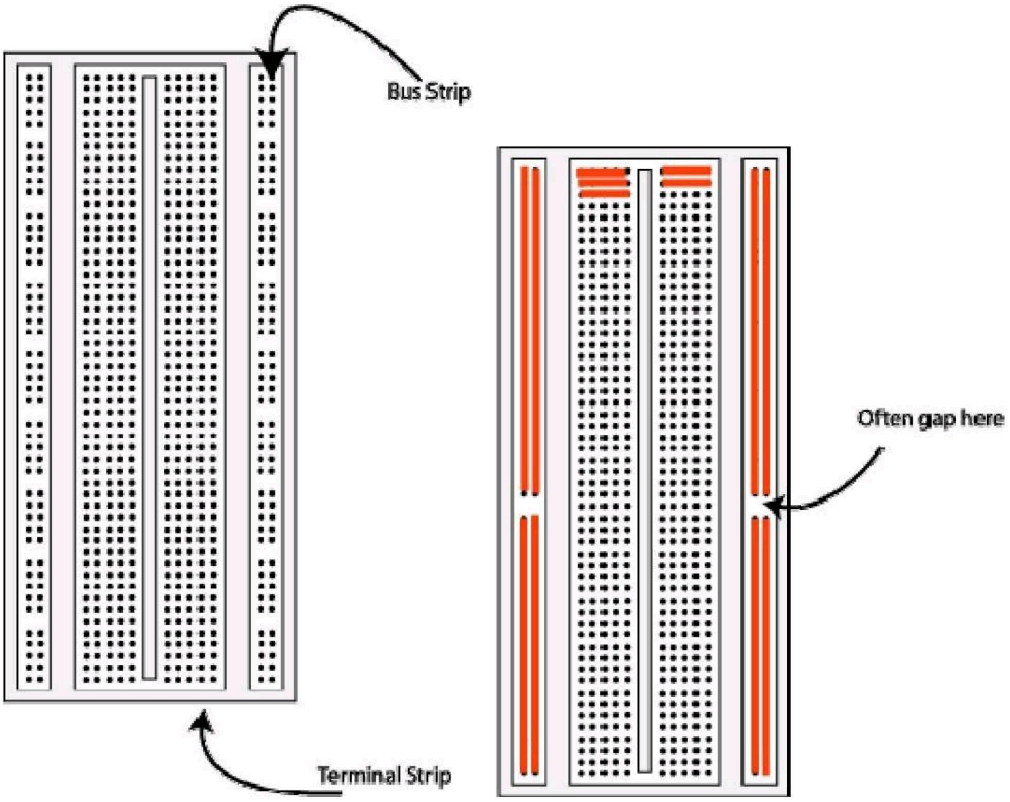


Fig1.The breadboard. The lines indicate connected holes.

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 <u>before</u> switching on power.

<u>Building the Circuit :</u>

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 +5 V 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+5VandGNDpinsofeachchiptothepowerandgroundbusstripsonthebreadboard.
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.

<u>Common Causes of Problems:</u>

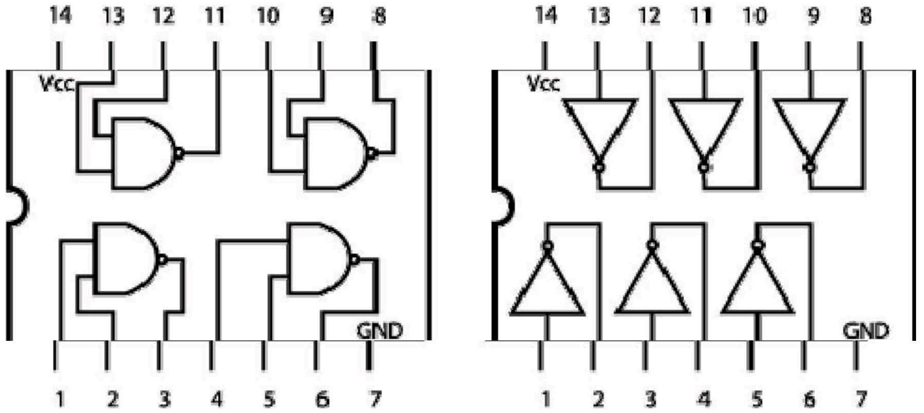
1 Not connecting the ground and/or power pins for all chips.
2 Not turning on the power supply before checking the operation of the circuit.
3 Leaving out wires.
4 Plugging wires into the wrong holes.
5 Driving a single gate input with the outputs of two or more gates
6 Modifying the circuit with the 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 some body 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 $\bar{A}$. You should use that notation during the write-up of your laboratory experiments.



Quad2 Input7400Hex7404Inverter

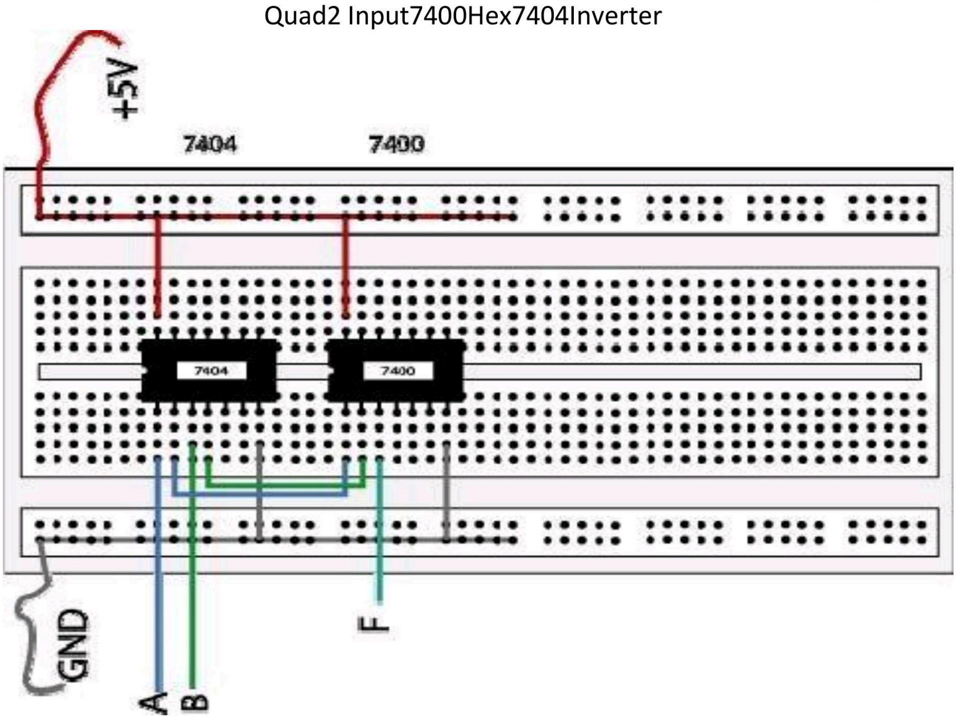


Fig2.The complete designed and connected circuit

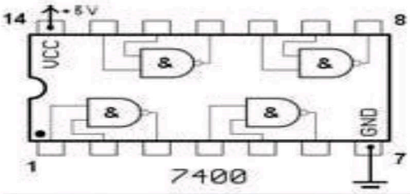
Sometimes the chip manufacturer may denote the first pin by a small indented circle above the first pin of the chip. Place your chips in the same direction, to save confusion at a later stage . Remember that you must connect power to the chips to

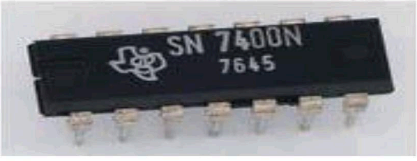
Get them to work.

