# HUST

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

# School of Electronics and Telecommunications
# Electronics Devices – ET2015E

ONE LOVE. ONE FUTURE.

TRƯỜNG ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

# Chapter 4. Digital logics

**Tutor:** Assoc. Prof. Nguyen Tien Dzung
**Department:** Electronics and Computer Engineering
**Emai:** tiendungbk@gmail.com; dzung.nguyentien@hust.edu.vn
**Mobile:** 0988.355.343
**Office:** 94 Le Thanh Nghi, R102, C-Bulding (Tại chức)

ONE LOVE. ONE FUTURE.

# Outline

4.1. Fundamentals of a logic algebra: number systems, logic variable and logic functions, Boolean algebra and properties

4.2. Representation of a logic function: truth table, logic function expression, Karnaugh table

4.3. Basic logical gates: AND, OR, NOT, NAND, NOR, XOR

4.4. Minimization of a logic function: algebraic method, K-table based method

4.5. Implementation of a logic function using logic gates: OR-AND, AND-OR, NAND-NAND, NOR-NOR

4.6. Some typical logic applications

# 4.1. Fundamentals of a logic algebra

1. Numeral system:

   a) Base or radix of a number system: $X_{10} = \sum_0^{n-1} a_i\, d^i$, where $d^i$ is weighting

   ✓ Decimal: d = 10;  digits $a_i$ = 0 to 9 – well-known number system

   ✓ Binary: d = 2; digits $a_i$ = 0,1 or bits or FALSE/TRUE

   ✓ Hexadecimal: d = 16, digits $a_i$ = 0 to 9 + letters A, B, C, D, E, F

   ✓ Octal: d = 8, digits $a_i$ = 0 to 7

   b) Conversion between number system:

   ✓ 2 to 10: $1101_2 = 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 8 + 4 + 0 + 1 = 13_{10}$, ($2^i$ – weighting)

   ✓ 10 to 2: $13_{10} = 8 + 4 + 0 + 1 = 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 = 1101_2$

   ✓ Example: $14_{10} = X_2$ ? $113_{10} = X_2$ ?

   c) Binary numeral system and Boolean algebra:

   ✓ Boolean algebra: Logical operations with bits

   ✓ Logical constant, variable, function: take values in set {0, 1} (e.g. A, F(A, B))

   ✓ Basic logical functions: AND (A AND B or F = A.B), OR (A OR B or F = A+B), NOT (or F = $\bar{A}$)

## d) Properties of Boolean algebra

✓ Relation between logic constants:

- AND: 0.0 = 0, 0.1 = 0, 1.0 = 0, 1.1 = 1
- OR: 0+0 = 0, 0+1 = 1, 1+0 = 1, 1+1 = 1
- NOT: $\bar{0} = 1$, $\bar{1} = 0$

✓ Relation between logic variables: assuming variables A, B, C …

- AND: 0.A = 0, A.0 = 0, A.1 = A, 1.A = A
- OR: 0+A = A, A+0 = A, 1+A = 1, A+1 = 1,
- Involution: $\bar{\bar{A}} = A$
- Idempotence: A.A = A, A + A = A
- Complement: $\bar{A} + A = 1$ , $\bar{A}.A = 0$
- Association: A.B + A.C = A(B+C), A + B + C = (A + B) + C = A + (B + C)
- Distribution: A(B+C) = A.B + A.C

✓ Useful relations:

- $\mathrm{A}.(\mathrm{B}+\bar{\mathrm{B}}) = A.1 = A;\ A.(\bar{A}\ + B) = A.\bar{A} + A.B = 0 +\ A.B = A.B; A.(A+B) = AB;\ A + A.B = B$;

# 4.1. Fundamentals of a logic algebra (cont.)

✓ De Morgan theorem: Applying for multiple variables

- $\overline{A.B} = \bar{A} + \bar{B}$ ; $\overline{A + B} = \bar{A}.\bar{B}$
- AND ⇔ OR, variable ⇔ negated variable

✓ Example:

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

# 4.2. Representation of a logic function

## Truth table, Logic expression, Karnaugh table

a) True table: list all $2^n$ variable combination of a n-variable function and corresponding outputs

- ✓ n = 2: possible $2^2$ = 4 variable combinations $=\bar{A}\bar{B}, \bar{A}B, A\bar{B}, AB$ corresponding to 00, 01, 10, 11 – logical combinations
- ✓ n = 3: possible $2^3$ = 8 variable combinations = $\bar{A}\bar{B}\bar{C}, \bar{A}\bar{B}C, \bar{A}B\bar{C}, \bar{A}BC, A\bar{B}\bar{C}, A\bar{B}C, AB\bar{C}, ABC$ - *000, 001, 010, 011, 100, 101, 110, 111*
- ✓ n = 4: possible $2^4$ = 16 variable combinations

$\bar{A}\bar{B}: 00 \Rightarrow 0$
$\bar{A}B: 01 \Rightarrow 1$
$A\bar{B}: 10 \Rightarrow 2$
$AB: 11 \Rightarrow 3$

| A | B | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

$\Rightarrow F = \overline{A}\overline{B} + \overline{A}B$

$\bar{A}\bar{B}\bar{C}: 000 \Rightarrow 0$
$\bar{A}\bar{B}C: 001 \Rightarrow 1$
$\bar{A}B\bar{C}: 010 \Rightarrow 2$
$\bar{A}BC: 011 \Rightarrow 3$
$A\bar{B}\bar{C}: 100 \Rightarrow 4$
$A\bar{B}C: 101 \Rightarrow 5$
$AB\bar{C}: 110 \Rightarrow 6$
$ABC: 111 \Rightarrow 7$

