Logic Simplification

Boolean Equation Minimization

EP
Revision 9/22/13

BOOLEAN ALGEBRA

SIMPLIFICATION OF CIRCUITS IS IMPERATIVE AS THIS PROCESS CAN HELP IN THE FOLLOWING CONSIDERATIONS:

- SAVE REAL ESTATE
- LOWER COST
- IMPROVE PERFORMANCE

Axioms of Boolean Algebra

OR FUNCTION

$0 + 0 = 0$

$0 + 1 = 1 + 0 = 1$

$1 + 1 = 1$

AND FUNCTION

$0 . 0 = 0$

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

$1 \cdot 1 = 1$

NOT FUNCTION

IF $x = 0 \rightarrow \overline{x} = 1$

IF $x = 1 \rightarrow \overline{x} = 0$

SINGLE VARIABLE THEOREMS

USING THE AXIOMS, WE CAN DEFINE:

$$A + 0 = A$$

$$A + 1 = 1$$

$$A + A = A$$

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

$$A \cdot 0 = 0$$

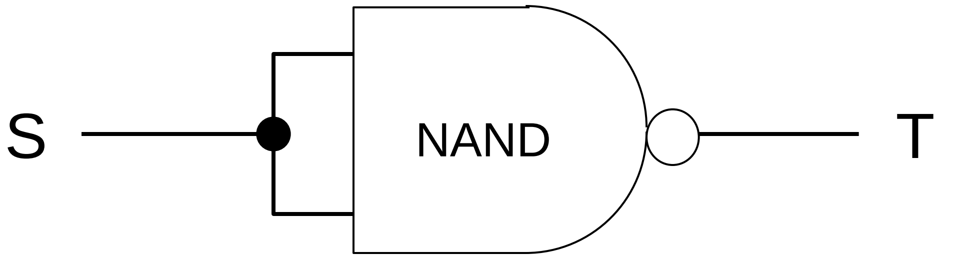
$$A \cdot 1 = A$$

$$A \cdot A = A$$

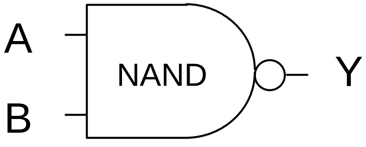
$$A \cdot \overline{A} = 0$$

$$\overline{\overline{A}} = A$$

NAND Gate - Special Application



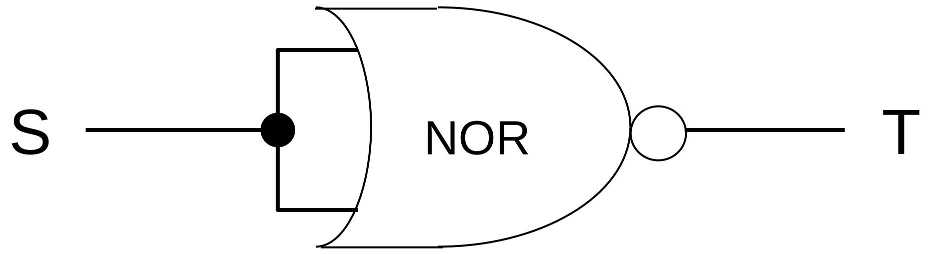
S	T
0	1
1	0



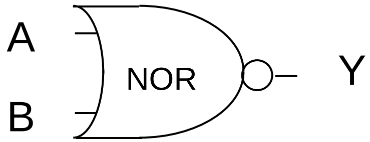
INPUTS		OUTPUT
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Equivalent To An Inverter Gate

NOR Gate - Special Application



S	T
0	1
1	0



INPUTS		OUTPUT
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Equivalent To An Inverter Gate

DUALITY PRINCIPAL

GIVEN A LOGIC EXPRESSION, ITS DUAL IS OBTAINED BY REPLACING ALL + OPERATORS WITH . OPERATORS AND VICE VERSA, AND BY REPLACING 0's WITH 1's AND VICE VERSA

USING THE DUALITY PRINCIPAL WE CAN GENERATE EQUIVALENT STATEMENTS

EXAMPLE:

$0 + 0 = 0 \rightarrow 1 . 1 = 1$

$0 . 0 = 0 \rightarrow 1 + 1 = 1$

SINGLE VARIABLE THEOREMS

USING THE AXIOMS, WE CAN DEFINE (REVISITED):

$$A + 0 = A \rightarrow A \cdot 1 = A$$

$$A + 1 = 1 \rightarrow A \cdot 0 = 0$$

$$A + A = A \rightarrow A \cdot A = A$$

$$A + A' = 1 \rightarrow A \cdot A' = 0$$

Boolean Algebra Properties

Property	***Expression***
Commutative	*Addition:* **A + B = B + A** *Multiplication:* **A B = B A**
Associative	*Addition:* **A + (B+C) = (A+B) + C = A+B+C** *Multiplication:* **A(BC) = (AB)C = ABC**
Distributive	**A(B+C) = AB + AC** **(A + B)(C + D) = AC+BC+AD+BD**

Boolean Algebra Properties

$A + AB = A$

Work the left side

$A(1 + B) = A(1) = A$

$A + A'B = A + B$

Work the right side

$(A + B) = (A + B)(1) = (A + B)(A + A')$

$AA + BA + AA' + BA'$

$A + AB + 0 + A'B$

$A + A'B$

Example: Prove that $A' + AB = A' + B$

Solution: Let $A' = X$ → Thus $A = X'$ → Substitute into the equation:

We get: $X + X'B = X + B$ which has been proven earlier, thus:

✓ $A' + AB = A' + B$

Boolean Algebra Properties

$AB + AB' = A$

$A(B + B') = A(1) = A$

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

$AA + AC + AB + BC = A + AC + AB + BC$

$A + AB + BC = A + BC$

WE CAN PROVE THE VALIDITY OF THESE PROPERTIES EITHER BY PERFECT INDUCTION OR BY PERFORMING ALGEBRAIC MANIPULATION

INDUCTION METHOD:

TRUTH TABLE FORMAT

ALGEBRAIC MANIPULATION METHOD:

USE AXIOMS, THEOREMS, PROPERTIES TO PROVE VALIDITY BY SHOWING THAT THE LHS = RHS

Boolean Simplification

Rule applied	$AABCBBC1 = X$
Commutative	$AABBBCC1 = X$
A·A = A	$ABC1 = X$
A·1 = A	$ABC = X$

Boolean Simplification

Rule applied	$KLMKMKS0 = X$
Commutative.	$KKKLMMS0 = X$
A·A = A	$KLMS0 = X$
A·0 = 0	$0 = X$

Boolean Simplification

Rule Applied	$CC + BCD + DDCB = X$
A · A = A	$C + BCD + DCB = X$
Commutative	$C + BCD + BCD = X$
A + A = A	$C + BCD = X$
Commutative.	$C + CBD = X$
A + AB = A	$C = X$

Boolean Simplification

Rule Applied	$DCB + 0 = X$
A + 0 = A	$DCB = X$

Boolean Simplification

Rule Applied	$CC + BD + 1 = X$
A · A = A	$C + BD + 1 = X$
A + 1 = 1	$1 = X$

Boolean Simplification

Rule Applied	$(C + CB) + (\overline{C}B + C) = X$
A + AB = A	$(C) + (\overline{C}B + C) = X$
A + $\overline{\text{A}}$B= A + B	$(C) + (C + B) = X$
Associative.	$C + C + B = X$
A + A = A	$C + B = X$

Boolean Simplification

Rule Applied	$\overline{A} + AB + \overline{C}D + D = X$
$\overline{A} + AB = \overline{A} + B$	$\overline{A} + B + \overline{C}D + D = X$
Commutative.	$\overline{A} + B + D + \overline{C}D = X$
$A + AB = A$	$\overline{A} + B + D = X$

