UNIVERSITAS
GADJAH MADA

Boolean Algebra

Azkario Rizky Pratama

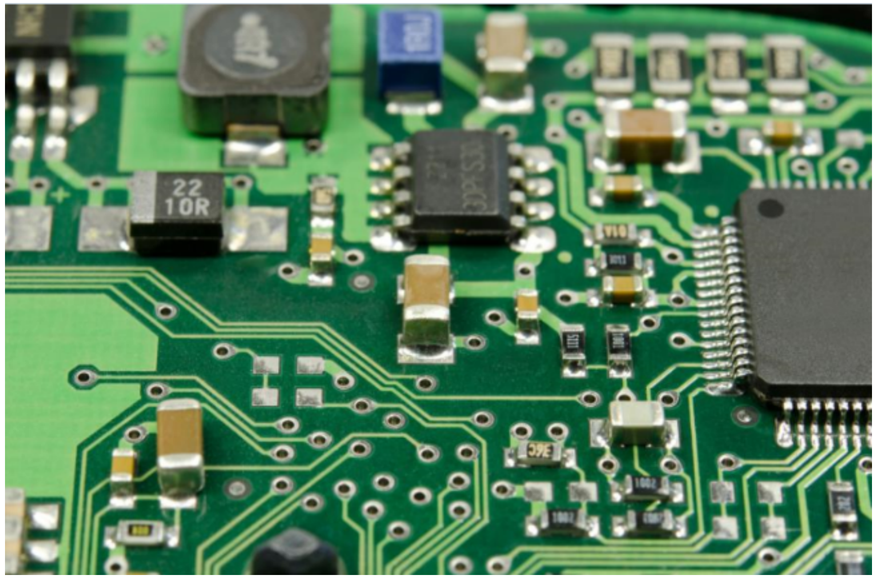
Overview

UNIVERSITAS GADJAH MADA

- Boolean Expressions
- Representing Boolean Functions
- Logic Gates
- Karnaugh Maps

Why is this important?

- We study Boolean algebra as a foundation for designing and analyzing digital systems!
- Boolean Algebra: a useful mathematical system for specifying and transforming logic functions.



allpcb.com

Boolean Expressions

Complement, Sum, Product

- Correspond to logical NOT, OR, and AND.
- We will denote the two logic values as 0:≡F and 1:≡T, instead of **False** and **True**.
  - Using numbers encourages algebraic thinking.
- New, more algebraic-looking notation for the most common Boolean operators:

$\bar{x} :\equiv \neg x$ $\quad x \cdot y :\equiv x \wedge y$ $\quad x + y :\equiv x \vee y$

Precedence order→

- Boolean values:
  - True
  - False
- Operation
  - Conjunction (AND)
    - A · B; A ∧ B
  - Disjunction (OR)
    - A + B; A ∨ B
  - Negation (NOT)
    - Ā ; ¬A ; A′

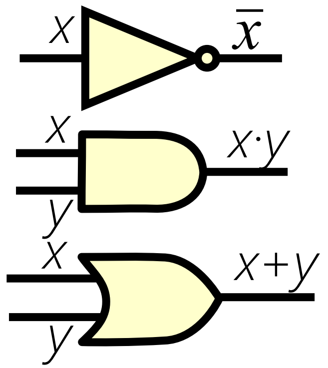
- Truth tables

A	B	A • B
0	0	0
0	1	0
1	0	0
1	1	1

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

A	A'
0	1
1	0

- Logic gates



Boolean Functions

- Let $B = \{0, 1\}$, the set of Boolean values.
- For all $n \in \mathbf{Z}^+$, any function $f{:}B^n \rightarrow B$ is called a *Boolean function of degree n*.
- There are 2^{2^n} (wow!) distinct Boolean functions of degree n.
  - B/c $\exists\ 2^n$ rows in truth table, w. 0 or 1 in each.

Degree	How many	Degree	How many
0	2	4	65,536
1	4	5	4,294,967,296
2	16	6	18,446,744,073,709,551,616.
3	256		

Boolean Expressions

UNIVERSITAS GADJAH MADA

- Let $x_1, \ldots, x_n$ be n different Boolean variables.
  - n may be as large as desired.
- A *Boolean expression* (recursive definition) is a string of one of the following forms:
  - Base cases: 0, 1, $x_1, \ldots,$ or x_n.
  - Recursive cases: $\overline{E_1}$, (E_1E_2), or (E_1+E_2), where E_1 and E_2 are Boolean expressions.
- A Boolean expression represents a Boolean function.
  - Furthermore, *every* Boolean function (of a given degree) can be represented by a Boolean expression.