UsefulICPindetails

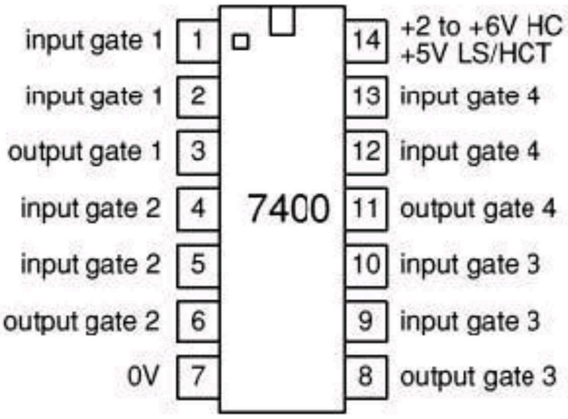
ICNUMBER	DescriptionofIC
7400	Quad2inputNANDGATE
7401	Quad2inputNANDGate(opencollector)
7402	Quad2 input NORGate
7403	Quad2inputNORGates(opencollector)
7404	HexInverts
7421	Dual4inputANDGates
7430	8inputNANDGate
7432	Quad2input ORGates
7486	Quad2input EX-ORGate
74107	Dualj-kFlipFlop
74109	Dualj-kFlipFlop
74174	Hex DFlipFlop
74173	QuadDFlip Flop
7473	Dualj-kFlipFlop
7474	DualDFlipFlop
7475	QuadBi-stablelatch

7476	Dual j-k Flip Flop

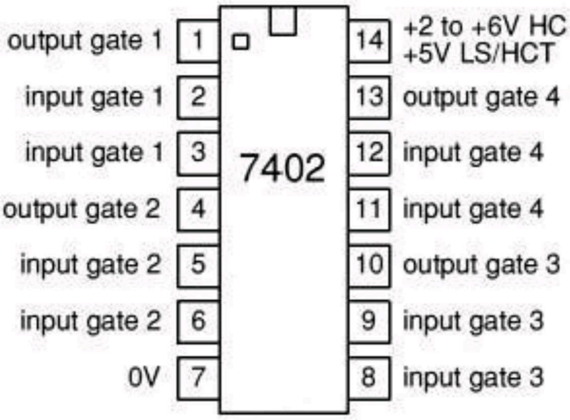




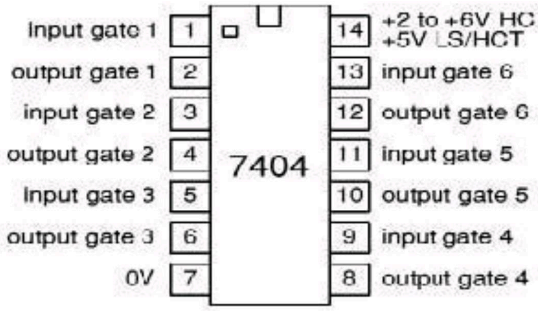
7400(NAND)



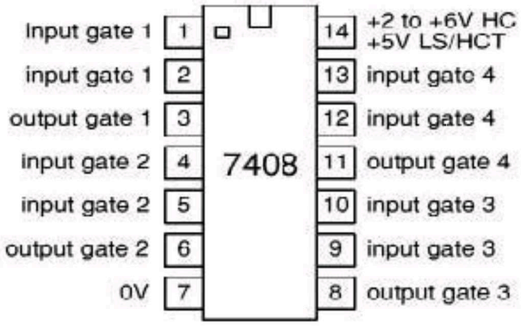
7402(NOR)



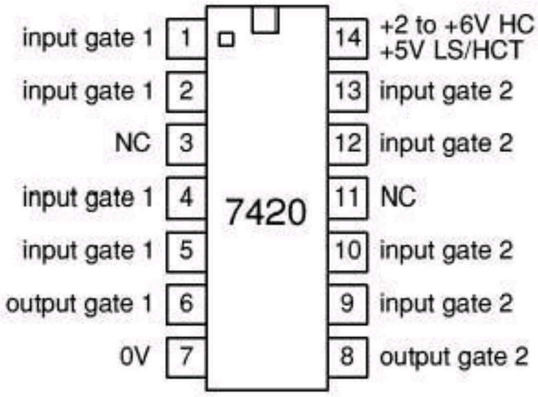
7404(NOT)



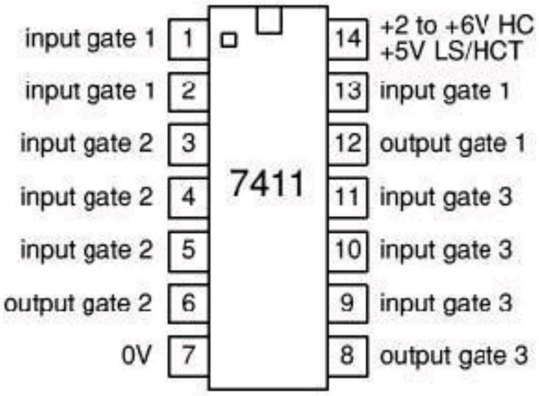
7408(AND)

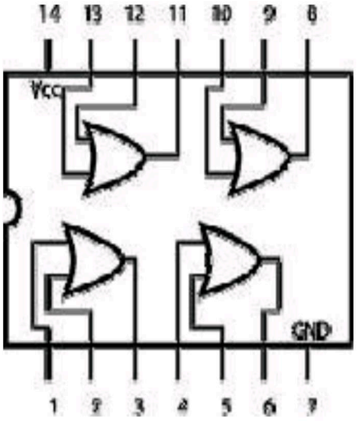


7420(4-i/pNAND)

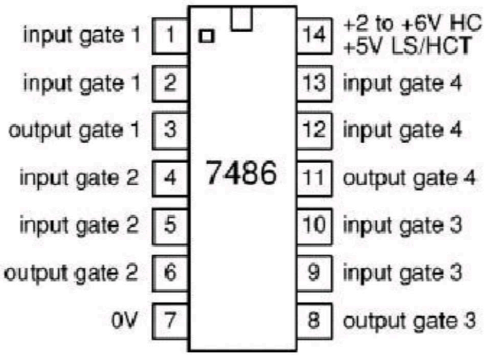


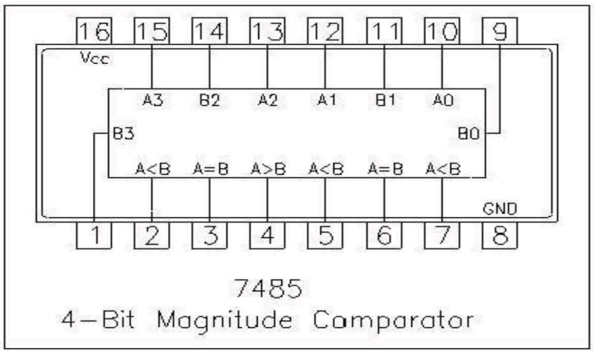
7411(3-i/p AND) 7432(OR)





7486(EX-R)





Experiment No. 2

Aim: To study and verify the Truth Tables of AND, OR, NOT, NAND, NOR, EXOR logic gates

Components: IC 7400, 7402, 7404,7408,7432,7486
Apparatus: Prototyping board (breadboard), DC Power Supply, Connecting Wires

Theory :

AND, OR and NOT gates are **basic gates**. XOR and XNOR are **universal gates**. Basically logic gates are electronic circuits because they are made up of number of electronic devices and components. Inputs and outputs of logic gates can occur only in two levels. These two levels are term HIGH and LOW, or TRUE and FALSE, or ON AND off, OR SIMPLY 1 AND 0. A table which lists all possible combinations of input variables and the corresponding outputs is called a „truth table‟. It shows how the logic circuit‟s output responds to various combinations of logic levels at the inputs.

AND GATE :-

An AND gate has two or more inputs but only one output. The output assumes the logic 1 state only when each one of its inputs is at logic 1 state. The output assumes logic 0 state even if one of its input is at logic 0 state. AND gate is also called an „all or nothing‟ gate.

The logic symbol & truth table of two input AND gate are shown in figure 1.a & 1.b respectively. The symbol for AND operation is „.‟.

With input variables A & B the Boolean expression for output can be written as;

$$X = A.B$$

Logic symbol:

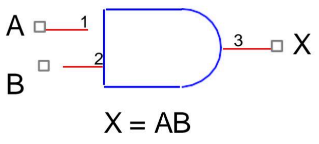


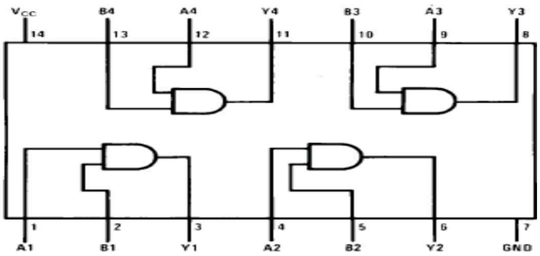
Fig.1.a

Truth table:

Input		Output
A	**B**	**X**
0	0	0
0	1	0
1	0	0
1	1	1

Fig.1.b

Pin diagram of IC**74LS08**:



OR GATE

Like an AND gate, an OR gate may have two or more inputs but only one output. The output assumes the logic 1 state, even if one of its inputs is in logic 1 state. Its output assumes logic 0 state, only when each one of its inputs is in logic 0 state. OR gate is also called an „any or all‟ gate. It can also be called an inclusive OR gate because it includes the condition „both the input can be present‟.

The logic symbol & truth table of two input OR gate are shown in figure 1.c & 1.d respectively. The symbol for OR operation is „+‟.

