ECE 2050 Digital Logic and Systems

Chapter 4 : Boolean Algebra and Logic Simplification

Instructor: Tinghuan Chen, Ph.D.

香港中文大學（深圳）
The Chinese University of Hong Kong, Shenzhen

Last Week

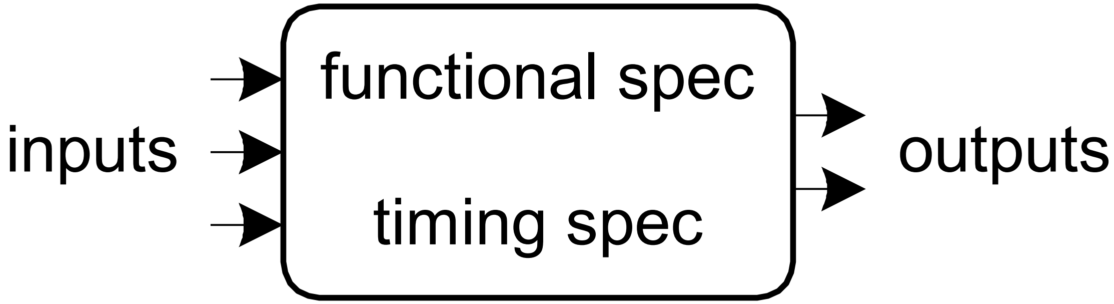
- Logic gates
  - Inverter, AND, OR, NAND, NOR, XOR, XNOR
  - Truth Table
  - Timing diagram
  - Logic expression
  - Distinctive Shape Symbols
- Logic Levels
  - Logic levels
  - Noise Margins

Combinational Circuits

Introduction

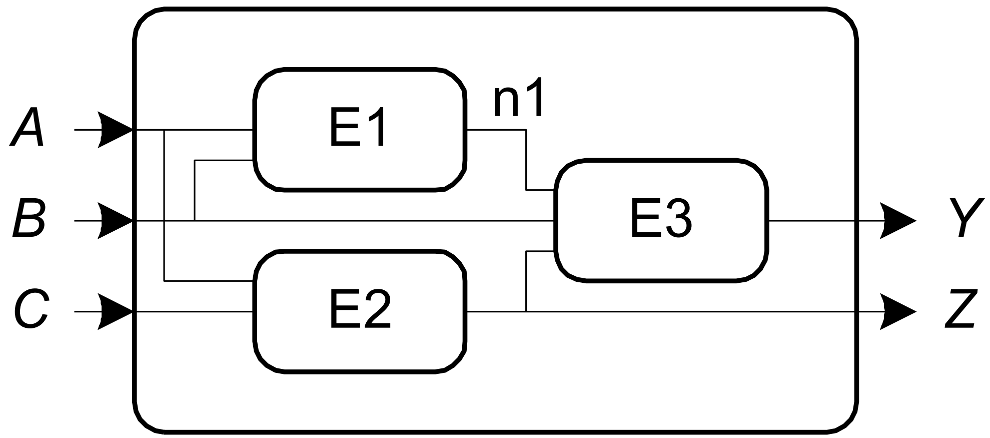
A logic circuit is composed of:

- Inputs
- Outputs
- Functional specification
- Timing specification



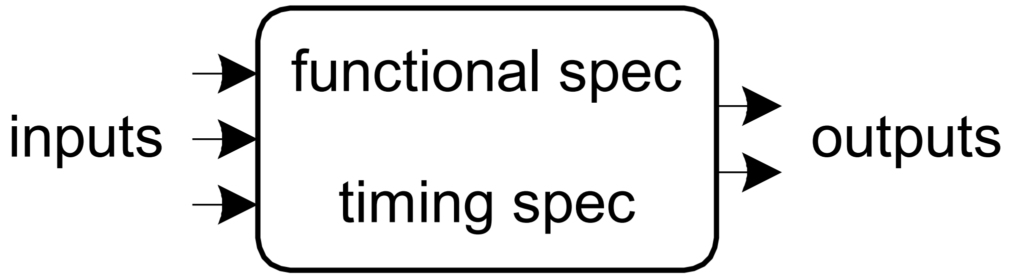
Circuits

- **Nodes**
  - Inputs: *A*, *B*, *C*
  - Outputs: *Y*, *Z*
  - Internal: n1
- **Circuit elements**
  - E1, E2, E3
  - Each a circuit



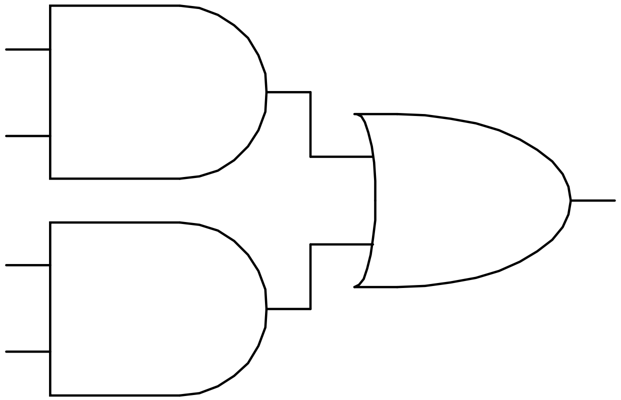
Types of Logic Circuits

- **Combinational Logic: Chapter 5**
  - Memory*less*
  - Outputs determined by current values of inputs
- **Sequential Logic: Chapter 6**
  - Has memory
  - Outputs determined by previous and current values of inputs



Rules of Combinational Composition

- Every element is combinational
- Every node is either an input or connects to *exactly one* output
- The circuit contains no cyclic paths
- **Example:**



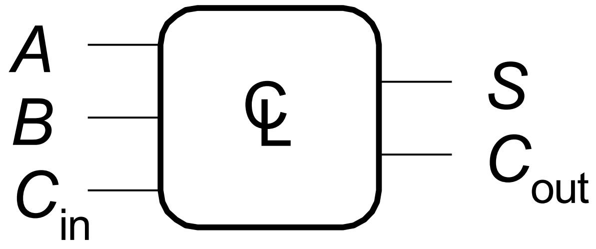
Boolean Equations

Boolean Equations

- Functional specification of outputs in terms of inputs

- **Example:** $S = F(A, B, C_{in})$

  $C_{out} = F(A, B, C_{in})$



$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = AB + AC_{in} + BC_{in}$$

Forming Boolean Expressions

Example 1:

We will go to the Park (*P* is the output) if it’s not Raining ($\overline{R}$) and we have Sandwiches (*S*).