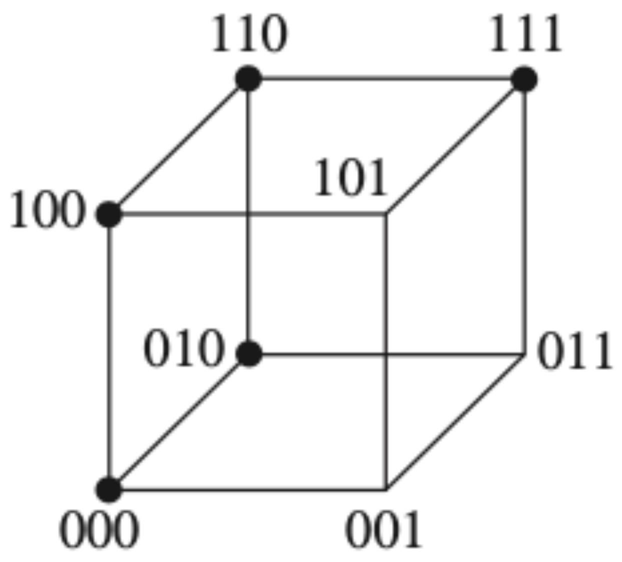
Boolean equivalents

UNIVERSITAS GADJAH MADA

- Two Boolean expressions e_1 and e_2 that represent the exact *same* function *f* are called *equivalent*. We write $e_1 \Leftrightarrow e_2$, or just $e_1 = e_2$.
  - Implicitly, the two expressions have the same value for *all* values of the free variables appearing in e_1 and e_2.
  - E.g., Boolean expressions xy, xy+0, xy · 1

Hypercube Representation

- A Boolean function of degree *n* can be represented by an *n*-cube (hypercube) with the corresponding function value at each vertex.



Boolean Identities

TABLE 5 **Boolean Identities.**

Identity	*Name*
$\overline{\overline{x}} = x$	Law of the double complement
$x + x = x$ $x \cdot x = x$	Idempotent laws
$x + 0 = x$ $x \cdot 1 = x$	Identity laws
$x + 1 = 1$ $x \cdot 0 = 0$	Domination laws
$x + y = y + x$ $xy = yx$	Commutative laws
$x + (y + z) = (x + y) + z$ $x(yz) = (xy)z$	Associative laws
$x + yz = (x + y)(x + z)$ $x(y + z) = xy + xz$	Distributive laws
$\overline{(xy)} = \overline{x} + \overline{y}$ $\overline{(x + y)} = \overline{x}\,\overline{y}$	De Morgan's laws
$x + xy = x$ $x(x + y) = x$	Absorption laws
$x + \overline{x} = 1$	Unit property
$x\overline{x} = 0$	Zero property

Example 10

- Prove the absorption law x(x+y) = x using other identities of Boolean algebra.

$x(x+y)$	$= (x+0)(x+y)$	Identity law for the Boolean sum
	$= x + 0 \cdot y$	Distributive law of the Boolean sum over the Boolean product
	$= x + y \cdot 0$	Commutative law for the Boolean product
	$= x + 0$	Domination law for the Boolean product
	$= x$	Identity law for the Boolean sum.

Duality

UNIVERSITAS GADJAH MADA

- The *dual* e^d of a Boolean expression e representing function f is obtained by exchanging $+$ with $\cdot$, and 0 with 1 in e.
  - The function represented by e^d is denoted f^d.

Find the duals of $x(y + 0)$ and $\overline{x} \cdot 1 + (\overline{y} + z)$.