With input variables A & B the Boolean expression for output can be written as;

$$\mathbf{X = A + B}$$

Logic symbol:

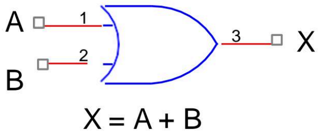


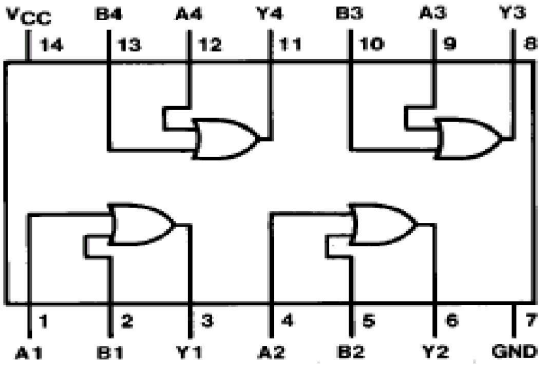
Fig.1.c

Truth table:

Input		Output
A	**B**	**X**
0	0	0
0	1	1
1	0	1
1	1	1

Fig.1.d

Pin diagram of IC**74LS32**:



NOT GATE

A NOT gate is also known an inverter, has only one input and only one output. It is a device whose output is always the complement of its input. That is the output of a not gate assumes the logic 1 state when its input is in logic 0 state and assumes the logic 0 state when its input is in logic 1 state.

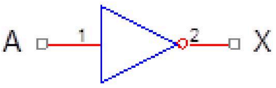
The logic symbol & truth table of NOT gate are shown in figure 1.e & 1.f respectively. The symbol for NOT operation is „-„-‟(bar).

With input variable A the Boolean expression for output can be written as;

$$X = \bar{A}$$

This is read as "X is equal to a bar".

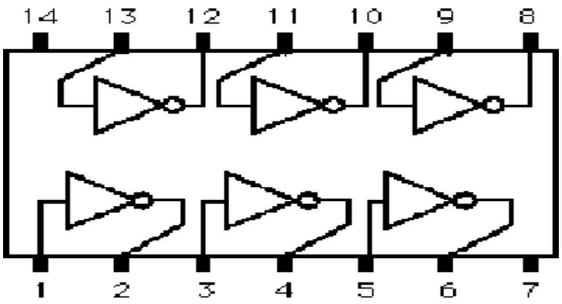
Logic symbol: Truth table:



Input	Output
A	**X**
0	1
1	0

Fig.1.e & Fig.1.f

Pin diagram for IC**74LS04**:



NAND GATE

NAND gate is universal gate. It can perform all the basic logic function. NAND means NOT AND that is, AND output is NOTed.so NAND gate is combination of an AND gate and a NOT gate. The output is logic 0 level, only when each of its inputs assumes a logic 1 level. For any other combination of inputs, the output is logic 1 level. NAND gate is equivalent to a bubbled OR gate.

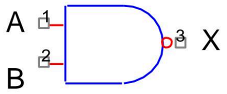
The logic symbol & truth table of two input NAND gate are shown in figure 1.g & 1.h respectively.

With input variables A & B the Boolean expression for

output can be written as;

$$X = \overline{AB}$$

Logic symbol:



$X = \overline{AB}$

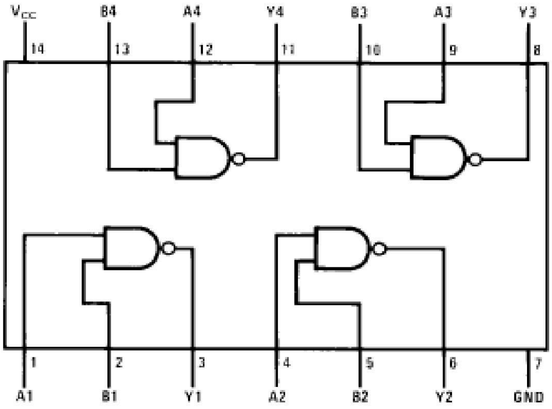
Fig.1.g

Truth table:

Input		Output
A	**B**	**X**
0	0	1
0	1	1
1	0	1
1	1	0

Fig.1.h

Pin diagram for IC**74LS00**:

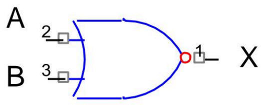


NOR GATE

NOR gate is universal gate. It can perform all the basic logic function. NOR means NOT OR that is, OR output is NOTed.so NOR gate is combination of an OR gate and a NOT gate. The output is logic 1 level, only when each of its inputs assumes a logic 0 level. For any other combination of inputs, the output is logic 0 level. NOR gate is equivalent to a bubbled AND gate. The logic symbol & truth table of two inputs NOR gate are shown in figure 1.i& 1.j respectively. With input variables A & B the Boolean expression for output can be written as;

$$X = \overline{A+B}$$

Logic symbol:



$X = \overline{A + B}$

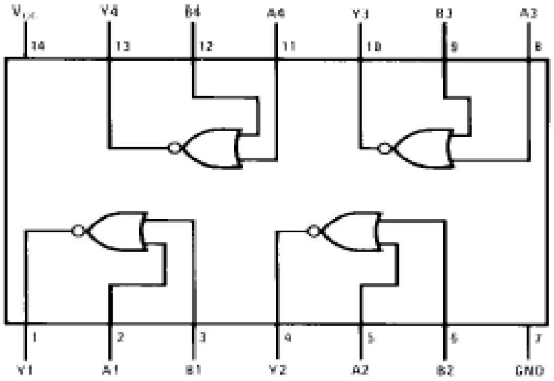
Fig.1.i

Truth table:

Input		Output
A	**B**	**X**
0	0	1
0	1	0
1	0	0
1	1	0

Fig.1.j

Pin diagram for IC**74LS02**:

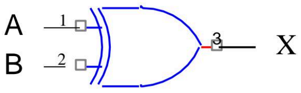


EXCLUSIVE-OR (X-OR) GATE

An X-OR gate is a two input, one output logic circuit, whose output assumes a logic 1 state when one and only one of its two inputs assumes a logic 1 state. Under the condition when both the inputs are same either 0 or 1, the output assumes a logic 0 state. Since an X-OR gate produces an output 1 only when the inputs are not equal, it is called as an anti-coincidence gate or inequality detector. The output of an X-OR gate is the modulo sum of its two inputs. The logic symbol & truth table of two input X-OR gate are shown in figure 1.k & 1.l respectively. The symbol for X-OR operation is „⊕“. With input variables A & B the Boolean expression for output can be written as;

$$X= A \oplus B$$

Logic symbol:



$X= A \oplus B$

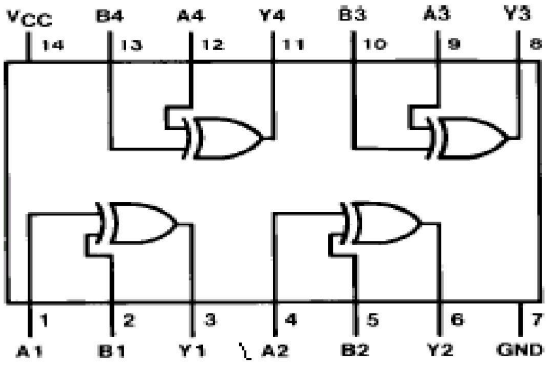
Fig.1.k

Truth table:

Input		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

Fig.1.l

Pin diagram for IC**74LS86**:



Procedure:-

- Set up the breadboard.
- Test all required ICs.
- Give various combinations of inputs and check the output.
- Repeat the procedure for each IC.

Observation Table: LED ON (RED light): Logic 1
LED OFF (Green Light): Logic 0
Input variables: A, B
Output variable: Y

Sr. No	Input(A) LED	Input(B) LED	Output (OR) $Y = \overline{A}$	Output (AND) Y=AB	Output (OR) Y=A+B	Output (NAND) $Y = \overline{AB}$	Output (NOR) $Y= \overline{A+B}$	Output (XOR) $Y=A\oplus B$
1								
2								
3								
4								

Results and Analysis:
NOT Gate: When logic 1 is applied to one of NOT gate of 7404 IC, then output becomes zero. When input LED is ON (RED), the output LED become OFF (Green) vice versa.

OR Gate: The output of an OR gate is a 1 if one or the other or both of the inputs are 1, but a 0 if both inputs are 0. When One or the other or Both of the input LEDS are ON (RED Light), then output LED is ON(RED) otherwise Output LED is OFF(Green Light)

AND Gate: The output of an AND gate is only 1 if both its inputs are 1. For all other possible inputs the output is 0.When both the LEDS are On, then output LED is ON (RED Light) otherwise Output LED is OFF.

NOR Gate: The output of the NOR gate is a 1 if both inputs are 0 but a 0 if one or the other or both the inputs are 1.
NAND Gate: The output of the NAND gate is a 0 if both inputs are 1 but a 1 if one or the other or both the inputs are 0.

EXOR gate: The output of the XOR gate is a 1 if either but not both inputs are 1 and a 0 if the inputs are both 0 and both 1.

Conclusion: Any Boolean expression can be realized using NOT, AND, OR, NAND,NOR, EXOR gates.

Experiment No. 3

Aim : - realization of logic gates using NAND and NOR

Gates. Components : - IC 7400, 7402

Apparatus : - Digital trainer kit, wires, probes, etc.

