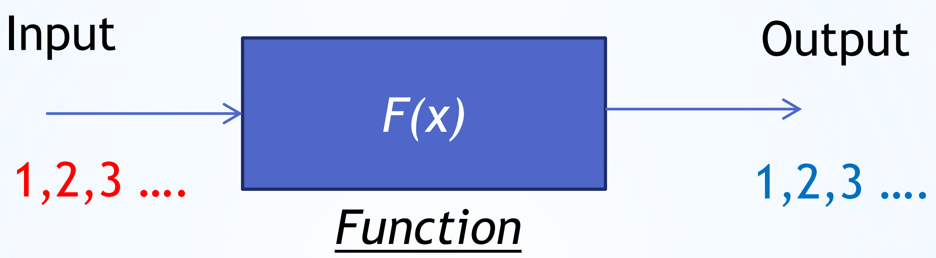
Logic Gates and Boolean Functions

- 3.1 Basic logic gates
- 3.2 Boolean algebra
- 3.3 Logic Circuit and Boolean Expression
- 3.4 Logic function – SOP, POS, minterm, maxterm, canonical form
- 3.5 Synthesis of logic function
- 3.6 Structure using one gate type

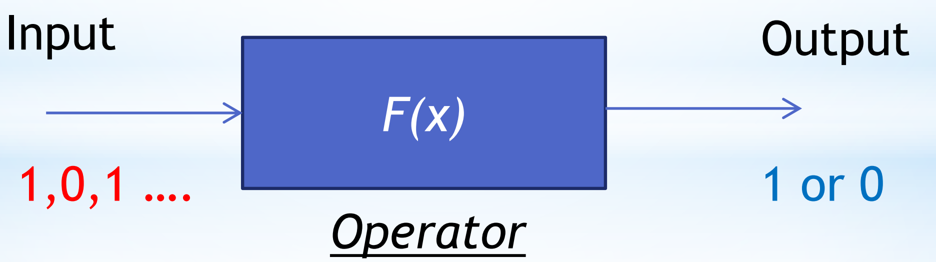
3.1 Basic Logic Gates

- The term **gate** describes a circuit that performs a basic logic operation.

Conventional numerical / mathematical function

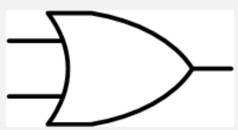
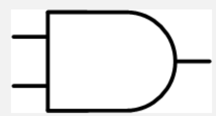
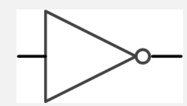


Logic operator/ function



Binary decision output e.g, Yes/No ; True/False and 1/0 .

Logic Operator

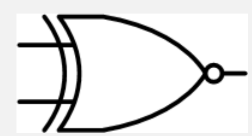
	OR gate	AND gate	NOT gate
Binary/Unary operator ?	Binary	Binary	Unary
Symbols	1: + 2: $\vee$	1: $\cdot$ 2: $\wedge$ 3: absence of an operator	1: ' 2: ~ 3: $\overline{}$
Examples	1: $a + b$ 2: $a \vee b$	1: $a \cdot b$ 2: $a \wedge b$ 3: ab	1: a' 2: $\sim a$ 3: $\bar{a}$
Logic Gate Symbol			

Truth table of OR	
a b	$a + b$
0 0	0
0 1	1
1 0	1
1 1	1

Truth table of AND	
a b	$a\,b$
0 0	0
0 1	0
1 0	0
1 1	1

Truth table of NOT	
a	$\bar{a}$
0	1
1	0

Logic Operator

Operation		NAND	NOR	XOR	XNOR
a	b	$(ab)'$	$(a + b)'$	$a \oplus b$	$\overline{a \oplus b}$
0	0	1	1	0	1
0	1	1	0	1	0
1	0	1	0	1	0
1	1	0	0	0	1
Logic Gate Symbol					

3.2 Boolean Algebra

- A set of element ***S*** with at least two different elements (***x*** , ***y***) satisfying binary operations (**+**) and (·) .
- For Boolean algebra in which ***S***= {0,1}, the formulation is referred as switching function.