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

$\Rightarrow F = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + AB\overline{C} + ABC$

| A | B | C | D | F |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |

| A | B | C | D | F |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

$\Rightarrow F = ?$

- ✓ Question: 1) If function F is given. Show the True table. 2) If $F = A + BC$, show its true table. 3) Show $\overline{F}$

# 4.2. Representation of a logic function (cont.)

## b) Logic expression: canonical form vs non-canonical form

- ✓ Sum of Product (SoP): sum of minterm $m_i$, where i is the order of the corresponding variable combination
  - Example: $\mathrm{F} = \overline{\mathrm{A}}\overline{\mathrm{B}} + \overline{\mathrm{A}}\mathrm{B} = \mathrm{m}_0 + \mathrm{m}_1$ or $\mathrm{F} = \{\mathrm{m}_0, \mathrm{m}_1\}$; $\mathrm{F} = \overline{\mathrm{A}}\overline{\mathrm{B}}\overline{\mathrm{C}} + \overline{\mathrm{A}}\overline{\mathrm{B}}\mathrm{C} + \mathrm{A}\mathrm{B}\overline{\mathrm{C}} + \mathrm{ABC} = \mathrm{m}_0 + \mathrm{m}_1 + \mathrm{m}_6 + \mathrm{m}_7$ or $\mathrm{F} = \{\mathrm{m}_0, \mathrm{m}_1, \mathrm{m}_6, \mathrm{m}_7\}$
  - Question:

  1) If $\mathrm{F} = \{\mathrm{m}_1, \mathrm{m}_3, \mathrm{m}_5, \mathrm{m}_7, \mathrm{m}_8, \mathrm{m}_{10}, \mathrm{m}_{12}, \mathrm{m}_{14}\}$, show the canonical form of F and Truth table

  2) If $F = A + BC$ show the canonical SoP of F and its Truth table

- ✓ Product of Sum (PoS): Product of maxterm $M_i$, where $m_i = \overline{M_i}$ or $M_i = \overline{m_i}$
  - ✓ Example: $\mathrm{F} = \mathrm{M}_0\mathrm{M}_1 = \overline{\mathrm{m}_0}.\overline{\mathrm{m}_1} = \overline{\overline{\mathrm{A}}\overline{\mathrm{B}}}.\overline{\overline{\mathrm{A}}\mathrm{B}} = (\mathrm{A} + \mathrm{B}).(\mathrm{A} + \overline{\mathrm{B}}) \Rightarrow \mathrm{M}_0 = \mathrm{A} + \mathrm{B}, \mathrm{M}_1 = \mathrm{A} + \overline{\mathrm{B}}$
  - ✓ Question:

  1) If canonical SoP of $\mathrm{F} = \{\mathrm{m}_0, \mathrm{m}_1, \mathrm{m}_6, \mathrm{m}_7\}$, show its canonical PoS

  2) If $\mathrm{F} = (\mathrm{A} + \mathrm{B})(\mathrm{B} + \mathrm{C})(\mathrm{A} + \mathrm{C})$, determine the canonical PoS of F

- ✓ Non-canonical form: OR-AND or AND-OR ➔ convert to canonical form
  - ✓ Question:

  1) $\mathrm{F} = \mathrm{A} + \mathrm{BC}$ ➔ SoP? PoS?

  2) $\mathrm{F} = (\mathrm{A} + \mathrm{B})(\mathrm{B} + \mathrm{C})(\mathrm{A} + \mathrm{C})$ ➔ SoP, PoS?

  3) $\mathrm{F} = \overline{(\mathrm{A}\overline{\mathrm{B}} + \overline{\mathrm{A}}\mathrm{B})\,\overline{\overline{\mathrm{B}}(\overline{\mathrm{A}}\mathrm{CD} + \overline{\mathrm{A}}\mathrm{C}\overline{\mathrm{D}})}}$ ➔ SoP, PoS?

c) Karnaugh table: arrange minterms in array where number of cells is $2^n$ for n-variable function

✓ Karnaugh table of logic variables: gray code and neighborhood property

| A \ B | 0 | 1 | |
|---|---|---|---|
| 0 | $m_0$ | $m_1$ | $\overline{A}$ |
| 1 | $m_2$ | $m_3$ | A |
| | $\overline{B}$ | B | |

| | $\overline{C}$ | C | | $\overline{C}$ | |
|---|---|---|---|---|---|
| A \ BC | 00 | 01 | 11 | 10 | |
| 0 | $m_0$ | $m_1$ | $m_3$ | $m_2$ | $\overline{A}$ |
| 1 | $m_4$ | $m_5$ | $m_7$ | $m_6$ | A |
| | $\overline{B}$ | | B | | |

| | | $\overline{D}$ | D | | $\overline{D}$ | |
|---|---|---|---|---|---|---|
| | AB \ CD | 00 | 01 | 11 | 10 | |
| $\overline{B}$ | 00 | $m_0$ | $m_1$ | $m_3$ | $m_2$ | $\overline{A}$ |
| | 01 | $m_4$ | $m_5$ | $m_7$ | $m_6$ | |
| B | 11 | $m_{12}$ | $m_{13}$ | $m_{15}$ | $m_{14}$ | A |
| $\overline{B}$ | 10 | $m_8$ | $m_9$ | $m_{11}$ | $m_{10}$ | |
| | | $\overline{C}$ | | C | | |

