Boolean Algebra

What is an Algebra?

What is an Algebra? (e.g. algebra of integers)

set of elements (e.g. 0,1,2,..)

set of operations (e.g. +, -, *,..)

postulates/axioms (e.g. 0 + x = x,..)

- **Boolean Algebra** named after George Boole (1815-1864) who used it to study human logical reasoning – calculus of proposition.
- simply a way of comparing individual bits.
- Uses operators to determine how the bits are compared.

Boolean Algebra Introduction

Boolean algebra forms the basis of logic circuit design. Consider very simple but common example: if (A is true) and (B is false) then print “the solution is found”. In this case, two Boolean expressions (A is true) and (B is false) are related by a connective ‘and’. How do we define these?

Boolean operators

- Events : *true* or *false*
- Connectives : a *OR* b; a *AND* b, *NOT* a
- The operators used most often are **AND** and **OR**.
  - The **AND** operation
    - if and only if **all** inputs are on, the output will be on.
    - The output will be off if any of the inputs are off.
  - The **OR** operation
    - **If any** input is on, the output will be on.
- Example: Either “it has rained” *OR* “someone splashed water”, “must be tall” *AND* “good vision”.
- It's easy to see all of the combinations by using what are called **truth tables**

Truth Tables

Truth tables show the result of combining boolean expressions of a logic function by listing all possible values the function can attain

a	b	a AND b
F	F	F
F	T	F
T	F	F
T	T	T

a	b	a OR b
F	F	F
F	T	T
T	F	T
T	T	T

a	NOT a
F	T
T	F

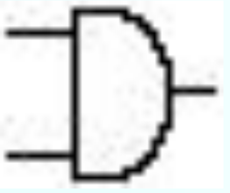
Truth Table: The AND operations

- the output will be on (or 1) if and only if **all** inputs are on (or 1).
- The output will be off (or 0) if any of the inputs are off (or 0)

TRUTH TABLE

AND OPERATIONS		
A	**B**	**C=A.B**
0	0	0
0	1	0
1	0	0
1	1	1

The **AND** Gate



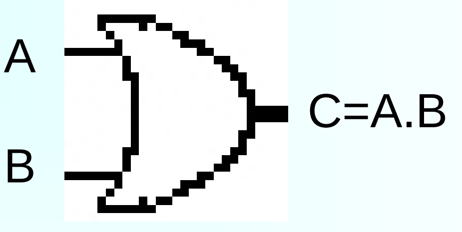
Truth Table: The OR operations

- the output will be on (or 1) if **ANY of** inputs are on (or 1).
- The output will be off (or 0) if **ALL** inputs are off (or 0)

TRUTH TABLE

OR OPERATIONS		
A	**B**	**C=A+B**
0	0	0
0	1	1
1	0	1
1	1	1

The **OR** Gate



Truth Table: the inverter or NOT operation

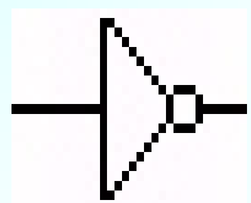
- has only **<u>one</u>** input and one output
- The output will be in **<u>opposite state</u>** to the input
- If input is 0, output is 1; If input is 1, output is 0;

<u>TRUTH TABLE</u>

The NOT Operations	
A	**C=** $\overline{\mathbf{A}}$
0	1
1	0

<u>The **NOT** Gate</u>

A



C= $\overline{A}$

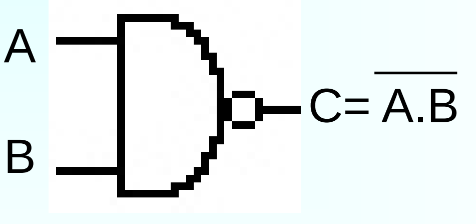
Truth Table: The NAND operations

- Has the same logic as **AND** operation, but with an inverted output.
- the output will be **OFF (or 0)** if and only if all inputs are **ON (or 1)**,.
- The output will be **ON (or 1)** if any of the inputs are **OFF (or 0)**

TRUTH TABLE

NAND OPERATIONS		
A	**B**	**C=** $\overline{\textbf{A.B}}$
0	0	1
0	1	1
1	0	1
1	1	0

The **NAND** Gate



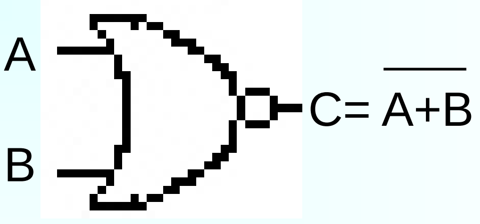
Truth Table: The NOR operation

- Inverts the OR operation
- if any input is **ON (or 1)**, the output will be **OFF (or 0)**

TRUTH TABLE

NOR OPERATIONS		
A	**B**	**C=$\overline{A+B}$**
0	0	1
0	1	0
1	0	0
1	1	0

The **NOR** Gate



Summary of 2-input logic gates

Inputs		Truth Table Outputs For Each Gate					
A	B	AND	NAND	OR	NOR	EX-OR	EX-NOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

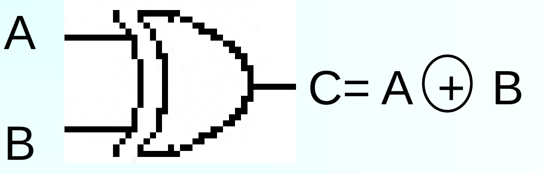
Truth Table: The Exclusive-OR or XOR operation

- the output will be **ON (or 1)** if the inputs are **DIFFERENT**.

TRUTH TABLE

XOR OPERATIONS		
A	**B**	**$C=A\oplus B$**
0	0	0
0	1	1
1	0	1
1	1	0

The **XOR** Gate



common logic functions and their equivalent Boolean notation.

Logic Function	Boolean Notation
AND	$A.B$
OR	$A+B$
NOT	$\overline{A}$
NAND	$\overline{A.B}$
NOR	$\overline{A+B}$
EX-OR	$(A.\overline{B}) + (\overline{A}.B)$ or $A \oplus B$
EX-NOR	$\overline{A}.\overline{B} + A.B$ or $\overline{A \oplus B}$

Example

For all possible combinations of A and B, draw the truth table for
$C = A.B + \overline{A}$

A	**B**	**A.B**	**$\overline{A}$**	**$A.B+\overline{A}$**
0	0	0	1	1
0	1	0	1	1
1	0	0	0	0
1	1	1	0	1