Boolean Algebra

PRESENTED BY NABANITA DAS

Boolean Algebra

- VERY nice machinery used to simplify Boolean functions
- George Boole (1815-1864): "An investigation of the laws of thought"
- Terminology:
  - *Literal:* A variable or its complement
  - *Product term:* literals connected by •
  - *Sum term:* literals connected by +
- **Application of Boolean algebra**
  - It is used to perform the logical operations in digital computer.
  - In digital computer **True** represent by '1' **(high volt)** and **False** represent by '**0**' **(low volt)**
  - Logical operations are performed by logical operators. The fundamental logical operators are:
    - AND (conjunction)
    - OR (disjunction)
    - NOT (negation/complement)

Boolean Algebra Summary

- We can express Boolean functions with either an expression or a truth table.
- Every Boolean expression can be converted to a circuit.
- Now, we'll look at how Boolean algebra can help simplify expressions, which in turn will lead to simpler circuits.
- We recall that the two binary values have different names:
  - 1/0
  - TRUE/ FALSE
- We use 1 and 0 to denote the two values.
- The three basic logical operations are:
  - AND is denoted by a dot (·).
  - OR is denoted by a plus (+).
  - NOT is denoted by an overbar (¯) or a single quote mark (') after, or (~) before the variable

Boolean Operator Precedence

- **The order of evaluation is:**
  - Parentheses
  - NOT
  - AND
  - OR
- **Consequence: Parentheses appear around OR expressions**
- **Example:** $F = A(B + C)(C + D)$

Boolean Algebra Theorems

Identity Name	AND Form	OR Form
Identity Law	$1x = x$	$0+x = x$
Null (or Dominance) Law	$0x = 0$	$1+x = 1$
Idempotent Law	$xx = x$	$x+x = x$
Inverse Law	$x\bar{x} = 0$	$x+\bar{x} = 1$
Commutative Law	$xy = yx$	$x+y = y+x$
Associative Law	$(xy)z = x(yz)$	$(x+y)+z = x+(y+z)$
Distributive Law	$x+yz = (x+y)(x+z)$	$x(y+z) = xy+xz$
Absorption Law	$x(x+y) = x$	$x+xy = x$
DeMorgan's Law	$(\overline{xy}) = \bar{x}+\bar{y}$	$(\overline{x+y}) = \bar{x}\bar{y}$
Double Complement Law	$\bar{\bar{x}} = x$	

Duality

The dual of an expression is obtained by exchanging (• and +), and (1 and 0) in it, provided that the precedence of operations is not changed.

Cannot exchange x with x’

Duality (cont'd)

With respect to duality,

Identities 1 – 8 have the following relationship:

1. $X + 0 = X$ 2. $X \bullet 1 = X$ (dual of 1)
3. $X + 1 = 1$ 4. $X \bullet 0 = 0$ (dual of 3)
5. $X + X = X$ 6. $X \bullet X = X$ (dual of 5)
7. $X + X' = 1$ 8. $X \bullet X' = 0$ (dual of 7)

Power of Duality

9. $x + x \bullet y = x$ is true, so $(x + x \bullet y)' = x'$
10. $(x + x \bullet y)' = x' \bullet (x' + y')$
11. $x' \bullet (x' + y') = x'$
12. Let $X = x'$, $Y = y'$
13. $X \bullet (X + Y) = X$, which is the dual of $x + x \bullet y = x$.
14. Other Boolean Theorem (CONSENSUS THEOREM)
    - $XY + X'Z + YZ = XY + X'Z$
    - $(X+Y)(X' + Z)(Y + Z) = (X + Y)(X' + Z)$

Proof of XY + X'Z + YZ = XY + X'Z

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

Consider LHS => xy+yz+x'z

yz term has to be eliminated, so multiply yz by (x+x').

[Note : x+x' = 1]

= xy+yz(x+x')+x'z

= xy+xyz+x'yz+x'z

= xy(1+z)+x'z(1+y) [Note : 1+z = 1]