$\bar{A} = m_0 + m_1 = \overline{A}\,\overline{B} + \overline{A}B$

$A = m_2 + m_3 = A\overline{B} + AB$

$\bar{B} = m_0 + m_2 = \overline{A}\,\overline{B} + A\overline{B}$

$B = m_1 + m_3 = \overline{A}B + AB$

$\bar{A} = m_0 + m_1 + m_3 + m_2 = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,C + \overline{A}BC + \overline{A}B\,\overline{C}$

$\overline{A} = m_4 + m_5 + m_7 + m_6 = A\,\overline{B}\,\overline{C} + A\,\overline{B}\,C + ABC + AB\overline{C}$

$B = m_4 + m_5 + m_7 + m_6 = A\,\overline{B}\,\overline{C} + A\,\overline{B}\,C + ABC + AB\overline{C}$

$BC = m_3 + m_7 = \overline{A}BC + ABC = BC$

$AC = m_5 + m_7 = A\overline{B}C + ABC = AC$

$\overline{A}B = m_3 + m_2 = \overline{A}BC + \overline{A}B\overline{C} = \overline{A}B$

$A =?\ B = ?\ C =?\ D = ?$

$AB =?\ BD = ?\ BC =?\ AD = ?$

$A\overline{B} =?\ \overline{B}D = ?\ B\overline{C} =?\ \overline{A}\ \overline{D} = ?$

$ABC =?\ A\overline{B}D = ?\ B\overline{C}D =?\ \overline{A}B\overline{D} = ?$

$\overline{A}\ \overline{B}D =?\ \overline{B}\ \overline{C}D = ?\ A\overline{C}D =?\ \overline{B}C\overline{D} = ?$

# 4.2. Representation of a logic function (cont.)

✓ Karnaugh table of logic function:

- Convert the logic function just to OR-AND form
- Determine the number of logic variables and show the Karnaugh table
- Fill 1’s in Karnaugh table in the cells corresponding to each product in OR-AND form
- Other cells will be filled by 0’s
- Karnaugh table is ready
- Example:

$F = A + B$

| F: A \ B | 0 | 1 | |
|---|---|---|---|
| 0 | 0 | 1 | $\overline{A}$ |
| 1 | 1 | 1 | A |
| | $\overline{B}$ | B | |

$F = \overline{A}B + BC + AC$

| F: A \ BC | 00 ($\overline{C}$) | 01 (C) | 11 (C) | 10 ($\overline{C}$) | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | $\overline{A}$ |
| 1 | 0 | 1 | 1 | 0 | A |
| | $\overline{B}$ | $\overline{B}$ | B | B | |

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

| F: A \ BC | 00 ($\overline{C}$) | 01 (C) | 11 (C) | 10 ($\overline{C}$) | |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | $\overline{A}$ |
| 1 | 1 | 1 | 1 | 1 | A |
| | $\overline{B}$ | $\overline{B}$ | B | B | |

$F = (A + B)(B + C)(A + C)$

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

Hint: Karnaugh table of $\overline{F}$, then ➔ F

| F: A \ BC | 00 ($\overline{C}$) | 01 (C) | 11 (C) | 10 ($\overline{C}$) | |
|---|---|---|---|---|---|
| 0 | | | | | $\overline{A}$ |
| 1 | | | | | A |
| | $\overline{B}$ | $\overline{B}$ | B | B | |

| F: A \ BC | 00 ($\overline{C}$) | 01 (C) | 11 (C) | 10 ($\overline{C}$) | |
|---|---|---|---|---|---|
| 0 | | | | | $\overline{A}$ |
| 1 | | | | | A |
| | $\overline{B}$ | $\overline{B}$ | B | B | |

- Question:

1) Determine Karnaugh table of $\overline{F}$
2) Determine canonical form of $\overline{F}$

# 4.2. Representation of a logic function (cont.)

- Question: Determine Karnaugh tables of the following logic functions

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

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

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

| F (AB \ CD) | $\overline{D}$ 00 | D 01 | D 11 | $\overline{D}$ 10 | |
|---|---|---|---|---|---|
| $\overline{B}$ 00 | | | | | $\overline{A}$ |
| B 01 | | | | | $\overline{A}$ |
| B 11 | | | | | A |
| $\overline{B}$ 10 | | | | | A |
| | $\overline{C}$ | $\overline{C}$ | C | C | |

| F (AB \ CD) | $\overline{D}$ 00 | D 01 | D 11 | $\overline{D}$ 10 | |
|---|---|---|---|---|---|
| $\overline{B}$ 00 | | | | | $\overline{A}$ |
| B 01 | | | | | $\overline{A}$ |
| B 11 | | | | | A |
| $\overline{B}$ 10 | | | | | A |
| | $\overline{C}$ | $\overline{C}$ | C | C | |

