

EXPERIMENT NO: -1

I. Investigate the logic behavior of AND, OR, NOT, NAND, NOR, EX-OR gates.

AIM

To investigate the logic behavior of AND, OR, NOT, NAND, NOR, EX-OR gates.

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	OR GATE	IC 7432	1
2	AND GATE	IC 7408	1
3	NOT GATE	IC 7404	1
4	NAND GATE	IC 7400	1
5	NOR GATE	IC 7402	1
6	XOR GATE	IC 7486	1
7	BREAD BOARD	-	1
8	RESISTOR	220 Ω	1
9	BATTERY	9V	1
10	LED	-	1
11	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

Logic Gate

A logic gate is defined as a logic circuit with one output and two (or) more logic inputs. The output signal occurs only for combination of input variables.

There are basically three logic gates

- AND gate
- OR gate
- NOT gate

These three gates are called as fundamental gates, because these gates form building block for most probably all digital circuits. From these fundamental gates other three gates are derived. They are called derived gates.

- NAND gate
-

- b. NOR gate
- c. XOR gate

These logic gates are generally terms as decision making elements.

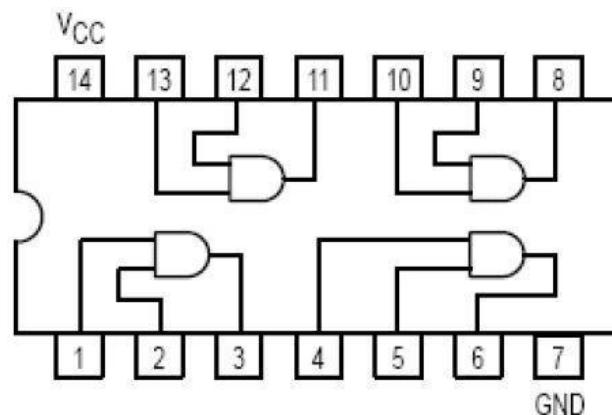
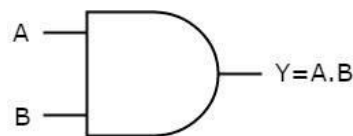
Basic Gates

AND Gate

An AND gate has two (or) more logic inputs and single output. The AND operation is defined as the output as (1) one if and only if all the inputs are (1) one. **7408** is the two inputs AND gate IC. A and B are the Input terminals & Y is the Output terminal.

Its logical equation is

$$Y = A.B$$



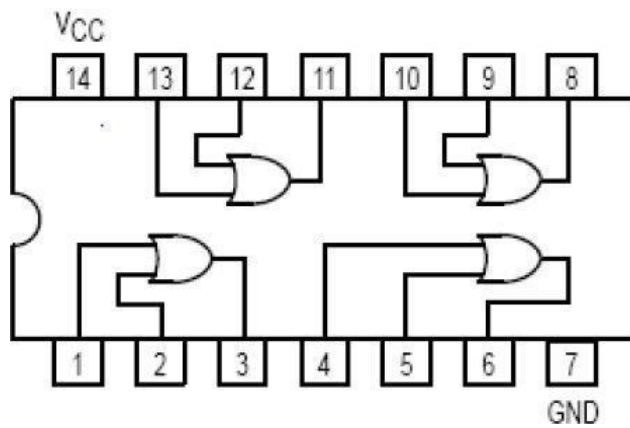
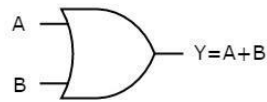
The truth table of two input AND gate is given as

INPUT A	INPUT B	OUTPUT Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

An OR gate has two (or) more logic inputs and single output. The OR operation is defined as the output as (1) one if one or more than inputs are (1) one. **7432** is the two Input OR gate IC. A & B are the input terminals & Y is the Output terminal. Its logical equation is

$$Y = A + B$$



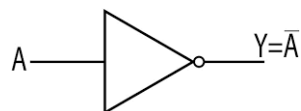
The truth table of two input OR gate is given as

INPUT A	INPUT B	OUTPUT Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT Gate

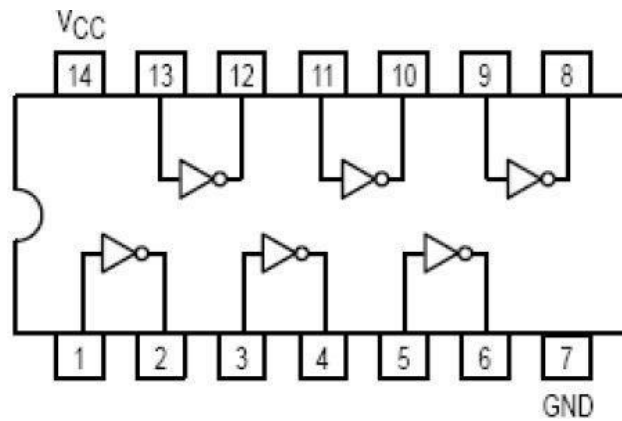
NOT gate has single input and single output. The NOT gate is also known as Inverter. It has one input (A) & one output (Y). IC No. is **7404**. Its logical equation is,