Theory:-

Universal gates: -

The Nand and Nor gates are called as universal gates, because it is possible to implement any Boolean expression with the help of only Nand or only Nor gates.

Hence a user can build any combinational circuit with the help of only Nand gates or only Nor gates.

The NAND & NOR gates are called as „Universal gates‟. Because it is possible to implement any Boolean expression with the help of only NAND or only NOR gate. We can construct AND, OR, NOT, X-OR & X-NOR gates.

The Boolean expression for NAND gate is,

$$X = \overline{AB}$$

The Boolean expression for NOR gate is,

$$X = \overline{A + B}$$

This is a great advantage because a user will have to make a stock of only Nand or Nor gates.

All gates using Nand Gate:-

1) Not using Nand:-

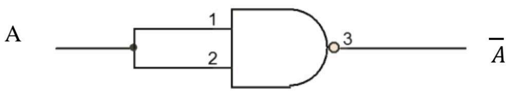
The Boolean expression for NOT gate is $A = \overline{A}$ Fig. shows the realization of a NOT gate using a two i/p NAND gate. As both i/p‟s are connected together we can write i/p A=B=A So o/p is given as

$$Y = \overline{A.B}$$

$$Y = \overline{A.A} \qquad \therefore A = B$$

But A.A=A ∴ by AND law

$$Y = \overline{A}$$



2) AND using NAND:-

The Boolean expression for an AND gate is Y=A.B

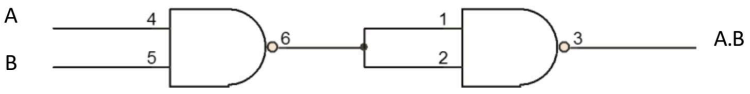
Taking double inversion,

$$Y = \overline{\overline{A.B}}$$

$$\text{But } \overline{\overline{A}} = A$$

$$Y=A.B$$

This equation can be realized using only NAND gate as shown in fig.



3) OR using NAND:-

The Boolean expression for an OR gate is Y=A + B

Taking double inversion,

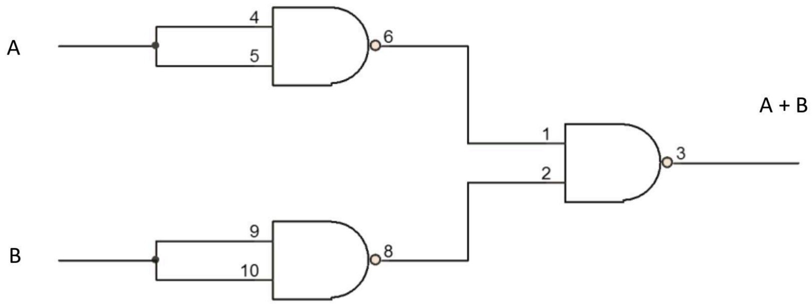
$$Y = \overline{\overline{A+B}}$$

But by DE-Morgan‟s theorem

$$\overline{AB} = \bar{A}\ \bar{B}$$

$$Y = \overline{\overline{AB}}$$

This is required expression for OR gate



4) NOR using NAND:-

The Boolean expression for an NOR gate is $Y = \overline{A+B}$

But by DE-Morgan‟s theorem

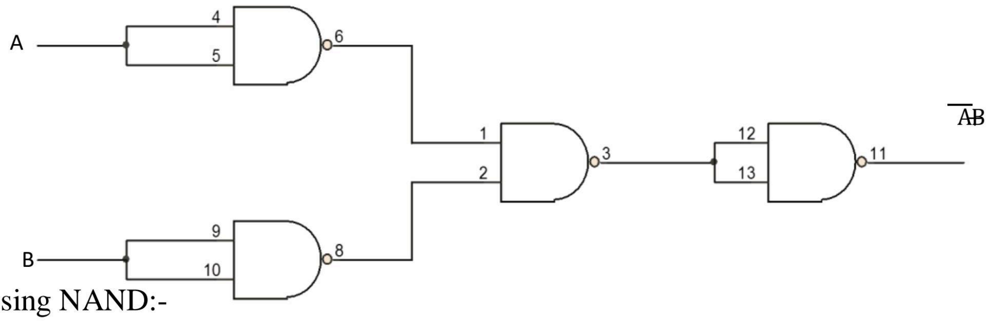
$$\overline{A+B} = \bar{A}\bar{B}$$

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

Taking double inversion,

$$Y = \overline{\overline{\bar{A}\bar{B}}}$$

This is required expression for NOR gate



5) Ex-OR using NAND:-

The expression for Ex- OR gate is

$$Y = A \oplus B$$

$$Y = \bar{A}.B + A.\bar{B}$$

Taking double inversion

$$Y = \overline{\overline{\bar{A}.B + A.\bar{B}}}$$

Let $\bar{A}.B$ =X and $A.\bar{B}$=Z

$$Y=\overline{\overline{X+Z}}$$

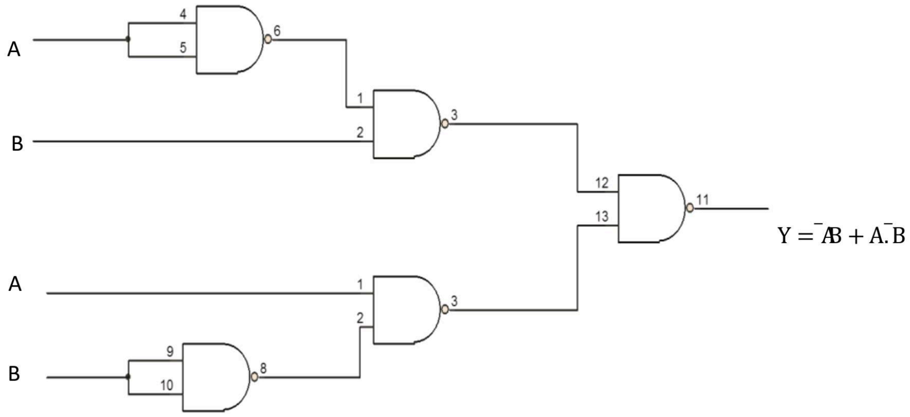
$$\text{using De} - \text{Morgan}'\text{stheorem}$$

$$\overline{X+Z} = \bar{X}\,.\,\bar{Z}$$

$$Y = \overline{\bar{X}\bar{Z}}$$

$$Y = \overline{(\overline{\bar{A}B})\,.\,(\overline{A.\bar{B}})}$$

This is required expression for Ex-OR gate using NAND gate.



All gates using NOR Gate:-

1) Not using NOR:-

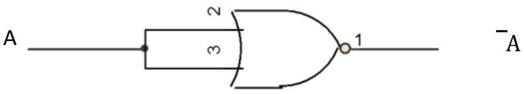
The Boolean expression for NOT gate is $A = \overline{A}$ Fig. shows the realization of a NOT gate using a two i/p NOR gate. As both i/p‟s are connected together we can write i/p A=B=A So o/p is given as

$$Y = \overline{A + B}$$

$$Y = \overline{A + A} \qquad \therefore A = B$$

$$\text{But } A+A=A \qquad \therefore \text{by OR law}$$

$$Y = \overline{A}$$



2) OR using NOR:-

The Boolean expression for an OR gate is Y=A+B

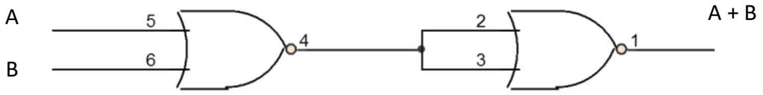
Taking double inversion,

$$Y = \overline{\overline{A + B}}$$

$$\text{But } \overline{\overline{A}} = A$$

$$Y=A+B$$

This equation can be realized using only NAND gate as shown in fig.



