EXPERIMENT No :- 1

AIM :→ Familiarization of AND, OR, NOT, XOR, XNOR, NAND, NOR gates and varification of using truth table.

REQUIRMENT :- Circuit Maker 2000 Software needs to be installed in the Computer.

Theory :-

- **Logic gates :-** Logic gates are the fundamental building blocks of digital System. The name Logic gates is derived from the ability of such a device to make dicisions, in the sense that it produces one output level when some Combination of input levels are present.

Logic gates are electonic circuit because they are made up of a number of electronic devices and Component. They are Constructed in a wide varity of forms. They are usually embedded in large Scale Integrated Circuit (LSI) and very large Scale Integrated Circuit (VLSI) along with a larger number of other device and are not easily accessible or identifiable.

Inputs and outputs of logic gates. Can exist only in two levels. These two levels are terms HIGH and LOW or TRUE and FALSE or ON and OFF or simply 1 and 0.

A table which lists all the possible Combination of input variable and the Corresponding outputs is Called truthtable.

Level logic may be positive logic OR negative logic.
A positive logic system is the one in which the heighen of the two voltage levels represents the logic 1 and the lower of the two voltage levels represent the logic '0'.
A Negative logic System is one in which the lower of the two voltage levels represent the logic '1' and the heigher of two voltage levels represents the logic '0'
In transistor - transistor logic the voltage level are +5v and 0v.

There are three basic types of gates as - AND OR and NOT. and there are two universal gates NAND and NOR.

Application of Gates :-

- Application of OR gates :- whenever the occurrence of any one or more than one event is needed to be detected or some Actions are to be taken after their occurrence in all those cases OR gate can be used.
- Application of AND gates :- Automatic Jam Control System is the one of the application of AND gets. another Application of AND gate is Fire alarm Security System.
- application of NOT gates :- NOT gates are also known as inverter because they invert the output given to them and shows the revers result. Now the CMOS inverter are commonly used to build Square wave oscillators which are used for generating Clock Signal.
- application of X-OR and X-NOR gate :- This type of logic gates are used in generations of parity generation and checking units.
- applications of NAND Gates :- the application of NAND gates are —
  1) Burglar Alarm
  2) Freezer warning buzzer

- Application of NOR Gates :- Logic NOR gates can be used to Construct EX-OR gates and some other real-time application. one of its real-time application is 'Mixer tank'.

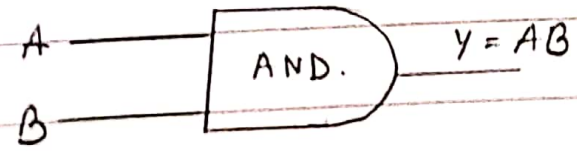
☐ Classification of logic GATES :-

1) AND Gates
2) OR Gates
3) NOT Gates
4) NAND Gates
5) NOR Gates
6) X-OR Gates
7) X-NOR Gates.

☐ Varification of logic gates using truth table :-

- AND Gates :-

Symbol :-

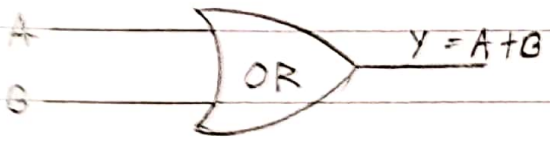


Truth table :-

input		output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR GATES :-

Symbol :-

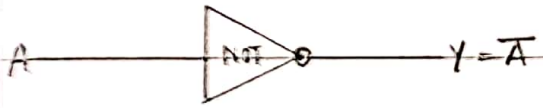


Truth table :-

Input		output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gates :-

Symbol :-



Truth table :-

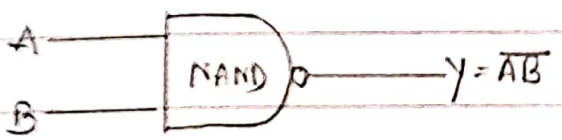
Input	output
A	Y
0	1
1	0

NAND Gates :-

Truth table :-

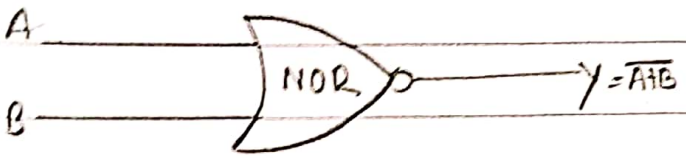
Inputs		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Symbol :-



- NOR Gates :-

Symbol :-

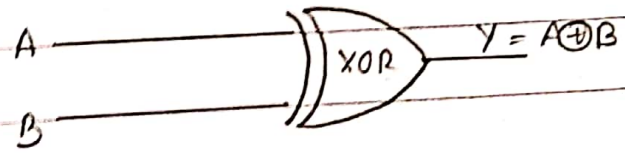


Truth table :-

Input		output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

- X-OR Gates :-

Symbol :-

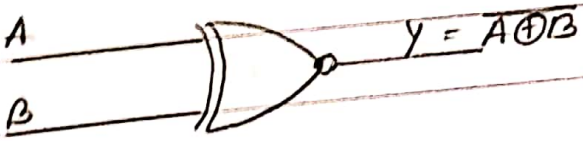


Truth table :-

Input		output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

- X-NOR Gates :-

Symbol



Truth table :-

inputs		outputs
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

• Procedure :-

STEP 1 :- open the Circuit Maker 2000 Software

STEP 2 :- Select the require logic gates under digital Basics > gates

STEP 3 :- Select the logic display under Display > digital > logic display.

STEP 4 :- Select the logic Switch under Switches > digital > logic Switch.

STEP 5 :- use the wire tool (/) to Connect the various devices with each other as required.

STEP 6 :- From menu bar, Select Digital Mode under Simulation menu.

STEP 7 :- Press F10 Key to run the tes

STEP 8 :- Check the outputs of the Test in according with the truth table of the devices

STEP 9 :- Save the project under the file > Save.

• Conclusion :- From the above experiment it is Conclude that using Circuit make 2000 Software we Can Carray out any operation of any type of logic gates. each logic gates has a unique truth table as theire output differ. In Some gates Can't use three input with are X-OR and X-NOR for that's 2XOR gate are used to take 3 inputs in X-OR gates, 2XOR and a inverter is use to take 3 input X-NOR gate

Experiment 2 :-

☐ AIM :→ Familiarization of AND, NOT, OR, X-OR, XNOR using universal gates NAND and NOR.

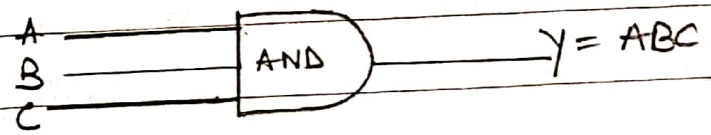
☐ REQUIRMENT :- Circuit maker 2000 Software needs to be installed in the Computer.

☐ THEORY :-

- Basic logic gates :- There are thereee types of basic logic gates — AND, OR and NOT.

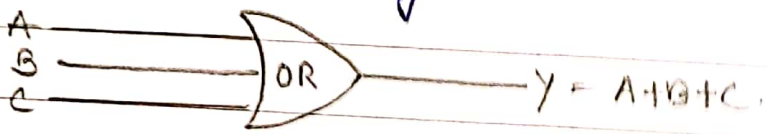
▲ AND Gates :-

An AND gate has two or more input but only one outputs. The outputs assumes the logic 1 state only when each one of its inputs is at logic 1 State. The output assume the logic 0 state even if one of its input at logic 0 State. The AND gates may, therefore be defined as a device whose output is 1 if and only if all its inputs are 1. Hence the AND Gate is also called all or nothing gates.

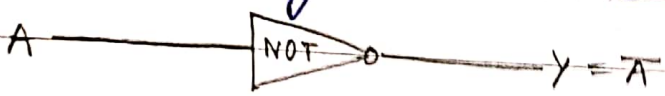


▲ OR Gates :-

An OR gates may have two or more input but only one output. The output assume the logic 1 state even if on of its inputs in the logic 1 state. Its outputs assume the logic 0 state only when each one of its inputs is in logic 0 state. An OR gates may therefore be define as a device whos output is 1 even if one of its inputs in 1. Hence an OR gate is also called an " any or All gates".



▲ NOT Gates :- A NOT GATE also called an inverter, has only one inputs and of Coure only one output it is a device whose output is always the Complement of its Input. That is the output of a NOT gate assume the logic 1 state when its Input in logic '0' state, and assume the output in the logic '0' state, when its input in logic '1' state.

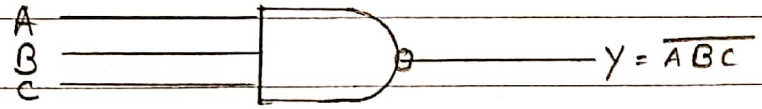


● Universal logic gates :-

There are two universal logic gates —
1) NAND gate. 2) NOR Gate.

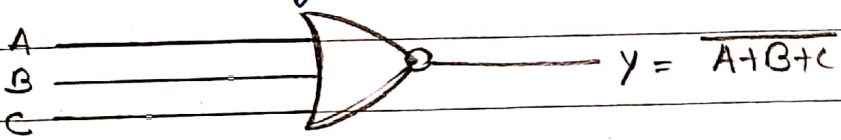
▲ NAND Gates :-

NAND means NOTAND i.e the AND output is TVOTed. So a NAND Grate is a Combination of an AND gates and a NOT gates. The output is logic 0 level, only when each of the inputs in logic '0' level. For any other Combination of inputs the output is a logic 1 level.



▲ NOR Gates :-

NOR means not OR ie the OR output is NOTed. a NOR Grate is Combination of OR gates and a NOT gates. The output is logic ● 1 level, only when each one of its Inputs assume a logic 0 level. for any other Combination of Inputs, the output is a logic '0' Levels.



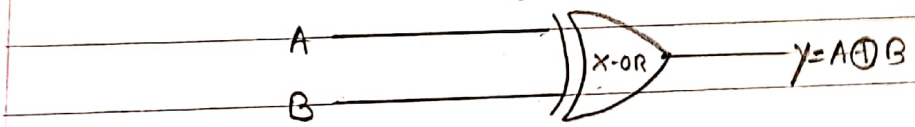
● why Called universal Grate :-

NAND and NOR Grate Call universal gate because all the other gates like AND, OR, NOT, XOR, XNOR Can be derived from it.

- The Exclusive-OR Gate :-

An X-OR gates is a two Input. one output logic Circuit. whose output assume a logic 1 State when one and only one of its two Inputs assume a logic 1 State. under the Condition when both the Inputs assume the logic 0 state or when both the input assume the logic 1 state the output assume the logic '0' State.

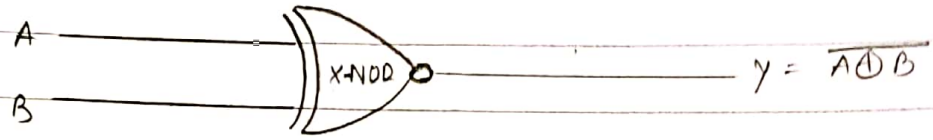
Since an x-OR gate produces an output 1 only when the inputs are not equal. it is called an "anti-coincidence gate" or "inequality detector".



- The Exclusive-NOR Gate :-

An X-NOR gate is a Combination of an X-OR gate and a NOT gate. The X-NOR gate is two Input and produce only one output. whose output assumes a 1 state only both the Inputs assume a '0' State or when both the Inputs assume a 1 State. The output assume a '0' State when 'd' of its Inputs assume a 0 State and other a 1 state. it is also called "Coincidence gate" because its output a 1 only when the Inputs Coincide. it can be used as an "equility

detector because it output a 1 only when its inputs are equal.



Procedure :-

Step 1 :- Open the Circuit maker 2000 Software.

Step 2 :- Select the required logic gate (NAND and NOR) under Digital Basic > Gates.

Step 3 :- Select the logic display under Display > digital > logic display.

Step 4 :- Select the logic Switch under Switches > Digita > logic Switch.

Step 5 :- use the wire tool (+) to connect the various devic with each other as required

Step 6 :- From menu bar Select digital mode under Simulation menu.

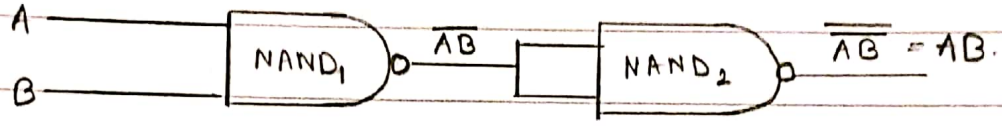
Step 7 :- Press F10 key to run the test.

Step 8 :- Check the output of the test. in accordence withe truth table of the device

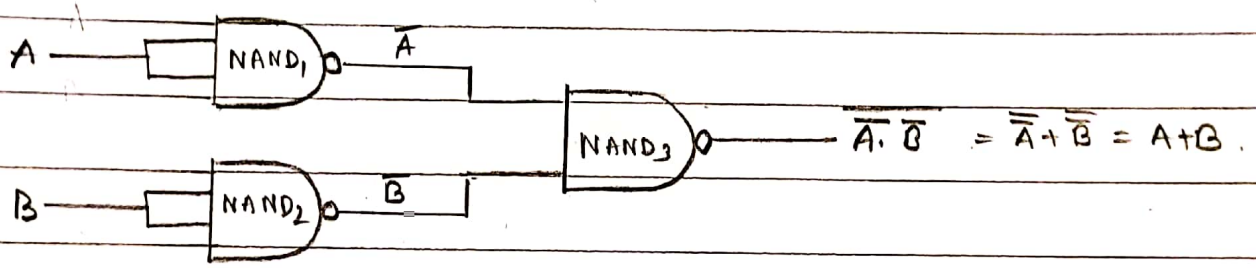
Step 9 :- Save the project under File > Save.

Implementation :-

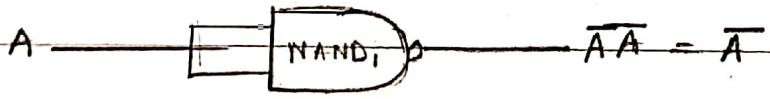
1) AND using NAND :-



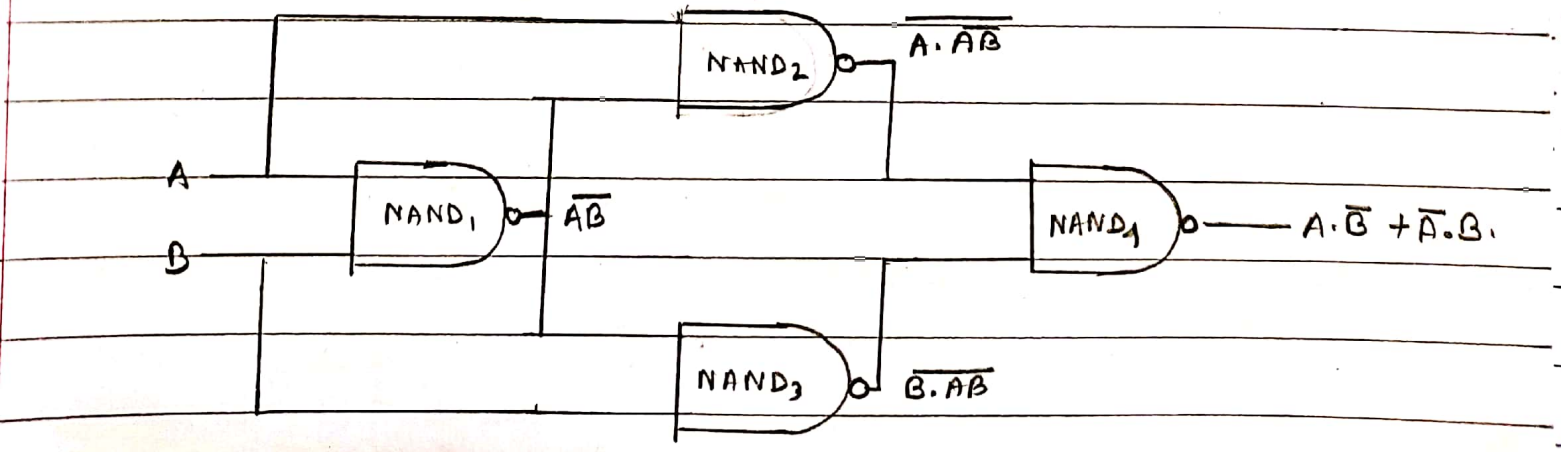
2) OR using NAND :-



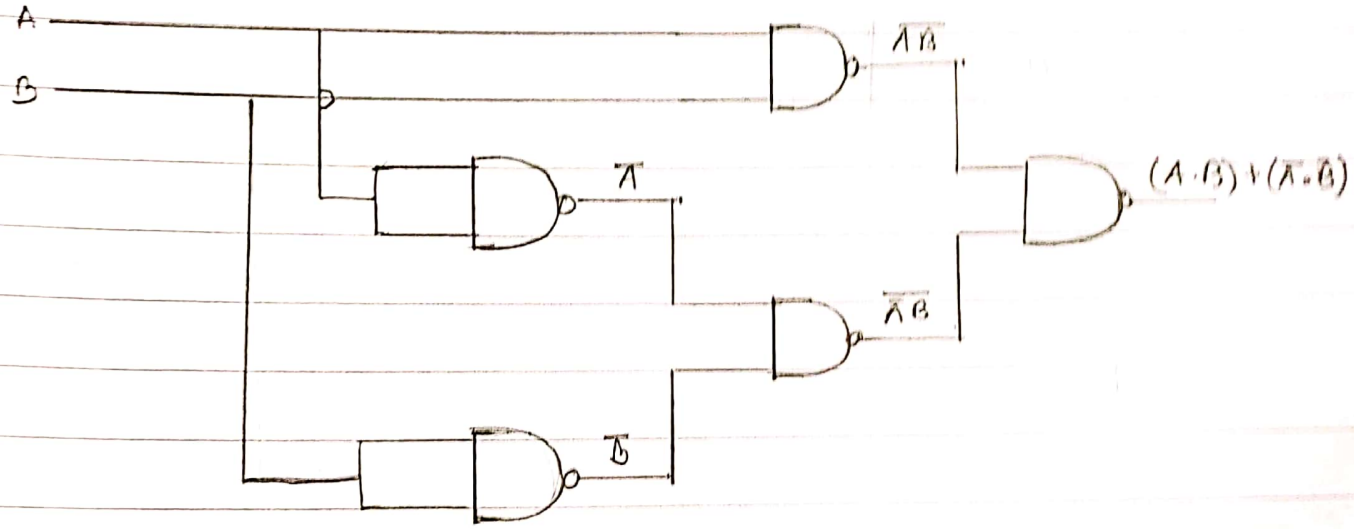
3) NOT using NAND :-



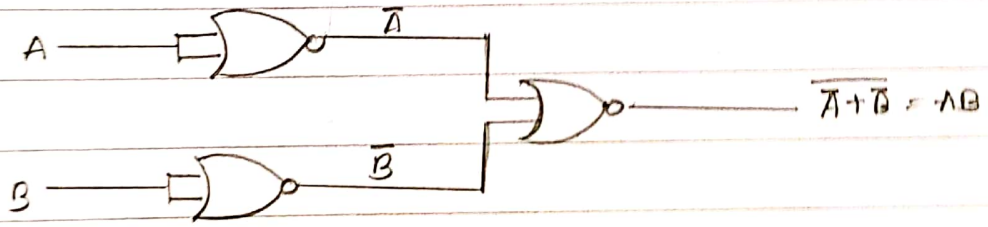
4) X-OR using NAND :-



5) X-NOR using NAND Gate :-

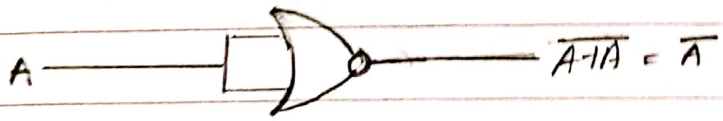


6) AND using NOR :-

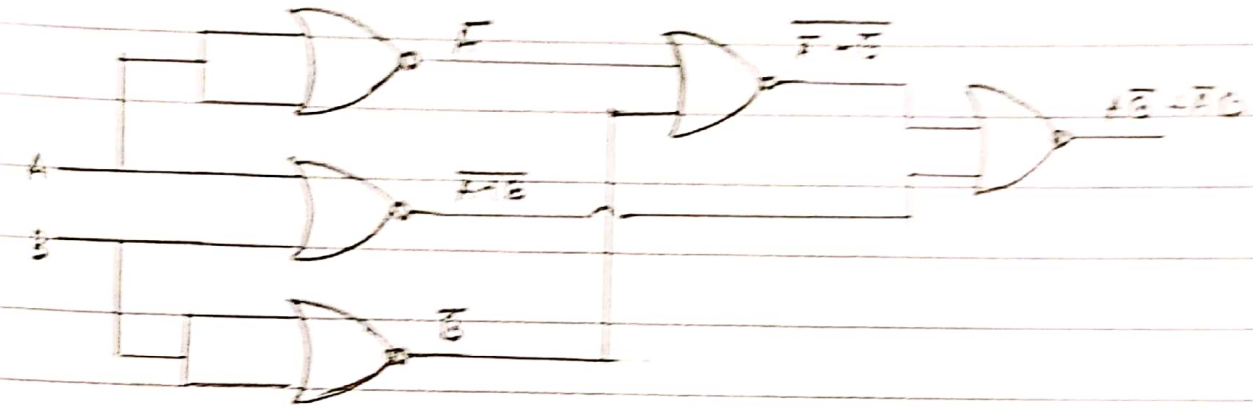


7) OR using NOR :-

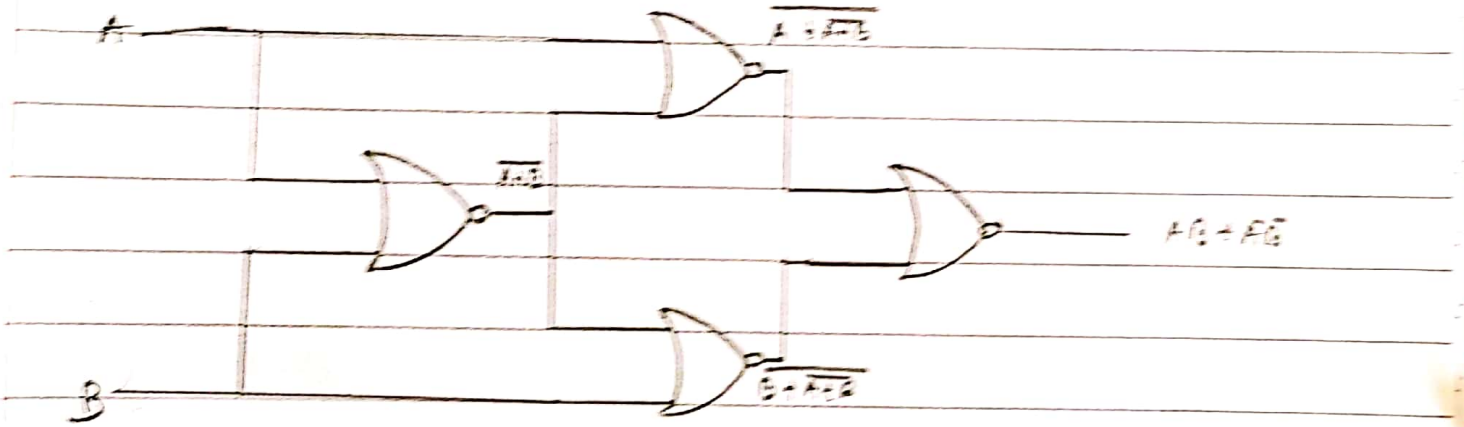
8) NOT using NOR :-



a) X-OR using NOR :-



b) X-NOR using NAND :-



Conclusion :- From the above experiment it is concluded that using Circuit maker 2000 Software. we can carry out any operation of any type of logic gates. In this Software by using the universal gates - NAND and NOR we can make any logic gates like AND - OR - NOT XOR and X-NOR. So it is called universal gates.

