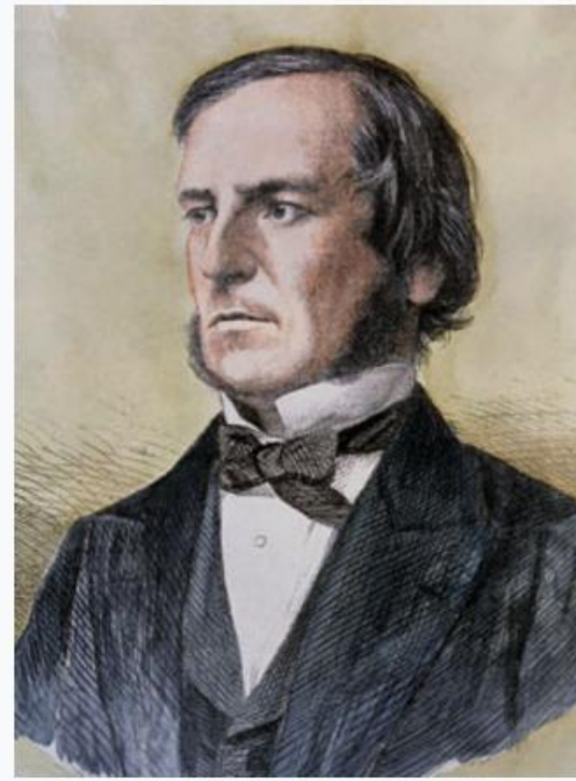
Boolean and POS/SOP

Dr. Noori Kim

Boolean algebra

- **George Boole** (/ˈbuːl/; 2 November 1815 – 8 December 1864)
- An English mathematician, educator, philosopher and logician.
- Worked in the fields of differential equations and algebraic logic
- Best known as **the author** of The Laws of Thought (1854) which contains **Boolean algebra**.

George Boole



Boole in about 1860

Boolean Axioms

Number	Axiom	Dual	Name
A1	$B = 0$ if $B \neq 1$	$B = 1$ if $B \neq 0$	Binary Field
A2	$\overline{0} = 1$	$\overline{1} = 0$	NOT
A3	$0 \bullet 0 = 0$	$1 + 1 = 1$	AND/OR
A4	$1 \bullet 1 = 1$	$0 + 0 = 0$	AND/OR
A5	$0 \bullet 1 = 1 \bullet 0 = 0$	$1 + 0 = 0 + 1 = 1$	AND/OR

Dual: Replace: $\bullet$ with $+$
0 with 1

***Do note that Axiom and Dual are not equivalent, but each of them is true.**

Boolean Theorems of One Variable

Number	Theorem	Dual	Name
T1	$B \bullet 1 = B$	$B + 0 = B$	Identity
T2	$B \bullet 0 = 0$	$B + 1 = 1$	Null Element
T3	$B \bullet B = B$	$B + B = B$	Idempotency
T4	$\overline{\overline{B}} = B$		Involution
T5	$B \bullet \overline{B} = 0$	$B + \overline{B} = 1$	Complements

Dual: Replace: $\bullet$ with +
0 with 1

Boolean Theorems of Several Vars

#	Theorem	Dual	Name
T6	$B \bullet C = C \bullet B$	$B+C = C+B$	Commutativity
T7	$(B \bullet C) \bullet D = B \bullet (C \bullet D)$	$(B + C) + D = B + (C + D)$	Associativity
T8	$B \bullet (C + D) = (B \bullet C) + (B \bullet D)$	$B + (C \bullet D) = (B+C)\ (B+D)$	Distributivity
T9	$B \bullet (B+C) = B$	$B + (B \bullet C) = B$	Covering
T10	$(B \bullet C) + (B \bullet \overline{C}) = B$	$(B+C) \bullet (B+\overline{C}) = B$	Combining
T11	$(B \bullet C) + (\overline{B} \bullet D) + (C \bullet D) =$ $(B \bullet C) + (\overline{B} \bullet D)$	$(B+C) \bullet (\overline{B}+D) \bullet (C+D) =$ $(B+C) \bullet (\overline{B}+D)$	Consensus

Axioms and theorems are useful for *simplifying* equations.

De Morgan's laws

In electrical and computer engineering, De Morgan's laws are commonly written as:

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

and

$$\overline{A + B} \equiv \overline{A} \cdot \overline{B},$$

where:

- $\cdot$ is a logical AND,
- $+$ is a logical OR,
- the $\overline{\text{overbar}}$ is the logical NOT of what is underneath the overbar.

Augustus De Morgan
1806–1871

Simplifying an Equation

Reducing an equation to the **fewest number of implicants**, where each implicant has the **fewest literals**

- Implicant: product of literals

  $A \bullet \overline{B} \bullet C$, $\overline{A} \bullet \overline{C}$, $B \bullet C$

- Literal: variable or its complement

  A, $\overline{A}$, B, $\overline{B}$, C, $\overline{C}$

Simplifying Boolean Equations

Example 1:

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

$Y = A$ T10: Combining

or

$= A \bullet (B + \overline{B})$ T8: Distributivity

$= A \bullet (1)$ T5': Complements

$= A$ T1: Identity

Simplifying Boolean Equations

Example 2:

$Y = A \bullet B' \bullet C + A \bullet B \bullet C + A' \bullet B \bullet C$

$= A \bullet B' \bullet C + \mathbf{A \bullet B \bullet C + A \bullet B \bullet C} + A' \bullet B \bullet C$ T3’: Idempotency