3) AND using NOR:-

The Boolean expression for an AND gate is Y=A.B

Taking double inversion,

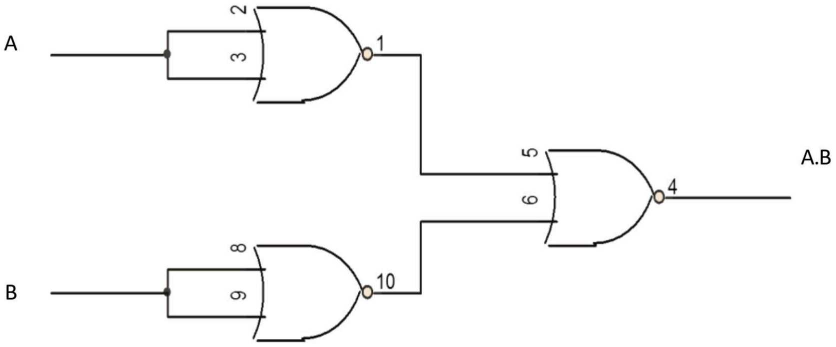
$$Y = \overline{\overline{A.B}}$$

But by DE-Morgan"s theorem

$$\overline{A.B} = \overline{A} + \overline{B}$$

$$Y = \overline{\overline{A.B}}$$

This is required expression for AND gate



4) NAND using NOR:-

The Boolean expression for an NOR gate is $Y = \overline{A.B}$

But by DE-Morgan"s theorem

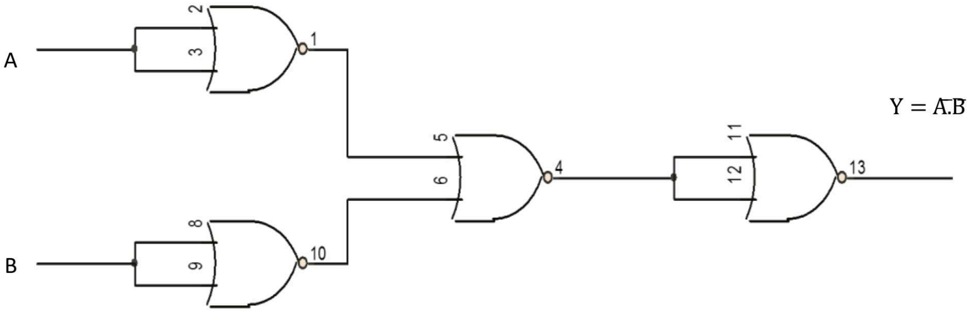
$$\overline{A.B} = \overline{A} + \overline{B}$$

$$Y = \overline{A} + \overline{B}$$

Taking double inversion,

$$Y = \overline{\overline{\overline{A} + \overline{B}}}$$

This is required expression for NAND gate



5) EX-OR using NOR:-

The expression for EX- OR gate is

$$Y = A \oplus B$$

$$Y = \bar{A}.B + A.\bar{B}$$

Taking double inversion

$$Y = \overline{\overline{\bar{A}.B + A.\bar{B}}}$$

Let $\bar{A}.B = X$ and $A.\bar{B} = Z$

$$Y = \overline{\overline{X + Z}}$$

$$\text{using De} - \text{Morgan's theorem}$$

$$\overline{X + Z} = \bar{X}.\bar{Z}$$

$$Y = \overline{\bar{X}\bar{Z}}$$

$$Y = \overline{\overline{(\bar{A}.B)}.\overline{(A.\bar{B})}}$$

But $\overline{\bar{A}B} = A + \bar{B}$ and $\overline{A.\bar{B}} = \bar{A} + B$

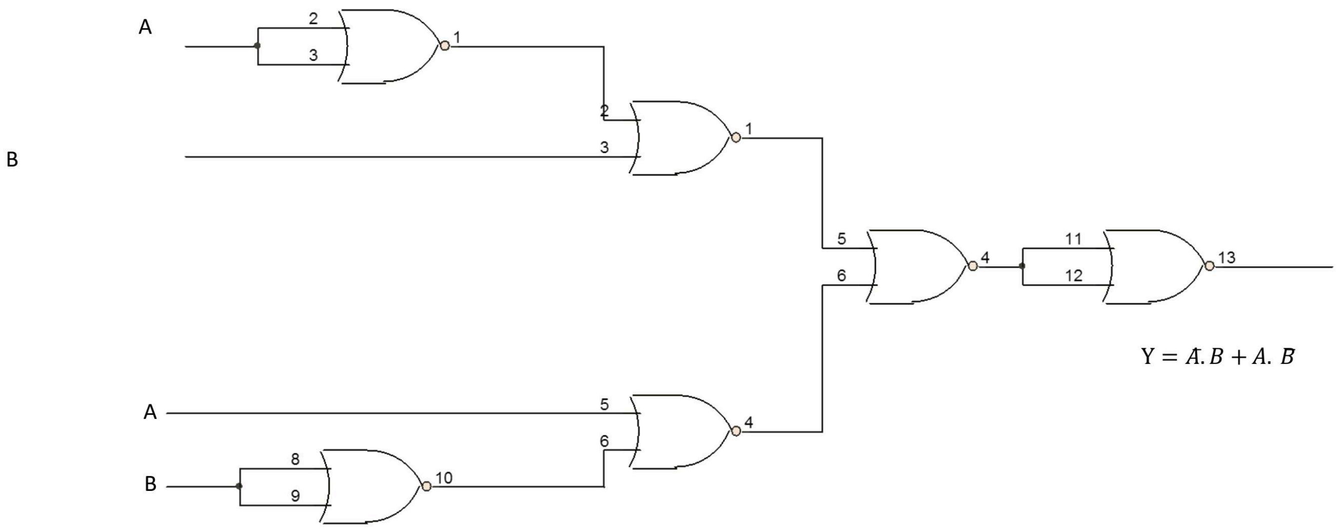
$$Y = \overline{(A + \bar{B})(\bar{A} + B)}$$

$$Y = \overline{(A + \bar{B})} + \overline{(\bar{A} + B)}$$

Taking double inversion, we get

$$Y = \overline{\overline{\overline{(A + \bar{B})} + \overline{(\bar{A} + B)}}}$$

This is required expression for EX-OR gate using NOR gate.



$Y = \bar{A}.B + A.\ \bar{B}$

Conclusion:-

All the gates are realized using NAND and NOR gates and truth tables are verified.

Experiment No. 4

AIM: - Implementation of half subtractor and Full subtractor using logicgates.

APPARATUS REQUIRED

1.IC 7486, IC 7432, IC 7408, IC 7404, IC 7400.

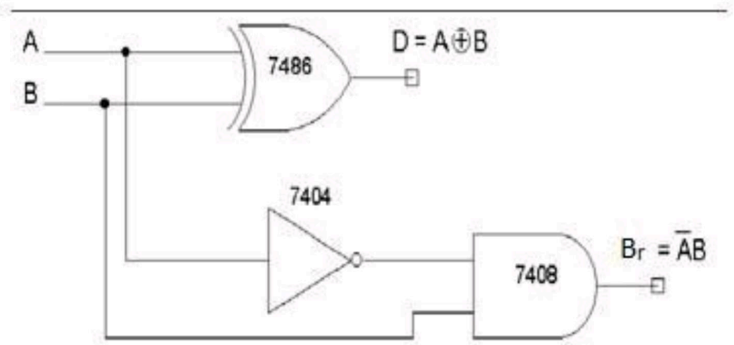
2. BreadBoard.

THEORY:

HALF SUBTRACTOR: Subtracting a single-bit binary value B from another A (i.e. A - B) produces a difference bit D and a borrow out bit B-out. This operation is called half subtraction and the circuit torealize it is called a half subtractor. The Boolean functions describing the half Subtractor are:

$D = A \oplus B$ $B_r = \bar{A} B$

Using X – OR and Basic Gates (a)Half Subtractor



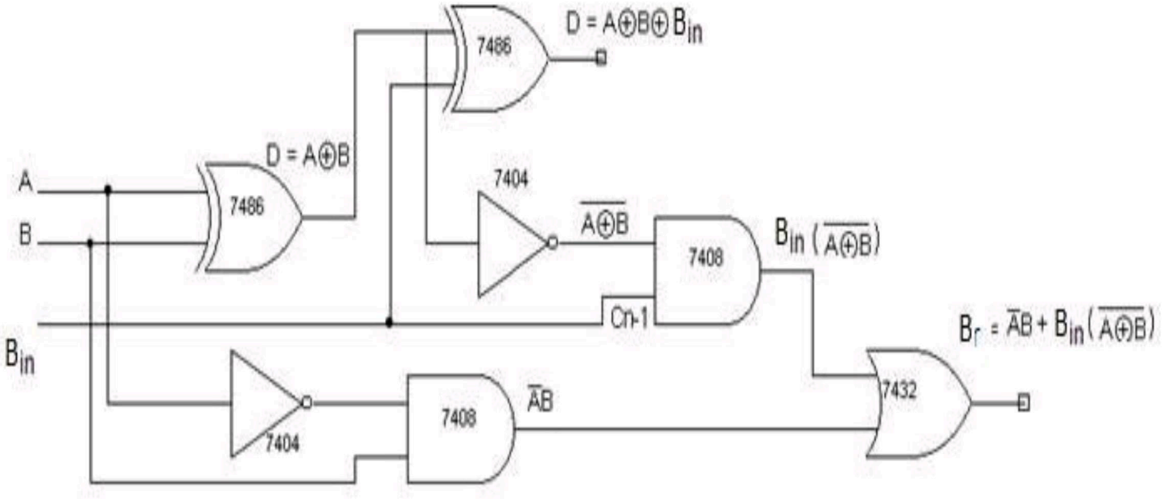
Truth table : Half Subtractor

Half Subtractor			
A	**B**	**D**	**B_r**
0	0		
0	1		
1	0		
1	1		

FULL SUBTRACTOR: Subtracting two single-bit binary values, B, Cinfrom a single-bit value A produces a difference bit D and a borrow out Br bit. This is called full subtraction. The Boolean functions describing the full-subtracter are:

$$D = (x \oplus y) \oplus Bin \qquad Br = \overline{A}B + \overline{A}\,(Bin) + B\,(Bin)$$

Using X – OR and Basic Gates (b) Full Subtractor



7486
$D = A \oplus B \oplus B_{in}$
$D = A \oplus B$
A
B
7486
7404
$\overline{A \oplus B}$
7408
$B_{in}\,(\overline{A \oplus B})$
Cn-1
$B_r = \overline{A}B + B_{in}(\overline{A \oplus B})$
B_{in}
7404
7408
$\overline{A}B$
7432

Truth table :- Full subtractor

Full Subtractor				
A	**B**	**B_{in}**	**D**	**B_r**
0	**0**	**0**		
0	**0**	**1**		
0	**1**	**0**		
0	**1**	**1**		
1	**0**	**0**		
1	**0**	**1**		
1	**1**	**0**		
1	**1**	**1**		

Procedure: -

1. Verify the gates.
2. Make the connections as per the circuit diagram.
3. Switch on VCC and apply various combinations of input according to thetruth table.
4. Note down the output readings for half and full subtractor difference and borrow bit for different combinations of inputs.