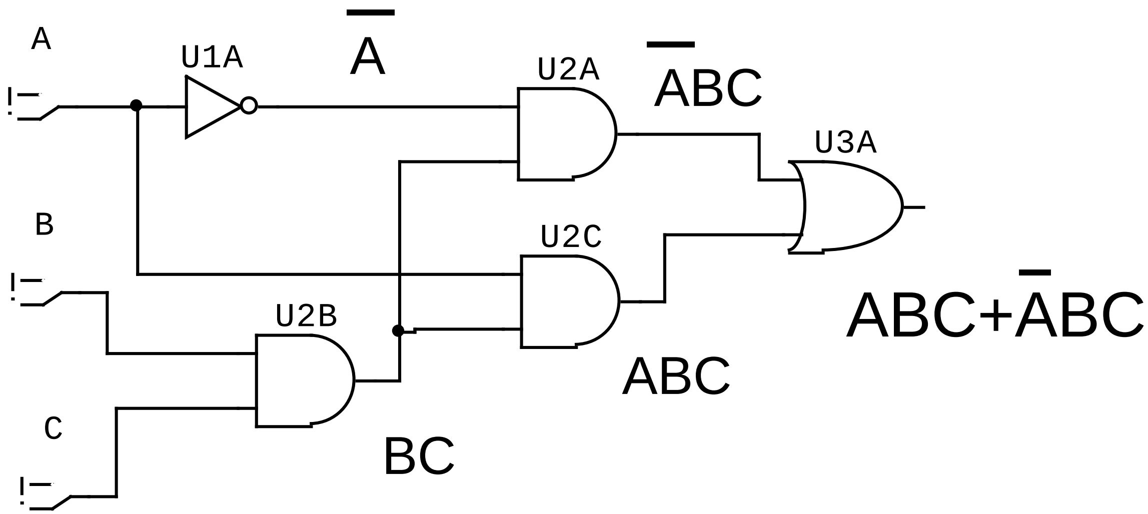
Boolean Simplification

Rule Applied	$\bar{A}BCD + \bar{A}BCD + \bar{A}\bar{B}CD + AB\bar{C}D = X$
A + A = A	$\bar{A}BCD + \bar{A}\bar{B}CD + AB\bar{C}D = X$
Distributive.	$\bar{A}CD(B + \bar{B}) + AB\bar{C}D = X$
A + $\bar{\text{A}}$ = 1	$\bar{A}CD(1) + AB\bar{C}D = X$
A · 1 = A	$\bar{A}CD + AB\bar{C}D = X$

Example # 1:

Develop a circuit and truth table from a Boolean expression

Boolean Expression: $\mathbf{X = \overline{A}\,B\,C + A\,B\,C}$



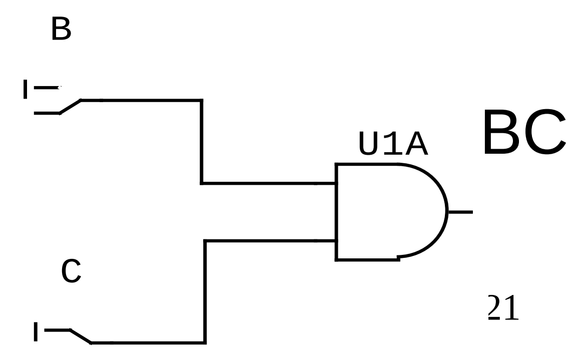
Simplify the Expression:

$X = A\,B\,C + \overline{A}\,B\,C$

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

$B\,C\,(\,1\,)$

Unsimplified Circuit



Example # 1: Unsimplified

$$X = \overline{A}\,B\,C + A\,B\,C$$

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

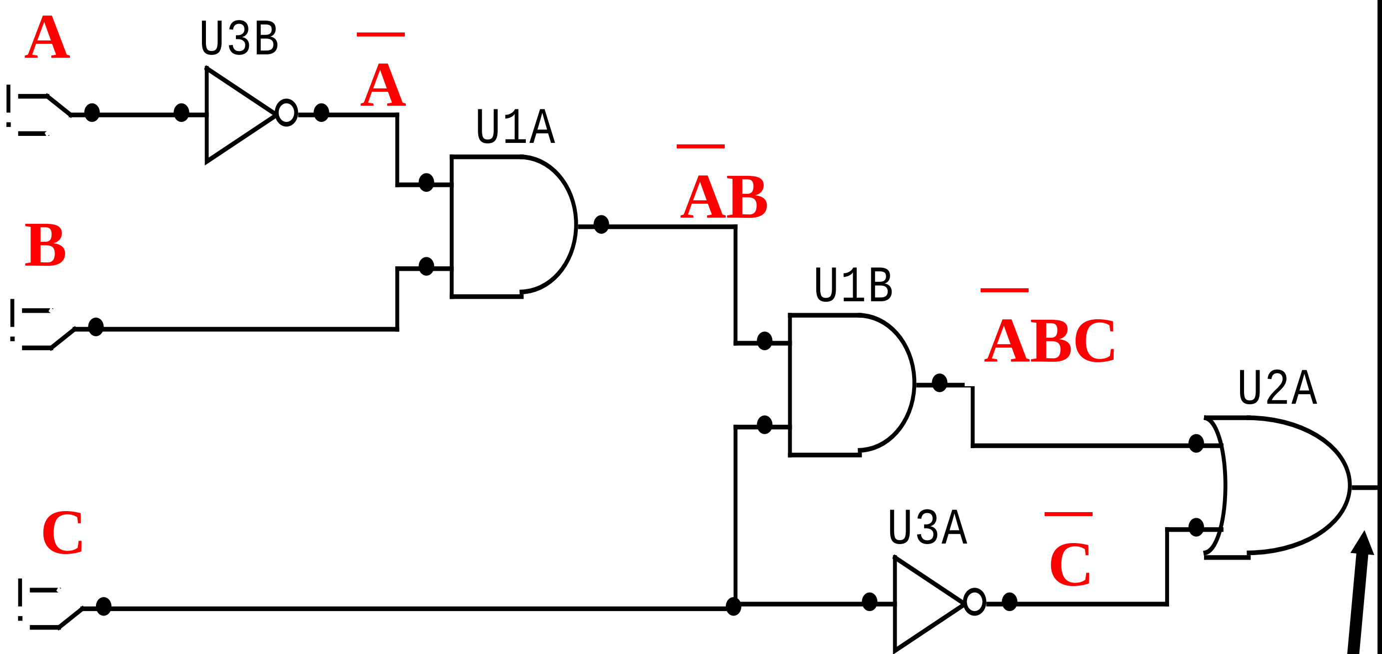
Example # 1: Simplified

$$X = B\,C$$

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Example # 2

Develop a truth table and Boolean Expression from a circuit:



A	B	C	X
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Boolean Expression:

$$X = \overline{A}\,B\,C + \overline{C}$$

Example # 2

Unsimplified Boolean Expression:

$$X = \overline{A}\,B\,C + \overline{C}$$

Simplified Boolean Expression:

$$\overline{A}\,B\,C + \overline{C} = \overline{C} + C\,[\overline{A}\,B]$$

Let: $C = X$ and $\overline{A}\,B = Z$

We get: $\overline{X} + X\,Z = \overline{X} + Z = \overline{C} + \overline{A}\,B$

We get:

$$X = \overline{A}\,B\,C + \overline{C} = \overline{A}\,B + \overline{C}$$

DeMorgan's Theorems

$$\overline{A\,.\,B} = \overline{A} + \overline{B} \quad \rightarrow \quad \overline{A + B} = \overline{A}\,.\,\overline{B}$$

$$A\,.\,B = \overline{\overline{A\,.\,B}} = \overline{\overline{A} + \overline{B}}$$

$$A + B = \overline{\overline{A + B}} = \overline{\overline{A}\,.\,\overline{B}}$$

