Boolean Algebra and Logic Gates

Logic Design of Digital Systems (300-1209) section 1

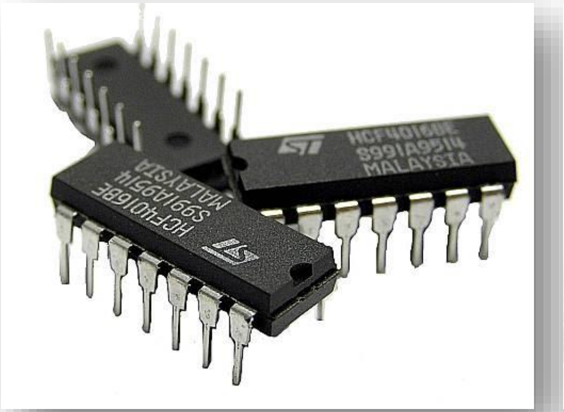
LECTURE 03

KRISADA PHROMSUTHIRAK

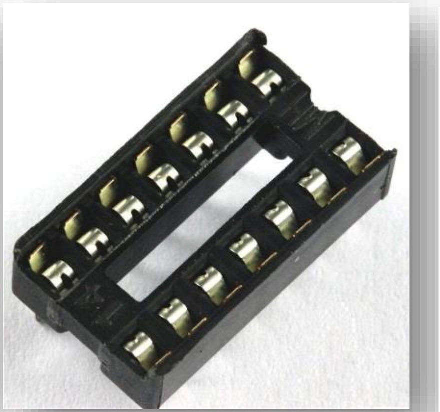
Digital Logic Gates

IC (Integrated Circuit)

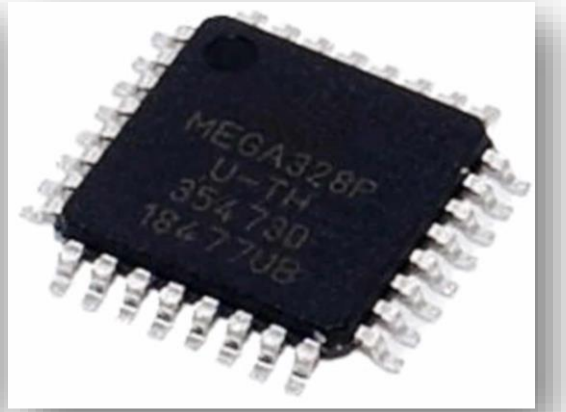
Digital circuits are invariably constructed with integrated circuits. An integrated circuit (IC) is a **small silicon semiconductor** crystal, called a chip, containing electrical components such as transistors, diodes, resistors, and capacitors.



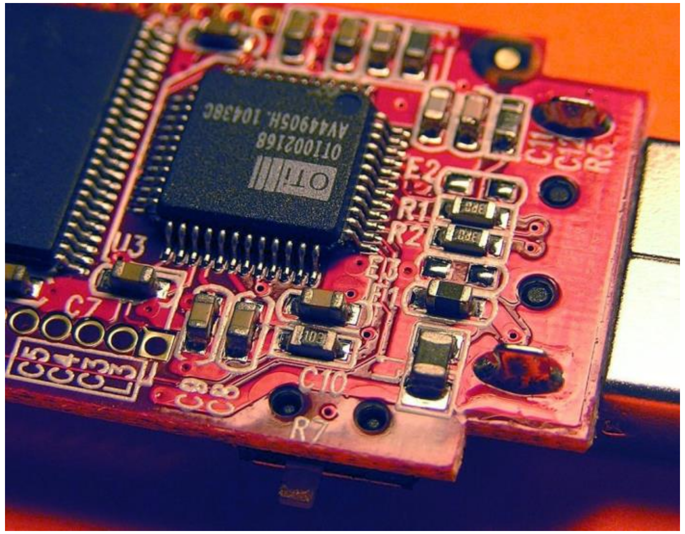
IC



IC Socket



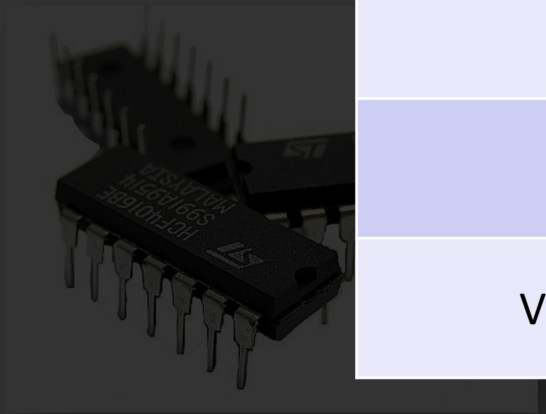
Surface Mount IC



IC (Integrated Circuit)

Digital circuits ar… **…all silicon semiconductor** c… , resistors, and capacitors.

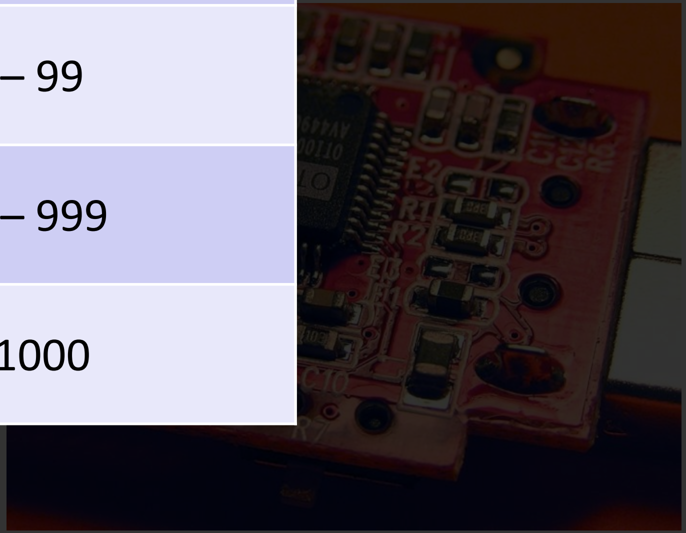
Levels	Number of Gates
Small Scale Integration (SSI)	<= 12
Medium Scale Integration (MSI)	13 – 99
Large Scale Integration (LSI)	100 – 999
Very Large-Scale Integration (VLSI)	>= 1000



