Boolean Expressions, Universal Gates and Truth-Tables

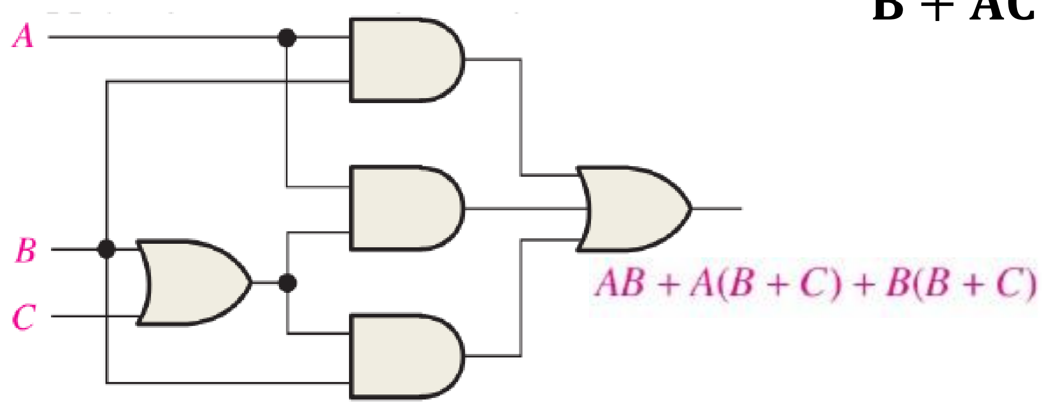
LOGIC SIMPLIFICATION

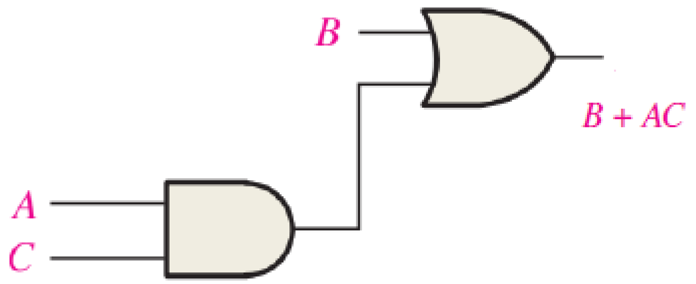
- A large Boolean Logic Expression can often be simplified to a simpler and shorter Boolean Logic Expression. This is done by applying the laws and rules of Boolean Algebra.
- Using Boolean algebra techniques, simplify this expression:

$$\mathbf{AB + A(B + C) + B(B + C)}$$

This can be simplified to:

$$\mathbf{B + AC}$$





LOGIC SIMPLIFICATION & IMPLEMENTATION

Simplify the Boolean expression $A\overline{B} + A(\overline{B + C}) + B(\overline{B + C})$.

Simplify the following Boolean expression:

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

Simplify the following Boolean expression:

$$\overline{A}BC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C + ABC$$

Simplify the following Boolean expression:

$$\overline{AB + AC} + \overline{A}\,\overline{B}C$$

Simplify the Boolean expression $\overline{AB} + \overline{AC} + \overline{A}\,\overline{B}\,\overline{C}$.

1. Simplify the following Boolean expressions:

   (a) $A + AB + A\overline{B}C$ (b) $(\overline{A} + B)C + ABC$ (c) $A\overline{B}C(BD + CDE) + A\overline{C}$

2. Implement each expression in Question 1 as originally stated with the appropriate logic gates. Then implement the simplified expression, and compare the number of gates.

***Applying rules of Boolean algebra and DeMorgan's Theorem show that

i) $\mathbf{MN + MO + MN O\; MN + O = 1}$

ii) $\mathbf{A\,BC + A\,BC + A(BC + ABC) = B}$

Universal Gates

- Although we can implement **any circuit** with **AND/OR/NOT**, we can also implement **any circuit** with **only NAND** or **NOR** gates.
- We might want to do this because of technology considerations, that is, these gates might be cheaper to implement in silicon or they might be the only type of gates we have available.
- Since we can always use only NAND or NOR gates, these gates are sometimes called **universal gates**.
- The “trick” (if you want to call it that) is to see that we can implement the three basic gates (AND, OR, NOT) in terms of NAND or NOR gates.