Experiment NO 3 :-

AIM :- Realisation and Conversion of the logical expression $x\bar{y}z + \bar{x}y\bar{z} + \bar{x}\bar{y}z$ from AOI Lgical to its equivalent NAND and NOR logic.

Requirment :- Circuit maker 2000 Software needs to be doconloaded.

Theory :- The Complement of a function is to be implemented. So, the designed Circuit Can be implemented using AND and OR gates only called AIO logic. or using AND/ OR/ NOT gates is called AOI Logic.

- Rules for Converting AOI logic to NAND logic :-

1) Draw the Circuit in AOI logic
2) NAND hardware is chosen. Add a circle at the output of each AND gate and at the inputs to all the OR gates.
3) add or subtract an Inverter on each line that recived a circle so that the polarity of signals on those lines remain unchanged. from a that of the original diagram.

4) Replace bubbled OR by NAND.
5) Eleminate Double Inversions.

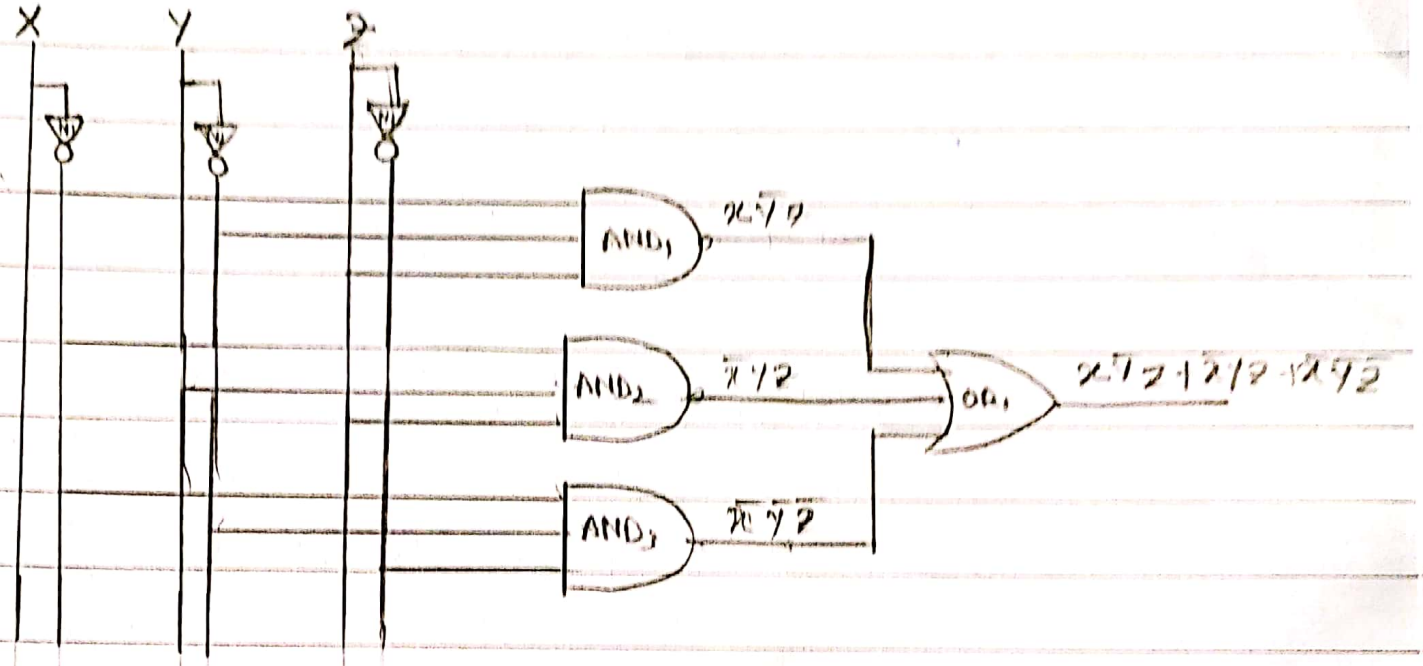
- Rules for Converting AOI logic to NOR logic :-

1) Draw the Circuit in AOI logic.
2) NOR Hardware is chosen. add a circle at the output of each OR gates and at the inputs to all the AND gates.
3) Add or Subtract an inverter on each line that recived a Circle So that the polarity of Signals on those line remains unchanged from that of the original diagram.
4) Replaced bubbled AND by NOR.
5) Eleminate double Inversion.

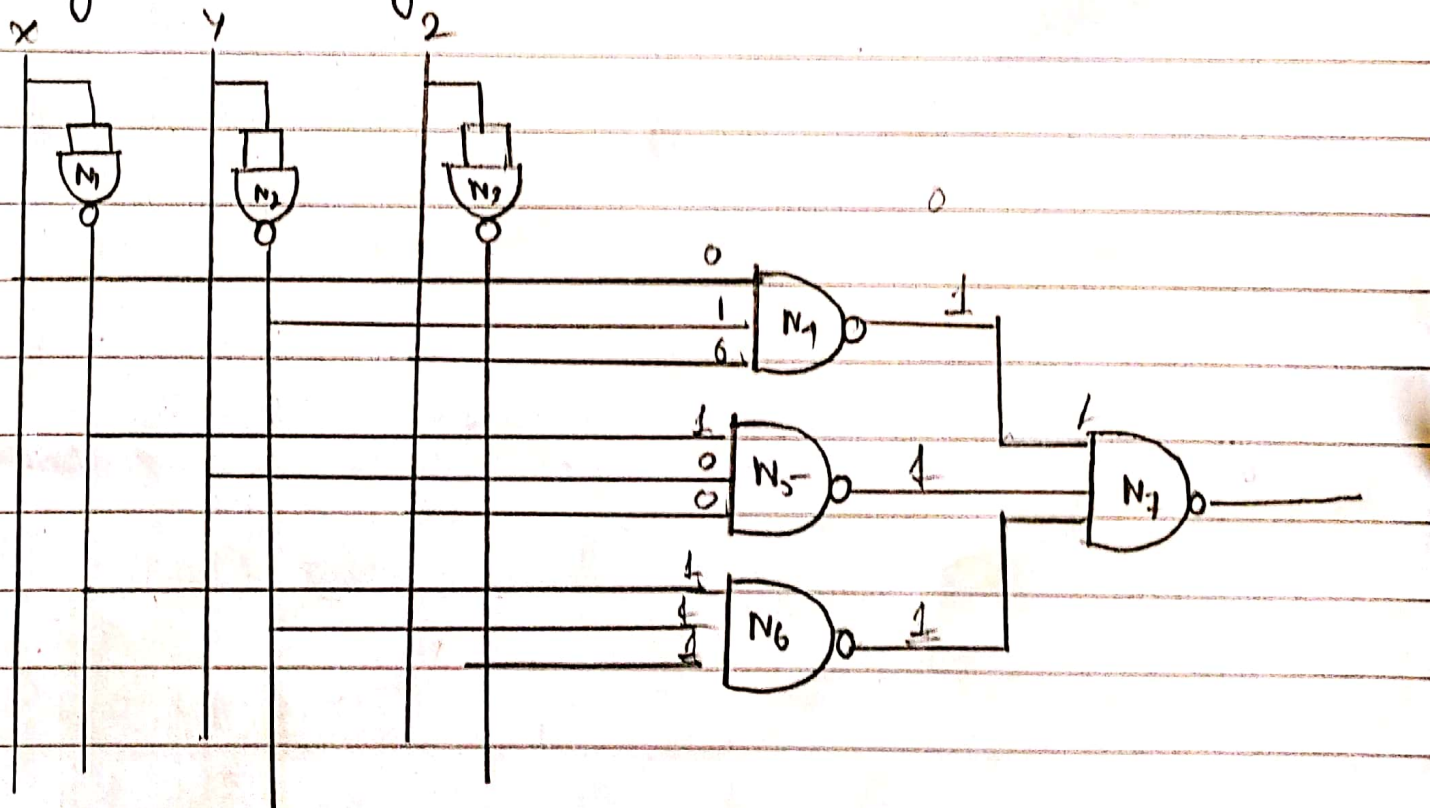
Implementation :-

Logical expression = $x\bar{y}z + \bar{x}yz + \bar{x}\bar{y}\bar{z}$

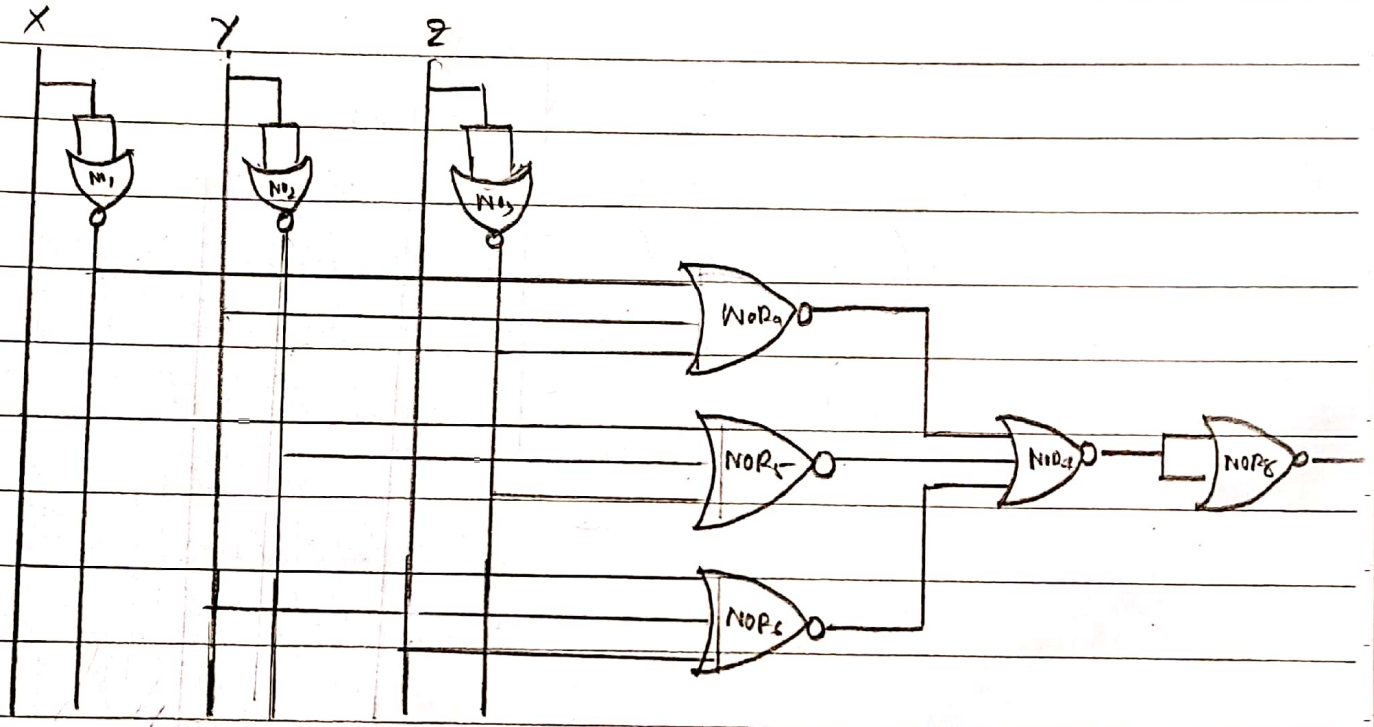
1) using AOI logic :-



Using NAND Logic :-



Using NOR logic :-



Truth table :-

x	y	z	$O_1 = x\bar{y}z$	$O_2 = \bar{x}yz$	$O_3 = \bar{x}\bar{y}\bar{z}$	$O = O_1 + O_2 + O_3$
0	0	0	0	0	1	1
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	1
1	0	0	0	0	0	0
1	0	1	1	0	0	1
1	1	0	0	0	0	0
1	1	1	0	0	0	0

Procedure :-

Step 1 : Open the Circuit maker 2000 Software.

Step 2 : Select either AND, OR, NOT, NAND or NOR under digital Basic > Gates.

Step 3 : Select the logical display under display > digital > logic Display.

Step 4 : Select the logic Switch under Switches > Digital > logic Switch

Step 5 : Use the wire tools (/) to connect the various devices with each other as per required.

Step 6 : From menu bar, Select digital mood under Simulation menue.

Step 7 : Press F10 Key to run the test

Step 8 : Check the output (s) of the test in accordence with the truth table of the device

Step 9 : Save the project under File > Save.

Conclusion :-

From the above experiment it is Concluded that using Circuit Maker 2000 Software. we Can Carry out any operation of any types of logic gates. In this Software by using AOI logic, NAND and NOR logic we Can make any diagram.

Experiment NO 1 :-

AIM :- Design and varification of D-morgan's theorem.

Requirment :- Circuit maker 2000 Software needs to be Installed in the Computer.

Theory :-

1] De Morgan's Theorem :-

De Morgan's Theorem represents two of the most powerful law in Boolean algebra.

Law 1 :- $\overline{A+B} = \overline{A} \cdot \overline{B}$

This law state that the Complement of a Sum of variables is equal to the product of their Individual Complement. what it means is that the Complement of two or more variables ORed together is the Same as the AND of the Complements of each of the Individual variables. This law Can be extended to any number of variables or Combination of variables.

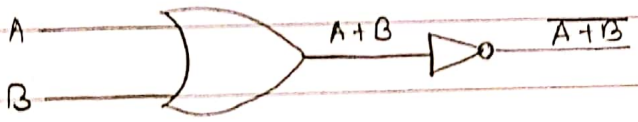
Law 2 : $\overline{AB} = \overline{A} + \overline{B}$

This law state that the Component of the product of variables is equal to the Sum of their Individual Complements. That is, the Complement of two or more variables ANDed together, is equal to the Sum of the Complements of each of the Individual variable. This law can be extended to any number of variables or Combination of Circuit.

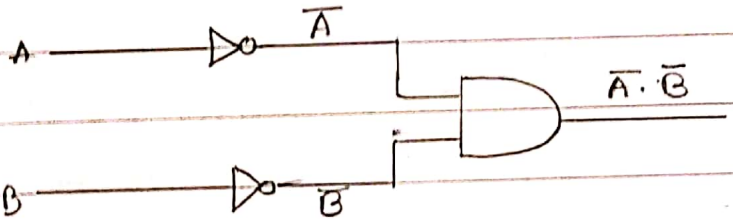
Implementation :-

Law 1 : $\overline{A+B} = \overline{A} \cdot \overline{B}$

LHS :



RHS :



Truth table :-

LHS:

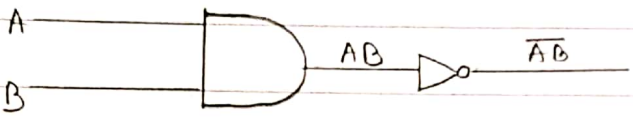
A	B	A+B	$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

RHS:

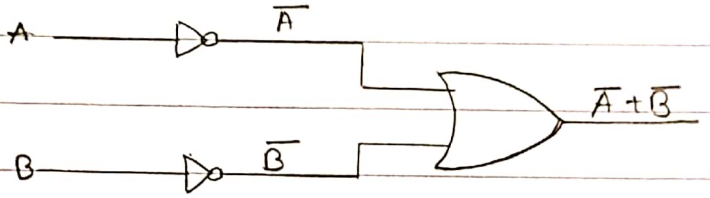
A	B	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Law 2 : $\overline{AB} = \overline{A} + \overline{B}$

LHS :



RHS :



Truth Table :

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

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

Procedure :-

Step 1 : open the Circuit maker 2000 Software

Step 2 : Select the required logic gates under digital basic > gates

Step 3 : Select the logic display under Display > digital > logic display

Step 4 : Select the logic Switch under Switches > digital > logic Switch

Step 5 : use the wire tools to Connect the various devices with each other as per [redacted] requirment

Step 6 : From menu bar Select digital mode under Simulation menu

Step 7 : Press F10 key to run the test.

Step 8 : Check the output of the test in accordance with the truth table of the device.

Step 9 : Save the project under File) Save.

Conclusion : From the above experiment. It is Concluded that using Circuit Maker 2000 Softwar, we can Carry out the De-Morgan's theorem. In this Software by using AOI logic the De-Morgan's Theorem Is satisfied.

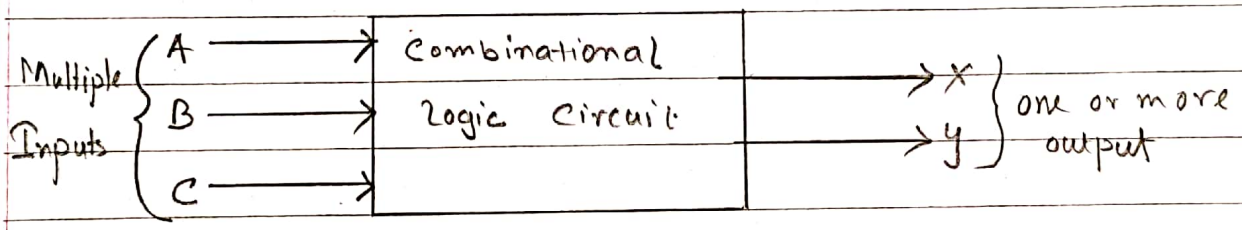
Experiment NO 5 :-

AIM :- Design and implement of Half adder, Full adder and Half Subtractor, Full Subtractor.

Requirement :- Circuit maker 2000 Software needs to be installed in the Computer.

Theory :-

About Combination logic Circuit :-



- Combinational Logic Circuit are memoryless digital logic Circuit whose output at any instant in time depends only on the Combination of its Input.

- The output of the Combinational logic Circuit are only determined by the logical function of their current Input State logic "0" or logic

Logic '1' at any given instant in time.

- Combinational logic circuit, the output is dependant at all times on the combination of its inputs. Thus a combination circuit is memory-less.

- Combinational logic circuit are made up from basic logic NAND, NOR or NOT gets that are "combined" or connected together to produce more complicated switching circuit.

- The three main ways of specifying the function of combinational logic circuit are :-
  1. Boolean Algebra.
  2. Truth table.
  3. Logic Diagram.

- As combinational logic circuit are made up from individual logic gates only, they can also also be considered as " decision making circuit

- Common combinational circuits made up from individual logic gates that carry out a desired application include, MUX, DMUX, Encoder, Decoder, ful and Half adder etc.

Arithmetic Logic Circuit :-

An Arithmetic Logic Circuit is a Combinational digital electronic circuit that performs arithmetic and bitwise operations on binary numbers.

The types of Arithmetic Circuit are -

1) Half adder
2) full Adder
3) Half Substructor
4) full Substructor.