$$Y = A \text{ NOT } B, Y = A'$$



The truth table of NOT gate is given as

INPUT A	OUTPUT Y
0	1
1	0

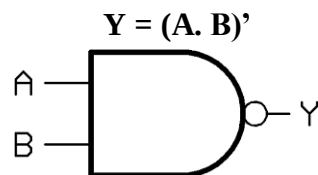


Derived Gates

NAND Gate

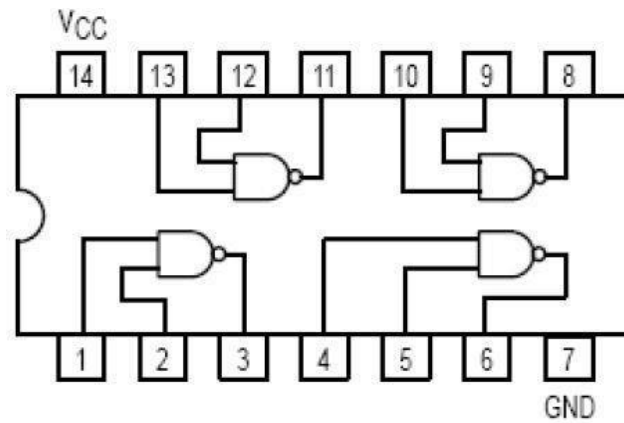
An NAND gate has two (or) more logic inputs and single output. The combination of NOT gate and an AND gate is called as NAND gate. The IC no. for NAND gate is **7400**. The NOT-AND operation is known as NAND operation. If all inputs are 1 then output produced is 0. NAND gate is inverted AND gate.

Its logical equation is



The truth table of two input NAND gate is given as

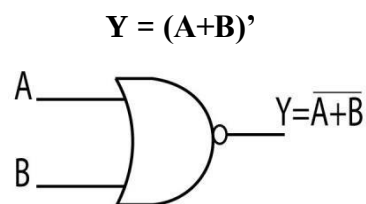
INPUT A	INPUT B	OUTPUT Y
0	0	1
0	1	1
1	0	1
1	1	0



NOR Gate

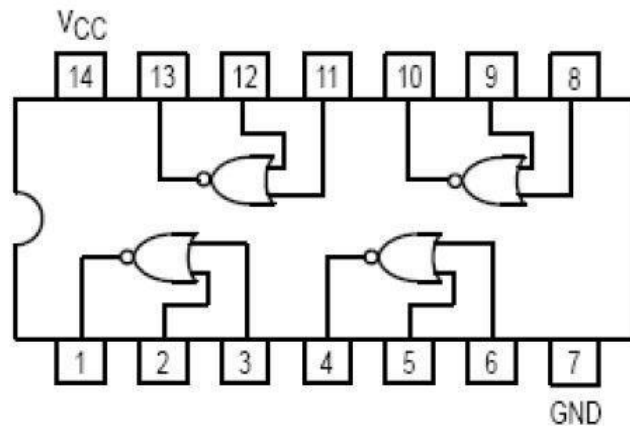
An NAND gate has two (or) more logic inputs and single output. The combination of NOT gate and an OR gate is called as NOR gate. IC **7402** is two I/P NOR gate IC. The NOT-OR operation is known as NOR operation. If all the inputs are 0 then the O/P is 1. NOR gate is inverted OR gate.

Its logical equation is



The truth table of two input NAND gate is given as

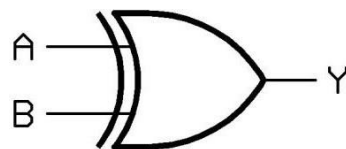
INPUT A	INPUT B	OUTPUT Y
0	0	0
0	1	1
1	0	1
1	1	1



XOR Gate

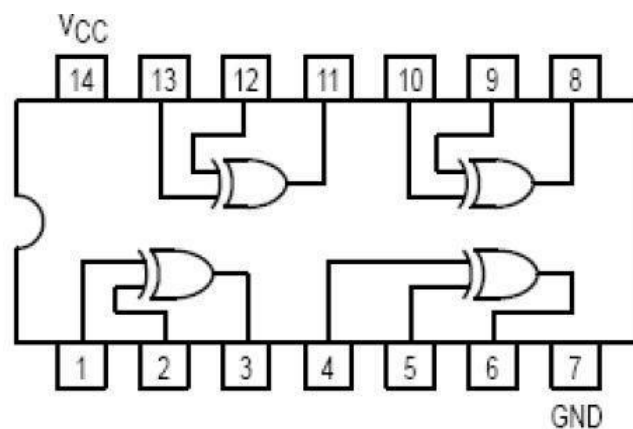
The XOR gate has two (or) more logic inputs and single output. **7486** is two inputs XOR gate IC. EX-OR gate is not a basic operation & can be performed using basic gates. Its logical equation is

$$Y = A'B + AB'$$



The truth table of two input XOR gate is given as

INPUT A	INPUT B	OUTPUT Y
0	0	0
0	1	1
1	0	1
1	1	0



Universal gate

The NAND and NOR gates are called Universal gates, because all the logic gates and logic functions can be implemented using NAND and NOR gates.

PROCEDURE

2 Input AND Gate

1. Position the IC [7408] properly on the solder less bread board.
2. Connect the pin 7 to ground using patch chord provided.
3. Connect the pin 14 to +5V supply using patch chord provided.
4. Connect the pin 1 and pin 2 to logic input of low (or) high.
5. Connect the pin 3 to logic output of LED indicator.
6. Give the logic input low (or) high and note the output through LED indicator.
7. Verify the truth table.

2 Input OR Gate

1. Position the IC [7432] properly on the solder less bread board.
2. Connect the pin 7 to ground using patch chord provided.
3. Connect the pin 14 to +5V supply using patch chord provided.
4. Connect the pin 1 and pin 2 to logic input of low (or) high.
5. Connect the pin 3 to logic output of LED indicator.
6. Give the logic input low (or) high and note the output through LED indicator.
7. Verify the truth table.