Basic Postulates

If $x,y \in S$,

$$x + y = y + x$$

$$x \cdot y = y \cdot x$$

commutative

If $x,y,z \in S$,

$$x + (y + z) = (x + y) + z$$

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z$$

associative

If $x,y,z \in S$,

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

distributive

Distributive Law

- Proof

$$x + y \cdot z = (x + y) \cdot (x + z)$$

$x\ \ y\ \ z$	$y \cdot z$	$x + y \cdot z$	$x + y$	$x + z$	$(x + y) \cdot (x + z)$
0 0 0	0	**0**	0	0	**0**
0 0 1	0	**0**	0	1	**0**
0 1 0	0	**0**	1	0	**0**
0 1 1	1	**1**	1	1	**1**
1 0 0	0	**1**	1	1	**1**
1 0 1	0	**1**	1	1	**1**
1 1 0	0	**1**	1	1	**1**
1 1 1	1	**1**	1	1	**1**

Duality

- If an expression is valid in Boolean algebra, the dual of the expression is also valid.

- Principle of duality:

$$0 \cdot x = 0 \qquad 1 + x = 1$$

$$1 \cdot x = x \qquad 0 + x = x$$

$$x \cdot x = x \qquad x + x = x$$

$$x \cdot x' = 0 \qquad x + x' = 1$$

The expressions are interchangeable by replacing “0” by “1” and “+” by “ $\cdot$ ” .

Theorem

Idempotent

$x + x = x$ $\qquad$ $x \cdot x = x$

Involution

$(x')' = x$

Absorption

$x + x\,y = x$ $\qquad$ $x\,(x + y) = x$

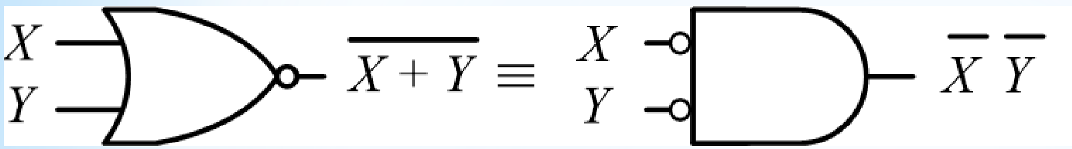
Logical adjacency

$x\,y + x\,y' = x$

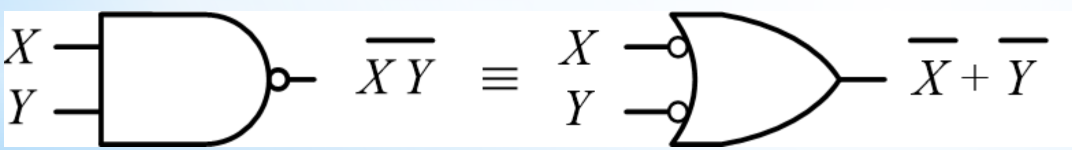
DeMorgan $\overline{(x + y)} = \overline{x}\;\overline{y}$ $\qquad$ $\overline{x\,y} = \overline{x} + \overline{y}$

- The complement of sum is equal to the product of the complement
- The complement of product is equal to the sum of the complement

DeMorgan



$X\;\;Y$	$\overline{X+Y}$	$\overline{X}\;\overline{Y}$
0 0	1	1
0 1	0	0
1 0	0	0
1 1	0	0