Boolean Equation:

$P = \overline{R}S$

Forming Boolean Expressions

Example 2:

You will be considered a Winner (**W** is the output) if we send you a Million dollars (**M**) or a small Notepad (**N**).

Boolean Equation:

$W = M + N$

Forming Boolean Expressions

Example 3:

You can Eat delicious food (***E*** is the output) if you Make it yourself (***M***) or you have a personal Chef (***C***) and she/he is talented (***T***) but not eXpensive ($\overline{\boldsymbol{X}}$).

Boolean Equation:

$$E = M + CT\overline{X}$$

Forming Boolean Expressions

Example 4:

You can Enter the building if you have a Hat and Shoes on or if you have a Hat on.

Boolean Equation:

$E = HS + H$

Forming Boolean Expressions

Example 5:

You can Enter the building if you have a Hat and Shoes on or if you have a Hat and no Shoes on.

Boolean Equation:

$$E = HS + H\overline{S}$$

Boolean Algebra: Axioms & Theorems

Boolean Algebra

- Axioms and theorems to **simplify** Boolean equations
- Like regular algebra, but simpler: variables have only two values (1 or 0)
- **Duality** (对偶性) in axioms and theorems:
    - ANDs and ORs, 0's and 1's interchanged

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

Boolean Operations and Expressions

- Boolean Addition : Equivalent to the OR operation

Determine the values of A, B, C, and D that make the sum term $A + \bar{B} + C + \bar{D} = 0$

Solution :

$A = 0, \quad \bar{B} = 0, \quad C = 0, \quad \bar{D} = 0 \quad \Rightarrow \quad A = 0, \quad B = 1, \quad C = 0, \quad D = 1$

- Boolean Multiplication : Equivalent to the AND operation

Determine the values of A, B, C, and D that make the product term $A\bar{B}C\bar{D} = 1$

Solution :

$A = 1, \quad \bar{B} = 1, \quad C = 1, \quad \bar{D} = 1 \quad \Rightarrow \quad A = 1, \quad B = 0, \quad C = 1, \quad D = 0$

Theorems of Boolean Algebra

Laws of Boolean Algebra

- Commutative laws $A + B = B + A$ $AB = BA$
- Associative laws $A + (B + C) = (A + B) + \mathrm{C}$ $A(BC) = (AB)\mathrm{C}$
- Distributive law $A(B + C) = AB + \mathrm{AC}$

Rules of Boolean Algebra

1. $A + 0 = A$	**7.** $A \cdot A = A$
2. $A + 1 = 1$	**8.** $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	**9.** $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	**10.** $A + AB = A$
5. $A + A = A$	**11.** $A + \overline{A}B = A + B$
6. $A + \overline{A} = 1$	**12.** $(A + B)(A + C) = A + BC$

A, B, or C can represent a single variable or a combination of variables.

How to Prove?

- **Method 1:** Perfect induction
- **Method 2:** Use other theorems and axioms to simplify the equation
  - Make one side of the equation look like the other

Proof by Perfect Induction

- Also called: **proof by exhaustion**
- Check every possible input value
- If the two expressions produce the same value for every possible input combination, the expressions are equal

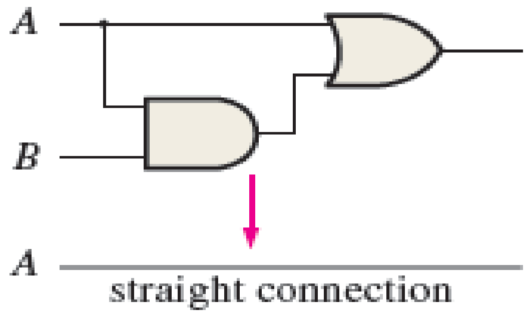
Proof of Rule 10

Number	Theorem
10	A + AB = A

Method 1: Perfect Induction

A	B	AB	$A + AB$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

equal



Proof of Rule 10

Number	Theorem
10	A + AB = A

Method 2: Prove true using other axioms and theorems.

$$
\begin{aligned}
A + AB = A \cdot 1 + AB = A(1 + B) & \quad \text{Factoring (distributive law)} \\
= A \cdot 1 & \quad \text{Rule 2: } (1 + B) = 1 \\
= A & \quad \text{Rule 4: } A \cdot 1 = A
\end{aligned}
$$

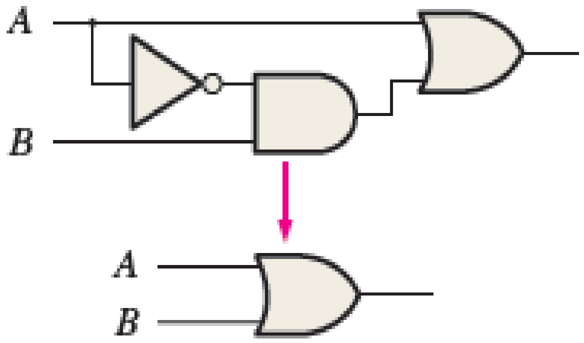
Proof of Rule 11

Number	Theorem
11	$A + \bar{A}B = A + B$

Method 1: Perfect Induction

A	B	$\bar{A}B$	$A + \bar{A}B$	$A + B$
0	0	0	0	0
0	1	1	1	1
1	0	0	1	1
1	1	0	1	1

equal



Proof of Rule 11

Number	Theorem
11	$A + \overline{A}B = A + B$

Method 2: Prove true using other axioms and theorems.