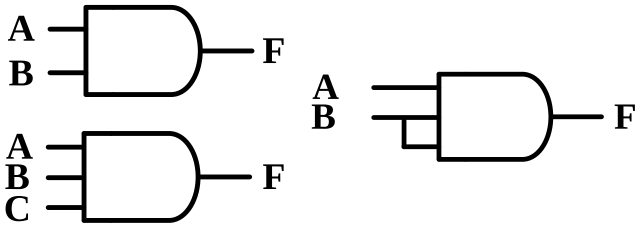
| F (AB \ CD) | $\overline{D}$ 00 | D 01 | D 11 | $\overline{D}$ 10 | |
|---|---|---|---|---|---|
| $\overline{B}$ 00 | | | | | $\overline{A}$ |
| B 01 | | | | | $\overline{A}$ |
| B 11 | | | | | A |
| $\overline{B}$ 10 | | | | | A |
| | $\overline{C}$ | $\overline{C}$ | C | C | |

# 4.3. Basic logical gates

## a. Example for 2-input:

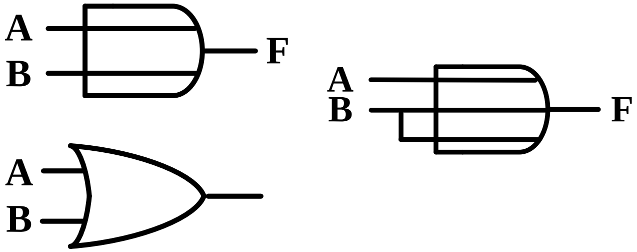
✓ AND: $\mathrm{F} = \mathrm{AB}$

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



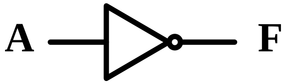

✓ OR: $\mathrm{F} = \mathrm{A} + \mathrm{B}$

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



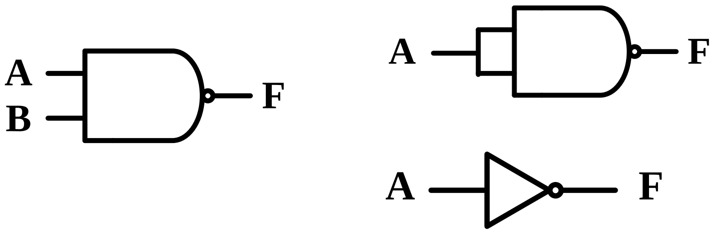

✓ NOT: $\mathrm{F} = \overline{\mathrm{A}}$

| A | F |
|---|---|
| 0 | 1 |
| 1 | 0 |



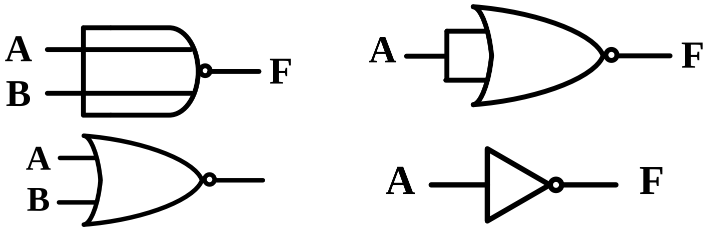

✓ NAND: $\mathrm{F} = \overline{\mathrm{A.B}}$

| A | B | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



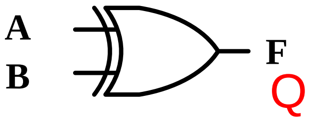

✓ NOR: $\mathrm{F} = \overline{A + B}$

| A | B | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |




✓ XOR: $\mathrm{F} = \mathrm{A}\overline{\mathrm{B}} + \overline{\mathrm{A}}\mathrm{B}$

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |




Question: 1) Implement function $F = \overline{A.B.C}$ using 2-input NAND gates

Question: 2) Implement function $F = \overline{A + B + C}$ using 2-input NOR gates

# 4. Minimization of logic function

a. Type of logic function expression: utilization in function implementation using desired gates

✓ OR-AND: $F = \overline{A}B + BC + AC = X + Y + Z$

✓ AND-OR: $F = (A + B)(B + C)(A + C) = X.Y.Z$

✓ NAND-NAND: $F = \overline{\overline{\overline{A}B}.\overline{BC}.\overline{AC}} = \overline{X.Y.Z}$

✓ NOR-NOR: $F = \overline{\overline{A + B} + \overline{B + C} + \overline{A + C}} = \overline{X + Y + Z}$

b. Optimization concept

✓ Output optimization in OR-AND form

✓ Fewest number of products in the OR-AND form

✓ Fewest number of variables/negated variables in each product above

✓ Optimization methods:

- Algebraic method: Utilization of Boolean algebra to minimize a logic function, but not guaranteed
- Karnaugh-table based method: Obvious and intuitive minimization of a logic function

# 4.4. Minimization of a logic function (cont.)

a. Algebraic optimization method:

✓ Utilization of properties in Boolean algebra and flexible use of De Morgan theorem

✓ Example:

- $F = A\overline{B}C + A\overline{B}\,\overline{C} = A\overline{B}(C + \overline{C}) = A\overline{B}$
- $F = A(BC + \overline{B}\,\overline{C}) + A(\overline{BC + \overline{B}\,\overline{C}}) = A((BC + \overline{B}\,\overline{C}) + \left(\overline{BC + \overline{B}\,\overline{C}}\right)) = A$
- $F = A\overline{B} + A\overline{B}CD(\overline{AB} + \overline{BD}) = A\overline{B}(1 + CD[\overline{AB} + \overline{BD}] = A\overline{B}$

b. Karnaugh-table based method: Follow the following steps

- Step 1: Represent logic function F in Karnaugh table (KT)
- Step 2: Group $2^k$ neighbor cells containing 1's following the rules (of course, always there is exception!)
    - Group size as largest as possible
    - Groups must not be nested
    - All 1's cells in KT must be included in a group
    - 1's cells may be re-used in grouping, however these cells must be included in different groups
- Step 3: Assign each group to the corresponding product of variable/negated variables
- Step 4: The minimization function is an OR-AND form of the products above
- Notes: The corner cells are neighbors; the end-most cells in a row or a column are also neighbors.