$= (A \bullet B' \bullet C + A \bullet B \bullet C) + (A \bullet B \bullet C + A' \bullet B \bullet C)$ T7’: Associativity

$= A \bullet C + B \bullet C$ T10: Combining

Simplifying Boolean Equations

Example 3:

$Y = (A + B \bullet C)(A + D \bullet E)$

Apply T8' first when possible: $W+X \bullet Z = (W+X) \bullet (W+Z)$

Make: $X = B \bullet C$, $Z = D \bullet E$ and rewrite equation

Y	$= (A+X) \bullet (A+Z)$	substitution ($X=B \bullet C$, $Z=D \bullet E$)
	$= A + X \bullet Z$	T8': Distributivity
	$= A + B \bullet C \bullet D \bullet E$	substitution

or

Y	$= A \bullet A + A \bullet D \bullet E + A \bullet B \bullet C + B \bullet C \bullet D \bullet E$	T8: Distributivity
	$= \mathbf{A + A \bullet D \bullet E} + A \bullet B \bullet C + B \bullet C \bullet D \bullet E$	T3: Idempotency
	$= \mathbf{A} \qquad\qquad + A \bullet B \bullet C + B \bullet C \bullet D \bullet E$	T9': Covering
	$= A + B \bullet C \bullet D \bullet E$	T9': Covering

This is called ***multiplying out***
an expression to getsum-of-products (SOP) form.

A sidebar: a common mistake

$$\bar{A} \cdot \bar{B} + A \cdot B = ??$$

$$\bar{A}\bar{B} + AB = ??$$

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

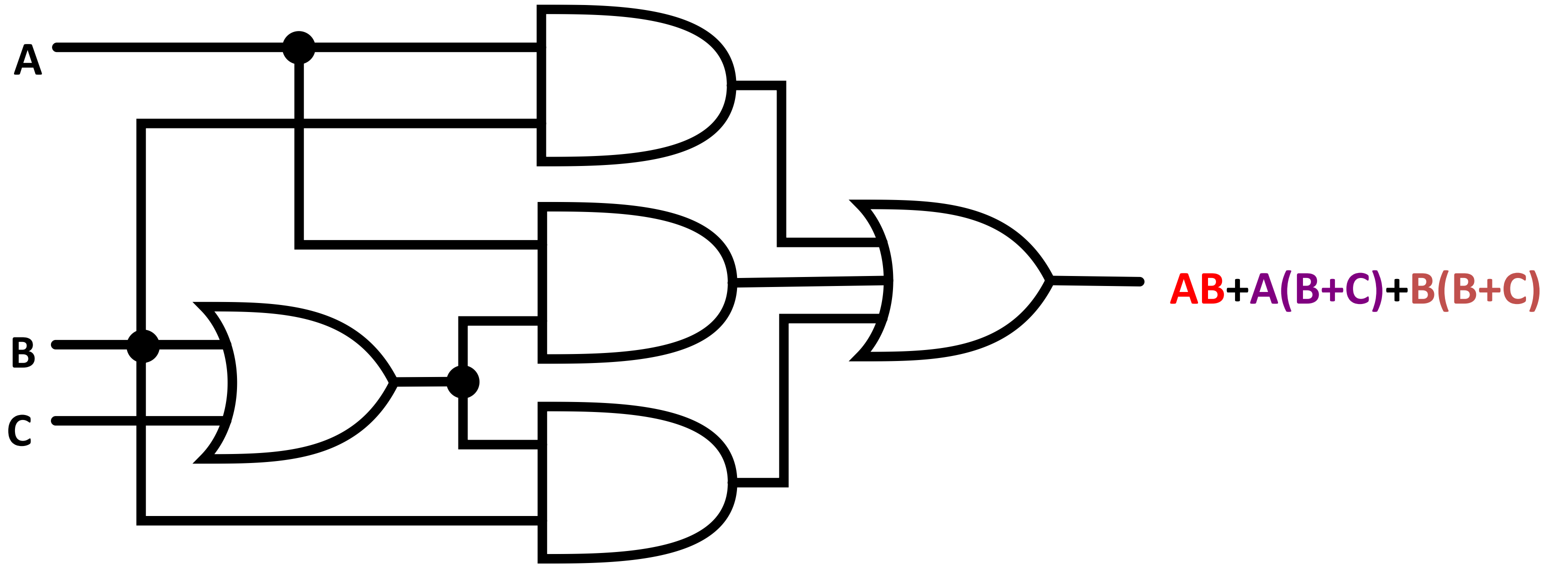
Are they same?