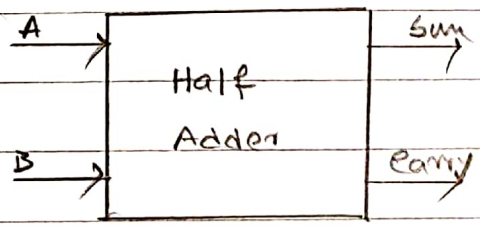
Half adder :-

A half-adder arithmetic circuit adds two binary digit given a sum bit and a Carry bit. The Sum (S) bit and carry (C) bit accordings to rules of binary addition, are given by

Truth table :-

input		output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

logic Symbol :-



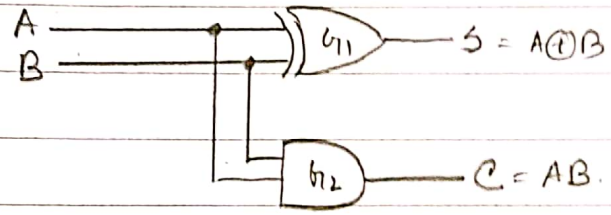
The Sum (S) is the X-OR of A and B therefore,

$$S = A\bar{B} + B\bar{A} = A \oplus B.$$

The Carry (C) is the AND of A and B therefore,

$$C = AB$$

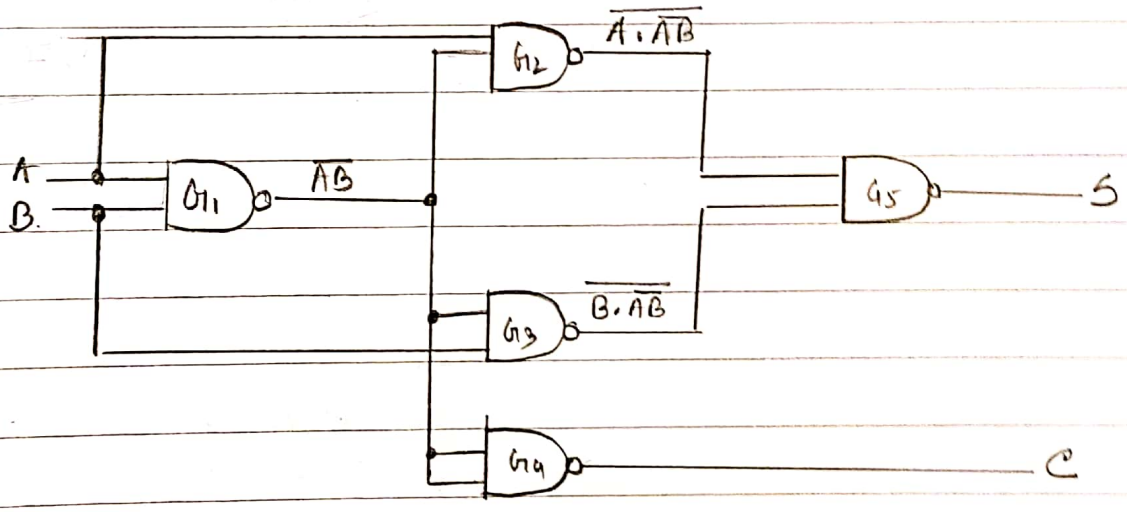
Logic diagram of Half adder :-



NAND logic :-

$$S = A\bar{B} + \bar{A}B$$
$$= A\bar{B} + A\bar{A} + \bar{A}B + B\bar{B}$$
$$= A(\bar{A}+\bar{B}) + B(\bar{A}+\bar{B})$$
$$= A \cdot \overline{AB} + B \cdot \overline{AB}$$
$$= \overline{A \cdot \overline{AB}} \cdot \overline{B \cdot \overline{AB}}$$

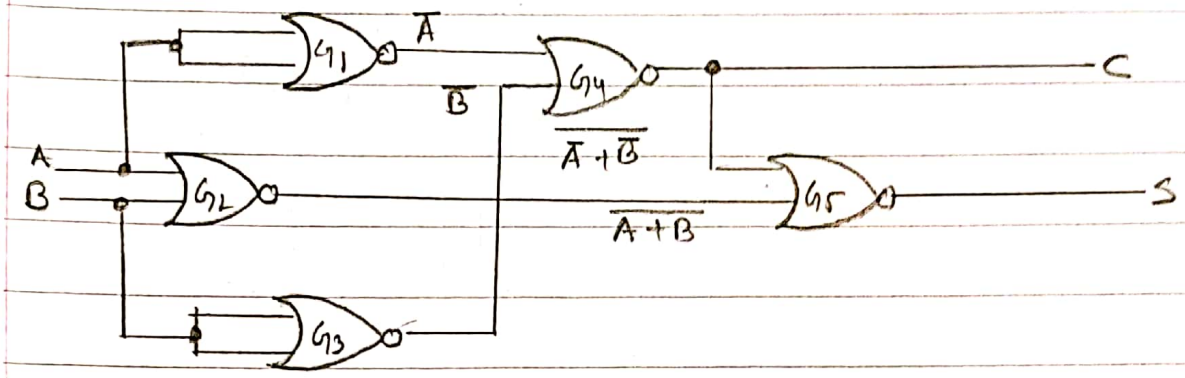
$$C = AB = \overline{\overline{AB}}$$



NOR logic :-

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

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

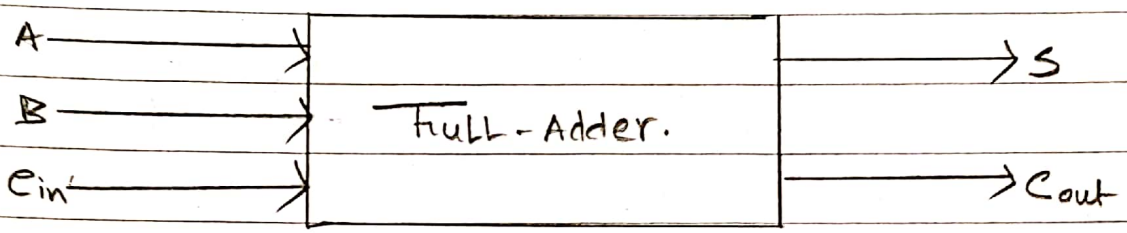


■ The full adder :-

A full adder is an arithmetic circuit that adds two bits and a carry and outputs a Sum bit and a Carry bits when we want to add two binary numbers, each having two or more bits, The LSBs can be added by using a half-adder. The carray resulted from the addition of the LSBs is carried over to the next Significant column and added to the two bits in that Column.

The full-adder adds the bits A and B and the carry from the previous column called the Carry-in Cin and outputs the Sum bit S and the carry bit called the Carry-out Cout.

logic Symbol :- or Block diagram :-



Truth table :-

Input			Sum	Carry
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S = \bar{A}\bar{B}Cin + \bar{A}B\bar{Cin} + A\bar{B}\bar{Cin} + ABCin.$

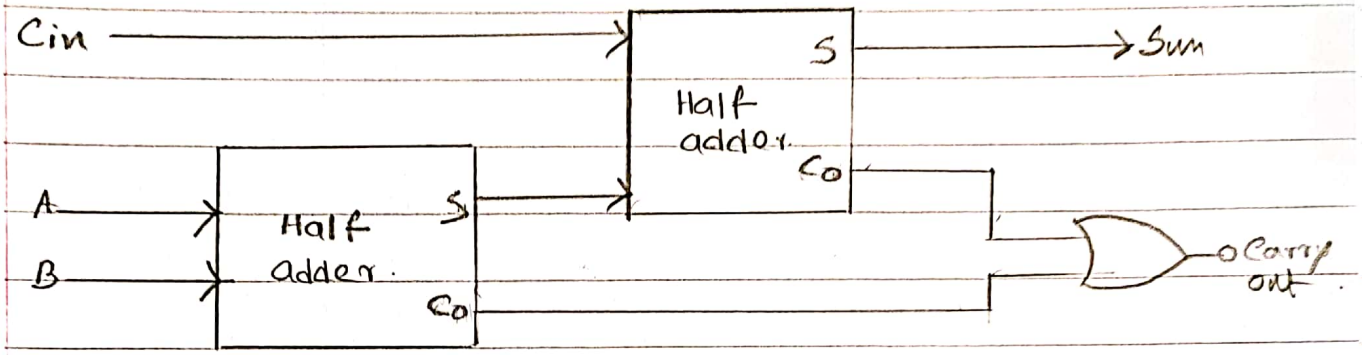
$= A \oplus B \oplus C.$

$Cout = \bar{A}BCin + A\bar{B}Cin + AB\bar{Cin} + ABCin$

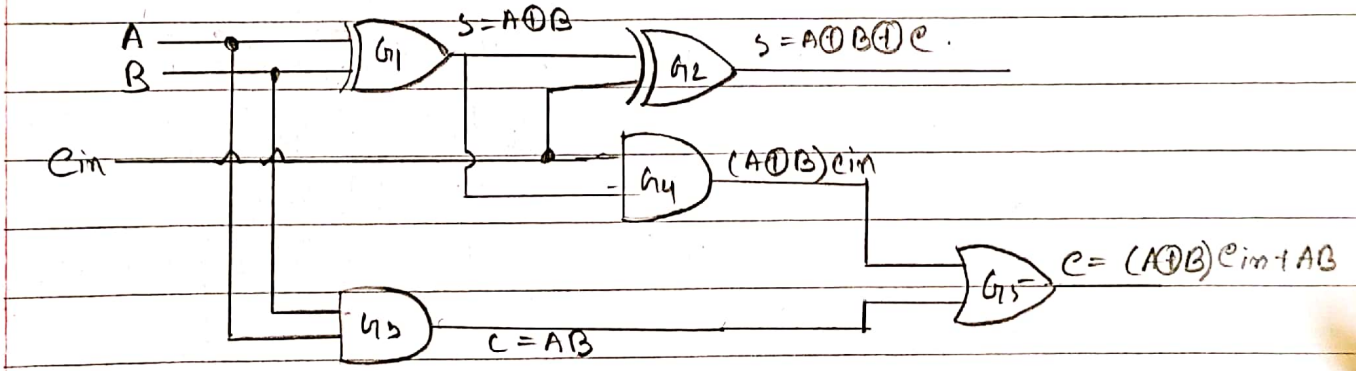
$= AB + (A \oplus B)Cin = AB + ACin + BCin.$

Logic diagram of full adder using two half adder.

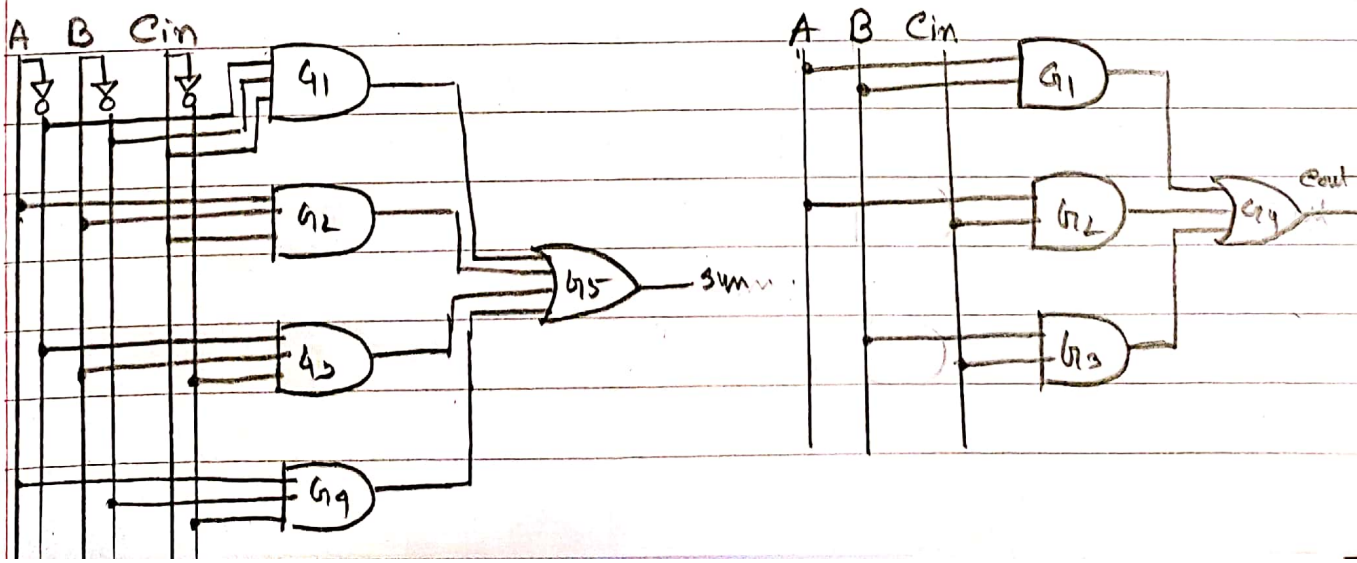
Block diagram :-



logic diagram :-



Sum and carry bits of a full adder using AOI logic.

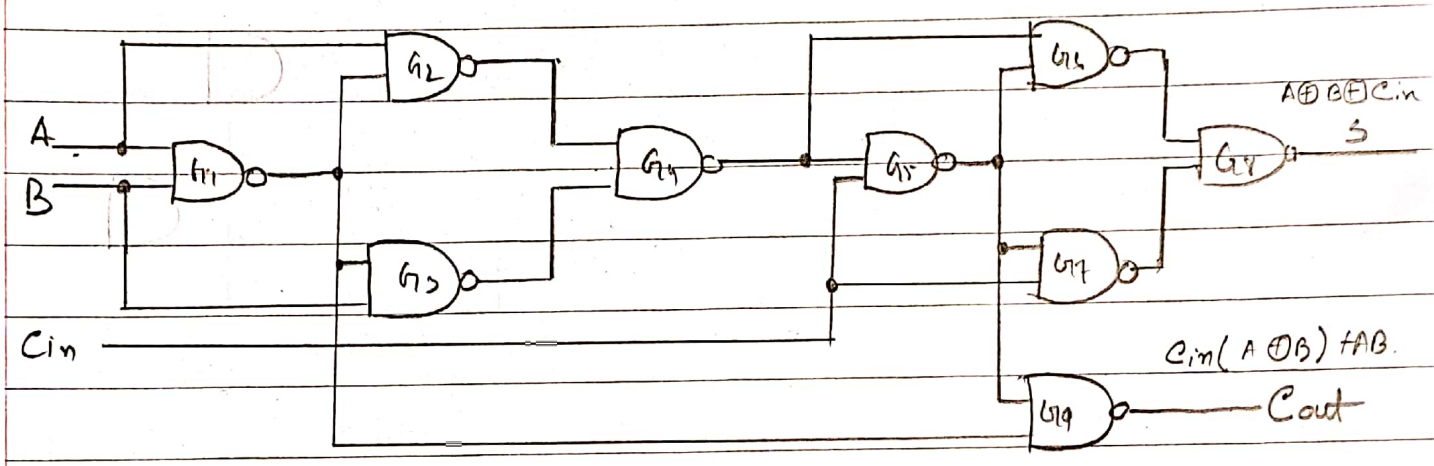


NAND logic :-

$A \oplus B = \overline{\overline{A \cdot \overline{AB}} \cdot \overline{B \cdot \overline{AB}}} = X$ then

$S = A \oplus B \oplus Cin = \overline{\overline{X \cdot \overline{X Cin}} \cdot \overline{Cin \cdot \overline{X Cin}}} = X \oplus Cin$

$Cout = Cin(A \oplus B) + AB = \overline{\overline{Cin(A \oplus B)} \cdot \overline{AB}}$

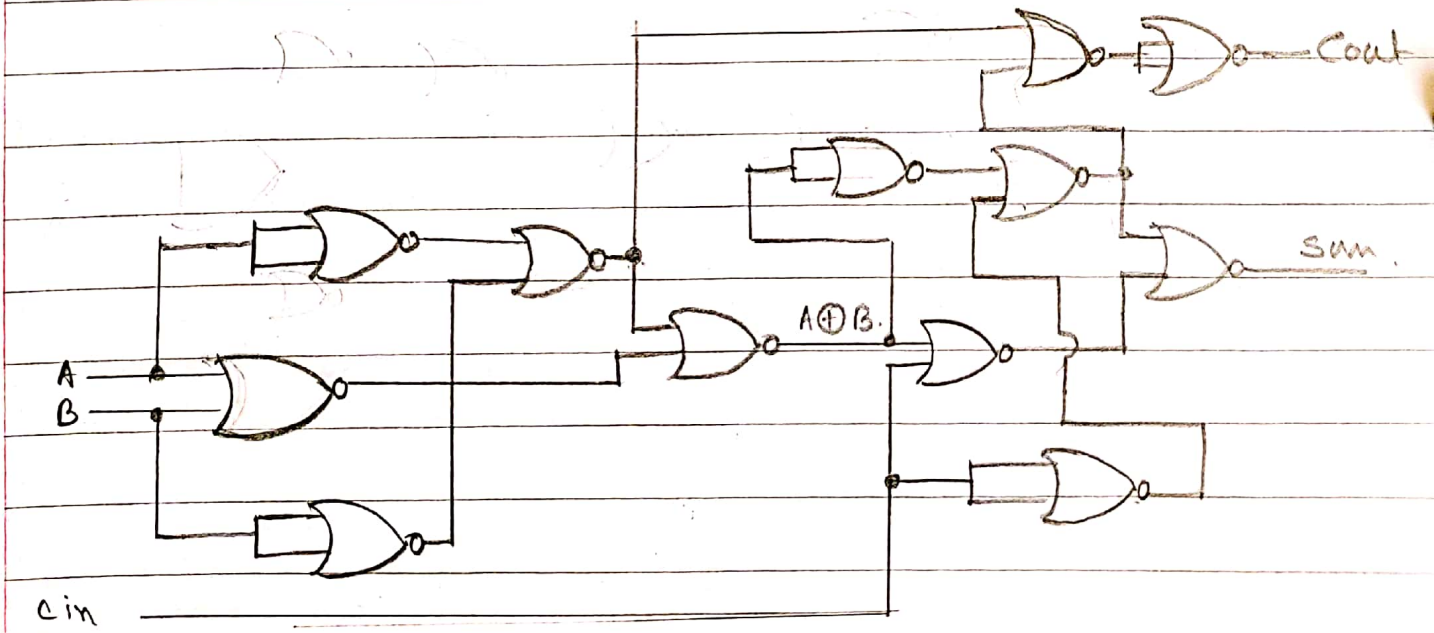


NOR - LOGIC :-

$A \oplus B = X = \overline{(A+B) + \overline{A+B}}$

$S = A \oplus B \oplus Cin = X \oplus Cin = \overline{X + Cin} + \overline{X + \overline{Cin}}$

$Cout = AB + Cin(A \oplus B) = \overline{\overline{A} + \overline{B}} + \overline{\overline{Cin} + \overline{A \oplus B}}$



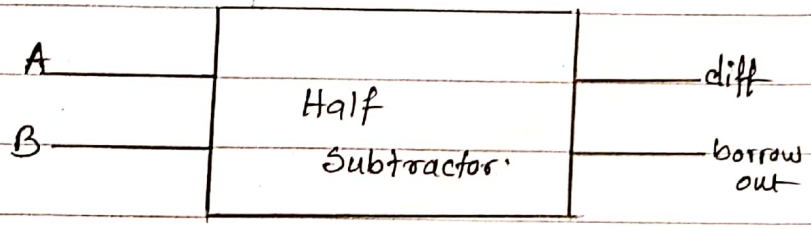
Half - [redacted] SUBTRACTOR :-

A half subtractor is an arithmetic circuit that subtracts one bits from the other. it is used to subtract the LSB of the subtrahend from the LSB of the minuend when one binary number is subtracted from the other.

Truth table :-

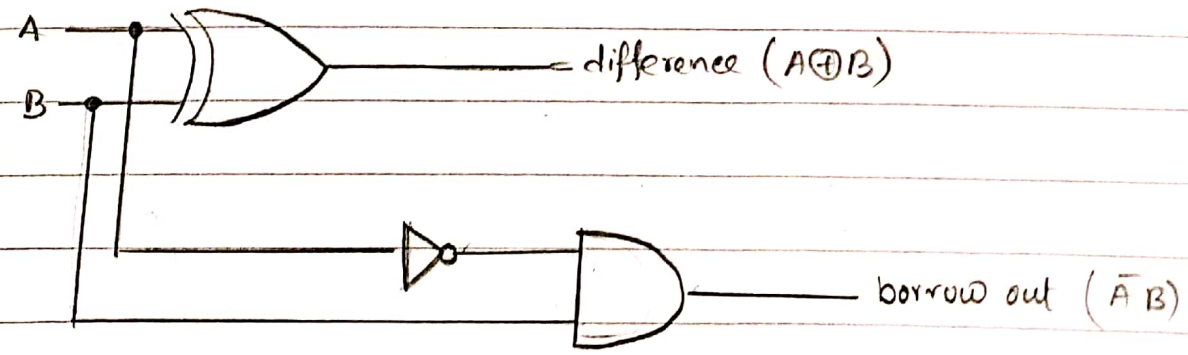
Input		Output	
A	B	d	b
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Block diagram :-

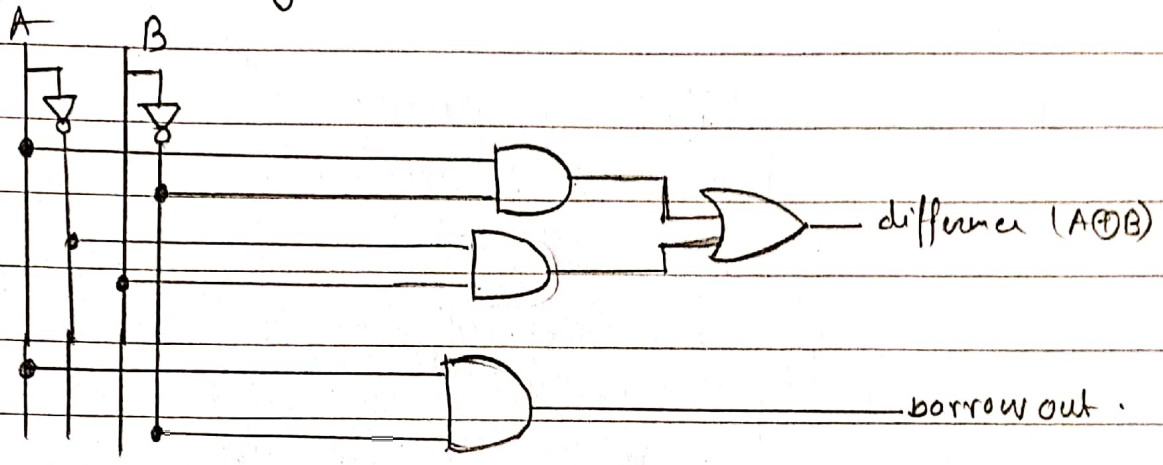


$d = A\bar{B} + B\bar{A} = A \oplus B$ and $b = \bar{A}B$

using X-OR gate with one NOT gate and AND Gate.