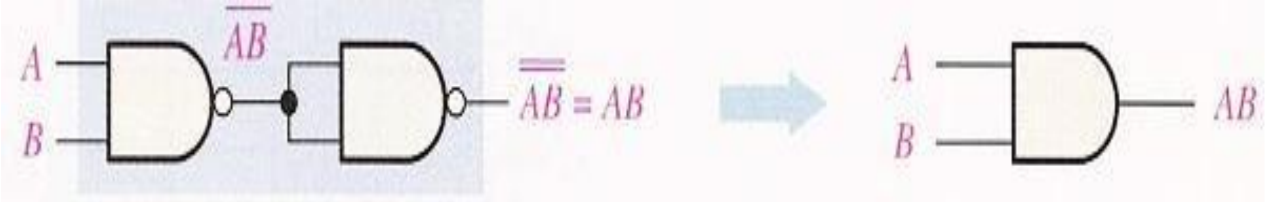
Universal Property of NAND and NOR gate

The NAND gate as a Universal logic gate:

- 1) <u>One NAND gate used as an inverter (**NOT gate**)</u>:

$$f = \overline{x \bullet x} = \overline{x}$$

- 2) <u>Two NAND gates used as an **AND gate**</u>

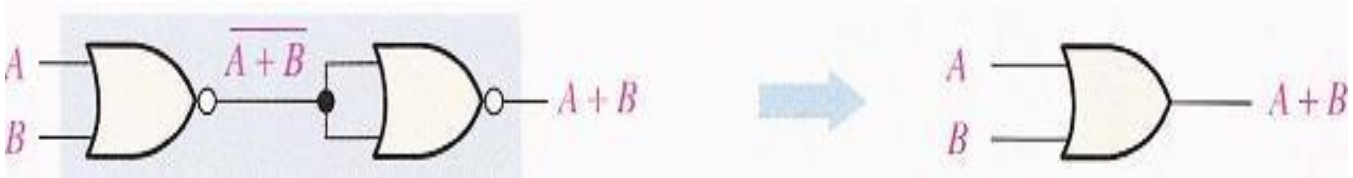


-) <u>Three NAND gates used as an **OR gate**</u>

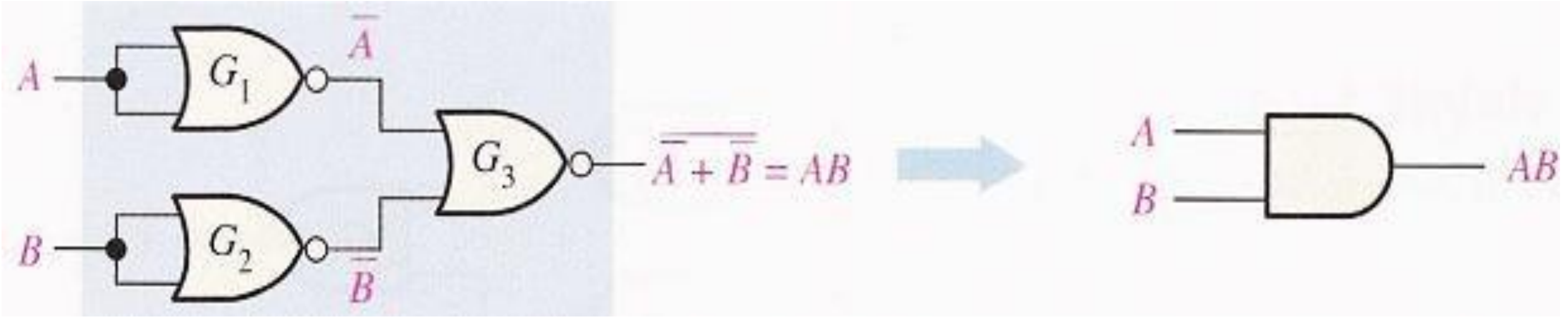
The NOR gate as a Universal logic gate:

- 1) <u>One NOR gate used as an inverter (**NOT gate**)</u>:

- 2) <u>Two NOR gates used as an **OR gate**</u>:



- 3) <u>Three NOR gates used as an **AND gate**</u>:



Steps for Construction of Logic circuits Using Universal Gates

1) First construct the logic expression using basic gates (AND/OR/NOT),
2) Replace each basic gate with its equivalent universal gate implementation,
3) Cancel two consecutive NOT gates (according to Boolean algebra),
4) Redraw the final circuit.