$A + \overline{A}B = (A + AB) + \overline{A}B$	Rule 10: $A = A + AB$
$= (AA + AB) + \overline{A}B$	Rule 7: $A = AA$
$= AA + AB + A\overline{A} + \overline{A}B$	Rule 8: adding $A\overline{A} = 0$
$= (A + \overline{A})(A + B)$	Factoring
$= 1 \cdot (A + B)$	Rule 6: $A + \overline{A} = 1$
$= A + B$	Rule 4: drop the 1

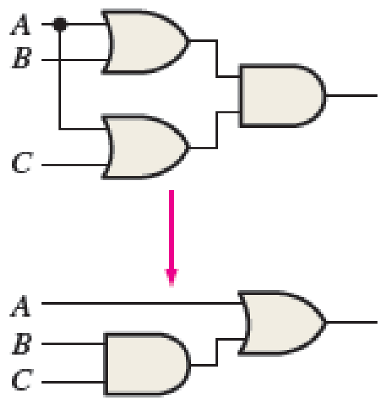
Proof of Rule 12

Number	Theorem
12	(A+B) (A+C) = A + (B•C)

Method 1: Perfect Induction

A	B	C	$A+B$	$A+C$	$(A+B)(A+C)$	BC	$A+BC$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

equal



Proof of Rule 12

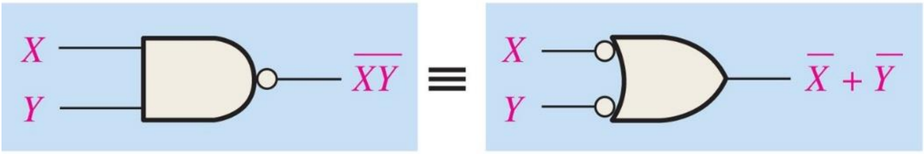
Number	Theorem
12	(A+B) (A+C) = A + (B•C)

Method 2: Prove true using other axioms and theorems.

$(A + B)(A + C) = AA + AC + AB + BC$	Distributive law
$= A + AC + AB + BC$	Rule 7: $AA = A$
$= A(1 + C) + AB + BC$	Factoring (distributive law)
$= A \cdot 1 + AB + BC$	Rule 2: $1 + C = 1$
$= A(1 + B) + BC$	Factoring (distributive law)
$= A \cdot 1 + BC$	Rule 2: $1 + B = 1$
$= A + BC$	Rule 4: $A \cdot 1 = A$

DeMorgan's Theorems

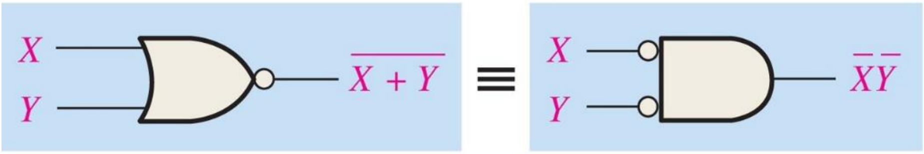
$$\overline{XY} = \bar{X} + \bar{Y}$$



NAND Negative-OR

Inputs		Output	
X	Y	$\overline{XY}$	$\overline{X} + \overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

$$\overline{X + Y} = \bar{X}\bar{Y}$$



NOR Negative-AND

Inputs		Output	
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

DeMorgan's Theorems: Examples

Example

Apply DeMorgan's theorems to each expression:

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

(b) $\overline{(\overline{A} + B) + CD}$

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

Basic rules of Boolean algebra.

1. $A + 0 = A$	**7.** $A \cdot A = A$
2. $A + 1 = 1$	**8.** $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	**9.** $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	**10.** $A + AB = A$
5. $A + A = A$	**11.** $A + \overline{A}B = A + B$
6. $A + \overline{A} = 1$	**12.** $(A + B)(A + C) = A + BC$

Solution

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

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

(c) $\overline{(A + B)\overline{C}\,\overline{D} + E + \overline{F}} = (\overline{(A + B)\overline{C}\,\overline{D}})(\overline{E + \overline{F}}) = (\overline{A}\,\overline{B} + C + D)\overline{E}F$

Boolean Algebra: Simplifying Equations

Logic Simplification Using Boolean Algebra

- Use the axioms and theorems of Boolean algebra to manipulate and simplify an expression.

Example 1:

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

$$Y = (\overline{A} + A)B$$

$$= (1)B$$

$$= B$$

Example 2:

$$Y = A\overline{B}C + ABC + \overline{A}BC$$

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

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

$$= AC + BC$$

Logic Simplification Using Boolean Algebra

Example 3:

$$AB + A(B + C) + B(B + C)$$

$$= \underbrace{AB + AB}_{\text{Rule 5}} + \mathrm{AC} + \underbrace{BB}_{\text{Rule 7}} + BC$$ distributive law

$$= AB + \mathrm{AC} + \underbrace{B + BC}_{\text{Rule 10}}$$

$$= \underbrace{\boxed{AB} + \mathrm{AC} + \boxed{B}}_{\text{Rule 10}}$$

$$= \mathrm{AC} + B$$

1. $A + 0 = A$	**7.** $A \cdot A = A$
2. $A + 1 = 1$	**8.** $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	**9.** $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	**10.** $A + AB = A$
5. $A + A = A$	**11.** $A + \overline{A}B = A + B$
6. $A + \overline{A} = 1$	**12.** $(A + B)(A + C) = A + BC$

Logic Simplification Using Boolean Algebra

Example 4:

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

$$= BC + \overline{B}D + (CDB + CD\overline{B})$$

$$= BC + \overline{B}D + BCD + \overline{B}CD$$

$$= BC + BCD + \overline{B}D + \overline{B}CD$$

$$= (BC + BCD) + (\overline{B}D + \overline{B}CD)$$

$$= BC + \overline{B}D$$

SOP, Standard SOP
POS, Standard POS

Some Definitions

- **Complement**: variable with a bar over it

  $\bar{A}, \bar{B}, \bar{C}$
- **Literal**: variable or its complement

  $A, \bar{A}, B, \bar{B}, C, \bar{C}$
- **Implicant**: product of literals

  $AB\bar{C}, \bar{A}C, BC$
- **Minterm**: product that includes all input variables

  $AB\bar{C}, A\bar{B}\bar{C}, ABC$
- **Maxterm**: sum that includes all input variables

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

Sum-of-Products (SOP) Form

- All **Boolean** equations can be written in SOP form
- Each row has a **minterm**
- A minterm is a **product** (AND) of literals
- Each minterm is **TRUE** for that row (and only that row)
- Form function by **ORing minterms** where output is **1**
- Thus, a sum (OR) of products (AND terms)