DeMorgan’s Theorem #1

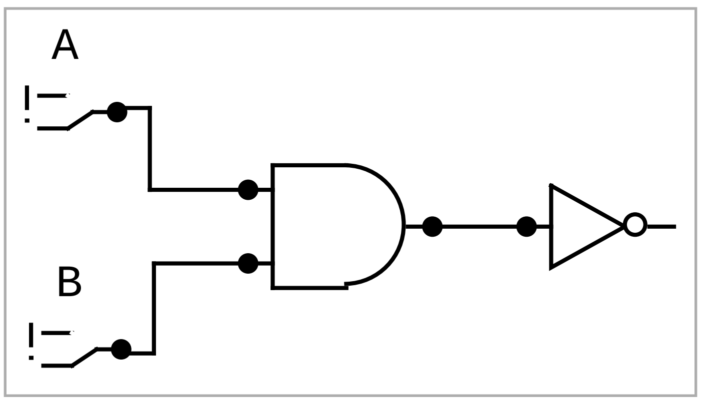
$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

A	B	$A \bullet B$	$\overline{A \bullet B}$		$\overline{A}$	$\overline{B}$	$\overline{A} + \overline{B}$
0	0	0	1		1	1	1
0	1	0	1		1	0	1
1	0	0	1		0	1	1
1	1	1	0		0	0	0