NOT Gate

1. Position the IC [7404] properly on the solder less bread board.
2. Connect the pin 7 to ground using patch chord provided.
3. Connect the pin 14 to +5V supply using patch chord provided.
4. Connect the pin 1 to logic input of low (or) high.
5. Connect the pin 2 to logic output of LED indicator.
6. Give the logic input low (or) high and note the output through LED indicator.
7. Verify the truth table.

2 Input NAND Gate

1. Position the IC [7400] properly on the solder less bread board.
 2. Connect the pin 7 to ground using patch chord provided.
 3. Connect the pin 14 to +5V supply using patch chord provided.
 4. Connect the pin 1 and pin 2 to logic input of low (or) high.
 5. Connect the pin 3 to logic output of LED indicator.
-

6. Give the logic input low (or) high and note the output through LED indicator.
7. Verify the truth table.

2 Input NOR Gate

1. Position the IC [7402] properly on the solder less bread board.
2. Connect the pin 7 to ground using patch chord provided.
3. Connect the pin 14 to +5V supply using patch chord provided.
4. Connect the pin 2 and pin 3 to logic input of low (or) high.
5. Connect the pin 1 to logic output of LED indicator.
6. Give the logic input low (or) high and note the output through LED indicator.
7. Verify the truth table.

2 Input XOR Gate

1. Position the IC [7486] properly on the solder less bread board.
2. Connect the pin 7 to ground using patch chord provided.
3. Connect the pin 14 to +5V supply using patch chord provided.
4. Connect the pin 2 and pin 3 to logic input of low (or) high.
5. Connect the pin 1 to logic output of LED indicator.
6. Give the logic input low (or) high and note the output through LED indicator.
7. Verify the truth table.

PRECAUTIONS:

1. Make the connections according to the IC pin diagram.
2. The connections should be tight.
3. The V_{CC} and ground should be applied carefully at the specified pin only.

OBSERVATION

AND Gate [L=logic 0, H=logic 1]

INPUT A	INPUT B	OUTPUT Y
L	L	
L	H	
H	H	
H	L	

OR Gate [L=logic 0, H=logic 1]

INPUT A	INPUT B	OUTPUT Y
L	L	
L	H	
H	L	
H	H	

NOT Gate [L=logic 0, H=logic 1]

INPUT A	OUTPUT Y
L	
H	

NAND Gate [L=logic 0, H=logic 1]

INPUT A	INPUT B	OUTPUT Y
L	L	
L	H	
H	L	
H	H	

NOR Gate [L=logic 0, H=logic 1]

INPUT A	INPUT B	OUTPUT Y
L	L	
L	H	
H	L	
H	H	

XOR Gate [L=logic 0, H=logic 1]

INPUT A	INPUT B	OUTPUT Y
L	L	
L	H	
H	L	
H	H	

RESULT: The logic behavior of AND, OR, NOT, NAND, NOR, EX-OR gates is verified.

QUESTIONS

1. Define Logic gates?
 2. Define IC?
 3. Define Universal gates?
 4. How many no. of input variables can a NOT Gate have?
 5. When will the output of a NAND Gate be 0?
-

II. Study and Verify NAND gate as a Universal Gate

AIM

To Study and Verify NAND gate as a Universal Gate.

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	NAND GATE	IC 7400	1
2	BREAD BOARD	-	1
3	RESISTOR	220 Ω	1
4	BATTERY	9V	1
5	LED	-	1
6	CONNECTING WIRE	-	AS PER REQUIREMENT

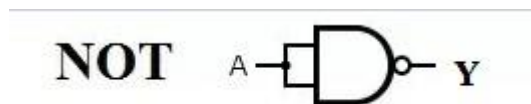
THEORY

Universal gate

The NAND and NOR gates are called Universal gates, because all the logic gates and logic functions can be implemented using NAND and NOR gates.

NAND Gate as Universal gate

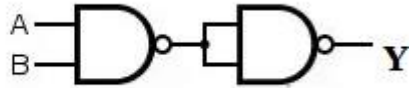
- a. **NAND GATE AS INVERTER:** The circuit diagram of implementation of NAND gate as inverter is shown below:



INPUT A	OUTPUT Y
0	1
1	0

- b. **NAND GATE AS AND GATE:** The circuit diagram of implementation of NAND gate as AND gate is shown below:
-

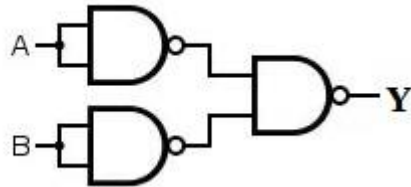
AND



INPUT A	INPUT B	OUTPUT Y
0	0	0
0	1	0
1	0	0
1	1	1

- c. **NAND GATE AS OR GATE:** The circuit diagram of implementation of NAND gate as OR gate is shown below:

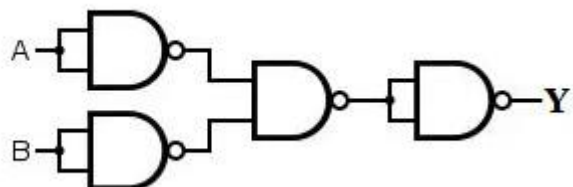
OR



INPUT A	INPUT B	OUTPUT Y
0	0	1
0	1	1
1	0	1
1	1	0

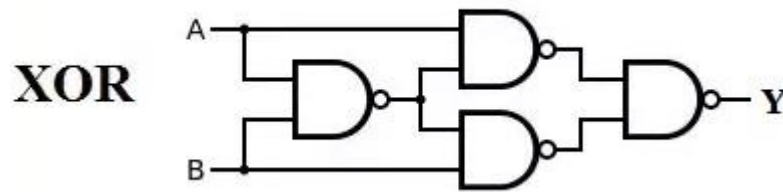
- d. **NAND GATE AS NOR GATE:** The circuit diagram of implementation of NAND gate as is shown below:

NOR



INPUT A	INPUT B	OUTPUT Y
0	0	1
0	1	0
1	0	0
1	1	0

- e. **NAND GATE AS EX-OR GATE** The circuit diagram of implementation of NAND gate as is shown below:



INPUT A	INPUT B	OUTPUT Y
0	0	0
0	1	1
1	0	1
1	1	0

PROCEDURE:

1. Make the connections as per the logic diagram.
2. Connect +5v to pin 14 & ground to pin 7.
3. Apply diff combinations of inputs to the i/p terminals
4. Note o/p for NAND as universal gate.
5. Verify the truth table.

RESULT: - The NAND gate as a Universal Gate is verified.

EXPERIMENT NO: -2

I. Implementation of the Given Boolean Function using Logic Gates.

AIM

To implement the given Boolean Function using Logic Gates.

$$Y = AB + A'B$$

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	NAND GATE	IC 7400	1
2	OR GATE	IC 7432	1
3	AND GATE	IC 7408	1
2	BREAD BOARD	-	1
3	RESISTOR	220Ω	1
4	BATTERY	9V	1
5	LED	-	1
6	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

It is desirable to connect gates together in as few a number as possible to create desired output result given a fixed set of input conditions. Alternatively, it may be necessary to utilize one type of gate to produce several other logic functions. Purchasing one IC type in bulk quantity has the advantage of reducing the cost of these chips. Two theories are used to facilitate these objectives.

- i) Boolean Algebra and
- ii) Demorgan's Theorem

Boolean algebra utilizes rules based on logic gate operations.

Karnaugh's Map Reduction Technique

The implication of logic circuits is based on Karnaugh's map. There are two fundamental approaches in logic design.

- i. Product of Sum method
- ii. Sum of product method

The Sum of product solution results in NAND circuit, while the Product of Sum solution results in NOR circuit corresponding to a given truth table.

The designers can select either method, usually selects the simpler circuit because it costs less and more reliable.

The complexity of the digital logic gates can be implemented in a Boolean function. The map method provides a simple straightforward procedure for minimizing the Boolean functions. This map method is known as “Karnaugh’s map”.

The map is the diagram made up squares. Each square represents one minterm. Since any Boolean functions can be expressed as a sum of minterm. The simplest algebraic expression is any one from a sum of products or product of sums that has a minimum number of literals.

Two variable map has 4 minterm. It can be represented as

m0	m1
m2	m3

$\bar{x}\bar{y}$	$\bar{x}y$
$x\bar{y}$	xy

Two Variable Map

A three variable map has 8 minterms for the three variables. This map has 8 squares. Three variable map is represents as shown below.

m0	m1	m3	m2
m4	m5	m7	m6

$\bar{x}\bar{y}\bar{z}$	$\bar{x}\bar{y}z$	$\bar{x}yz$	$\bar{x}y\bar{z}$
$x\bar{y}\bar{z}$	$x\bar{y}z$	xyz	$xy\bar{z}$

Three Variable Map

PROCEDURE

1. Position the IC’s [7408, 7432, and 7404] properly on the solder less bread board.
 2. Connect the pin 7 of all the chips to ground using patch chord provided.
 3. Connect the pin 14 of all the chips to +5V supply using patch chord provided.
 4. Input A signal and B signals are connected at pin no 1 and 2 of the IC 7408.
 5. Input A signal is complemented by connecting pin 1 of the IC 7404.
 6. The inverted input A and B signals are connected to the pin 4 and 5 of IC 7432.
 7. The output (7408) pin 3 and 6 are concerted to input pin 1 and 2 of OR gate.
 8. The output pin 3 of the OR gate give the Boolean function result.
-

OBSERVATION [L=logic 0, H=logic 1]

INPUT A	INPUT B	$Y = AB + A'B$
L	L	
L	H	
H	L	
H	H	

RESULT: The given Boolean Function using Logic Gates is verified.

II. Verification of Demorgan's theorem for 2 variables

AIM

To verify Demorgan's theorem for 2 variables.

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	NAND GATE	IC 7400	1
2	OR GATE	IC 7432	1
3	AND GATE	IC 7408	1
2	BREAD BOARD	-	1
3	RESISTOR	220Ω	1
4	BATTERY	9V	1
5	LED	-	1
6	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

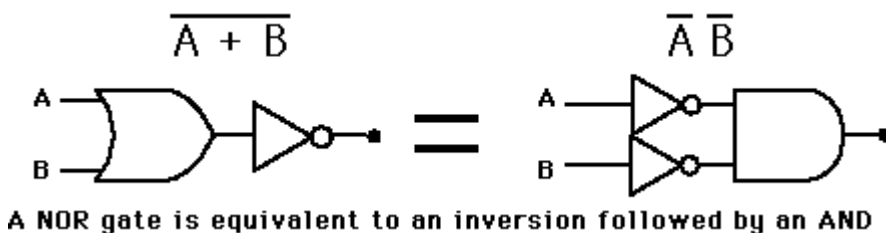
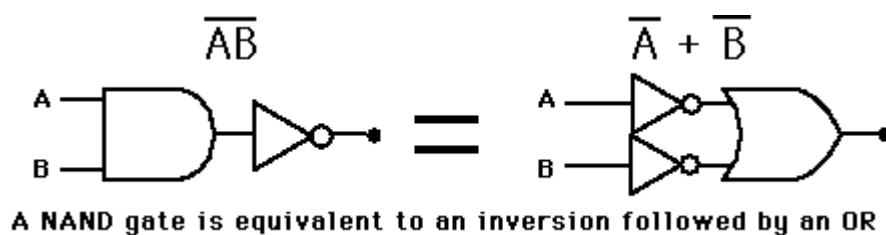
Theorem 1: The compliment of the product of two variables is equal to the sum of the compliment of each variable. Thus according to De-Morgan's laws or De-Morgan's theorem if A and B are the two variables or Boolean numbers. Then accordingly,