$$\overline{A \cdot B} \; ??? \; \bar{A} \cdot \bar{B}$$

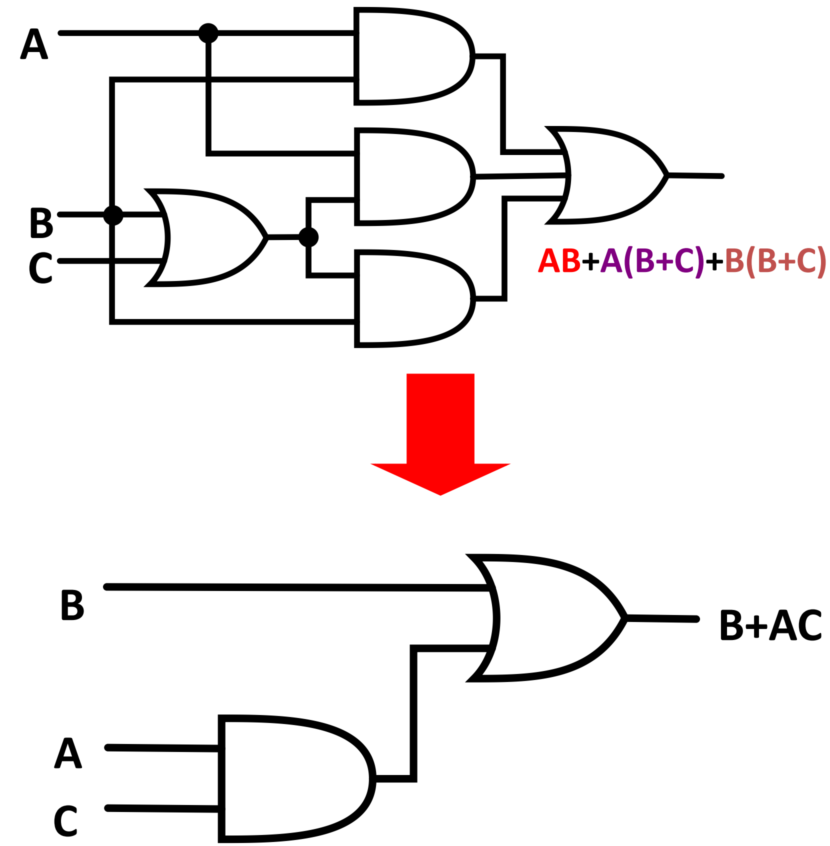
$$\overline{A \cdot B} \neq \bar{A} \cdot \bar{B}$$

Boolean AND Logic gates

- A simplified Boolean expression uses the fewest gates possible to implement a given expression.



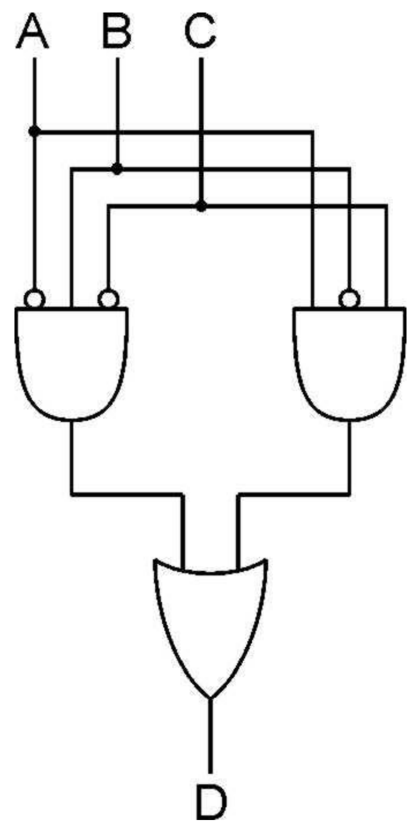
- A•B+A• (B+C)+B• (B+C)
  - (distributive law)
    - AB+AB+AC+BB+BC
  - (BB=B)
    - AB+AB+AC+**B**+BC
  - (AB+AB=AB)
    - **AB**+AC+B+BC
  - (B+BC=B)
    - AB+AC+**B**
  - (AB+B=B)
    - **B**+AC



POS/SOP

- Can implement ANY truth table with AND, OR, NOT.

A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0



1. AND combinations that yield a "1" in the truth table.

2. OR the results of the AND gates.

Standard Forms of Boolean Expressions

- All Boolean expressions, regardless of their form, can be converted into either of two standard forms:
  - The sum-of-products (SOP) form
  - The product-of-sums (POS) form

Sum-of-Products (SOP)

The Sum-of-Products (SOP) Form

- Two or more product terms are summed by Boolean addition.
  - Examples:

    $$AB + ABC$$

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

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

  - Also:

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

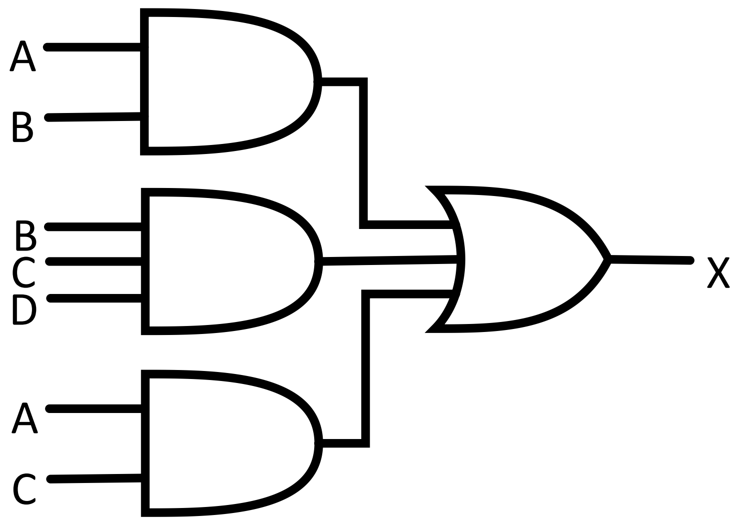
- In an SOP form, <u>a single overbar cannot extend over more than one variable</u>; however, more than one variable <u>in a term</u> can have an overbar:
  - example: $\overline{A}\,\overline{B}\,\overline{C}$ is OK!
  - But not: $\overline{ABC}$