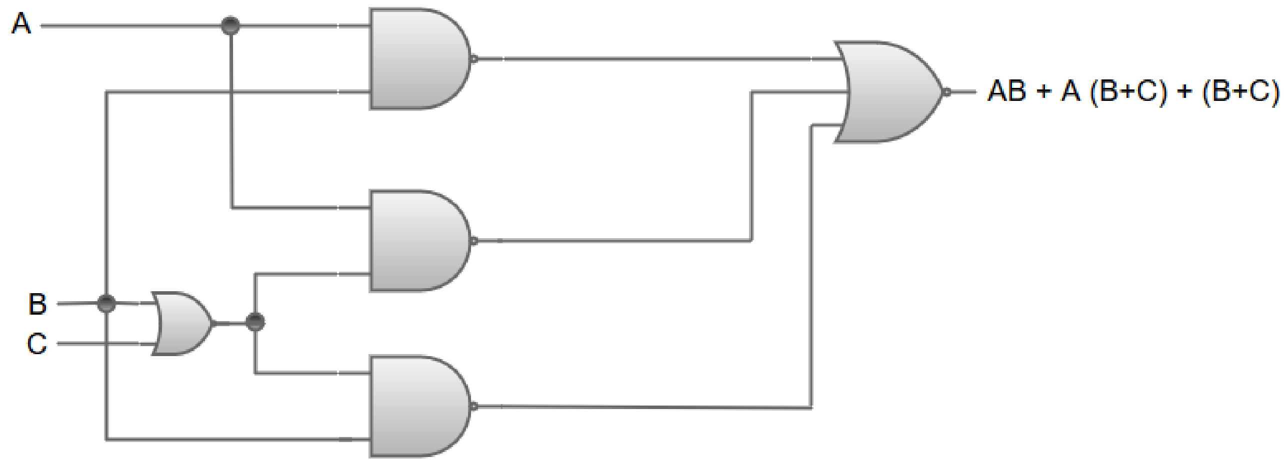
= xy(1)+x'z(1)

= XY + X'Z Proved.

Simplification using Boolean algebra

Let us consider an example of a Boolean function: **AB+A (B+C) + B (B+C)**

The logic diagram for the Boolean function AB+A (B+C) + B (B+C) can be represented as:



Simplification using Boolean algebra

We will simplify this Boolean function on the basis of rules given by Boolean algebra.

AB + A (B+C) + B (B+C)

=AB + AB + AC + BB + BC {Distributive law; A (B+C) = AB+AC, B (B+C) = BB+BC}

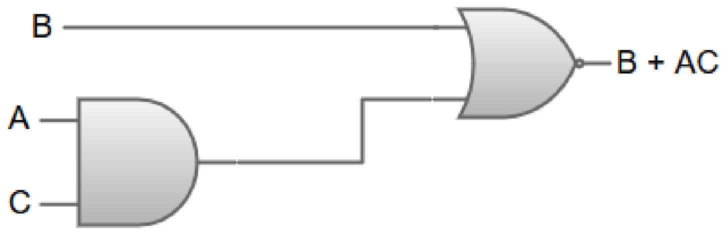
=AB + AB + AC + B + BC {Idempotent law; BB = B}

=AB + AC + B + BC {Idempotent law; AB+AB = AB}

=AB + AC +B {Absorption law; B+BC = B}

=B + AC {Absorption law; AB+B = B}

Hence, the simplified Boolean function will be B + AC.



The logic diagram for Boolean function B + AC

One more example of logic simplification

Example:

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

It can be proved as follows:

$(A + B)(A + C) = AA + AC + AB + BC$ Distributive law

$= A + AC + AB + BC$ Rule : $AA = A$

$= A(1 + C) + AB + BC$ Rule : $1 + C = 1$

$= A. 1 + AB + BC$

$= A(1 + B) + BC$ Rule : $1 + B = 1$

$= A. 1 + BC$ Rule : $A . 1 = A$

$= A + BC$

Converting from Truth Table to Boolean Function

- There are two ways to convert from truth tables to Boolean functions:
  1. Using Sum of Products / Minterms (SOP)
  2. Using Product of Sums / Maxterms (POS)

Sum of Product

- The sum-of-products (**SOP**) form is a method (or form) of simplifying the Boolean expressions of logic gates.
- Sum and product derived from the symbolic representations of the OR and AND functions.
- OR (+) , AND (.) , addition and multiplication.

Sum

$$f(A,B,C) = ABC + A'BC'$$

Product terms

Product of Sum

- When two or more sum terms are multiplied by a Boolean OR operation.
- Sum terms are defined by using OR operation and the product term is defined by using AND operation.

Product

$$f(A,B,C) = (A'+B) \, . \, (B+C')$$

Sum terms

Canonical form of Boolean Expression (Standard form)

- The canonical forms are the special cases of SOP and POS forms.
- In standard **SOP** and **POS** each term of Boolean expression must contain all the literals (with and without bar) that has been used in Boolean expression.
- If the above condition is satisfied by the Boolean expression, that expression is called Canonical form of Boolean expression.
- Each individual term in the POS form is called **Maxterm**.
- Each individual term in the SOP form is called **Minterm**.
- We perform **Sum of Minterm** also known as Sum of products (SOP) .
- We perform **Product of Maxterm** also known as Product of sum (POS).

Convert SOP to standard SOP form

1. For getting the standard SOP form of the given non-standard SOP form, we will add all the variables in each product term which do not have all the variables. By using the Boolean algebraic law, $(x + x' = 1)$
2. Repeat **Step 1**, until all resulting product terms contain all variables.
3. For each missing variable in the function, the number of product terms doubles.

