Digital Circuits

ECS 371

Dr. Prapun Suksompong

prapun@siit.tu.ac.th

Lecture 9

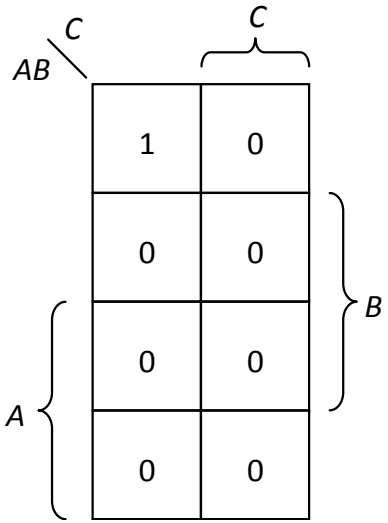
Office Hours:
BKD 3601-7
Monday 9:00-10:30, 1:30-3:30
Tuesday 10:30-11:30

ECS371.PRAPUN.COM

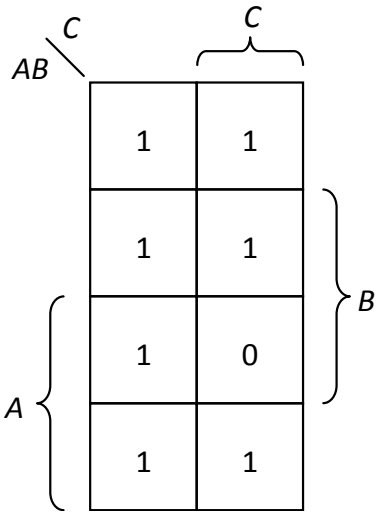
Caution

When you see $\overline{A}\overline{B}C$ or $\overline{A}\overline{B}\overline{C}$ on quiz/HW/exam, please always **double-check** whether the bars on the top are disconnected.

This is the K-map for $X = \overline{A}\overline{B}\overline{C}$ which is the same as $X = \overline{A} \cdot \overline{B} \cdot \overline{C}$



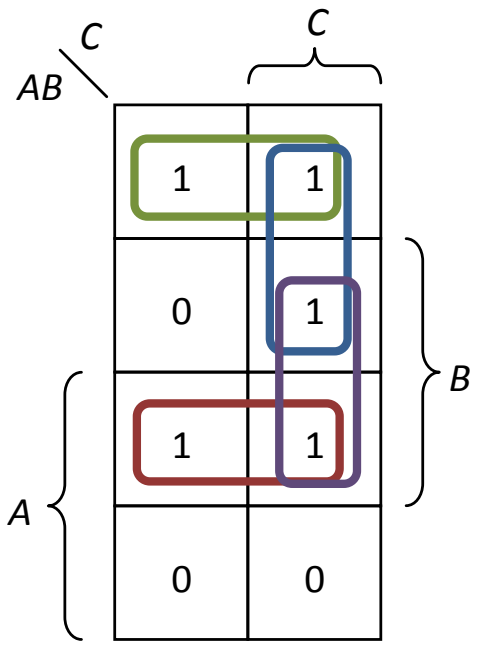
This is the K-map for $X = \overline{ABC}$ which is equivalent to $X = \overline{A} + \overline{B} + \overline{C}$