IC

IC Socket

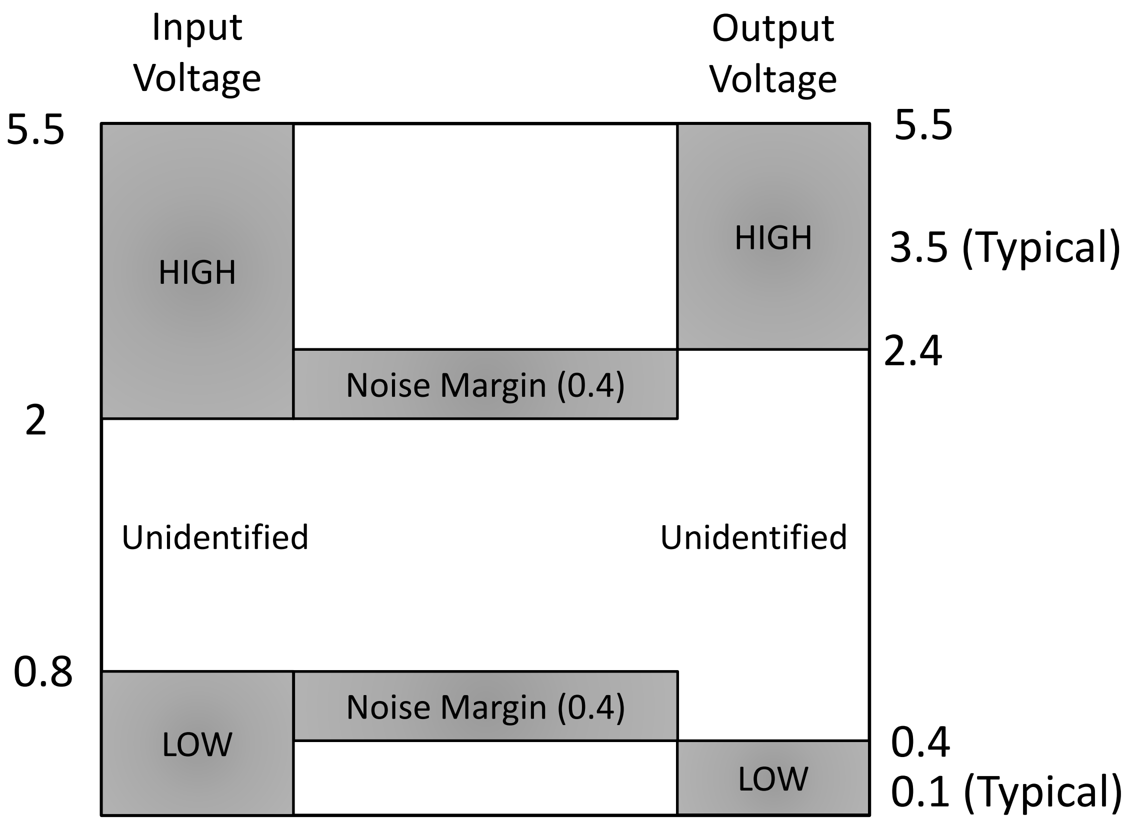
Surface Mount IC



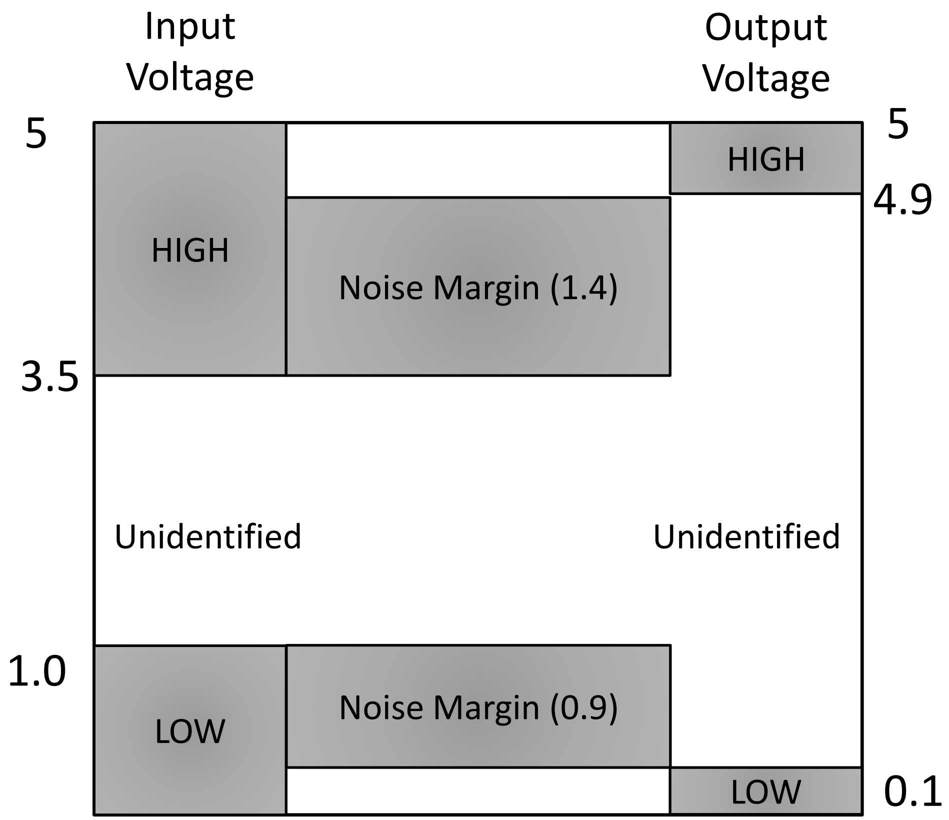
IC (Integrated Circuit)