# 4.4. Minimization of a logic function (cont.)

- Example:

| F (A \ B) | $\overline{B}$ 0 | B 1 |
|---|---|---|
| $\overline{A}$ 0 | 1 | 0 |
| A 1 | 1 | 1 |

| F (A \ BC) | $\overline{C}$ 00 | C 01 | C 11 | $\overline{C}$ 10 |
|---|---|---|---|---|
| $\overline{A}$ 0 | 0 | 1 | 1 | 0 |
| A 1 | 1 | 0 | 0 | 1 |
| | $\overline{B}$ | $\overline{B}$ | B | B |

| F (AB \ CD) | $\overline{D}$ 00 | D 01 | D 11 | $\overline{D}$ 10 |
|---|---|---|---|---|
| $\overline{A}$ $\overline{B}$ 00 | 1 | 1 | 0 | 0 |
| $\overline{A}$ B 01 | 1 | 1 | 0 | 0 |
| A B 11 | 1 | 1 | 1 | 1 |
| A $\overline{B}$ 10 | 1 | 1 | 1 | 1 |
| | $\overline{C}$ | $\overline{C}$ | C | C |

| AB \ CD | $\overline{D}$ 00 | D 01 | D 11 | $\overline{D}$ 10 |
|---|---|---|---|---|
| $\overline{A}$ $\overline{B}$ 00 | 1 | 1 | 0 | 1 |
| $\overline{A}$ B 01 | 1 | 1 | 0 | 0 |
| A B 11 | 0 | 0 | 1 | 1 |
| A $\overline{B}$ 10 | 1 | 0 | 1 | 1 |
| | $\overline{C}$ | $\overline{C}$ | C | C |

| F (AB \ CD) | $\overline{D}$ 00 | D 01 | D 11 | $\overline{D}$ 10 |
|---|---|---|---|---|
| $\overline{A}$ $\overline{B}$ 00 | 1 | 0 | 1 | 1 |
| $\overline{A}$ B 01 | 1 | 0 | 1 | 0 |
| A B 11 | 0 | 1 | 1 | 1 |
| A $\overline{B}$ 10 | 1 | 0 | 1 | 1 |
| | $\overline{C}$ | $\overline{C}$ | C | C |

| F (A \ BC) | $\overline{C}$ 00 | C 01 | C 11 | $\overline{C}$ 10 |
|---|---|---|---|---|
| $\overline{A}$ 0 | 1 | 1 | 1 | 0 |
| A 1 | 1 | 1 | 1 | 0 |
| | $\overline{B}$ | $\overline{B}$ | B | B |

| F (A \ BC) | $\overline{C}$ 00 | C 01 | C 11 | $\overline{C}$ 10 |
|---|---|---|---|---|
| $\overline{A}$ 0 | 0 | 0 | 1 | 1 |
| A 1 | 0 | 1 | 1 | 0 |
| | $\overline{B}$ | $\overline{B}$ | B | B |

| F (AB \ CD) | $\overline{D}$ 00 | D 01 | D 11 | $\overline{D}$ 10 |
|---|---|---|---|---|
| $\overline{A}$ $\overline{B}$ 00 | 0 | 1 | 0 | 0 |
| $\overline{A}$ B 01 | 1 | 1 | 0 | 1 |
| A B 11 | 1 | 1 | 1 | 0 |
| A $\overline{B}$ 10 | 1 | 0 | 0 | 0 |
| | $\overline{C}$ | $\overline{C}$ | C | C |

| F (AB \ CD) | $\overline{D}$ 00 | D 01 | D 11 | $\overline{D}$ 10 |
|---|---|---|---|---|
| $\overline{A}$ $\overline{B}$ 00 | 1 | 1 | 0 | 1 |
| $\overline{A}$ B 01 | 0 | 1 | 1 | 1 |
| A B 11 | 1 | 1 | 1 | 0 |
| A $\overline{B}$ 10 | 1 | 0 | 1 | 1 |
| | $\overline{C}$ | $\overline{C}$ | C | C |

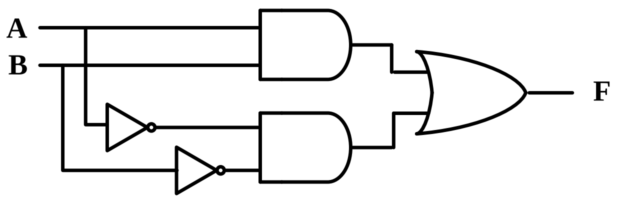
| F (AB \ CD) | $\overline{D}$ 00 | D 01 | D 11 | $\overline{D}$ 10 |
|---|---|---|---|---|
| $\overline{A}$ $\overline{B}$ 00 | 1 | 0 | 0 | 1 |
| $\overline{A}$ B 01 | 1 | 1 | 0 | 1 |
| A B 11 | 0 | 1 | 1 | 0 |
| A $\overline{B}$ 10 | 1 | 0 | 1 | 1 |
| | $\overline{C}$ | $\overline{C}$ | C | C |