Non-uniqueness

Use a K-map to minimize the following expression

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



Solution 1: $AB + \overline{A}\overline{B} + \overline{A}C$

Solution 2: $AB + \overline{A}\overline{B} + BC$

K-Map POS Minimization

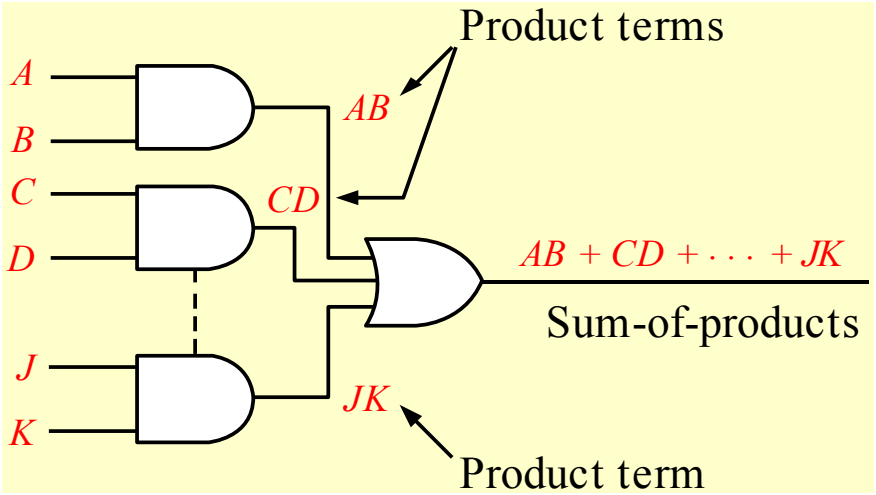
- Appendix B in the textbook.
- For a POS expression in standard form, a 0 is placed on the K-map for each sum term in the expression.
- The cells that do not have a 0 are the cells for which the expression is 1.
- Group 0s to produce instead of grouping 1s.

Combinational Logic

- Chapter 5 and 6
- Reading Assignment:
  - Read Section 5-1 to 5-5.
- Definition: A **combinational logic** is a combination of logic gates interconnected to produce a specified Boolean function with no storage or memory capability.
- Sometimes called **combinatorial logic**.

SOP Implementation: AND-OR Circuit

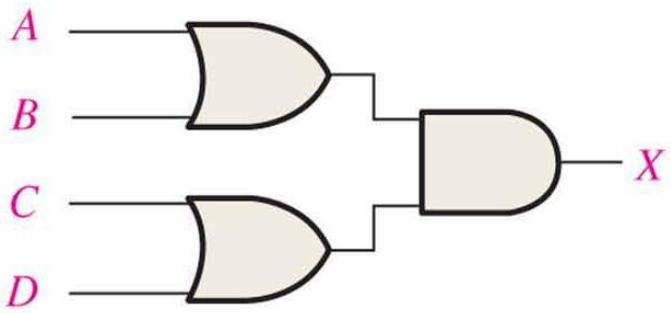
In Sum-of-Products (SOP) form, basic combinational circuits can be directly implemented with AND-OR combinations: first forming the AND terms; then the terms are ORed together.



This is called the **AND-OR configuration**.

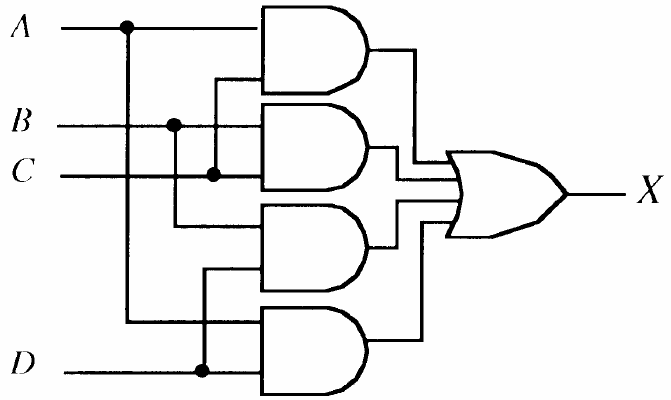
Example

Write the output expression of the following circuit as it appears in the figure and then change it to an equivalent AND-OR configuration.