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

Theorem 2:-

The compliment of the sum of two variables is equal to the product of the compliment of each variable. Thus according to De Morgan's theorem if A and B are the two variables then,

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



De-Morgan's laws can also be implemented in Boolean algebra in the following steps:

1. While doing Boolean algebra at first replace the given operator. That is (+) is replaced with (.) and (.) is replaced with (+).
2. Compliment of each of the term is to be found.

PROCEDURE

1. Realize the Demorgan's theorem using logic gates.
2. Connect V_{CC} and ground as shown in the pin diagram.
3. Make connections as per the logic gate diagram.
4. Apply the different combinations of input according to the truth tables and verify that the results are correct.

OBSERVATION [L=logic 0, H=logic 1]

Theorem 1

INPUT A	INPUT B	$A \cdot B$	$A + B$
L	L		
L	H		
H	L		
H	H		

Theorem 2

INPUT A	INPUT B	$A \oplus B$	$A \cdot B$
L	L		
L	H		
H	L		
H	H		

RESULT: The truth table of Demorgan's theorem for 2 variables is verified.

QUESTIONS

1. Define K-map?
 2. Define SOP?
 3. Define POS?
 4. What are combinational circuits?
 5. If there are four variables how many cells the K-map will have?
 6. Which code is used for the identification of cells?
-

EXPERIMENT NO: -3

I. Realize binary adder circuits using logic gates.

AIM

To realize half adder using Logic gates.

APPARATUS REQUIRED

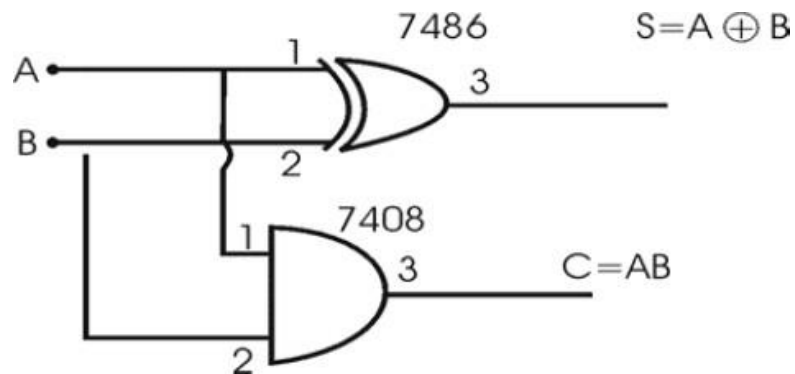
SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	XOR GATE	IC 7486	1
2	AND GATE	IC 7408	1
3	BREAD BOARD	-	1
4	RESISTOR	220 Ω	1
5	BATTERY	9V	1
6	LED	-	1
7	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

Half-Adder: A combinational logic circuit that performs the addition of two data bits, A and B, is called a half-adder. Addition will result in two output bits; one of which is the sum bit S, and the other is the carry bit, C. The Boolean functions describing the half-adder are:

$$S = A \oplus B$$

$$C = A.B$$



Truth Table

INPUT A	INPUT B	$Y = A \oplus B$	$C=A.B$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

PROCEDURE: -

1. Position the IC's [7486, 7408] properly on the solder less bread board.
2. Connect the pin 7 of all the chips to ground using patch chord provided.
3. Connect the pin 14 of all the chips to +5V supply using patch chord provided.
4. Connect the circuit as shown and get the output of Sum and Carry separately.
5. Take input from pin no. 1&2 of IC no.7486 and take output at pin no.3.
6. Pin no.3 is connected with LED.
7. Short pin no.1 of IC no.7486 to pin no.1 of IC no.7408.
8. Similarly, short pin no.2 of IC no.7486 to pin no.2 of IC no.7408.
9. Take output at Pin no.3 of IC no.7408 and connect to LED.

PRECAUTIONS: -

1. Supply should not exceed 5V.
2. Connections should be tight and easy to inspect.
3. Use L.E.D. with proper sign convention and check it before connecting in circuit.

OBSERVATION [L=logic 0, H=logic 1]

INPUT A	INPUT B	$Y = A \oplus B$	$C=A.B$
L	L		
L	H		
H	L		
H	H		

RESULT: -The truth table of half adder is verified.

AIM

To realize full adder using Logic gates.