TTL Transistor-transistor logic

CMOS Complementary metal-oxide semiconductor



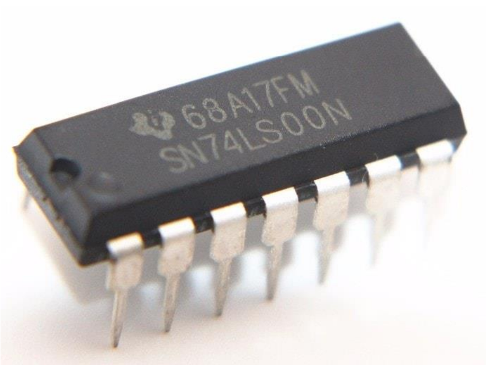
TTL

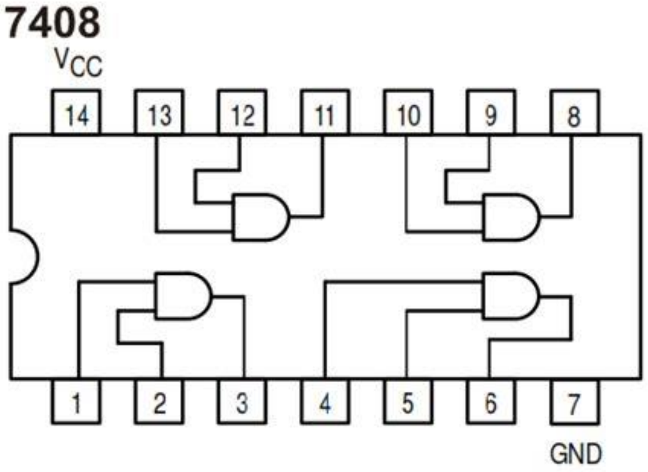


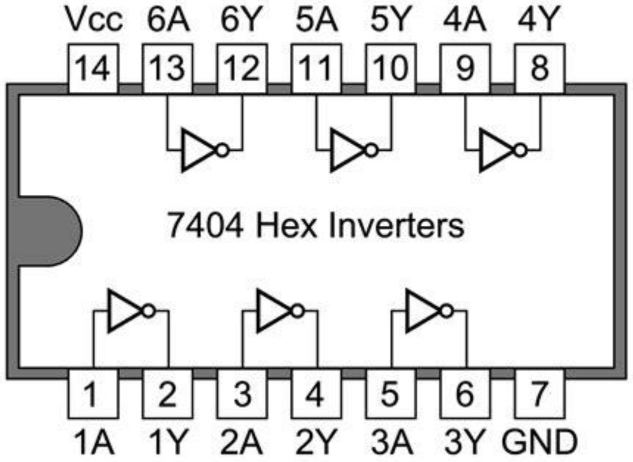
CMOS

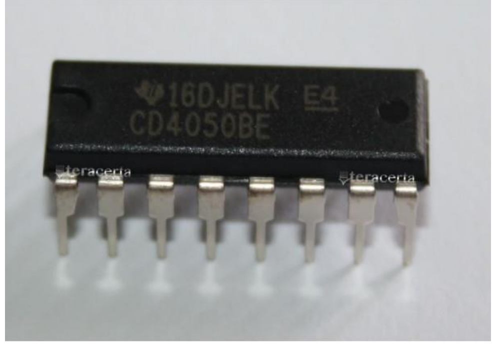
IC (Integrated Circuit)

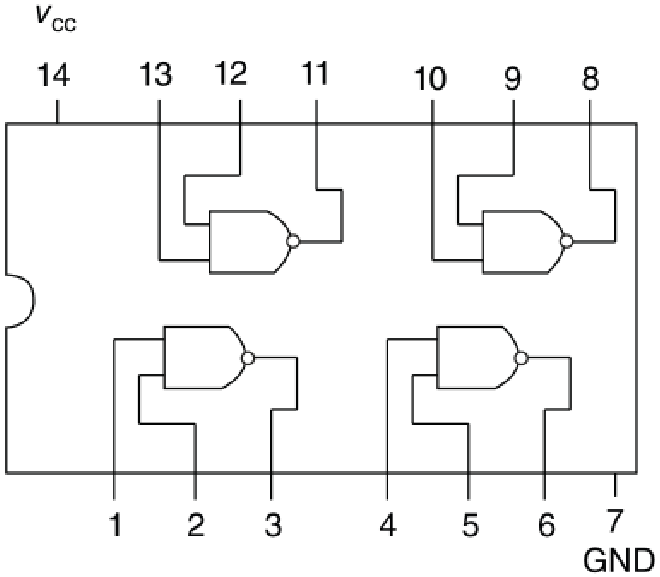
	TTL	CMOS
Propagation Delays	Around 10 ns	Between 20 and 50 ns
Electromagnetic Disruptions	Less susceptible to electromagnetic disruptions	More sensitive to electromagnetic disruptions
Basic Logic Gates	It contains only NAND gates	Feature NAND and NOR gates to carry out logic function



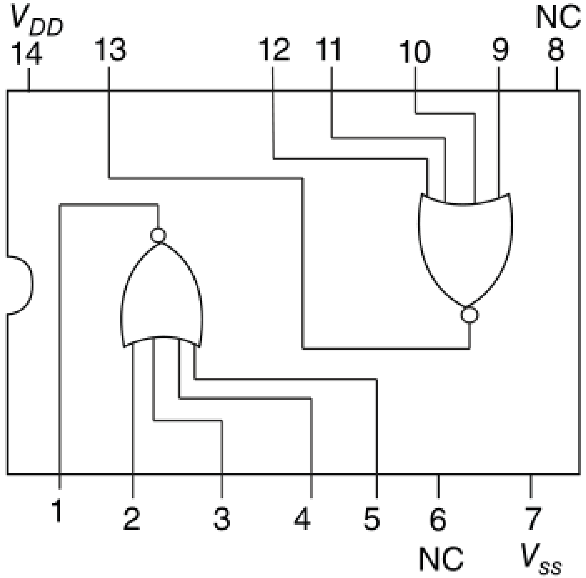




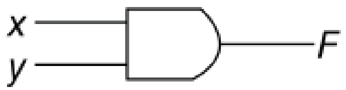
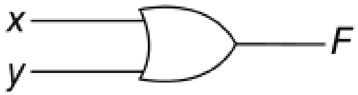
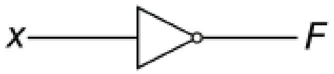
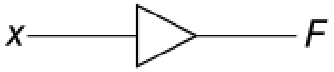
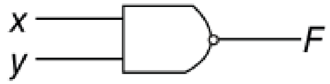
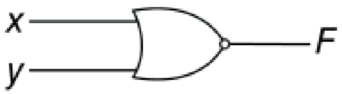
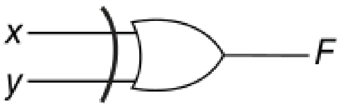
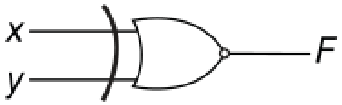




7400–Quadruple 2-input NAND gates

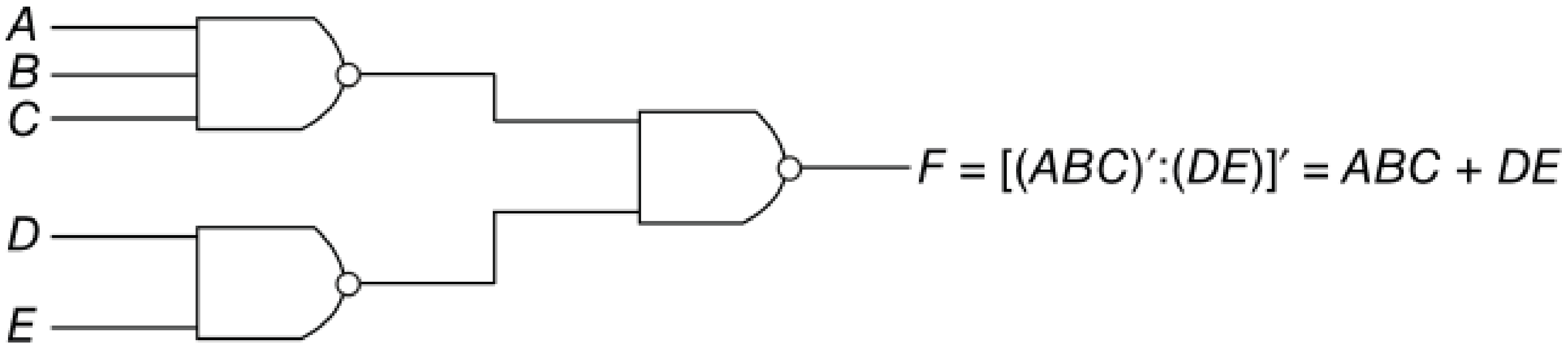


4002–Dual 4-input NOR gates.