Solution: Interchanging $\cdot$ signs and $+$ signs and interchanging 0s and 1s in these expressions produces their duals. The duals are $x + (y \cdot 1)$ and $(\overline{x} + 0)(\overline{y}z)$, respectively. ◀

The Abstract Definition of a Boolean Algebra

A *Boolean algebra* is a set B with two binary operations $\vee$ and $\wedge$, elements 0 and 1, and a unary operation $^{-}$ such that these properties hold for all x, y, and z in B:

$$\left.\begin{aligned} x \vee 0 &= x \\ x \wedge 1 &= x \end{aligned}\right\} \quad \text{Identity laws}$$

$$\left.\begin{aligned} x \vee \overline{x} &= 1 \\ x \wedge \overline{x} &= 0 \end{aligned}\right\} \quad \text{Complement laws}$$

$$\left.\begin{aligned} (x \vee y) \vee z &= x \vee (y \vee z) \\ (x \wedge y) \wedge z &= x \wedge (y \wedge z) \end{aligned}\right\} \quad \text{Associative laws}$$

$$\left.\begin{aligned} x \vee y &= y \vee x \\ x \wedge y &= y \wedge x \end{aligned}\right\} \quad \text{Commutative laws}$$

$$\left.\begin{aligned} x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z) \\ x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) \end{aligned}\right\} \quad \text{Distributive laws}$$

Tentukan ekspresi Boolean dari tabel kebenaran berikut ini dan sederhanakan!

x	y	z	$F(x, y, z)$
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	0
0	0	0	1

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

Dengan identitas diatas fungsi tersebut dapat kita sederhanakan menjadi

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

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

$= xyz + \bar{z}$

Dengan *absorption law*

$= xyz + (\bar{z} + \bar{z}xy)$

$= xyz + xy\bar{z} + \bar{z}$

$= xy(z + \bar{z}) + \bar{z}$

$= xy + \bar{z}$

Tentukan ekspresi Boolean dari tabel kebenaran berikut ini dan sederhanakan!

x	y	z	F(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

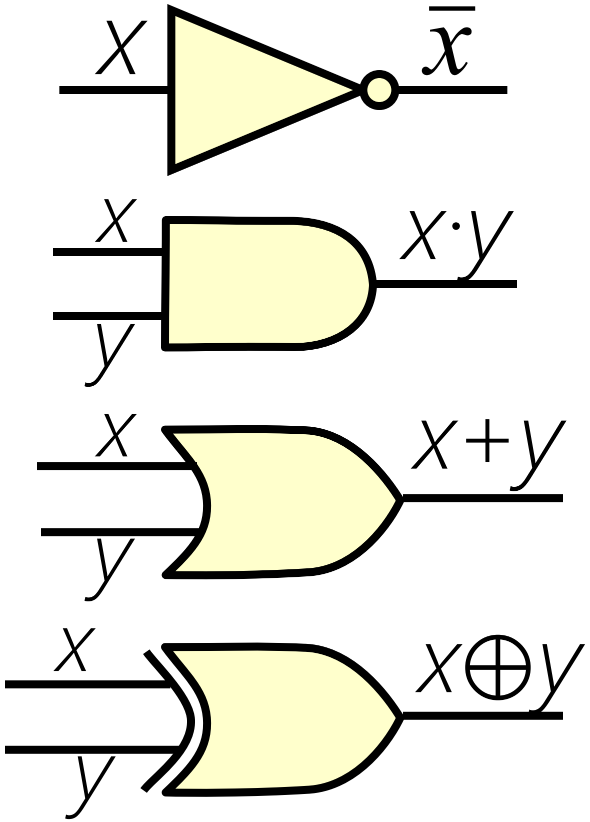
UNIVERSITAS GADJAH MADA

Logic Gates

Logic Gate Symbols

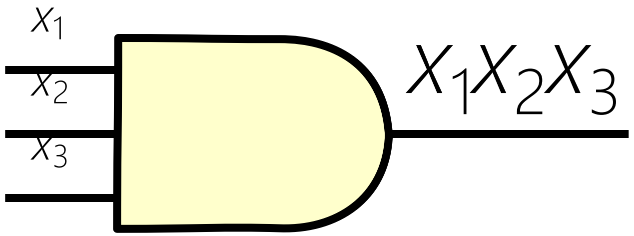
UNIVERSITAS GADJAH MADA

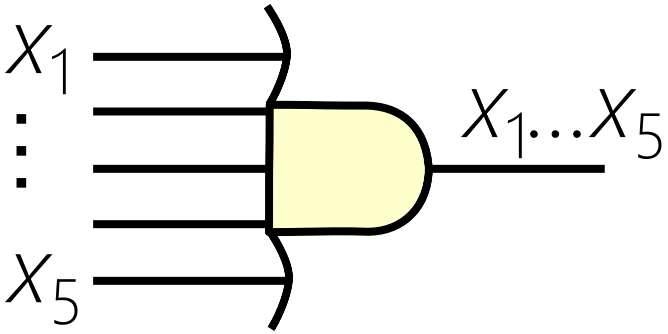
- Inverter (logical NOT, Boolean complement).
- AND gate (Boolean product).
- OR gate (Boolean sum).
- XOR gate (exclusive-OR, sum mod 2).