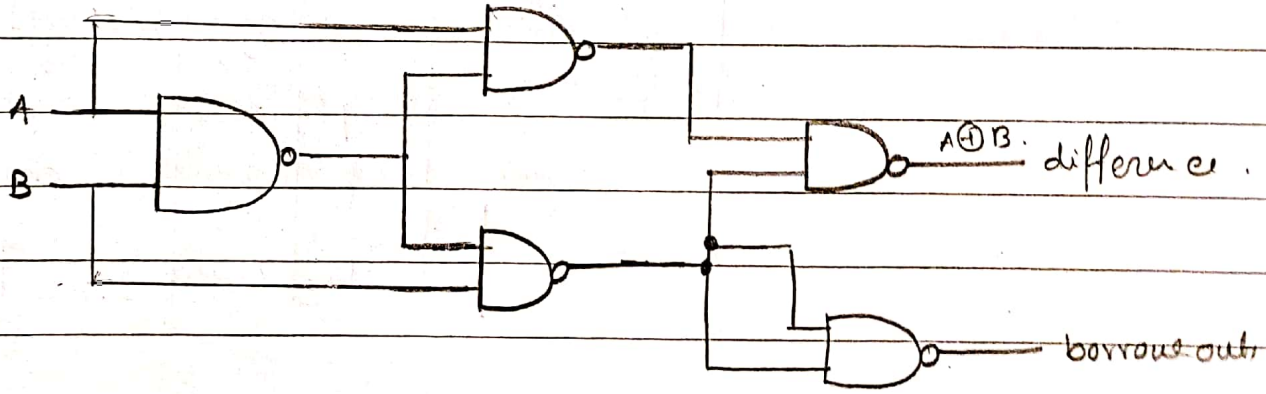
Using AOI logic :-



NAND logic :-

$$d = A \oplus B = \overline{\overline{A \cdot \overline{AB}} \cdot \overline{B \cdot \overline{AB}}}$$

$$b = \overline{A}B = B(\overline{A}+\overline{B}) = B(\overline{AB}) = \overline{B \cdot \overline{AB}}$$



NOR LOGIC :-

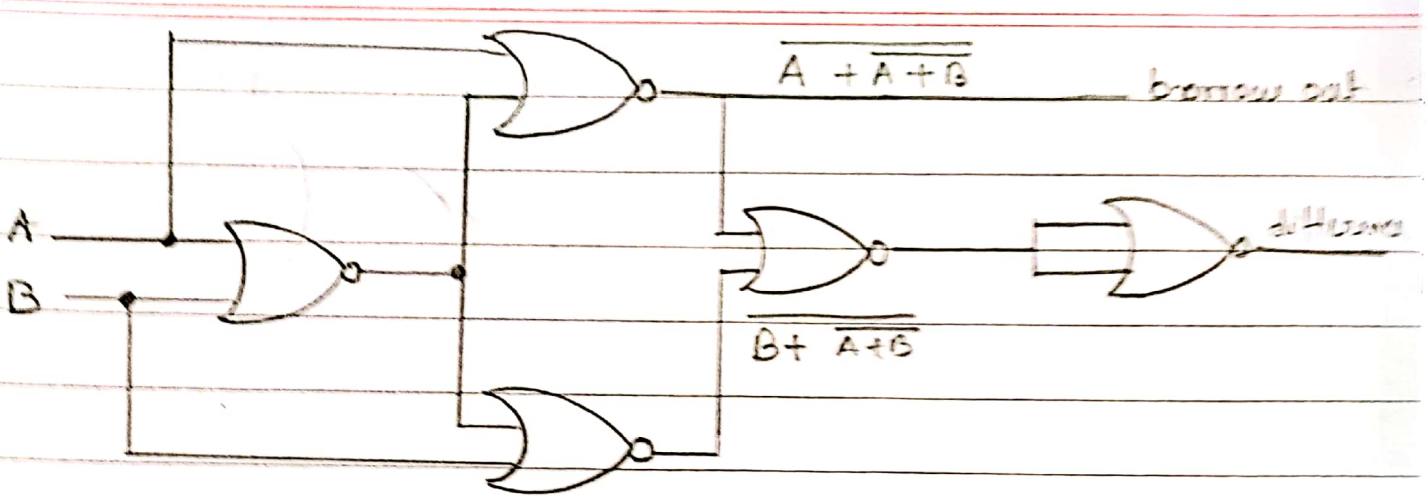
$$d = A \oplus B = A\overline{B} + \overline{A}B = A\overline{B} + B\overline{B} + \overline{A}B + A\overline{A}$$

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

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

$$b = \overline{A}B$$

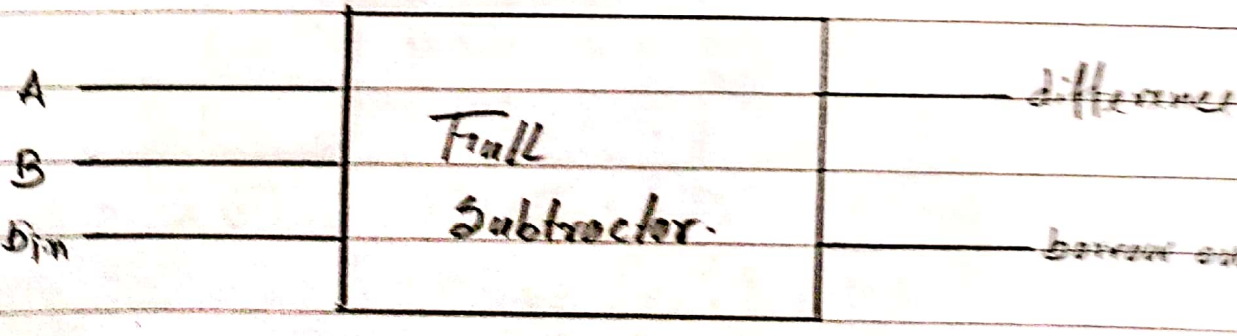
$$= \overline{A}(A+B) = \overline{\overline{A(A+B)}} = \overline{A + [A+B]}$$



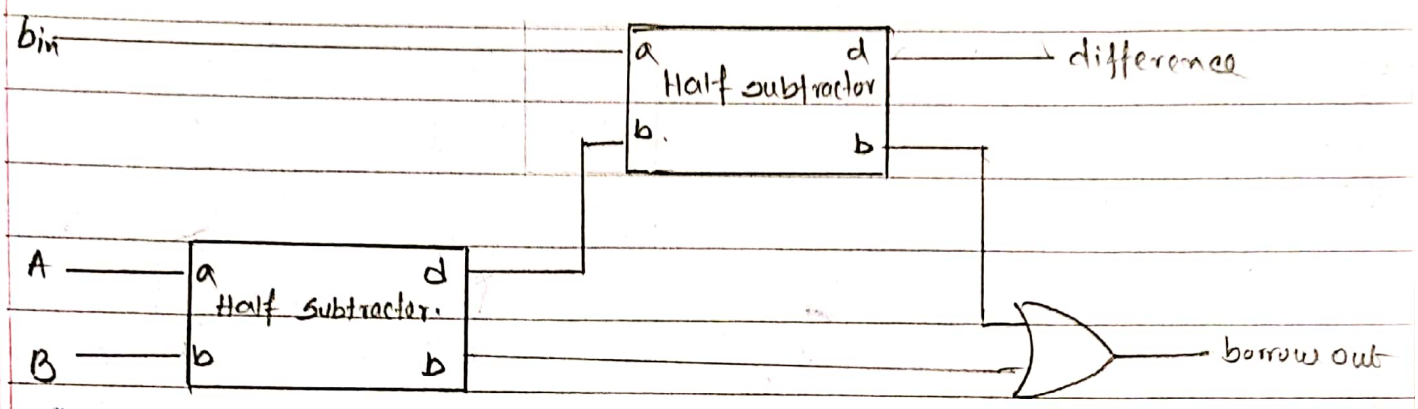
FULL SUBTRACTOR :-

The half-subtractor can be used only for LSB subtraction. if there is a borrow during the first subtraction of the LSB, it affects the subtraction in the next higher column ; the subtrahend bit is subtracted from the minuened bit. Considering the borrow from that column used from the subtraction in the preceding column such a subtraction is performed by full-subtractor.

Block diagram :-



Block diagram :- Using two half subtractor :-



Truth table :-

input			output	
A	B	Bin	difference	borrow out
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$d = \bar{A}\bar{B}\,bin + \bar{A}B\,\overline{bin} + A\bar{B}\,\overline{bin} + AB\,bin$$

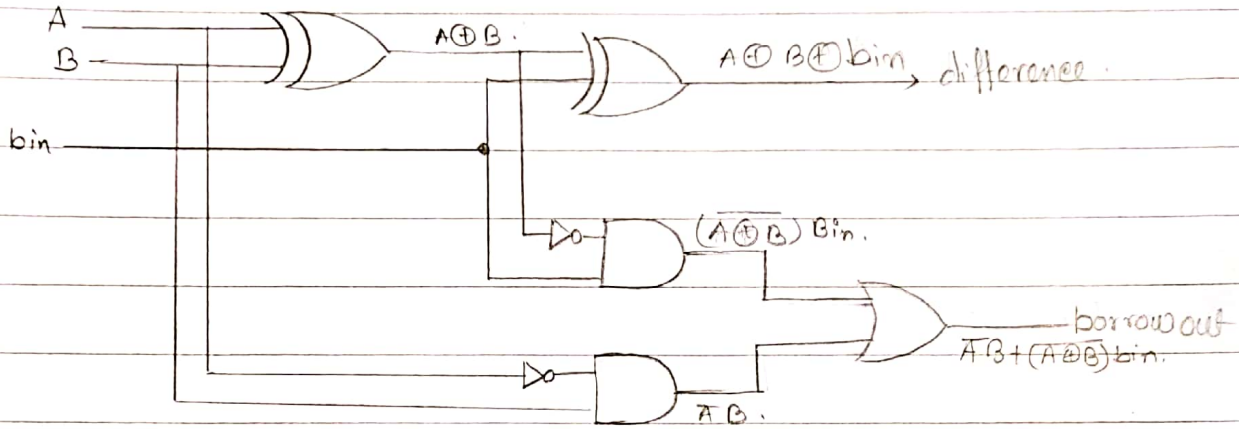
$$= bin\,(\bar{A}\bar{B} + AB) + \overline{bin}\,(A\bar{B} + \bar{A}B)$$

$$= bin\,(\overline{A \oplus B}) + \overline{bin}\,(A \oplus B)$$

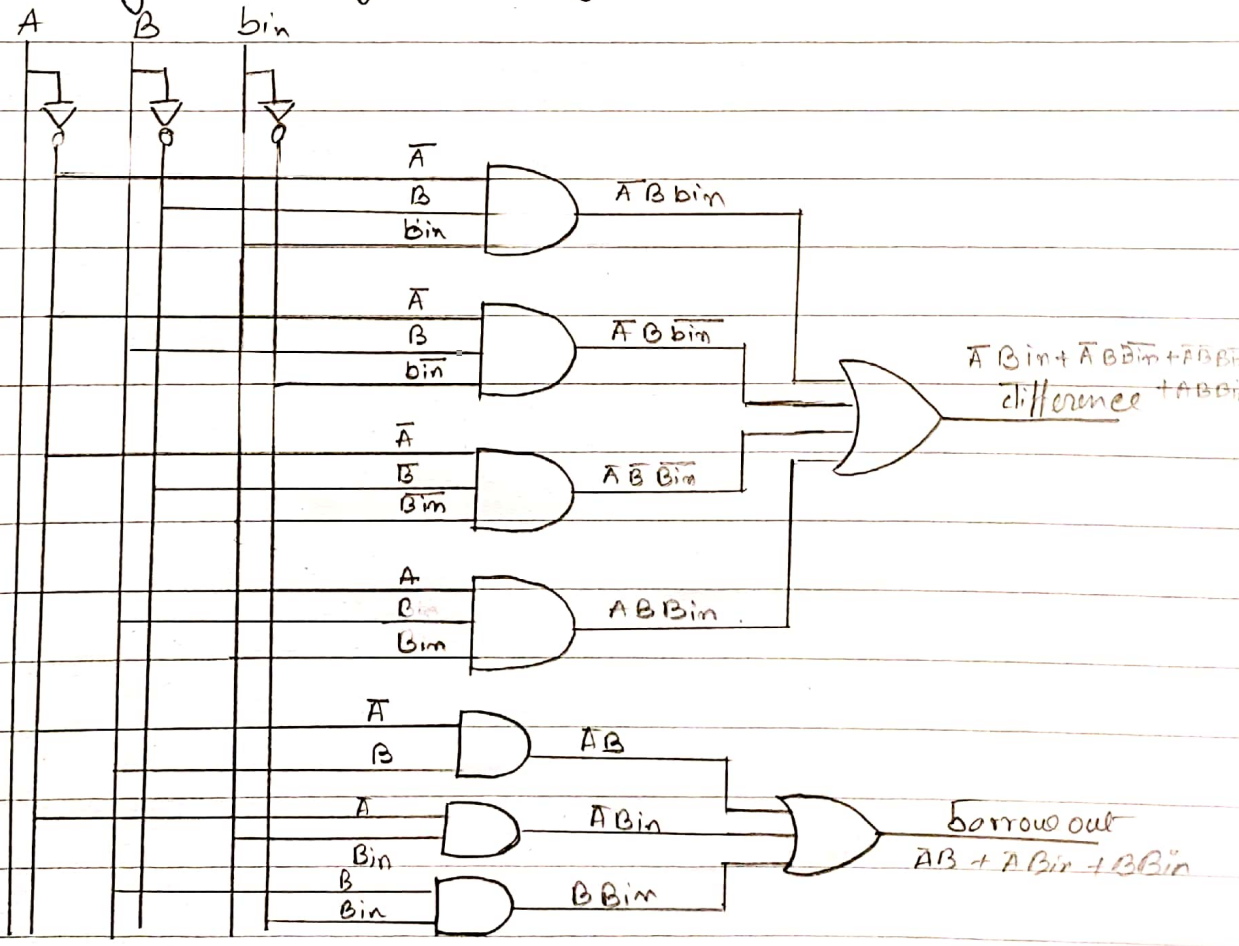
$$= A \oplus B \oplus bin$$

$$b = \bar{A}\bar{B}\,bin + \bar{A}B\,\overline{bin} + \bar{A}B\,bin + AB\,bin$$
$$= \bar{A}B + (\overline{A \oplus B})\,bin. = \bar{A}B + \bar{A}Bin + BBin.$$

■ block diagram using XOR NOT and and OR gate :-



■ block diagram using AOI logic :-

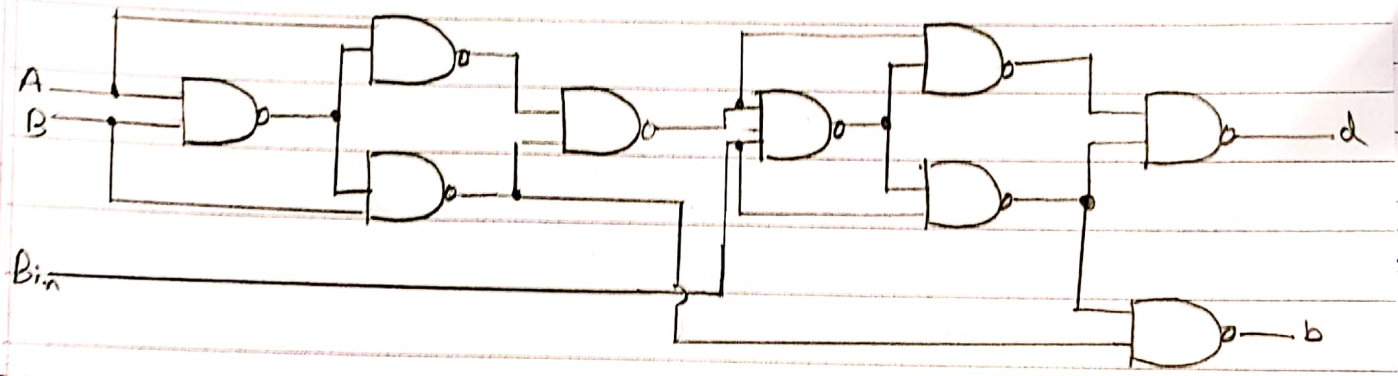


■ NAND LOGIC :- $d = A \oplus B \oplus bin = \overline{(A \oplus B)(\overline{A \oplus B})bin} \cdot \overline{bin(A \oplus B)bin}$

$b = \overline{A}B + bin(\overline{A \oplus B}) = \overline{\overline{A}B + bin(A \oplus B)} = \overline{\overline{A}B} \cdot \overline{bin(A \oplus B)}$

$= \overline{B(\overline{A}+\overline{B})} \cdot \overline{bin[bin + (A \oplus B)}$

$= \overline{B \cdot AB} \cdot \overline{bin[bin \cdot (A \oplus B)]}$



■ NOR LOGIC :- $d = A \oplus B \oplus bin = \overline{(A \oplus B)bin + (\overline{A \oplus B})\overline{bin}}$

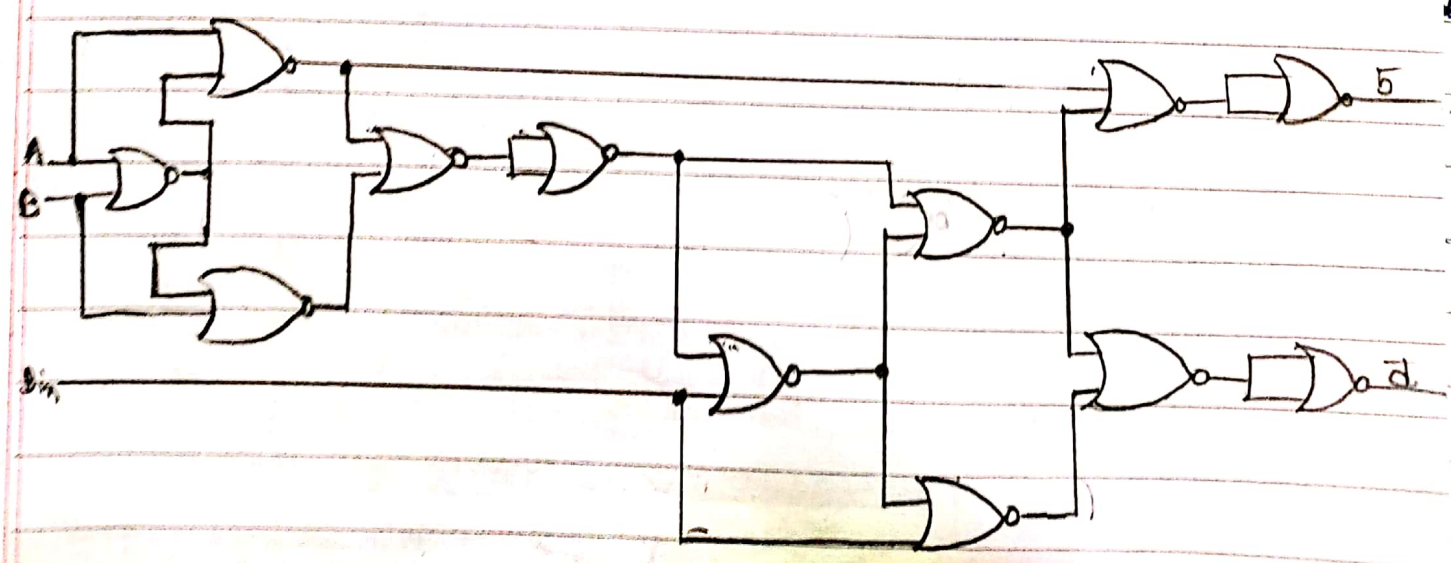
$= \overline{[(A \oplus B) + (\overline{A \oplus B})\overline{bin}][bin + (A \oplus B)\overline{bin}]}$

$= \overline{\overline{A \oplus B} + \overline{(A \oplus B) + bin} + \overline{bin + (A \oplus B) + bin}}$

$= \overline{(A \oplus B) + (\overline{A \oplus B}) + bin + \overline{bin + (A \oplus B) + bin}}$

$b = \overline{A}B + bin(\overline{A \oplus B}) = \overline{A}(A+B) + (\overline{A \oplus B})[(A \oplus B) + bin]$

$= \overline{A + (\overline{A+B}) + (A \oplus B) + (\overline{A \oplus B}) + bin}$



PROCEDURE :-

Step 1 : Open the Circuit Maker 2000 Software.

Step 2 : Select [redacted] the required logic gates under digital Basic > Gates

Step 3 :- Select logic display under Displays > Digital > logic display.

Step 4 :- Select Inverter

Step 5 :- Select the logic Switch under Switch > digital > logic Switch.

Step 6 :- use the wire (+) tools to Connect the various devices with each other as required

Step 7 :- From menu bar Select digital mode under Simulation menu

Step 8 :- press F10 key to run the test.

Step 9 :- Check the output of the test in accordence with the truth table

Step 10 :- Save the project under the file check.

Conclusion :-

From the above experiment it is Concluded that using Circuit maker 2000 Software we can Carry out any operation using any types of logic gates. we can perform one and two bit addition and Subtraction by adder and Subtractor. and the Circuit also implemented : AOI logic and NAND and NOR Logic

DATE

EXPT. NO.

Experiment No 6 :-

■ **Aim :-** Design and implement of a 4-bit adder / Substructor Circuit using a 4-bit binary adder (using IC74LS83)

■ **Requirment :-** Circuit Maker 2000 Software needs to be installed in the Computer.

■ **Theory :-**

▼ **n-bits binary adder :-** The single bit binary adders can be constructed from basic logic gates. But what if we wanted to add together two n-bits number then n-number of 1-bit full adder need to be connected or 'cascade' together to produce what is known as n-bit binary adder. It is also known as Ripple Carry adder

- Example :- 4-bit binary adder.