Conclusion: -

Half subtractor and full subtractor are constructed and their truth tables areVerified.

Experiment No. 5

BCD TO 7-SEGMENT DECODER/DRIVER

AIM: To set up and test a 7-segment static display system to display numbers 0 to 9.

LEARNING OBJECTIVE:

- To learn about various applications of decoder
- To learn and understand the working of IC 7447
- To learn about types of seven-segment display

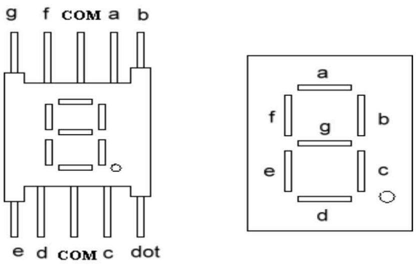
COMPONENTS REQUIRED:

IC7447, 7-Segment display (common anode), Patch chords, resistor (1KΩ) & IC Trainer

Kit THEORY:

The Light Emitting Diode (LED) finds its place in many applications in these modern electronic fields. One of them is the Seven Segment Display. Seven-segment displays contains the arrangement of the LEDs in "Eight" (8) passion, and a Dot (.) with a common electrode, lead (Anode or Cathode). The purpose of arranging it in that passion is that we can make any number out of that by switching ON and OFF the particular LED's. Here is the block diagram of the Seven Segment LED arrangement.

The Light Emitting Diode (LED), finds its place in many applications in this modern electronic fields. One of them is the Seven Segment Display. Seven-segment displays contains the arrangement of the LEDs in "Eight" (8) passion, and a Dot (.) with a common electrode, lead (Anode or Cathode). The purpose of arranging it in that passion is that we can make any number out of that by switching ON and OFF the particular LED's. Here is the block diagram of the Seven Segment LED arrangement.

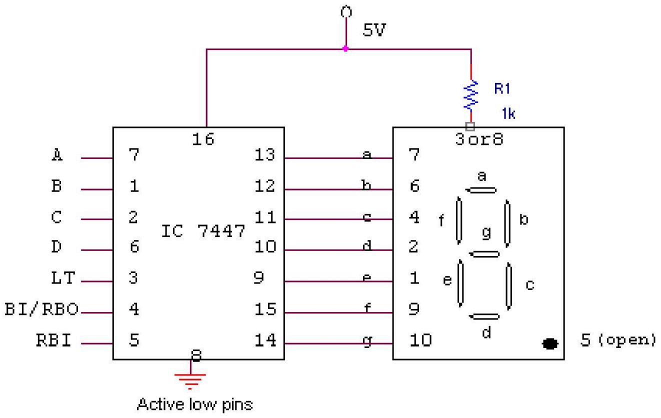


Seven-Segment Display

LED's are basically of two types-
Common Cathode (CC) -All the 8 anode legs uses only one cathode, which is common. Common Anode (CA)-The common leg for all the cathode is of Anode type.

A decoder is a combinational circuit that connects the binary information from 'n' input lines to a maximum of 2^n unique output lines. The IC7447 is a BCD to 7-segment pattern converter. The IC7447 takes the Binary Coded Decimal (BCD) as the input and outputs the relevant 7 segment code.

CIRCUIT DIAGRAM:



TRUTH TABLE:

BCD Inputs				Output Logic Levels from IC 7447 to 7-segments							Decimal number display
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	1	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	1	1	0	0	9

PROCEDURE:

- Check all the components for their working.
- Insert the appropriate IC into the IC base.
- Make connections as shown in the circuit diagram.
- Verify the Truth Table and observe the outputs.

VIVA QUESTIONS:

1. What are the different types of LEDs?
2. Draw the internal circuit diagram of an LED.
3. What are the applications of LEDs?

Experiment No. 6

AIM:To design and implement 4-bit adder and subtractor using 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 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:

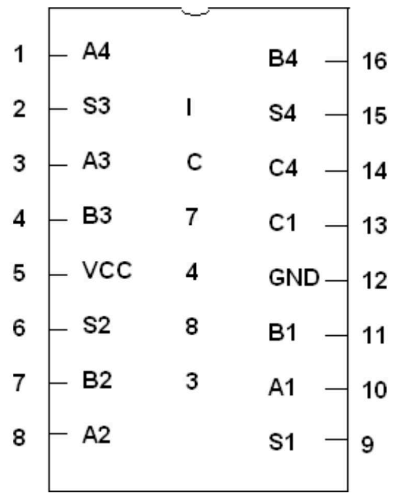
The addition and subtraction operation can be combined into one circuit with one common binary adder. The mode input M controls the operation. When M=0, the circuit is adder circuit. When M=1, it becomes subtractor.

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 decimal digits must be represented in BCD and should appear in the form listed in the columns.

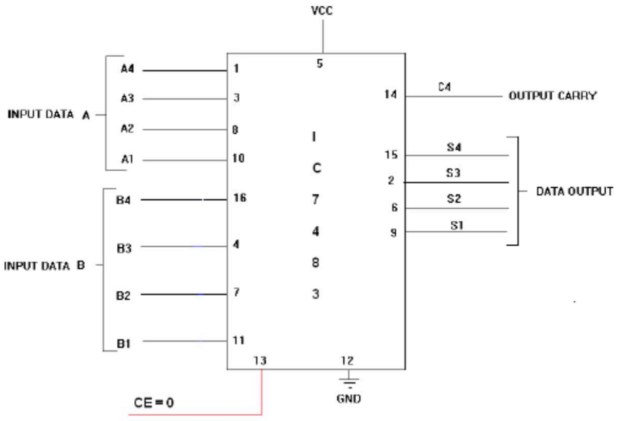
ABCD adder that adds 2 BCD digits and produce a sum digit in BCD. The 2 decimal digits, together with the input carry, are first added in the top 4 bit adder to produce the binary sum.

PIN DIAGRAM FOR IC 7483:



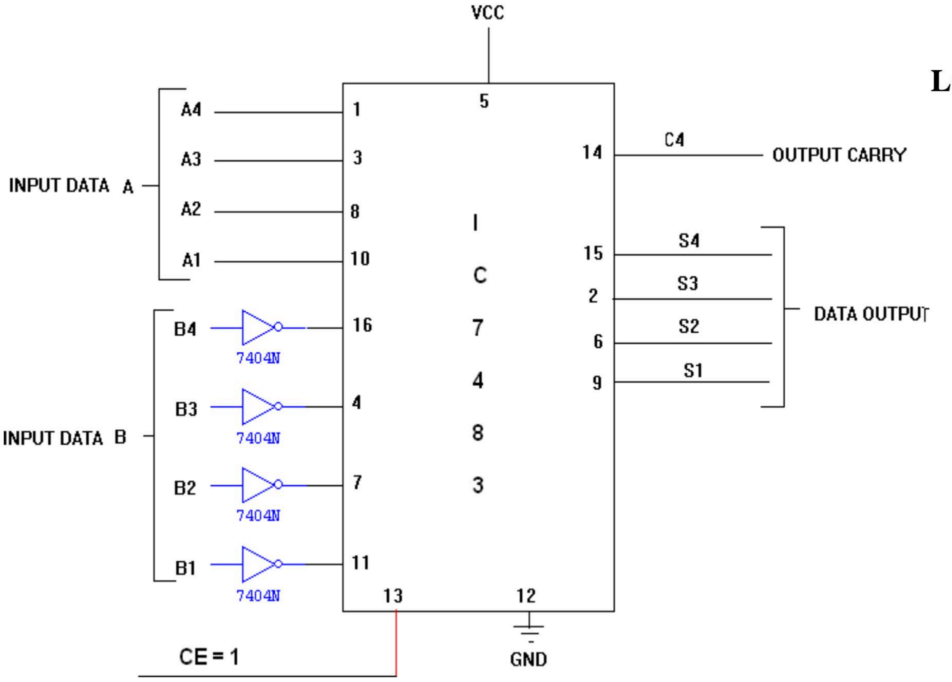
LOGIC DIAGRAM:

4-BIT BINARY ADDER



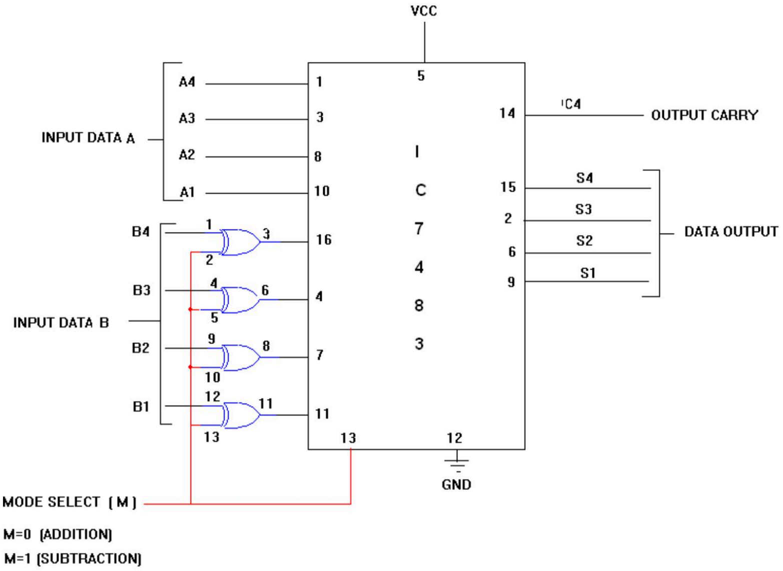
LOGIC DIAGRAM:

4-BIT BINARY SUBTRACTOR



LOGIC DIAGRAM:

4-BIT BINARY ADDER/SUBTRACTOR

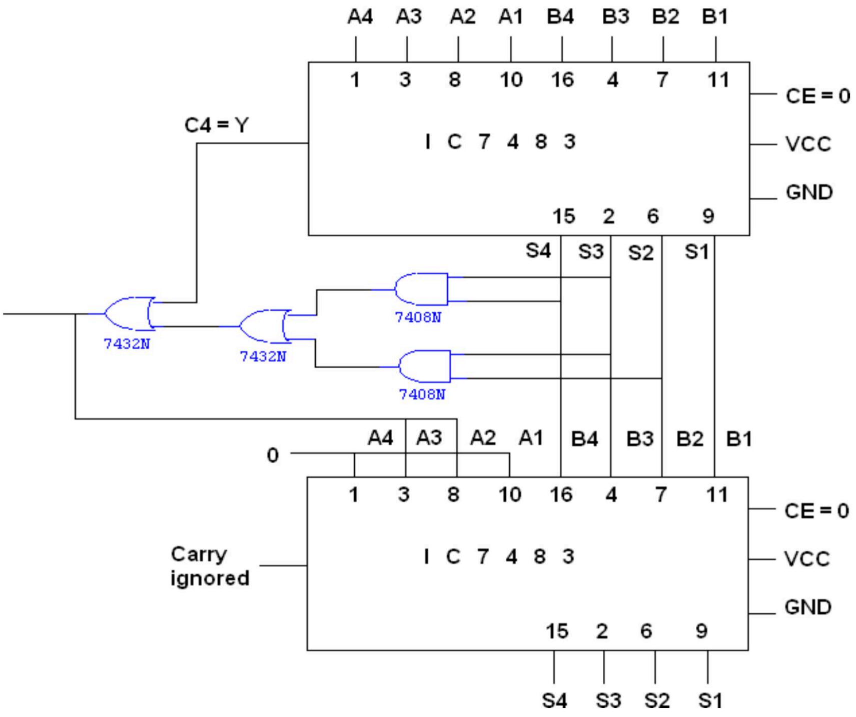


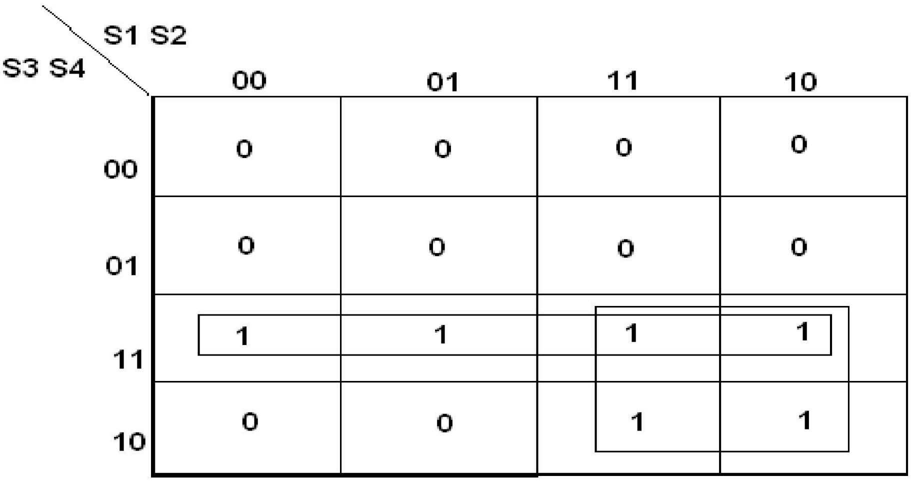
TRUTH TABLE:

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

LOGIC DIAGRAM:

BCD ADDER





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

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

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.

Experiment No: 7

AIM: To design and implement encoder and decoder using logic gates and study of IC 7445 and IC 74147.

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:

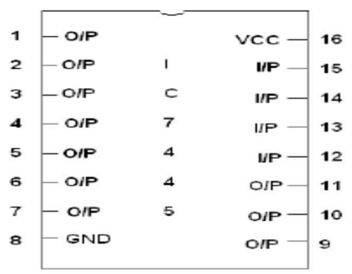
An encoder is a digital circuit that performs inverse operation of a decoder. An encoder has 2^n input lines and n output lines. In encoder the output lines generates the binary code corresponding to the input value. In octal to binary encoder it has eight inputs, one for each octal digit and three output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

DECODER:

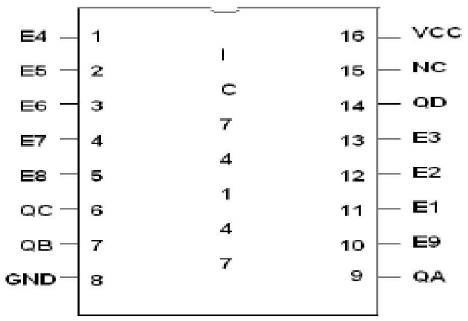
A decoder is a multiple input multiple output logic circuit which converts coded input into coded output where input and output codes are different. The input code generally has fewer bits than the output code. Each input code word produces a different output code word i.e there is one to one mapping 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 output 2^n-1

PIN DIAGRAM FOR IC 7445:

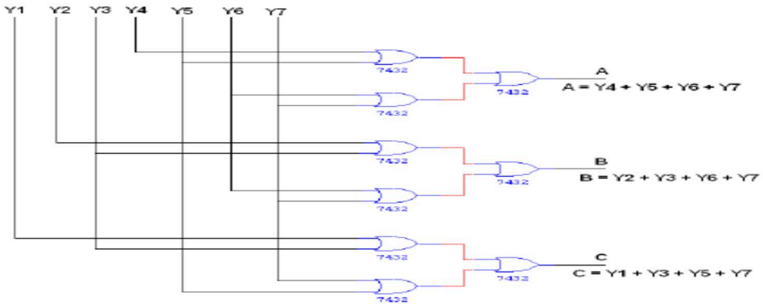
BCD TO DECIMAL DECODER:



PIN DIAGRAM FOR IC 74147:

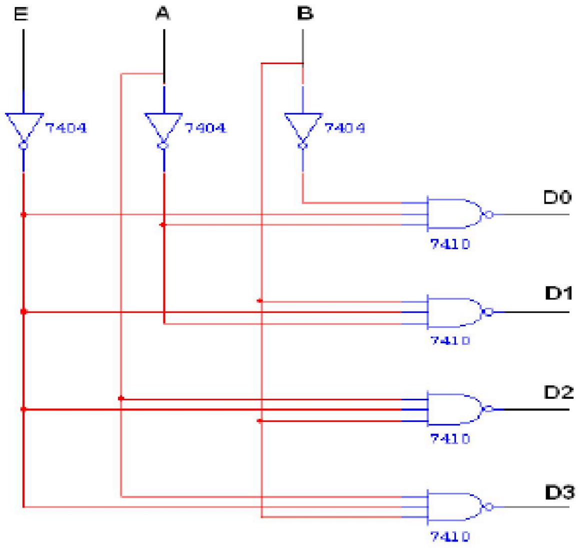


LOGIC DIAGRAM FOR ENCODER

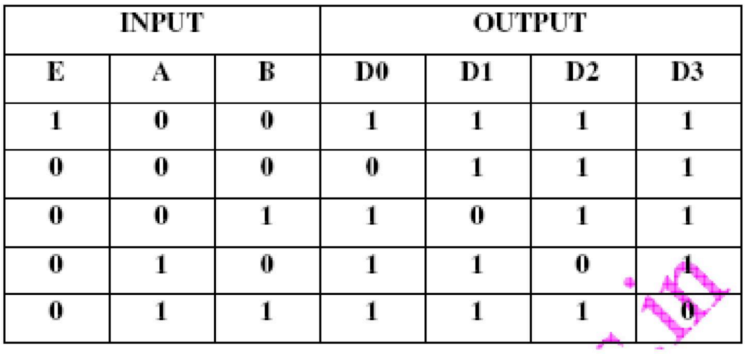


INPUT							OUTPUT		
Y1	Y2	Y3	Y4	Y5	Y6	Y7	A	B	C
1	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	1	1
0	0	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	1	1	1	1

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

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 design and implementation of encoder and decoder using logic gates and study of IC 7445 and IC 74147 were done.