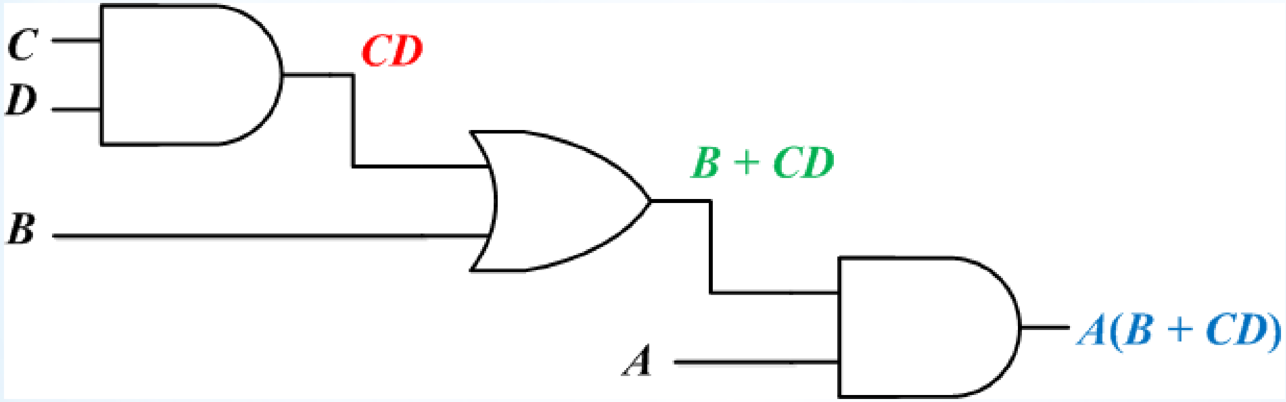


$X\;\;Y$	$\overline{X\,Y}$	$\overline{X}+\overline{Y}$
0 0	1	1
0 1	1	1
1 0	1	1
1 1	0	0

Note that $\overline{X+Y} \neq \overline{X}+\overline{Y}$ and $\overline{X\,Y} \neq \overline{X}\;\overline{Y}$, a very common mistake !!

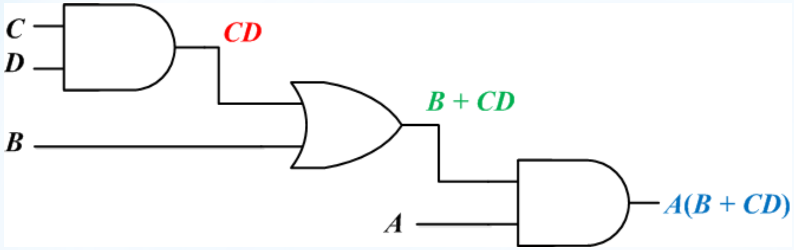
3.3 Logic Circuit and Boolean Expression

Boolean expression from a logic circuit



- Write down the output expression from all logic operators
- The Boolean function of this circuit is $A\ (B + CD)$
- Construct a truth table for above logic circuit

Truth table for a logic circuit



Examples of numbering systems		Inputs				Output
Decimal	Hexadecimal	A	B	C	D	A(B+CD)
0	0	0	0	0	0	0
1	1	0	0	0	1	0
2	2	0	0	1	0	0
3	3	0	0	1	1	0
4	4	0	1	0	0	0
5	5	0	1	0	1	0
6	6	0	1	1	0	0
7	7	0	1	1	1	0
8	8	1	0	0	0	0
9	9	1	0	0	1	0
10	A	1	0	1	0	0
11	B	1	0	1	1	1
12	C	1	1	0	0	1
13	D	1	1	0	1	1
14	E	1	1	1	0	1
15	F	1	1	1	1	1

Completed solution for a logic circuit design must include:

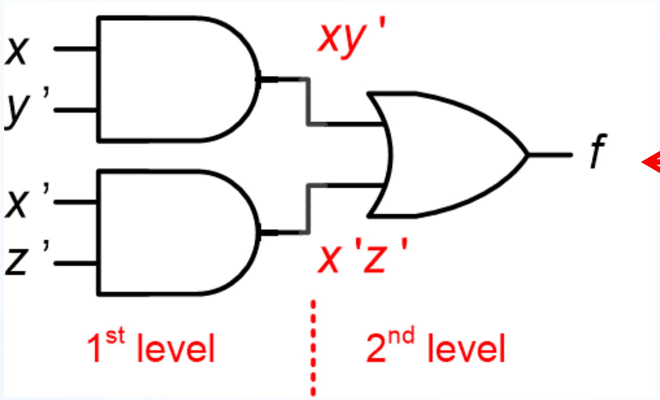
1. Boolean Algebra
2. Circuit schematic
3. Truth table

For the truth table, find the output as a following:

1. Write down all possible input combinations
2. Write down the final stage output (i.e. $A(B + CD)$)

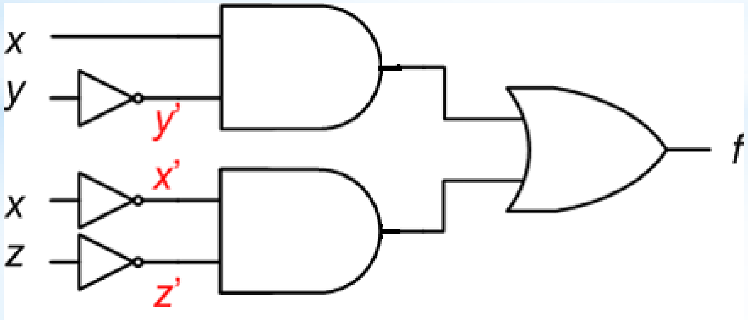
Logic circuit from a Boolean expression

Provided that a Boolean function $f(x,y,z)=xy'+x'z'$, then the logic circuit can be formed as:



a 2-level circuit

or



Boolean function $\rightarrow$ Truth Table

Example: $f(x,y,z)=xy'+x'z'$

Input(s)		Output
x y z	xy' $x'z'$	$xy' + x'z'$
0 0 0	0 1	1
0 0 1	0 0	0
0 1 0	0 1	1
0 1 1	0 0	0
1 0 0	1 0	1
1 0 1	1 0	1
1 1 0	0 0	0
1 1 1	0 0	0

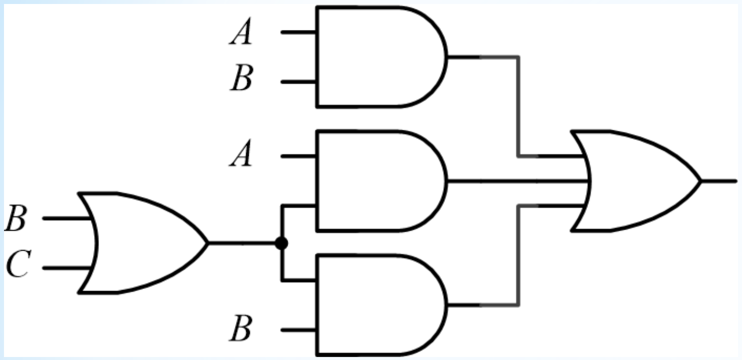
Truth Table → Boolean function

Inputs	Output
a b c	*f*
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	0
1 1 1	0

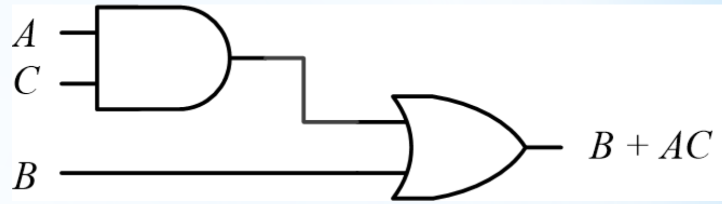
f is 1 if $\{(a = 0)$ AND $(b = 0)$ AND $(c = 1)\}$ OR
$\{(a = 1)$ AND $(b = 0)$ AND $(c = 0)\}$ OR
$\{(a = 1)$ AND $(b = 0)$ AND $(c = 1)\}$

$$f(a,b,c) = a'b'c + \underbrace{ab'c' + ab'c}_{\text{Is it the simplest form?}}$$

Application of Boolean Algebra - simplification



(a)



(b)

Prove that the above Circuit (a) is equivalent to Circuit (b).

Solution by Boolean Algebra Simplification

$AB + A(B + C) + B(B + C)$		
$= AB + AB + AC + BB + BC$		
$= AB + AB + AC + B + BC$	$BB=B$	*Idempotent*
$= AB + AC + B + BC$	$AB+AB=AB$	*Idempotent*
$= AB + AC + B$	$B+BC=B$	*Absorption*
$= B + AC$	$AB+B=B$	*Absorption*

3.4 Logic Function - SOP and POS

Logic functions are generally expressed with least number of literals (variables).

- ***Sum of products*** **(SOP)**

$$f(a,b,c,d) = ab'c + b'd' + a'cd$$

- ***Product of sums*** **(POS)**

$$f(a,b,c,d) = (a' + b + c)(b' + c + d')(a + c')$$

3.4.1 Minterm and Maxterm

Minterm :- For a function of *n* variables, if a ***product term*** contains all *n* variables **exactly one time** in ***its*** complemented or uncomplemented form, the product term is called *minterm*. Complement = 0 and Uncomplement =1.

Function	Minterm	Not minterm	Not minterm
f(A, B, C)	*A' B' C*	*(A B)' C*	*A'B' or AABC*

Maxterm :- If a ***sum term*** of a function of *n* variables contains all *n* variables **exactly one time** in ***its*** complemented or uncomplemented form, the sum term is called a *maxterm*. Complement = 1 and Uncomplement =0.