Question: Minimization of $\overline{F}$

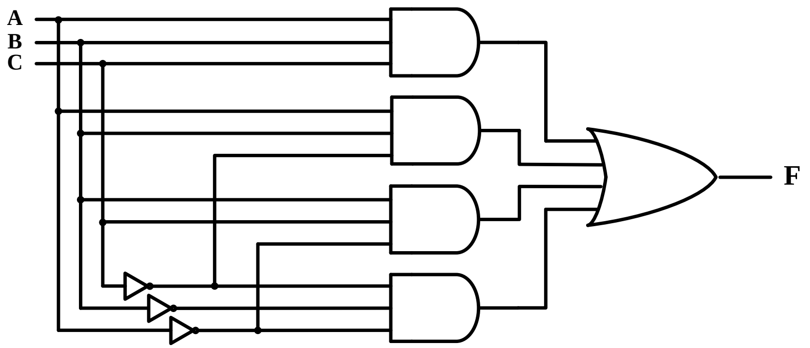
# 4.5. Implementation of a logic function using logic gates

a. OR-AND expression form: Typical output from minimization process

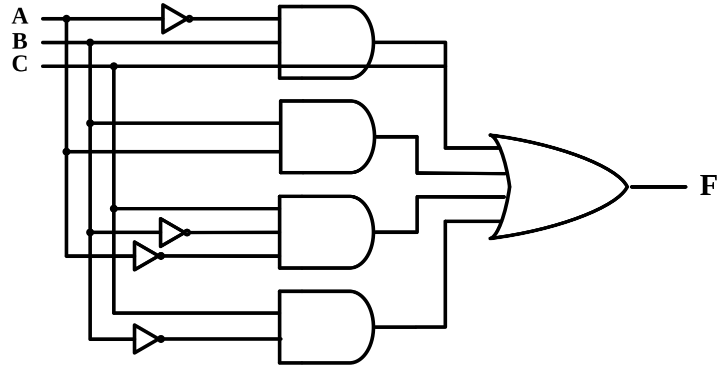
$$F = AB + \bar{A}\,\bar{B}$$




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

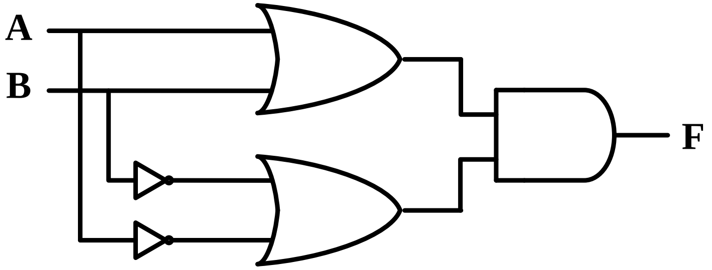



$$F = \bar{A}BC + AB + \bar{A}\bar{B}C + B\bar{C}$$

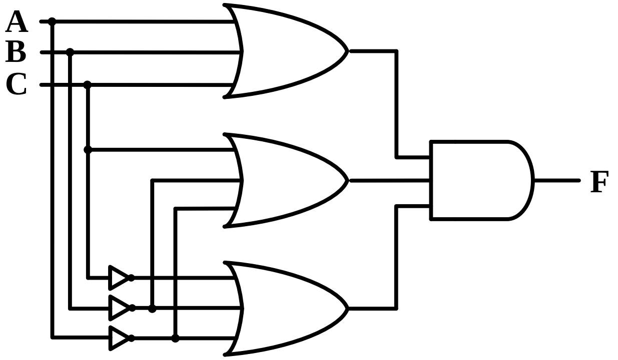



b. AND-OR expression:

$$F = (A + B)(\bar{A} + \bar{B})$$



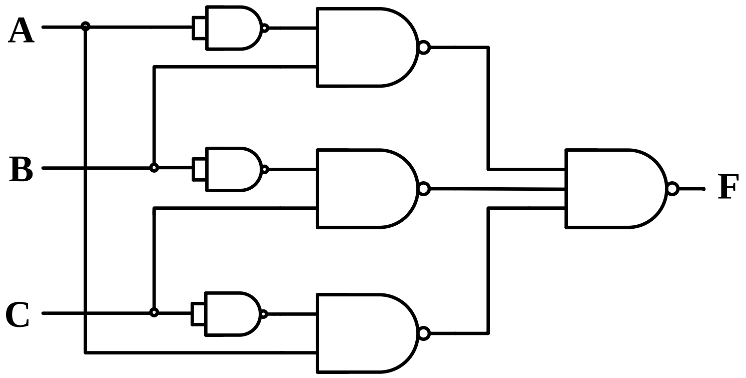

$$F = (A + B)(B + C)(A + C)$$



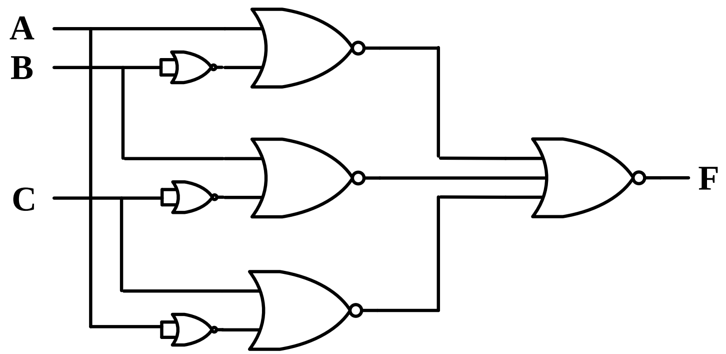
# 4.5. Implementation of a logic function using logic gates (cont.)