Experiment No. 8

AIM: To design and implement Multiplexer and Demultiplexer using logic gates and study of IC 74150 and IC 74154.

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.	PATCH CORDS	-	

THEORY:

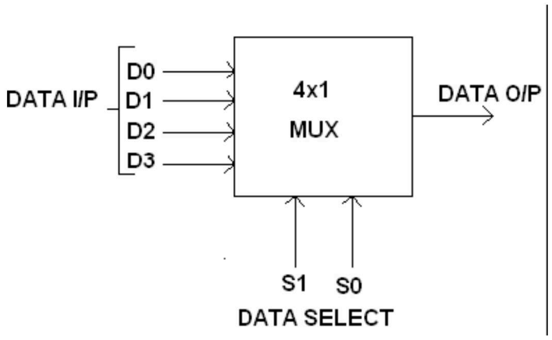
MULTIPLEXER:

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.

DEMULTIPLEXER:

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 4:1 MULTIPLEXER:

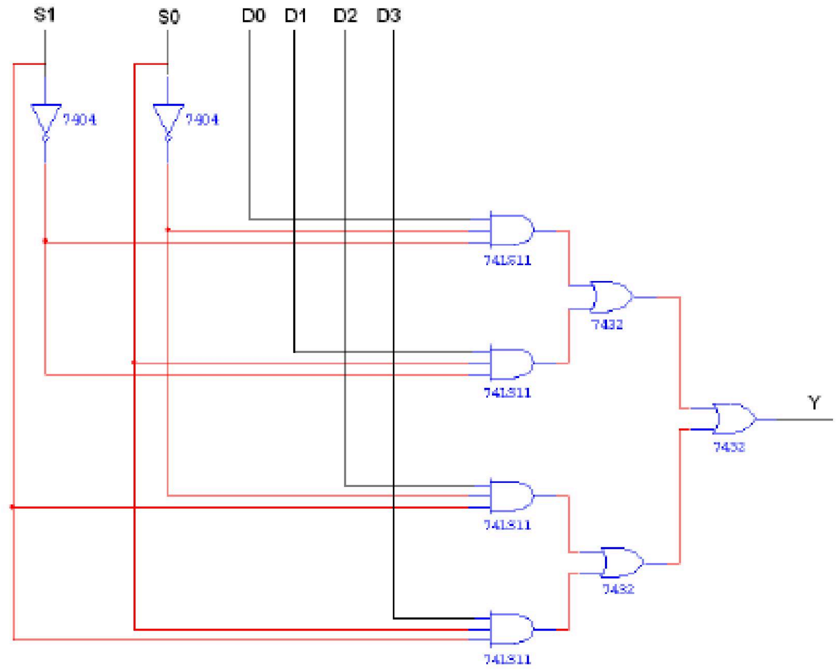


FUNCTION TABLE:

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

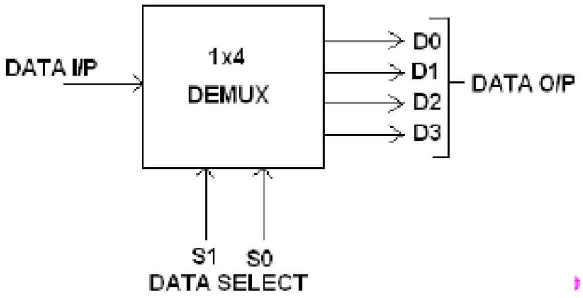
CIRCUIT DIAGRAM FOR MULTIPLEXER:



TRUTH TABLE:

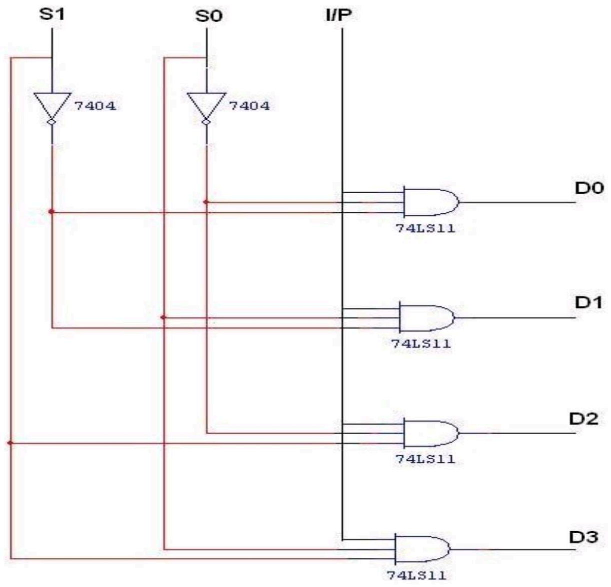
S1	S0	Y = OUTPUT
0	0	D0
0	1	D1
1	0	D2
1	1	D3

BLOCK DIAGRAM FOR 1:4 DEMULTIPLEXER:

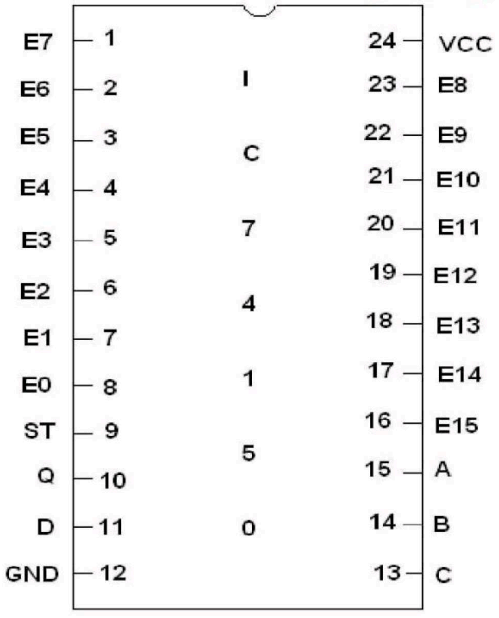


S1	S0	INPUT
0	0	X → D0 = X S1’ S0’
0	1	X → D1 = X S1’ S0
1	0	X → D2 = X S1 S0’
1	1	X → D3 = X S1 S0

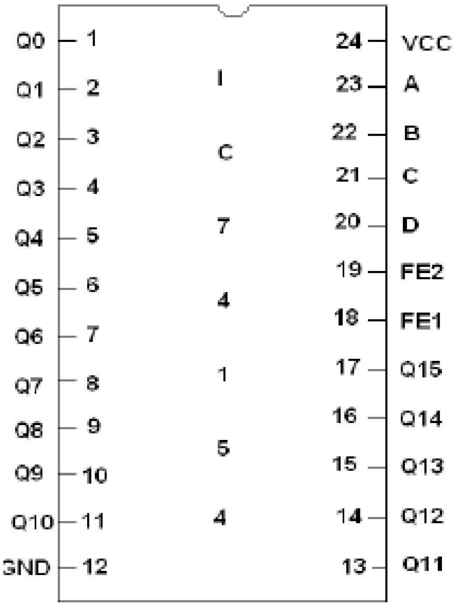
Y = X S1’ S0’ + X S1’ S0 + X S1 S0’ + X S1 S0



PIN DIAGRAM FOR IC 74150:



PIN DIAGRAM FOR IC 74154



TRUTH TABLE:

INPUT			**OUTPUT**			
S1	**S0**	**I/P**	**D0**	**D1**	**D2**	**D3**
0	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**1**	**1**	**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**	**1**	**0**
1	**1**	**0**	**0**	**0**	**0**	**0**
1	**1**	**1**	**0**	**0**	**0**	**1**

PROCEDURE:

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

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

RESULT:

Thus the design and implementation of Multiplexer and Demultiplexer using logic gates and study of IC 74150 and IC 74154 were done