▼ **Binary addition process using 4-bit binary Adder :-**

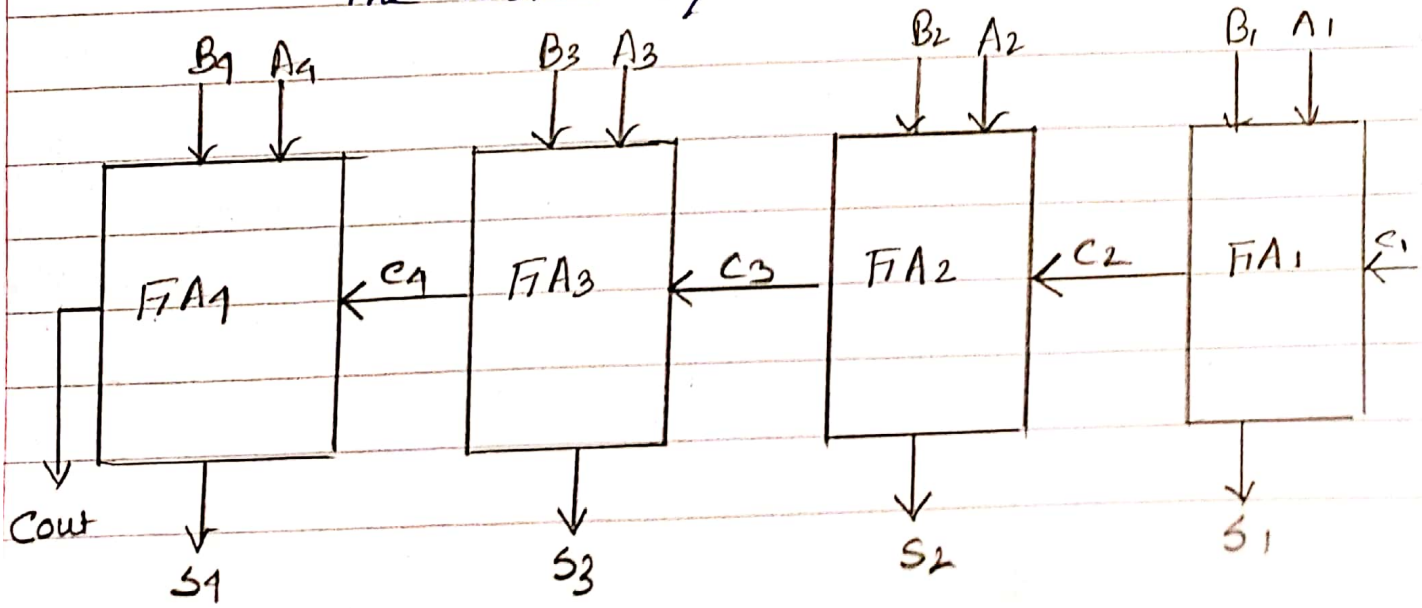
- **Process :-**

Step 1 : firstly the full adder FA1 adds A_1 and

along with the Carray C_1 to generate the Sum S_1 (The first bit of output Sum) and the Carry C_2 which is Connected to the next adder in the chain.

Step 2 :- Next the full adder FA_2 uses this Carry bit C_2 to add with the input bits A_2 and B_2 to generate the Sum S_2 (The second bit of the output Sum) and the Carry C_3 which is again further Connected to the next adder in the chain and so on.

Step-3 :- The process Continues till the last full adder (FA_4) use the Carry bit C_4 to add with its input A_4 and B_4 to generate the last bit (S_4) of the output along the last Carry bit C-out.



DATE

EXPT. NO.

PAGE NO. 49

Let us examine the justification of the above circuit by taking an example of addition of two 4 bit numbers.

Let us add 1011 with 1101

here, $A_1 = 1$ $A_2 = 1$ $A_3 = 0$ $A_4 = 1$

$B_1 = 1$ $B_2 = 0$ $B_3 = 1$ $B_4 = 1$

As there is no previous carry $C_0 = 0$

now $0 + A_1 + B_1 = 0 + 1 + 1 = 10 \rightarrow S_1 = 0, C_1 = 1$

$C_1 + A_2 + B_2 = 1 + 1 + 0 = 10 \rightarrow S_2 = 0, C_2 = 1$

$C_2 + A_3 + B_3 = 1 + 0 + 1 = 10 \rightarrow S_3 = 0, C_3 = 1$

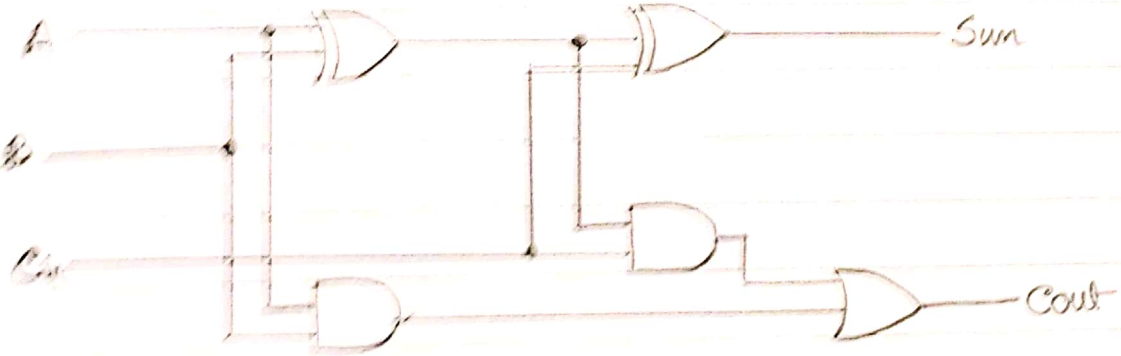
$C_3 + A_4 + B_4 = 1 + 1 + 1 = 11 \rightarrow S_4 = 1, C_{out} = 1$

therefore, final result of the addition would be

$C_{out} S_4 S_3 S_2 S_1 = 11000$.

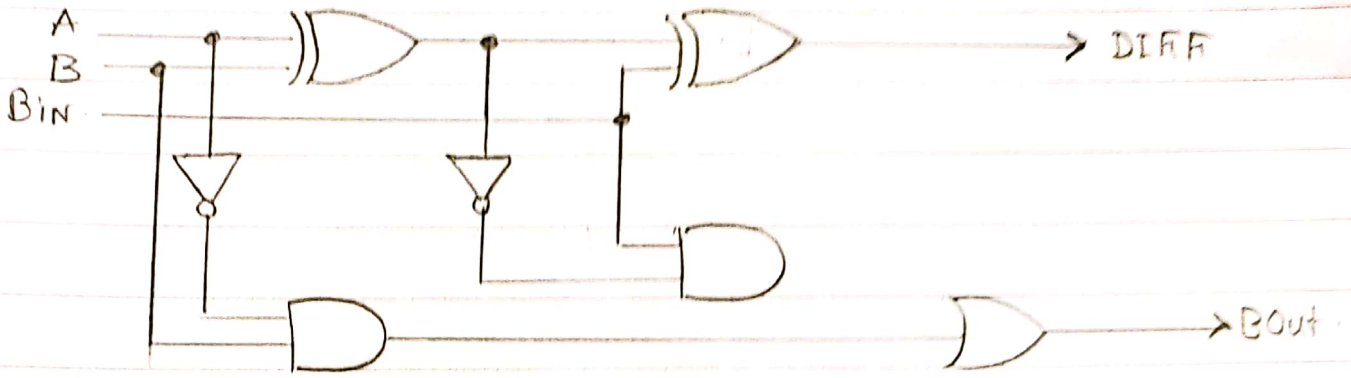
- Binary Subtraction Circuit and Addition circuit :-

- Binary Addition Circuit :-



EXPT. NO.

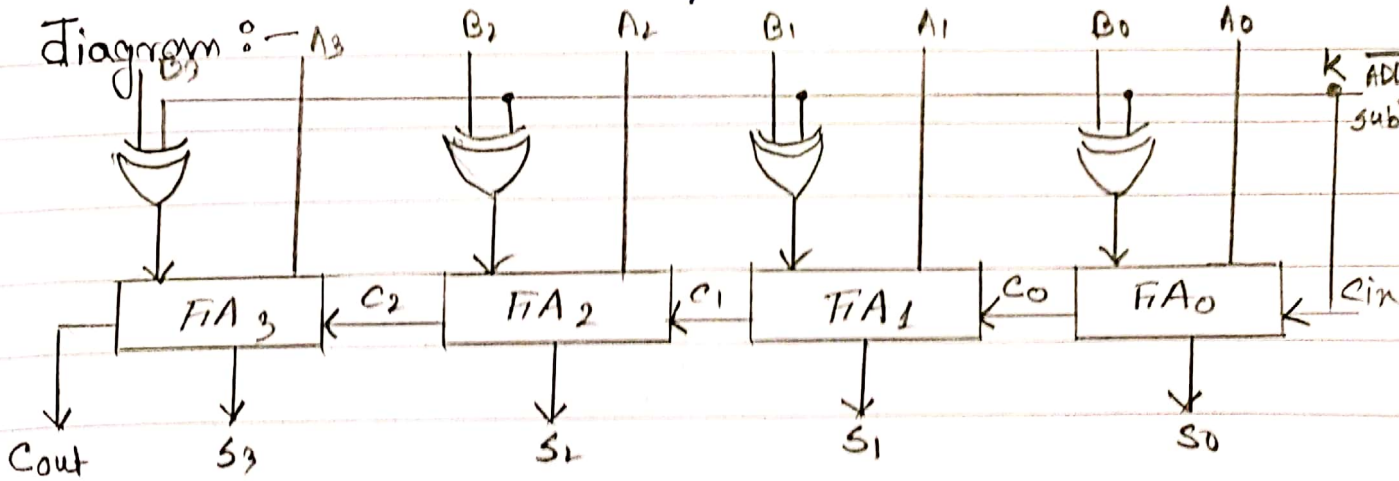
Binary Subtraction Circuit :-



4-bit binary adder Subtractor Circuit using 1 bit Binary full adder.

The 4-bit binary adder / Subtractor produces either the addition or the subtraction of two 4-bit numbers based on the value of initial carry or borrow C_0. Let the 4-bit binary numbers, $A_0\ A_1\ A_2\ A_3$ and $B_0\ B_1\ B_2\ B_3$. The operation of 4-bit Binary Adder Subtractor is similar to that of 4 bit binary adder and 4 bit Binary Subtractor.

Diagram :-



Block Diagram.

- When the mode input (K) is at a low logic, i.e '0' then the circuit acts as an adder, and when the mode input is at a high logic i.e '1' the circuit acts as a subtractor.

- The exclusive-OR gate connected in series recived input K and one of the input B.

- When K is at low logic we have $B \oplus 0 = B$ the full adder recived the value of B, the input carry is 0 and the circuit perform A+B.

- when M is at a high logic, we have $B \oplus 1 = B'$ and $c_0 = 1$ the B inputs are complemented, and a 1 is added through the input carry The circuit perform the operation A plus the 2's of B

Procedure :-

Step 1 :- open the circuit makes 2000 software.

Step 2 :- Select the required logic gate(s) under digital Basics > Gates

Step 3 :- Select the logic Display under display > Digita > logic Display

Step 4 :- Select the logic Switch under Switches > digital > logic Switch.

Step 5 :- Select the Ic 74LS83 under digital by function > arithmetic > 7483

Step 6 :- use the wire tool (+) to connect the various devices with each other as required.

Step 7 :- From Menu Bar, Select digital mode under Simulation menu

step 8 :- press F10 Key to run the test.

Step 9 :- check the output (s) of the teste in accordence with the truth table of the device. and create a observation table.

step 10 :- Save the project under file > Save.

Observation Table :-

- binary adder :-

SL	A_4	A_3	A_2	A_1	B_4	B_3	B_2	B_1	Cout	S_4	S_3	S_2	S_1
1.	1	0	1	0	0	1	0	1	0	1	1	1	1

SL.	A_4	A_3	A_2	A_1	B_4	B_3	B_2	B_1	Cout	S_4	S_3	S_2	S_1
2.	1	0	1	1	1	0	0	1	1	0	1	0	0
3.	1	0	0	0	1	1	1	0	1	0	1	1	0
4.	1	1	1	1	1	1	1	1	1	1	1	1	0
5.	0	1	0	1	0	1	1	0	0	1	0	1	1

- Binary Subtractor :-

SL.	A_4	A_3	A_2	A_1	B_4	B_3	B_2	B_1	Bout	S_4	S_3	S_2	S_1
1.	1	0	1	0	0	1	0	1	1	0	1	0	1
2.	1	0	0	0	1	1	1	0	0	1	0	1	0
3.	0	0	1	1	0	1	0	1	0	1	1	1	0
4.	1	1	0	0	1	0	0	1	1	0	0	1	1
5.	1	1	1	1	0	0	0	0	1	1	1	1	1

- Conclusion :- From the above experiment it is concluded that using Circuit maker 2000 Software, we can carry out any operation of any type of logic gate(XOR) and A IC- 7415 83. we can perform addition and Subtraction operation in the same Circuit by changing the control line input. i.e if the input of control line is 1 then the Circuit behave as a Subtractor Circuit if it is '0' then then circuit behave as a Adder circuit.

Experiment NO 7 :-

AIM :- Design and implement using xor logic gate
ⓐ 4 bit gray to binary Converter.
ⓑ 4bit binary to gray Converter.

REQUIRMENT :- Circuit maker 2000 software needs to be installed in the Computer.

Theory :-

■ Binary to Gray Converter :-

The input to the circuit is a 4-bit binary and output of the Circuit is a 4-bit bray Code.

■ Truth table :-

4 bit binary				4bit Gray			
B_4	B_3	B_2	B_1	G_4	G_3	G_2	G_1
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

B_4	B_3	B_2	B_1	G_4	G_3	G_2	G_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

The 4bit binary and the corrosponding gray code shown in the truth table we observe that :

1. The entries for G_4 are exactly the same as those for B_4 Therefore $G_4 = B_4$
2. The entries for G_3 are :

   $G_3 = 1$ only when either $B_4 = 1$ or $B_3 = 1$, $G_3 = 0$ $G_3 = 0$ for $B_4 = B_3 = 0$ and $B_4 = B_3 = 1$ This is an XOR operation of B_4 and B_3. Therefore $G_3 = B_4 \oplus B_3$
3. The entries for G_2 are :

   $G_2 = 1$ only when $B_4 = 1$ or $B_3 = 1$ or $G_2 = 0$ for for both $B_3 = B_2 = 1$ and $B_3 = B_2 = 0$ This is an XOR operation of B_3 and B_2 therefore $G_2 = B_3 \oplus B_2$

4. The entries for G_1 are:

$G_1 = 1$ only when $B_2 = 1$ or $B_1 = 1$. $G_1 = 0$ for both $B_2 = B_1 = 1$ and $B_2 = B_1 = 0$. This is an x-OR operation of B_2 and B_1. Therefore $G_1 = B_1 \oplus B_2$.

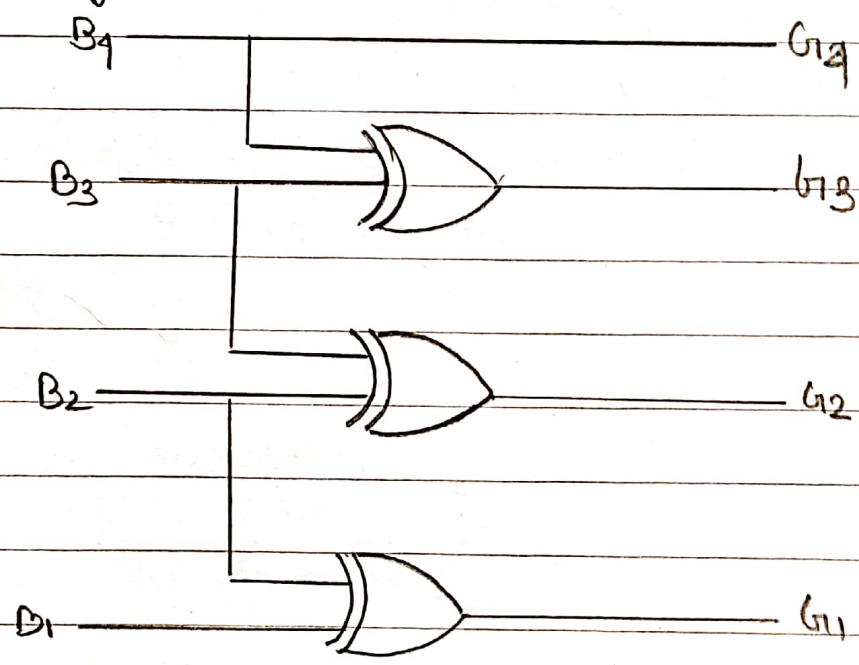
$G_4 = B_4$

$G_3 = \overline{B_4} B_3 + B_4 \overline{B_3} = B_4 \oplus B_3$

$G_2 = \overline{B_3} B_2 + B_3 \overline{B_2} = B_3 \oplus B_2$

$G_1 = \overline{B_2} B_1 + \overline{B_1} B_2 = B_2 \oplus B_1$

Logic diagram :-



☐ Gray to binary Converter :-

The 4 bit input Gray Code and Corresponding output binary number shown in the truth table.

Truth table :-

Input				Output			
G_4	G_3	G_2	G_1	B_4	B_3	B_2	B_1
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	0	0	0	1	0	0	0
1	0	0	1	1	0	0	1
1	0	1	1	1	0	1	0
1	0	1	0	1	0	1	1
1	1	1	0	1	1	0	0
1	1	1	1	1	1	0	1
1	1	0	1	1	1	1	0
1	1	0	0	1	1	1	1

1. The entries for B_4 are exactly the same as those for G_4. Therefore $B_4 = G_4$
2. The entries for B_3 are.
   $B_3 = 1$, Only when the number of 1s in G_4

and G_{13} is an odd number. Otherwise $B_3 = 0$ so, B_3 is the modulo sum of G_{14} and G_{13}. Therefore $B_3 = G_{14} \oplus G_{13}$

3. The entries for B_2 are:

$B_2 = 1$, only when the number of 1s in G_{14}, G_{13} and G_{12} is an odd number. Otherwise $B_2 = 0$. So B_2 is the modulo sum of G_{14}, G_{13} and G_{12} i.e modulo sum of B_3 and G_{12}. Therefore $B_2 = B_3 \oplus G_{12}$

4. The entries for B_1 are:

$B_1 = 1$, only when the number of 1s in G_{14}, G_{13}, G_{12} and G_{11} is an odd number. Otherwise $B_1 = 0$. So B_1 is the modulo sum of G_{14}, G_{13}, G_{12} and G_{11}, i.e modulo sum of B_2 and G_{11}. Therefore $B_1 = B_2 \oplus G_{11}$

$$B_4 = G_{14}$$

$$B_3 = \overline{G_{14}}\, G_{13} + G_{14}\, \overline{G_{13}} = G_{14} \oplus G_{13}$$

$$B_2 = \overline{G_{14}}\, G_{13}\, \overline{G_{12}} + \overline{G_{14}}\, \overline{G_{13}}\, G_{12} + G_{14}\, \overline{G_{13}}\, \overline{G_{12}} + G_{14}\, G_{13}\, G_{12}$$

$$= \overline{G_{14}}\, (G_{13} \oplus G_{12}) + G_{14}\, (\overline{G_{13} \oplus G_{12}})$$

$$= G_{14} \oplus G_{13} \oplus G_{12} = B_3 \oplus G_{12}$$

$$B_1 = \overline{G_{14}}\, \overline{G_{13}}\, \overline{G_{12}}\, \overline{G_{11}} + \overline{G_{14}} + \overline{G_{13}}\, G_{12}\, \overline{G_{11}} + \overline{G_{14}}\, G_{13}\, \overline{G_{12}}\, \overline{G_{11}} + \overline{G_{14}}\, G_{13}\, G_{12}\, G_{11}$$

$$+ G_{14}\, G_{13}\, \overline{G_{12}}\, G_{11} + G_{14}\, G_{13}\, G_{12}\, \overline{G_{11}} + G_{14}\, \overline{G_{13}}\, \overline{G_{12}}\, G_{11} + G_{14}\, \overline{G_{13}}\, G_{12}\, G_{11}$$

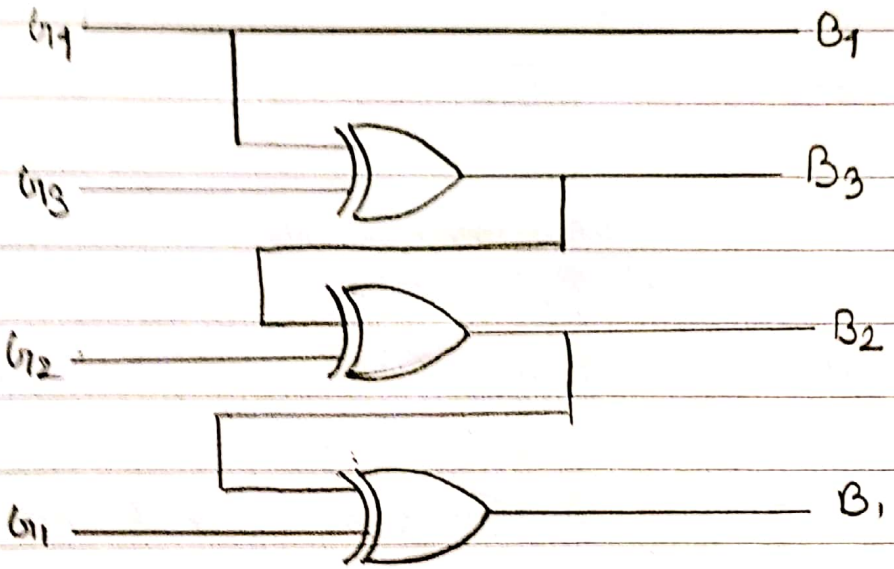
$$= \overline{G_{14}}\, \overline{G_{13}}\, (G_{12} \oplus G_{11}) + G_{14}\, G_{13}\, (G_{12} \oplus G_{11}) + \overline{G_{14}}\, G_{13}\, (\overline{G_{12} \oplus G_{11}})$$

$$+ G_{14}\, \overline{G_{13}}\, (\overline{G_{13} \oplus G_{11}})$$

$$= (G_{11} \oplus G_{11})\, (\overline{G_{14} \oplus G_{13}}) + (\overline{G_{12} \oplus G_{11}})\, (G_{14} \oplus G_{13})$$

$$= G_{14} \oplus G_{13} \oplus G_{12} \oplus G_{11} = B_2 \oplus G_{11}$$

Logic Diagram :-



Procedure :-

STEP 1 : Open the circuit maker 2000 Software.

STEP 2 : Select the required logic gate (XOR) under digital basic > Gates.

STEP 3 : Select the logic display under display > Digital > logic display.

STEP 4 : Select the logic Switch under Switches > digita > logic Switch.

STEP 5 : use the wire tool to Connect the various devices with each other as required.