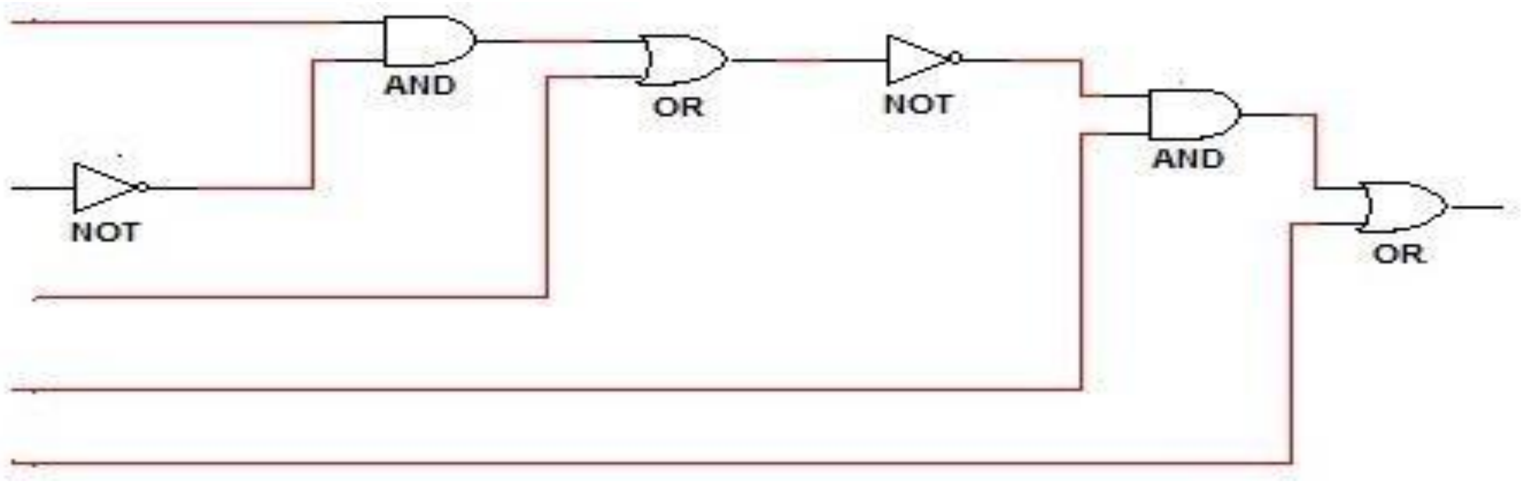
Example: Implement the following expression with (a) NAND gates ONLY, (b) NOR gates ONLY.

Y=(AB'+C)'D+E

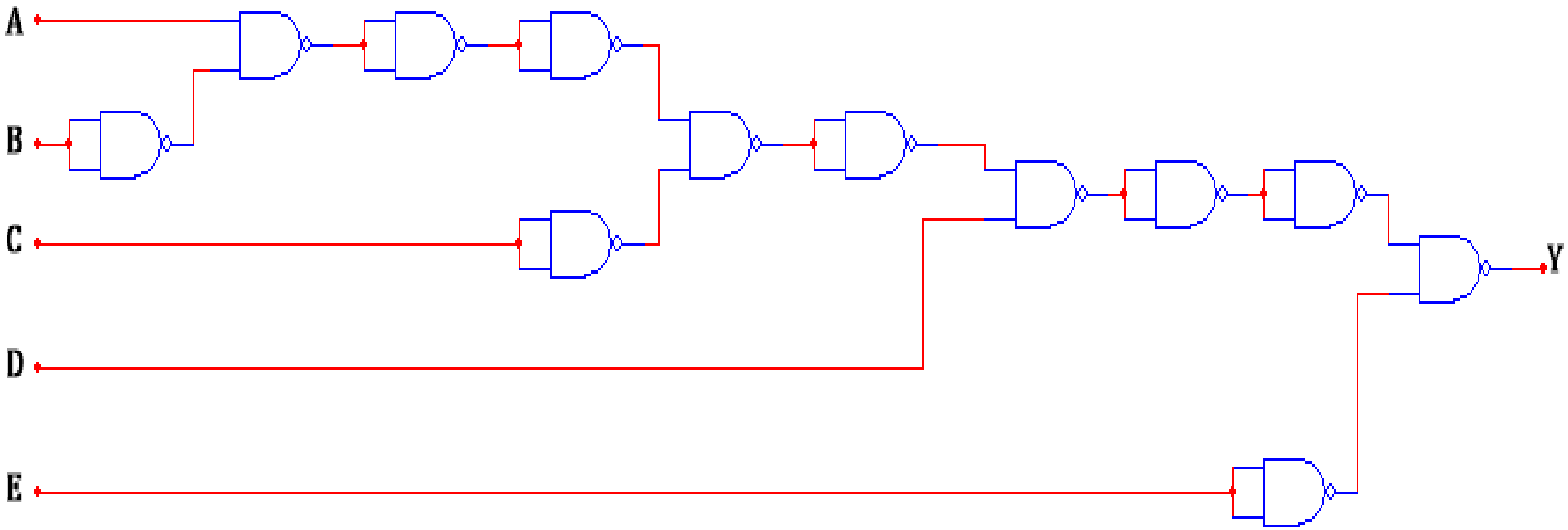
Solution:

Implementation using NAND gates ONLY:

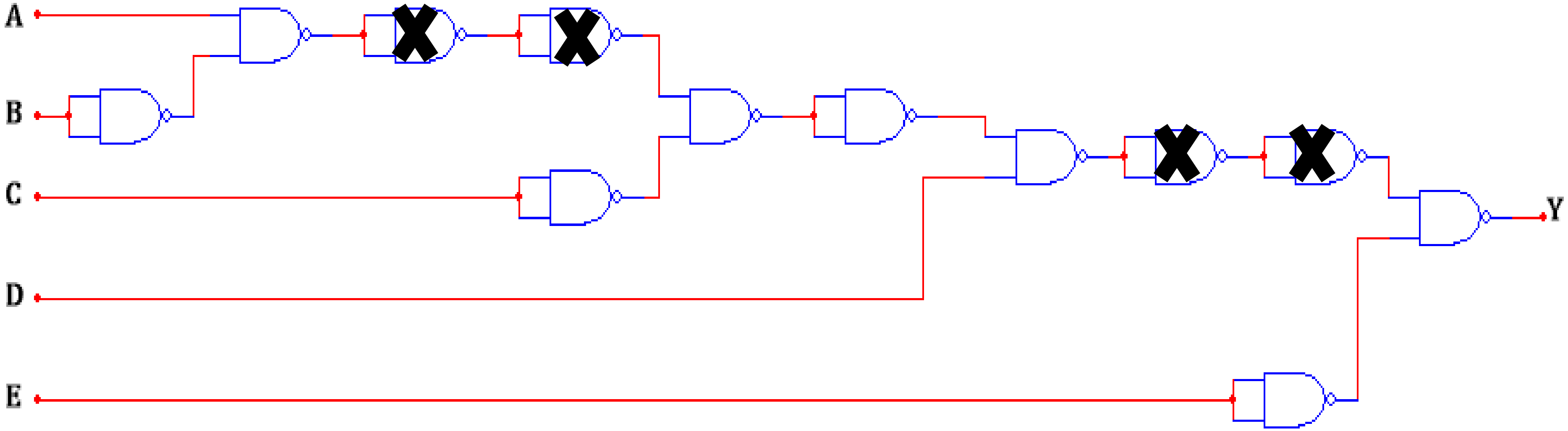
i) Implementation with Basic gates:



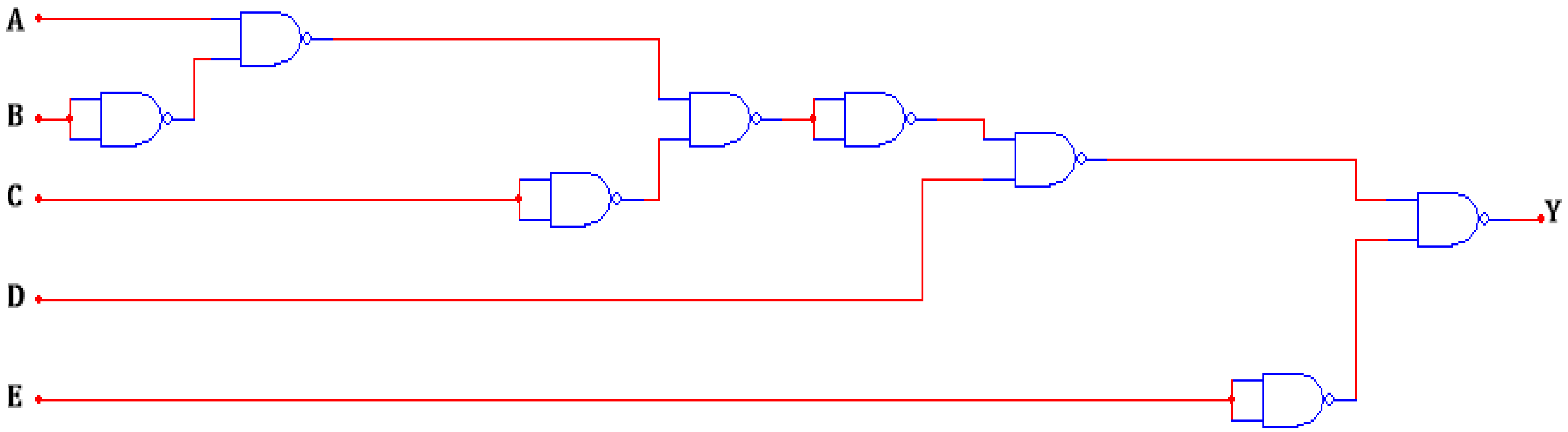
(ii) Replace each basic gate with its equivalent NAND gate implementation from Table 1.



(iii) Cancel two consecutive NOT equivalent gates (according to Boolean algebra).

