(T)EE2026
Digital Fundamentals
Boolean Algebra

Massimo Alioto

Dept of Electrical and Computer Engineering

Email: ***massimo.alioto@nus.edu.sg***

Outline

- What is Boolean Algebra
- Theorems
- Boolean functions and truth table
- Boolean function simplification using algebra manipulation

What is Boolean Algebra?

Brief History:

- Boolean was developed in 1854 by George Boole (An English mathematician, philosopher, and logician)
- Huntington formulated the postulates in1904 as the formal definition
- Boolean Algebra is the mathematical foundation for digital system design, including computers
- It was first applied to the practical problem (Analysis of networks of relays) in late 1930s by C.E Shannon (MIT) who later introduced "Switching algebra" in 1938
- Switching algebra is a Boolean algebra in which the number of elements is precisely two

Boolean Algebra

- Boolean algebra is defined by
  - a set of elements, ***B***, and
  - two binary operators, $\cdot\,(AND), +(OR)$
  - unary operator $\overline{}$ (*NOT*)
- Boolean algebra satisfies six Huntington postulates

*Elements → integer number (i.e. 0, 1, …)

*Variables → symbols (i.e. x, y, z, …) are natural numbers

Postulates of Boolean Algebra

Six Huntington postulates:

There are 6 Huntington Postulates that define the Boolean Algebra:

1. Closure - For all elements x and y in the set ***B***
   - *i.* $x + y$ *is an element of* $\boldsymbol{B}$ *and*
   - *ii.* $x \cdot y$ *is an element of* $\boldsymbol{B}$
2. There exists a 0 and 1 element in ***B***, such that
   - *i.* $x + 0 = x$
   - ii. $x \cdot 1 = x$
3. Commutative Law
   - i. $x + y = y + x$
   - ii. $x \cdot y = y \cdot x$

Postulates of Boolean Algebra – cont.

4. Distributive Law
   - *i.* $x \cdot (y + z) = x \cdot y + x \cdot z$ $\quad(\cdot\ over +)$
   - *ii.* $x + (y \cdot z) = (x + y) \cdot (x + z)$ $\quad(+\ over\ \cdot)$

5. For every element x in the set ***B***, there exists an element $\bar{x}$ in the set ***B***, such that
   - *i.* $x + \bar{x} = 1$
   - *ii.* $x \cdot \bar{x} = 0$

   ($\bar{x}$ is called the **complement** of x)

6. There exist at least two distinct elements in the set ***B***

Switching Algebra

- Switching algebra is a <u>two-valued</u> Boolean Algebra, that is, the number of elements in the set ***B*** is two {0,1}
- Switching algebra represents bistable electrical switching circuits (On or Off)
- There are two main operators (***AND, OR***)
  - Binary operators (two arguments involved)
    - ***AND*** → "**.**"
    - ***OR*** → "**+**"
  - Plus, one unary operator (only one argument involved)
    - ***NOT*** → "$\overline{\ }$" (*Complement* operator represented by an overbar)
- Switching algebra satisfies six Huntington postulates

The Three Operators in Two-Valued Boolean Algebra (*B={0,1}*)

OR: $A + B$

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

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 1$$

AND: $A \cdot B$

A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

NOT: $\overline{A}$

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

$$\mathrm{A} = 0 \rightarrow \overline{A} = 1$$
$$\mathrm{A} = 1 \rightarrow \overline{A} = 0$$

Priority: NOT has highest precedence, followed by AND and OR
NOT(A · B + C) = NOT((A · B) + C)

Boolean vs. Ordinary Algebra

Boolean algebra	Ordinary algebra
No associative law. But it can be derived from the other postulates	Associative law is included: $a + (b + c) = (a + b) + c$
Distributive law: $x + (y \cdot z) = (x + y) \cdot (x + z)$ valid	Not valid
No additive or multiplicative inverses, therefore there are no subtraction and division operation	Subtraction and division operations exist
Complement operation available	No complement operation
Boolean algebra: Undefined set of elements; Switching algebra: a two-valued Boolean algebra, whose element set only has two elements, 0 and 1.	Dealing with real numbers and constituting an infinite set of elements

Theorems of Boolean Algebra

#	Theorem		
1	$A + A = A$	$A \cdot A = A$	Tautology Law
2	$A + 1 = 1$	$A \cdot 0 = 0$	Union Law
3	$\overline{(\bar{A})} = A$		Involution Law
4	$A + (B + C)$ $= (A + B) + C$	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	Associative Law
5	$\overline{A + B} = \bar{A} \cdot \bar{B}$	$\overline{A \cdot B} = \bar{A} + \bar{B}$	De Morgan's Law
6	$A + A \cdot B = A$	$A \cdot (A + B) = A$	Absorption Law
7	$A + \bar{A} \cdot B = A + B$	$A \cdot (\bar{A} + B) = A \cdot B$	
8	$AB + A\bar{B} = A$	$(A + B)(A + \bar{B}) = A$	Logical adjacency
9	$AB + \bar{A}C + BC$ $= AB + \bar{A}C$	$(A + B)(\bar{A} + C)(B + C)$ $= (A + B)(\bar{A} + C)$	Consensus Law

Duality (OR and AND, 0 and 1 can be interchanged)

When is the output of an AND gate equal to 1?

when all inputs are 1

when some of the inputs are 1

when some of the inputs are 0

A + A = ?

When poll is active, respond at **PollEv.com/massimoaliot866**

Top

A + NOT(A) * B = ?

When poll is active, respond at **PollEv.com/massimoaliot866**

Top

Boolean Functions and Truth Table

- A Boolean function expresses the logical relationship between binary variables
- It is evaluated by determining the binary value of the expression for all possible values of the variables
- Examples

$$F_1 = A + B \qquad F_3 = A + BC$$

$$F_2 = A \cdot B \qquad F_4 = \bar{A}\bar{B}C + AB\bar{C}$$

Truth Table

- ***Truth table*** is a tabular technique for listing all possible combinations of input variables and the values of function for each combination.

$$F_1 = A + B$$

A	B	F_1
0	0	0
0	1	1
1	0	1
1	1	1

$$F_3 = A + BC$$

A	B	C	F_3
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Truth Table - examples

Prove the De Morgan's Law:

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

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

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

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

Prove : $A + \bar{A} \cdot B = A + B$

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

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

A	B	$A \cdot (A + B)$	A
0	0	0	0
0	1	0	0
1	0	1	1
1	1	1	1

Truth Table – examples (cont.)

Prove : $A + (B \cdot C) = (A + B) \cdot (A + C)$

A	B	C	$A + (B \cdot C)$	$(A + B) \cdot (A + C)$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

SOP and POS

SOP → Sum of Products

Example:

$$F_1(A,B,C) = \overbrace{\overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}C}^{\textit{Sum}}$$

Products

POS → Product of Sums

$$F_2(A,B,C) = \overbrace{(A+B+C)(A+\overline{B}+\overline{C})\cdot(\overline{A}+B+C)(\overline{A}+\overline{B}+C)(\overline{A}+\overline{B}+\overline{C})}^{\textit{Products}}$$

Sums

Minterm and Maxterm

- ***Minterm*** is a <u>*product term*</u> that contains all variables in the function
- ***Maxterm*** is a <u>*sum term*</u> that contains all variables in the function
- For ***n*** variables, there are 2^n different *minterms* or *maxterms*
- For example, in a Boolean Function: $Z = f(A, B, C)$
  - $ABC,\ A\bar{B}\bar{C},\ \bar{A}BC$ are *minterms* in SOP (contain all variables)
  - $AB,\ \bar{A}C, BC$ are ***not*** *minterms* in SOP
  - $(A + B + C),\ (\bar{A} + \bar{B} + C),\ (A + B + \bar{C})$ are *maxterms* in POS (contain all variables)
  - $(A + C),\ (B + C),\ (\bar{A} + \bar{B})$ are ***not*** *maxterms* in POS

Minterm and Maxterm – cont.

A	B	C	F	Minterm	Maxterm
0	0	0	0	$\bar{A} \cdot \bar{B} \cdot \bar{C}$	$A + B + C$
0	0	1	0	$\bar{A} \cdot \bar{B} \cdot C$	$A + B + \bar{C}$
0	1	0	1	$\bar{A} \cdot B \cdot \bar{C}$	$A + \bar{B} + C$
0	1	1	0	$\bar{A} \cdot B \cdot C$	$A + \bar{B} + \bar{C}$
1	0	0	1	$A \cdot \bar{B} \cdot \bar{C}$	$\bar{A} + B + C$
1	0	1	1	$A \cdot \bar{B} \cdot C$	$\bar{A} + B + \bar{C}$
1	1	0	1	$A \cdot B \cdot \bar{C}$	$\bar{A} + \bar{B} + C$
1	1	1	0	$A \cdot B \cdot C$	$\bar{A} + \bar{B} + \bar{C}$

minterms (maxterms) that are equal to 1 (0) only for a given input

- Minterm
  - AND all the variables
  - If the variable in truth table is “0”, take its complement in the minterm
- Maxterm
  - OR all the variables
  - If the variable in truth table is “1”, take its complement in the maxterm

Canonical Form

- A Boolean functions is said to be in canonical form if it is expressed as
  - a ***sum*** of ***minterms*** (Canonical SOP - CSOP) or
  - a ***product*** of ***maxterms*** (Canonical POS - CPOS)

SOP: $F_1(A,B,C) = \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C}$ ← Canonical form

$F_{1a}(A,B,C) = \overline{A}B\overline{C} + A\overline{B} + \overline{B}C + AB\overline{C}$ ← Non Canonical form

Non *minterms*

POS: $F_2(A,B,C) = (A+B+C)(A+B+\overline{C})(A+\overline{B}+\overline{C})$ ← Canonical form

$F_2(A,B,C) = (A+B+C)(A+\overline{C})(A+\overline{B}+\overline{C})$ ← Non Canonical form

Non *maxterm*

SOP and POS → Truth Table

Are the following two Boolean functions same?

$$F_1(A,B,C) = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}C$$

$$F_2(A,B,C) = (A+B+C)(A+\overline{B}+\overline{C})(\overline{A}+B+C)(\overline{A}+\overline{B}+C)(\overline{A}+\overline{B}+\overline{C})$$

Let's use truth table to check:

Truth table

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

SOP:

- If any *PRODUCT* in SOP is "1", the function is "1". Otherwise, the function is "0"

POS:

- If any *SUM* in POS is "0", the function is "0". Otherwise, the function is "1"

- **SOP and POS are different ways to present the same Boolean function**

Truth Table → CSOP or CPOS

Write the Boolean function represented by the Truth table below in SOP and POS, respectively

Truth table:

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

$$F_1(A,B,C) = \overline{A}B\overline{C} + A\overline{B}\overline{C} + A\overline{B}C + AB\overline{C}$$

- CSOP → Only includes the terms that make F = 1

$$F_2(A,B,C) = (A+B+C)(A+B+\overline{C})(A+\overline{B}+\overline{C})(\overline{A}+\overline{B}+\overline{C})$$

- CPOS → Only includes the terms that make F = 0

Any Boolean function can be obtained from a given truth table and expressed in either CSOP or CPOS
(if you can choose, pick CSOP if truth table has few 1's and many 0's, CPOS otherwise)

Why did we introduce the canonical form (CSOP, CPOS)?

Truth Table → SOP → POS

Truth table:

A	B	C	F_1	$\overline{F_1}$
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

Use SOP: start from CSOP of NOT(F) (otherwise, complemented POS is obtained from SOP)

$$\overline{F_1}(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + ABC$$

Apply De Morgan's Law:

$$F_1(A,B,C) = \overline{\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + ABC}$$

Use POS directly from truth table:

clearly the same

$$F_1(A,B,C) = (A+B+C)(A+B+\bar{C})(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+\bar{C})$$

POS can be obtained from SOP (and vice versa) by starting from complemented SOP of F and applying the De Morgan's Law

Example-1: Non-Canonical → Canonical Form via Truth Table

Example: For the given Boolean function below, find a canonical *minterm* and *maxterm* expression.

1) obtain the truth table from the given function
2) find *minterm* or *maxterm* expression from truth table (CSOP or CPOS)

$F(x,y,z) = \bar{x} + y\bar{z}$ ← Non Canonical form

Truth table:

x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Canonical *minterm* expression:

$$F(x,y,z) = \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z + \bar{x}y\bar{z} + \bar{x}yz + xy\bar{z}$$

(only contains the *minterms* that make the function = 1)

Canonical *maxterm* expression:

$$F(x,y,z) = (\bar{x} + y + z)(\bar{x} + y + \bar{z})(\bar{x} + \bar{y} + \bar{z})$$

(only contains the *maxterms* that make the function = 0)

Example-2: Non-Canonical → Canonical Form via Postulates and Theorems

Example: For the given Boolean functions below, convert it to canonical *minterm or maxterm* expression.

(*Using postulates/theorem to expand the given function to canonical form)

SOP → CSOP:
(CSOP – Canonical SOP)

$$F(x,y,z) = \bar{x}y + xz$$
$$= \bar{x}y \cdot 1 + x \cdot 1 \cdot z$$
$$= \bar{x}y(z+\bar{z}) + x(y+\bar{y})z$$
$$= \bar{x}yz + \bar{x}y\bar{z} + xyz + x\bar{y}z$$

For missing literals, complete minterms through postulates:
$A \cdot 1 = A \quad and \quad A + \bar{A} = 1$

SOP → CPOS:
(CSOP – Canonical POS)

1)
$$F(x,y,z) = \bar{x}y + xz$$
$$= \overline{\overline{\bar{x}y + xz}} \quad \text{a}$$
$$= \overline{(x+\bar{y})(\bar{x}+\bar{z})} \quad \text{b}$$
$$= \overline{x\bar{x} + x\bar{z} + \bar{x}\bar{y} + \bar{y}\bar{z}} \quad \text{c}$$
$$= (x+y)(\bar{x}+z)(y+z) \quad \text{d}$$
2)
$$= (x+y+z)\cdot(x+y+\bar{z})\cdot(\bar{x}+y+z)\cdot(\bar{x}+\bar{y}+z)$$
$$\cdot(x+y+z)\cdot(\bar{x}+y+z)$$

1) express SOP as POS
- a) complement twice
- b) apply De Morgan's law
- c) expand
- d) re-apply De Morgan's law

2) for missing literals, complete maxterms through distribution postulate

Use distribution postulate:
$A + (BC) = (A+B)(A+C)$
(A = incomplete sum,
C=NOT(B)=missing literal)

$x+y = (x+y) + z\bar{z}$
$= (x+y+z)(x+y+\bar{z})$

Summary

- Postulates and theorems of Boolean algebra
- Three binary operators: AND, OR and NOT
- Boolean Functions
- Truth table and Boolean function evaluation using truth table
- Boolean function in SOP or POS form
- Obtain SOP or POS from truth table
- Minterm and maxterm
- Canonical form of Boolean function
- Convert non-canonical form to canonical SOP or POS expressions.