Name	Graphic symbol	Algebraic function	x	y	F
AND		$F = xy$	0	0	0
			0	1	0
			1	0	0
			1	1	1
OR		$F = x + y$	0	0	0
			0	1	1
			1	0	1
			1	1	1
Inverter		$F = x'$	0		1
			1		0
Buffer		$F = x$	0		0
			1		1
NAND		$F = (xy)'$	0	0	1
			0	1	1
			1	0	1
			1	1	0
NOR		$F = (x + y)'$	0	0	1
			0	1	0
			1	0	0
			1	1	0
Exclusive-OR (XOR)		$F = xy' + x'y = x \oplus y$	0	0	0
			0	1	1
			1	0	1
			1	1	0
Exclusive-NOR or equivalence		$F = xy + x'y' = x \odot y$	0	0	1
			0	1	0
			1	0	0
			1	1	1

Example

$$F = [(ABC)'(DE)']' = ABC + DE$$

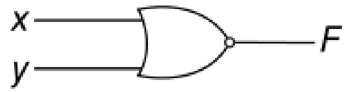


Cascaded NAND gate

Example: Construct AND gate[2] using NOR gate[2]

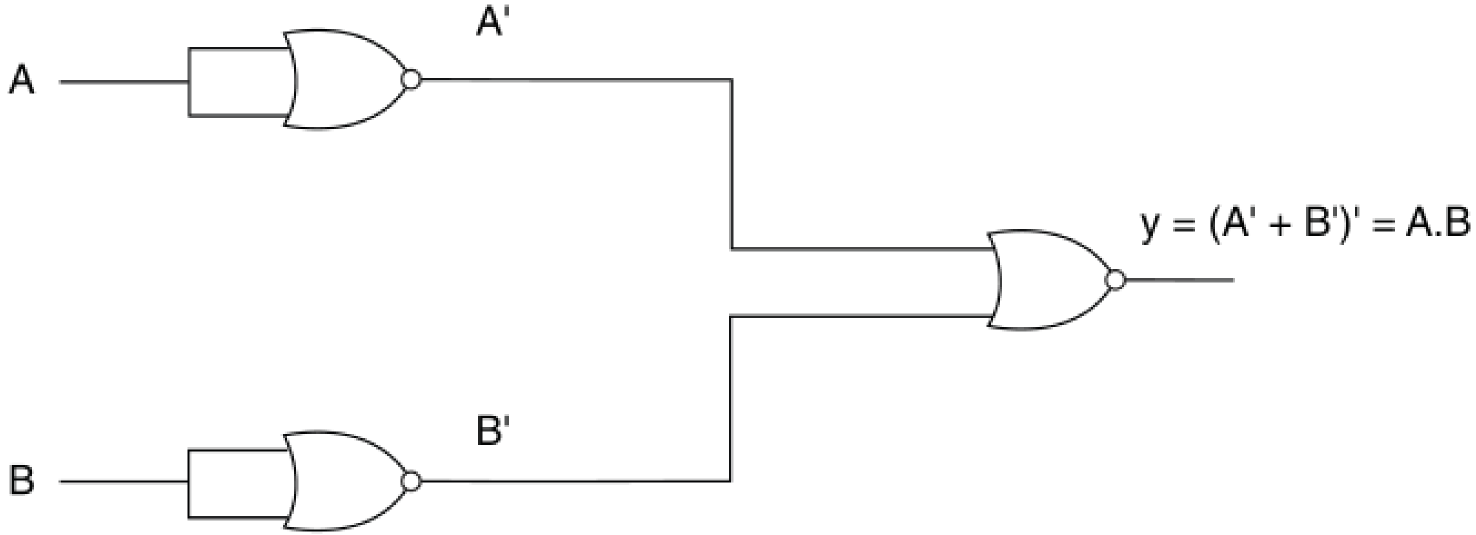
$$A \cdot B = \left((A \cdot B)'\right)' = (A' + B')'$$

NOR



$F = (x + y)'$

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

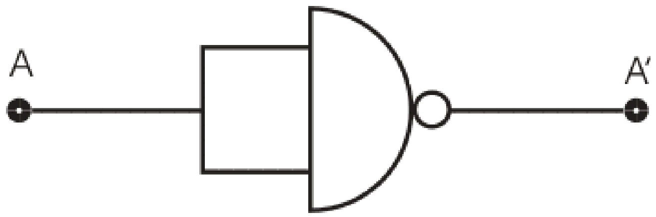


De Morgan's theorem

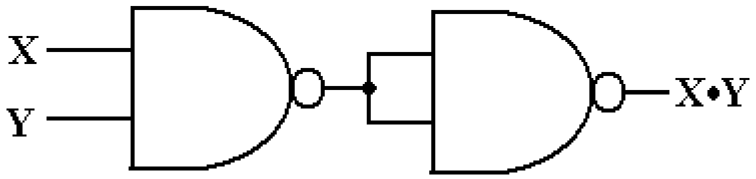
Logic.ly

#1 Logic Gate 101

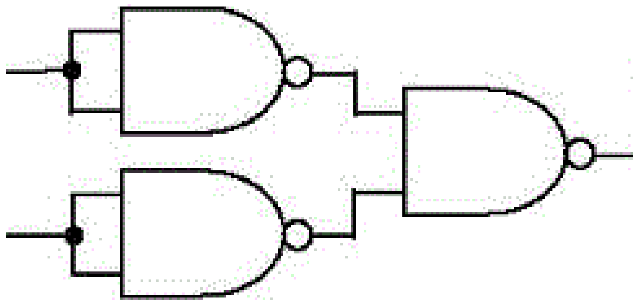
#2 Universal Gate (NAND)



NOT Gate (Inverter)

