BOOLEAN ALGEBRA

- used to analyze and simplify the digital (logic) circuits.
- Uses only 0 and 1 (binary numbers)
- Also called Binary Algebra or Local Algebra
- Invented by George Boole in 1854.

Rules for Boolean Algebra:

- variable can have only 2 values 0 and 1
- complement of a variable is represented by $(\bar{\ })$
- OR ing of the variable is represented by (+)
- like $A+B+C$
- AND ing of 2 or more variables represented with a dot $(A.B)$
  ex. $A.B.C$
- 1 for HIGH 0 for LOW

"LAWS"

| Law/Theorem | Law of Addition | Law of Multiplication |
|---|---|---|
| Identity | $x+0=x$ | $\cancel{x.1=1}$ $x.1=x$ |
| Complement | $x+x'=1$ | $x.x'=0$ |
| Idompetent | $x+x=x$ | $x.x=x$ |
| Dominant | $x+1=1$ | $x.0=0$ |
| Ivolution | $(x')'=x$ | |
| Commutative | $x+y=y+x$ | $x.y=y.x$ |
| Associative | $x+(y+z)=(x+y)+z$ | $x.(y.z)=(x.y).z$ |
| Distributive | $x.(y+z)=x.y+x.z$ | $x+y.z=(x+y).(x+z)$ |
| De morgan's | $(x+y)'=x'.y'$ | $(x.y)'=x'+y'$ |
| Absorption | $x+(x.y)=x$ | $x.(x+y)=x$ |

Date :
Page no.

## Simplification of Boolean Expressions

① $ABC + AB'C + ABC'$

$$
\begin{aligned}
F &= ABC + AB'C + ABC' \\
&= AC(B+B') + ABC' && B+B' = 1 \\
&= AC + ABC' \\
&= A(C+BC') && C+BC' = C+B. \\
&= A(C+B)
\end{aligned}
$$

② $AB' + (A' + B' + C \cdot C')'$

$$
\begin{aligned}
F &= AB' + (A' + B' + C \cdot C')' && C \cdot C' = 0 \\
&= AB' + (A' + B' + 0)' \\
&= AB' + (A' + B')' && (A+B)' = A' \cdot B' \\
&= AB' + (A'' \cdot B'') && A'' = A \\
&= AB' + AB \\
&= A(B+B') && B+B' = 1 \\
&= A
\end{aligned}
$$

### Map simplification

The map method provides a simple, straight forward procedure for simplifying boolean expressions

This method is a pictorial arrangement of the truth table which allows an easy ~~representation~~ interpretation for choosing the minimum number of terms required to express the function algebrically.

Date :
Page no.

## Simplification of Boolean Expressions

① $ABC + AB'C + ABC'$

$$
\begin{aligned}
F &= ABC + AB'C + ABC' \\
&= AC(B+B') + ABC' && B+B'=1 \\
&= AC + ABC' \\
&= A(C+BC') && C+BC' = C+B. \\
&= A(C+B)
\end{aligned}
$$

② $AB' + (A' + B' + C\cdot C')'$

$$
\begin{aligned}
F &= AB' + (A' + B' + C\cdot C')' && C\cdot C' = 0 \\
&= AB' + (A' + B' + 0)' \\
&= AB' + (A' + B')' && (A+B)' = A'\cdot B' \\
&= AB' + (A''\cdot B'') && A'' = A \\
&= AB' + AB \\
&= A(B+B') && B+B'=1 \\
&= A
\end{aligned}
$$

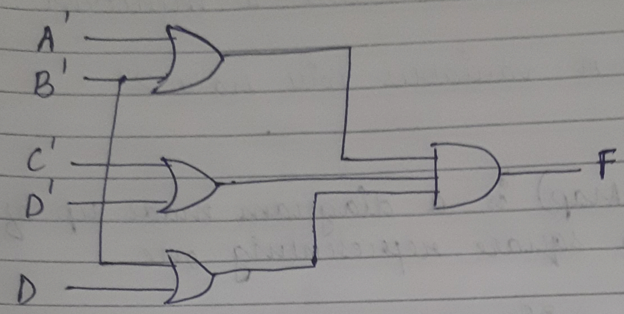
### Map simplification

The map method provides a simple, straight forward procedure for simplifying boolean expressions

This method is a pictorial arrangement of the truth table which allows an easy ~~representation~~ interpretation for choosing the minimum number of terms required to express the function algebrically.

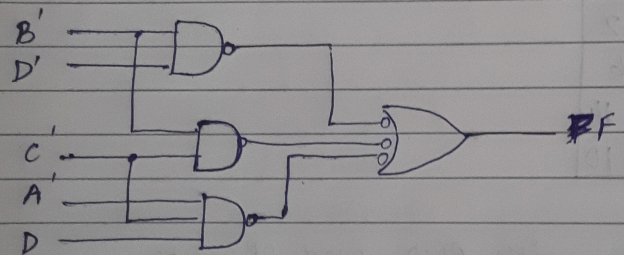
Date :
Page no.

(b) Product of sums $F = (A'+B')(C'+D')(B'+D)$



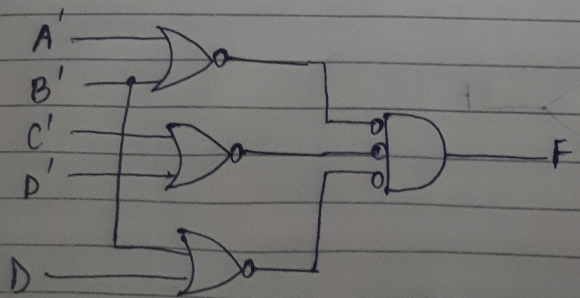

Nand and Nor Implementations for the same

(a) $F = B'D' + B'C' + A'C'D$. with NAND gates




Sum of products

(b) $F = (A'+B')(C'+D')(B'+D)$ with NOR gates




Product of sums.