DIGITAL SYSTEMS LABORATORY

YEAR/SEM: II/III **Date :09.07.2025**

Faculty Name: Dr.P.Mohamed Fathimal

List of Experiments

Unit I

1. **Verification of Boolean Theorems using basic gates**
2. **Design and implementation of combinational circuits using basic gates and universal gates for arbitrary functions**

Unit II

3. **Design and implementation of Parity generator / checker.**
4. **Design and implementation of Magnitude Comparator.**
5. **Design and implementation of Code converters.**
6. **Design and implementation of an application using multiplexers.**
7. **Combinational circuits using HDL.**

Unit III

8. **Design and implementation of shift –registers.**
9. **2. Design and implementation of synchronous counters.**
10. **3. Sequential circuits using HDL.**

Unit IV

11. **Design and implementation of asynchronous counters.**

Unit V

12. **Design and implementation of a simple digital system.**

Unit I

Experiment 1: To study logic gates and verify their truth tables.

APPARATUS REQUIRED:

SL.NO.	COMPONENT	SPECIFICATION	QTY
1.	AND GATE	IC 7408	1
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
4.	NAND GATE 2 I/P	IC 7400	1
5.	NOR GATE	IC 7402	1
6.	X-OR GATE	IC 7486	1
7.	NAND GATE 3 I/P	IC 7410	1
8.	IC TRAINER KIT	-	1
9.	PATCH CORD	-	14

THEORY:

Circuit that takes the logical decision and the process are called logic gates. Each gate has one or more input and only one output.

OR, AND and NOT are basic gates. NAND, NOR and X-OR are known as universal gates. Basic gates form these gates.

AND GATE:

The AND gate performs a logical multiplication commonly known as AND function. The output is high when both the inputs are high. The output is low level when any one of the inputs is low.

OR GATE:

The OR gate performs a logical addition commonly known as OR function. The output is high when any one of the inputs is high. The output is low level when both the inputs are low.

NOT GATE:

The NOT gate is called an inverter. The output is high when the input is low. The output is low when the input is high.

AND GATE:

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

NOR GATE:

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

X- OR GATE:

The output is high when any one of the inputs is high. The output is low when both the inputs are low and both the inputs are high.

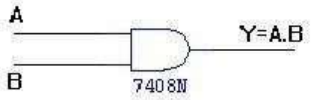
PROCEDURE:

(i) Connections are given as per circuit diagram.
(ii) Logical inputs are given as per circuit diagram.
(iii) Observe the output and verify the truth table.

XI-

AND GATE

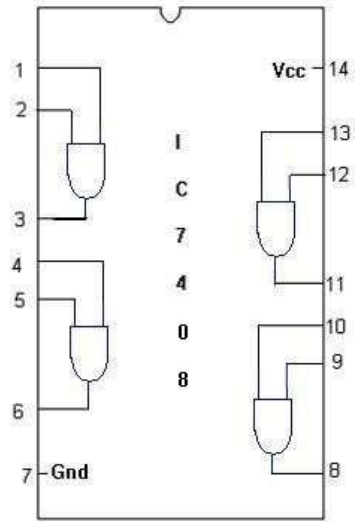
SYMBOL



TRUTH TABLE

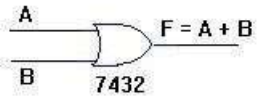
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

PIN DIAGRAM



OR GATE

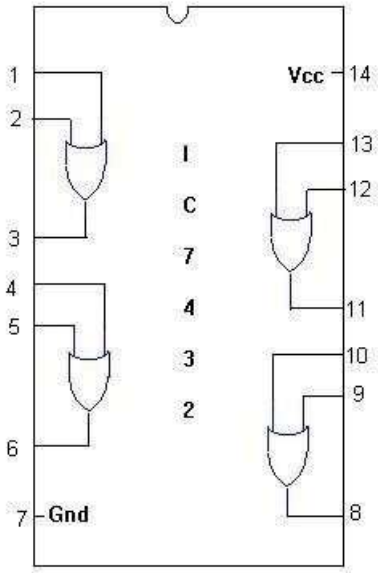
SYMBOL :