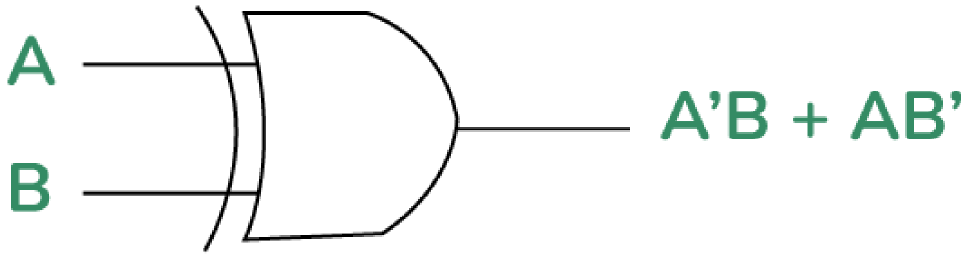


AND Gate



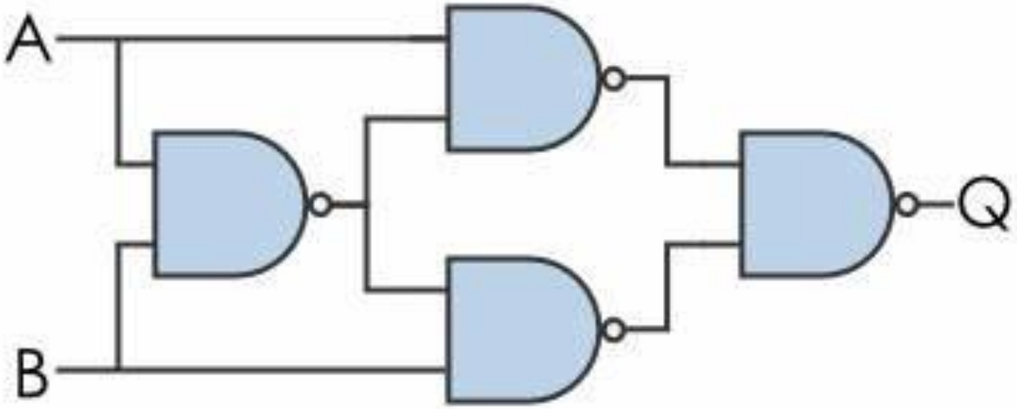
OR Gate

XOR Gate (Exclusive-OR Gate)



Truth Table

A (Input 1)	B (Input 2)	X = A'B + AB'
0	0	0
0	1	1
1	0	1
1	1	0



XOR Gate

Boolean Algebra

Boolean algebra is the category of algebra in which the variable's values are the truth values, **true** and **false**, ordinarily denoted 1 and 0 respectively. It is used to analyze and simplify digital circuits or digital gates.

Like other deductive math systems, it starts with defining the set elements, set operators.

If S is a set and x, y is a certain objects.

$x \in S$ → x is a member of S

$y \notin S$ → y is not an element of S

Boolean Algebra

The postulates of a mathematical system form the basic assumptions from which it is possible to deduce the rules, theorems, and properties of the system.

The most common postulates used to formulate various algebraic structures are

[1] Closure

[2] Associative law

[3] Commutative law

Boolean Algebra

The postulates of a mathematical system form the basic assumptions from which it is possible to deduce the rules, theorems, and properties of the system.

The most common postulates used to formulate various algebraic structures are

[4] Identity element

Boolean Algebra

The postulates of a mathematical system form the basic assumptions from which it is possible to deduce the rules, theorems, and properties of the system.

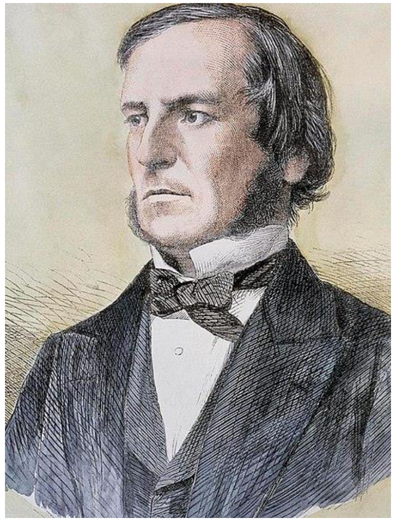
The most common postulates used to formulate various algebraic structures are

[5] Inverse

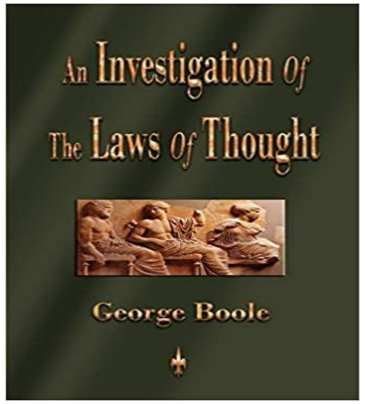
[6] Distributive law

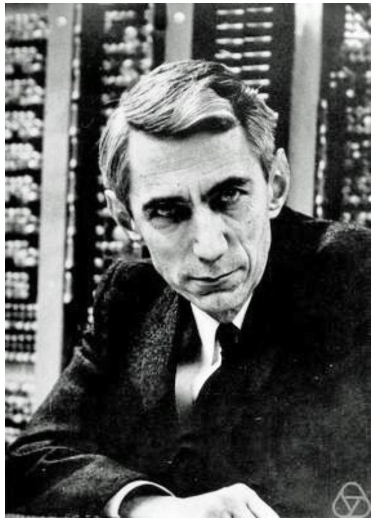
Axiomatic Definition of Boolean Algebra

In 1854 *George Boole* introduced a systematic treatment of logic and developed for this purpose an algebraic system now called **Boolean algebra.** In 1938 *C. E. Shannon* introduced a two-valued Boolean algebra called switching algebra, in which he demonstrated that the properties of bistable electrical switching circuits can be represented by this algebra. For the formal definition of Boolean algebra, we shall employ the postulates formulated by *E. V. Huntington* in 1904.

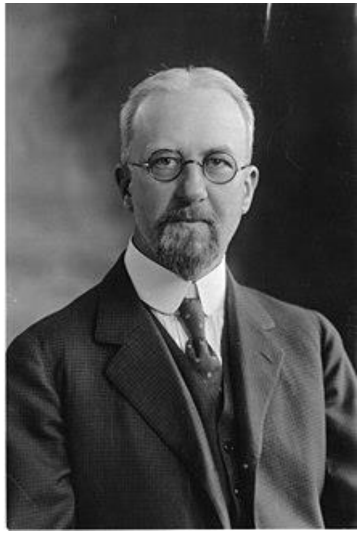


George Boole





Claude Elwood Shannon



Edward Vermilye Huntington

Axiomatic Definition of Boolean Algebra

Huntington postulates

1	(a) **Closure** with respect to the operator $+$ (b) **Closure** with respect to the operator $\cdot$
2	(a) An **identity element** with respect to $+$, designated by 0: $x + 0 = 0 + x = x$ (b) An **identity element** with respect to $\cdot$, designated by 1: $x \cdot 1 = 1 \cdot x = x$
3	(a) **Commutative** with respect to the operator $+$: $x + y = y + x$ (b) **Commutative** with respect to the operator $\cdot$: $x \cdot y = y \cdot x$

Axiomatic Definition of Boolean Algebra

Huntington postulates

4	(a) $\cdot$ is **distributive** over $+$: $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$ (b) $+$ is **distributive** over $\cdot$: $x + (y \cdot z) = (x + y) \cdot (x + z)$
5	For every element $x \in B$, there exists an element $\acute{x} \in B$ (called the **complement** of x) that (a) $x + \acute{x} = 1$ (b) $x \cdot \acute{x} = 0$
6	There exists at least two elements $x, y \in B$ such that $x \neq y$