A	*B*	*Y*	minterm	minterm name
0	0	0	$\overline{A}\,\overline{B}$	m_0
0	1	1	$\overline{A}\,B$	m_1
1	0	0	$A\,\overline{B}$	m_2
1	1	1	$A\,B$	m_3

$$Y = \mathrm{F}(A, B) = \overline{A}B + AB$$

Product-of-Sums (POS) Form

- All Boolean equations can be written in POS form
- Each row has a **maxterm**
- A maxterm is a **sum** (OR) of literals
- Each maxterm is **FALSE** for that row (and only that row)
- Form function by **ANDing maxterms** where output is **0**
- Thus, a **product** (AND) of **sums** (OR terms)

A	***B***	***Y***	**maxterm**	**maxterm name**
0	0	**0**	$A + B$	M_0
0	1	1	$A + \overline{B}$	M_1
1	0	**0**	$\overline{A} + B$	M_2
1	1	1	$\overline{A} + \overline{B}$	M_3

$$Y = \mathrm{F}(A, B) = (A + B) \bullet (\overline{A} + B)$$

Boolean Equations Examples

- You are going to the cafeteria for lunch
  - You won't eat lunch (E = 0)
    - If it's not clean (C = 0) or
    - If they only serve meatloaf (M = 1)
- Write a truth table for determining if you will eat lunch (E).

C	M	E
0	0	0
0	1	0
1	0	1
1	1	0

SOP & POS Form

SOP – sum-of-products

C	M	E	minterm
0	0	0	$\bar{C}\,\bar{M}$
0	1	0	$\bar{C}\,M$
1	0	1	$C\,\bar{M}$
1	1	0	$C\,M$

$$E = C\bar{M}$$

POS – product-of-sums

C	M	E	maxterm
0	0	0	$C + M$
0	1	0	$C + \bar{M}$
1	0	1	$\bar{C} + M$
1	1	0	$\bar{C} + \bar{M}$

$$\begin{aligned}
E &= (C+M)(C+\bar{M})(\bar{C}+\bar{M}) \\
&= (C + M\bar{M}) * (\bar{C}+\bar{M}) \\
&= (C + 0) * (\bar{C}+\bar{M}) \\
&= C * (\bar{C}+\bar{M}) \\
&= C\bar{C} + C\bar{M} \\
&= 0 + C\bar{M} \\
&= \boxed{C\bar{M}}
\end{aligned}$$

same

SOP and Standard SOP Form

An expression is in SOP form when all products contain literals only.

SOP form: $Y = AB + B\overline{C} + DE$

NOT SOP form: $Y = DF + E(\overline{A}+B)$

SOP form: $Z = A + BC + D\overline{EF}$

❑ The Standard SOP Form

- all the variables in the domain appear in each product term in the expression.

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

POS and Standard POS Form

An expression is in **POS** form when all sums contain literals only.

POS form:	$Y = (A+B)(C+D)(\overline{E}+F)$
NOT POS form:	$Y = (D+E)(\overline{F}+GH)$
POS form:	$Z = A(B+C)(D+\overline{E})$

- The Standard POS Form
  - all the variables in the domain appear in each product term in the expression.

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

Rule12: X+YZ=(X+Y)(X+Z)

The Standard POS Form

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

POS Form ?? An expression is in **POS** form when all sums contain literals only.

Standard POS Form ?? Recall **12.** $(A + B)(A + C) = A + BC$

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

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

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

Boolean Expressions and Truth Tables (I)

Example Develop a truth table for the standard SOP expression : $\bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$

A	B	C	$\bar{A}\bar{B}C$	$A\bar{B}\bar{C}$	ABC	Output
0	0	0				0
0	0	1	**1**	Don't Care		1
0	1	0				0
0	1	1				0
1	0	0	Don't Care	**1**	Don't Care	1
1	0	1				0
1	1	0				0
1	1	1	Don't Care		**1**	1

Boolean Expressions and Truth Tables (II)

Example   Determine the truth table for the following standard POS expression:

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

<table>
<tr><th>A</th><th>B</th><th>C</th><th>$(A+B+C)$</th><th>$(A+\bar{B}+C)$</th><th>$(A+\bar{B}+\bar{C})$</th><th>$(\bar{A}+B+\bar{C})$</th><th>$(\bar{A}+\bar{B}+C)$</th><th>Output</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="4">Don’t Care</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Don’t Care</td><td>0</td><td colspan="3">Don’t Care</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td></td><td></td><td>0</td><td></td><td></td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td></td><td></td><td></td><td></td><td></td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td colspan="3">Don’t Care</td><td>0</td><td>Don’t Care</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td colspan="4">Don’t Care</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td>1</td></tr>
</table>

From Truth Tables to Standard Expressions

Example Determine the truth table for the following standard SOP 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

$\bar{A}BC$

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

$AB\bar{C}$

ABC

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

From Truth Tables to Standard Expressions

Example Determine the truth table for the following 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

$A + B + C$

$A + B + \bar{C}$

$A + \bar{B} + C$

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

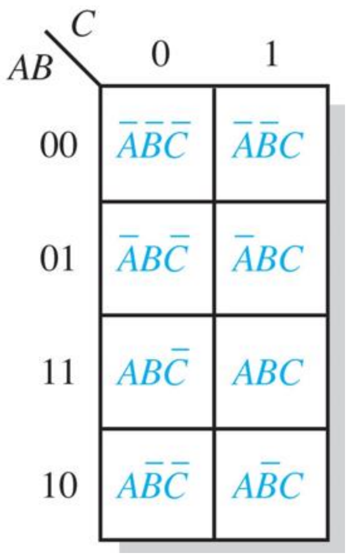
$$X = (A + B + C)(A + B + \bar{C})(A + \bar{B} + C)(\bar{A} + B + \bar{C})$$

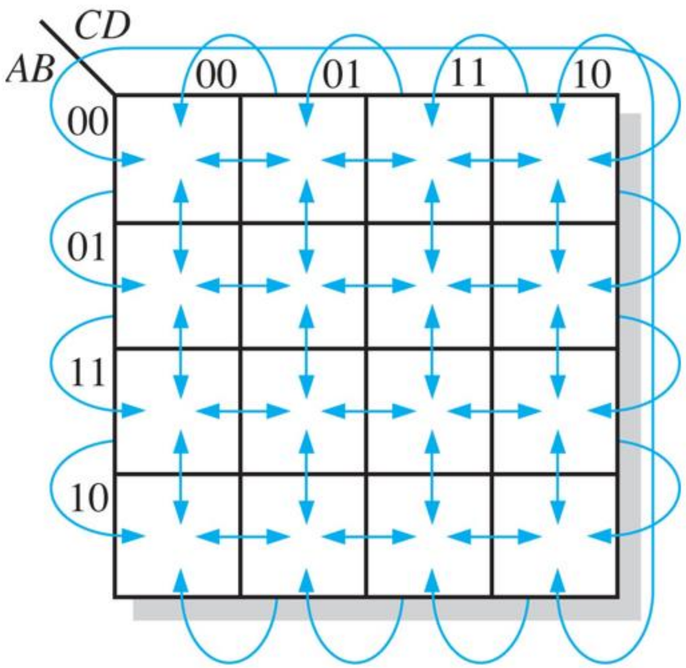
Karnaugh Maps

The Karnaugh Map (K-Map)

- A systematic method to simply Boolean expressions to their simplest SOP/POS expressions, aka. the minimum expressions
- Cells : each represents a binary value of the input variables
  - # of **cells** : the total # of possible input variable combinations
  - Adjacent cells are indexed with the **Gray code**, i.e. only a single variable change between adjacent cells.

A 3-variable K-map





A 4-variable K-map

K-Map Rules

- **Every 1 must be circled** at least once
- Circles may be **horizontal or vertical**, but not diagonal
- Each circle must span a **power of 2** (i.e., 1, 2, 4, …2^n) cells in each direction
- Each circle must be as **large** as possible
- A circle may **wrap around the edges**

3-Input K-Map

Truth Table

A	*B*	*C*	*Y*
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

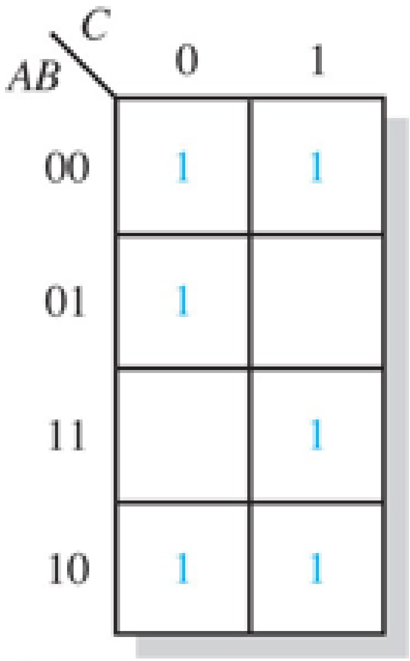
K-Map

Y — *C* \ *AB*	00	01	11	10
0	0	1	0	0
1	0	1	1	0

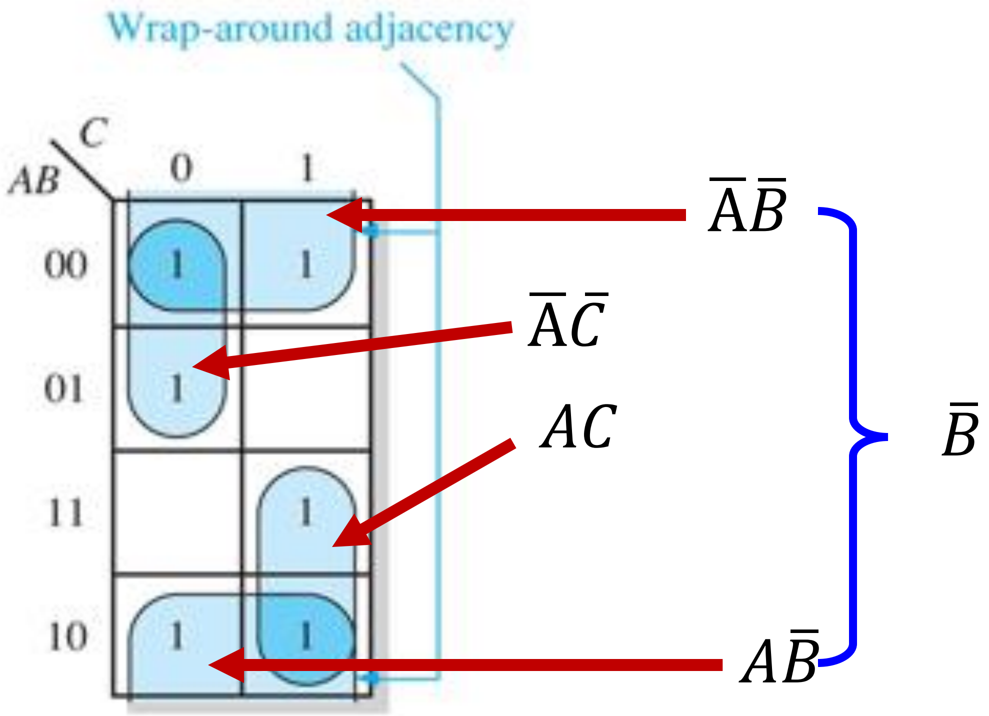
$$Y = \bar{A}B + BC$$

3-Input K-Map

Example $\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC + A\bar{B}\bar{C} + A\bar{B}C \;= \bar{A}\bar{C} + AC + \bar{B}$



(b)



(b)

4-Input K-Map

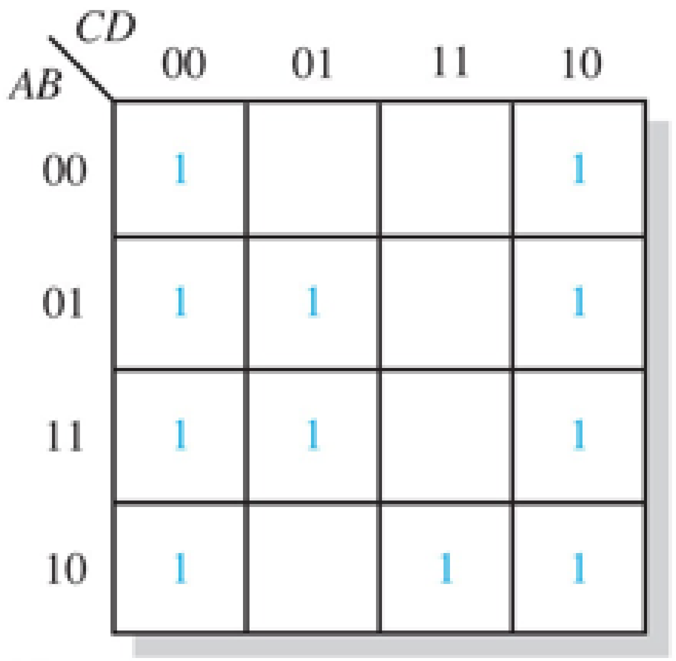
A	*B*	*C*	*D*	*Y*
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Y — *CD* \ *AB*	00	01	11	10
00	1	0	0	1
01	0	1	0	1
11	1	1	0	0
10	1	1	0	1

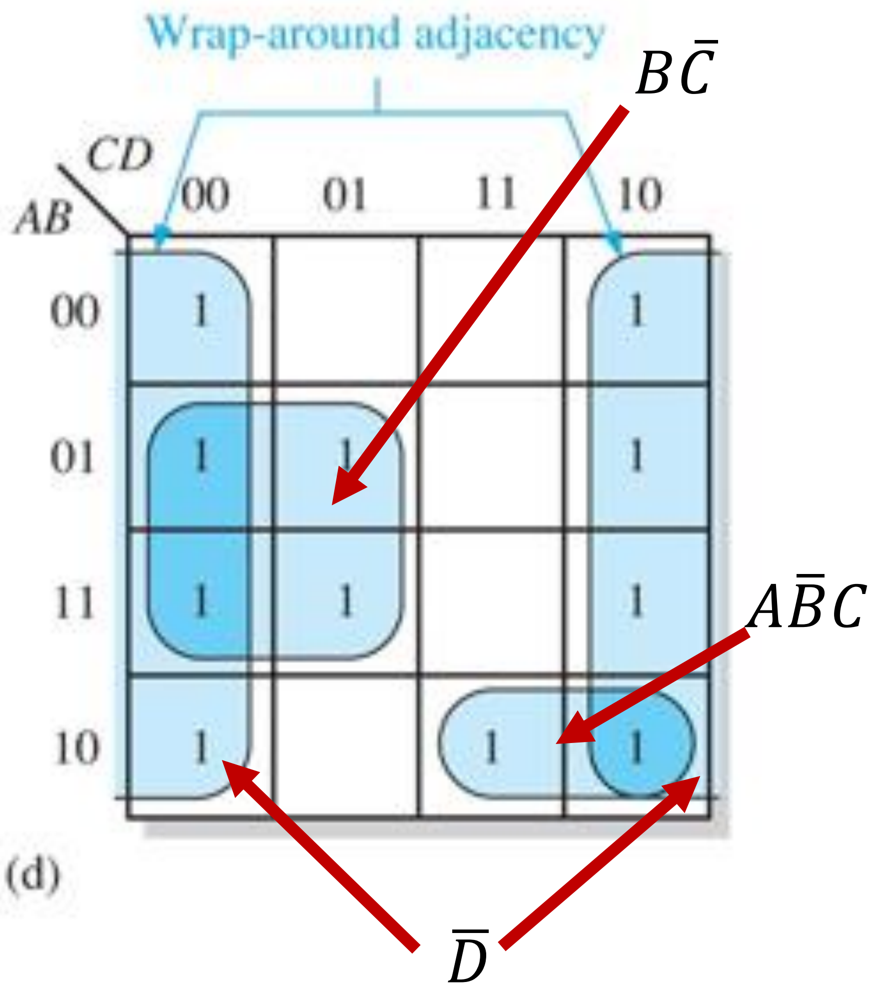
Y — *CD* \ *AB*	00	01	11	10
00	1	0	0	1
01	0	1	0	1
11	1	1	0	0
10	1	1	0	1

$$Y = \bar{A}C + \bar{A}BD + A\bar{B}\bar{C} + \bar{B}\bar{D}$$

4-Input K-Map

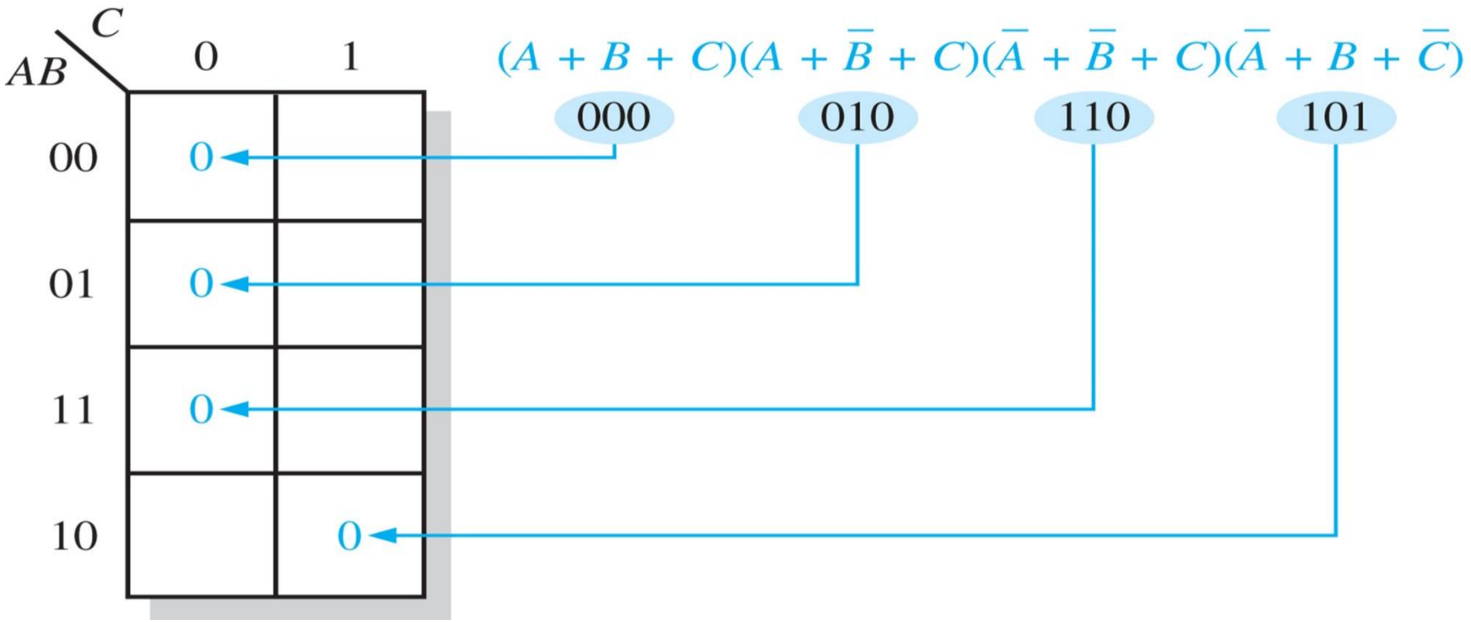


(d)



K-Map POS Minimization (I)

- In SOP minimization, we focus on those 1's
- In **POS** minimization, we focus on those **0's**

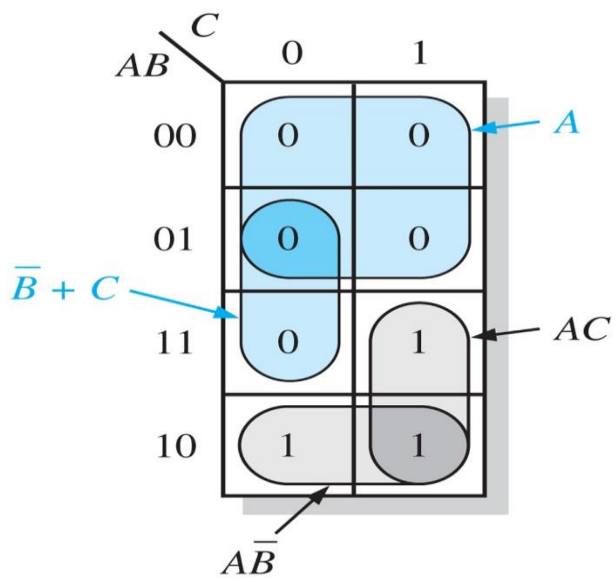


Example of mapping a standard POS expression.

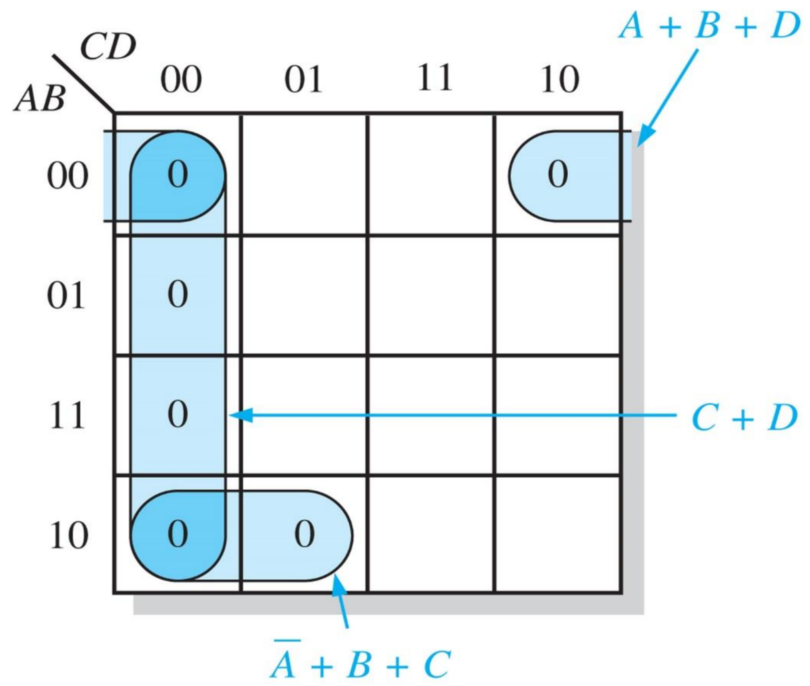
K-Map POS Minimization (I)

Example 4-34 $(A+B+C)(A+B+\bar{C})(A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+C)$

000 001 010 011 110



$AC + A\bar{B} = A(\bar{B}+C)$



Example 4-35 $(B+C+D)(A+B+\bar{C}+D)(\bar{A}+B+C+\bar{D})(A+\bar{B}+C+D)(\bar{A}+\bar{B}+C+D)$

X000 0010 1001 0100 1100

Converting btw POS & SOP Using K-Map

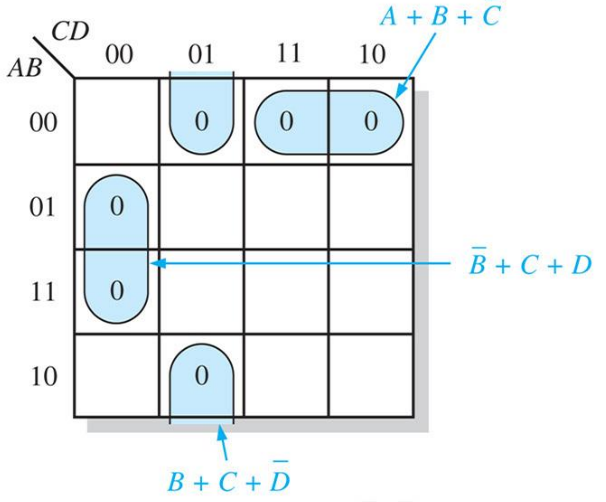
Example

Using a K-map, convert the following standard POS expression into a min. POS expression, a standard SOP expression, and a minimum SOP expression.

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

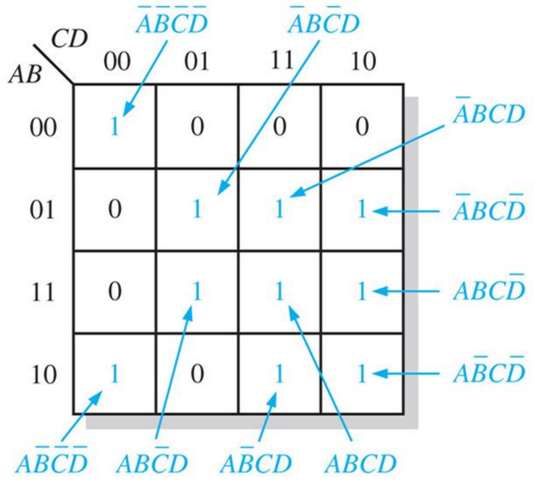
1 Find all 0's 1100 0100 0001 0011 1001 0010

