Boolean algebra

A **boolean algebra** models logical statements based on two values and relationships “and” (**conjunction**), “or” (**disjunction**) and “not” (**negation**). The values are usually called “true” and “false”, or “1” and “0” in circuit design.
If A is a boolean variable, i.e. a name whose interpretation can only either be “true” or “false”, then

$$A = 1$$

reads “A is true”, and

$$A = 0$$

reads “A is false.”

Boolean algebra/Operators

The three operators of a boolean algebra are as follows.

The binary operator **and**, symbolised by a dot: $A \cdot B$ reads "A and B." If the meaning is clear, the dot can be omitted, as in AB. This operator is defined by the equations

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

The binary operator **or**, symbolised by '$+$': $A + B$ reads "A or B." This operator is defined by the equations

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

Boolean algebra/Operators (cont)

The last operator is the unary operator **negation**, symbolised by a bar over its argument. It is also called **complement**, or **one-complement** but, be careful: in the context of 2-complement binary numbers (page 23), for example, negation is **not** the complement.
This operator is simply defined as

$$\overline{1} = 0$$
$$\overline{0} = 1$$

Warning: Page 36, $\overline{B}$ means the 1-complement of an **integer**, not one bit. In other words, B was *not* a boolean variable.

Boolean algebra/Identities

For any boolean variable A, the following identities hold:

$$1 + A = 1 \tag{3}$$

$$0 + A = A \tag{4}$$

$$A + A = A \tag{5}$$

$$A + \overline{A} = 1 \tag{6}$$

$$0 \cdot A = 0 \tag{7}$$

$$1 \cdot A = A \tag{8}$$

$$A \cdot A = A \tag{9}$$

$$A \cdot \overline{A} = 0 \tag{10}$$

$$\overline{\overline{A}} = A \tag{11}$$

Boolean algebra/Identities

These identities can be proved by replacing A by 0 and 1, and use the definitions of "and", "or" and "not" pages 48 and 49. In the same way, we can prove:

Commutative laws

$$A + B = B + A \tag{12}$$

$$A \cdot B = B \cdot A \tag{13}$$

Distributive laws

$$A + (B \cdot C) = (A + B)(A + C) \tag{14}$$

$$A(B + C) = (A \cdot B) + (A \cdot C) \tag{15}$$

Boolean algebra/Identities (cont)

It is common to give precedence to negation over conjunction, which in turn has precedence over disjunction. This allows to omit some parentheses and, for instance, rewrite the distributive laws as

$$
\begin{aligned}
A + BC &= (A + B)(A + C) \\
A(B + C) &= AB + AC
\end{aligned}
$$

Boolean algebra/Identities (cont)

Then, we have more identities:

Absorption laws

$$A + AB = A \tag{16}$$

$$A(A + B) = A \tag{17}$$

Logic adjacency

$$AB + A\overline{B} = A \tag{18}$$

$$(A + B)(A + \overline{B}) = A \tag{19}$$

Boolean algebra/Identities (cont)

De Morgan

$$\overline{A+B} = \overline{A}\cdot\overline{B} \tag{20}$$

$$\overline{AB} = \overline{A}+\overline{B} \tag{21}$$

For example

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

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

Boolean algebra/Truth tables

One way to define a boolean function is to write its **truth table**, i.e. a table whose columns are the arguments of the function and its result, and the rows provide all the cases of 0 and 1 for the arguments. For example, function $L(A, B, S)$ is defined by

A	B	S	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Boolean algebra/Min-terms

Consider a boolean function $F(A, B)$, defined by the truth table

A	B	AB	$A\overline{B}$	$\overline{A}B$	$\overline{A}\,\overline{B}$	$F(A,B)$
0	0	0	0	0	1	1
0	1	0	0	1	0	0
1	0	0	1	0	0	0
1	1	1	0	0	0	1

We inserted in the truth table the values of the boolean expressions AB, $A\overline{B}$, $\overline{A}B$ and $\overline{A}\,\overline{B}$, which are called **min-terms**. The interesting point is that each min-term is true only for **one** combination of values of A and B.

This leads to a way of finding a definition of F based on boolean expressions instead of a truth table.

Boolean algebra/Min-terms (cont)

Indeed, for each combination of values of A and B such that $F(A, B)$ is 1, consider the corresponding min-term and make a disjunction of them. We get here

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

Why does it work? For each combination (A, B) for which $F(A, B) = 1$, one of the min-terms is 1, i.e., $\overline{A}\,\overline{B} = 1$ or $AB = 1$, so the disjunction is also 1. For each combination (A, B) for which $F(A, B) = 0$, both min-terms will be 0, so is their disjunction.

Boolean algebra/Min-terms (cont)

Therefore, given a truth table of a boolean function, form the expression made of the disjunction of the min-terms for which the result is 1.
This form is called, in general, **normal disjunctive form**, or **sum of product** (**SOP**).
Counter-example: $\overline{A+B}$ is **not** a SOP: the negations must apply to variables or 0 or 1.
Any boolean expression can be transformed, by means of the identities we gave, into a SOP.