Solution:

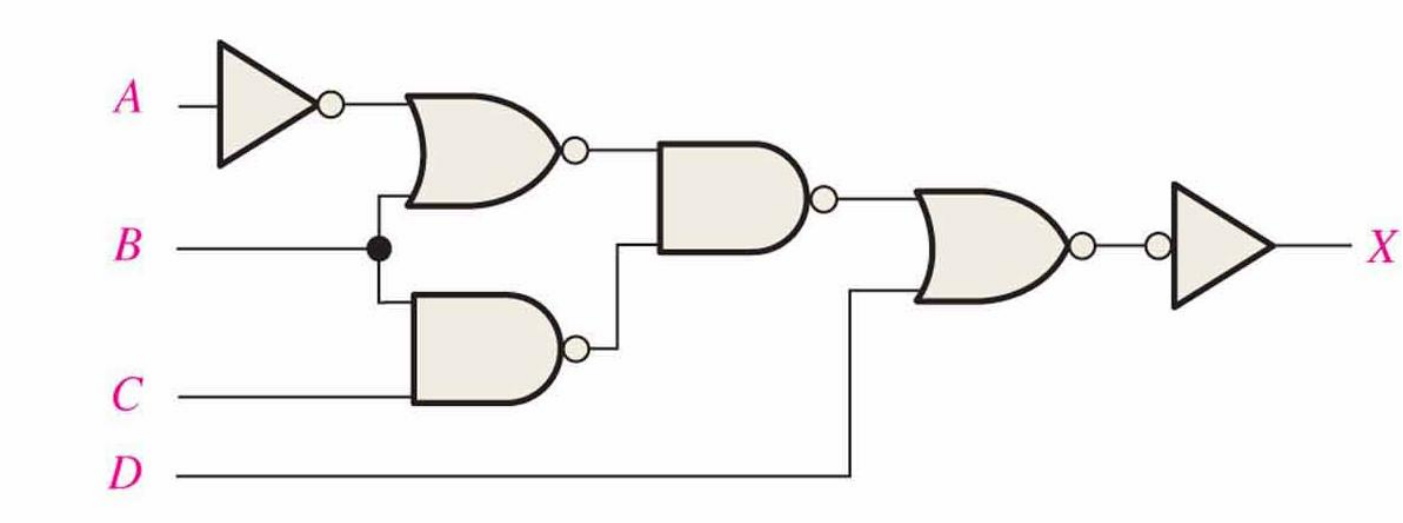
$$X = (A+B)\cdot(C+D)$$
$$= (A+B)\cdot C + (A+B)\cdot D$$
$$= AC + BC + AD + BD$$



(a)

Example

Write the output expression of the following circuit as it appears in the figure and then change it to an equivalent AND-OR configuration.



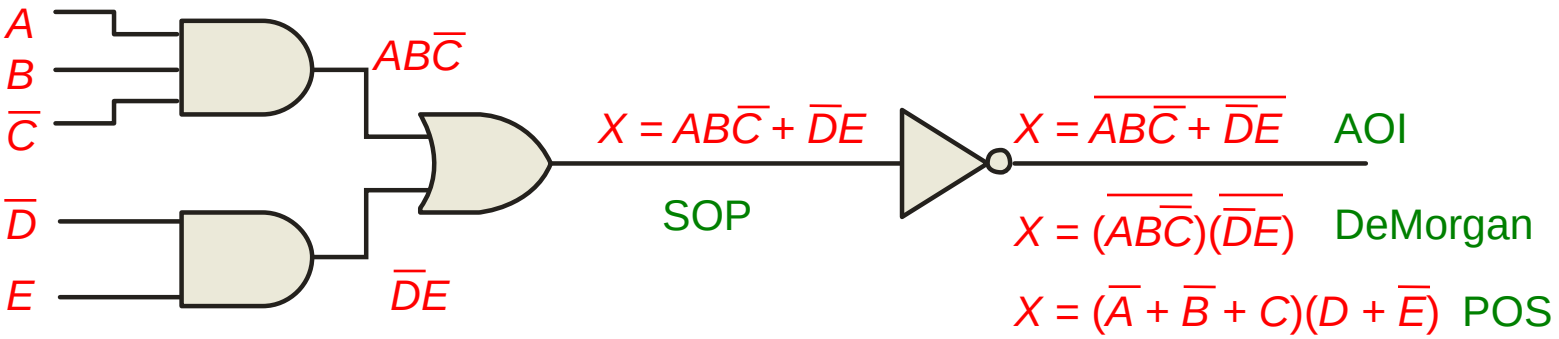
Remark

1. From any logic expression, you can construct a truth table.
2. From the truth table you can get a canonical sum or a minterm list. (This can be simplified to a minimal sum. In any case, you get a SOP expression)
3. Any SOP expression can be implemented using AND gates, OR gates, and inverters.