STEP 6 : From menu bar, select digital mode under Simulation menu.

STEP 7 : press F10 Key to run the test

STEP 8 :- check the output of the test in according with the truth table of the device. an observation table and create

STEP 9 :- Save the project under file > Save.

Observation table :-

Binary to Gray Conversion :-

B_4	B_3	B_2	B_1	G_4	G_3	G_2	G_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
0	1	0	1	0	1	1	1

Gray to Binary Conversion :-

G_4	G_3	G_2	G_1	B_4	B_3	B_2	B_1
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
0	1	1	1	0	1	0	1

Conclusion :-

from the above experiment it is Concluded that using Circuit maker 2000 Software we can carray out any operation of any types of logic gates (XOR). we can convert Binary to gray Code and Gray to binary using those circuit very easily.

Experiment NO :- 8

Aim :- Design an implement using XOR and NOT gates.

A. Even parity generator.
B. odd parity generator.
C. Even parity checker.
D. odd parity checker.

Requirment :- Circuit maker 2000 Software needs to be installed in the Computer.

Theory :-

■ Parity bit :-

Binary data, when transmitted and processed, is susceptible to noise that can alter its 1s to 0's or 0's to 1's. To detect such errors, an additional bit called Parity bit is added to data bits.

■ parity Generator :-

A parity Generator is a Combinational logic Circuit that generates the parity bit in the transmitter.

A parity bit is used for the purpose of deteting Error

during Transmission of binary information.

it is an extra bit included with a binary number / message to make the number of 1's either odd or even.

There are two types of parity generator :-

1) even parity generator.
2) odd parity generator.

☐ even parity generator :-

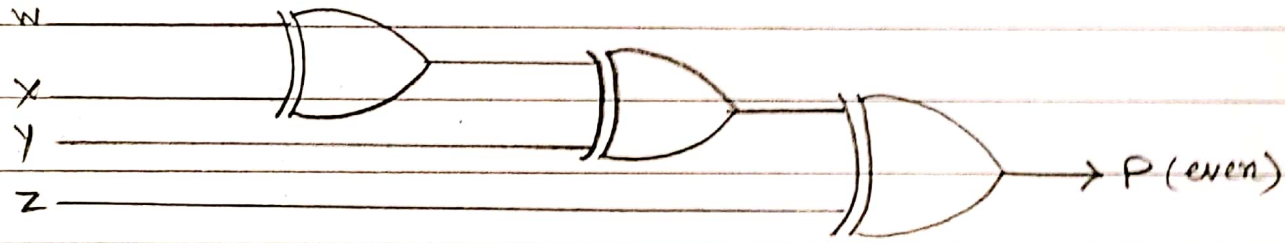
for even parity, a parity bit 1 is added such that the total number of 1's in the 4 bit input and the parity bit together is even.

☐ Truth table :-

w	x	y	z	even parity bit.
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1

1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Circuit Diagram :-



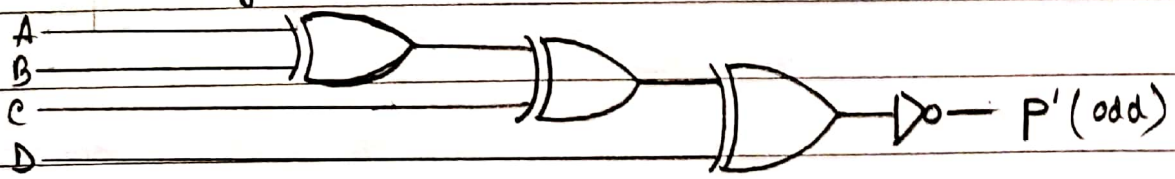
[] add parity generator :-

An add parity bit generator output a 1 when the number of 1's in the data bits is even so that the total numbe of 1's in the data bits and the parity bits together is odd.

☐ truth table :-

A	B	C	D	output odd parity bit (P)
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

Circuit diagram :-



Parity checker :-

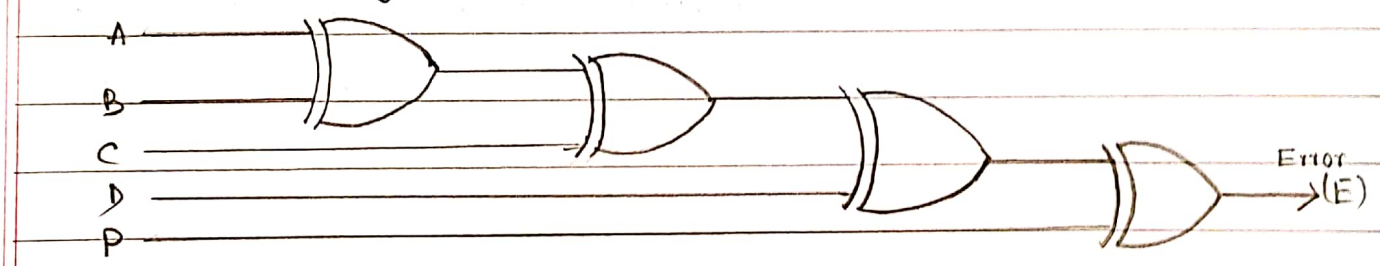
- A Circuit that checks the parity in the Reciver is called Parity Checker.
- The parity checker Circuit checks for possible Errors in the Transmission.
- Since the information Transmitted with even parity, the Received must have an even number of 1's. if it has odd number of 1's, its indicates that there is an error occured during transmission.
- The output of the both (even and odd) parity checker is denoted by PEC (Parity Error checker). if there is error, that is, if it has odd number of 1's it will indicates 1 if no then PEC will indicate 0.
- that means. if it is even parity checker. if PEC will indicate 1 that means 'error' if PEC will indicate 0 that means no error.
  - if it is odd parity checker. if PEC will indicates 1 that means 'no error' if PEC will indicate 0 that means 'error'

☐ Truth table :-↓

Sl. No.	A	B	C	D	P	Parity error (E)
0	0	0	0	0	0	0
1	0	0	0	0	1	1
2	0	0	0	1	0	1
3	0	0	0	1	1	0
4	0	0	1	0	0	1
5	0	0	1	0	1	0
6	0	0	1	1	0	0
7	0	0	1	1	1	1
8	0	1	0	0	0	1
9	0	1	0	0	1	0
10	0	1	0	1	0	0
11	0	1	0	1	1	1
12	0	1	1	0	0	0
13	0	1	1	0	1	1
14	0	1	1	1	0	1
15	0	1	1	1	1	0
16	1	0	0	0	0	1
17	1	0	0	0	1	0
18	1	0	0	1	0	0
19	1	0	0	1	1	1
20	1	0	1	0	0	0
21	1	0	1	0	1	1

22.	1	0	1	1	0	1
23	1	0	1	1	1	0
24	1	1	0	0	0	0
25	1	1	0	0	1	1
26	1	1	0	1	0	1
27	1	1	0	1	1	0
28	1	1	1	0	0	1
29	1	1	1	0	1	0
30	1	1	1	1	0	0
31.	1	1	1	1	1	1

Circuit Diagram :-



Basic working principle of parity generator / checker. :-

In Order to check and generate the proper parity bit in a given code word, the basic principle used is. "The modulo Sum of an even number of 1's is always a 0 and the ~~the~~ modulo Sum of an odd number of 1's is always 1".

Therefore in order to check for an error, all the bits in the received word are added. if the modulo sum is a 0 for an odd parity system or a 1 for an even parity system an error is detected.

PROCEDURE :-

Step 1 :- Open circuit maker 2000 Software.

Step 2 :- Select the require logic gates under digital Basic > Gates.

Step 3 :- Select logic display under display > digital > logic Display

Step 4 :- Select Inverter.

Step 5 :- Select the logic Switch under Switch > digital > logic Switch.

Step 6 :- use the wire (+) tools to connect the various device with each other as required.

Step 7 :- From menu bar Select digital mode under Simulation menu.

Step 8 :- Press F10 Key to run the test.

Step 9 :- Check the output of the test in accordence with truth table and create a observation table.

Step 10 :- Save the project under the file check.

Observation table :->

- EVEN PARITY CHECKER GENERATOR :-

Sl NO	SENDER Message.				PARITY	RECIVER					ERROR	Remarks.
	D3	D2	D1	D0	P	D3	D2	D1	D0	P	E	
1	1	0	0	1	0	1	1	0	1	0	1	Error
2.	1	0	0	0	1	1	0	0	1	1	1	Error

3	1 1 1 1	0	1 0 1 1 0	1	error
4	1 0 1 0	0	1 0 1 0 0	0	no error
5	1 1 1 0	1	1 1 1 0 1	0	no error.

Odd parity checker Generator :-

Sl. No.	Sender message	parity	Receiver	Error	Remark
	D_3 D_2 D_1 D_0	P	D_3 D_2 D_1 D_0 P	E	
1.	0 0 1 0	0	0 1 1 0 0	0	error
2.	0 0 1 1	1	0 0 0 1 1	0	error
3.	1 0 0 1	1	1 0 0 1 1	1	no error
4.	1 1 0 1	0	1 0 0 1 0	0	error.
5.	0 0 1 1	1	0 0 1 1 1	1	no error

Conclusion :- from the above experiment it is Conclude that using Circuit maker 2000 software we can carry out any operation of any types of logic gates. we can [redacted] Create a even and odd parity generator / checker Circuit and by this Circuit we can find out a single bit error at the reciver end if any.

EXPERIMENT NO. 9

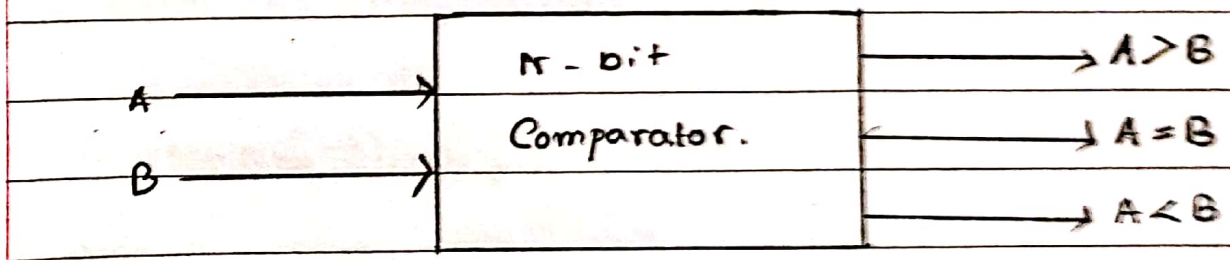
Aim :- Design an 4 bit Comparator using a Single 7485IC.

Requirment :- Circuit maker 2000 Software needs to be installed in the Computer.

Theory :-

Comparator :-

A digital Comparator is a Combinational Circuit that Compares two digital or binary numbers in order to find out whether one binary number is equal, less than or greater than the other binary number. we logically designed a Circuit for which we will have to two inputs one for A and another for B and three output terminals one for A>B Condition, one for A=B Condition and one for A<B Condition.



1-Bit Magnitude Comparator :-

A Comparator used to Compare two bits is called a single bit Comparator. It Consists of two inputs each for two single bit numbers and three outputs to generate less than, equal to and greater than between two binary numbers.

Truth table :-

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

2- bit Comparator :-

A Comparator used to Compare two binary numbers each of two bits is called a 2-bit Comparator. It Consist of four inputs and three outputs to generate less than, equal to and greater than between two binary numbers.

Truth table :→

SL NO.	Input				Output		
	A1	A0	B1	B0	A < B	A = B	A > B
1.	0	0	0	0	0	1	0

2.	0	0	0	1	1	0	0
3.	0	0	1	0	1	0	0
4.	0	0	1	1	1	0	0
5.	0	1	0	0	0	0	1
6.	0	1	0	1	0	1	0
7.	0	1	1	0	1	0	0
8.	0	1	1	1	1	0	0
9.	1	0	0	0	0	0	1
10.	1	0	0	1	0	0	1
11.	1	0	1	0	0	1	0
12.	1	0	1	1	1	0	0
13.	1	1	0	0	0	0	1
14.	1	1	0	1	0	0	1
15.	1	1	1	0	0	0	1
16.	1	1	1	1	0	1	0

4 bit Comparator :-

A Comparator used to Compare two binary numbers each of four bits is called a 4 bit Comparator. It Consists of eight Inputs each for two four bit numbers and three outputs to generate less than, equal to and greater than between two binary numbers.

In a 4-bit Comparator the Condition of A>B can possible in the following four cases:

1. If $A_3 = 1$ and $B_3 = 0$
2. If $A_3 = B_3$ and $A_2 = 1$ and $B_2 = 0$
3. If $A_3 = B_3$ and $A_2 = B_2$ and $A_1 = 1$ and $B_1 = 0$
4. If $A_3 = B_3$, $A_2 = B_2$, $A_1 = B_1$ and $A_0 = 1$ and $B_0 = 0$

Similarly the Condition for A<B can be possible in the following four cases:

1. if $A_3 = 0$ and $B_3 = 1$
2. if $A_3 = B_3$ and $A_2 = 0$ and $B_2 = 1$
3. if $A_3 = B_3$, $A_2 = B_2$ and $A_1 = 0$ and $A_2 = 1$
4. if $A_3 = B_3$, $A_2 = B_2$, $A_1 = B_1$ and $A_0 = 0$ and $B_0 = 1$

The Condition of A=B is possible only when all the individual bits of one number exactly coincide with Corresponding bits of another number.

Application of Comparator :-

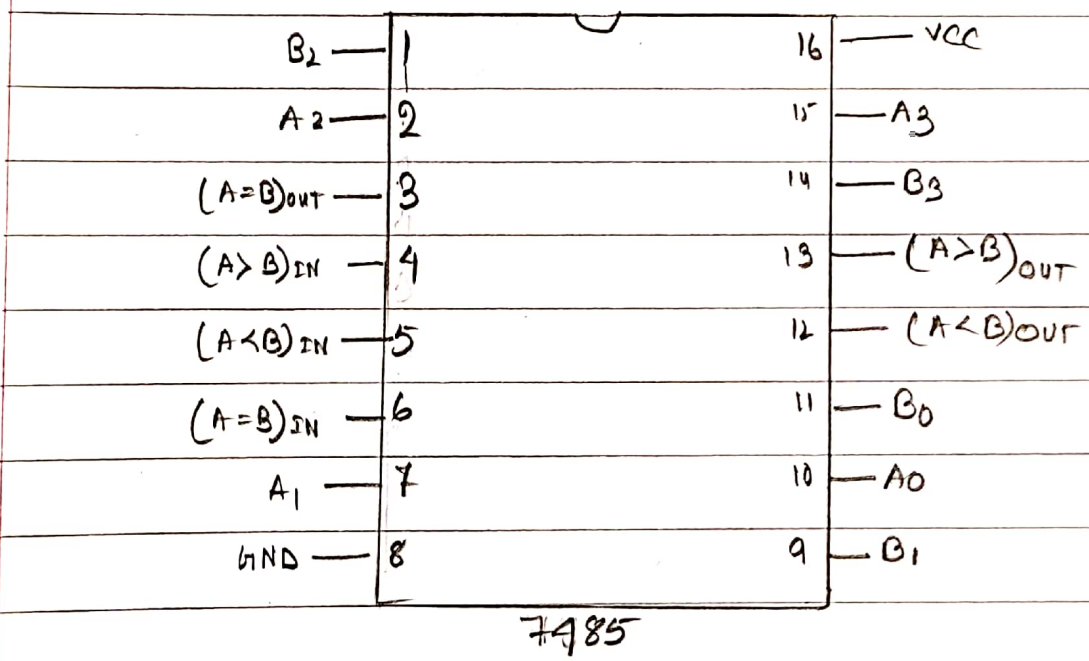
1. Comparators are used in central processing unit and micro Controller
2. These are used in control application in which the binary numbers representing phyical variable such as temperature, position., etc. are Compared with reference value.
3. Comparators are also used as process Controllers and for Servo motor control
4. used in password verification and biometric application.

IC Comparator :-

the 7485 is the 1 bit IC Comparator. pin labelled $(A>B)_{IN}$, $(A<B)_{IN}$ and $(A=B)_{IN}$ are used for Cascading. Pin labelled $(A>B)_{OUT}$, $(A<B)_{OUT}$ and $(A=B)$ is used from take the output.

'Vcc is used for power Supply (5 V dc). GND is Ground. A_0 A_1 A_2 A_3 and B_0 B_1 B_2 B_3 are two binary number.

Pin diagram :-



Procedur :-

Step 1 :- Open the Circuit maker 2000 Software.

Step 2 :- Select the logic Switch under Switch > digital > logic Switch.

DATE

EXPT. NO. 6.1

Step 3 :- Select the logic display under the display > digital > logic display

Step 4 :- Select the IC 7485 under digital by function > Comparators > 7485.

Step 5 :- Use the wire tools (1) to connect the various devices with each other as required.

Step 6 :- From menu bar select digital mode under Simulation menu.

Step 7 :- Press the key to run the test

Step 8 :- Check the output(s) of the test in accordance with the truth table of the devices and Create a observation table

Step 9 :- Save the project under file > Save.

Observation table :-

Sl. No.	Number A				Number B				A > B	A = B	A < B
	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0			
1.	1	0	1	0	1	0	1	0	0	1	0
2.	1	0	1	1	0	1	1	0	1	0	0
3.	0	1	1	1	1	1	1	1	0	0	1
4.	1	0	0	0	1	0	0	1	0	0	1
5.	0	0	1	1	0	0	0	1	1	0	0

Conclusion :- For the above experiment it is Concluds that using circuit make 2000 we can carry out any operation by logic gates and IC (7485). By implementing a 4-bit Comparator we can find out the Comparision of two 4bit number that which one is greater or less or both are equal.

PIONEER

EXPERIMENT NO :- 10

Aim :- Design and Implement the following Combinational Circuit —

(a) Binary to Octal decoder with enable input using 74LS138 IC.

(b) octal - to Binary encoder with enable input using 74148 IC.

Requirment :- Circuit maker 2000 Software needs to be download.

Theory :-

■ **Decoder :-**

A decoder is a logic Circuit that Converts an N-bit binary input Code into M output lines Such that only one output line is activated for each one of the possible Combinations of inputs.

A decoder Circuit takes binary data of 'n' inputs into 2^n unique output. In addition to input pin the decoder has a enable pin. This enables the pin when negated, makes the Circuit inactive.

Binary to octal decoder :-

the circuit for a decoder with 3 input and eight output. it used all AND gates and therefore, the output are active heigh. For active low output, NAND gates are used.

This decoder can be refferred to in several ways.

It can be called a "3 line to 8 line" decoder because it has three input line and eight output line.

it also called "Binary to octal decoder" because it take three binary inputs and active one of the eight (octal) outputs.

it also referred to as a "1-of-8 decoder" because only one of the eight output is active at a time.

Enable input :-

Some decoder have one or more ENABLE inputs. that are used to control the operation of the decoder. if a common enable line is connected to the fourth input of the of each gate. a particular output as deterr by the A, B, C input code will go high only when enable line is held high.

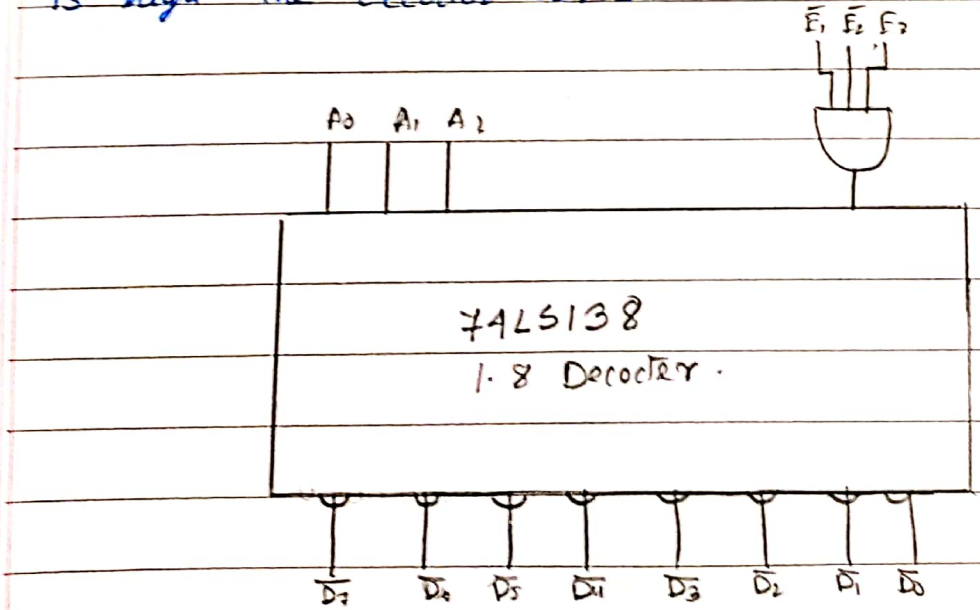
when the enable is low, all the output will be forced to low state regardless of the levels at the A, B and C inputs.

Truth table :-

Input A	B	C	Output D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

74LS138 Decoder :-

A_2 A_1 A_0 is the input code and E_3, $\bar{E}_2$ and $\bar{E}_1$ are separate enable inputs that are combined in the NAND gate. The $E_3\bar{E}_2\bar{E}_1$ is the enable signal only when enable is high the decoder works.



Encoder :-

An encoder is a device whose inputs are decimal digit and/or alphabetic characters and whose output are coded representation of those input In other words an encoders may be said to be a combinational logic circuit that perform the revers 'operation' of the decoder. The opposite of the decoding process is encoding.

an encoder has a number of input line, only one of which is active at a given time and produce an n-bit output code depending on the which input are active.

Octal to Binary encoder :-

An octal to Binary encoder accept 8 input lines and produce 3 bit output code corresponding to the activated input.

Truth table :-

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A	B	C
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

from the truth table, the output line C is active when the input octal digit is 1, 3, 5 or 7. Similarly B is 1 when input octal digit is 2, 3, 6 or 7 and A is 1 for input octal digit 4, 5, 6 or 7 Hence therefor.

$$C = D_1 + D_2 + D_5 + D_5$$

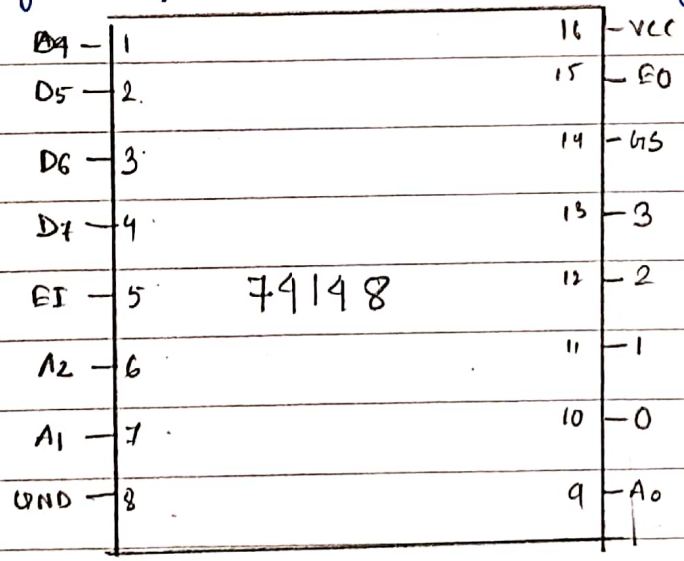
$$B = D_2 + D_3 + D_6 + D_7$$

$$A = D_4 + D_5 + D_6 + D_7.$$

we see theat D_0 is not present in any of expression so, D_0 is a Don't care.

IC 74184 :-

4, 5, 6, 7, 3, 2, 1, 0 those 8 are decimal data input (active low) Vcc is Supply voltage. GND is Ground, A_0, A_1 A_2 Binary address output (active low) EI is enable input (Active low) GS goes low when EI is low and any input is low, EO goes high when EI is low and any input is low. ($EO = \overline{GS}$)



Procedure :-

Step 1 : open circuit maker 2000 softwar.

Step 2 :- Select required logic gates under digital basic > gates

Step 3 :- Select logic display under display > digital > logic display

Step 4 :- Select the logic Switch under Switch > digital > logic Switch

Step 5 :- Select the IC 74138 under digital by function > Decoder / Demux > 74138 and and IC 74148 under digital by function > 74198

Step 6 :- use the wire (+1) tools to Connect the various device with each other as required.

Step 7 :- from menu bar select digital mode under Simulation menu.

Step 8 :- Press F10 Key to run the test.

Step 9 :- Check the output(s) of the test in accordance with the truth table of the device. and Create observration table.

Step 10 :- Save the project under file > save.

Observation table for Binary to Octal decoder :-

E_3	$\bar{E}_2$	$\bar{E}_1$	A_2	A_1	A_0	$\bar{D}_7$	$\bar{D}_6$	$\bar{D}_5$	$\bar{D}_4$	$\bar{D}_3$	$\bar{D}_2$	$\bar{D}_1$	$\bar{D}_0$
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1

PIONEER©

Observation table for Octal to binary encoder :-

$\overline{EI}$	$\overline{I_7}$	$\overline{I_6}$	$\overline{I_5}$	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	$\overline{I_0}$	$\overline{GS}$	A_2	A_1	A_0
0	0	1	1	1	1	1	1	1	0	0	0	0
0	1	0	1	1	1	1	1	1	0	0	0	1
0	1	1	0	1	1	1	1	1	0	0	1	0
0	1	1	1	0	1	1	1	1	0	0	1	1
0	1	1	1	1	0	1	1	1	0	1	0	0
0	1	1	1	1	1	0	1	1	0	1	0	1
0	1	1	1	1	1	1	0	1	0	1	1	0
0	1	1	1	1	1	1	1	0	0	1	1	1

Conclusion :- From the above experiment we can conclude that using Circuit maker 2000 Software we can carry any operation using any types of logic gates and IC's.
- So we can observed that encoder is very much simple Combinational Circuit with input line 2^n and output line n. which can be applied email, video encoder etc. where decoder is a very Complex Combinational Circuit with input line n and output line 2^n which can be applied microprocessor, or memory chip etc.

ASSIGNMENT → 11

Aim :- Design and realization of the following mux AND/OR Demultiplex with Suitable logic Symbol and logic diagram.

a) 2-input mux using IC 74157
b) 4-input mux using IC 74153
c) 8-input mux using IC 74151A
d) 1-line to 4-line demux using IC 74155
e) 1-line to 8 line demux using IC 74155

Requirment :- Circuit maker 2000 Software needs to be Installed in the Computer.

Theory :-

- **Multiplexing :-**

multiplexing means sharing. A common example of multiplexing or sharing occurs when several periferal devices share a single transmission line or bus to communicate with a Computer.

There are two types of multiplexing

1. time multiplexing.
2. frequency multiplexing.

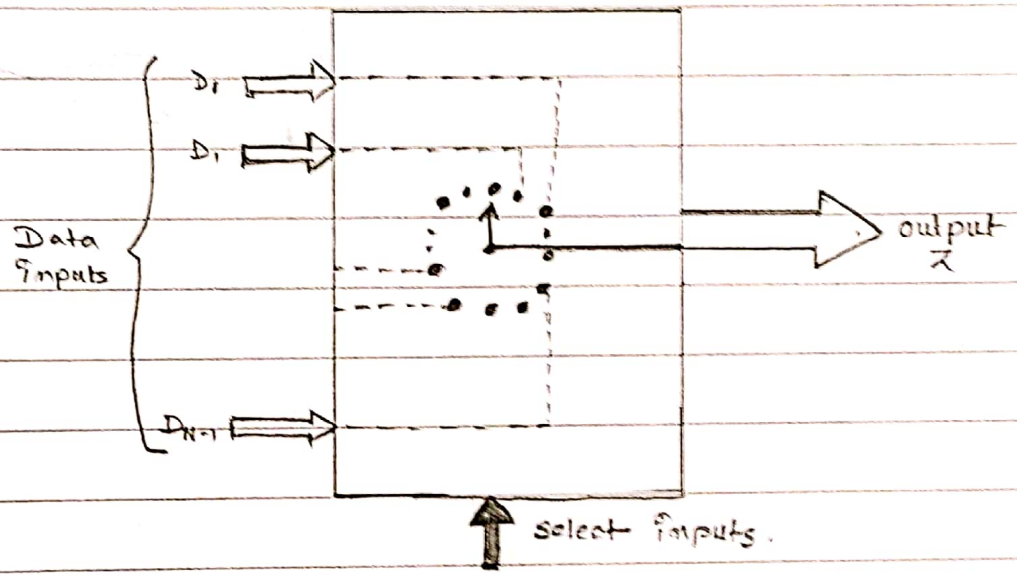
DATE

EXPT. NO.

PAGE NO 58

- **Multiplexer :-** A multiplexer (MUX) or data selector is a logic circuit that accepts several data inputs and allows only one of them at a time to get through the output. The routing of the desired data input to the output is controlled by SELECT inputs.

- **Diagram :-**



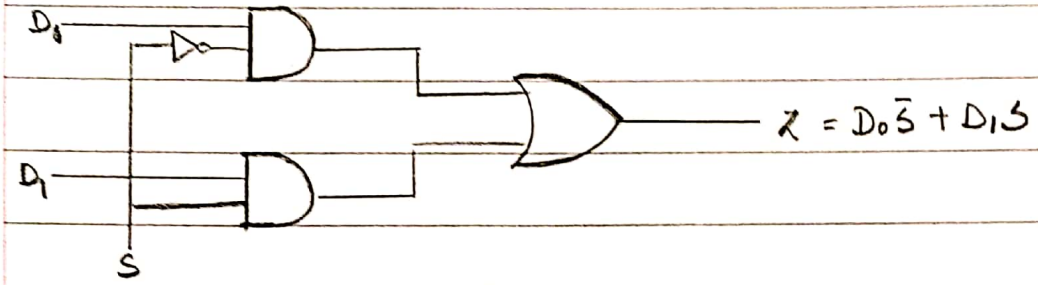
- **2 input multiplexer :-**

2 input multiplexer with data inputs D0 and D1 and and data select input S. the logic level applied to the S input determines which AND gate is enabled. So that its data input passes through the OR gate to the output. the output $Z = \bar{S}D_0 + SD_1$

when S=0 AND gate 1 is enable and AND gate 2 is disabled so Z=D0. when S=1 AND gate 1 is disable and AND gate 2 is enable so, Z=D1.

PIONEER

- logic diagram :-



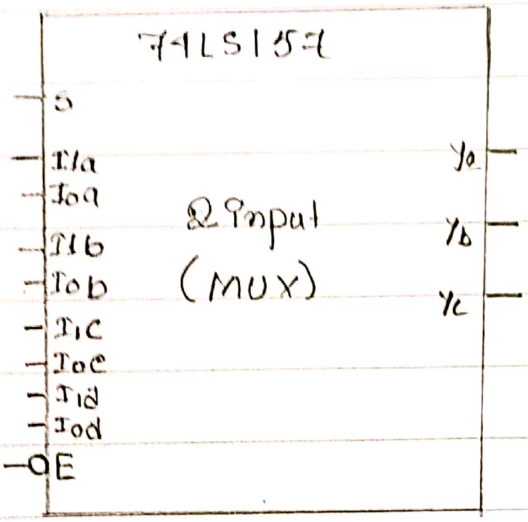
- Function table :-

S	output (Z)
0	D_0
1	D_1

- 2input mux using IC 74157 :-

The 74157 Consist of four Separate 2-input multiplexer on a Single chip. Each of the four multiplexer share a Common Data Select line and a Common $\overline{E}$ (enable) line. Because there are only two Inputs to be Selected from each multiplexer, a Single data select Input is Sufficient. A Low on the $\overline{E}$ Input allows the Selected Input data to pass through to the output. A HIGH on the $\overline{E}$ Input prevent data from going through to the output. i.e disables the multiplexer.

■ Pin diagram :-



■ 4 Input multiplexer :-

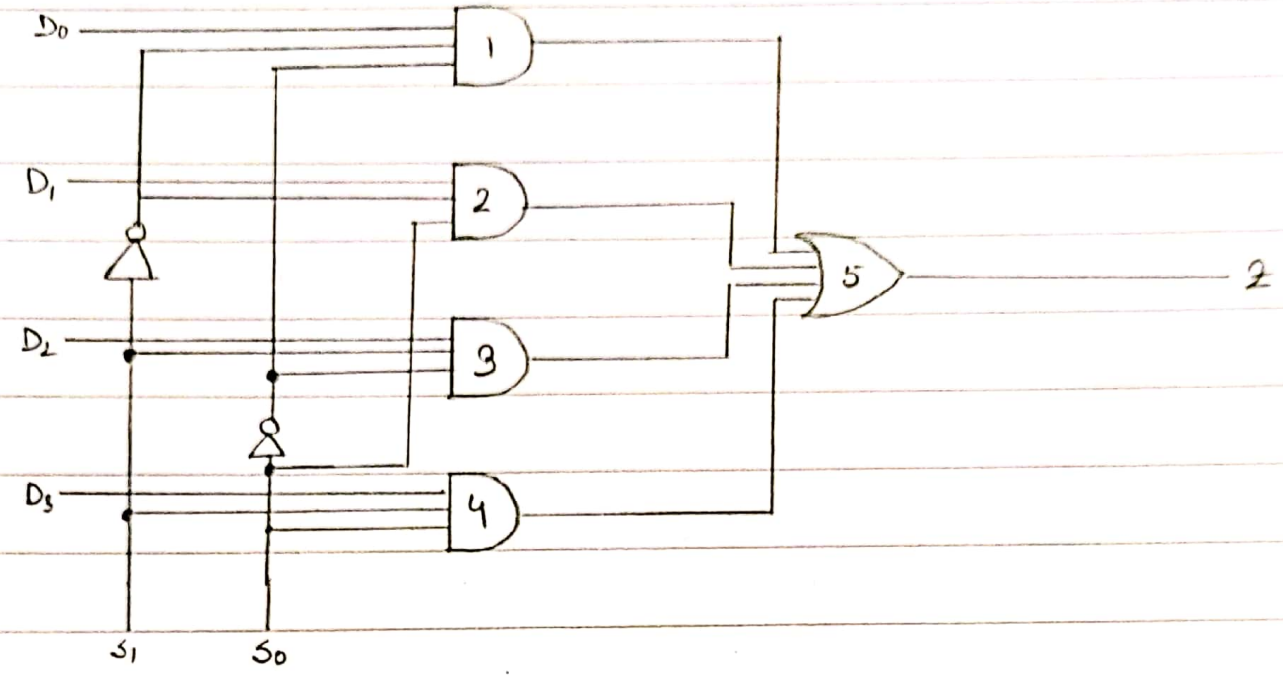
4 input multiplexer with data input D_0, D_1 D_2 and D_3, and data select input S_0 and S_1. the logic level applied to the S_0 and S_1 inputs determine which AND gate in enabled, so that its Data input passes through OR gate to the output.

$$Z = \bar{S_1}\,\bar{S_0}\,D_0 + \bar{S_1}\,S_0\,D_1 + S_1\,\bar{S_0}\,D_2 + S_1\,S_0\,D_3$$

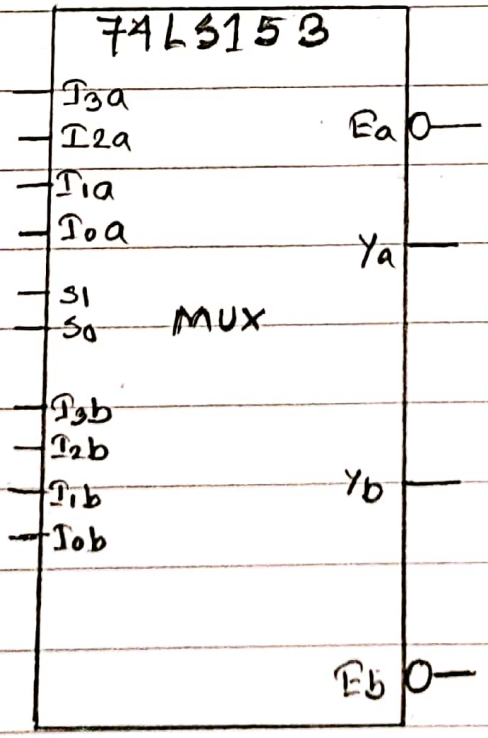
■ function table :-

Select Inputs		output
S_1	S_0	Z
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

- logic diagram :



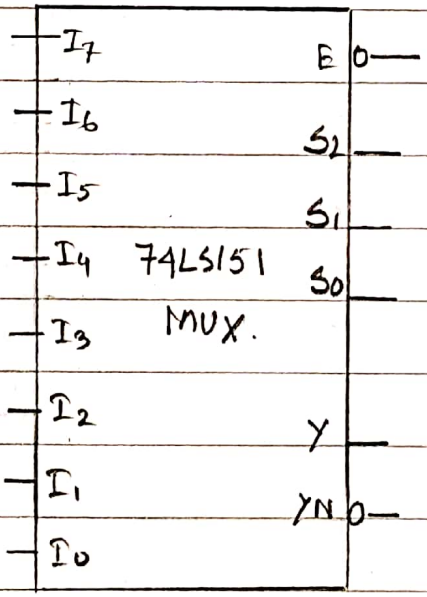
- 4 input mux using IC 74153 :-
  - Pin diagram :↓



- The 74151A - 8-Input Data Selector / multiplexer:-

This multiplexer has 8 data inputs, 3 data Select input and an enable input $\overline{E}$ (enable). A LOW on the $\overline{E}$ input allows the selected input data to pass through to the output. when $\overline{e} = 1$, the multiplexer is disable, so that $Z = 0$, regardless of the Select Input Code. note that the output as well as its Complement or available.

- pin diagram :-



Function table :-

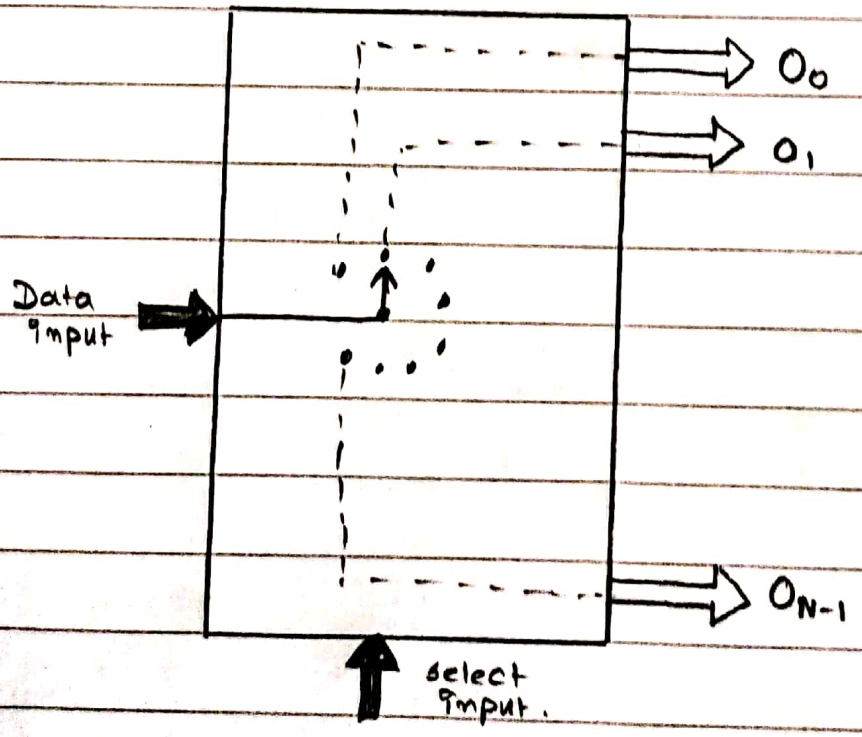
Input				output
E	S_2	S_1	S_0	Z
H	X	X	X	L
L	L	L	L	D_0
L	L	L	H	D_1
L	L	H	L	D_2
L	L	H	H	D_3
L	H	L	L	D_4
L	H	L	H	D_5
L	H	H	L	D_6
L	H	H	H	D_7

- Appication of multiplexer :-
  1. Data Selection
  2. Data routing
  3. operation Sequence
  4. parallel - to Serial Conversion
  5. wave-form generation
  6. logic function Generation.

- Demultiplexer :- A demultiplexer perform the reverse operation It takes a single Input and distributes It over several outputs. So a demultiplexer can be thought of as a 'distributor', Science It transmits the same data In different destinations.

  A demultiplexer is a 1 to N device.

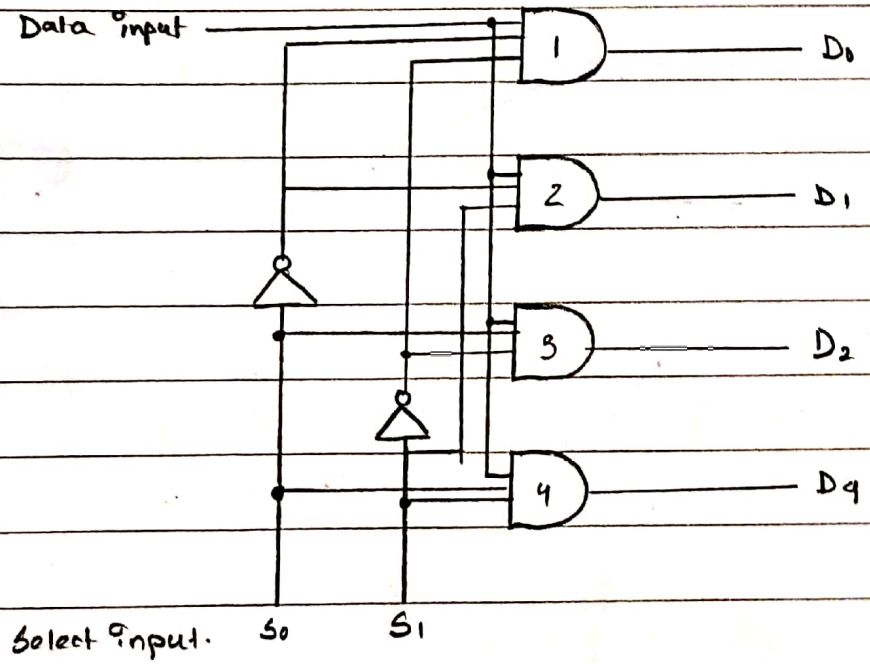
- Block Diagram :-



1-line to 4 line Demultiplexer :-

The input data line goes to all of the AND gates. The two select lines S_0 and S_1 enable only one gate at a time and the data appearing on the input line will pass through the selected gate to the associated output line.