Boolean OK

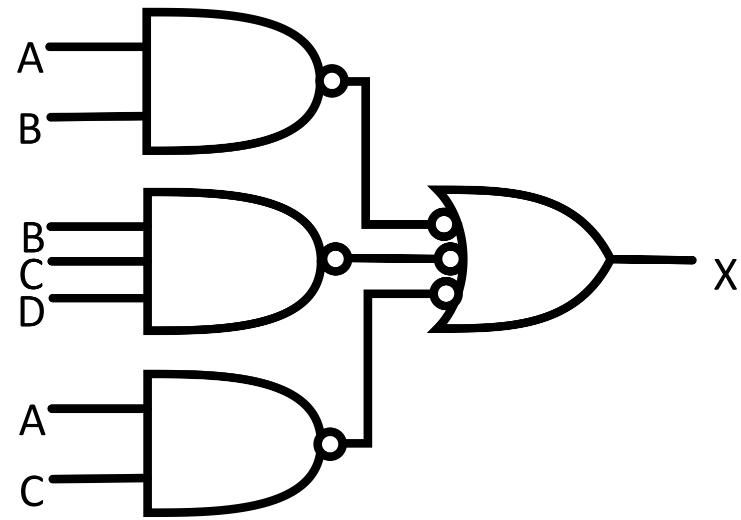
Implementation of an SOP

$$X = AB + BCD + AC$$

- AND/OR implementation



- NAND/NAND implementation



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.
  - Example: $$A\overline{B}CD + \overline{A}\,\overline{B}C\overline{D} + AB\overline{C}\,\overline{D}$$
- Standard SOP expressions are important in:
  - Constructing truth tables
  - The Karnaugh map simplification method

Converting Product Terms to Standard SOP

- Convert the following Boolean expression into standard SOP form:

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

Product-of-Sums (POS)

The Product-of-Sums (POS) Form

- Two or more sum terms are multiplied:
  - Examples:

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

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

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

  - Also:

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

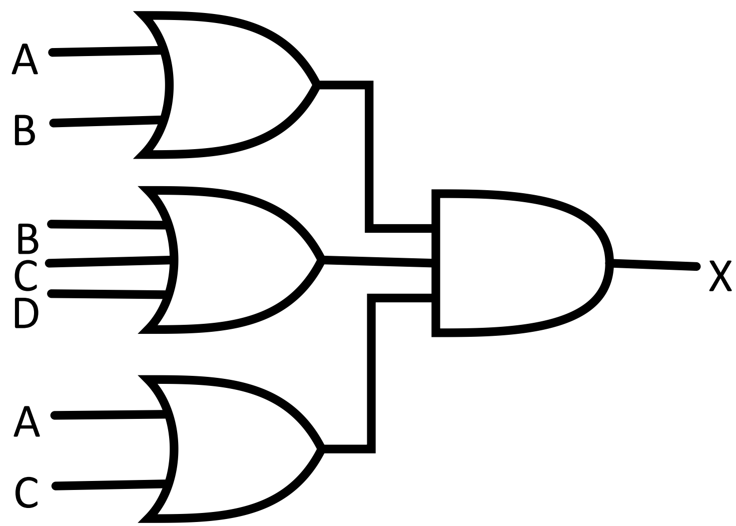
- In a POS form, <u>a single overbar cannot extend over more than one variable</u>; however, more than one variable <u>in a term</u> can have an overbar:
  - example: $\overline{A}+\overline{B}+\overline{C}$ is OK!
  - But not: $\overline{A+B+C}$

Boolean OK

Implementation of a POS

$$X=(A+B)(B+C+D)(A+C)$$

- OR/AND implementation



The Standard POS Form

- A **<u>standard</u>** POS expression is one in which *all* the variables in the domain appear <u>in each sum term</u> in the expression.
  - Example: $(\overline{A}+\overline{B}+\overline{C}+\overline{D})(A+\overline{B}+C+D)(A+B+\overline{C}+D)$

- Standard POS expressions are important in:
  - Constructing truth tables
  - The Karnaugh map simplification method

Converting a Sum Term to Standard POS (example)

- Convert the following Boolean expression into standard POS form:

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

SOP/POS conversion

Converting SOP Expressions to Truth Table Format

- Develop a truth table for the standard SOP expression

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

Inputs			Output	Product Term
A	B	C	X	
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

- Develop a truth table for the standard SOP expression

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

Inputs			Output	Sum Term
A	B	C	X	
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 Expression from a Truth Table (example)

I/P			O/P
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

SOP

- There are <u>four 1s</u> in the output and the corresponding binary value are 011, 100, 110, and 111.

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

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

POS

- There are <u>four 0s</u> in the output and the corresponding binary value are 000, 001, 010, and 101.

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

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

The Karnaugh Map

What is K-Map

- An array of cells in which each cell represents a binary value of the input variables.
- Simplification of a given expression is simply a matter of properly **grouping the cells**.
- K-maps can be used for expressions with 2+ variables: 3 and 4 variables will be discussed to illustrate the principles.