c. NAND-NAND expression: Derive to OR-AND expression and apply De Morgan to get NAND-NAND form

$\mathrm{OR-AND:}\ F = \bar{A}B + \bar{B}C + A\bar{C} = \overline{\overline{\bar{A}B + \bar{B}C + A\bar{C}}} = \overline{\overline{\bar{A}B}.\overline{\bar{B}C}.\overline{A\bar{C}}}$




Question: Implement function F using NOR gate




d. NOR-NOR expression: Derive OR-AND of $\bar{F}$ .
Take negation of $\bar{F}$ to get $\bar{\bar{F}} = F$, and apply De Morgan 1st time. Take involution of F to result $\bar{\bar{F}}$ and apply De Morgan 2nd time for first negation to get NOR-NOR form

Question: Implement function $F_{\text{NOR-NOR}}$ using 2-input NOR gates only

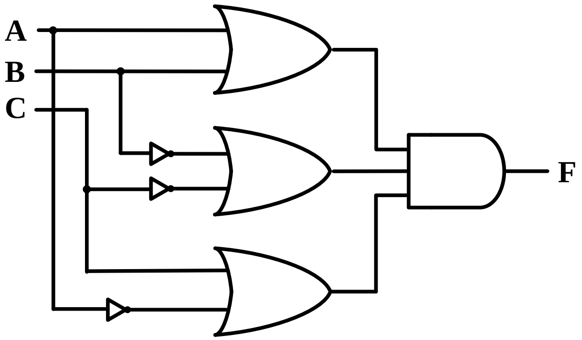
$\mathrm{OR-AND\ of}\ \bar{F} = \bar{A}B + \bar{B}C + A\bar{C}$

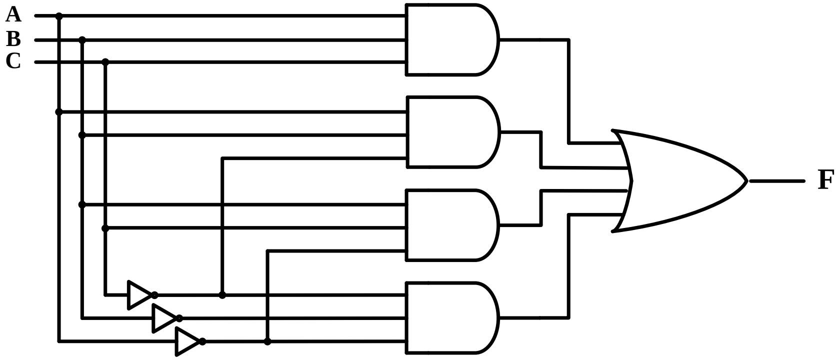
$\mathrm{Negation\ of}\ \bar{F} \Rightarrow \bar{\bar{F}} = F = \overline{\bar{A}B + \bar{B}C + A\bar{C}} = \overline{\bar{A}B}.\overline{\bar{B}C}.\overline{A\bar{C}} = (A + \bar{B})(B + \bar{C})(\bar{A} + C)$

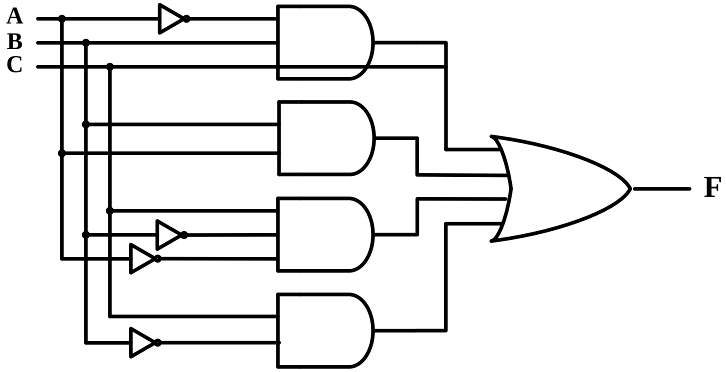
$\mathrm{Involution\ of}\ F \Rightarrow \bar{\bar{F}} = F = \overline{\overline{(A + \bar{B})(B + \bar{C})(\bar{A} + C)}} = \overline{\overline{A + \bar{B}} + \overline{B + \bar{C}} + \overline{\bar{A} + C}}$