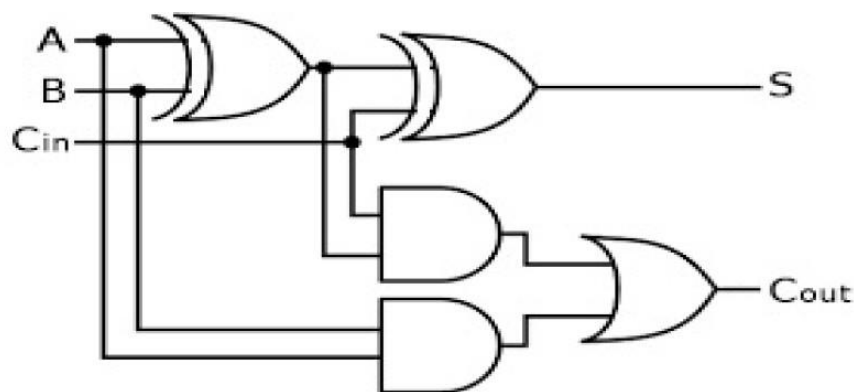
APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	XOR GATE	IC 7486	1
2	AND GATE	IC 7408	1
3	OR GATE	IC 7432	
3	BREAD BOARD	-	1
4	RESISTOR	220 Ω	1
5	BATTERY	9V	1
6	LED	-	1
7	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

Full Adder: The half-adder does not take the carry bit from its previous stage into account. This carry bit from its previous stage is called carry-in bit. A combinational logic circuit that adds two data bits, A and B, and a carry-in bit, C_{in} is called a full-adder. The Boolean functions describing the full-adder are:

$$S = A \oplus B \oplus C_{in}$$
$$C = A.B + C_{in}(A \oplus B)$$



Truth Table

INPUT A	INPUT B	INPUT C_{in}	$S = A \oplus B \oplus C_{in}$	$C = A.B + C_{in}(A \oplus B)$
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

PROCEDURE: -

1. Position the IC's [7486, 7408, and 7432] properly on the solder less bread board.
2. Connect the pin 7 of all the chips to ground using patch chord provided.
3. Connect the pin 14 of all the chips to +5V supply using patch chord provided.
4. Connect the circuit as shown and get the output of Sum and Carry separately.
5. Make the connections as per the circuit diagram for the full adder circuit, on the trainer kit.
6. Switch on the V_{CC} power supply and apply the various combinations of the inputs according to the respective truth tables.
7. Verify that the outputs are according to the expected results.

OBSERVATION [L=logic 0, H=logic 1]

INPUT A	INPUT B	INPUT C_{in}	$S = A \oplus B \oplus C_{in}$	$C = A.B + C_{in}(A \oplus B)$
L	L	L		
L	L	H		
L	H	L		
L	H	H		
H	L	L		
H	L	H		
H	H	L		
H	H	H		

RESULT: -The truth table of full adder is verified.

QUESTIONS

1. Give the basic rules for binary addition?
 2. Specify the no. of I/P and O/P of Half adder?
 3. What is the drawback of half adder?
 4. Write the equation for sum & carry of half adder?
 5. Write the equation for sum & carry of full adder?
 6. How many half adders will be required for implementing full adder?
 7. Define Bit?
-

II. To realize subtractor circuits using Logic gates.

AIM

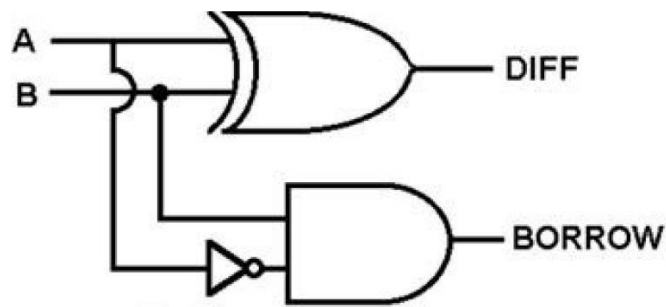
To realize half subtractor using Logic gates.

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	XOR GATE	IC 7486	1
2	AND GATE	IC 7408	1
3	NOT GATE	IC 7404	
3	BREAD BOARD	-	1
4	RESISTOR	220Ω	1
5	BATTERY	9V	1
6	LED	-	1
7	CONNECTING WIRE	-	AS PER REQUIREMENT

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 B₀. This operation is called half subtraction and the circuit to realize it is called a half subtractor. The Boolean functions describing the half-subtractor are:



$$D = A \oplus B$$

$$B_0 = A'B$$

Truth Table

INPUT A	INPUT B	$D = A \oplus B$	$B_0 = A'B$
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	0

PROCEDURE: -

1. Position the IC's [7408, 7404, and 7432] properly on the solder less bread board.
2. Connect the pin 7 of all the chips to ground using patch chord provided.
3. Connect the pin 14 of all the chips to +5V supply using patch chord provided.
4. Connect the circuit as shown and get the output of difference and borrow separately.
5. Take input from pin no. 1&2 of IC no.7486 and take output at pin no.3.
6. Pin no.3 is connected with LED.
7. Short pin no.1 of IC no.7486 to pin no.1 of IC no.7404 and take output at pin no.2 of IC no 7404.
8. Short pin no.2 of IC no.7404 to pin no.1 of IC no.7408.
9. Similarly, short pin no.2 of IC no.7486 to pin no.2 of IC no.7408.
10. Take output at Pin no.3 of IC no.7408 and connect to LED.

OBSERVATION [L=logic 0, H=logic 1]

INPUT A	INPUT B	$D = A \oplus B$	$B_0 = A'B$
L	L		
L	H		
H	L		
H	H		

RESULT: -The truth table of half subtractor is verified.

AIM

To realize full subtractor using Logic gates.