The 3 Variable K-Map

- There are 8 cells as shown:

AB \ C	0	1
00	$\overline{A}\,\overline{B}\,\overline{C}$	$\overline{A}\,\overline{B}C$
01	$\overline{A}B\overline{C}$	$\overline{A}BC$
11	$AB\overline{C}$	ABC
10	$A\overline{B}\,\overline{C}$	$A\overline{B}C$

Grey Code

Decimal	Binary	Gray	Gray as decimal
0	000	000	0
1	001	001	1
2	010	011	3
3	011	010	2
4	100	110	6
5	101	111	7
6	110	101	5
7	111	100	4

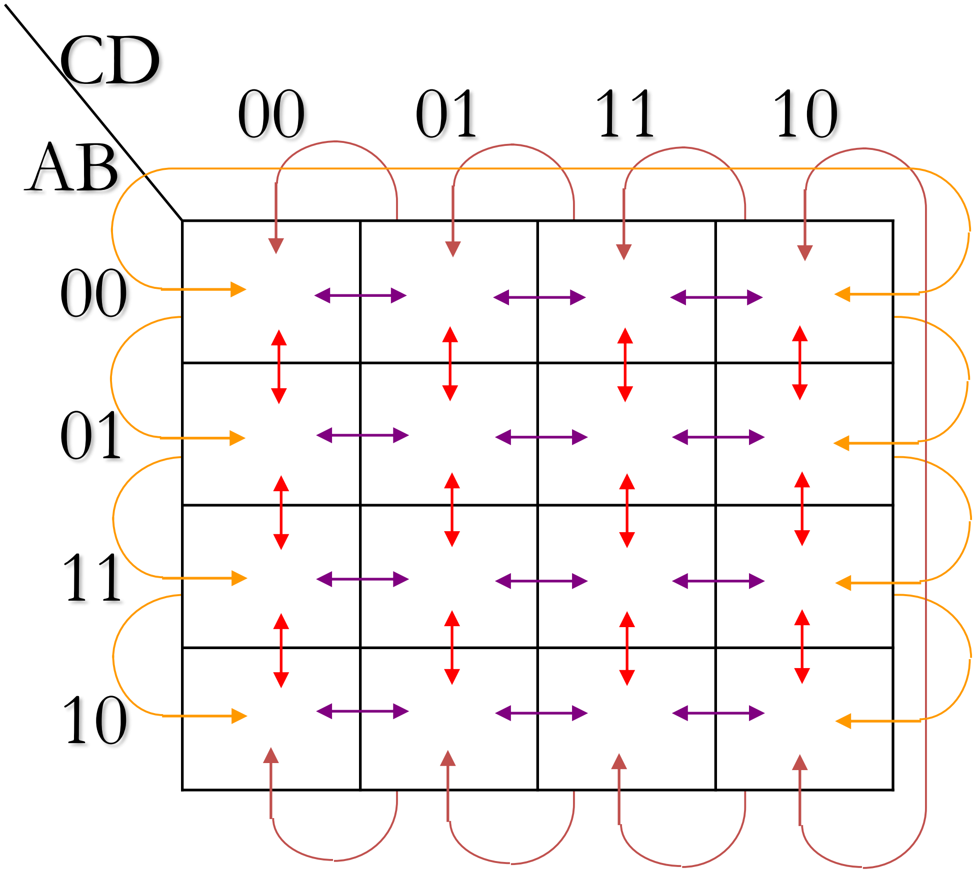
Gray code by bit width

2-bit	4-bit
00	0000
01	0001
11	0011
10	0010
	0110
	0111
3-bit	0101
000	0100
001	1100
011	1101
010	1111
110	1110
111	1010
101	1011
100	1001
	1000

The 4-Variable K-Map

AB / CD	00	01	11	10
00	$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D}$	$\overline{A}\,\overline{B}\,\overline{C}D$	$\overline{A}\,\overline{B}CD$	$\overline{A}\,\overline{B}C\overline{D}$
01	$\overline{A}B\overline{C}\,\overline{D}$	$\overline{A}B\overline{C}D$	$\overline{A}BCD$	$\overline{A}BC\overline{D}$
11	$AB\overline{C}\,\overline{D}$	$AB\overline{C}D$	$ABCD$	$ABC\overline{D}$
10	$A\overline{B}\,\overline{C}\,\overline{D}$	$A\overline{B}\,\overline{C}D$	$A\overline{B}CD$	$A\overline{B}C\overline{D}$

Cell Adjacency



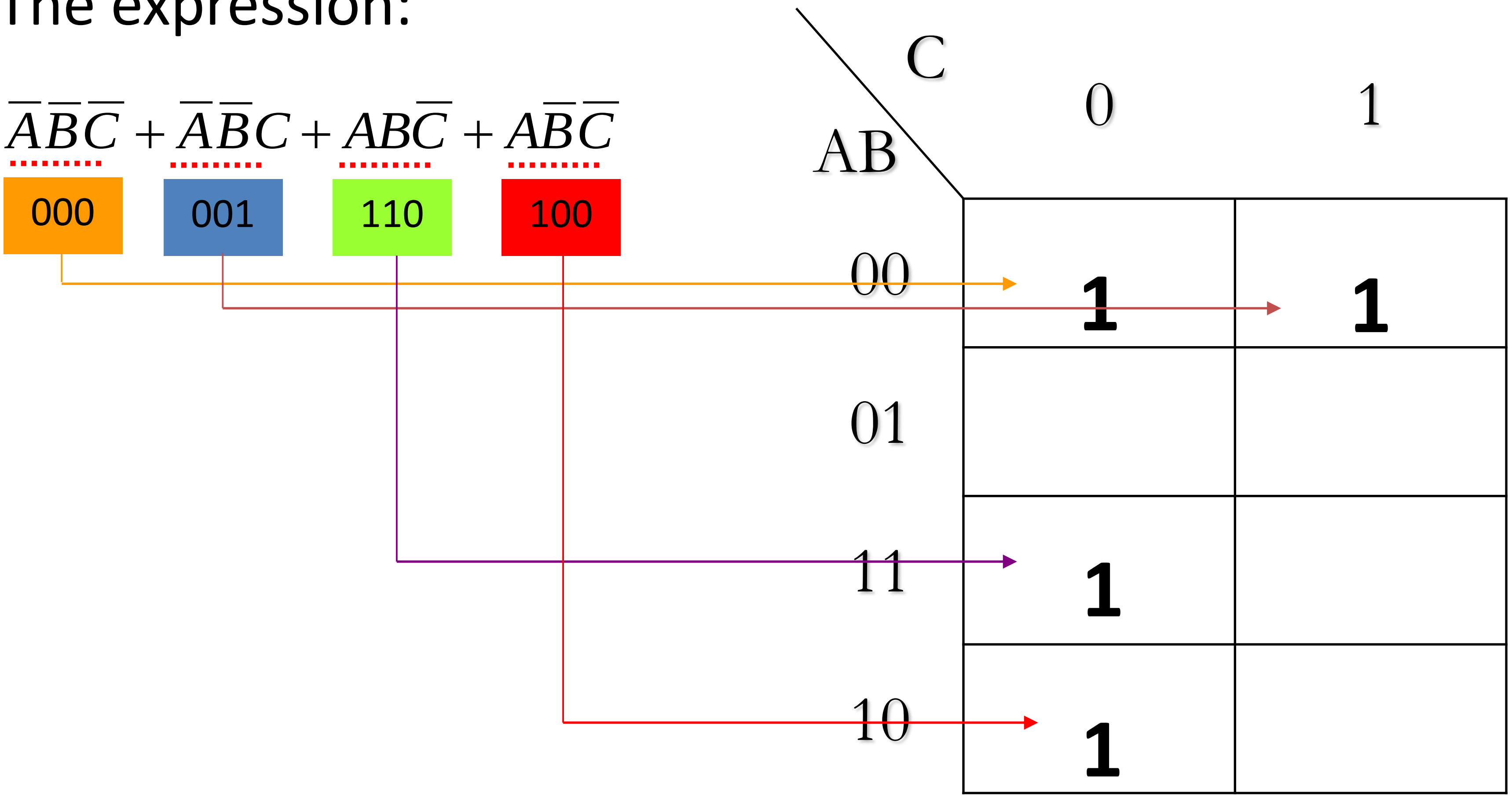
K-Map SOP Minimization

- The K-Map is used for simplifying Boolean expressions to their minimal form.
- A minimized SOP expression contains the fewest possible terms with fewest possible variables per term.
- Generally, a minimum SOP expression can be implemented with fewer logic gates than a standard expression.

Mapping a Standard SOP Expression (full example)

The expression:

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



Practice $\overline{A}\,\overline{B}C + \overline{A}B\overline{C} + AB\overline{C} + ABC$

AB \ C	0	1
00		
01		
11		
10		

Practice

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

AB \ C	0	1
00		
01		
11		
10		

K-Map Simplification of SOP Expressions

- After an SOP expression has been mapped, we can do the process of *minimization:*
  - Grouping the 1s
  - Determining the minimum SOP expression from the map