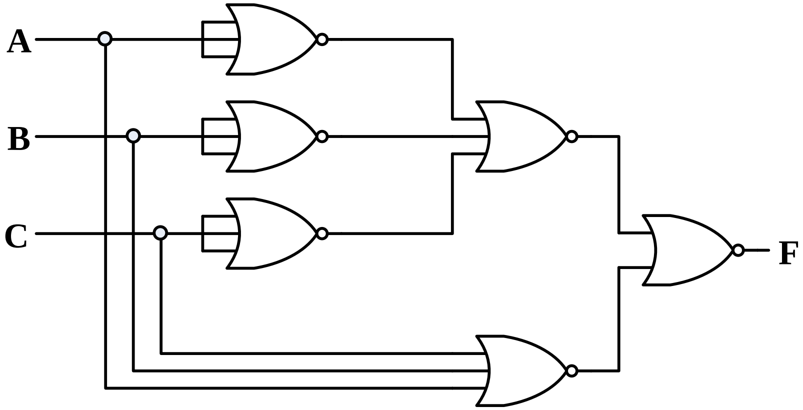
# 4.5. Implementation of a logic function using logic gates (cont.)

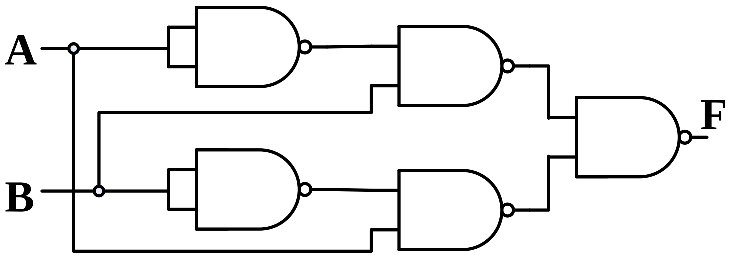
- Example: Determine logic function of the following logic circuits

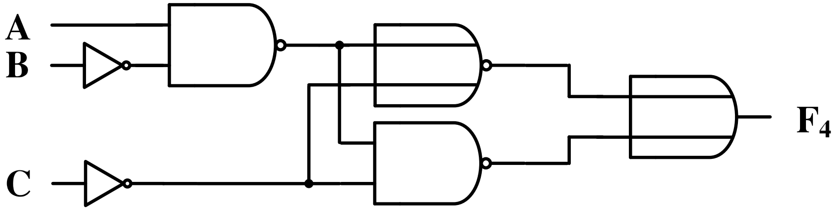


# 4.6. Some typical logic applications

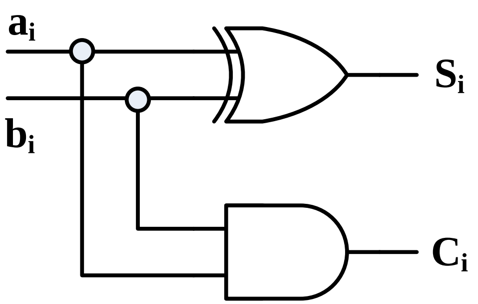
a. One-bit half adder: Determine Truth table

- $a_i, b_i$ : i-th bits of binary numbers A and B to be added
- $S_i$: Sum bit, adding bits $a_i$ and $b_i$
- $C_i$: carry bit

$$S_i = \bar{a}_i b_i + a_i \bar{b}_i$$

$$C_i = a_i b_i$$

| A | B | $S_i$ | $C_i$ |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |

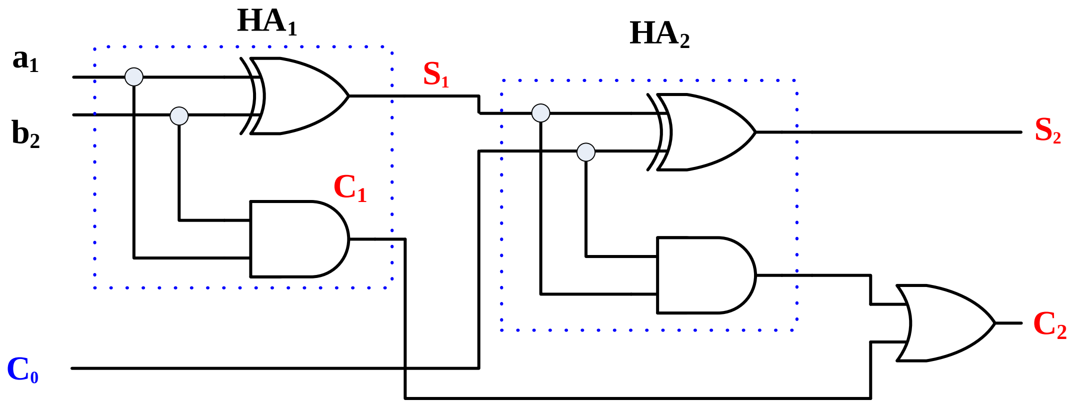



b. One-bit full adder: Determine Truth table

$S_i =$

$C_i =$

| $a_i$ | $b_i$ | $C_{i-1}$ | $S_i$ | $C_i$ |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |




Question: 8-bit adder implementation

# 4.6. Some typical logic applications

a. Comparator: Compare two binary number
b. Binary Coded Decimal (BCD): encode numbers 0-9 to corresponding binary outputs
c. 7-segment coder: encode number 0-9 to monitor in led-segments
d. Carry-Look-Ahead adder (CLA): Fast implementation of an n-bit adder

HUST

hust.edu.vn fb.com/dhbkhn

HUST

hust.edu.vn fb.com/dhbkhn

# THANK YOU !