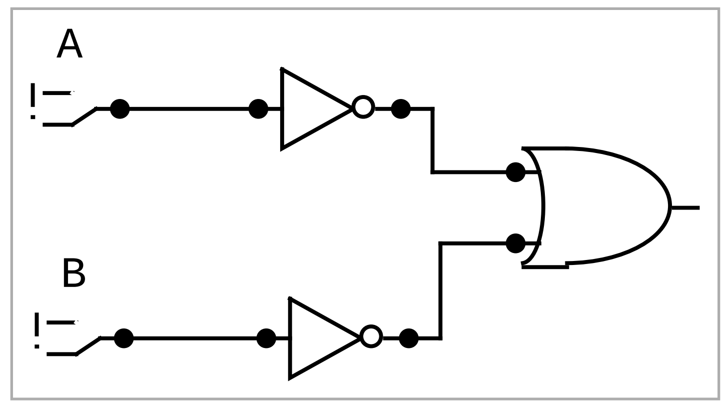
EQUAL

DeMorgan’s Theorem #1

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



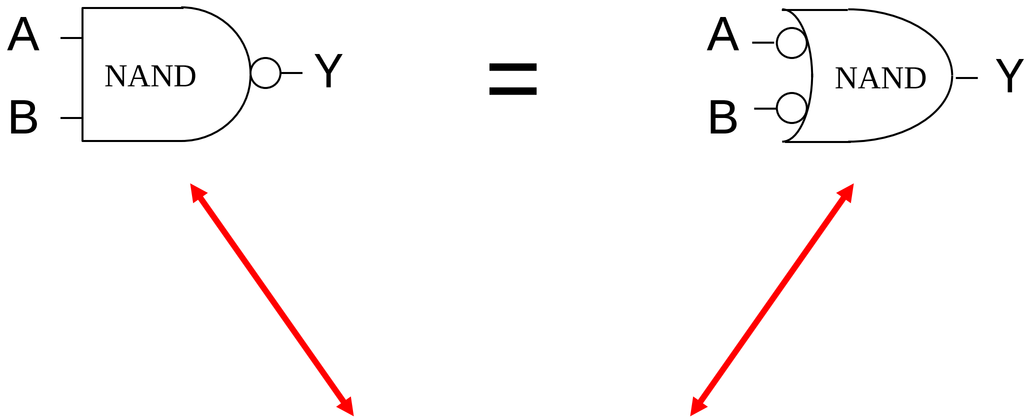
=



Invert output of an AND gate

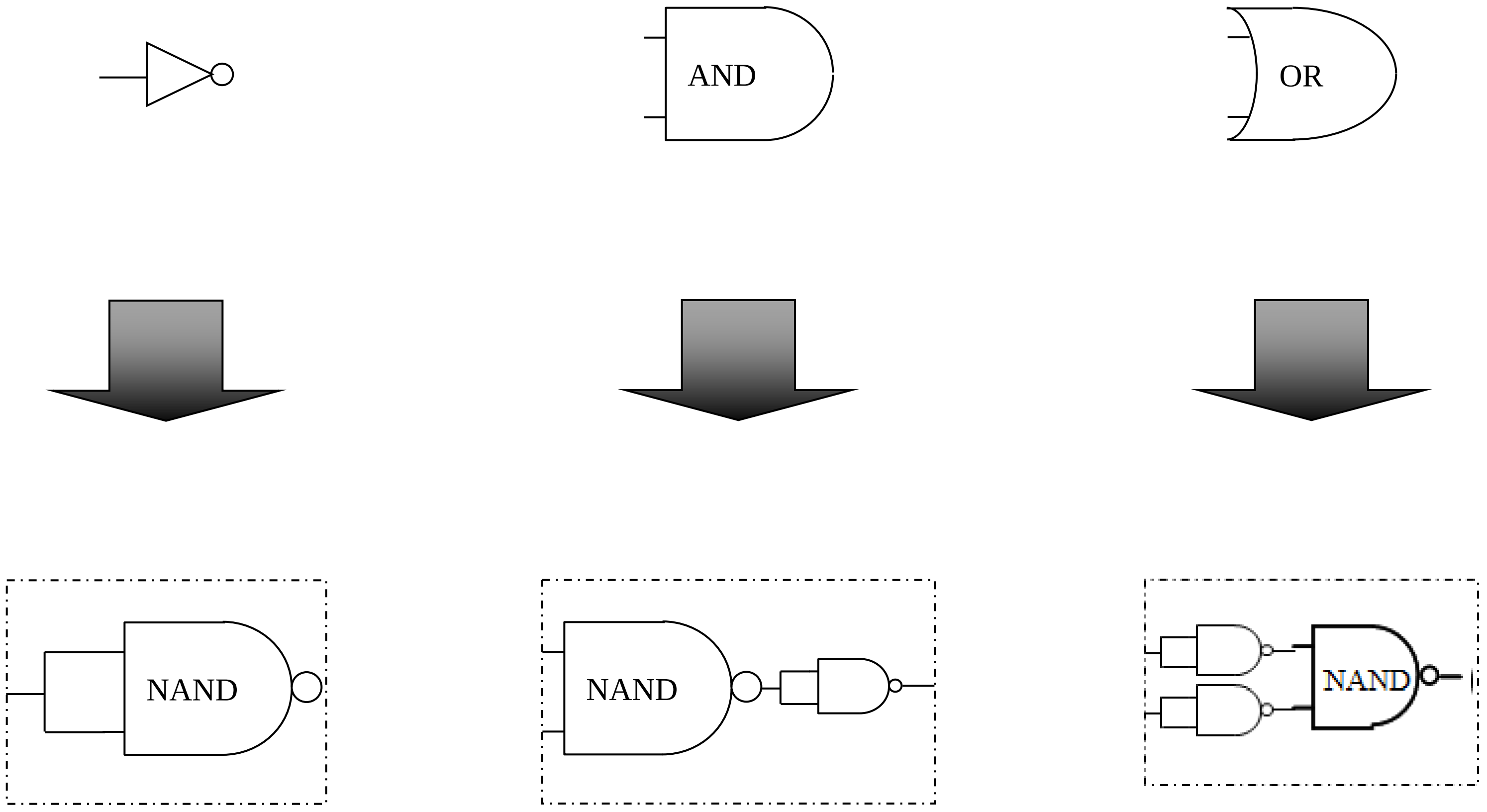
Invert the inputs of an OR gate

NAND Logic

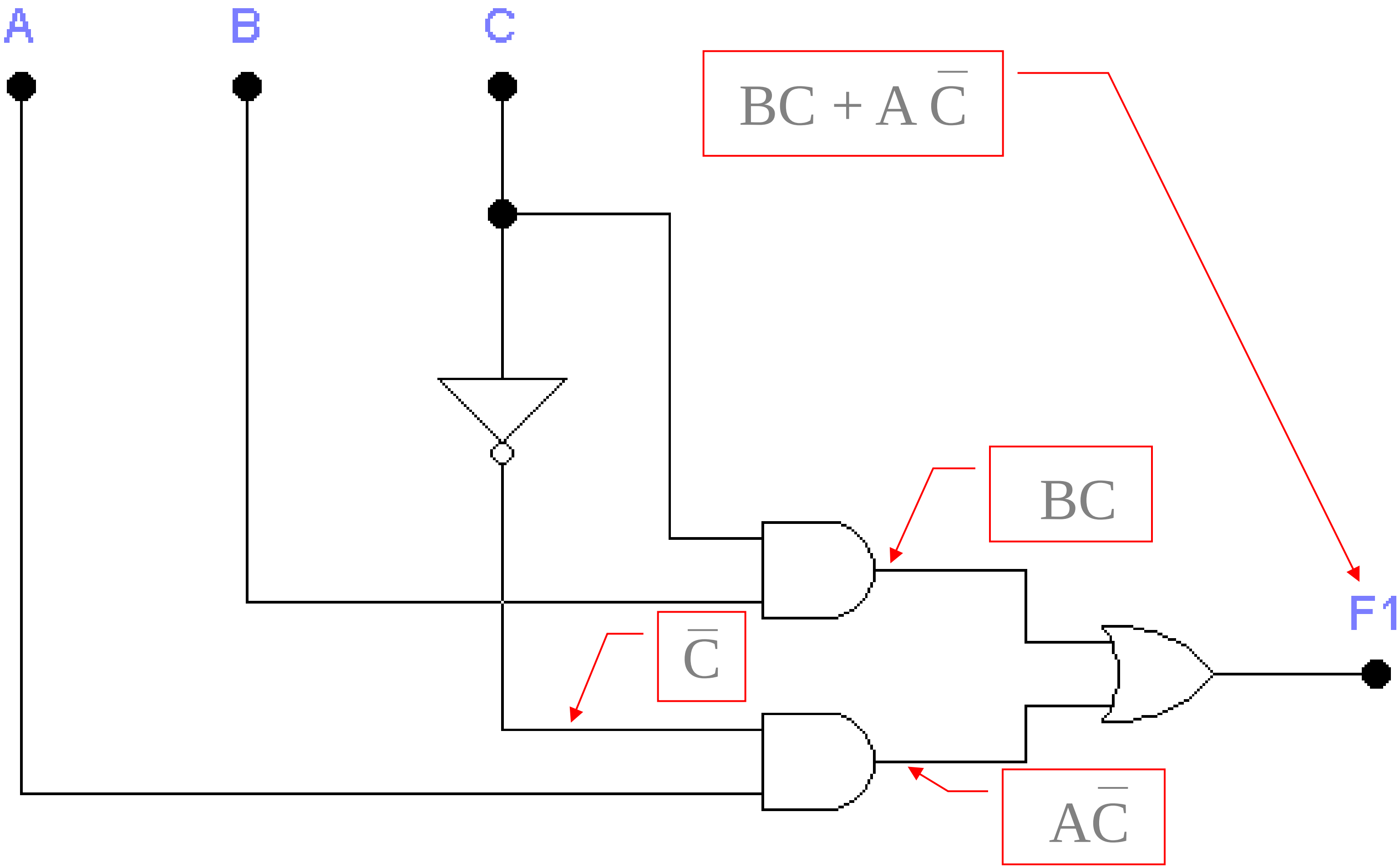


$$Y = \overbrace{\overline{A \times B}} = \overbrace{\overline{A} + \overline{B}}$$