Example:

$$f(A,B,C) = AB + BC + AC$$

Step 1: Find the missing literals in each product term.

$$f(A,B,C) = AB + BC + AC$$

Literal B is missing

Literal A is missing

Literal C is missing

Step 2: AND the product term with missing literal + its complement.

$$f(A,B,C) = AB \cdot (C+C') + BC \cdot (A+A') + AC \cdot (B+B')$$

Missing literals and their complements

Step 3: Expends the term and reorder the literals.

$$f(A,B,C) = AB \cdot (C+C') + BC \cdot (A+A') + AC \cdot (B+B')$$

Expand & Reorder:

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

Step 4: Omit repeated product terms.

$$f(A,B,C)=ABC + ABC' + ABC + A'BC + ABC + AB'C$$

$$f(A,B,C)= ABC + ABC' + A'BC + AB'C$$

Convert POS to standard POS form

1. For getting the standard POS form of the given non-standard POS form, we will add all the variables in each product term that do not have all the variables. By using the Boolean algebraic law ($x * x' = 0$).
2. Applying Boolean algebraic law, $x + y z = (x + y) * (x + z)$
3. By repeating step 1, until all resulting sum terms contain all variables

Example:

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

Step 1: Find The Missing Literals In Each Sum Term.

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

Literal B is missing

Literal A is missing

Literal C is missing

Step 2: OR the sum term with missing literal . Its complement.

$$f(A,B,C) = (A + B+(C.C')). (B + C+(A.A')) \,.\, (A + C+(B.B'))$$

Missing literals and their complements

Step 3: Expends The Term And Reorder The Literals.

f (A,B,C) = (A + B+(C.C’)). (B + C+(A.A’)) . (A + C+(B.B’))

Expand & Reorder:

by applying, distributive law [A + BC = (A + B) (A + C)]

f(A,B,C)=(A+B+C).(A+B+C’).(A+B+C).(A’+B+C).(A+B+C).(A+B’+C)

Step 4: Omit repeated sum terms.

f(A,B,C)=(A+B+C).(A+B+C’).(A+B+C).(A’+B+C).(A+B+C).(A+B’+C)

f(A,B,C)=(A+B+C).(A+B+C’).(A’+B+C).(A+B’+C)

Minterms

- Minterms are AND terms with every variable present in either true or complemented form.
- Given that each binary variable may appear normal (e.g., x) or complemented (e.g.x'), there are 2^n minterms for n variables.

Example: Two variables (X and Y)produce 2 x 2 = 4 combinations:

XY(both normal)

XY'(X normal, Y complemented)

X'Y(X complemented, Y normal)

X'Y'(both complemented)

Thus there are four minterms of two variables.

Maxterms

- Maxterms are OR terms with every variable in true or complemented form.
- Given that each binary variable may appear normal (e.g., x) or complemented (e.g. x’), there are 2^n maxterms for n variables.

Example: Two variables (X and Y) produce 2x2=4 combinations:

X+Y(both normal)

X+Y’(x normal, y complemented)

X’+Y(x complemented, y normal)

X’+Y’(both complemented)

Minterms & Maxterms for **3** variables

A	B	C	Minterms	Maxterms
0	0	0	$A'B'C' = m_0$	$A+B+C = M_0$
0	0	1	$A'B'C = m_1$	$A+B+C' = M_1$
0	1	0	$A'BC' = m_2$	$A+B'+C = M_2$
0	1	1	$A'BC = m_3$	$A+B'+C' = M_3$
1	0	0	$AB'C' = m_4$	$A'+B+C = M_4$
1	0	1	$AB'C = m_5$	$A'+B+C' = M_5$
1	1	0	$ABC' = m_6$	$A'+B'+C = M_6$
1	1	1	$ABC = m_7$	$A'+B'+C' = M_7$

Converting from Truth Table to Boolean Function

Truth Table *to* Boolean Function

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$$F = \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + ABC$$

Using Minterms

Converting from Truth Table to Boolean Function

Truth Table *to* Boolean Function

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

A	B	C	F'
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Using Maxterms

Converting from Truth Table to Boolean Function

Sum of Minterms

$$F = \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + ABC$$

$$F = m_1 + m_4 + m_5 + m_7$$

$$\boxed{F = \sum (1,4,5,7)}$$

Product of Maxterms

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

$$F = M_0 \quad M_2 \quad M_3 \quad M_6$$

$$F = \boxed{\prod (0,2,3,6)}$$

FOR MINTERM (SOP):
A=1
B=1
C=1

FOR MAXTERM (POS):
A=0
B=0
C=0

	A	B	C	F	F'
0	0	0	0	0	1
1	0	0	1	1	0
2	0	1	0	0	1
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	1	0
6	1	1	0	0	1
7	1	1	1	1	0