AND-OR-Invert (AOI) circuit

When the output of a SOP form is inverted, the circuit is called an **AND-OR-Invert circuit**.

The AOI configuration lends itself to product-of-sums (POS) implementation.

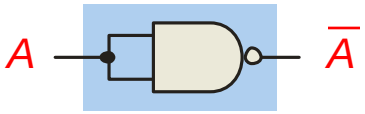


Universal gate

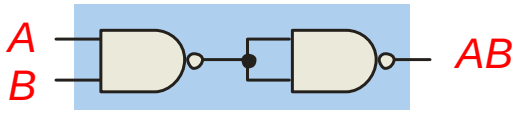
- The term **universal** refers to a property of a gate that permits any logic function to be implemented by that gate or by a combination of gates of that kind.
- Example: NAND gates, NOR gates

NAND Gate as a Universal Gate

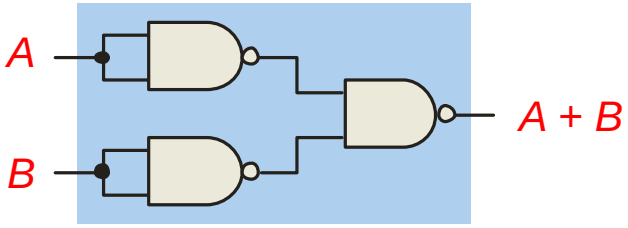
NAND gates are sometimes called **universal** gates because they can be used to produce the other basic Boolean functions.