Multi-input AND, OR, XOR

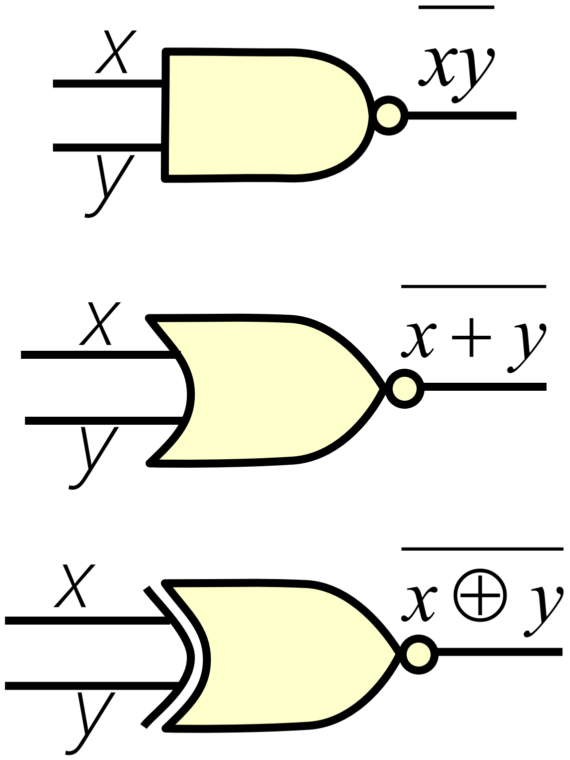
- Can extend these gates to arbitrarily many inputs.
- Two commonly seen drawing styles:
  - Note that the second style keeps the gate icon relatively small.



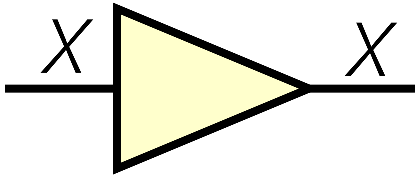


NAND, NOR, XNOR

- Just like the earlier icons, but with a small circle on the gate's output.
  - Denotes that output is complemented.
- The circles can also be placed on inputs.
  - Means, input is complemented before being used.



Buffer



- What about an inverter symbol *without* a circle?
- This is called a *buffer*. It is the identity function.
- It serves no logical purpose, but…
- It represents an explicit delay in the circuit.
  - This is sometimes useful for timing purposes.
- All gates, when physically implemented, incur a non-zero delay between when their inputs are seen and when their outputs are ready.

- Gates: The basic elements of electronic circuits
- Combinational circuits: the circuits whose output depends only on the input and not on the current state of the circuit (no memory).

Karnaugh Maps

- K-map is a graphical method for finding terms to combine for Boolean functions involving a relatively small number of variables.

$$x\bar{y} + \bar{x}y + \bar{x}\,\bar{y}.$$

	y	$\bar{y}$
x		1
$\bar{x}$	1	1

$$\bar{x} + \bar{y}.$$

$$xy\bar{z} + x\bar{y}\,\bar{z} + \bar{x}yz + \bar{x}\,\bar{y}\,\bar{z}$$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x		1	1	
$\bar{x}$	1		1	

$$wx\bar{y}\,\bar{z} + w\bar{x}yz + w\bar{x}y\bar{z} + w\bar{x}\,\bar{y}\,\bar{z} + \bar{w}x\bar{y}\,\bar{z} + \bar{w}\,\bar{x}y\bar{z} + \bar{w}\,\bar{x}\,\bar{y}\,\bar{z}$$

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx			1	
$w\bar{x}$	1	1	1	
$\bar{w}\bar{x}$		1	1	
$\bar{w}x$			1	

Don’t Care Conditions

- When we care only about the output for some combinations of input values, because other combinations of input values are not possible or never occur.
    - Those combinations that never occur can be arbitrarily chosen.