Digital Circuits
ECS 371

Dr. Prapun Suksompong

prapun@siit.tu.ac.th

Lecture 7

Office Hours:
BKD 3601-7
Monday 9:00–10:30, 1:30–3:30
Tuesday 10:30–11:30

ECS371.PRAPUN.COM

Announcement

- HW2 posted on the course web site
  - Chapter 4: 6(b,c,d,e), 13, 15, 20, 22
  - **Write down all the steps** that you have done to obtain your answers.
  - Due date: July 2, 2009 (Thursday)
  - Please submit your HW to the instructor 3 minutes BEFORE your class starts.
  - Late submission will NOT be accepted.
  - Earlier submission is possible. There are two HW boxes in the EC department (6th floor) for ECS 371. (One for CS. Another one for IT.)
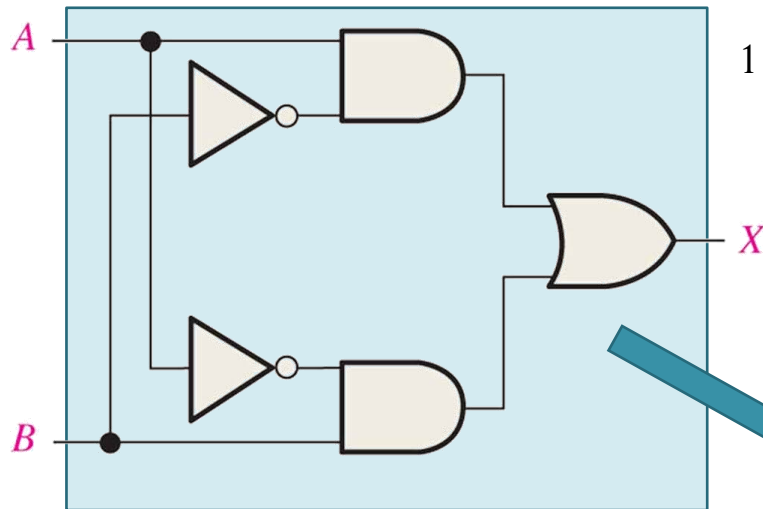
Announcement

- Reading Assignment
  - Chapter 4: 4.1to 4-9

Our Goals

- Ultimate goals:
    1. Analyze digital circuits.
    2. Design digital circuits to do something useful.

1. Analyze digital circuit:
    - Given a digital circuit, find out what it does.
    - One way to achieve this goal is to look at the truth table.

Example: Analyzing Logic Circuit

1. Suppose we are given a digital circuit.

2. Analyze it via the truth table

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

It may be useful to find a Boolean expression first.

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

3. Interpretation: This is a circuit that checks whether A ≠ B.
Output X = 1 if A and B are not the same.
Output X = 0 if A is the same as B.

We can apply the same technique to more complicated circuits.

Sometimes, we may be able to simplify the circuit by simplifying its Boolean expression.

Our Goals

- Ultimate goals:
  1. Analyze digital circuits.
  2. Design digital circuits to do something useful.

1. Analyze digital circuit:

2. Design digital circuits:

- Given a task, build a SIMPLE digital circuit that perform the specified task.

- A recipe:
  - Directly construct a Boolean expression corresponding to the task or start with a truth table and then turn it into a Boolean expression.
  - Simplify the expression. ← Many examples last week
  - Construct a circuit from the simplified expression.

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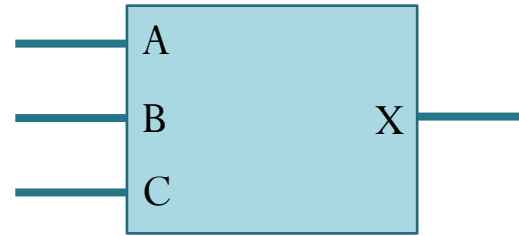
$$X = A\bar{B}C + \bar{A}BC + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C}$$

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

$= \bar{A}BC + \left(A\bar{B}C + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} \right)$

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

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

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

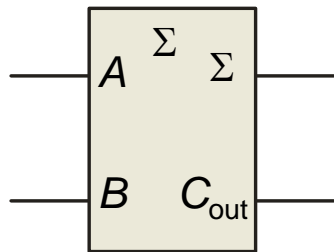Direct construction of a circuit from this equation will be complicated.

Simplification

Construction of an equivalent circuit from this equation is simpler.

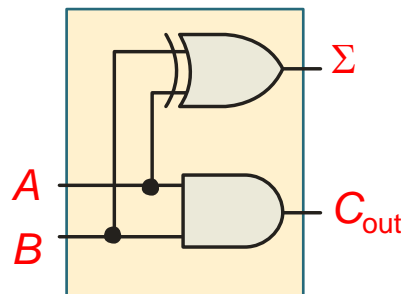$$X = \bar{A}C + \bar{B}$$

Example: Designing Logic Circuit

1. Suppose we want to build a circuit that perform a one-bit addition

2. We can summarize its task by a truth table.

Inputs		Outputs	
A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

3. Can we turn this truth table into a simple circuit?

Example: Designing Logic Circuit (2)

1. Suppose we want to build a circuit that perform a one-bit addition with input carry

2. We can summarize its task by a truth table.

3. Can we turn this truth table into Boolean expression?

How?

Inputs			Outputs	
A	B	C_{in}	C_{out}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$\Sigma = (A \oplus B) \oplus C_{in}$

$(A \oplus B)C_{in}$

$C_{out} = AB + (A \oplus B)C_{in}$

AB
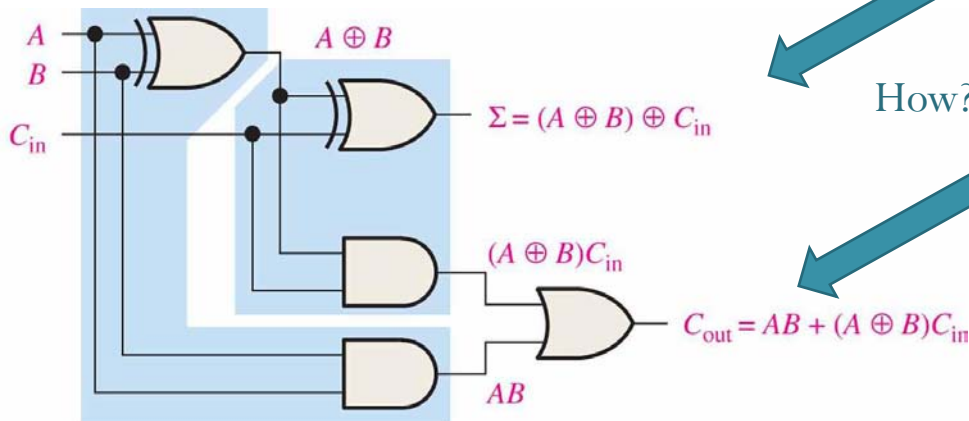
Truth Table ➡ Boolean Expression

- How can we turn a truth table into Boolean Expression?

- Let's first study the relationship between truth table and Boolean expression.

- There are some forms of Boolean expressions that are easily converted into their corresponding truth tables.
  - Sum-of-products form
  - Product-of-sums form

Product Term

A single literal or a product of two or more literals.

Example:

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

$$A \cdot C$$

$$A$$

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

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

Caution:

$$\overline{A \cdot B \cdot C} \text{ is not a product term.}$$

$Q:$ When does $A \cdot \overline{B} \cdot C = 1$?

$A:$ IFF $(A, B, C) = (1, 0, 1)$

So, it is very easy to construct a truth table for

$$X = A \cdot \overline{B} \cdot C$$

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

Sum Term

A single literal or a sum of two or more literals.

Example:

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

$$A + C$$

$$A$$

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

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

Caution:

$\overline{A + B + C}$ is not a sum term.

Q : When does $A + \overline{B} + C = 1$?

It might be easier to ask:

Q : When does $A + \overline{B} + C = 0$?
A : IFF $(A, B, C) = (0, 1, 0)$

Q : When does $A + \overline{B} + C = 1$?
A : IFF $(A, B, C) \neq (0, 1, 0)$

It is easy to construct a truth table for

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

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

Again

1. It is easy to find the case when a product term = 1.

$$A \cdot \overline{B} \cdot C = 1 \text{ iff } (A, B, C) = (1, 0, 1)$$

2. It is easy to find the case when a sum term = 0.

$$A + \overline{B} + C = 0 \text{ iff } (A, B, C) = (0, 1, 0)$$

Sum-of-Product (SOP) Form

A product term or a sum of product terms

Example: $A\overline{B} + ABC$

$$AB + C$$

$$ABC + CDE$$

Example: Develop a truth table for $X = A\overline{B} + ABC$

Observe that when a product term does not contain all variables, it corresponds to multiple 1s in the output column of the truth table. This is because the missing variable does not affect the product term and hence can be 0 or 1.

A	B	C	X
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

Observe that when a product term contains all the variables, it corresponds to a unique 1 in the output column of the truth table.

Standard SOP (Canonical Sum)

Each product term contains all variables.

Example:

Nonstandard product term → $A\overline{B} + ABC$ is not a standard SOP expression.

$A\overline{B}C + A\overline{B}\,\overline{C} + ABC$ is a standard SOP expression.

Minterm: A product term in a standard SOP expression.

Example: $A\overline{B}C + A\overline{B}\,\overline{C} + ABC$ contains 3 minterms

Observe these!

- A minterm is equal to 1 for only one combination of variable values.

- A minterm corresponds to a unique 1 in the output column of the truth table.

- A minterm corresponds to a unique row of the truth table.

- A canonical sum is a sum of minterms.

Remember these:

- **SOP** = Sum of products
- *Standard* **SOP** = Canonical sum
- **Minterm** = A product term in canonical sum

Domain

Domain of an expression:

   Set of variables contained in the expression.

Example: Domain of $A\overline{B} + ABC$ is $\{A, B, C\}$.

So, a **standard** SOP expression is one in which all the variables in the domain appear in each product term in the expression.

Row Number of the Truth Table

Row #	A	B	C	X
0	0	0	0	
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	
6	1	1	0	
7	1	1	1	

Think of these as one unsigned binary number. The corresponding decimal value is the row #.

Minterm/Row Number & Truth Table

Row #	A	B	C	Minterm
0	0	0	0	$\overline{A} \cdot \overline{B} \cdot \overline{C}$
1	0	0	1	$\overline{A} \cdot \overline{B} \cdot C$
2	0	1	0	$\overline{A} \cdot B \cdot \overline{C}$
3	0	1	1	$\overline{A} \cdot B \cdot C$
4	1	0	0	$A \cdot \overline{B} \cdot \overline{C}$
5	1	0	1	$A \cdot \overline{B} \cdot C$
6	1	1	0	$A \cdot B \cdot \overline{C}$
7	1	1	1	$A \cdot B \cdot C$

Summary: Each row of the truth table corresponds to
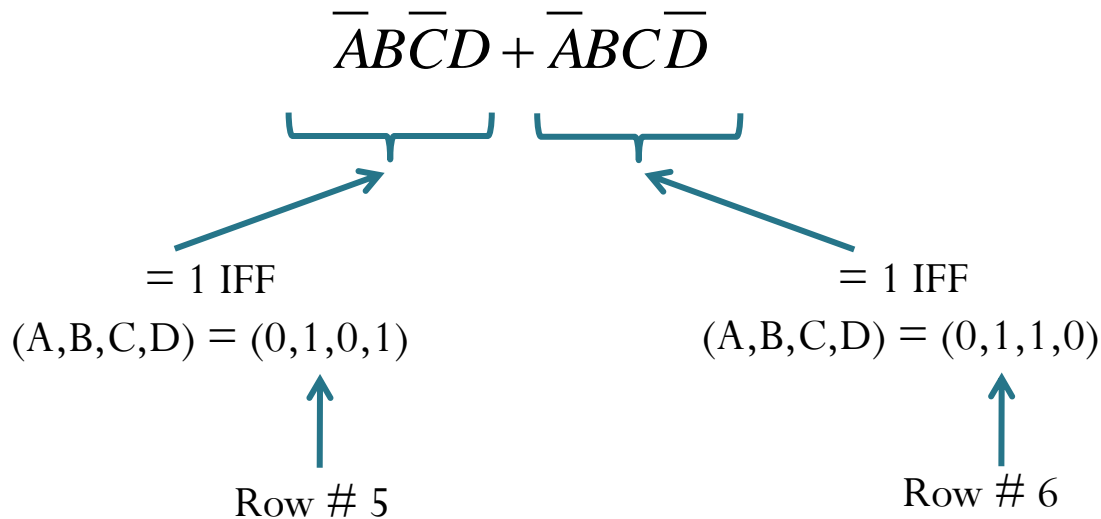1. A combination of input (A,B,C).
2. A row #.
3. A minterm.