Function	Maxterm	Not maxterm	Not maxterm
f(A, B, C)	*A'+ B'+ C*	*(A + B)' C*	*A'+ B' or A'+B+B+C*

Noted that the minterm and maxterm cannot be simplified.

Minterms and maxterms for 3 variables logic function

			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	$x'y\,z'$	m_2	$x + y' + z$	M_2
0	1	1	$x'y\,z$	m_3	$x + y' + z'$	M_3
1	0	0	$x\,y'z'$	m_4	$x' + y + z$	M_4
1	0	1	$x\,y'z$	m_5	$x' + y + z'$	M_5
1	1	0	$x\,y\,z'$	m_6	$x' + y' + z$	M_6
1	1	1	$x\,y\,z$	m_7	$x' + y' + z'$	M_7

The number of minterms and maxterm of a logic function of n variables equals to 2^n, e.g. f(a,b,c,d) ➔ 16 minterms (maxterms);

f(a,b,c,d,e) ➔ 32 minterms (maxterms).

3.4.2 Canonical form

The canonical form of a logic function is a presentation in either minterms of maxterms.

In minterm form (with logic output “1”):

$$f(A,B,C) = \overline{A}B\overline{C} + AB\overline{C} + \overline{A}BC + ABC$$

Minterm	Code	Number
A’BC’	010	m_2
ABC’	110	m_6
A’BC	011	m_3
ABC	111	m_7

$$f(A,B,C) = m_2 + m_3 + m_6 + m_7$$

Canonical form $$f(A,B,C) = \sum m(2,3,6,7)$$

Noted for minterm: Complement = 0 and Uncomplement = 1

The canonical form of a logic function is a presentation in either minterms of maxterms.

In maxterm form (with logic output "0"):

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

Maxterm	Code	Number
$A + B + C$	000	M_0
$A + B + C'$	001	M_1
$A' + B + C$	100	M_4
$A' + B + C'$	101	M_5

$$f(A,B,C) = M_0 M_1 M_4 M_5$$

Canonical form $f(A,B,C) = \prod M(0,1,4,5)$

Noted for maxterm: Complement = 1 and Uncomplement = 0

Canonical SOP representation:

$$f(a,b,c) = a + \bar{b}c$$

$$= a(b+\bar{b}) + \bar{b}c(a+\bar{a})$$

$$= ab(c+\bar{c}) + a\bar{b}(c+\bar{c}) + \bar{b}c(a+\bar{a})$$

$$= abc + ab\bar{c} + a\bar{b}c + a\bar{b}\bar{c} + a\bar{b}c + \bar{a}\bar{b}c$$

$$= m_1 + m_4 + m_5 + m_6 + m_7$$

$$f(a,b,c) = \sum m(1,4,5,6,7)$$

Canonical POS representation:

$$
\begin{aligned}
f(x,y,z) &= xy + \bar{x}z \\
&= (xy + \bar{x})(xy + z) = (x + \bar{x})(y + \bar{x})(x + z)(y + z) \\
&= (\bar{x} + y)(x + z)(y + z) = (\bar{x} + y + z\bar{z})(x + z + y\bar{y})(y + z + x\bar{x}) \\
&= (\bar{x} + y + z)(\bar{x} + y + \bar{z})(x + z + y)(x + \bar{y} + z)(x + y + z)(\bar{x} + y + z) \\
&= (x + y + z)(x + \bar{y} + z)(\bar{x} + y + z)(\bar{x} + y + \bar{z})
\end{aligned}
$$

$$
f(x,y,z) = M_0 M_2 M_4 M_5 = \Pi M(0,2,4,5)
$$

Conversion between canonical forms

Relationship

$\overline{maxterm_i} = minterm_i$ (i.e. $\overline{M_i} = m_i$)