- Logic Diagram :-



- Function Table :-

input		output			
S_1	S_0	D_0	D_1	D_2	D_4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

1 line to 8 line Demultiplexing :

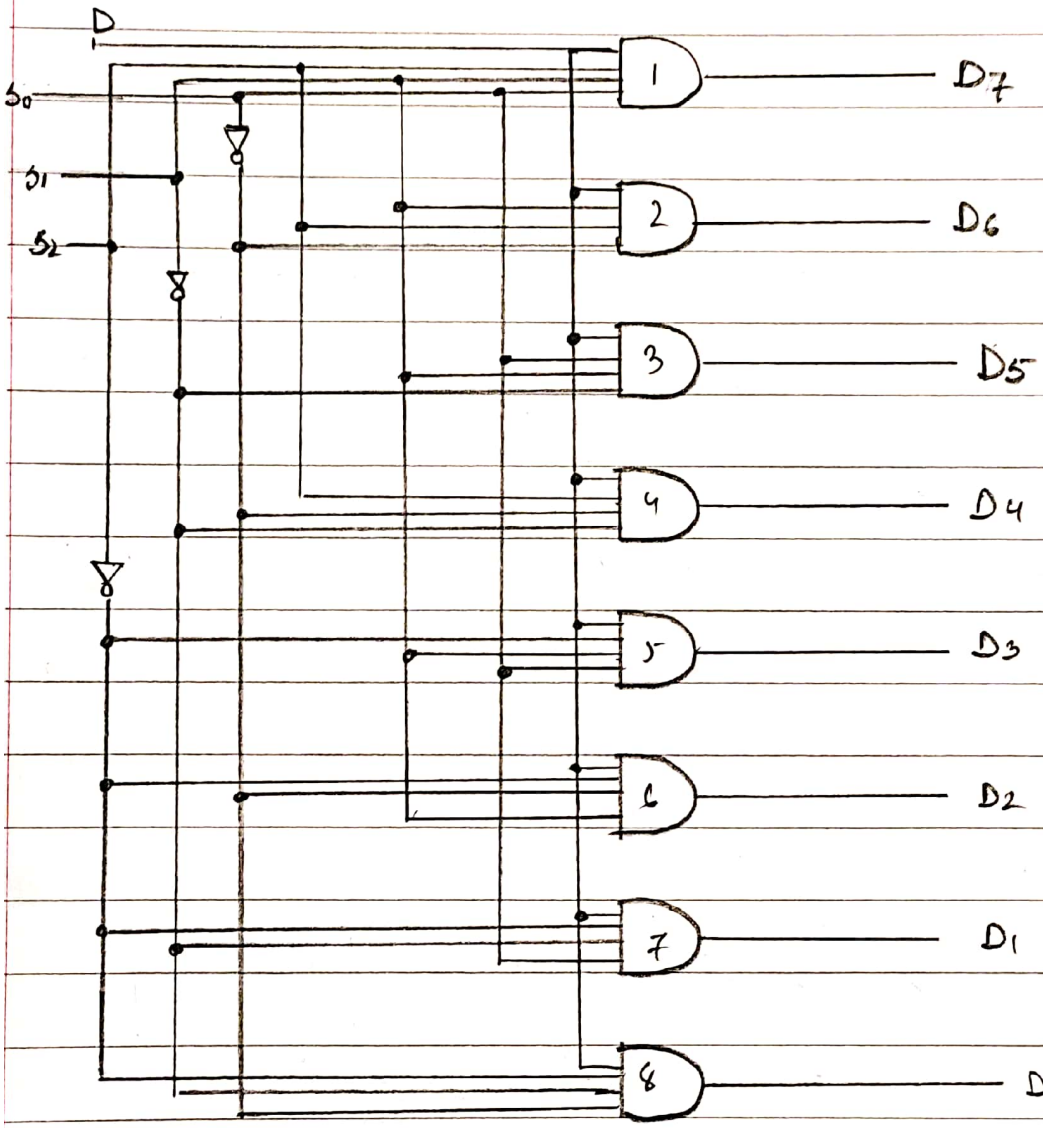
a demultiplexer that distributes one input line to eight output line, the single data input line D is connected to all eight AND gates but only one of these gates will be enabled by the Select Input Lines

For example: with $S_2 S_1 S_0 = 000$, only the AND gate O_0 will be enabled and the data input D will appear at output O_0. other selected codes cause input D to reach to other output.

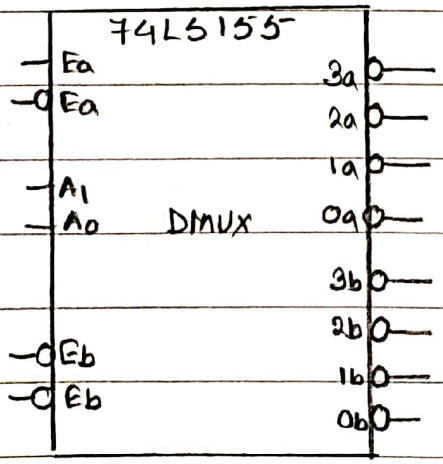
function table :-

Select Code			output							
S_2	S_1	S_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	D
0	0	1	0	0	0	0	0	0	D	0
0	1	0	0	0	0	0	0	D	0	0
0	1	1	0	0	0	0	D	0	0	0
1	0	0	0	0	0	D	0	0	0	0
1	0	1	0	0	D	0	0	0	0	0
1	1	0	0	D	0	0	0	0	0	0
1	1	1	D	0	0	0	0	0	0	0

Logic Diagram :-



■ Pin diagram LS74155 using both 1 to 4 and 1 to 8 line DMUX :-



■ Procedure :-

Step 1 : open the circuit maker 2000 Software

Step 2 : Select the required logic display under display > Digital > logic display

Step 3 : Select the logic Switch under Switches > digital > logic Switch.

Step 4 : Select the IC 74LS157 74LS153, 74LS151, 74LS155 under > (multiplexing > 157, 153, 151) and > (demux > 155)

Step 5 : use the wire tools (+) to Connect the various devices with each other as required.

Step 6 : From menu bar Select digital mode under Simulation menu.

Step 7 : press F10 key do run the test.

Step 8 : Check the output(s) of the test in accordance with the truth table of the devices and create a observation table.

Step 10 : Save the project under file > Save.

■ Observation table :- 2 Input MUX :-

E	So	I0a	I0b	Yo
1	X	X	X	L
0	L	L	X	L
0	L	H	X	H
0	H	X	L	L
0	H	X	H	H

4-input max :- Observation table :-

$\overline{E_a}$	S_0	S_1	I_a	Y_a
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1

$\overline{E_b}$	S_0	S_1	I_b	Y_b
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1

8-input max :- Observation table :-

$\overline{E}$	S_2	S_1	S_0	I	Y
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	0
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	0	1	1
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	0
0	1	1	1	1	1

1 line to 4 line Demultiplexing observration table :-

E_A	$\overline{E_A}$	A_1	A_0	$\overline{D}_0$	$\overline{D}_1$	$\overline{D}_2$	$\overline{D}_3$
1	0	0	0	0	1	1	1
1	0	0	1	1	0	1	1
1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	0

1 line to 8 Line Demultiplexing observration table :-

E_a	$\overline{E_a}/\overline{E_b}$	$\overline{E_b}$	A_1	A_0	$\overline{0}_b$	$\overline{1}_b$	$\overline{2}_b$	$\overline{3}_b$	$\overline{0}_a$	$\overline{1}_a$	$\overline{2}_a$	$\overline{3}_a$
0	0	1	0	0	0	1	1	1	1	1	1	1
0	0	1	0	1	1	0	1	1	1	1	1	1
0	0	1	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	1	1	1	0	1	1	1	1
1	0	0	0	0	1	1	1	1	0	1	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	0

Conclusion :-

from the above experiment it is Conclude that using Circuit maker 2000 Software we can Carry out any operation of any type of IC's. here we use 74155, 74151, and 74153, 74157. So we Can observed that the 'multiplexer' is a very useful function and has a multitude of

application. the selection of a particular input line is controlled by a set of selection line, normay there are 2^n input line and n selection line whose bit combinations determine which input is selected. multiplexer can be used as for 'Communication system', 'Computer memory', 'Telephone network', 'Transmission from the Computer system of a Stateliite' and the 'Demultiplexer' is also a very useful function that has one input and multiple output lines. which is used to send a signal to one of the various devices. demultiplexer can be used for, 'Communication system', 'Arithmetic logic unit' and 'Serial to parallel Converter.'

ASSIGNMENT 12

Aim :- Design and realization of the following Sequencial Circuit using logic gates ~~AND~~ / OR

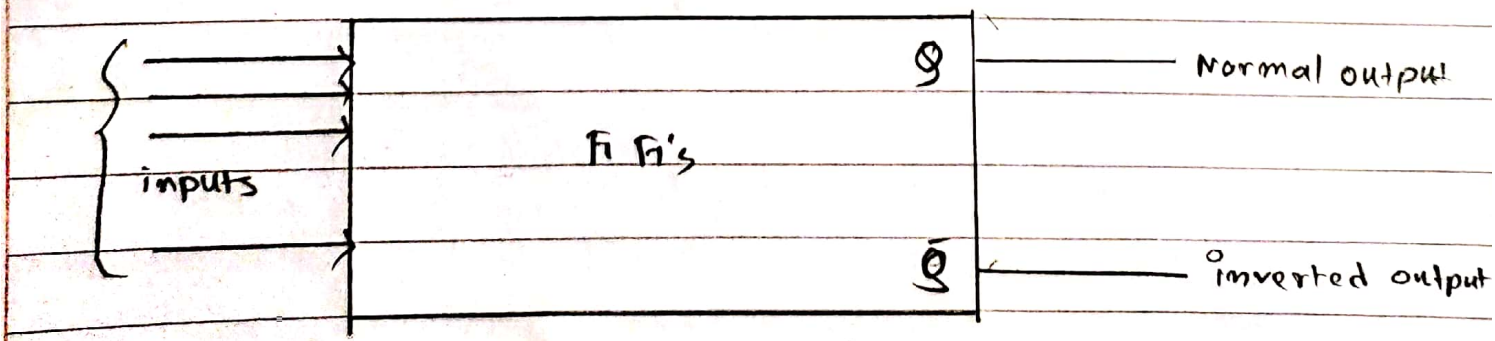
1. D-Flip-Flop
2. S-R Flip-Flop
3. J-K Flip Flop
4. ↑ Flip Flop

Requirment :- Circuit maker 2000 Software need to Installed in the Computer.

Theory :-

- **FLIP-FLOP :-** Flip-Flop are digital logic Circuit that Can be in one of two state. It's state Can be changed by applying the proper trigger Signal.

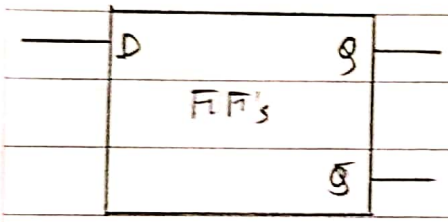
Block diagram :-



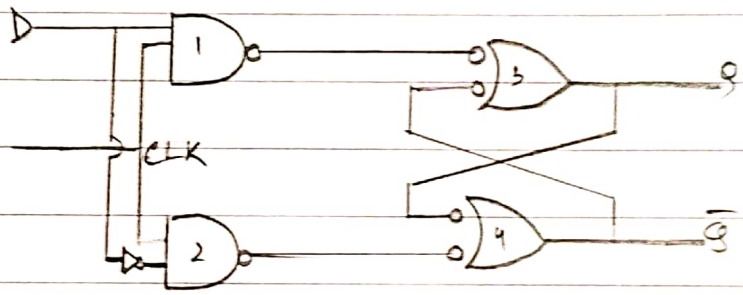
D FLIP-FLOP :-

The edge trigger D-flipflop has only one input terminal - the D flip flop may be obtained from an S-R flip flop by just putting one inverter between the S-R terminal. The FFs has only one synchronous control input in addition to the clocks. this is called the D input.

logic Symbol



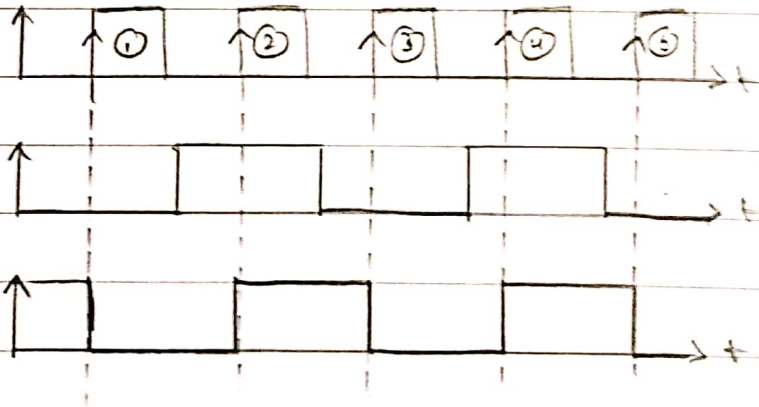
logic diagram :-



Truth table :-

Input	Clock	output	Comment
D	CLK	Q	
0	↑	0	Reset
1	↑	1	Set

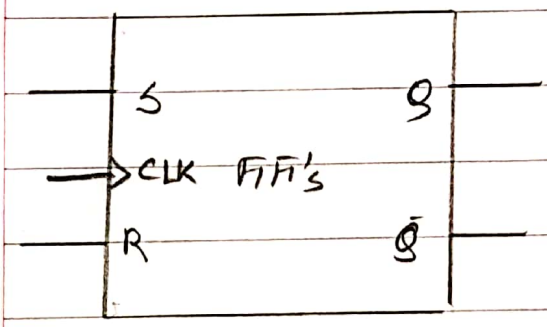
timing Diagram :-



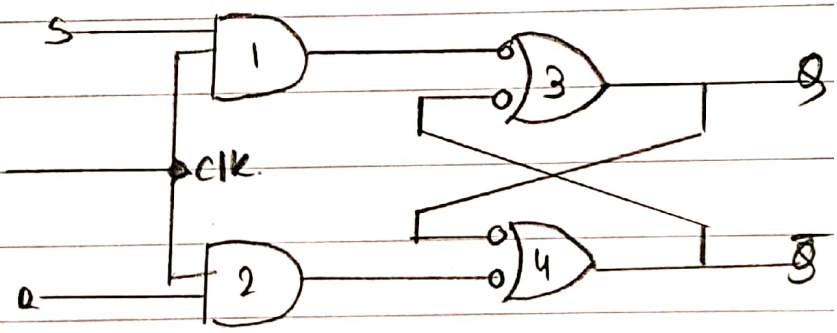
SR flip-flop :-

The S-R inputs of the S-R flip-flop are called Synchronous Control Input because data on these inputs affect the flip-flop's output only on the triggering edges of the clock pulse. without a clock pluse the SR flip flop cannot effect the output.

logic Symbol :-



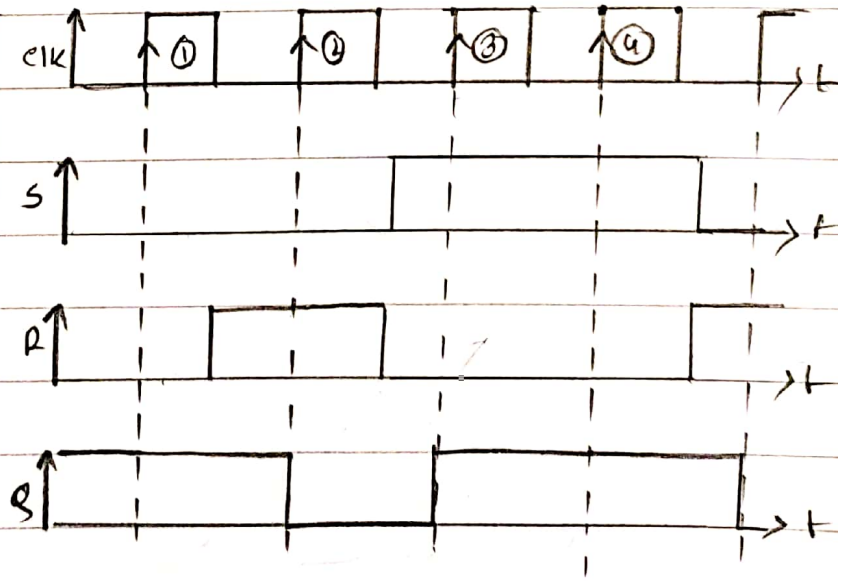
logic Diagram :-



truth table :-

Input		Clock	output	Comment
S	R	CLK	Q	
0	0	↑	Q_0	nochange.
0	1	↑	0	RESET
1	0	↑	1	SET
1	1	↑	?	Invalid

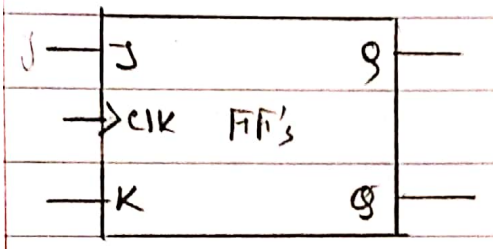
Timing Diagram :-



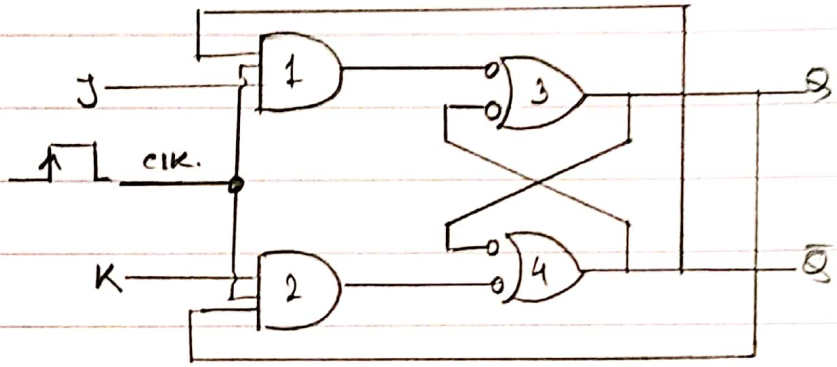
J-K Flip Flop :-

The J-K flipflop is very versatile and most widely used. It function is similar to the S-R flip-flop except that it has no invalid state like that of S-R flip-flop. when both input are high that is when J=K=1 the flip-flop toggled.

logic Symbol



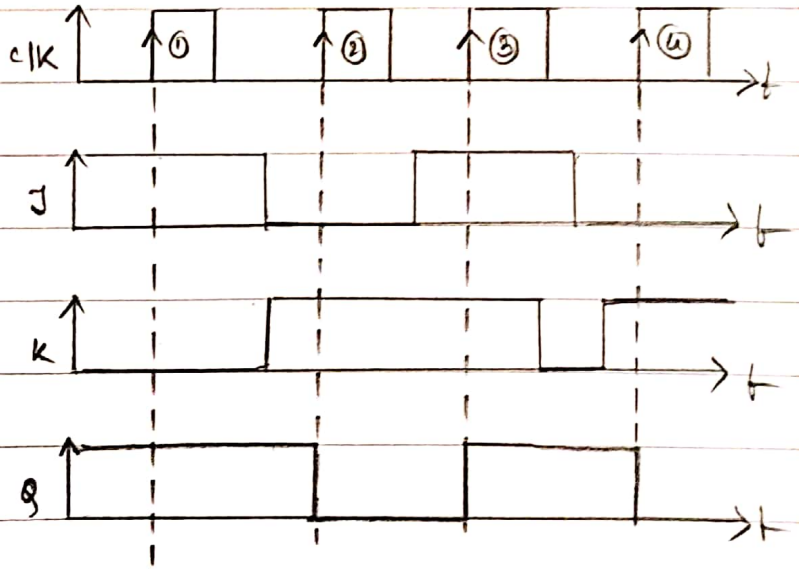
logic diagram :-



Truth table :-

input		clock	output	Comment.
J	K	CLK		
0	0	↑	Q_0	nochange
0	1	↑	0	Reset
1	0	↑	1	Set.
1	1	↑	$\bar{Q}_0$	toggol

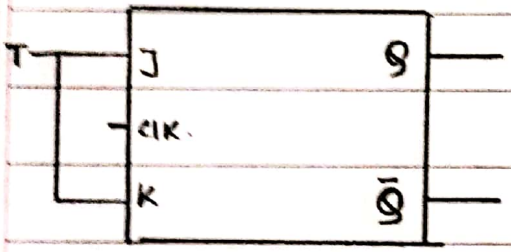
timing Diagram :-



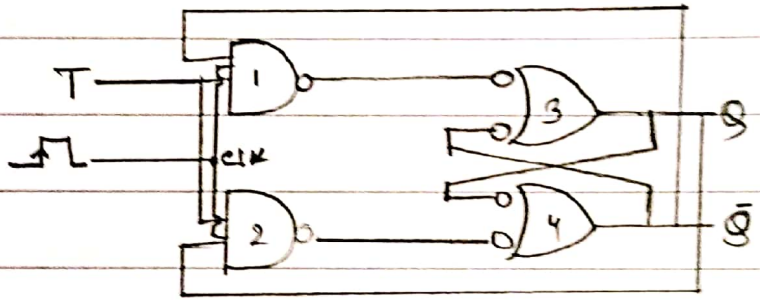
'T' flipflop :-

A 'T' flip flop has a single control input labelled 'T' for toggle. when T is high the ff toggles on every new clock pluse. when T is low the ff remain for the same state.

- logic Symbol :-



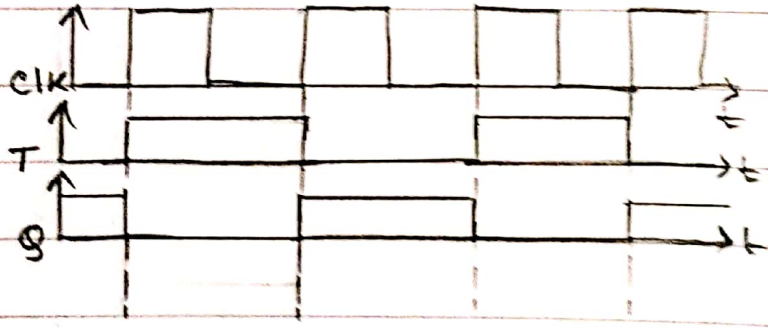
- logic diagram :-



- Truth table :-

input	output
T	Q
0	Q_0
1	$\bar{Q}_0$

- Timing Diagram.



- Procedure :-

Step 1 : open the Circuit maker 2000 Software.

Step 2 : Select the required logic gates under digital Basic > Gates.

Step 3 : Select the logic display under Display > Digital > logic display.

Step-4 :- Select the logic Switch under Switcher > digital > logic Switch.

Step-5 :- Select the Flip-Flop under Flip-Flops.

Step-6 :- use the wire tools to Connect the various devices with each other as required.

Step 7 :- From menu bar Select digital mode under Simulation menu.

Step 8 :- Press F10 key to run the test.

Step 9 :- Check the output(s) of the test in accordance with the truth table of the devices.

Step 10 :- Save the project under file > Save.

Conclusion :-

From the above experiment it is Conclude that using circuit maker 2000 Software we can Carry out any operation of any type of flip flop. So we Can Observed that the output given by a flip flop Can be fed as input to some another flip flop. and among the two output one is the normal one and the another one is it's Complement. Flip flop Can be apply as storage parallel data, storage Serial data, transfer of data, Serial to parallel, parallel to Serial Conversion, Counting, and frequency devision.