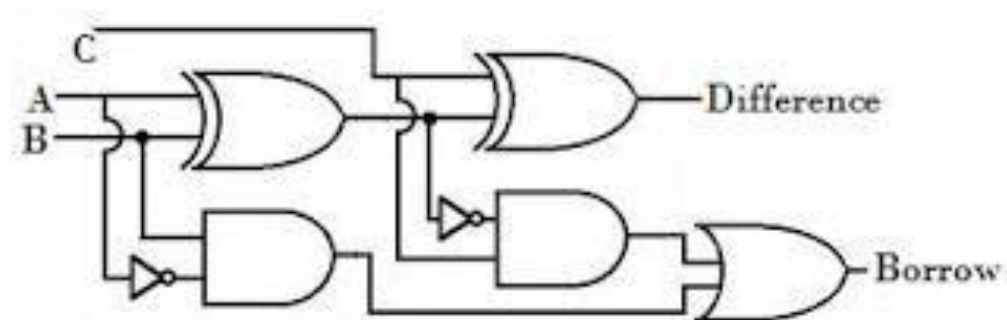
APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	XOR GATE	IC 7486	1
2	AND GATE	IC 7408	1
3	OR GATE	IC 7432	
4	NOT GATE	IC 7404	
5	BREAD BOARD	-	1
6	RESISTOR	220Ω	1
7	BATTERY	9V	1
8	LED	-	1
9	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

Full-subtractor: Subtracting two single-bit binary values, B, C from a single-bit value A produces a difference bit D and a borrow out B₀ bit. This is called full subtraction. The Boolean functions describing the full subtractor.

$$D = A \oplus B \oplus C$$
$$B_0 = A'B + A'C + BC$$



Truth Table

INPUT A	INPUT B	INPUT C_{in}	$D = A \oplus B \oplus C$	$B_0 = A'B + A'C + BC$
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	0	1

PROCEDURE: -

1. Position the IC's [7486, 7408, 7402 and 7432] properly on the solder less bread board.
2. Connect the pin 7 of all the chips to ground using patch chord provided.
3. Connect the pin 14 of all the chips to +5V supply using patch chord provided.
4. Connect the circuit as shown and get the output of Difference and Borrow separately.
5. Make the connections as per the circuit diagram for the full subtractor circuit, on the trainer kit.
6. Switch on the V_{CC} power supply and apply the various combinations of the inputs according to the respective truth tables.
7. Verify that the outputs are according to the expected results.

OBSERVATION [L=logic 0, H=logic 1]

INPUT A	INPUT B	INPUT C_{in}	$D = A \oplus B \oplus C$	$B_0 = A'B + A'C + BC$
L	L	L		
L	L	H		
L	H	L		
L	H	H		
H	L	L		
H	L	H		
H	H	L		
H	H	H		

RESULT: -The truth table of full subtractor is verified.

QUESTIONS

1. What is half subtractor?
 2. For implementing half subtractor how many EX-OR, AND gates and Not gates are required?
 3. What are the logical equations for difference & borrow?
 4. How full subtractor is different from half subtractor.
 5. How many bits we use in half subtractor for subtraction?
-

EXPERIMENT NO: -4

Design, implement and test a given design example with (i) NAND Gates only (ii) NOR Gates only

AIM

To design, implement and test a given design example with (i) NAND Gates only (ii) NOR Gates only.

APPARATUS REQUIRED

SL NO.	COMPONENT	SPECIFICATION	QUANTITY
1	NAND GATE	IC 7400	1
2	BREAD BOARD	-	1
3	RESISTOR	220Ω	1
4	BATTERY	9V	1
5	LED	-	1
6	CONNECTING WIRE	-	AS PER REQUIREMENT

THEORY

Canonical Forms (Normal Forms): Any Boolean function can be written in disjunctive normal form (sum of min-terms) or conjunctive normal form (product of max-terms). A Boolean function can be represented by a Karnaugh map in which each cell corresponds to one minterm/ maxterm.

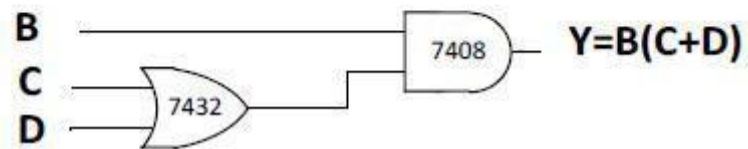
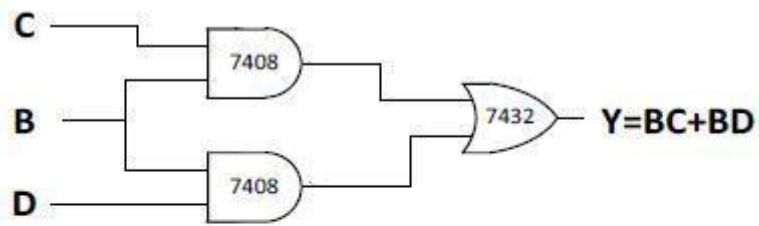
1. Sum of minterms: Sum of Product (SOP)
2. Product of maxterms: Product of Sum (POS)

Simplified Boolean expression:

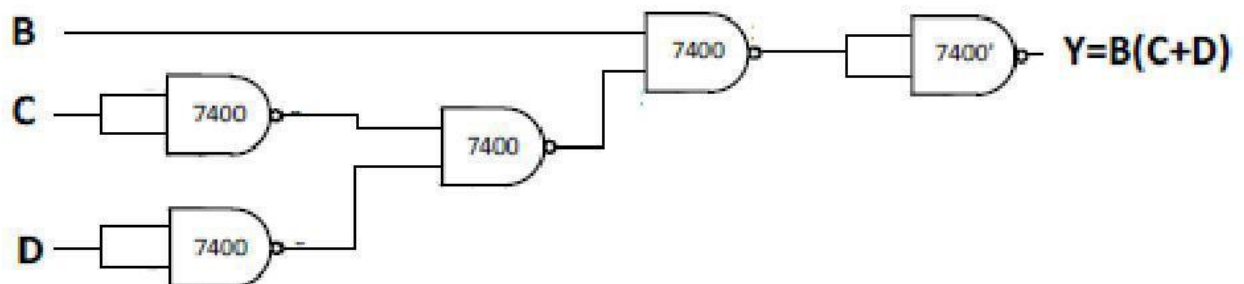
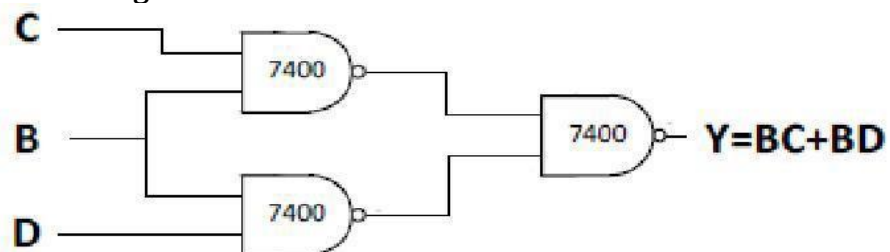
$$\text{SOP form } Y = f(A, B, C, D) = BC + BD$$

$$\text{POS form } Y = f(A, B, C, D)$$

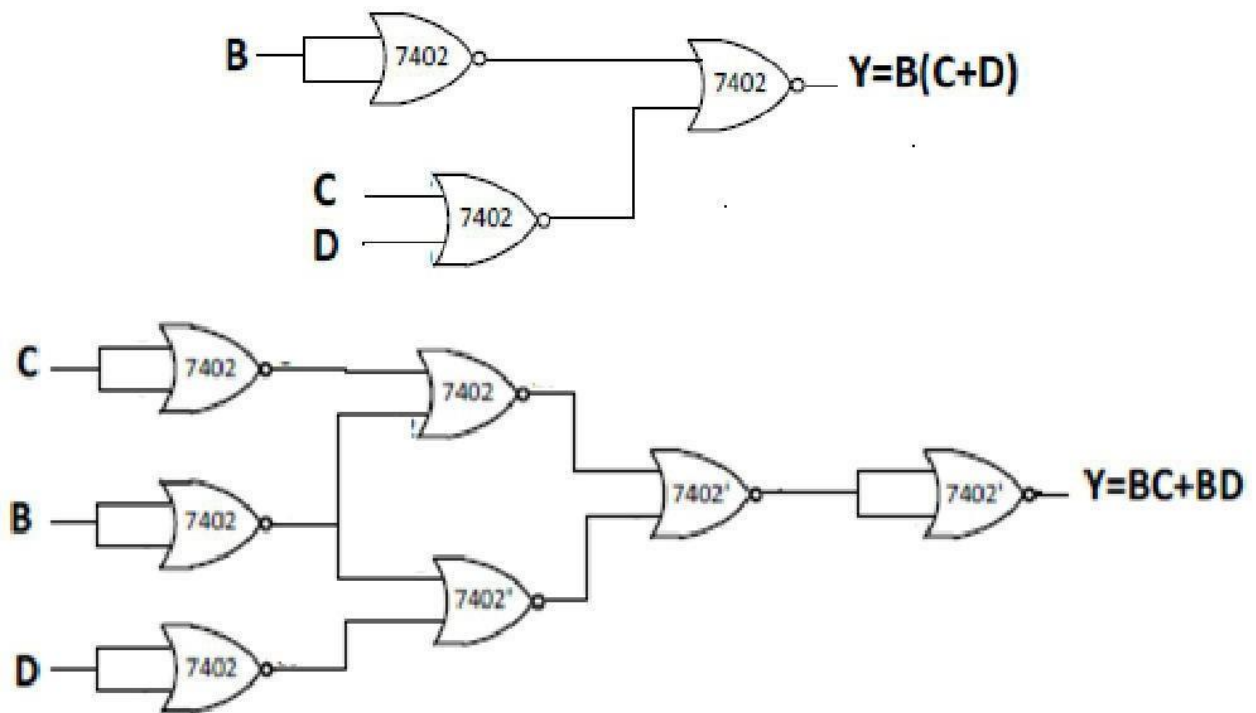
(i) Simplification using Basic Gates:



(ii) Simplification using NAND Gate:



(iii) Simplification using NOR Gate:



PROCEDURE

1. Verify that the gates are working.
2. Construct a truth table for the given problem.
3. Draw a Karnaugh's Map corresponding to the given truth table.
4. Simplify the given Boolean expression manually using the Karnaugh's Map.

a. Implementation Using Logic Gates:

1. Realize the simplified expression using logic gates.
2. Connect V_{CC} and ground as shown in the pin diagram.
3. Make connections as per the logic gate diagram.
4. Apply the different combinations of input according to the truth tables.
5. Verify that the results are correct.

b. Implementation Using Universal Gates:

1. Convert the AND-OR logic into NAND-NAND and NOR-NOR logic.
2. Realize the simplified Boolean expressions using only NAND gates, and Then by using only NOR gates.
3. Connect the circuits according to the circuit diagrams, apply inputs according to the truth table and verify the results.

RESULTS: - The given design example is implemented using NAND gates only.

QUESTIONS

1. Define universal gates?
2. Which IC is used for NAND and NOR gates?
3. Why NAND-NOR gates are called as universal gate?
4. Design XOR gate using NAND gate.
5. Design NAND gate using NOR gate.