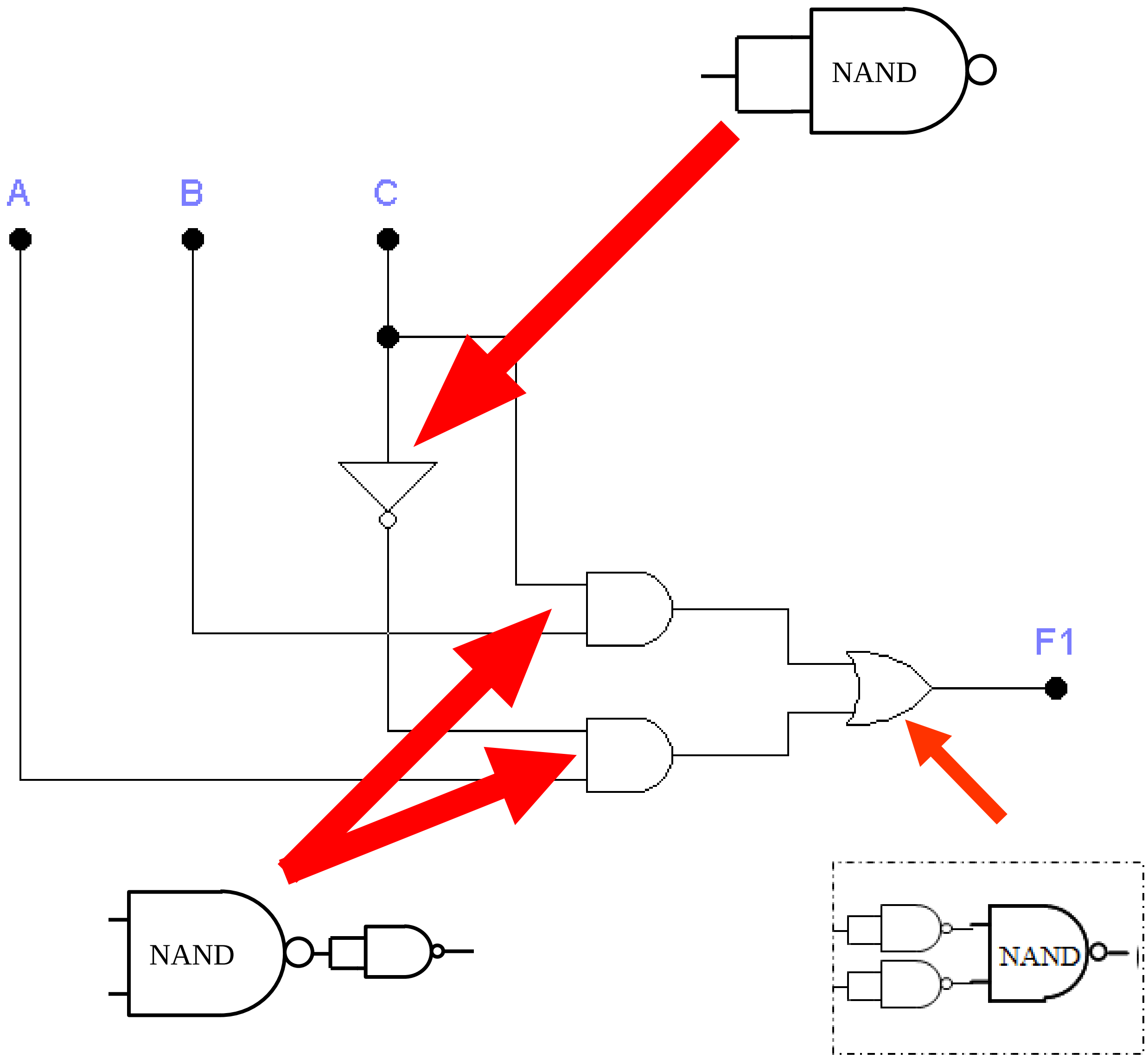
A-O-I ⇒ NAND Only Logic



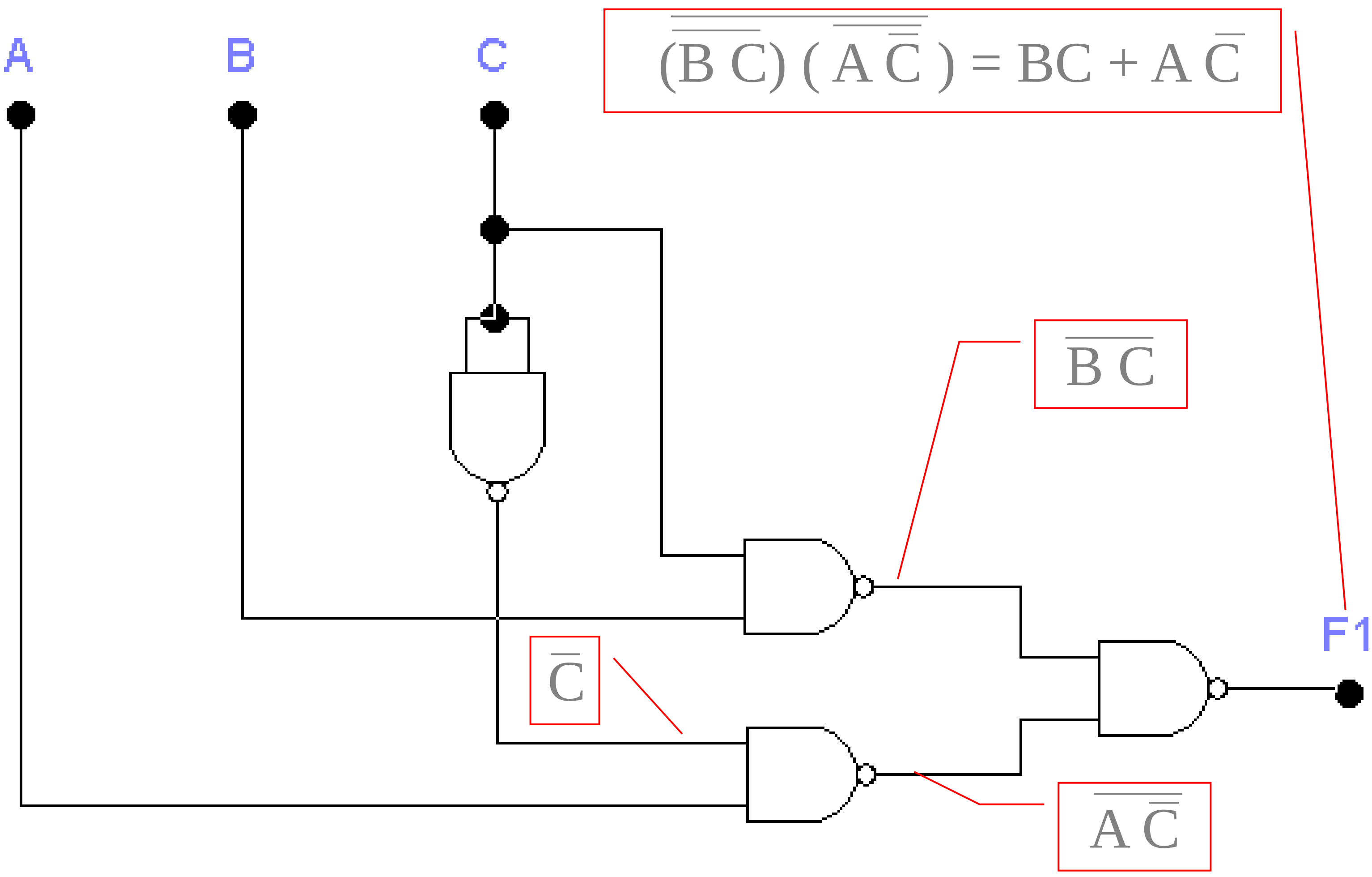
A-O-I Design Example



A-O-I $\Rightarrow$ NAND Only Logic



NAND-ONLY Design Example



DeMorgan’s Theorem #2

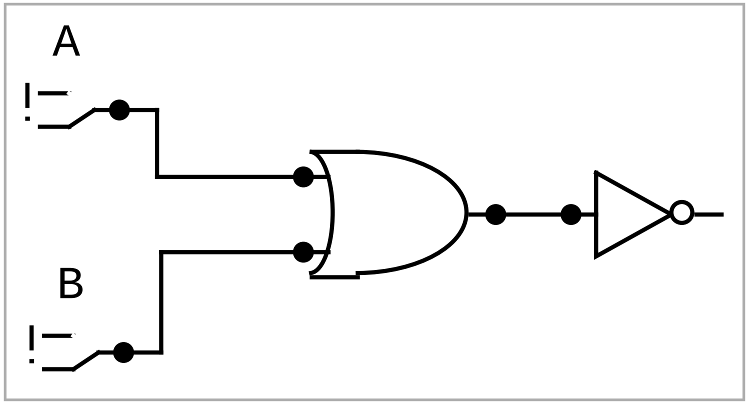
$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

A	B	A + B	$\overline{A + B}$		$\overline{A}$	$\overline{B}$	$\overline{A} \times \overline{B}$
0	0	0	1		1	1	1
0	1	1	0		1	0	0
1	0	1	0		0	1	0
1	1	1	0		0	0	0