$\overline{minterm_i} = maxterm_i$ (i.e. $\overline{m_i} = M_i$)

e.g. $M_3' = (a + b' + c')$
$= a'bc$ (De Morgan's Theorem)
$= m_3$

Canonical form conversion: SOP ←→ POS

For minterm representation (**choose all the terms with output 1**):

$$f(x,y,z) = \overline{x}\,\overline{y}z + x\overline{y}\,\overline{z} + x\overline{y}z + xyz$$

$$f(x,y,z) = m_1 + m_4 + m_5 + m_7 = \sum m(1,4,5,7)$$

For maxterm representation (**choose all the terms with output 0**):

$$\overline{f(x,y,z)} = \overline{x}\,\overline{y}\,\overline{z} + \overline{x}y\overline{z} + \overline{x}yz + xy\overline{z}$$

$$f(x,y,z) = \overline{\overline{f(x,y,z)}} = \overline{\overline{x}\,\overline{y}\,\overline{z} + \overline{x}y\overline{z} + \overline{x}yz + xy\overline{z}}$$

$$= (x+y+z)(x+\overline{y}+z)(x+\overline{y}+\overline{z})(\overline{x}+\overline{y}+z)$$

$$f(x,y,z) = M_0 M_2 M_3 M_6 = \Pi M(0,2,3,6)$$

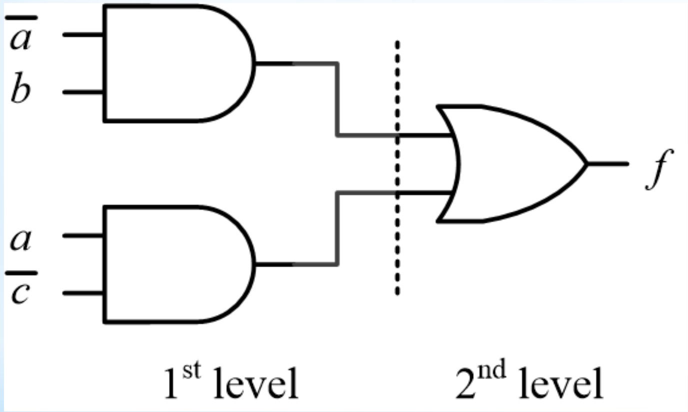
x y z	*f*
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	1
1 0 1	1
1 1 0	0
1 1 1	1

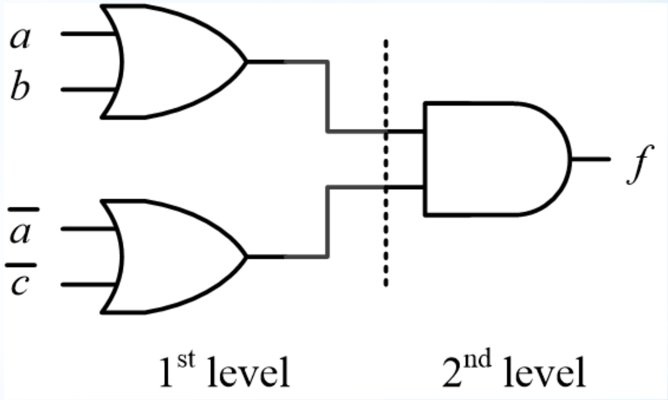
Noted that for maxterm the uncomplemented variable = 0, complemented variable = 1

3.5 Synthesis of logic function

$$f(a,b,c) = \overline{a}b + a\overline{c}$$

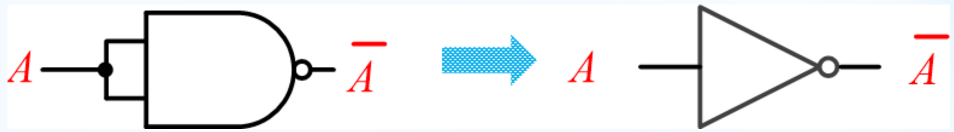
$$f(a,b,c) = (a+b)(\overline{a}+\overline{c})$$