2 Fill 0's



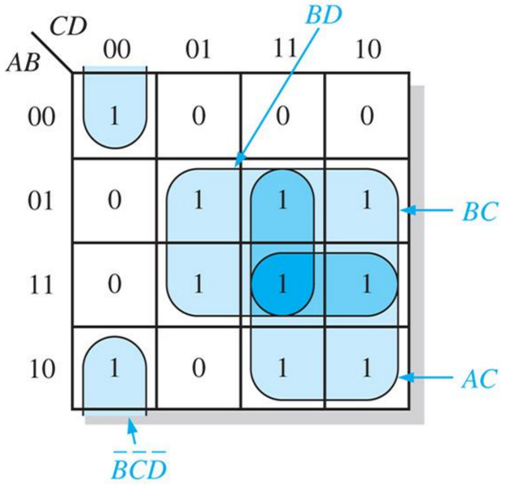
(a) Minimum POS: $(A+B+C)(\bar{B}+C+D)(B+C+\bar{D})$

3 Fill 1's



(b) Standard SOP:
$\bar{A}\bar{B}\bar{C}\bar{D}+\bar{A}B\bar{C}D+\bar{A}BCD+\bar{A}BC\bar{D}+ABC\bar{D}+A\bar{B}C\bar{D}+$
$A\bar{B}\bar{C}\bar{D}+AB\bar{C}D+A\bar{B}CD+ABCD$

4 K-Map



(c) Minimum SOP: $AC+BC+BD+\bar{B}\bar{C}\bar{D}$

K-Map with Don't Cares

K-Map Rules

- **Every 1 must be circled** at least once
- Circles may be **horizontal or vertical**, but not diagonal
- Each circle must span a **power of 2** (i.e., 1, 2, 4, …2^n) cells in each direction
- Each circle must be as **large** as possible
- A circle may **wrap around the edges**
- Circle a “don't care” (X) only if it helps minimize the equation

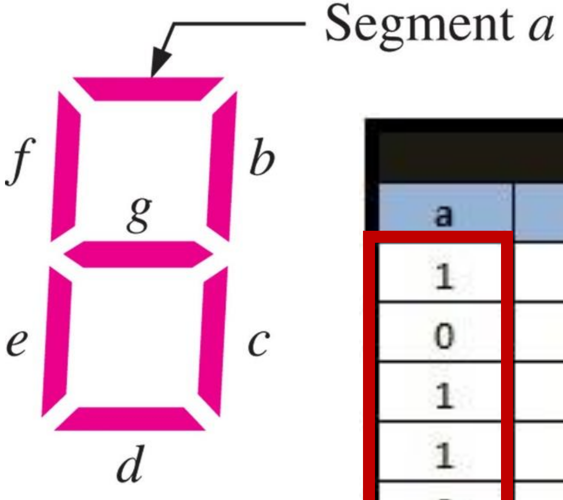
K-Maps with Don’t Cares

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	X
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Y — CD \ AB	00	01	11	10
00	1	0	X	1
01	0	X	X	1
11	1	1	X	X
10	1	1	X	X

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

K-Maps with Don't Cares: Example



Segments Inputs							7 Segment Display Output				
a	b	c	d	e	f	g		A	B	C	D
1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	1
1	1	0	1	1	0	1	2	0	0	1	0
1	1	1	1	0	0	1	3	0	0	1	1
0	1	1	0	0	1	1	4	0	1	0	0
1	0	1	1	0	1	1	5	0	1	0	1
1	0	1	1	1	1	1	6	0	1	1	0
1	1	1	0	0	0	0	7	0	1	1	1
1	1	1	1	1	1	1	8	1	0	0	0
1	1	1	1	0	0	1	9	1	0	0	1

Each digit can be represented by a BCD code

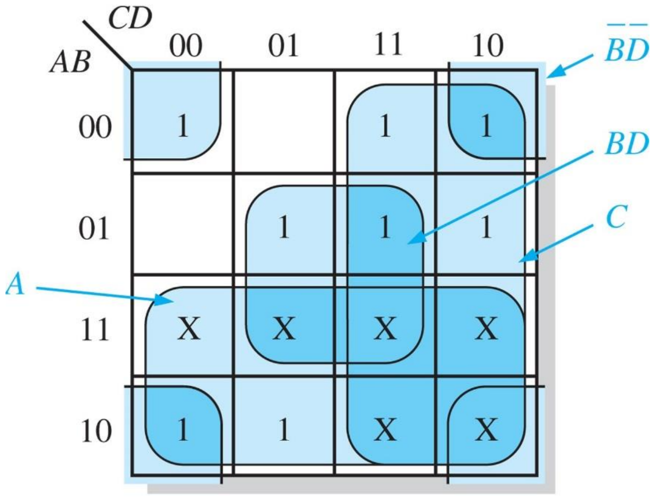
Example 4-32 (II)

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	X
1	0	1	1	X
1	1	0	0	X
1	1	0	1	X
1	1	1	0	X
1	1	1	1	X

Truth table for Segment ‘a’

AB \ CD	00	01	11	10
00	1		1	1
01		1	1	1
11	X	X	X	X
10	1	1	X	X

$$a = A + C + BD + \bar{B}\bar{D}$$



Chapter Review

- Combinational Circuits
- Boolean Equations
- Axioms & Theorems
    - Commutative laws
    - Associative laws
    - Distributive law
    - Rules of Boolean Algebra
    - DeMorgan's Theorems
- Simplifying Equations
    - SOP and POS
- Karnaugh Maps
    - Don't Cares

True/False Quiz

- ✓ Variable, complement, and literal are all terms used in Boolean algebra.
- ✗ Addition in Boolean algebra is equivalent to the NOR function.
- ✓ Multiplication in Boolean algebra is equivalent to the AND function.
- ✓ The commutative law, associative law, and distributive law are all laws in Boolean algebra.
- ✗ The complement of 0 is 0 itself.
- ✗ When a Boolean variable is multiplied by its complement, the result is the variable
- ✓ "The complement of a product of variables is equal to the sum of the complements of each variable" is a statement of DeMorgan's theorem.
- ✓ SOP means sum-of-products.
- ✓ Karnaugh maps can be used to simplify Boolean expressions.
- ✗ A 3-variable Karnaugh map has six cells.