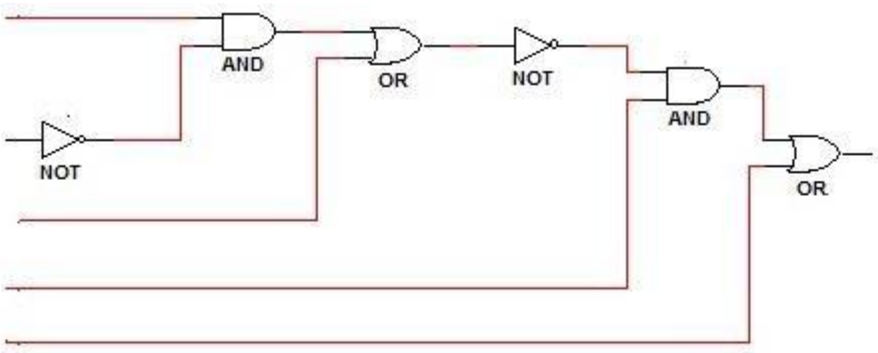


(iv)Redraw the final circuit

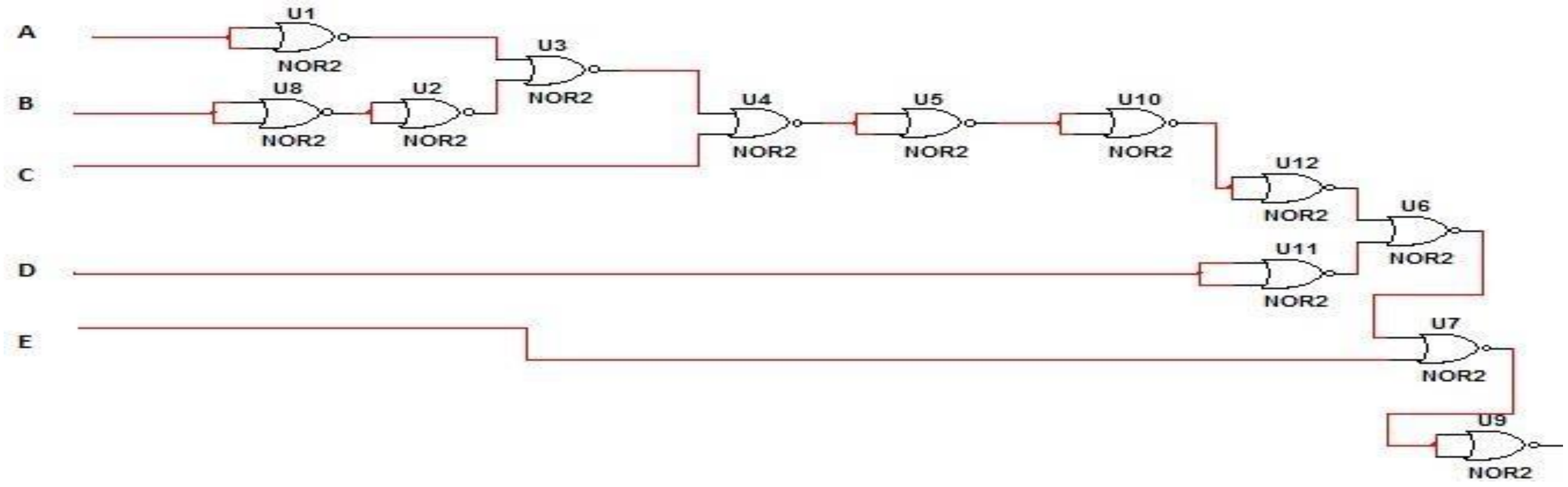


(b) Implementation using NOR gates ONLY:

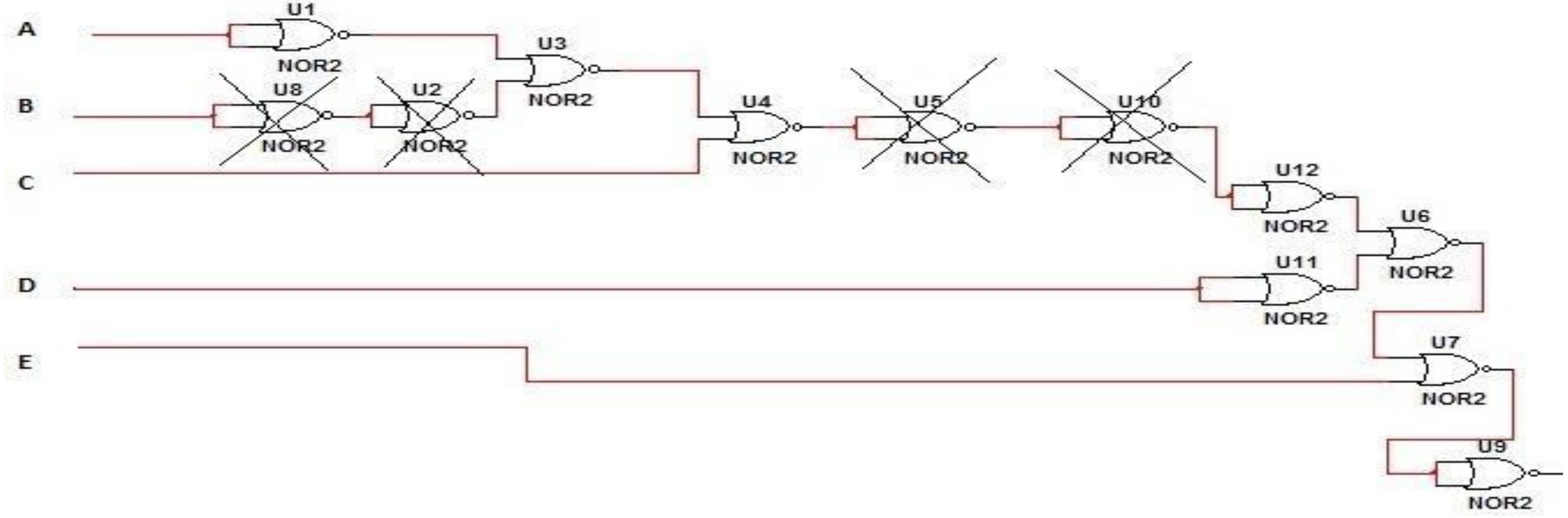
(i) Implementation with Basic gates:



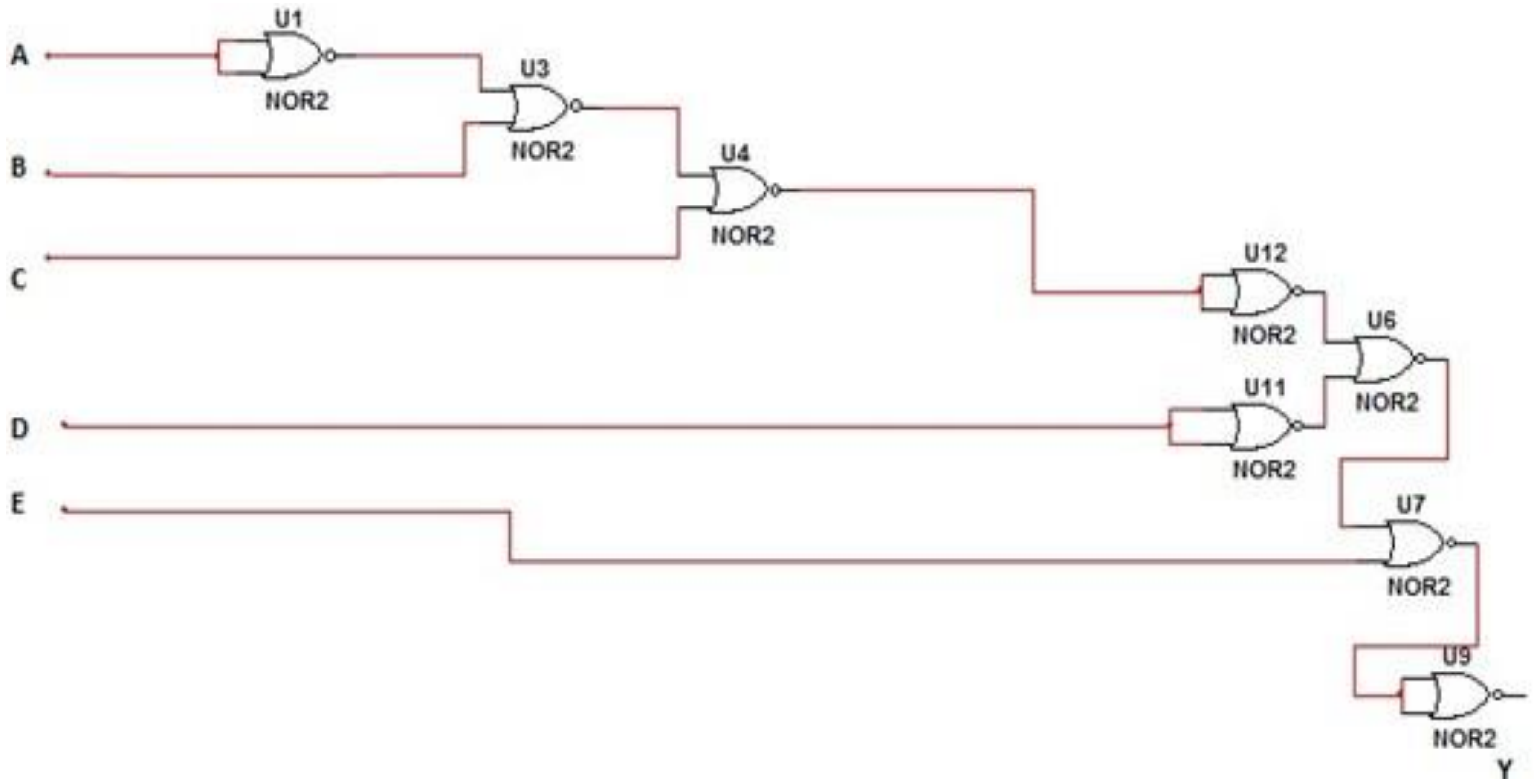
(ii) Replace each basic gate with its equivalent NOR gate implementation from Table 1.