Axiomatic Definition of Boolean Algebra

Huntington postulates **4 ข้อที่ถูกใช้งานบ่อย**

Postulate 2: Identity	(a) $x + 0 = x$	(b) $x \cdot 1 = x$
Postulate 3: Commutative	(a) $x + y = y + x$	(b) $xy = yx$
Postulate 4: Distributive	(a) $x(y + z) = xy + xz$	(b) $x + yz = (x + y)(x + z)$
Postulate 5: Complement	(a) $x + \acute{x} = 1$	(b) $x \cdot \acute{x} = 0$

Basic Theorems and Properties of Boolean Algebra

Basic Theorems

Theorem 1	(a) $x + x = x$	(b) $x \cdot x = x$
Theorem 2	(a) $x + 1 = 1$	(b) $x \cdot 0 = 0$
Theorem 3: Involution	$(x')' = x$	
Theorem 4: Associative	(a) $x + (y + z) = (x + y) + z$	(b) $x(yz) = (xy)z$
Theorem 5: DeMorgan's	(a) $(x + y)' = x'y'$	(b) $(xy)' = x' + y'$
Theorem 6: Absorption	(a) $x + xy = x$	(b) $x(x + y) = x$

Basic Theorems and Properties of Boolean Algebra

Basic Theorems **Theorem 1 (a):** $x + x = x$

Basic Theorems and Properties of Boolean Algebra

Basic Theorems **Theorem 1 (b):** $x \cdot x = x$

Basic Theorems and Properties of Boolean Algebra

Basic Theorems **Theorem 2 (a):** $x + 1 = 1$

Basic Theorems and Properties of Boolean Algebra

Basic Theorems **Theorem 2 (b):** $x \cdot 0 = 0$

Basic Theorems and Properties of Boolean Algebra

Basic Theorems **Theorem 3:** $(x')' = x$

Basic Theorems and Properties of Boolean Algebra

Basic Theorems **Theorem 5 (DeMorgan's):**

(a) $(x+y)' = x'y'$

(b) $(xy)' = x' + y'$

x	y	$x+y$	$(x+y)'$	x'	y'	$x'y'$
0	0	0	1	1	1	[
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Basic Theorems and Properties of Boolean Algebra

Basic Theorems **Theorem 6 (a):** $x + xy = x$

Basic Theorems and Properties of Boolean Algebra

Basic Theorems **Theorem 6 (b):** $x \cdot (x + y) = x$

Basic Theorems and Properties of Boolean Algebra

Huntington postulates & Basic Theorems

(Only frequently used topics)

Postulate 2: Identity	(a) $x + 0 = x$	(b) $x \cdot 1 = x$
Postulate 3: Commutative	(a) $x + y = y + x$	(b) $xy = yx$
Postulate 4: Distributive	(a) $x(y + z) = xy + xz$	(b) $x + yz = (x + y)(x + z)$
Postulate 5: Complement	(a) $x + x\' = 1$	(b) $x \cdot x\' = 0$

Theorem 1:	(a) $x + x = x$	(b) $x \cdot x = x$
Theorem 2:	(a) $x + 1 = 1$	(b) $x \cdot 0 = 0$
Theorem 3: Involution	$(x')' = x$	
Theorem 5: DeMorgan's	(a) $(x + y)' = x'y'$	(b) $(xy)' = x' + y'$
Theorem 6: Absorption	(a) $x + xy = x$	(b) $x(x + y) = x$

Operator Precedence (ลำดับความสำคัญของตัวดำเนินการ)

The operator precedence for evaluating Boolean expressions is (1) parentheses, (2) NOT, (3) AND, and (4) OR.

Basic Theorems and Properties of Boolean Algebra

[1] Using basic Boolean theorem prove: $(x + y)(x + z) = x + yz$

Basic Theorems and Properties of Boolean Algebra

[2] Using basic Boolean theorem prove: $xy + xz + yz' = xz + yz'$

Basic Theorems and Properties of Boolean Algebra

Venn Diagram

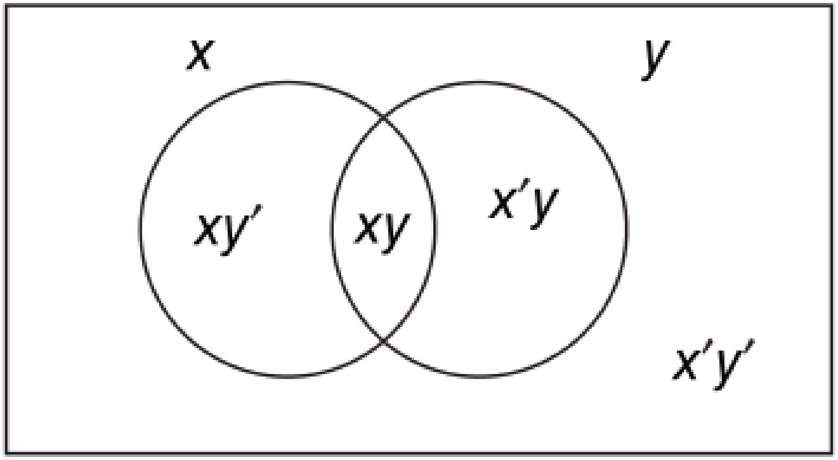


Figure 2.1 Venn diagram for two variables

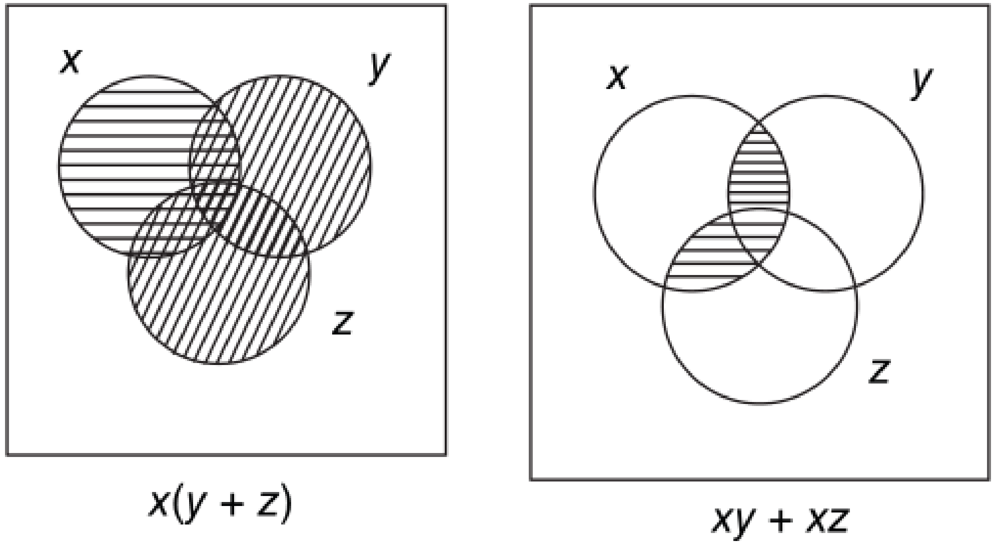


Figure 2.3 Venn diagram illustration of the distributive law

Postulate 4: Distributive $x(y+z) = xy + xz$

Boolean Functions

SECTION BREAK

Boolean Functions

A Boolean function is an expression formed with binary variables, the two binary operators OR and AND, the unary operator NOT, parentheses, and equal sign.