TRUTH TABLE

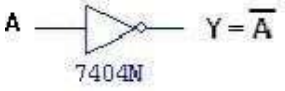
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

PIN DIAGRAM :

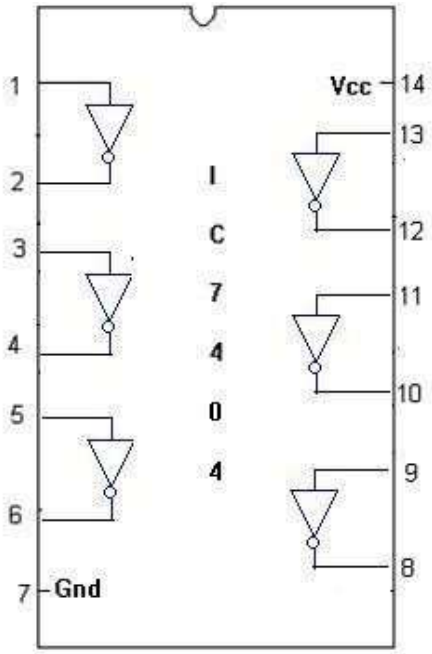


NOT GATE

SYMBOL



PIN DIAGRAM

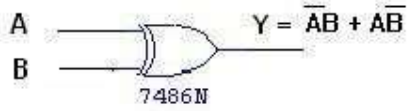


TRUTH TABLE :

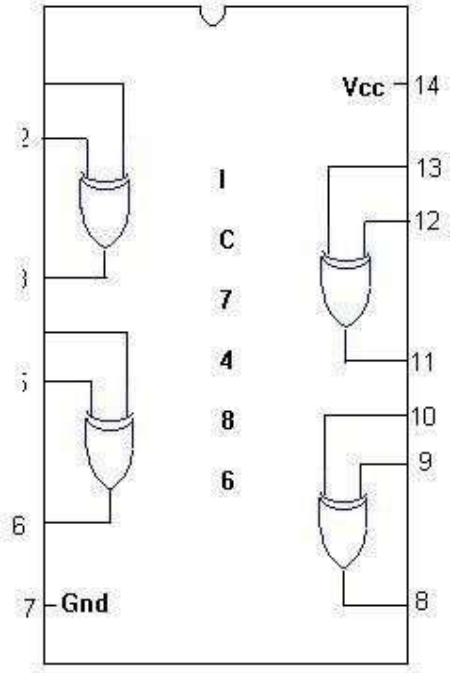
A	$\overline{A}$
0	1
1	0

EX-OR GATE

SYMBOL



PIN DIAGRAM



TRUTH TABLE :

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

EXPERIMENT #1: LOGIC GATES AND BOOLEAN ALGEBRA

OBJECTIVES:

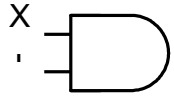
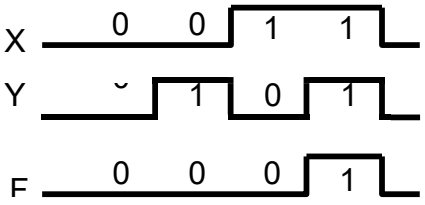
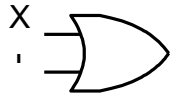
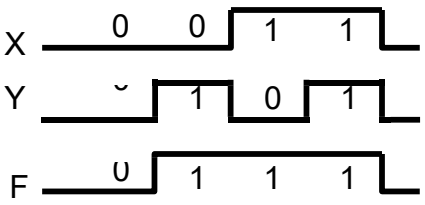
- Study the operation of basic logic gates
- Obtain Boolean expressions from a logic circuit
- Build a logic circuit from Boolean expressions
- Simplify Boolean expressions using Boolean Algebra theorems and postulates
- Obtain truth tables and compute circuit cost for logic circuits

Equipment and ICs:

- Mini-Lab ML-2001 lab station
- 1 IC-7404 (Hex Inverters)
- 1 IC-7408 (Quadruple 2-input AND gates)
- 2 IC-7411 (Triple 3-input AND gates)
- 1 IC-7432 (Quadruple 2-input OR gates)
- 1 IC-7400 (Quadruple 2-input NAND gates)
- 1 IC-7402 (Quadruple 2-input NOR gates)
- 1 IC-7486 (Quadruple 2-input XOR gates)

Introduction:

The three basic logic gates are AND, OR, and NOT. These logic gates are the building blocks of all digital circuits. Other logic gates such as NAND, NOR, XOR, XNOR are derived from the three basic logic gates. The graphic symbol, timing diagrams, and truth table for each logic gate is given below:

Graphic Symbol	Timing diagrams	Truth Table
 F = X.Y **AND**		X Y F 0 0 0 0 1 0 1 0 0 1 1 1
 F = X+Y **OR**		X Y F 0 0 0 0 1 1 1 0 1 1 1 1

LOGIC DIAGRAMS:

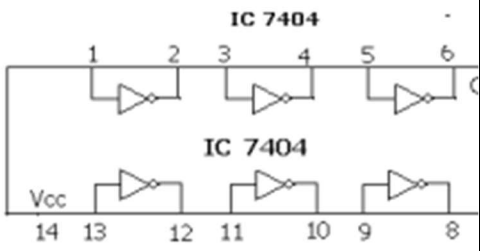
NOT GATE

SYMBOL

TRUTH TABLE

Dec	I/P (A)	O/P $\overline{(A)}$
0	0	1
1	1	0

IC 7404



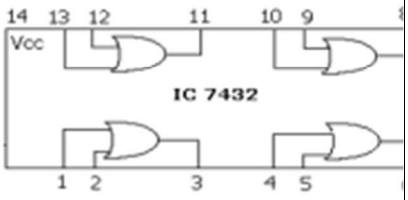
OR GATE

SYMBOL

TRUTH TABLE

Dec Eq	Inputs		Output
	A	B	Y
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

IC 7432



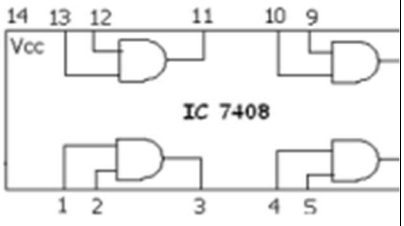
AND GATE

SYMBOL

TRUTH TABLE

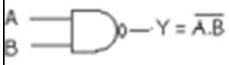
Dec Eq	Inputs		Output
	A	B	Y
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

IC 7408



NAND GATE

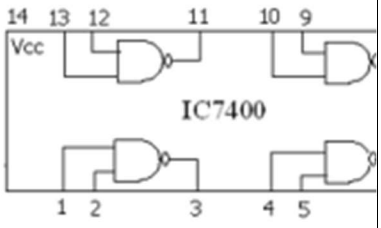
SYMBOL

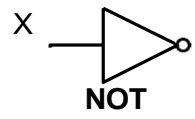
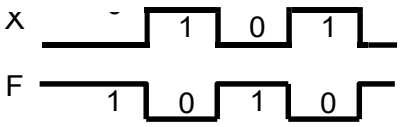
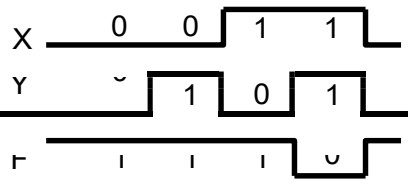
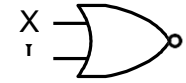
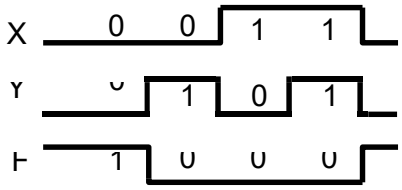
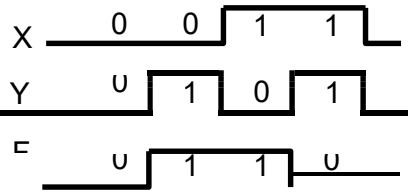
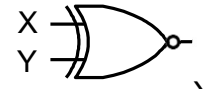
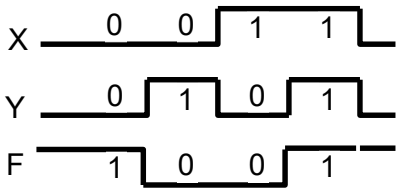


TRUTH TABLE

Dec Eq	Inputs		Output
	A	B	Y
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0

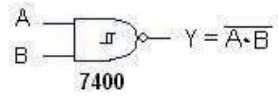
IC 7400



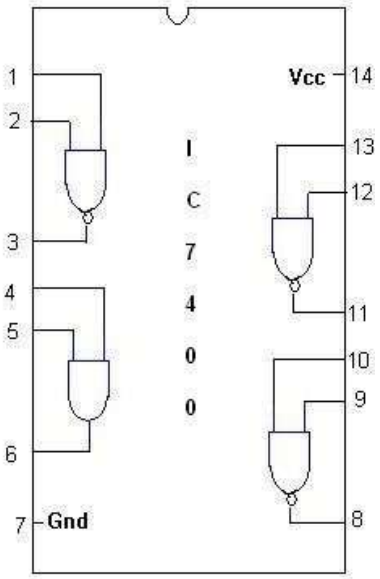
Graphic Symbol	Timing Diagrams	Truth Table
		X F 0 1 1 0
		X Y F 0 0 1 1 0 1 1 1 0
		X Y F 0 0 1 0 1 0 1 0 0 1 1 0
		X Y F 0 0 0 1 0 1 1 1 0
		X Y F 0 0 1 0 1 0 1 0 0 1 1 1

2-INPUT NAND GATE

SYMBOL



PIN DIAGRAM

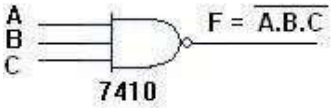


TRUTH TABLE

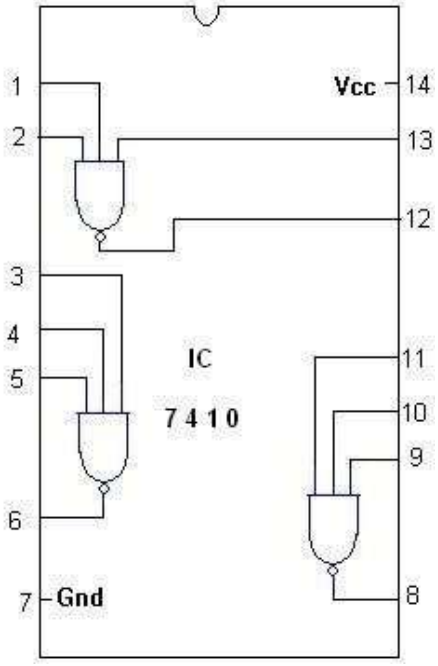
A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

3-INPUT NAND GATE

SYMBOL :



PIN DIAGRAM :

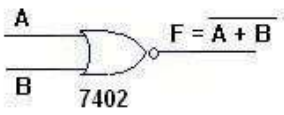


TRUTH TABLE

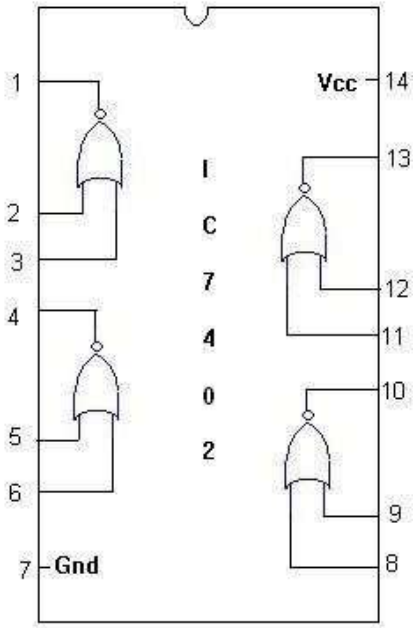
A	B	C	$\overline{A.B.C}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

NOR GATE

SYMBOL :



PIN DIAGRAM :



TRUTH TABLE

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

LOGIC DIAGRAMS:

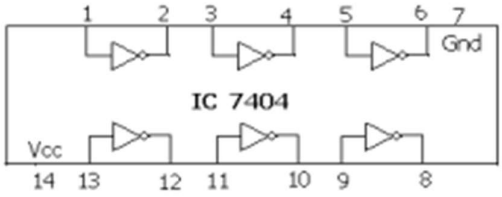
NOT GATE

SYMBOL

TRUTH TABLE

Dec	I/P (A)	O/P $\overline{(A)}$
0	0	1
1	1	0

IC 7404



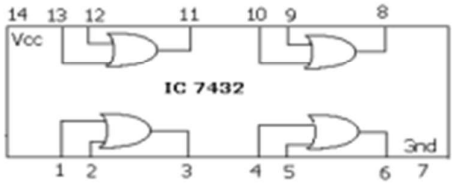
OR GATE

SYMBOL

TRUTH TABLE

Dec Eq	Inputs		Output
	A	B	Y
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

IC 7432



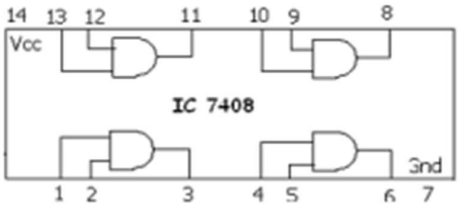
AND GATE

SYMBOL

TRUTH TABLE

Dec Eq	Inputs		Output
	A	B	Y
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

IC 7408



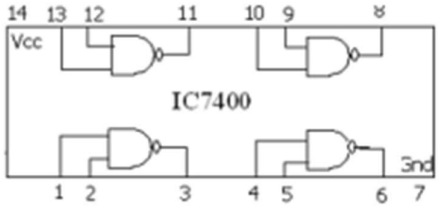
NAND GATE

SYMBOL

TRUTH TABLE

Dec Eq	Inputs		Output
	A	B	Y
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0

IC 7400



Postulates and Theorems of Boolean Algebra

X + 0 = X	X.1 = X	
X + X' = 1	X.X' = 0	
X + X = X	X.X = X	
X + 1 = 1	X.0 = 0	
(X')' = X		Involution
X + Y = Y + X	XY = YX	Commutative
X + (Y + Z) = (X + Y) + Z	X(YZ) = (XY)Z	Associative
X(Y + Z) = XY + XZ	X + YZ = (X + Y)(X + Z)	Distributive
(X + Y)' = X'Y'	(XY)' = X' + Y'	DeMorgan
X + XY = X	X(X + Y) = X	Absorption

Questions

1. List out the basic gate.
2. Mention the universal gate.
3. What are the applications of gates?
4. Write the truth table of AND gate.
5. Write the truth table of OR gate.
6. Write the truth table of NOT gate.
7. Write the truth table of NAND gate.
8. Write the truth table of NOR gate.
9. Write the truth table of EX- OR gate.

Ex.No.-1b VERIFICATION OF BOOLEAN THEOREMS USING DIGITAL LOGIC GATES

AIM:

To verify the Boolean Theorems using logic gates.

APPARATUS REQUIRED:

SL. NO.	COMPONENT	SPECIFICATION	QTY.
1.	AND GATE	IC 7408	1
2.	OR GATE	IC 7432	1
3.	NOT GATE	IC 7404	1
4.	IC TRAINER KIT	-	1
5.	CONNECTING WIRES	-	As per required

THEORY:

BASIC BOOLEAN LAWS

1. Commutative Law

The binary operator OR, AND is said to be commutative if,

1. A+B = B+A
2. A.B=B.A

2. Associative Law

The binary operator OR, AND is said to be associative if,

1. A+(B+C) = (A+B)+C
2. A.(B.C) = (A.B).C

3. Distributive Law

The binary operator OR, AND is said to be distributive if,

1. A+(B.C) = (A+B).(A+C)
2. A.(B+C) = (A.B)+(A.C)

4. Absorption Law

1. A+AB = A
2. A+AB = A+B

5. Involution (or) Double complement Law

1. A = A

6. Idempotent Law

1. A+A = A
2. A.A = A

7. Complementary Law

1. A+A' = 1
2. A.A' = 0

8. De Morgan's Theorem

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

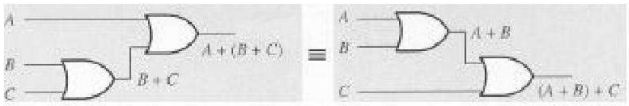
(A+B)' = (A)'. (B)'

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

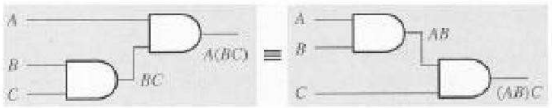
(A.B)' = (A)'+ (B)'

Associative Laws of Boolean Algebra

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



A • (B • C) = (A • B) • C



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

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

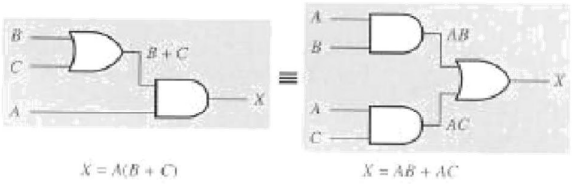
Proof of the Associative Property for the AND operation: (A·B)·C = A·(B·C)

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

Distributive Laws of Boolean Algebra

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

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



Proof of Distributive Rule

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

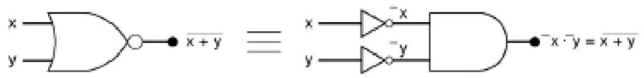
Proof of Distributive Rule

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

Demorgan's Theorem

a) Proof of equation (1):

Construct the two circuits corresponding to the functions A'. B' and (A+B)' respectively. Show that for all combinations of A and B, the two circuits give identical results. Connect these circuits and verify their operations.



(a)

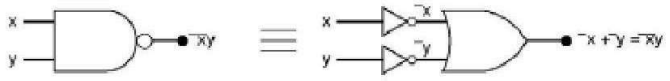
(b)

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

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

b) Proof of equation (2)

Construct two circuits corresponding to the functions A'+B' and (A.B)' A.B, respectively. Show that, for all combinations of A and B, the two circuits give identical results. Connect these circuits and verify their operations.



(a)

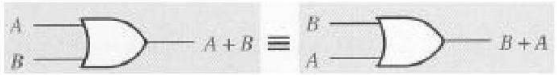
(b)

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

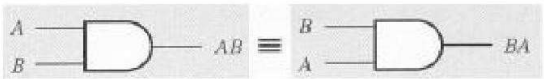
A	B	A+B	$\overline{A + B}$	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Commutative Laws of Boolean Algebra

A + B = B + A



A • B = B • A



We will also use the following set of postulates:

P1: Boolean algebra is closed under the AND, OR, and NOT operations.

P2: The identity element with respect to • is one and + is zero. There is no identity element with respect to logical NOT.

P3: The • and + operators are commutative.

P4: • and + are distributive with respect to one another. That is,

A • (B + C) = (A • B) + (A • C) and A + (B • C) = (A + B) • (A + C).

P5: For every value A there exists a value A' such that A•A' = 0 and A+A' = 1. This value is the logical complement (or NOT) of A.

P6: • and + are both associative. That is, (A•B)•C = A•(B•C) and (A+B)+C = A+(B+C).

You can prove all other theorems in boolean algebra using these postulates.

PROCEDURE:

1. Obtain the required IC along with the Digital trainer kit.
2. Connect zero volts to GND pin and +5 volts to Vcc.
3. Apply the inputs to the respective input pins.
4. Verify the output with the truth table.

QUESTIONS

1. What is Demorgan's law?
2. What is associative law?
3. What is mean by compliment gate?
4. Explain the basic laws in digital electronics
5. What is double complement?
6. What is absorption law?