

Boolean Algebra

Table of contents

Notation conventions	1
Boolean identities	1
Sum of products	2

Logic functions can be expressed as expressions of variables (literals, e.g. A, B, X) and functions (e.g. $+, \cdot, \oplus, \bar{}$). Variables and functions can only take values 0 or 1.

Notation conventions

- **Inversion:** overline, e.g. $\text{NOT}(A) = \bar{A}$ (“A bar”).
- **AND:** dot, or implied by adjacency, e.g. $\text{AND}(A, B) = AB = A \cdot B$.
- **OR:** plus sign, e.g. $\text{OR}(A, B, C) = A + B + C$.
- Other examples: $\text{XOR}(A, B) = A \oplus B = \bar{A}B + A\bar{B}$; $\text{NAND}(A, B, C) = \overline{ABC}$; $\text{NOR}(A, B) = \overline{A + B}$.

Boolean identities

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan’s law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

De Morgan's law also means **AND** and **OR** gates can be interchanged if you invert all the inputs and the output — this is why NAND/NOR-only circuits (see logic-gates¹) can implement any function.

Sum of products

Any logic function can be implemented as the **OR of AND combinations** of its inputs (a *sum of products*):

1. For each row where the truth table output is 1, write the AND term of the inputs (complementing wherever the input is 0) that produces that row.
2. OR all of those terms together.

This always works, but doesn't necessarily give the minimum number of gates — the resulting expression can often be simplified further using the identities above. E.g. $Z = AB + AC = A(B + C)$ (distributive law) uses fewer gates than the raw sum-of-products form.

¹courses/csse2010/logic-gates.pdf