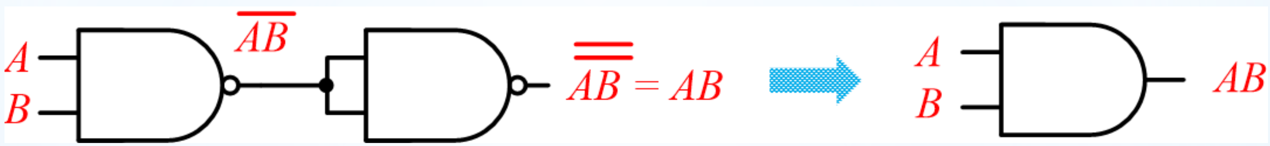


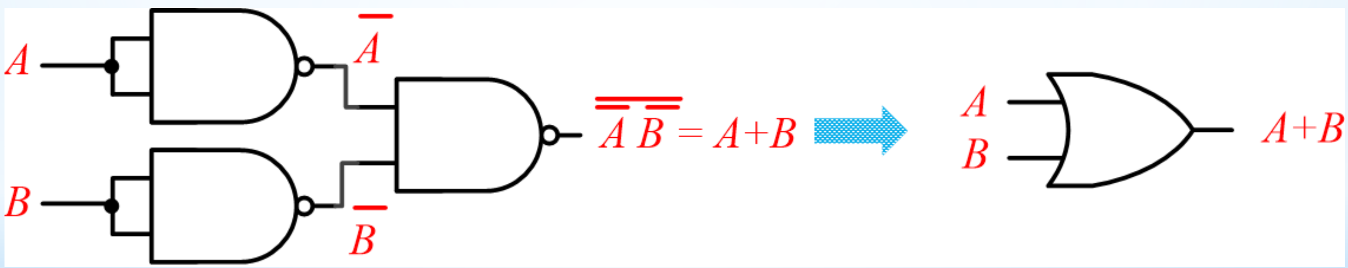


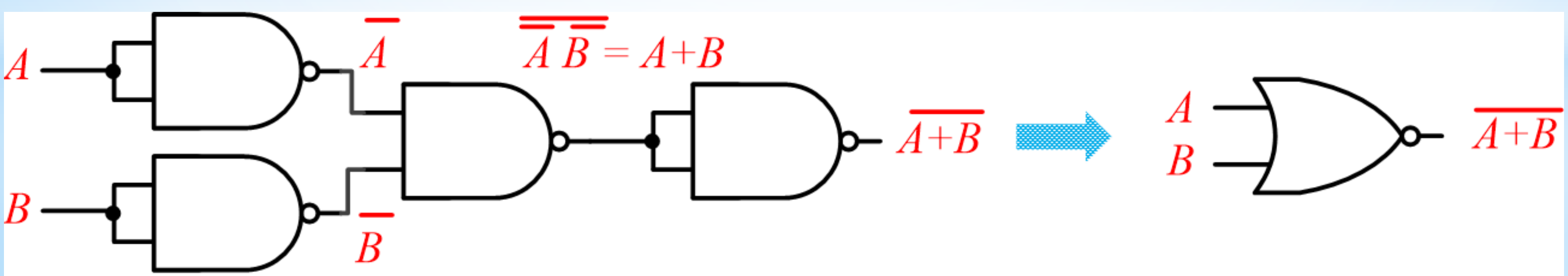
3.6 Structure of one gate type

Universal gate**: NAND** gate

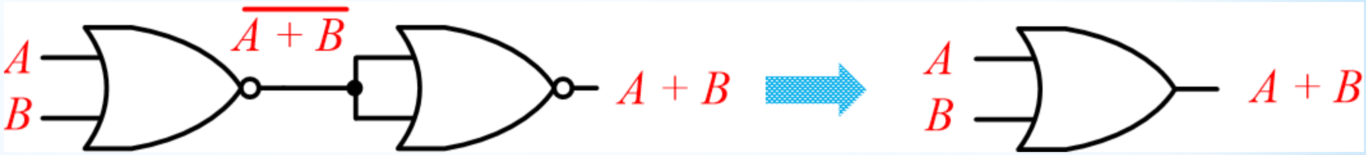


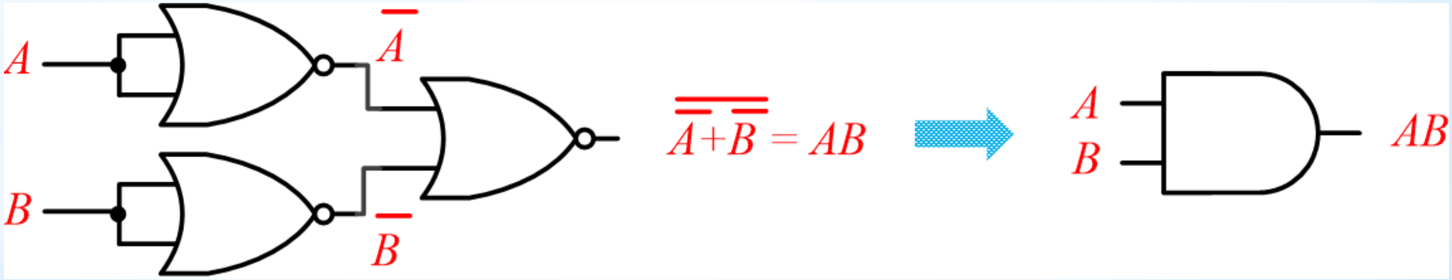


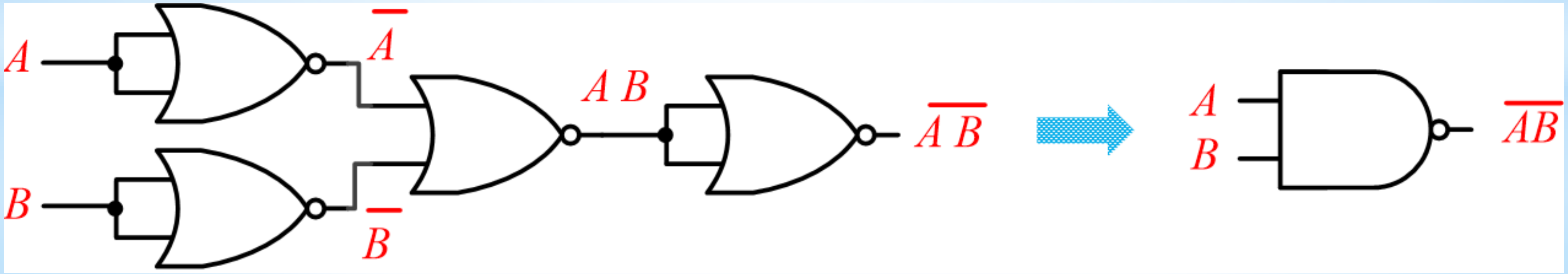




Universal gate: NOR gate







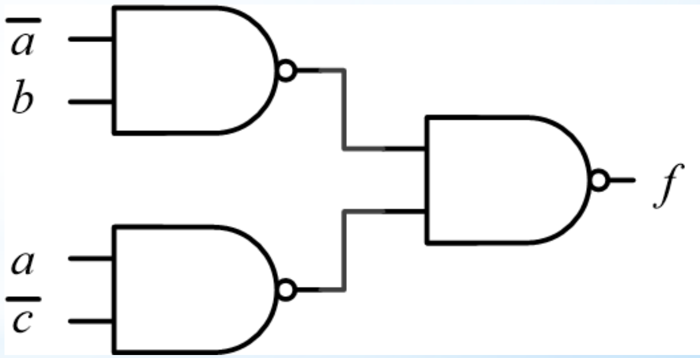
NAND implementation

Modify the Boolean function to get rid of the "OR" operation

$$f(a,b,c) = \bar{a}b + a\bar{c}$$

$$= \overline{\overline{\bar{a}b + a\bar{c}}}$$

$$= \overline{\overline{\bar{a}b}\;\overline{a\bar{c}}}$$

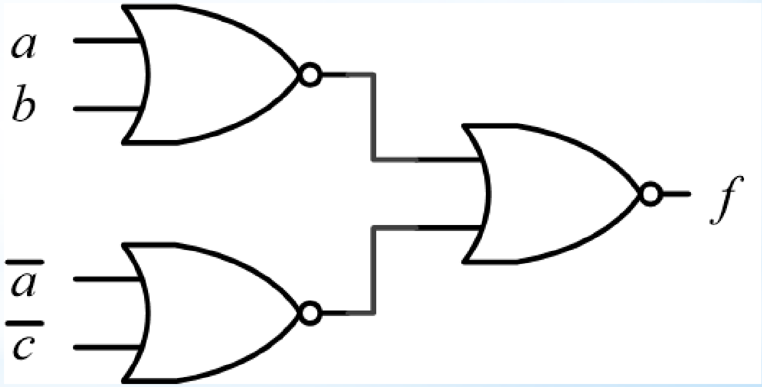


Note inverter can be implemented using NAND gate.

NOR implementation

Modify the Boolean function to get rid of the "AND" operation

$$
\begin{aligned}
f(a,b,c) &= (a+b)(\bar{a}+\bar{c}) \\
&= \overline{\overline{(a+b)(\bar{a}+\bar{c})}} \\
&= \overline{\overline{a+b}+\overline{\bar{a}+\bar{c}}}
\end{aligned}
$$



Note inverter can be implemented using NOR gate.