Binary Representation

$$X = \overline{A}B\overline{C}D + \overline{A}BC\overline{D}$$

(of minterm and of any Boolean expression)

Example: Consider a canonical sum:

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

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

= 1 IFF
(A,B,C,D) = (0,1,0,1)

= 1 IFF
(A,B,C,D) = (0,1,1,0)

Row # 5

Row # 6

This is called a **minterm list**

$$\overline{A}B\overline{C}D + \overline{A}BC\overline{D} = \sum_{A,B,C,D}(5,6)$$

It means "the sum of minterms 5,6 with variable A, B, C, D"

Canonical Sum and Minterm List

- Example: Convert $A\overline{B}C + \overline{A} \cdot \overline{B}$ into canonical sum.

$$\text{Hint}: X = X \cdot 1 = X \cdot \left(Y + \overline{Y}\right)$$
$$= X \cdot Y + X \cdot \overline{Y}$$

- Example: Convert $A\overline{B}C + \overline{A} \cdot \overline{B}$ into a minterm list.

- Any logic function/expression can be expressed as a canonical sum (which is equivalent to a minterm list).

Truth Table ➡ Boolean Expression

(A Revisit!)

Add all the minterms corresponding to a 1 in the truth table.

A	B	C	X
0	0	0	1
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

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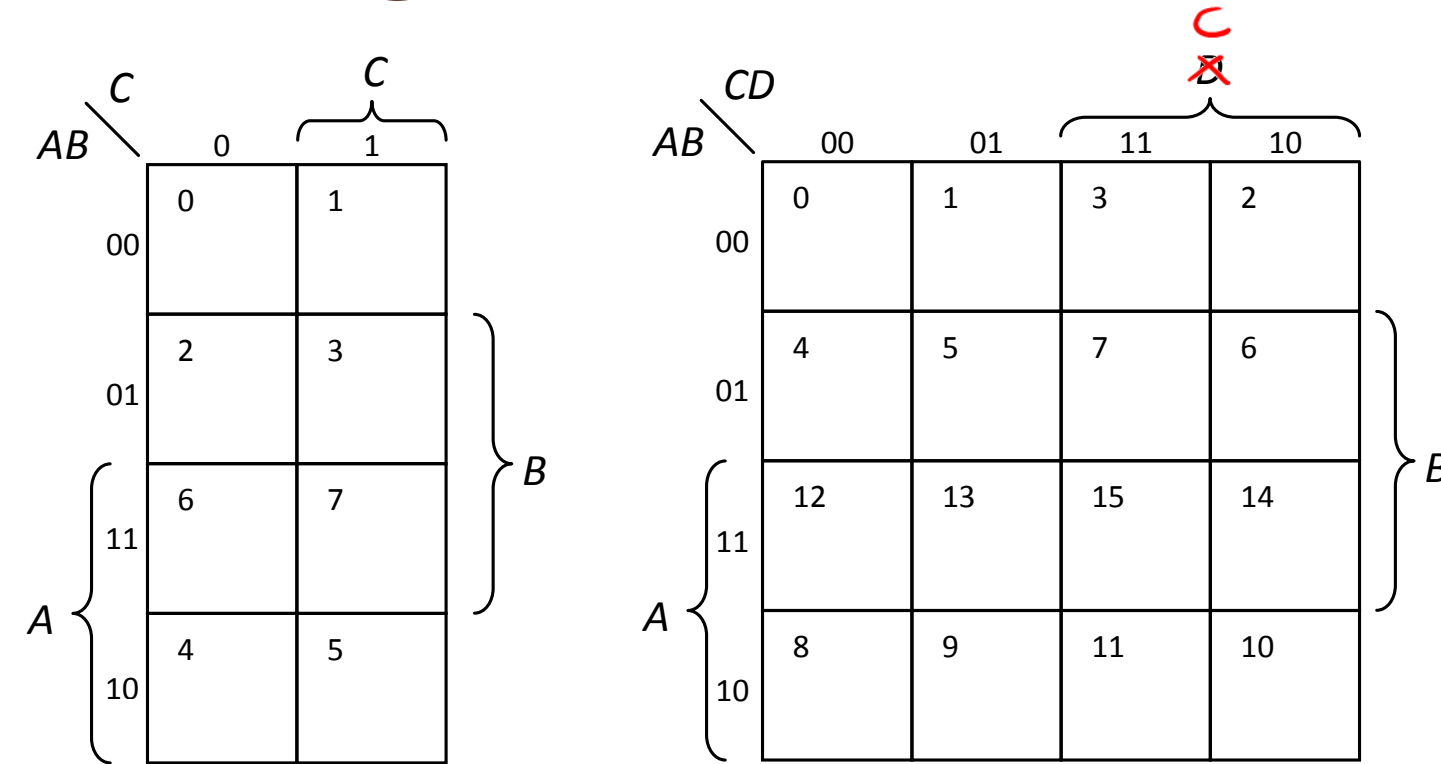$$+ A \cdot B \cdot \overline{C} + A \cdot B \cdot C$$