$$F_1 = xy\acute{z} \qquad F_3 = \acute{x}\acute{y}z + \acute{x}yz + x\acute{y}$$

$$F_2 = x + \acute{y}z \qquad F_4 = x\acute{y} + \acute{x}z$$

x	y	z	F_1	F_2	F_3	F_4
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	1	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	0

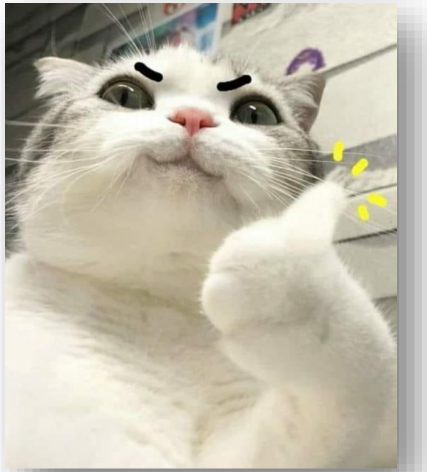
Boolean Functions

Complement of a Function

The complement of a function F is F′ and is obtained from an interchange of 0's for 1's and 1's for 0's in the value of F. The complement of a function may be derived algebraically through De Morgan's theorem.

De Morgan's theorems can be extended to three or more variables. The three-variable form of the first De Morgan's theorem is derived below.

$$
\begin{aligned}
(A + B + C)' &= (A + X)' && \text{let } B + C = X \\
&= A'X' && \text{by theorem 5(a) (De Morgan)} \\
&= A' \bullet (B + C)' && \text{substitute } B + C = X \\
&= A' \bullet (B'C') && \text{by theorem 5(a) (De Morgan)} \\
&= A'B'C' && \text{by theorem 4(b) (associative)}
\end{aligned}
$$



Boolean Functions

Complement of a Function

Example 1: Find complement of Function $F_1 = x'yz' + x'y'z$

Boolean Functions

Complement of a Function

<u>Example</u> 2: Find complement of Function $F_2 = x(y'z' + yz)$

Canonical and Standard Forms

Minterms and Maxterms

A binary variable may appear in either form:

x : Normal form

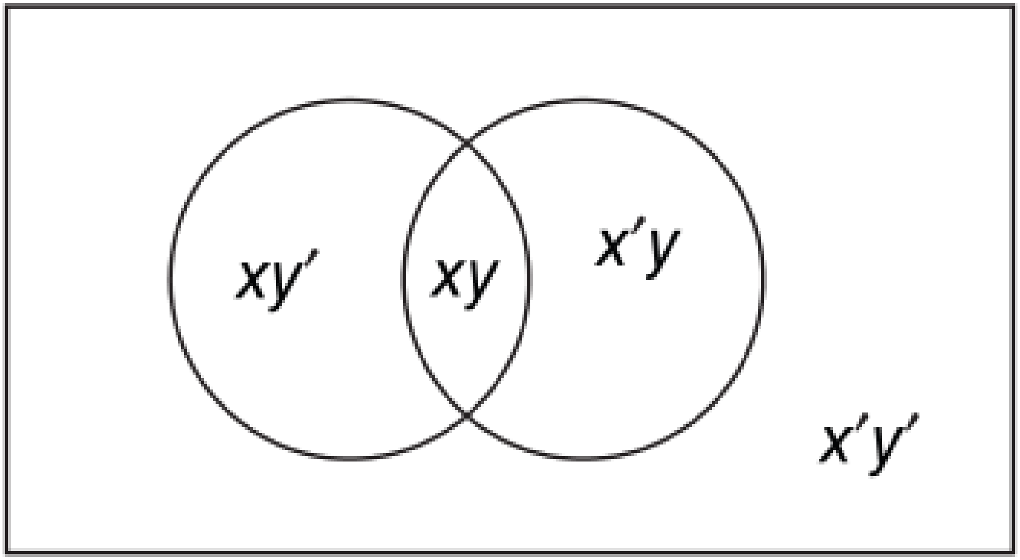
$\acute{x}$: Complement form

Suppose we have two variables that are **being combine** with **AND operator**.
There are four possible outcomes.

$$xy\ ,\ x\acute{y}\ ,\ \acute{x}y\ ,\ \acute{x}\acute{y}$$

Each of these four AND terms represents one of the distinct areas in the Venn diagram

So, it called **minterm** or **standard product**



Canonical and Standard Forms

Minterms and Maxterms

In a similar fashion, if we change the **operation to OR**. We called it **maxterm** or **standard sum**

Example of minterm and maxterm for three binary variables

			Minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x+y+z$	M_0
0	0	1	$x'y'z$	m_1	$x+y+z'$	M_1
0	1	0	$x'yz'$	m_2	$x+y'+z$	M_2
0	1	1	$x'yz$	m_3	$x+y'+z'$	M_3
1	0	0	$xy'z'$	m_4	$x'+y+z$	M_4
1	0	1	$xy'z$	m_5	$x'+y+z'$	M_5
1	1	0	xyz'	m_6	$x'+y'+z$	M_6
1	1	1	xyz	m_7	$x'+y'+z'$	M_7

Canonical and Standard Forms

Minterms and Maxterms

Any Boolean function can be expressed as a sum of minterms
(by “sum” is meant the ORing of terms).

Any Boolean function can be expressed as a product of maxterms
(by “product” is meant the ANDing of terms).

Canonical Form

Canonical and Standard Forms

ขั้นตอนต่อไปนี้จะเริ่มยากแล้วนะ

Minterms and Maxterms

x	y	z	Function f_1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	I	0	0
1	1	1	1

Minterms

$$f_1 = x'y'z + xy'z' + xyz = m_1 + m_4 + m_7$$

Maxterms

$$f_1 = (x + y + z)(x + y' + z)(x + y' + z')(x' + y + z')(x' + y' + z)$$
$$= M_0 \cdot M_2 \cdot M_3 \cdot M_5 \cdot M_6$$

Canonical and Standard Forms

Step
1

Sum of Minterms (Canonical Forms)

Example: Express the Boolean function $F = A + B'C'$ in a sum of minterms

Canonical and Standard Forms

Step
2

Sum of Minterms (Canonical Forms)

Example: Express the Boolean function $F = A + B'C'$ in a sum of minterms

Canonical and Standard Forms

$$A + BC = (A + B)(A + C)$$

Step
1

Product of Maxterms (Canonical Forms)

Example: Express the Boolean function $F = xy + x'z$ in a product of maxterms

Canonical and Standard Forms

$$A + BC = (A + B)(A + C)$$

Step 2

Product of Maxterms (Canonical Forms)

Example: Express the Boolean function $F = xy + x'z$ in a product of maxterms

Canonical and Standard Forms

$$A + BC = (A + B)(A + C)$$

Step 3

Product of Maxterms (Canonical Forms)

Example: Express the Boolean function $F = xy + x'z$ in a product of maxterms

Canonical and Standard Forms

Standard Forms

The two canonical forms of Boolean algebra are basic forms that one obtains from reading a function from the truth table. These forms are very seldom the ones with the least number of literals, because each minterm or maxterm must contain, by definition, all the variables either complemented or uncomplemented.