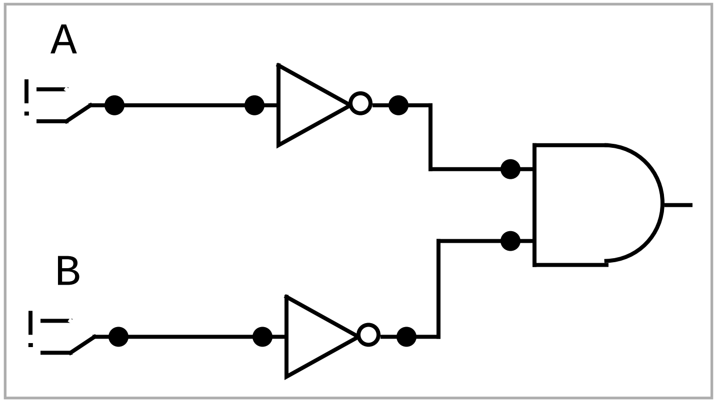
EQUAL

DeMorgan's Theorem #2

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



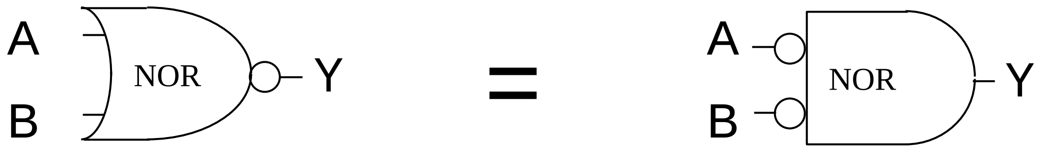
=



Invert output of an OR gate

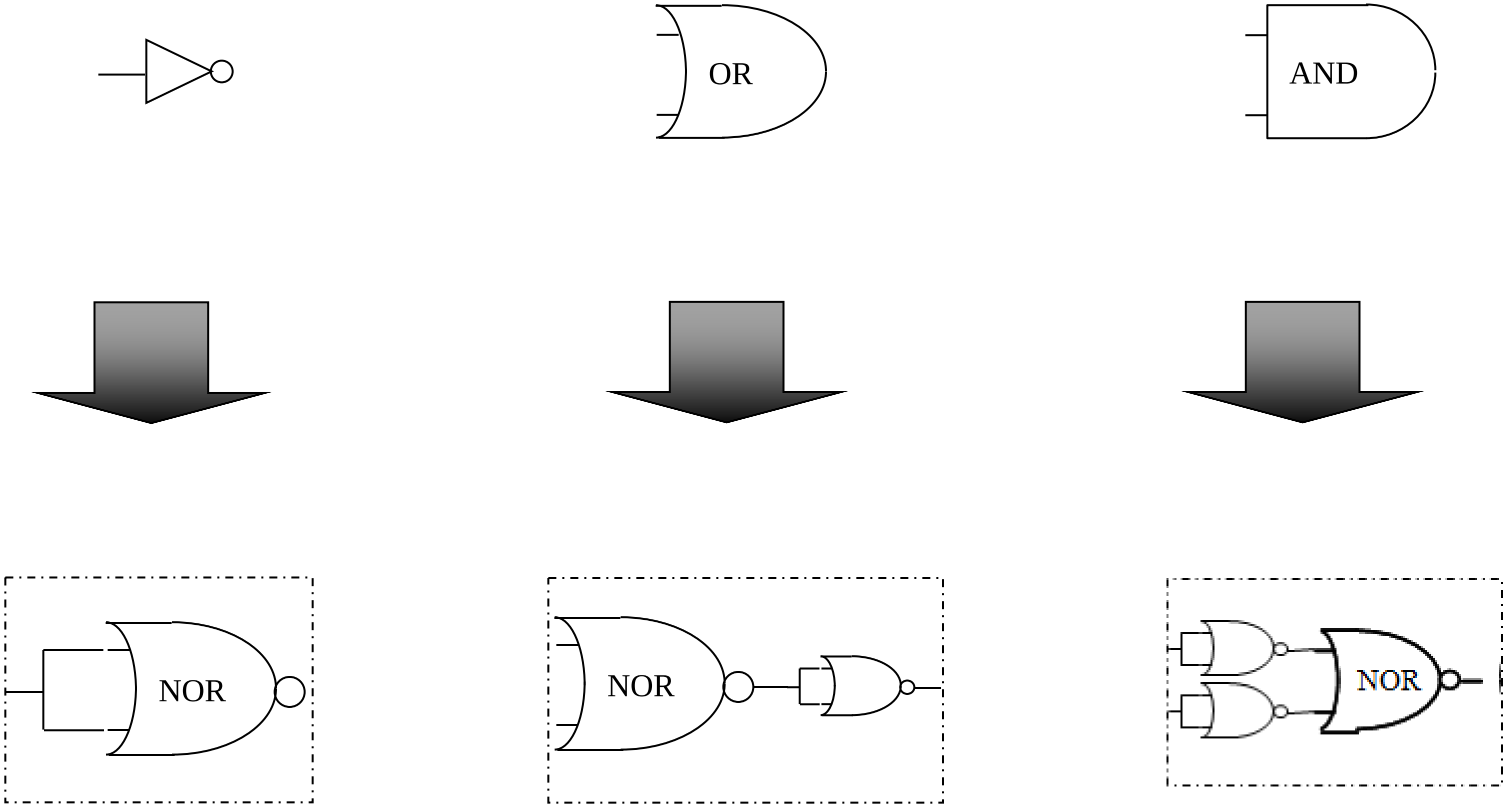
Invert the inputs of an AND gate

NOR Logic

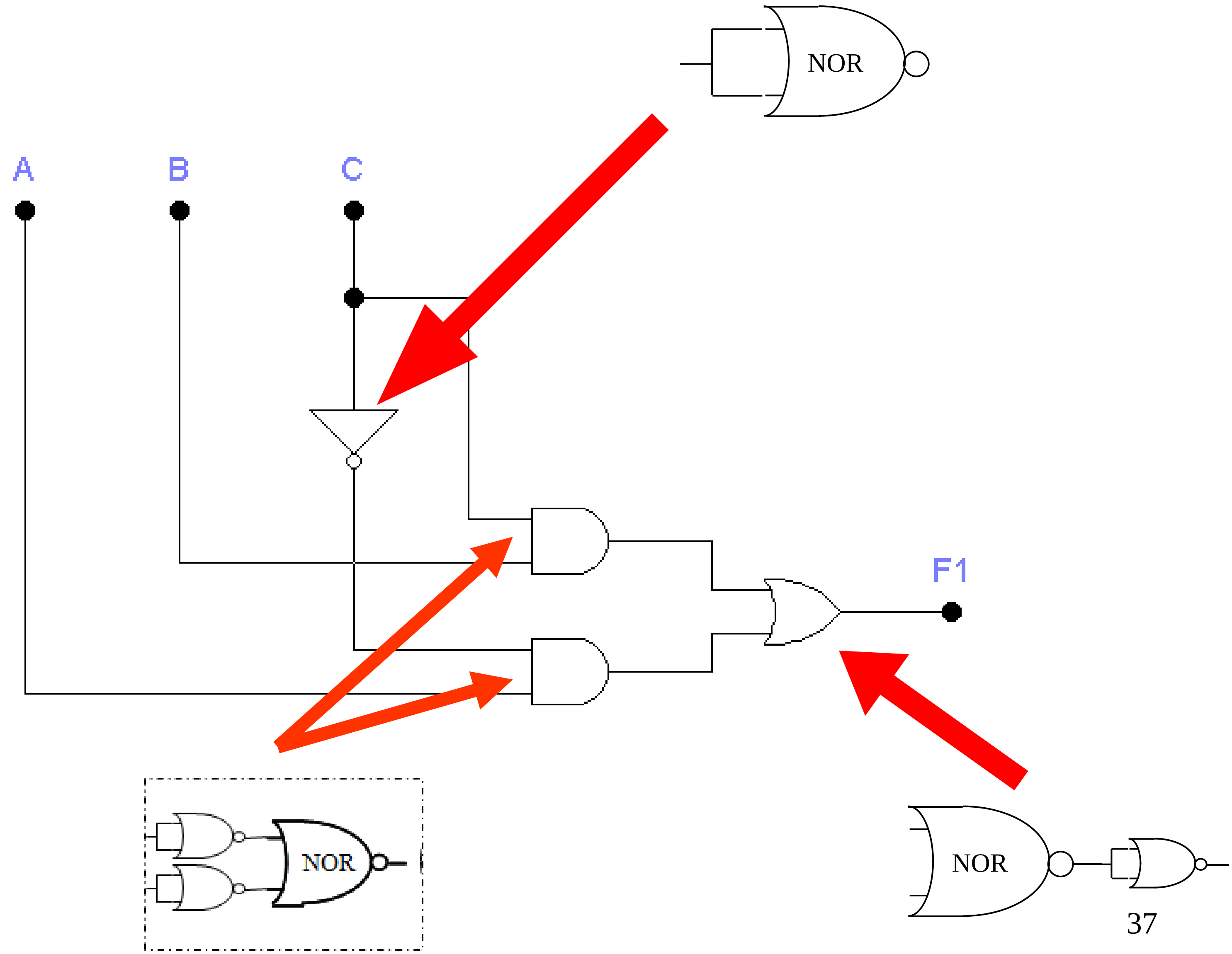


$$Y = \overline{A + B} = \overline{A} \times \overline{B}$$

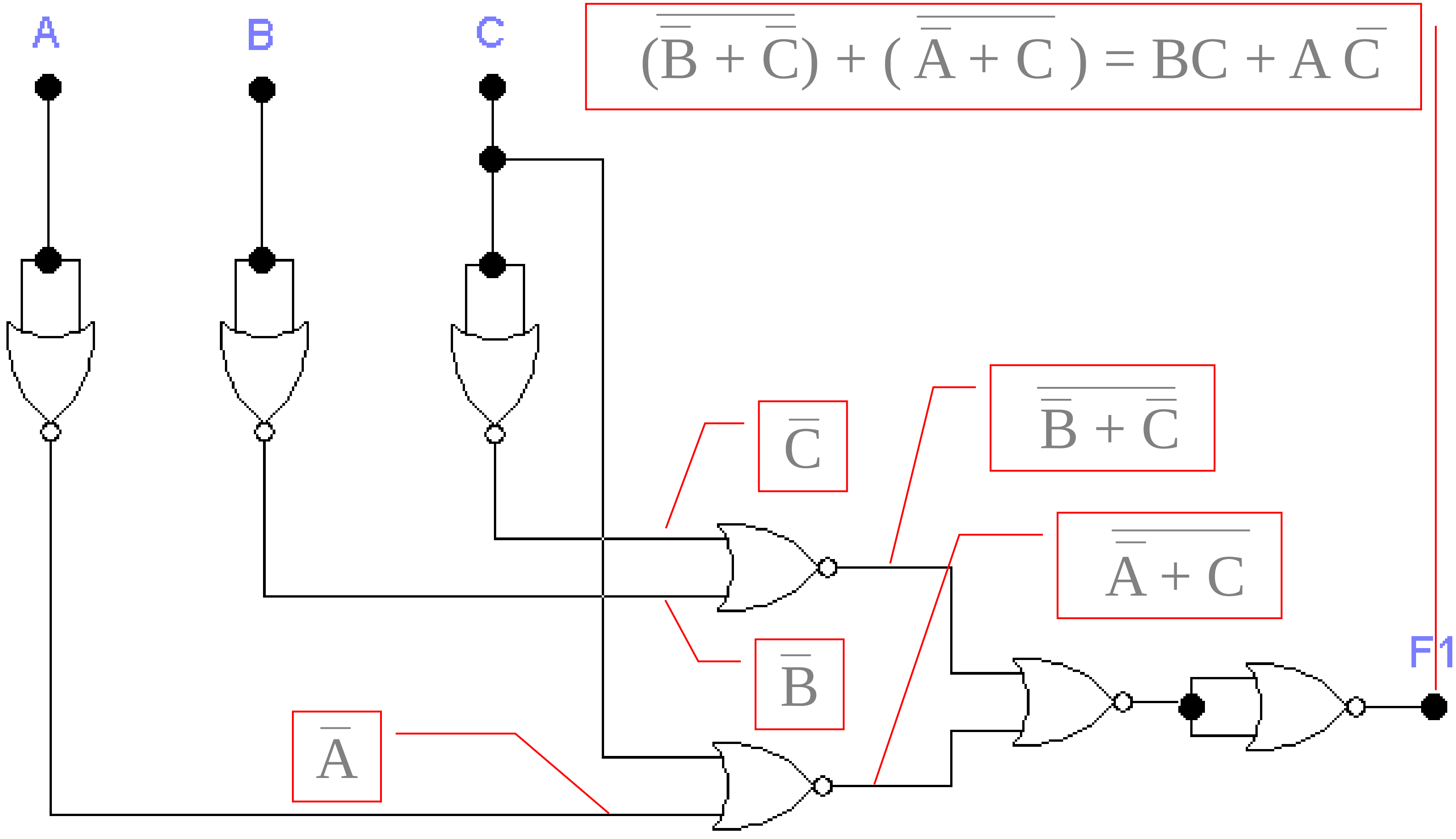
A-O-I ⇒ NOR Only



A-O-I ⇒ NOR



NOR-Only Design Example



DeMorganizing an Expression

	$\overline{\overline{(A\overline{C})} + (B\overline{C}) + (A+C)(B\overline{C})} = X$
DeMorg.	$(A\overline{C})(\overline{B}+C)(\overline{A}\,\overline{C}) + (\overline{B}+C) = X$
Dist.	$(A\overline{C}\,\overline{B} + A\overline{C}C)(\overline{A}\,\overline{C}) + (\overline{B}+C) = X$
Dist.	
A’ A = 0 C’ C = 0	$(0+0) + (\overline{B}+C) = X$

DeMorganizing an Expression