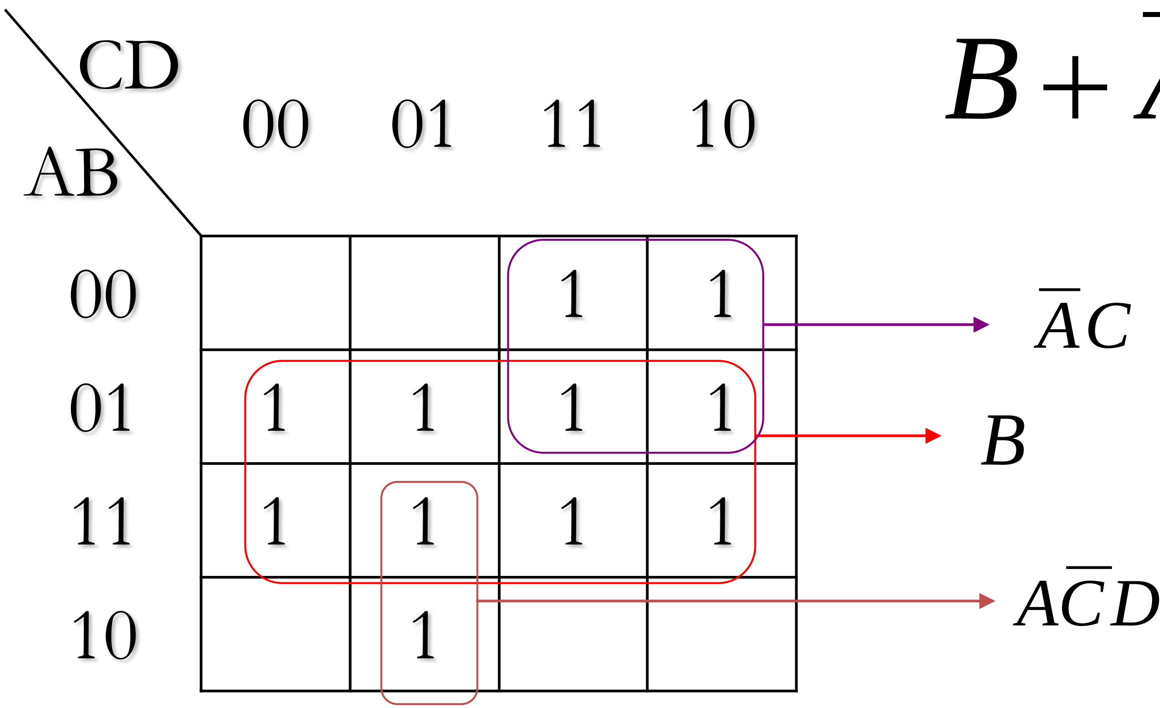
Grouping the 1s

AB \ C	0	1
00	1	
01		1
11	1	1
10		

AB \ C	0	1
00	1	1
01	1	
11		1
10	1	1

Determining the Minimum SOP Expression from the Map

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



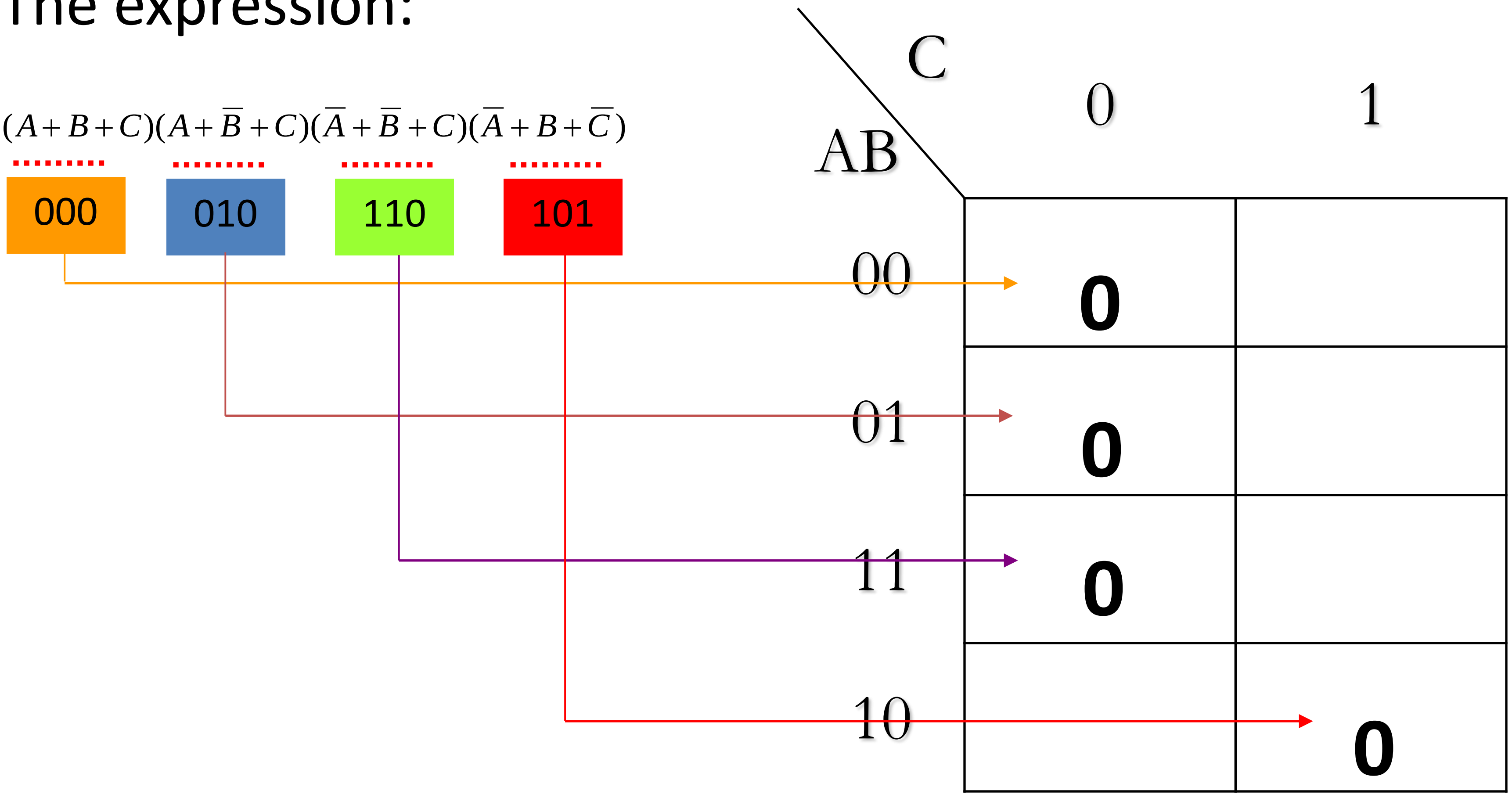
K-Map **POS** Minimization

- The approaches are similar to SOP except that
  - with **POS expression**, 0s representing the standard sum terms placed on the K-map.

Mapping a Standard POS

The expression:

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



Mapping a Standard POS/SOP

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

POS: 000,001,010,011,110 → missing 100,111,101 : SOP

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

AB \ C	0	1
00	0	0
01	0	0
11	0	1
10	1	1

A

$\overline{B}+C$

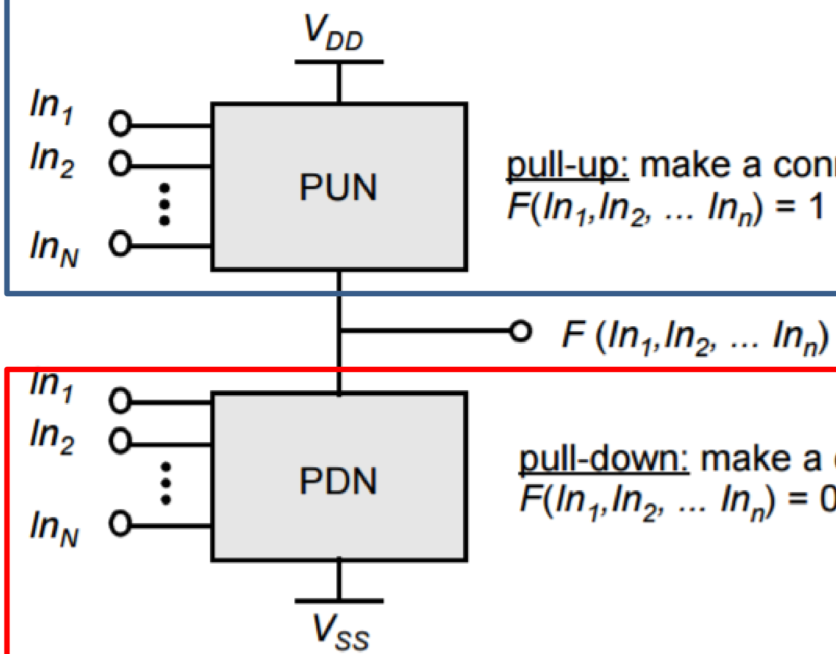
$A(\overline{B}+C)$

$A\overline{B}+AC$

AC

$A\overline{B}$

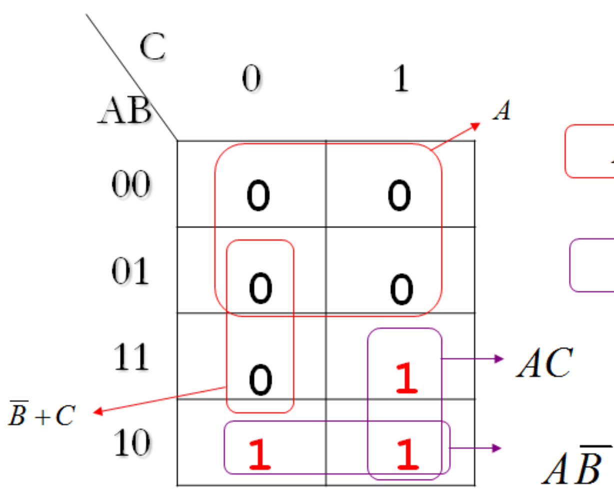
$A\overline{B}\,\overline{C}+ABC+A\overline{B}C$



pull-up: make a connection from V_{DD} to F when $F(In_1, In_2, \ldots In_n) = 1$

pull-down: make a con

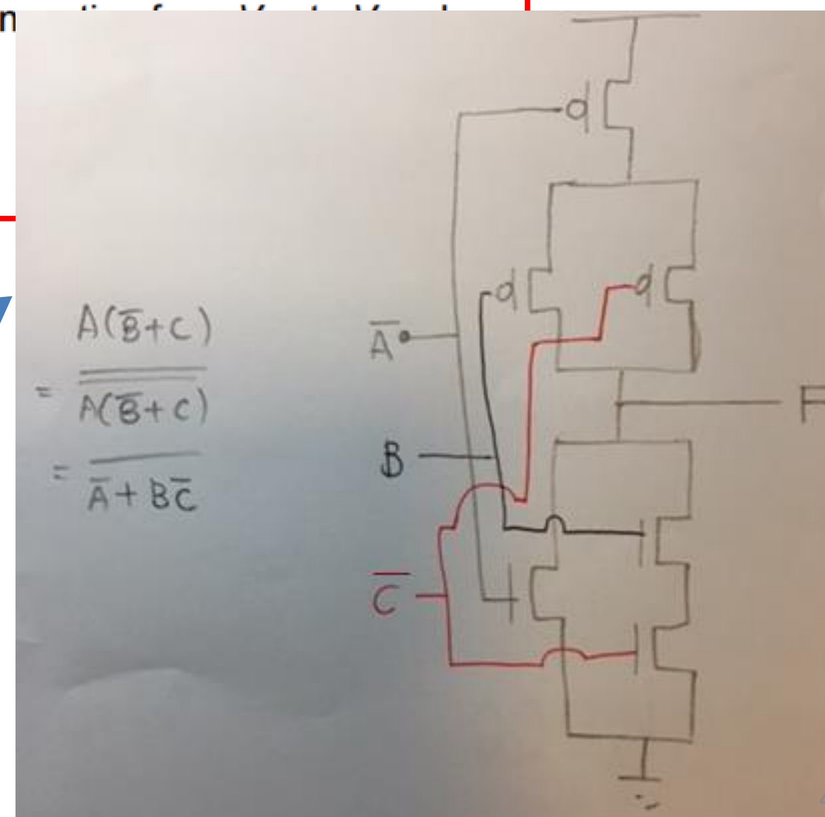
$F(In_1, In_2, \ldots In_n) = 0$



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

=

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



Summary: Standard vs. Non-standard?

1. Non standard SOP

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

2. Convert 1 to Standard SOP

AB'+AC = AB'(C+C')+AC(B+B')
=AB'C+AB'C'+ABC+AB'C
=AB'C'+ABC+AB'C

3. Standard SOP

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

1. Non standard POS

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

2. Convert 1 to Standard POS

A(B'+C)=A(B'+C +A'A)=A(B'+C+A')(B'+C+A)
=(A+BB')(B'+C+A')(B'+C+A)
=(A+B)(A+B')(B'+C+A')(B'+C+A)
=(A+B+CC')(A+B'+CC')(B'+C+A')(B'+C+A)
=(A+B+C)(A+B'+C)(A+B+C')(A+B'+C')(B'+C+A')(B'+C+A)
=(A+B+C)(A+B+C') (A+B'+C)(A+B'+C')(A'+B'+C)

3. Standard POS

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