(iii) Cancel two consecutive NOT equivalent gates (according to Boolean algebra).



(iv)Redraw the final circuit.



STANDARD FORM OF BOOLEAN EXPRESSION

- Sum of Products: A Product is defined as a term consisting of products of the literals. When two or more products are summed in a Boolean expression, it is called the Sum-of-Products (SOP).

$$\mathbf{A + AB + B\overline{C}}$$

- Standard SOP: It is an expression where all the variables are present in each product terms. Each of the products are called min terms $(\mathbf{m_i})$.

$$\mathbf{AB\overline{C} + \overline{AB}C + A\overline{B}C}$$

- Product of Sum: A Sum is defined as a term consisting of sum of the literals. When two or more sum terms are multiplied in a Boolean Expression, it is called the Product-of- Sum (POS).

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

- Standard POS: It is an expression where all the variables are present in each sum terms. Each sum terms are called max terms $(\mathbf{M_i})$.

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

Minterms and Maxterms

- Each variable in a Boolean expression is a literal
- Boolean variables can appear in normal (x) or complement form (x')
- Each AND combination of terms is a <u>minterm</u>
- Each OR combination of terms is a <u>maxterm</u>

For example:
Minterms

x	y	z	Minterm	
0	0	0	x'y'z'	m_0
0	0	1	x'y'z	m_1
…				
1	0	0	xy'z'	m_4
…				
1	1	1	xyz	m_7

For example:
Maxterms

x	y	z	Maxterm	
0	0	0	x+y+z	M_0
0	0	1	x+y+z'	M_1
…				
1	0	0	x'+y+z	M_4
…				
1	1	1	x'+y'+z'	M_7

SOP & POS

- Conversion of SOP to Standard SOP
  - Step1: Multiply each of the non-standard terms with a term made up of the sum of the missing variable and its complement. This do not change the function as we are just multiplying by 1.
  - Step2: Repeat step 1 until all the non-standard terms become standard terms.
- Conversion of POS to Standard POS
  - Step1: Add to each non-standard product terms a term made up of the product of the missing variable and its complement. This does not change the expression as we are just adding a 0.
  - Step2: Apply rule 12: $\mathbf{A+BC=(A+B)(A+C)}$
  - Step3: Repeat step 1 until all the sum terms contain all the variable in the domain.

SOP & POS

Convert the following Boolean expression into standard SOP form:

$$A\overline{B}C + \overline{A}\,\overline{B} + AB\overline{C}D$$

Convert the expression $W\overline{X}Y + \overline{X}Y\overline{Z} + WX\overline{Y}$ to standard SOP form.

Convert the following Boolean expression into standard POS form:

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

Convert the expression $(A + \overline{B})(B + C)$ to standard POS form.

The sum-of-product (SOP) form

When two or more product terms are summed by Boolean addition. the resulting expression is a sum-of-products (SOP). Some examples are

$AB + ABC$

$ABC + CDE + \overline{B}C\overline{D}$

$\overline{A}B + \overline{A}B\overline{C} + AC$

Also, an SOP expression can contain a single-variable term, as in $A + \overline{A}\,\overline{B}C + BC\overline{D}$.

an SOP expression can have the term $\overline{A}\,\overline{B}\,\overline{C}$ but not $\overline{ABC}$.

Conversion of a General Expression to SOP Form

$$A(B + CD) = AB + ACD$$

Convert each of the following Boolean expressions to SOP form:

(a) $AB + B(CD + EF)$ **(b)** $(A + B)(B + C + D)$ **(c)** $\overline{(\overline{A + B}) + C}$

Solution **(a)** $AB + B(CD + EF) = AB + BCD + BEF$

(b) $(A + B)(B + C + D) = AB + AC + AD + BB + BC + BD$

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

The product of sum (POS) form

When two or more sum terms are multiplied, the resulting expression is a product-of-sums (POS). Some examples are

$$(\overline{A} + B)(A + \overline{B} + C)$$
$$(\overline{A} + \overline{B} + \overline{C})(C + \overline{D} + E)(\overline{B} + C + D)$$
$$(A + B)(A + \overline{B} + C)(\overline{A} + C)$$

A POS expression can contain a single-variable term, as in $\overline{A}(A + \overline{B} + C)(\overline{B} + \overline{C} + D)$ have the term $\overline{A} + \overline{B} + \overline{C}$ but not $\overline{A + B + C}$.