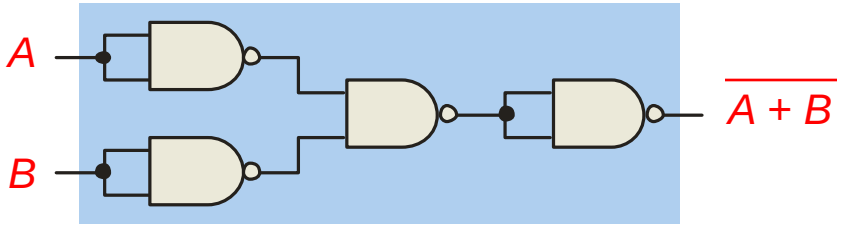
Inverter



AND gate



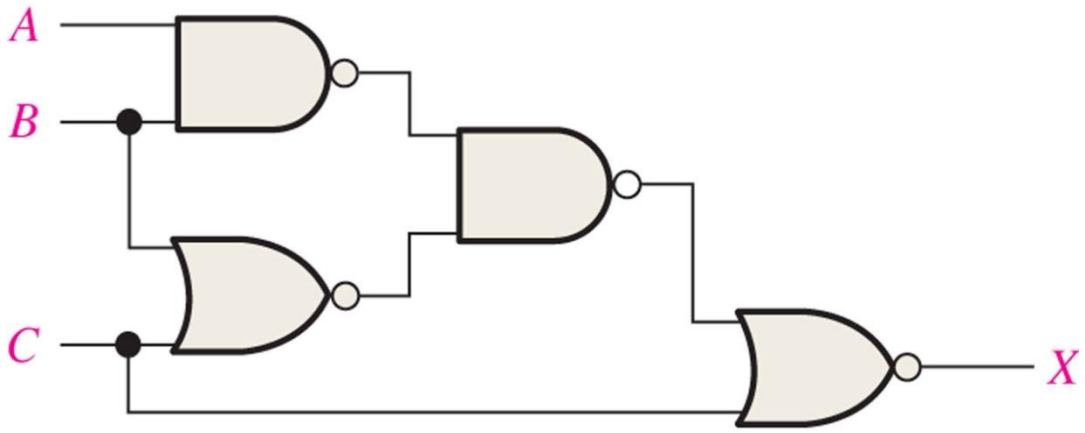
OR gate



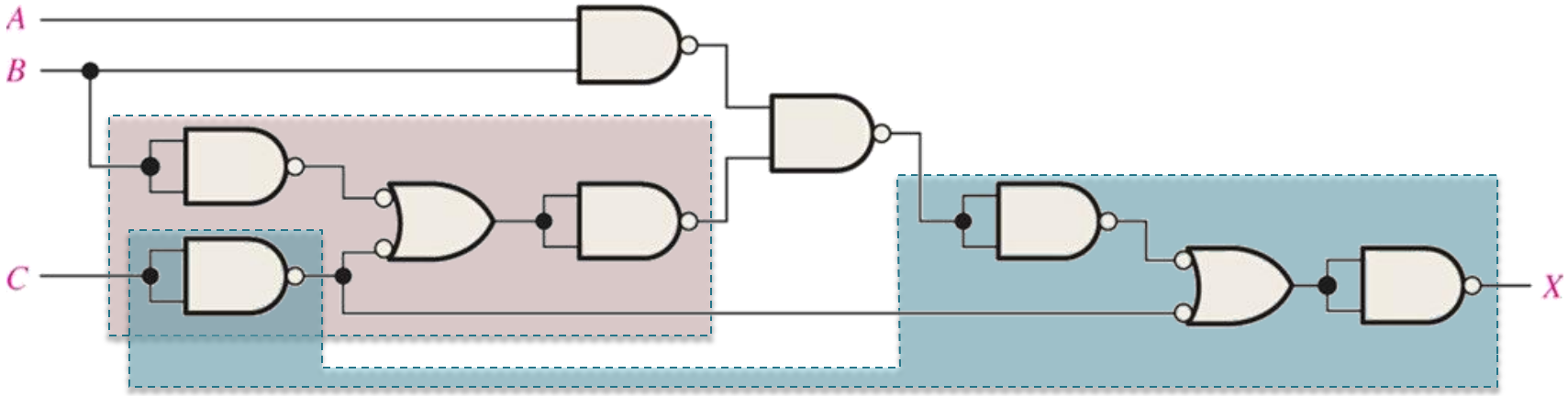
NOR gate

Example

Implement the following logic circuit using only NAND gates:

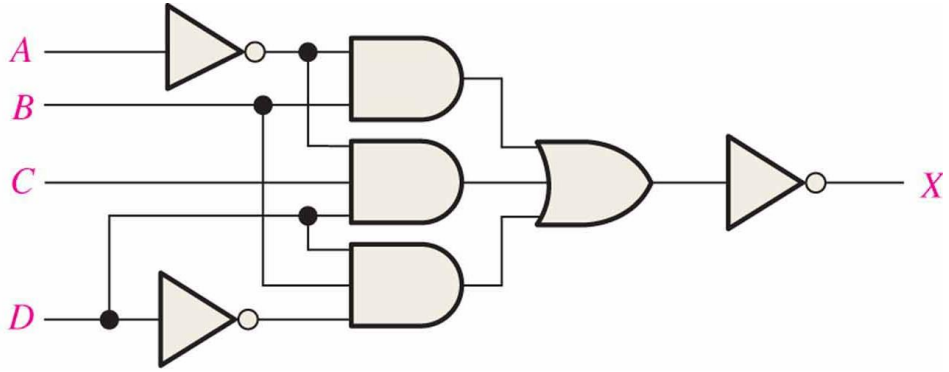


Solution:

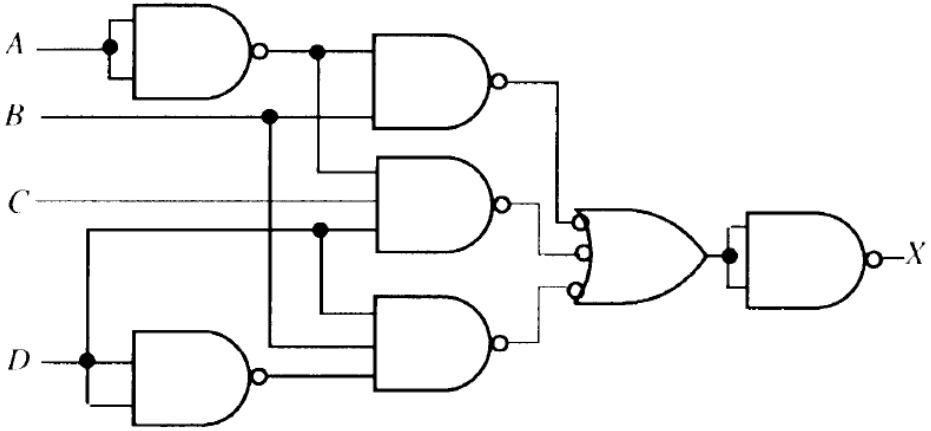


Example

Implement the following logic circuit using only NAND gates:

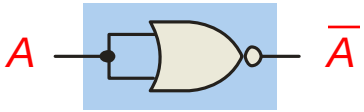


Solution:

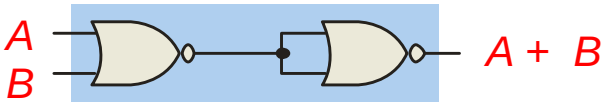


NOR Gate as a Universal Gate

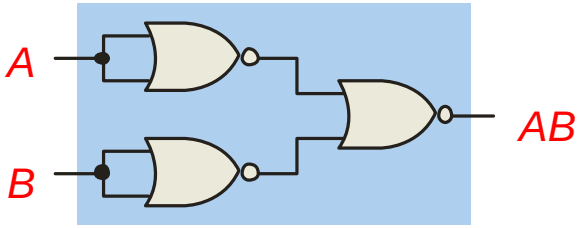
NOR gates are also **universal** gates and can form all of the basic gates.



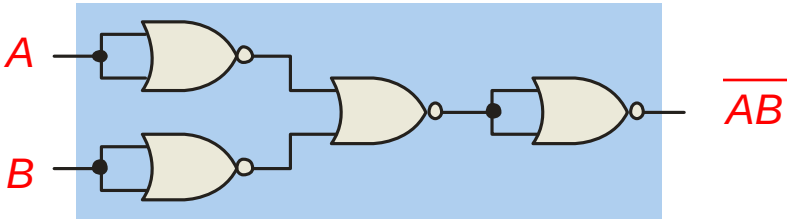
Inverter



OR gate



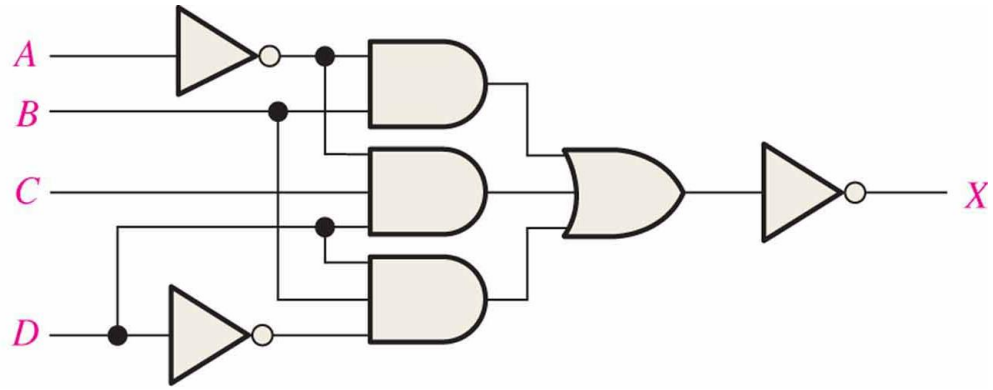
AND gate



NAND gate

Example

Implement the following logic circuit using only NOR gates:



Solution: