Logic Gate of Boolean Algebra with logics

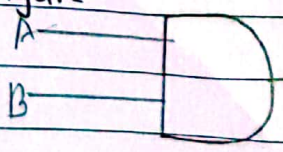
A logic gate is a basic building block of a digital circuit that has two inputs and one output. The relationship between the input and the output is based on a certain logic. These gates are implemented using electronic switches like transistors, diodes.

The basic logic gate are categorized into seven: AND, OR, XOR, NAND, NOR, XNOR and NOT. These logic gates with their logic gate symbol and truth table are explained below:

Basic logical Gate:

- AND Gate: The AND gate is a digital gate with inputs and output, which perform logical conjunction based on the combination of its input. The output of this gate is high only when all the inputs are high otherwise output will be low. The symbol and truth table of an AND gate with two input is shown:

Figure

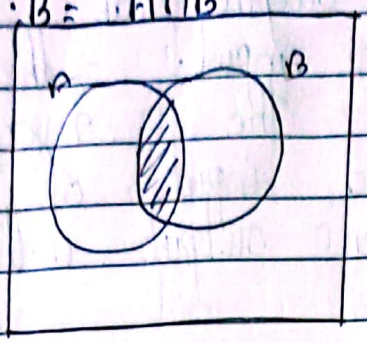


Truth table

Input		output
A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

DATE _____ PAGE _____

Set diagram:
$A \cdot B = A \cap B$



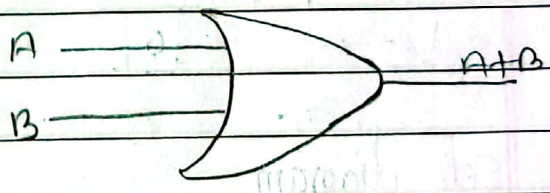
OR Gate: The OR gate is a digital logic gate with input and one outputs that performs a logical conjunction based on the combination of it input. The output of the OR gate is high only when one or more inputs are ~~true~~high. ~~When~~ Otherwise output will be low.

Algebraic expression: $f = A+B$

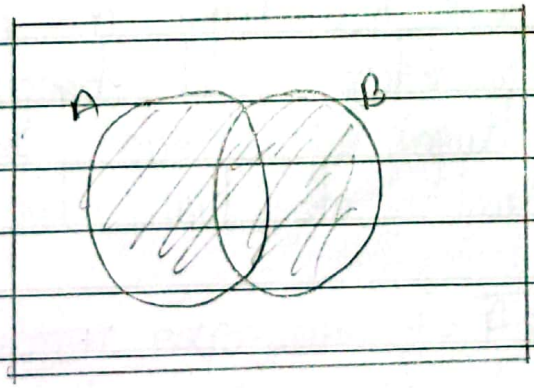
Truth table

Input		output
A	B	f
0	0	0
0	1	1
1	0	1
1	1	1

figure:



Set: $A \cup B = A+B$

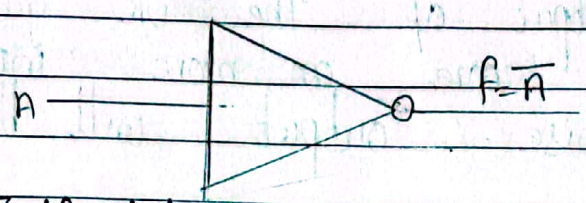


DATE _____ PAGE _____

Not Gate: The Not gate is a digital logic gate with one input and one output that an inverter operation of the input. The output of NOT gate is the reverse of the input. When the input of the Not gate is high then output will be low and viceversa.

Algebraic expression: $f = A'$

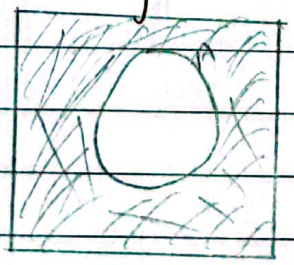
Symbol



Truth table

Input	output
A	f=A'
0	1
1	0

Set diagram



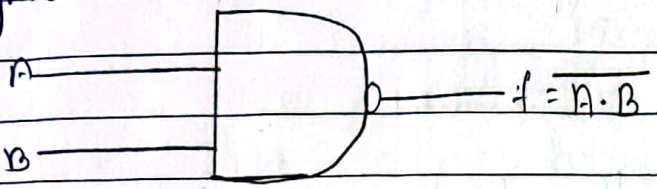
Universal Gate:

DATE _____ PAGE _____

- NAND Gate: The NAND Gate is a digital logical gate with 'n' input and one output. that performs the operation of the AND gate followed by the operation of NOT gate. NAND gate is designed by combining AND and NOT gates. If the input of NAND gate high, then the output of gate will be low

Algebraic expression: $f = \overline{A \cdot B}$

Symbol:



Truth table

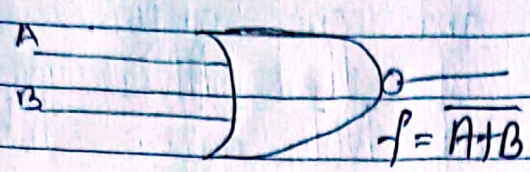
Input			output	
A	B	A·B	f	$\overline{A \cdot B}$
0	0	0		1
0	1	0		1
1	0	0		1
1	1	1		0

- NOR Gate: The NOR gate is a digital logic gate with n input and one output. NOR gate is designed by combining the OR and NOT gate. When one of input of NOR gate is high the output will be low.

Algebraic expression $f = \overline{A+B}$

DATE _____ PAGE _____

Symbol



Truth table

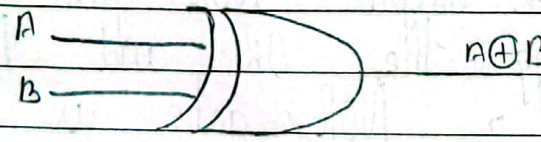
Input		Output	Output
A	B	A+B	$f = \overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Other logic Gates:

- Exclusive - OR Gate: The Exclusive - OR Gate is digital logic gate with two inputs and one output. The short form of this gate is Ex-OR. It performs based on the operation of OR-gate. If any one of the inputs of the gate is high, then the output of Ex-OR gate will be high.

Algebraic expression: $A'B + AB'$ $(A \oplus B)$

Symbol



DATE _____ PAGE _____

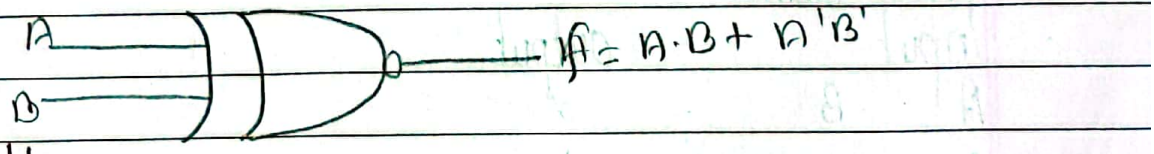
Truth table:

Input						output
A	B	A'	B'	A'B	AB'	A'B+AB'
0	0	1	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	1	0	0	0	0	0

- Exclusive-NOR Gates: The Exclusive-NOR gate is a digital logic gate with two input and one output. The short form of the gate is Ex-NOR. When both the inputs of this gate are high, then the output will be high But if any one of input is high the output will be low.

Algebraic expression: $F = A \cdot B + A'B' / A \odot B$

Symbol



Truth table

Input						
A	B	A'	B'	A'B'	AB	f = A·B + A'B'
0	0	1	1	1	0	1
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	1	0	0	0	1	1

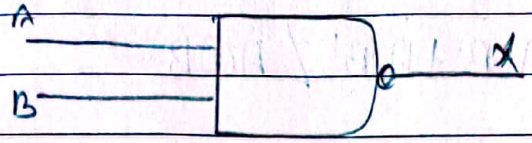
DATE _____ PAGE _____

Universal Gate : Universal gate are the logic gate which are capable of implementing any Boolean function without requiring any other type of gate. Universal gate can realize all the binary operation and all the basic logic gate can be derived from them, that is why they are called Universal gate.

NAND Gate is a universal Gate.

NAND Gate can be connected to form any other logic gate. NAND Gate can be connected to form Inverter, And and OR gates.

logic symbol



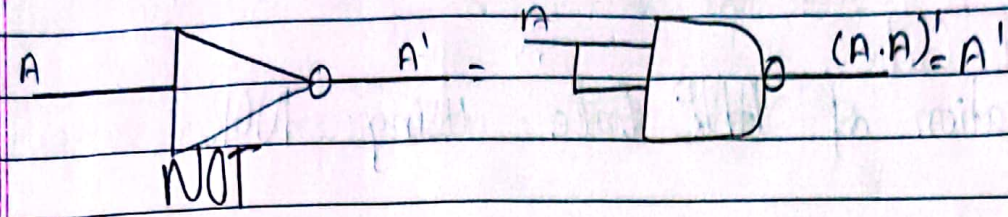
Truth table

Input		output
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

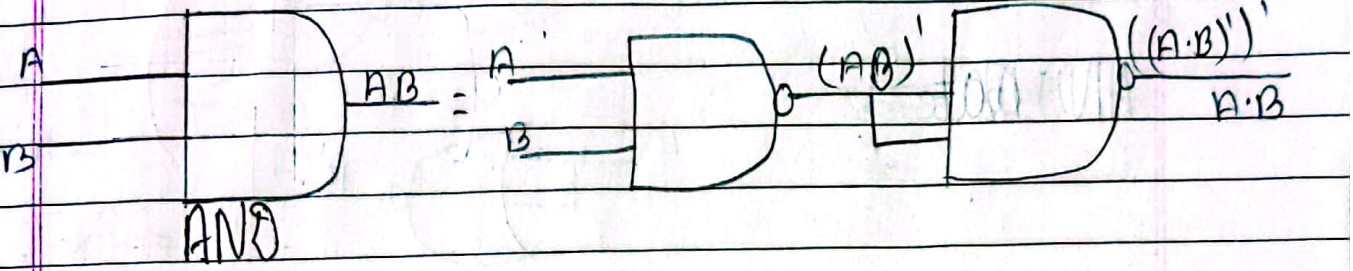
DATE _____ PAGE _____

Implementing

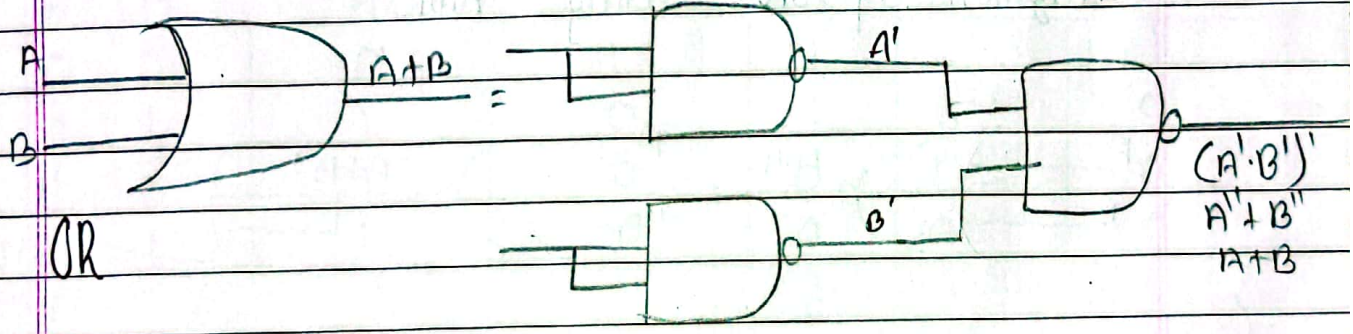
- ~~Relea~~ Realization of Not gate from NAND



- Realization of AND gate from NAND



Realization of OR gate using NAND



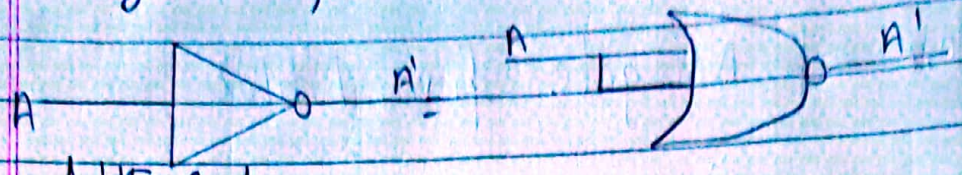
NOR Gate:

NOR Gate is constructed by connecting a NOT Gate at the output teemnal of the OR Gate.

NOR Gate ~~is~~ OR Gate + NOT Gate

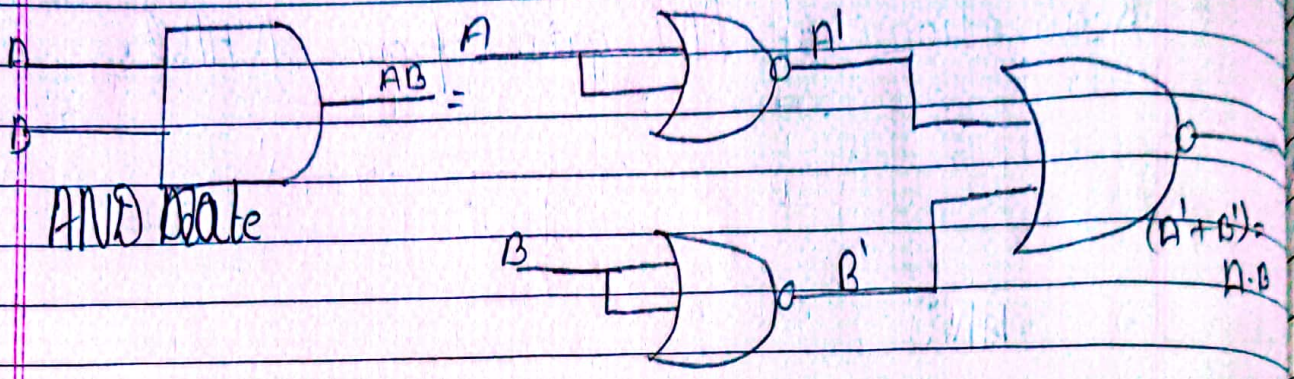
DATE _____ PAGE _____

- Realization of NOT Using NOR



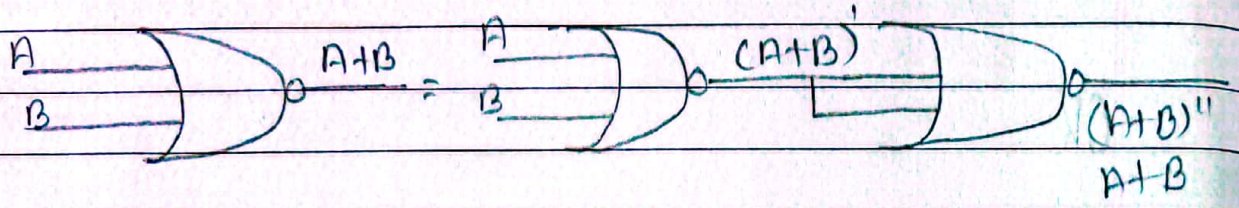
NOT Gate

- Realization of AND Gate using NOR



AND Gate

Realization of OR using NOR



From above analysis all three basic gate can be independently prepared by NAND and NOR gate. So they any gate can be derived from NAND or NOR gate.

DATE _____ PAGE _____

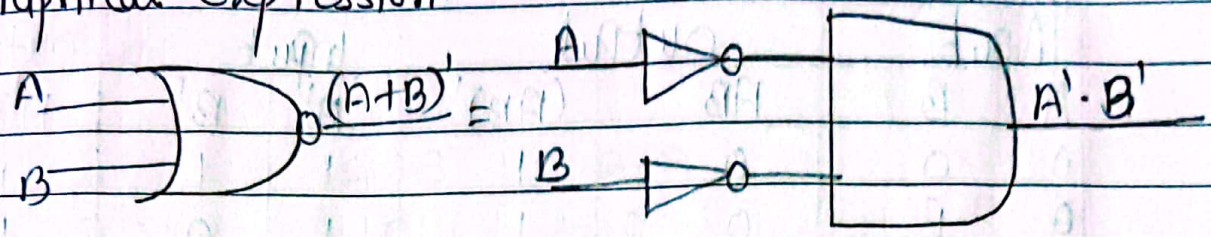
De-Morgan's law

- first law:

"The complement of a sum is equal to the product of the component."