Another way to express Boolean functions is in standard form. In this configuration, the terms that form the function may contain one, two or any number of literals. There are two types of standard forms:

Sum of Product: $F_1 = \acute{y} + xy + \acute{x}y\acute{z}$

Product of Sum: $F_2 = x(\acute{y} + z)(\acute{x} + y + \acute{z})$

Canonical and Standard Forms

Standard Forms

Boolean function may not be in the standard form, but we can change it by postulates and theorems.

<u>Example</u>

$$F_3 = (AB + CD)\left(\acute{A}\acute{B} + \acute{C}\acute{D}\right)$$

We can change into…

$$F_3 = \acute{A}\acute{B}CD + AB\acute{C}\acute{D}$$

Canonical and Standard Forms

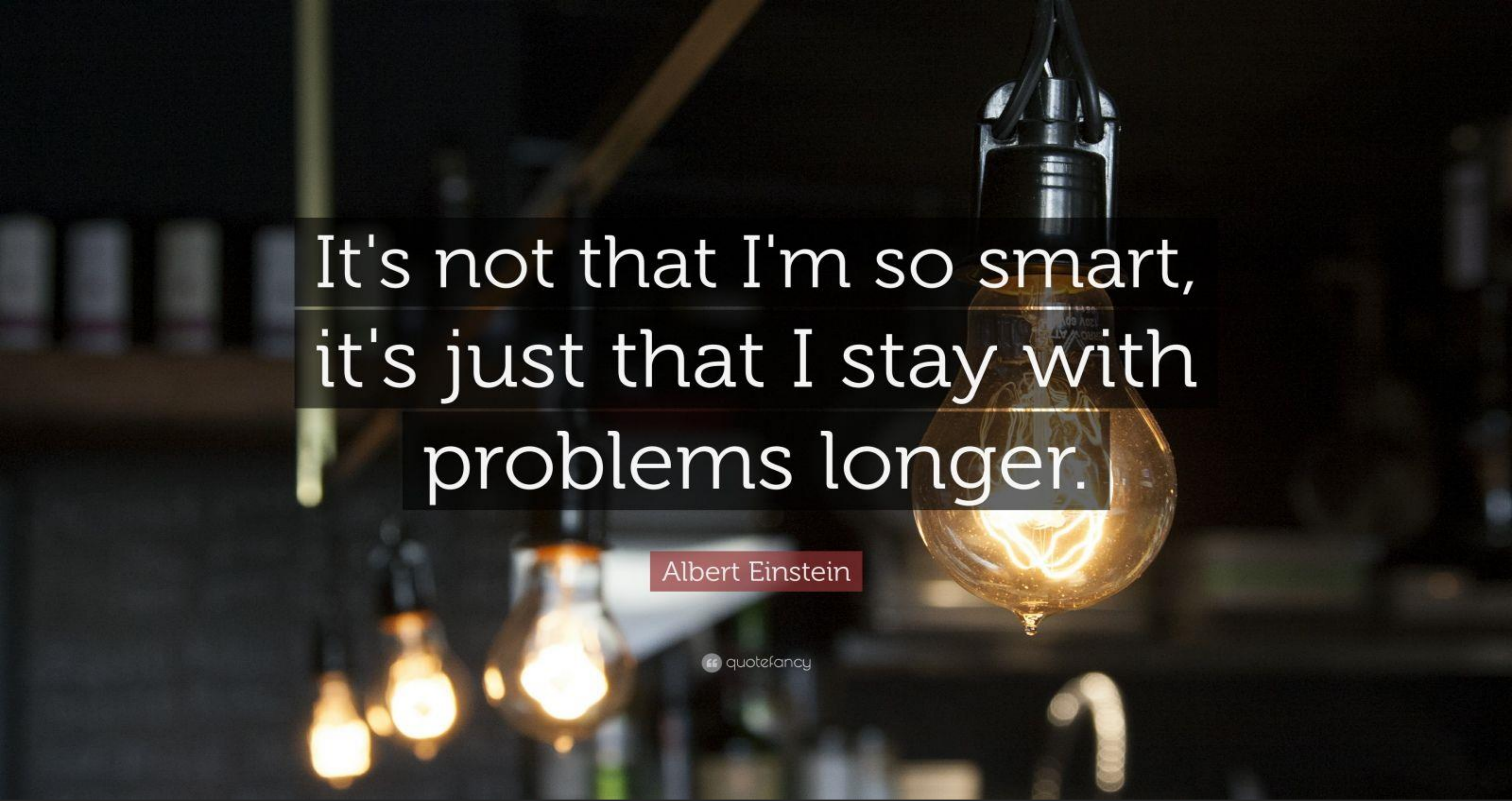
Other Logic Operator

x	y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Operator Symbol				$/$		$/$		$\oplus$	$+$	$\downarrow$	$\odot$		$\subset$		$\supset$	$\uparrow$	

All possible outcomes of two variables

Boolean functions	Operator symbol	Name
$F_0 = 0$		Null
$F_1 = xy$	$x \cdot y$	AND
$F_2 = xy'$	x/y	Inhibition
$F_3 = x$		Transfer
$F_4 = x'y$	y/x	Inhibition
$F_5 = y$		Transfer
$F_6 = xy' + x'y$	$x \oplus y$	Exciusive-OR
$F_7 = x + y$	$x + y$	OR
$F_8 = (x + y)'$	$x \downarrow y$	NOR
$F_9 = xy + x'y'$	$x \odot y$	Equivalence*
$F_{10} = y'$	y'	Complement
$F_{11} = x + y'$	$x \subset y$	Implication
$F_{12} = x'$	x'	Complement
$F_{13} = x' + y$	$x \supset y$	Implication
$F_{14} = (xy)'$	$x \uparrow y$	NAND
$F_{15} = 1$		Identity

**Equivalence* is also known as *equality*, *coincidence*, and *exclusive-NO*


It's not that I'm so smart,
it's just that I stay with
problems longer.
Albert Einstein
quotefancy