Alternatively, we may write

$$X = \sum_{A,B,C}(0,3,4,6,7)$$

Minimization

"The process that results in an expression containing the fewest possible terms with the fewest possible variables."

The expression can then be implemented with fewer logic gates.

Karnaugh Map

The small number inside each cell is the corresponding row number in the truth table, assuming that the truth table inputs are labeled alphabetically from left to right (e.g. A, B, C) and the rows are numbered in binary counting order.

Row #	A	B	C
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Row #	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

24

Karnaugh Map: Basic Layout

- Similar to a truth table.
  - Present all of the possible values of input variables and the resulting output for each value.
- Instead of being organized into columns and rows, it is an **array of cells**.
- The number of cells in a K-map, as well as the number of rows in a truth table, is equal to the total number of possible input variable combinations.
- The cells are arranged in a way so that simplification of a given expression is simply a matter of properly grouping the cells.
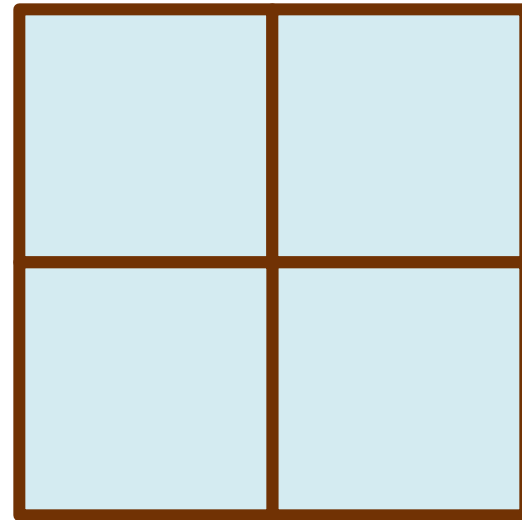- Used for simplifying Boolean expressions to their "minimum form".
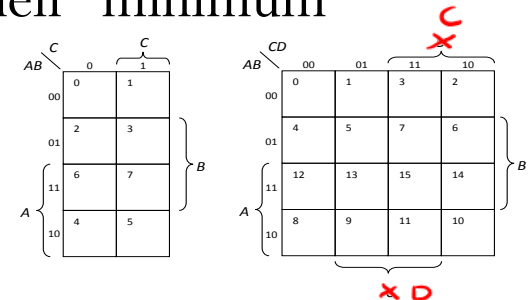
The 2-variable Karnaugh Map (2x2 K-Map)

$\text{Domain} = \{A, B\}$

Each cell represent one minterm.

A	B	Minterm
0	0	$\overline{A} \cdot \overline{B}$
0	1	$\overline{A} \cdot B$
1	0	$A \cdot \overline{B}$
1	1	$A \cdot B$

The 3-variable Karnaugh Map

$$\text{Domain} = \{A, B, C\}$$

(a)   (b)

The 4-variable Karnaugh Map

Domain = $\{A, \cancel{B}, C\}$ $\{A, B, C, D\}$

CD AB	00	01	11	10
00				
01				
11				
10				

(a)