Algebraic expression : $(A+B)' = A' \cdot B'$

Graphical expression:



Truth table:

INput		Output		INput		Output
A	B	A+B	(A+B)'	A'	B'	A'·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

Conclusion:

Comparing the value of $(A+B)'$ and $A' \cdot B'$ from truth table, both are equal, hence proved.

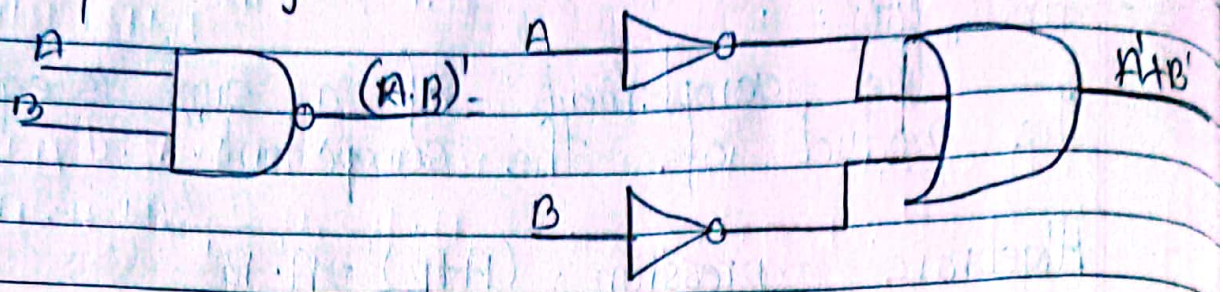
+ Second law

Statement: "The component of product is equal to the sum of complement

Algebraic expression = $(A \cdot B)' = A' + B'$

DATE _____ PAGE _____

Graphical symbol:



Truth Table

Input		output		Input		output
A	B	AB	(AB)'	A'	B'	A'+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

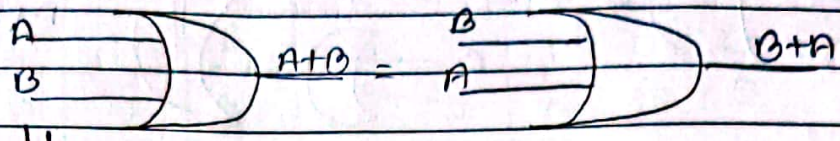
Conclusion: Comparing the value of (AB)' and A'+B' from truth table both are equal, hence proved.

Laws of Booleans Algebra

1. Commutative law:

a. $(A+B) = (B+A)$

Graphical symbol

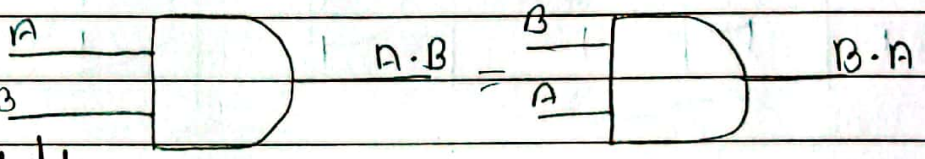


Truth table

Input		output	
A	B	A+B	B+A
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

b. $(A \cdot B) = (B \cdot A)$

Graphical symbol:



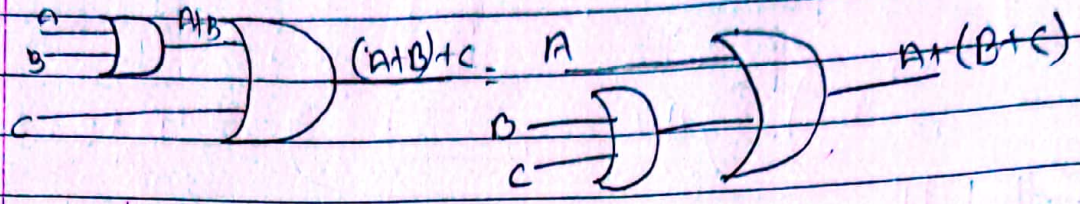
Truth table:

Input		Output	
A	B	A·B	B·A
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

DATE _____ PAGE _____

2. Associative law
Algebraic expression: $(A+B)+C = A+(B+C)$

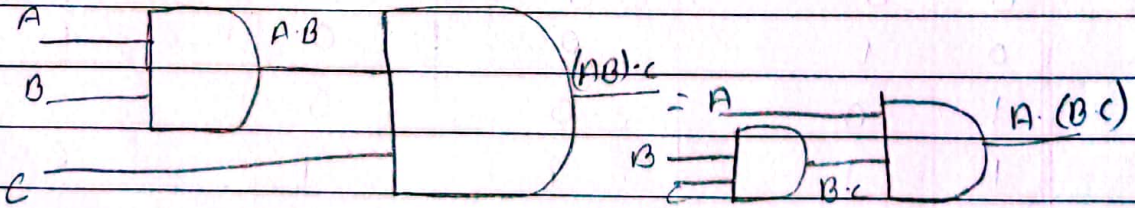
Graphical symbol:



Truth table
Input

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

$(A.B).C = A.(B.C)$
Graphical symbol



DATE _____ PAGE _____

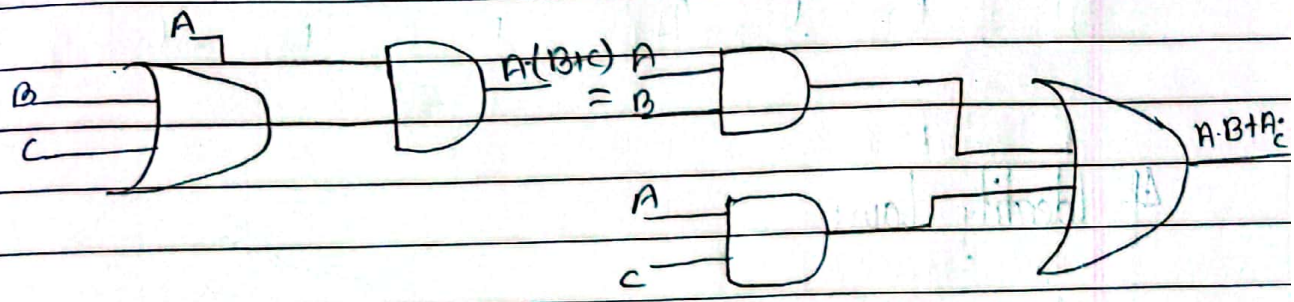
Truth table

A	B	C	A·B	(A·B)·C	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	0	1	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

3. Distributive law

Graphical Symbol:

$A.(B+C) = A.B + A.C$



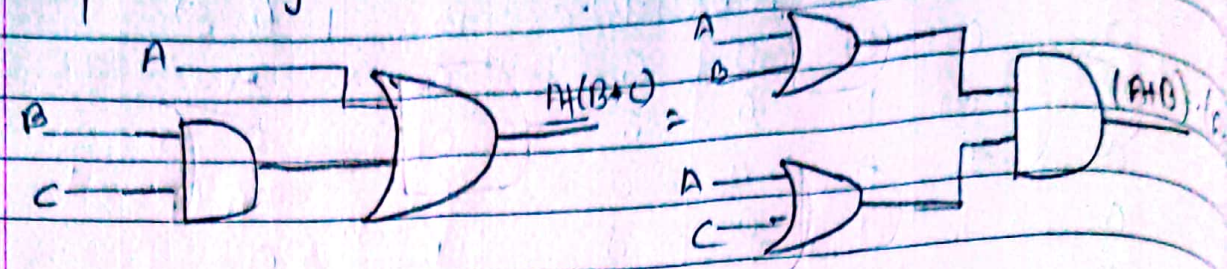
Truth table

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

DATE _____ PAGE _____

H $A+(B.C) = (A+B).(A+C)$

Graphical symbol



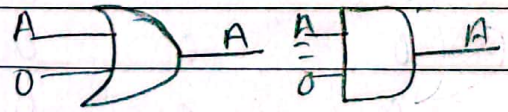
Truth table

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

4. Identity law:

H $A+0 = A$ $[\because A = 1A]$

Graphical symbol



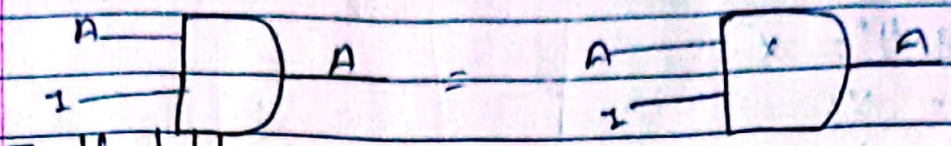
Truth table

A	B	A+0
0	0	0
1	0	1

DATE _____ PAGE _____

H $A \cdot 1 = A$

Graphical symbol:



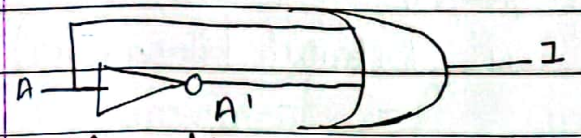
Truth table

A	1	A·1
0	1	0
1	1	1

5. Complement law

H $A + A' = 1$

Graphical symbol

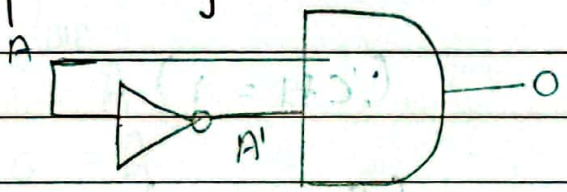


Truth table

A	A'	A+A'
0	1	1
1	0	1

+ $A \cdot A' = 0$

Graphical symbol:



A	A'	A·A'
0	1	0
1	0	0

DATE _____ PAGE _____

Simplification Theorem

* $XY + X \cdot Y' = X$
* $(X+Y) \cdot (X+Y)' = X$
* $X + XY = X$
* $X(X+Y) = X$
* $(X+Y')Y = XY$
* $XY' + Y = X + Y$

Consensus Theorem

$XY + X'Z' + YZ = XY + X'Z$

$(X+Y)(X'+Z)(Y+Z) = (X+Y)(X'+Z)$

Duality

$(X+Y+Z) = XYZ$

$(A+B)(A+C) = A + BC$

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

$= AA + AC + AB + BC$ (Distributive law)

$= A + AC + AB + BC$ ($\because AA = A$)

$A(1+C) + B(A+C)$ ($\because C+1 = 1$)

$A + AB + BC$ (factoring)

$A(1+B) + BC$ ($\because 1+B = 1$)

$A + BC$ ($A \cdot 1 = A$)

DATE _____ PAGE _____

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

$= (A + AB) + \bar{A}B \quad (\because A = A + AB)$

$= (A \cdot A + AB) + \bar{A}B \quad (\because A = A \cdot A)$

$= AA + AB + \bar{A}A + \bar{A}B \quad (\because \bar{A}A = 0)$

$= A(A + B) + \bar{A}(A + B) \quad (\because A(A+B) = AA + AB)$

$(A + A')(A + B)$

$1 \cdot (A + B)$

Other law of boolen algebra

1. Complement law:
A 'complement' $= A'$ i.e input 1 then output 0.

2. AND law
- $A \cdot A = A$
- $A \cdot 0 = 0$
- $A \cdot 1 = A$
- $A \cdot A' = 0$

3. OR law
i. $A + A = A$
ii. $A + 0 = A$
iii. $A + 1 = 1$
$A + A' = 1$

DATE _____ PAGE _____

4. Distributive law

$A.(B+C) = AB + AC$

$A+(B.C) = (A+B)(A+C)$

Example

$A+A'B$

$(A+A').(A+B)$

$1 = (A+B)$ let $(A+B) = x$

$1 \cdot x$

x

$A+B$

$BAC' + B'AC' + BC'$

$= AC'(B+B') + BC'$ $(B+B'=1)$

$= AC' + BC'$

$= C'(A+B)$

$AB + AB'C + AB'C'$

$= AB + AB'(C+C')$

$= A(B+B')$

A

DATE _____ PAGE _____

3 $(A+B+C).(A+B'+C)(A+B+C')$

let $A+B=x$

$= (x+C)(A+B'+C)(x+C')$

$= (x+C)(x+C')(A+B'+C)$

$= (x+C.C')(A+B'+C)$

$= x(A+B'+C)$

$(A+B).(A+B'+C)$

$= AA+AB'+AC+A+B(B'+C)$

$= A+B.B'+BC$

$= A+BC$

4. $(A+B)(A+B')(A'+B)(A'+B')$

Sol^n

$= (A+BB')(A'+BB')$

$= (A+0).(A'+0)$

$= A.A'$

$= 0$

DATE _____ PAGE _____

Redundancy Theorem:
Condition
i Three variable
ii Each variable is repeated twice
iii One variable is complemented
Solution
iv Take complimented variable

5. $AB + A'C + BC$
$= AB + A'C$

6. $XY + XZ + XZ'$
$= XZ + XZ'$

5. Commutative law
$A+B = B+A$
$A \cdot B = B \cdot A$

6. Associative law
$(A \cdot B) \cdot C = A \cdot (B \cdot C)$
$(A+B)+C = A+(B+C)$

7. Demorgan's Theorem
$(A+B)' = A'B'$
$(A \cdot B)' = A'+B'$

8. Priority:
i) Braces ii) NOT iii) AND iv) OR

DATE _____ PAGE _____

A	B	C	B+C	A.(B+C)	AB	AC	AB+AC
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Boolean algebra it representation
i SOP
ii POS

- SOP (Sum of product):
If only consider the miniterms which have high value. SOP is symbolized by Σ and miniterms is represented by small m.

Example
Truth table

Miniterms	Digit	A	B	C	f
A'B'C	0	0	0	0	0
A'B'C	1	0	0	1	0
A'BC'	2 (m_2)	0	1	0	1
A'BC	3	0	1	1	0
AB'C'	4 (m_4)	1	0	0	1
AB'C	5 (m_5)	1	0	1	1
ABC'	6 (m_6)	1	1	0	1
ABC	7 (m_7)	1	1	1	1

$F = m_2 + m_4 + m_5 + m_6 + m_7$

DATE _____ PAGE _____

Standard formal contains all the variable in it minterms.

$A'BC' + AB'C' + AB'C + ABC' + ABC$

$= A'BC' + AB'(C'+C) + AB(C'+C)$

$= A'BC' + AB' + AB$

$= A'BC' + A(B'+B)$

$= A'BC' + A$

let $BC' = X$ $= A + A'B = A + B$

$A + A'BC'$

$A + A'X$

$A + X$

$A + BC'$ (Minimal sop formal doesnot contain all the variable in its minterms)

* $F(A,B,C) = m_2 + m_4 + m_5 + m_6 + m_7$

Example

$f(A,B) = \Sigma m(0,2,3)$

Digital	A	B	f
0	0	0	1
1	0	1	0
2	1	0	1
3	1	1	1

$f = A'B' + AB' + AB$

$B'(A'+A) + AB$

$B' + AB$ $(\because A + A'B = A + B)$

DATE _____ PAGE _____

B'+A

SOP form : having all the variable

1st type $F = \Sigma m(0,1,3)$

2nd type $f = A'B' + AB + A'B'$

0 1	1 1	0 0
1	1	0
m_1	m_3	m_0

3rd type

Minterm	A	B	f
m_0	0	0	1
m_1	0	1	1
m_2	1	0	0
m_3	1	1	1

$B(A'+A) \cdot + A'B'$

$B \cdot 1 + A'B'$

$B + A'B'$ ($A + A'B = A + B$

$B + B'A'$

$B + A'$

$F(A,B) \Sigma m(0,2,3)$

$m_0 + m_2 + m_3$

$A'B' + AB' + AB$

Rough

$0 = 00$ $A'B'$

$2 = 10$ AB'

$3 = 11 = AB$

DATE _____ PAGE _____

Example $\Sigma m(1,3)$ Convert the above expression into minimal SOP form

$F = m_1 + m_3$
$= A'B + AB$
$= B$

01	11
$A'B$	AB

Example $f(A,B,C) = \Sigma m(0,1,4,5,7)$
$m_0 + m_1 + m_4 + m_5 + m_7$

$A'B'C' + A'B'C + AB'C' + AB'C + ABC$
$= A'B' + AB' + ABC$
$= B'(A+A') + ABC$
$= B' + ABC$
$= B' + B(AC)$
$= B' + AC$

0 = 000 $A'B'C'$
1 = 001 $A'B'C$
4 = 100 = $AB'C'$
5 = 101 = $AB'C$
7 = 111 = ABC

POS (Product of sum):
It only consider the maxterms which have low value
POS is symbolized by Π and maxterm is represented by Capital M

Truth table

Maxterm	Digit	A	B	C	f
$A+B+C$	0	0	0	0	0
$A+B+C'$	1	0	0	1	0
$A+B'+C$	2	0	1	0	1
$A+B'+C'$	3	0	1	1	0
$A'+B+C$	4	1	0	0	1
$A'+B+C'$	5	1	0	1	1
$A'+B'+C$	6	1	1	0	1
$A'+B'+C'$	7	1	1	1	1

DATE _____ PAGE _____

$F(A,B,C) = M_0 \cdot M_1 \cdot M_3$

$f(A,B,C) = \Pi M(0,1,3)$

$= (A+B+C)(A+B+C')(A+B'+C')$

$=$ let $A+B = X$

$= (X+C)(X+C')(A+B'+C')$

$= (X+C\cdot C')(A+B'+C')$

let $B'+C' = Y$

$(A+B)(A+Y)$

$A+BY$

$A+B(B'+C')$

$A+BC'$

for Pos

For $(ABC) = \Sigma m(2,4,5,6,7)$

010 = $A'B'C'$

100 = $AB'C'$

101 = $AB'C$

110 = ABC'

111 = ABC

$F(ABC) = A'B'C' + AB'C + AB'C + ABC$

$f(AB) = \Sigma m(0,2,3)$

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

$= B'(A+A') + AB$

$= B' + BA \quad (\because A + AB = A+B)$

$= B' + A$

DATE PAGE

Q1. Simplify the following in Boolean algebra.

a. $AC + BC' + A'C' + B'C'$

Soln

$C'(A + B + A' + B')$

$C'(A + A' + B + B')$

C'

b. $C(B + C)(A + B + C)$

$C(B + C + 0)(A + B + C)$

let $B + C = X$

$= C(X + A \cdot 0)(A + B + C)$

$= C(X + A \cdot 0)$

$= C(B + C)$

$= BC + C \cdot C$

$= BC + C$

$= C(B + 1)$

C

c. $AB'C + A'B'C + AB'C' + A'B'C'$

Soln

DATE _____ PAGE _____

$AB'C + A'B'C + AB'C' + A'B'C'$

$= AB'(C+C') + A'B'(C+C')$

$= AB' + A'B'$

$= B'(A+A')$

$= B'$

d. $A+B(A+B) + A(A'+B)$

$= A + AB + BB + A \cdot A' + AB$

$= A + AB + B + 0$

$= A + B$

e. $(AB'C' + AB'C + ABC + ABC')(A+B)$

$[AB'(C'+C) + AB(C+C')](A+B)$

$[AB' + AB](A+B)$

$[A(B'+B)](A+B)$

$= A + (A+B)$

$= A$

DATE _____ PAGE _____

* $F = (A \cdot B \cdot C)' + A'BC + (AC)'$

$= A' + B' + C' + A'BC + A' + C'$

$A' \cdot 1 + B' + B' + A'BC$

$A'(1 + BC) + B' + C'$

$A' \cdot 1 + B' + C'$

$(ABC)'$

* Comparing SoP & Pos

Digit	A	B	C	f
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

DATE _____ PAGE _____

SOP

$Y(A,B,C) = \Sigma m(0,2,3,6,7)$

$m_0 + m_2 + m_3 + m_6 + m_7$

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

$= A'B'C' + A'B(C'+C) + AB(C'+C)$

$= A'B'C' + A'B + AB$

$= A'B'C' + B(A'+A)$

$= A'B'C' + B$

let $A'C' = x$

$= xB' + B \quad (A + A'B = A + B)$

$= B + A'C'$

POS

$Y = (A,B,C) = \pi M(1,4,5)$

M_1, M_4, M_5

$(A+B+C')(A'+B+C)(A'+B+C')$

$(A+B+C') \cdot (A'+B+C \cdot \bar{C})$

$(A+B+C')(A'+B)$

$[B + (A+\bar{C}) \cdot \bar{A}]$

$B + (A \cdot \bar{A} + \bar{A} \cdot \bar{C})$

$B + \bar{A}\bar{C}$

$(\bar{A}+B)(B+\bar{C})$

DATE _____ PAGE _____

K-MAP (Karnaugh map)

mapping structure to arrange boolean algebra most easiest form to express boolean algebra into minimal form.

groups are formed and while forming groups

i) try to make large group as possible
ii Size of group is give 2 to the power n

SOP form in K-map

$f(A,B) = (m_0, m_2, m_3)$

A \ B	B'	B
A'	00 / A'B'	01 A'B
A	10 AB'	11 AB

$= A'B' + AB + AB$
$= B'(A'+A) + AB$
$= B' + AB$
$= B' + A$

DATE _____ PAGE _____

Using k-map

a. $f = (W,X,Y,Z) = \Sigma(7, 13, 14, 15)$

WX \ YZ	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

1st part = WXZ

2nd part = XYZ

3rd part = WXY

f = WXZ + XYZ + WXY

b. $f(W,X,Y,Z) = \Sigma(1, 3, 4, 6, 9, 11, 14, 15)$

WX \ YZ	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

1st part = W'X'Z

2nd part = W'XZ'

3 part = WXY

4 part = WYZ

f = W'X'Z + W'XZ' + WXY + WYZ

DATE _____ PAGE _____

d. $f(W,X,Y,Z) = \Sigma(1,3,4,5,7,8,9,13,15)$

Sol:

WX/YZ	00	01	11	10
00	0	1 1	2	1 3
01	1 4	1 5	1 7	6
11	12	1 13	1 15	14
10	1 8	1 9	11	10

1st part = XZ
2nd part = Y'Z
3rd part = W'XY'
4th part = WX'Y'

$f = XZ + Y'Z + W'XY' + WX'Y'$

*

AB\CD	00	01	11	10
00	1 0	1 1	1 3	1 2
01	4	5	7	6
11	12	1 13	1 15	1 14
10	1 8	9	11	1 10

1st group A'B'
2nd group B'D'
3rd group ABCD
4th group ACD'

$F = A'B' + B'D' + ABD + ACD'$

DATE _____ PAGE _____

*

AB\CD	00		01		11		10	
00	1	0	1	1	1	3		2
01	1	4	1	5		7		6
11	1	12	1	13		15		14
10		8	1	9	1	11		10

$f \Sigma m (0, 1, 3, 5, 13, 9, 11, 14, 12)$

1st group : BC'
2nd group = $B'D$
3rd group = $A'C'$

$F = BC' + B'D + A'C'$

* $f = \Sigma m(0, 1, 2, 4, 5, 6)$

A\BC	00		01		11		10	
0	1	0	1	1		3	1	2
1	1	4	1	5		7	1	6

1st part = B'
2nd part = C'
$f = B' + C'$

DATE _____ PAGE _____

* $f(w,x,y,z) = \pi(1,4,5,6,11,12,13,14,15)$

wx \ yz	0+0	0+1	1+1	1+0
0+0	0	0 1	3	2
0+1	0 4	0 5	7	0 6
1+1	0 12	0 13	0 15	0 14
1+0	8	9	0 11	10

1st part $x'+y$
2nd part = ~~$w+y+z'$~~
3rd = $w'+y'+z'$
4th = $x'+y+z'$

final part $(x'+y)(w+y+z')(w'+y'+z')(x'+y+z')$

* $f(w,x,y,z) = \pi(0,4,5,7,8,9,13,15)$

w+x \ y+z	0+0	0+1	1+1	1+0
0+0	0 0	1	3	2
0+1	0 4	0 5	0 7	6
1+1	12	0 13	0 15	14
1+0	0 8	0 9	11	10

1st part = $x'+z'$
2nd part = $w+y+z$
3rd part = $w+x+y$

$f = (x'+z')(w+y+z)(w+x+y)$

DATE _____ PAGE _____

11. Simplify the following Boolean expression

a. $w'z + xz + x'y + wx'z$

w'x \ yz	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

DATE _____ PAGE _____

Imp Short Answer

1. Why alphabet are used for representing number 9 in hexadecimal number system?

→ Alphabets are used for representing number above 9 in hexadecimal number system to make the problem easier to slove and reduce the complexity of digit.

2. Why NAND gate is said to be universal gate?

→ NAND gate are called universal gate because any logic function can be done using them and thus can produce a inverter.

3. What are the three different method that are used for representing the negative number in decimal form into binary?

→ The three different method that are used for representing the negative number in decimal form into binary are:
- Sign magitude representation
- One's complement form
- Two's complement form

4. What is the use of error detection code?

→ The use of error detection code is to transmit binary information from one place to another by electric source.

DATE _____ PAGE _____

5. What is Boolean algebra?

→ Boolean algebra of logic which deals with study of binary variable and logic operation

7. What is the advantage of representing a data in hexa-decimal form?

→ The advantage of representing a data in hexadecimal form is that it is very compact and by using a base 16 means that the number is usually less than in binary or decimal.

8. Differentiate between Pos and Sop expression.

- Sop: It is used more number of AND gate. It takes OR function of product term.

Pos: It is used more number of OR gate. It take AND function of sum term.

9. Why do we need don't care in k-map?

→ Don't care conditions allow us to replace the empty cells of a k-map of form a larger group of the variables. While forming a group of cells we can simply ignore the cell.

DATE _____ PAGE _____

10. Convert a binary number 1101 into equivalent gray c

1101
↓
= 1011

15. What is the importance of k-map?

→ k-map are used to simplify real-world logic requirements so that they can be implemente using a minimum number of physical logic gates. It is also used for

12. Why gray code are used in k-map instead of binary codes?

- Gray code actually follow Adjacency property i.e between two successive code there is only one bit change and in the binary code, there will be multiple value change. Because of Adjacency property when one plots o k-map, literals can be grouped keeping other literal constant.

13. Mention advantage of parity method of error det detection.

- Error on a noisy line can be caught quickly.
- Only the errant word has to be re transmitte

DATE _____ PAGE _____

14. Is gray code weight or un-weighted code?

→ It is the non weighted code and it is not ~~arith~~ arithmetic codes. That means there are no specific weights assigned to the bit position. It has a very special feature that, only one bit will change each time the decimal number is incremented.

15. Decimal number are weighted number justify.

- Yes because each digit position is being assigned with specific weight.

16. What is advantage of BCD code.

- It is fast and efficient system to convert the decimal number into binary number as compare to pure binary system.

17. What are error detection code? Give example?

→ Error detection is the process of detecting the error that are present in the data transmitted form transmitter to receiver, in a communication system. We use some redundancy code to detect these errors, by adding to the data while it is transmitted from source. These codes are called error detecting code.

DATE _____ PAGE _____

18. Why NAND and NOR gate are called universal gate?

- The NAND and NOR gate are known as universal gate, since logic function can be implemented by NAND or NOR gates.

Circuits using NAND and NOR are popular as they are easier to design and therefore cheap function of other gate can easily be implemented using NAND and NOR gate; for this reason they are called universal gate.

19. State De-morgan's theorem mathematically