The Standard SOP Form

A *standard SOP expression* is one in which *all* the variables in the domain appear in each product term in the expression. For example, $A\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + AB\overline{C}\,\overline{D}$ is a standard SOP expression. Standard SOP expressions are important in constructing truth tables,

Converting Product Terms to Standard SOP

a nonstandard SOP expression is converted into standard form using Boolean algebra rule 6 ($A + \overline{A} = 1$) from Table 4–1: A variable added to its complement equals 1.

EXAMPLE

Convert the following Boolean expression into standard SOP form:

$$A\overline{B}C + \overline{A}\,\overline{B} + AB\overline{C}D$$

$$A\overline{B}C = A\overline{B}C(D + \overline{D}) = A\overline{B}CD + A\overline{B}C\overline{D}$$

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

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

$$A\overline{B}C + \overline{A}\,\overline{B} + AB\overline{C}D = A\overline{B}CD + A\overline{B}C\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + \overline{A}\,\overline{B}\,\overline{C}D + \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + AB\overline{C}D$$

Binary Representation of a Standard Product Term

Remember, a product term is implemented with an AND gate whose output is 1 only if each of its inputs is 1. Inverters are used to produce the complements of the variables as required.

$$A\overline{B}C\overline{D} = 1 \cdot \overline{0} \cdot 1 \cdot \overline{0} = 1 \cdot 1 \cdot 1 \cdot 1 = 1$$

The Standard POS Form

A standard POS expression is one in which *all* the variables in the domain appear in each sum term in the expression. For example,

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

Converting a Sum Term to Standard POS : Add $(A \cdot \overline{A} = 0)$ to each nonstandard term.

EXAMPLE

Convert the following Boolean expression into standard POS form:

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

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

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

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

Binary Representation of a Standard Sum Term

is implemented with an OR gate whose output is 0 only if each of its inputs is 0. Inverters are used to produce the complements of the variables as required.

$$A + \overline{B} + C + \overline{D} = 0 + \overline{1} + 0 + \overline{1} = 0 + 0 + 0 + 0 = 0$$

Converting Standard SOP to Standard POS

Convert the following SOP expression to an equivalent POS expression:

$$\overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C + ABC$$

$$\overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C + ABC$$

The evaluation is as follows:

$$000 + 010 + 011 + 101 + 111$$

Since there are three variables in the domain of this expression, there are a total of eight (2^3) possible combinations. The SOP expression contains five of these combinations, so the POS must contain the other three which are 001, 100, and 110

Remember, these are the binary values that make the sum term 0. The equivalent POS expression is

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

BOOLEAN EXPRESSIONS AND TRUTH TABLES

All standard Boolean expressions can be easily converted into truth table format using binary values for each term in the expression. The truth table is a common way of presenting, in a concise format, the logical operation of a circuit. Also. standard SOP or POS expressions can be determined from a truth table. You will find truth tables in data sheets and other literature related to the operation of digital circuits.

Converting SOP Expressions to Truth Table Format

EXAMPLE

Develop a truth table for the standard SOP expression $\overline{A}\,\overline{B}C + A\overline{B}\,\overline{C} + ABC$.

INPUTS			OUTPUT	
A	*B*	*C*	*X*	PRODUCT TERM
0	0	0	0	
0	0	1	1	$\overline{A}\,\overline{B}C$
0	1	0	0	
0	1	1	0	
1	0	0	1	$A\overline{B}\,\overline{C}$
1	0	1	0	
1	1	0	0	
1	1	1	1	ABC

Converting POS Expressions to Truth Table Format

EXAMPLE

Determine the truth table for the following standard POS expression:

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

INPUTS A	B	C	OUTPUT X	SUM TERM
0	0	0	0	$(A + B + C)$
0	0	1	1	
0	1	0	0	$(A + \overline{B} + C)$
0	1	1	0	$(A + \overline{B} + \overline{C})$
1	0	0	1	
1	0	1	0	$(\overline{A} + B + \overline{C})$
1	1	0	0	$(\overline{A} + \overline{B} + C)$
1	1	1	1	

Determining Standard Expressions from a Truth Table

EXAMPLE

From the truth table in Table 4–8, determine the standard SOP expression and the equivalent standard POS expression.

INPUTS			OUTPUT
A	*B*	*C*	*X*
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

$011 \longrightarrow \overline{A}BC$
$100 \longrightarrow A\overline{B}\,\overline{C}$
$110 \longrightarrow AB\overline{C}$
$111 \longrightarrow ABC$

$000 \longrightarrow A + B + C$
$001 \longrightarrow A + B + \overline{C}$
$010 \longrightarrow A + \overline{B} + C$
$101 \longrightarrow \overline{A} + B + \overline{C}$

SOP expression for the output X is

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

The resulting standard POS expression for the output X is

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

Textbooks:

- [1] Thomas L. Floyd, “Digital Fundamentals” 11th edition, Prentice Hall.
- [2] M. Morris Mano, “Digital Logic & Computer Design” Prentice Hall.