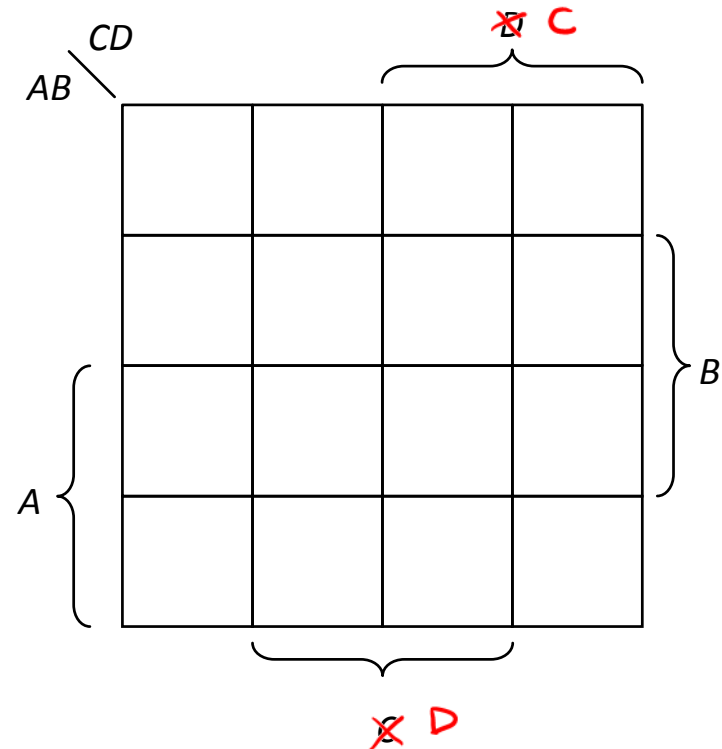| CD AB	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}CD$	$\bar{A}\bar{B}C\bar{D}$
01	$\bar{A}B\bar{C}\bar{D}$	$\bar{A}B\bar{C}D$	$\bar{A}BCD$	$\bar{A}BC\bar{D}$
11	$AB\bar{C}\bar{D}$	$AB\bar{C}D$	$ABCD$	$ABC\bar{D}$
10	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}D$	$A\bar{B}CD$	$A\bar{B}C\bar{D}$

(b)

Mapping (Canonical Sum)

Map the following expression on a Karnaugh map

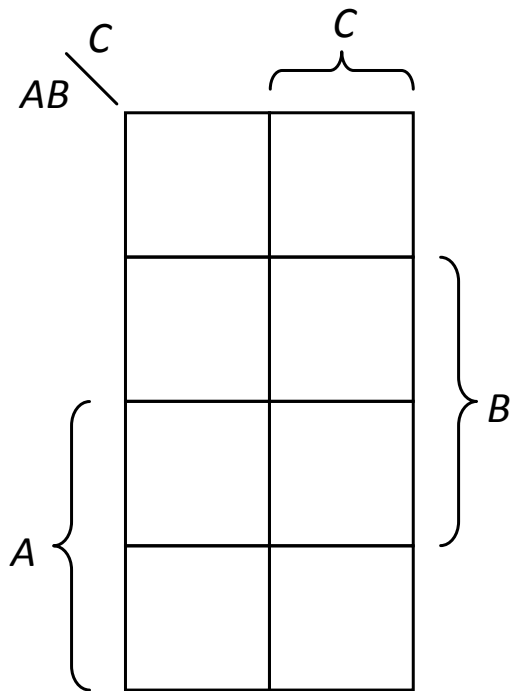$$\overline{A}\,\overline{B}C + \overline{A}B\overline{C} + AB\overline{C} + ABC$$
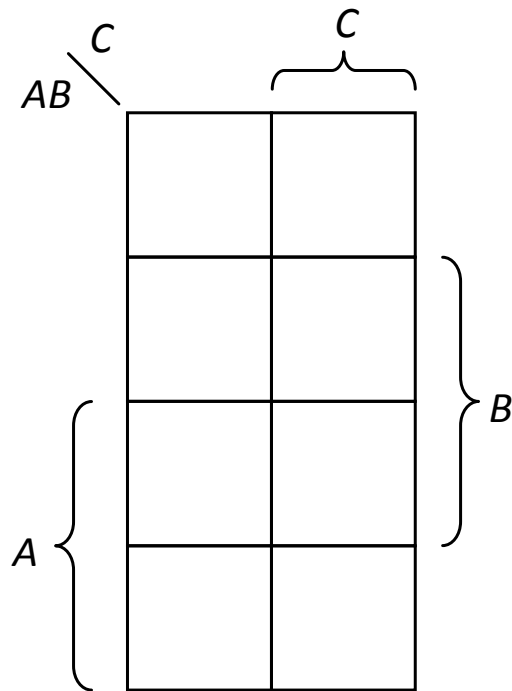
$$\overline{A}\,\overline{B}CD + \overline{A}B\overline{C}\,\overline{D} + AB\overline{C}D + ABCD + AB\overline{C}\,\overline{D}$$
$$+ \overline{A}\,\overline{B}\,\overline{C}D + A\overline{B}C\overline{D}$$

Mapping (SOP)

Map the following expression on a Karnaugh map

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

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