$$(\overline{C}\,\overline{B}0 + 00) + (\overline{B} + C) = X$$

A 0 = 0

A + 0 = A

$$(0) + (\overline{B} + C) = X$$

$$(\overline{B} + C) = X$$

BOOLEAN THEOREMS

1. A + B = B + A *COMMUTATIVITY*
   AB = BA
2. A + (B + C) = (A + B) + C *ASSOCIATIVITY*
   A(B C) = (A B)C
3. A(B + C) = AB + AC *DISTRIBUTIVITY*
   (A + B)(C + D) = AC + AD + BC + BD

LAWS

1. A . 0 = 0
2. A . 1 = A
3. A + 0 = A
4. A + 1 = 1
5. A . A = A
6. A + A = A
7. A . A’ = 0
8. A + A’ = 1
9. A” = A

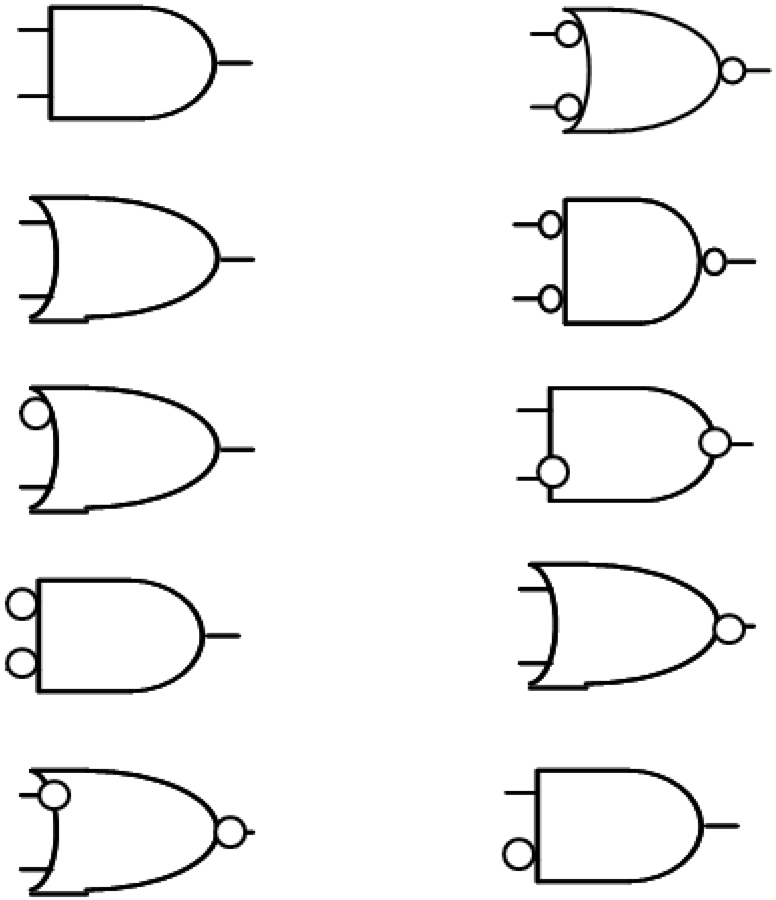
10. A + AB = A
    A(A + B) = A
    A + A’B = A + B
    A’ + AB = A’ + B

RULES

1. (A + B)’ = A’ . B’
2. (A . B)’ = A’ + B’
3. (A + B) = (A’ . B’)’
4. (A . B) = (A’ + B’)’

DeMorgan

BUBBLE PUSHING



SYNTHESIS USING AND, OR, NOT GATES

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

<u>f is true (1) when</u>:

- x1 is false (not true) AND x2 is false (not true)

OR

- x1 is false (not true) AND x2 is true

OR

- x1 is true AND x2 is true

f(x1, x2) = x1’ x2’ + x1’ x2 + x1 x2

SYNTHESIS USING AND, OR, NOT GATES

$f(x1, x2) = x1' x2' + x1' x2 + x1 x2$

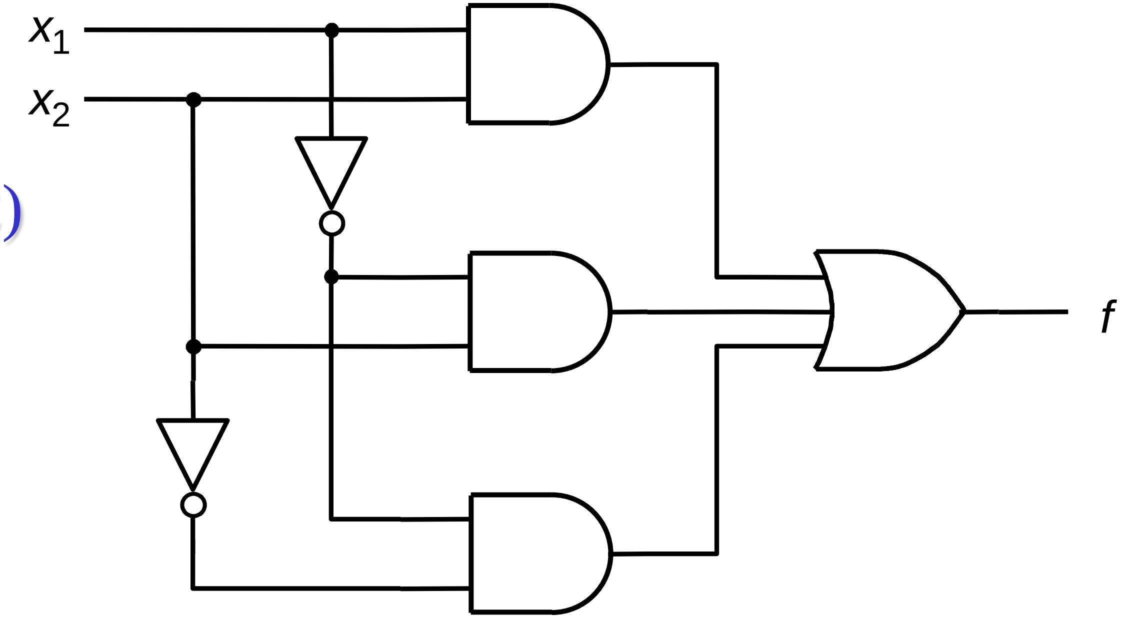
0 0 → 0' 0' → 1 1 → 1

0 1 → 0' 1 → 1 1 → 1

1 1 → 1

3 PRODUCT TERMS FORM THIS SUM OF PRODUCTS (SOP)

CREATE A PRODUCT TERM THAT HAS A VALUE OF 1 FOR EACH TRUTH TABLE ROW THAT CAUSES THE OUTPUT TO BE TRUE (f = 1) f(x1, x2) = x1’ x2’ + x1’ x2 + x1 x2 = x1’ x2’ + x1’ x2 + x1 x2 + x1’x2



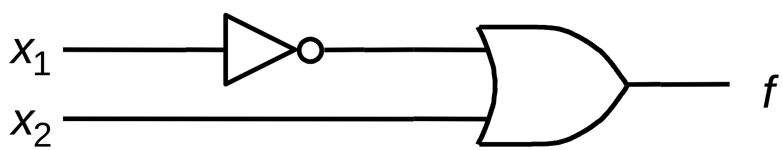
x2(x1’+x1)+ x1’(x2’+x2)

x2 . 1 + x1’ . 1

x2 + x1’

(a) Canonical sum-of-products

f(x1, x2) = x2 + x1’



(b) Minimal-cost realization

Figure 2.16 Two implementations of a function

CANONICAL FORM SOP → EACH PRODUCT TERM IS A MINTERM (ALL VARIABLES ARE IN THE PRODUCT TERM)

Row number	x_1	x_2	x_3	Minterm	Maxterm
0	0	0	0	$m_0 = \overline{x}_1\overline{x}_2\overline{x}_3$	$M_0 = x_1 + x_2 + x_3$
1	0	0	1	$m_1 = \overline{x}_1\overline{x}_2x_3$	$M_1 = x_1 + x_2 + \overline{x}_3$
2	0	1	0	$m_2 = \overline{x}_1x_2\overline{x}_3$	$M_2 = x_1 + \overline{x}_2 + x_3$
3	0	1	1	$m_3 = \overline{x}_1x_2x_3$	$M_3 = x_1 + \overline{x}_2 + \overline{x}_3$
4	1	0	0	$m_4 = x_1\overline{x}_2\overline{x}_3$	$M_4 = \overline{x}_1 + x_2 + x_3$
5	1	0	1	$m_5 = x_1\overline{x}_2x_3$	$M_5 = \overline{x}_1 + x_2 + \overline{x}_3$
6	1	1	0	$m_6 = x_1x_2\overline{x}_3$	$M_6 = \overline{x}_1 + \overline{x}_2 + x_3$
7	1	1	1	$m_7 = x_1x_2x_3$	$M_7 = \overline{x}_1 + \overline{x}_2 + \overline{x}_3$

Used in SOP　　Used in POS

f → 1　　f → 0

Figure 2.17　Three-variable Minterms and Maxterms

EXAMPLE

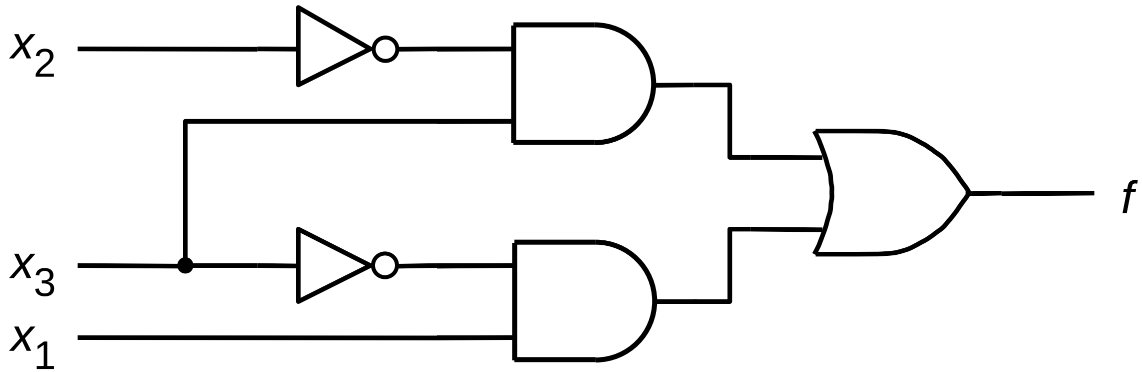
Row number	x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

f(x3,x2,x1) = x1’ x2’ x3 + x1 x2’ x3’ + x1 x2’ x3 + x1 x2 x3’

LSB → MSB

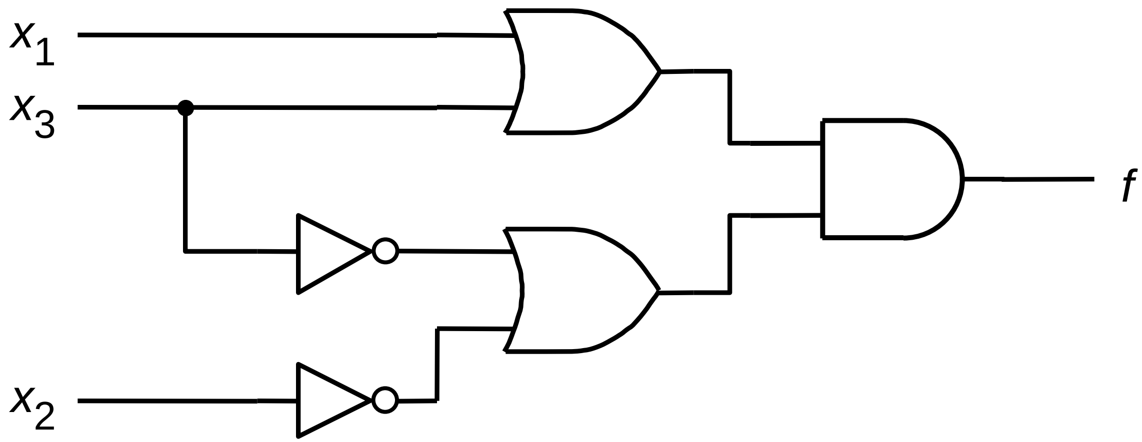
Figure 2.18 A three-variable function

Derive



(a) A minimal sum-of-products realization

Derive



(b) A minimal product-of-sums realization

Figure 2.19 Two realizations of a function

Example # 3:

Develop a logic circuit and Boolean expression from a truth table

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Boolean Expression:

$$X = \bar{A}\,B\,\bar{C} + \bar{A}\,B\,C + A\,\bar{B}\,C$$

Simplify:

$$\bar{A}B(\cancelto{1}{\bar{C} + C}) + A\bar{B}C$$

$$\bar{A}B + A\bar{B}C$$