BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

# ĐẠI HỌC BÁCH KHOA HÀ NỘI
# VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# Electronics for Information Technology

**(Điện tử cho Công nghệ Thông tin)**

IT3420E


Đỗ Công Thuần

Department of Computer Engineering

Email: thuandc@soict.hust.edu.vn

# General Information

- Course: **Electronics for Information Technology**
- ID Number: IT3420
- Credits: 2 (2-1-0-4)
- Lecture/Exercise: 32/16 hours (48 hours, 16 weeks)
- Evaluation:
    - Midterm examination and weekly assignment: **50%**
    - Final examination: **50%**
- Learning Materials:
    - Lecture slides
    - Textbooks
        - ***Introductory Circuit Analysis*** (2015), 10th – 13th ed., Robert L. Boylestad
        - ***Electronic Device and Circuit Theory*** (2013), 11th ed., Robert L. Boylestad, Louis Nashelsky
        - ***Microelectronics Circuit Analysis and Design*** (2006), 4th ed., Donald A. Neamen
        - ***Digital Electronics: Principles, Devices and Applications*** (2007), Anil K. Maini

# Contact Your Instructor

- You can reach me through office in **Room 802, B1 Building**, HUST.
  - You should make an appointment by email before coming.
  - If you have urgent things, just come and meet me!
- You can also reach me at the following **email** any time. This is the best way to reach me!
  - thuandc@soict.hust.edu.vn

# Course Contents

- The Concepts of Electronics for IT
- **Chapter 1**: Passive Electronic Components and Applications
- **Chapter 2**: Semiconductor Components and Applications
- **Chapter 3**: Operational Amplifiers
- **Chapter 4**: Fundamentals of Digital Circuits
- **Chapter 5**: Logic Gates
- **Chapter 6**: Combinational Logic
- **Chapter 7**: Sequential Logic

# Chapter 5:
# **Logic Gates**

1. Introduction to Boolean Algebra
2. Variables, Literals & Terms in Boolean Expressions
3. Laws and Rules of Boolean Algebra
4. Logic Simplification
5. Basic Logic Gates

*References:*

***Digital electronics: Principles, Devices, and Applications, Anil Kumar Maini 2007 John Wiley & Sons***

***Fundamentals of Logic Design, Seventh Edition, Charles H. Roth, Jr. and Larry L. Kinney***

***Digital Fundamentals, Thomas L. Floyd, Eleventh Edition, Pearson Education Limited 2015***

# Contents

# Contents

# Introduction to Boolean Algebra

- Boolean algebra was introduced by George Boole in the 1850s.
- A mathematical system for formulating logic statements with symbols so that problems can be written and solved in a manner similar to ordinary algebra.
- Boolean algebra is applied in the **design and analysis of digital systems**.

# Introduction to Boolean Algebra

- In a digital circuit (logic circuit), the signal has only one of two discrete levels at a time.
  - Each level is interpreted as one of two different states (on/off, true/false, 1/0, ...).
  - Ex: $0 \rightarrow 0.8V$ : logic 0
    $2.5 \rightarrow 5V$ : logic 1

**Boolean algebra can be efficiently used to analyze and design digital systems.**

# Introduction to Boolean Algebra

- **Binary variable** is used to represent the input or output of a switching circuit, and can take on only 2 different values, “0” and “1”.
- **Boolean function/expression** consists of binary variables, the constants “0” and “1”, and the logic operation symbols. A Boolean function also have 2 different values, “0” and “1”.
- **3 basic logic functions**:
  - “AND”
  - “OR”
  - “NOT”

# Introduction to Boolean Algebra

| Logic 0 | Logic 1 |
|---|---|
| False | True |
| Off | On |
| Low | High |
| No | Yes |
| Open switch | Closed switch |

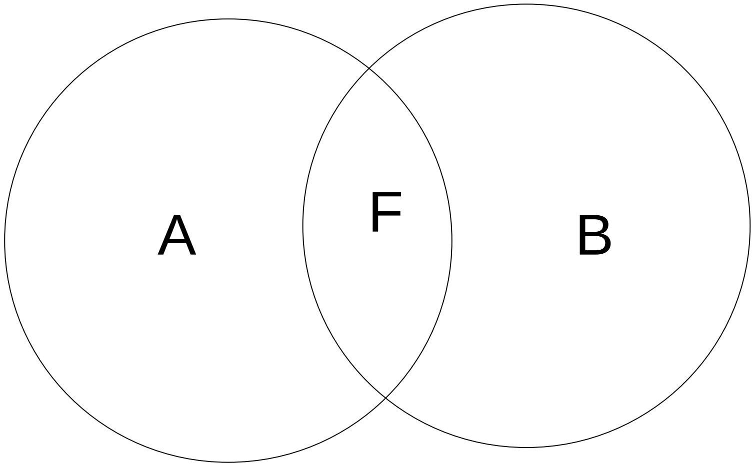
# Contents

# Variables, Literals & Terms in Boolean Expressions

- Venn Diagram
- Logic Function
- Truth Table
- Karnaugh Map
- Timing Diagram

# Venn Diagram

- A Venn diagram uses simple closed curves (circles or ellipses) drawn on a plane to represent sets.
- It shows all possible logical relations between a finite collection of different sets.
- Each logic varible is composed of 2 different sets, corresponding to *Logic 0* and *Logic 1*.
- Example: F = A **AND** B



# Logic Function

- Basic functions:
  - **AND**, **.**
  - **OR**, **+**
  - **NOT**, $\overline{\phantom{A}}$

- Examples:
  - F = A AND B or F = A.B
  - F = A OR B or F = A+B
  - F = NOT(A) or F = $\bar{A}$

# Truth Table

- A true table lists all possible combinations of input binary variables and the corresponding outputs of a logic system.
- If a logic circuit has ***n*** binary inputs, its truth table will have:
  - $2^n$ possible input combinations, or $2^n$ rows
  - (n+1) columns

$$F = A+B$$

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

# Karnaugh Map

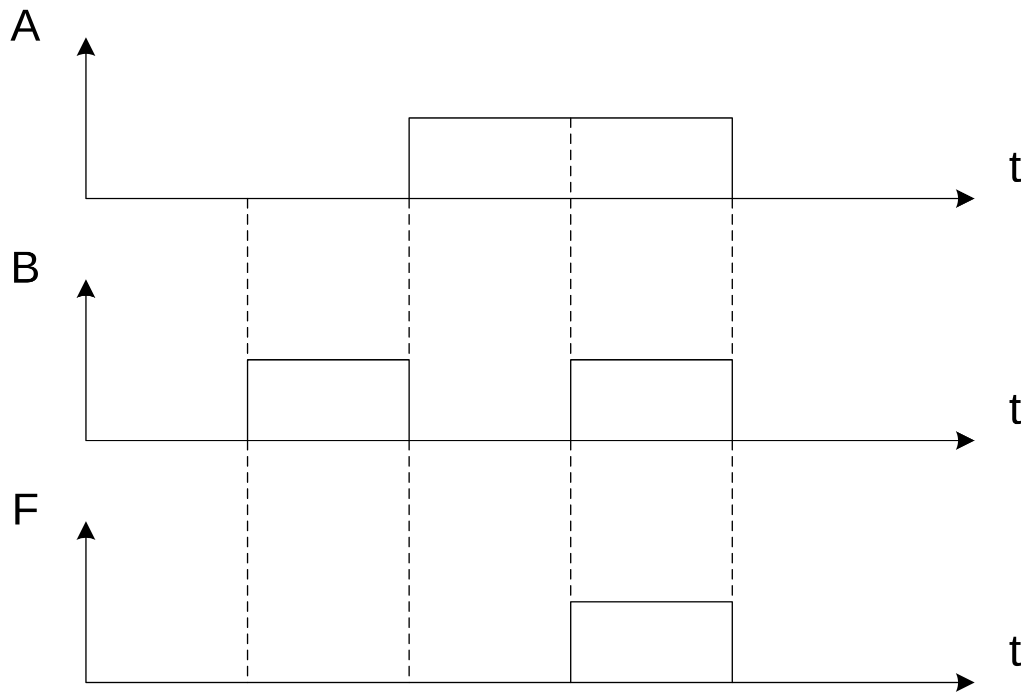
- A Karnaugh map is a graphical representation of the logic system.
- It is an array of cells in which each cell represents a binary value of the input variables.
- The cells are arranged in a way so that simplification of a given expression is simply a matter of properly grouping the cells.

$$F = A.B$$

| a\b | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

# Timing Diagram

- A timing diagram is a graphical representation of a set of signals in the time domain.
- Example, for **F = A.B**:



# Contents

# Laws and Rules of Boolean Algebra

- Equivalent, Complement, Dual of Boolean Expression
- 10 postulates
- 17 basic theorems **(Homework)**

# Equivalent of Boolean Expression

- Two given Boolean expressions are said to be equivalent if **one of them equals '1' only when the other equals '1'** and also **one equals '0' only when the other equals '0'**.

# Complement of Boolean Expression

- Two given Boolean expressions are said to be the complement of each other if one expression equals '1' only when the other equals '0', and vice versa.
- **Method**: complementing each literal, changing
- all '.' to '+' and all '+' to '.', all 0s to 1s and all 1s to 0s.
  - Given expression:
    $\overline{A}.B + A.\overline{B}$
  - Corres. complement:
    $(A + \overline{B}).(\overline{A} + B)$
  - Given expression:
    $\overline{A}.\overline{B} + A.B$
  - Corres. complement:
    $(A + B).(\overline{A} + \overline{B})$

# Example 1

- Find the complement expression of:

$$[(A.\overline{B}+\overline{C}).D+\overline{E}].F$$

✓Corresponding complement:

$$[(\overline{A}+B).C+\overline{D}].E+\overline{F}$$

# Dual of Boolean Expression

- The dual of a Boolean expression is obtained by replacing all '.' operations with '+' operations, all '+' operations with '.' operations, all 0s with 1s and all 1s with 0s and leaving all literals unchanged.

- Given expressison:

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

- Corres. dual:

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

- Given expressison:

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

- Corres. dual:

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

# Example 2

- Find the dual of:

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

✓Corresponding dual:

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

# Example 3

- Simplify:

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

✓Solution:

$$(A.B + C.D).[(\overline{A} + \overline{B}).(\overline{C} + \overline{D})] = 0$$

# Postulates

- $1 \times 1 = 1$
- $1 \times 0 = 0$
- $0 \times 1 = 0$
- $0 \times 0 = 0$
- $\overline{0} = 1$
- $\overline{1} = 0$

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 1$

# Theorems of Boolean Algebra

- Theorem 1: (Operations with '0' and '1')

$$\text{(a) } 0.X = 0$$

$$\text{(b) } 1 + X = 1$$

- Proof:

$$X = 0$$

$$\text{LHS} = 0.X = 0.0 = 0 = \text{RHS}$$

$$X = 1$$

$$\text{LHS} = 0.1 = 0 = \text{RHS}$$

# Theorems of Boolean Algebra

- Theorem 2 (Operations with ‘0’ and ‘1’)

$$\text{(a) } 1.X = X$$

$$\text{(b) } 0 + X = X$$

- Proof:

$$X = 0 \qquad \text{LHS} = 1.0 = 0 = \text{RHS}$$

$$X = 1 \qquad \text{LHS} = 1.1 = 1 = \text{RHS}$$

# Theorems of Boolean Algebra

- Theorem 3 (Idempotent or Identity Laws)

$$\text{(a) } X.X.X.\dots.X = X$$

$$\text{(b) } X + X + X + \cdots + X = X$$

- Example:

$$\begin{aligned}(A.\overline{B}.\overline{B} + C.C).(A.\overline{B}.\overline{B} + A.\overline{B} + C.C) &= (A.\overline{B} + C).(A.\overline{B} + A.\overline{B} + C) \\ &= (A.\overline{B} + C).(A.\overline{B} + C) \\ &= A.\overline{B} + C\end{aligned}$$

# Theorems of Boolean Algebra

- Theorem 4 (Complementation Law)

$$(a)\ X.\overline{X} = 0$$

$$(b)\ X + \overline{X} = 1$$

- Proof:

$$X = 0,\ \overline{X} = 1$$

$$X.\overline{X} = 0.1 = 0$$

$$X = 1,\ \overline{X} = 0$$

$$X.\overline{X} = 1.0 = 0$$

- Further illustration:

$$(A + B.C)(\overline{A + B.C}) = 0$$

$$(A + B.C) + (\overline{A + B.C}) = 1$$

# Example 4

- Simplify the following:

$$[1 + L.M + L.\overline{M} + \overline{L}.M].[(L + \overline{M}).(\overline{L}.M) + \overline{L}.\overline{M}.(L + M)]$$

✓Solution:

$$1.(0 + 0) = 1.0 = 0.$$

# Theorems of Boolean Algebra

- Theorem 5 (Commutative Laws)

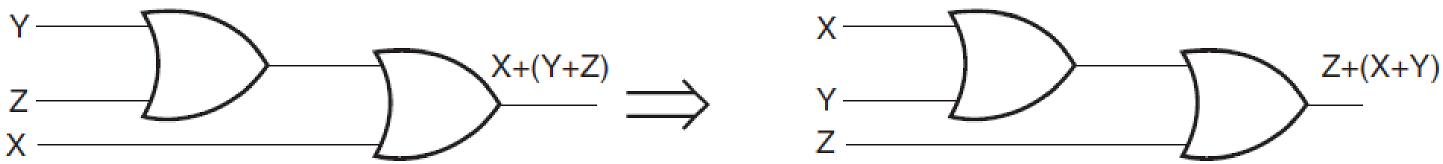
$$\text{(a) } X + Y = Y + X$$

$$\text{(b) } X.Y = Y.X$$

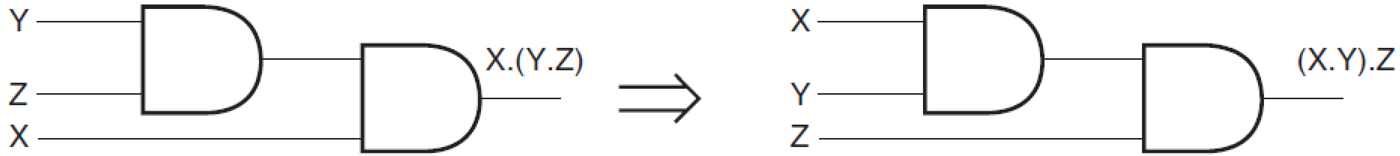
# Theorems of Boolean Algebra

- Theorem 6 (Associative Laws)

$$(a)\ X + (Y + Z) = Y + (Z + X) = Z + (X + Y)$$




$$(b)\ X.(Y.Z) = Y.(Z.X) = Z.(X.Y)$$



# Theorems of Boolean Algebra

- Theorem 7 (Distributive Laws)

$$\text{(a) } X.(Y+Z) = X.Y + X.Z$$

$$\text{(b) } X + Y.Z = (X+Y).(X+Z)$$

- Proof:

| X | Y | Z | Y+Z | XY | XZ | X(Y+Z) | XY+XZ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

# Example 5

- Simplify the following:

$$\overline{A}.\overline{B}+\overline{A}.B+A.\overline{B}+A.B=$$

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

$$=\overline{A}.1+A.1$$

$$=\overline{A}+A$$

$$=1$$

# Example 6

- Simplify the following:

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

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

$$=(\overline{A}+0).(A+0)$$

$$=\overline{A}.A$$

$$=0$$

# Theorems of Boolean Algebra

- Theorem 8:

$$\text{(a) } X.Y + X.\overline{Y} = X$$

$$\text{(b) } (X+Y).(X+\overline{Y}) = X$$

- Proof:

$$X.Y + X.\overline{Y} = X.(Y+\overline{Y}) = X.1 = X$$

$$(X+Y).(X+\overline{Y}) = X + Y.\overline{Y} = X + 0 = X$$

# Example 7

- Simplify the following:

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

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

✓Solution:

$$=A$$

# Theorems of Boolean Algebra

- Theorem 9:

$$\text{(a)}\ (X+\overline{Y}).Y = X.Y$$

$$\text{(b)}\ X.\overline{Y}+Y = X+Y$$

- Theorem 9(b) is the dual of theorem 9(a) and hence stands proved.

# Theorems of Boolean Algebra

- Theorem 10: (Absorption Law or Redundancy Law)

$$\text{(a)}\ X + X.Y = X$$

$$\text{(b)}\ X.(X + Y) = X$$

- Proof:

$$X + X.Y = X.(1 + Y) = X.1 = X$$

# Example 8

- Simplify the LHS:

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

- Simplify the LHS :

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

# Theorems of Boolean Algebra

- Theorem 11:

$$(a)\ Z.X + Z.\overline{X}.Y = Z.X + Z.Y$$

$$(b)\ (Z + X).(Z + \overline{X} + Y) = (Z + X).(Z + Y)$$

- Proof:

| X | Y | Z | ZX | ZY | $Z\overline{X}$ | $Z\overline{X}Y$ | $ZX + Z\overline{X}Y$ | ZX+ZY |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |

# Example 9

- Simplify the following:

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

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

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

# Theorems of Boolean Algebra

- Theorem 12 (Consensus Theorem)

$$\text{(a) } X.Y+\overline{X}.Z+Y.Z=X.Y+\overline{X}.Z$$

$$\text{(b) } (X+Y).(\overline{X}+Z).(Y+Z)=(X+Y).(\overline{X}+Z)$$

- Proof:

| X | Y | Z | XY | $\overline{X}Z$ | YZ | $XY+\overline{X}Z+YZ$ | $XY+\overline{X}Z$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |

# Example 10

- Simplify the following:

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

✓Solution:

$$= A.B.C + C.D$$

# Example 11

- Prove that:

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

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

$$= A.B$$

✓Solution:

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

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

# Theorems of Boolean Algebra

- Theorem 13 (DeMorgan's Theorem)

$$(a)\ [\overline{X_1 + X_2 + X_3 + \ldots + X_n}] = \overline{X_1}.\overline{X_2}.\overline{X_3}.\ \ldots\ .\overline{X_n}$$

$$(b)\ [\overline{X_1.X_2.X_3.\ \ldots\ .X_n}] = [\overline{X_1} + \overline{X_2} + \overline{X_3} + \ldots + \overline{X_n}]$$

- Proof:

$$\text{LHS} = [\overline{X_1 + X_2 + X_3 + \cdots + X_n}] = [\overline{0+0+0+\cdots+0}] = \overline{0} = 1$$

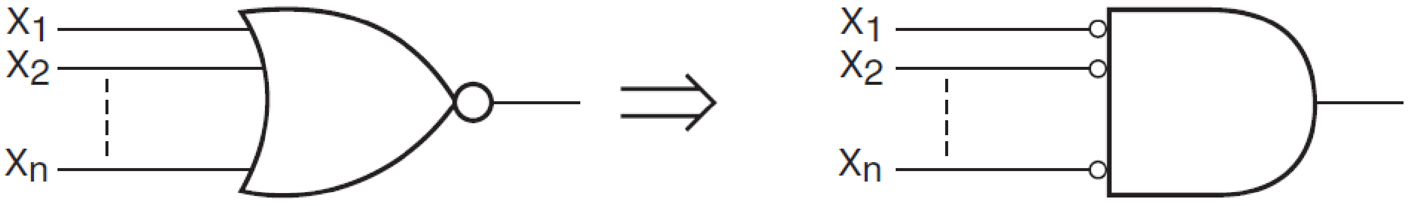
$$\text{RHS} = \overline{X_1}.\overline{X_2}.\overline{X_3}.\ \ldots\ .\overline{X_n} = \overline{0}.\overline{0}.\overline{0}.\ \ldots\ .\overline{0} = 1.1.1.\ \ldots\ .1 = 1$$

$$\text{LHS} = [\overline{X_1 + X_2 + X_3 + \cdots X_n}] = [\overline{1+0+0+\cdots+0}] = \overline{1} = 0$$

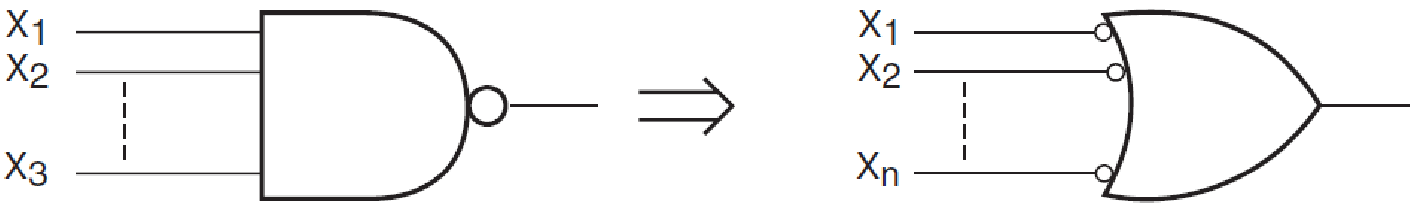
$$\text{RHS} = \overline{X_1}.\overline{X_2}.\overline{X_3}.\ \ldots\ .\overline{X_n} = \overline{1}.\overline{0}.\overline{0}.\ \ldots\ .\overline{0} = 0.1.1.\ \ldots\ .1 = 0$$

# Theorems of Boolean Algebra

- Theorem 13 (DeMorgan's Theorem)




(a)




(b)

# Theorems of Boolean Algebra

- Theorem 14 (Transposition Theorem)

$$\text{(a) } X.Y + \overline{X}.Z = (X+Z).(\overline{X}+Y)$$

$$\text{(b) } (X+Y).(\overline{X}+Z) = X.Z + \overline{X}.Y$$

- Proof:

| X | Y | Z | XY | $\overline{X}Z$ | X+Z | $\overline{X}+Y$ | $XY+\overline{X}Z$ | (X+Z)($\overline{X}+Y$) |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

# Example 12

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

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

# Theorems of Boolean Algebra

- Theorem15:

$$(a)\ X.f(X, \overline{X}, Y, Z, \ldots) = X.f(1, 0, Y, Z, \ldots)$$

$$(b)\ X + f(X, \overline{X}, Y, Z, \ldots) = X + f(0, 1, Y, Z, \ldots)$$

- Or:

$$(a)\ \overline{X}.f(X, \overline{X}, Y, Z, \ldots) = \overline{X}.f(0, 1, Y, Z, \ldots)$$

$$(b)\ \overline{X} + f(X, \overline{X}, Y, Z, \ldots) = \overline{X} + f(1, 0, Y, Z, \ldots)$$

# Example 13

- Simplify the following:

$$A.[\overline{A}.B + A.\overline{C} + (\overline{A} + D).(A + \overline{E})] =$$

✓Solution:

$$= A.[0.B + 1.\overline{C} + (0 + D).(1 + \overline{E})]$$

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

# Example 14

- Simplify the flollowing:

$$\overline{A} + [\overline{A}.B + A.\overline{C} + (\overline{A} + B).(A + \overline{E})] =$$

✓Solution:

$$= \overline{A} + [0.B + 1.\overline{C} + (0 + B).(1 + \overline{E})]$$

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

# Theorems of Boolean Algebra

- Theorem 16:

(a) $f(X, \overline{X}, Y, \ldots, Z) = X.f(1, 0, Y, \ldots, Z) + \overline{X}.f(0, 1, Y, \ldots, Z)$

(b) $f(X, \overline{X}, Y, \ldots, Z) = [X + f(0, 1, Y, \ldots, Z)][\overline{X} + f(1, 0, Y, \ldots, Z)]$

- Proof:

$$\begin{aligned} f(X, \overline{X}, Y, \ldots, Z) &= X.f(X, \overline{X}, Y, \ldots, Z) + \overline{X}.f(X, \overline{X}, Y, \ldots, Z) \\ &= X.f(1, 0, Y, \ldots, Z) + \overline{X}.f(0, 1, Y, \ldots, Z) \end{aligned}$$

$$\begin{aligned} f(X, \overline{X}, Y, \ldots, Z) &= [X + f(X, \overline{X}, Y, \ldots, Z)][\overline{X} + .f(X, \overline{X}, Y, \ldots, Z)] \\ &= [X + f(0, 1, Y, \ldots, Z)][\overline{X} + f(1, 0, Y, \ldots, Z)] \end{aligned}$$

# Theorems of Boolean Algebra

- Theorem 17 (Involution Law)

$$\overline{\overline{X}} = X$$

# Example 15

- Prove the following:

$$L.(M+\overline{N})+\overline{L}.\overline{P}.Q=(L+\overline{P}.Q).(\overline{L}+M+\overline{N})$$

✓Solution:

- Assume: $L = X, (M+\overline{N}) = Y$ and $\overline{P}.Q = Z$

$$L.(M+\overline{N})+\overline{L}.\overline{P}.Q = X.Y+\overline{X}.Z$$

$$= (X+Z).(\overline{X}+Y)$$

$$= (L+\overline{P}.Q)(\overline{L}+M+\overline{N})$$

$$= \text{RHS}$$

# Example 16

- Prove the following:

$$[A.\overline{B}+\overline{C}+\overline{D}].[D+(E+\overline{F}).G]=$$

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

✓Solution:

- Assume: $\overline{D}=X,\ A.\overline{B}+\overline{C}=Y$ and $(E+\overline{F}).G=Z$

$$[A.\overline{B}+\overline{C}+\overline{D}].[D+(E+\overline{F}).G]=$$

$$=(X+Y).(\overline{X}+Z)=X.Z+\overline{X}.Y$$

$$=\overline{D}.G.(E+\overline{F})+D.(A.\overline{B}+\overline{C})=\text{RHS}$$

# Example 17

- Simplify the following:

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

✓Solution:

$$= 1$$

# Example 18

- Simplify the following:

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

✓Solution:

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

# Example 19

- Simplify the following:

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

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

$$+\overline{A}.\overline{B}.C.\overline{D}.\overline{E}+A.\overline{B}.\overline{C}.D.\overline{E}=$$

✓Solution:

$$=B.E+\overline{B}.D.\overline{E}+\overline{B}.\overline{D}.\overline{E}$$

$$=B.E+\overline{B}.\overline{E}$$

# Summary

- Basic laws:

| | |
|---|---|
| **Commutative** | $X.Y = Y.X, \quad X + Y = Y + X$ |
| **Associative** | $X.(Y.Z) = (X.Y).Z, \quad X + (Y + Z) = (X + Y) + Z$ |
| **Distributive** | $X.(Y + Z) = X.Y + X.Z, \quad (X + Y).(X + Z) = X + Y.Z$ |

# Summary

- Basic theorems:

| # | Theorem | Minterm | Maxterm |
|---|---|---|---|
| 1 | 0, 1 | $\mathbf{X.1 = X}$ | $\mathbf{X + 0 = X}$ |
| 2 | 0, 1 | $\mathbf{X.0 = 0}$ | $\mathbf{X + 1 = 1}$ |
| 3 | Complement | $X.\overline{X} = 0$ | $X + \overline{X} = 1$ |
| 4 | Idempotent | $\mathbf{X.X = X}$ | $\mathbf{X + X = X}$ |
| 5 | Absorption | $\mathbf{X + X.Y = X}$ | $\mathbf{X.(X + Y) = X}$ |
| 6 | Involution | $\overline{\overline{X}} = X$ | |
| 7 | DeMorgan's | $\overline{(X.Y.Z...)} = \overline{X} + \overline{Y} + \overline{Z} + ...$ | $\overline{(X + Y + Z + ...)} = \overline{X}.\overline{Y}.\overline{Z}...$ |

# Contents

1. Introduction to Boolean Algebra
2. Variables, Literals & Terms in Boolean Expressions
3. Laws and Rules of Boolean Algebra
4. Logic Simplification
5. Basic Logic Gates

# Logic Simplification

- The primary objective of all simplification procedures is to obtain an expression that has the minimum number of terms.
- If there is more than one possible solution with the same number of terms, the one having the minimum number of literals is the choice.
- Some simplification techniques:
  - Algebraic method
  - Karnaugh map method, or
  - Quine-McCluskey method

# Algebraic Method

- This method uses Boolean theorems to simplify a logic function.
- Example: Simplify the following expression

$$f = AB + \overline{A}C + BC$$

- Apply $A + \overline{A} = 1$ and $X + XY = X$

$$f = AB + \overline{A}C + BC\left(A + \overline{A}\right)$$

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

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

# Example 20

- Simplify this expression:

$$f = AB + BCD + \overline{A}C + \overline{B}C$$

- Apply $A + \overline{A} = 1$ and $X + XY = X$

$$f = AB + BCD(A + \overline{A}) + \overline{A}C + \overline{B}C$$

$$= (AB + ABCD) + (\overline{A}BCD + \overline{A}C) + \overline{B}C$$

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

$$= AB(1 + C) + \overline{AB}.C$$

$$= AB + C$$

# Truth Table (Review)

- Getting the Boolean expression from the Truth Table: **f(A,B,C)**

$= ABC = m_7$

$= (A+B+C)(A+B+C')(A+B'+C)(A+B'+C')(A'+B+C)\ (A'+B+C')(A'+B'+C)$

| m | A | B | C | f |
|---|---|---|---|---|
| $m_0$ | 0 | 0 | 0 | 0 |
| $m_1$ | 0 | 0 | 1 | 0 |
| $m_2$ | 0 | 1 | 0 | 0 |
| $m_3$ | 0 | 1 | 1 | 0 |
| $m_4$ | 1 | 0 | 0 | 0 |
| $m_5$ | 1 | 0 | 1 | 0 |
| $m_6$ | 1 | 1 | 0 | 0 |
| $m_7$ | 1 | 1 | 1 | 1 |

# Std. Forms of Boolean Expressions

- A given Boolean function can be in either of 2 forms: minterm (SoP) or maxterm (PoS).
- A standard form (or canonical form) contain as many literals as the Boolean function has.
- In general, a Boolean function of n variables, it can be represented into the SoP form ($m_i$: minterm)

$$f(X_{n-1},\ldots,X_0)=\sum_{i=0}^{2^n-1} a_i m_i$$

or the PoS form ($M_i$: maxterm):

$$f(X_{n-1},\ldots,X_0)=\prod_{i=0}^{2^n-1} (a_i + M_i) \qquad a_i = 0 \text{ or } 1$$

# Minterm

- In general: $f(X_{n-1},\ldots,X_0)=\sum_{i=0}^{2^n-1} a_i m_i$ $\quad a_i = 0 \text{ or } 1$

| $A$ | $B$ | $C$ | $Y$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

$$Y = \overline{A}\ .\overline{B}.\overline{C} + \overline{A}.B.C + A.B.\overline{C} + A.\overline{B}.C + A.B.C$$

$$\Rightarrow f(A,B,C) = \sum 0,3,5,6,7$$

# Maxterm

- In general: $$f(X_{n-1},\ldots,X_0)=\prod_{i=0}^{2^n-1}(a_i+M_i)$$ $a_i = 0$ or $1$

| $A$ | $B$ | $C$ | $Y$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

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

$$\Rightarrow f(A,B,C)=\prod 1,2,4$$

# Maxterm → Minterm

- ✓Multipling out the given expression
- ✓Removing redundancy

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

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

$$+ A.B.C$$

# Minterm → Maxterm

- ✓Taking the dual of the given expression
- ✓Multiplying out different terms to get the SoP form
- ✓Removing redundancy
- ✓Taking a dual to get the equivalent PoS form

$$
\begin{aligned}
Y &= A.B + \overline{A}.\overline{B} \\
\text{Dual} &= (A+B).(\overline{A}+\overline{B}) \\
&= A.\overline{A} + A.\overline{B} + B.\overline{A} + B.\overline{B} \\
&= 0 + A.\overline{B} + B.\overline{A} + 0 \\
&= A.\overline{B} + \overline{A}.B
\end{aligned}
$$

$$\text{The dual of } (A.\overline{B} + \overline{A}.B) = (A+\overline{B}).(\overline{A}+B)$$

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

# Karnaugh Map Method

- Construction of a K-Map
  - An n-variable K-map has $2^n$ squares.
  - Each possible combination of inputs is allotted a square.
- In the case of a minterm K-map:
  - '1' is placed in the squares for which the output is '1'.
  - '0' is placed in the square for which the output is '0'.
  - 'X' is placed in the squares corresponding to 'don't' care' conditions.
- In the case of a maxterm K-map:
  - '1' is placed in square for which the output is '0'.
  - '0' is placed in the square for which the output is '1'.
  - 'X' is placed in the square corresponding to 'don't' care' conditions.
- **The designation of adjacent rows and adjacent columns should be the same except for one of the literals being complemented.**
- Also, the extreme rows and extreme columns are considered adjacent.

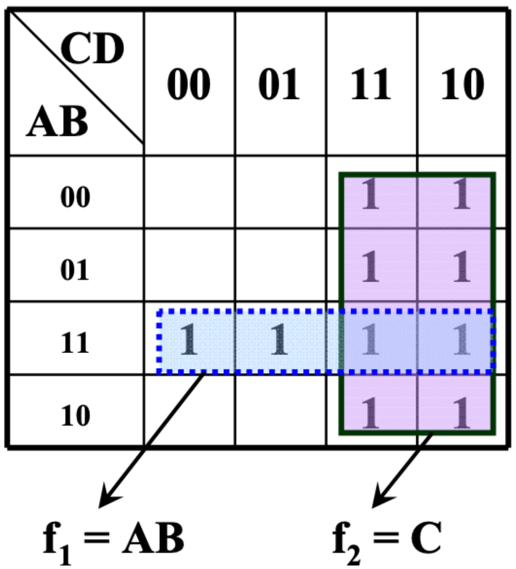
| A \ B | 0 | 1 |
|---|---|---|
| 0 | | |
| 1 | | |

| A \ BC | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |

| AB \ CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |

# Karnaugh Map Method

- Efficient for $\leq 5$ variables.
- **Guidelines:**
  - Each square containing a ‘1’ must be considered at least once, although it can be considered as often as desired.
  - The objective should be to account for all the marked squares in the minimum number of groups.
  - The number of squares in a group must always be a power of 2, i.e. groups can have 1, 2, 4, 8, 16, … squares.




  - Each group should be as large as possible; a group of two squares should not be made if the involved squares can be included in a group of four squares and so on.
  - ‘Don’t care’ entries (marked ‘X’) can be used in accounting for all of 1-squares to make optimum groups. Such entries that can be used to advantage should be used.

# Quine McCluskey Method

- Efficient for > 5 variables.
- Based on the complementation theorem:

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

- **Step-by-step procedure:**
    1. Divide all the minterms (and don't cares) of a function (F) into groups.
    2. Merge minterms from adjacent groups to form a new implicant table.
    3. Repeat Step 2 until no more merging is possible.
    4. Put all prime implicants (PIs) in a cover table (**don't cares excluded**).
    5. Identify essential minterms, and hence essential prime implicants (EPIs).
    6. Add prime implicants to the minimum expression of F until all minterms of F are covered.

# Quine McCluskey Method

- Example: $f(A, B, C, D) = \sum(10,11,12,13,14,15)$

| Bảng a | | Bảng b | |
|---|---|---|---|
| Hạng tích sắp xếp | Nhị phân (ABCD) | Rút gọn lần 1 *(ABCD)* | Rút gọn lần thứ 2 *(ABCD)* |
| 10 | 1 0 1 0 | 1 0 1 - # (10,11) | 1 1 - - (12,13,14,15) |
| 12 | 1 1 0 0 | 1 - 1 0 # (10,14) | 1 - 1 - (10,11,14,15) |
| 11 | 1 0 1 1 | 1 1 0 - # (12,13) | |
| 13 | 1 1 0 1 | 1 1 - 0 # (12,14) | |
| 14 | 1 1 1 0 | 1 - 1 1 # (11,15) | |
| 15 | 1 1 1 1 | 1 1 - 1 # (13,15) | |
| | | 1 1 1 - # (14,15) | |

# Quine McCluskey Method

- Example: $\mathrm{f}(\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}) = \sum(10,11,12,13,14,15)$

| *A BCD* | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|
| 1 1 - - | | | x | x | x | x |
| 1 - 1 - | x | x | | | x | x |

# Contents

# Basic Logic Gates

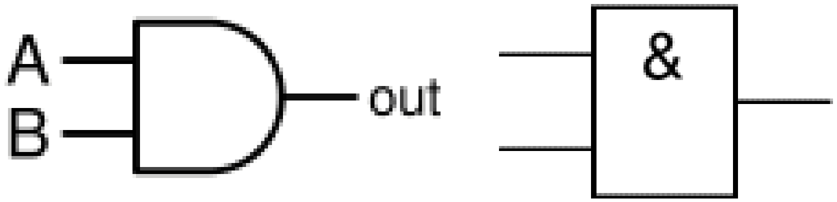
- Concepts
- Implementation of AND, OR gates using diodes
- Implementation of a NOT gate using transistors
- Integrated circuits (ICs)

# Concepts

- 3 basic logic functions:
  - AND
  - OR
  - NOT
- 3 basic logic gates to implement logic functions:
  - **AND** gate
  - **OR** gate
  - **NOT** inverter
- Other logic gates: NAND, NOR, XOR, XNOR

# AND Gate

- Functionality:
  - Performing an ANDing operation on two or more than two logic variables.
- 2-input AND gate:
  - Symbol:

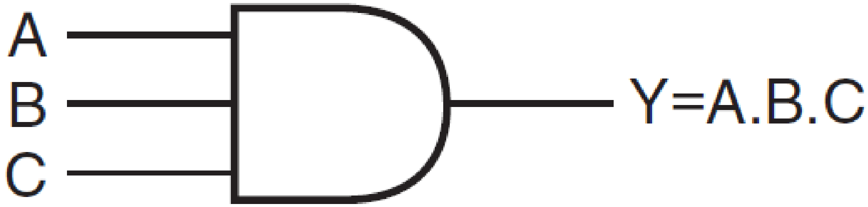



  - Truth Table:
  - Expression: out = A . B

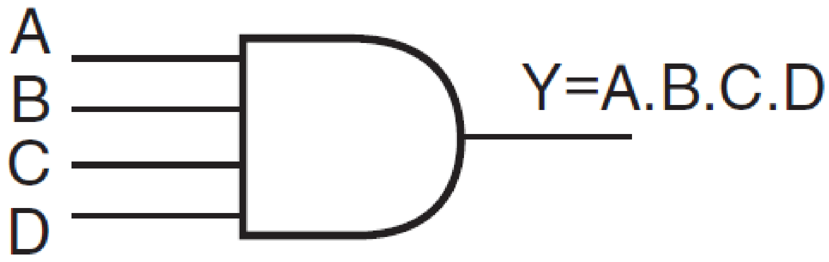
| A | B | out |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# AND Gate

- 3-input AND gate:




- 4-input AND gate:




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

# OR Gate

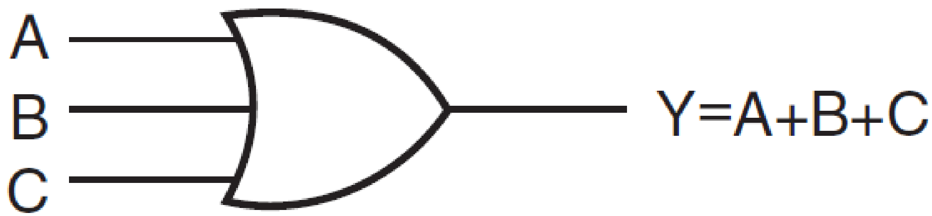
- Functionality:
  - Performing an ANDing operation on two or more than two logic variables.
- 2-input OR gate:
  - Symbol:


  - Truth Table:
  - Expression: out = A + B

| A | B | out |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

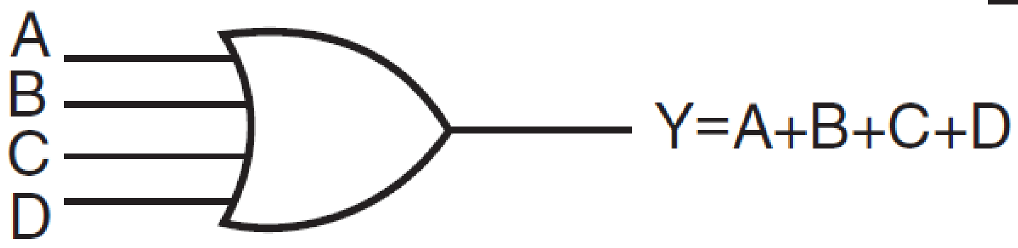
# OR Gate

- 3-input OR gate:



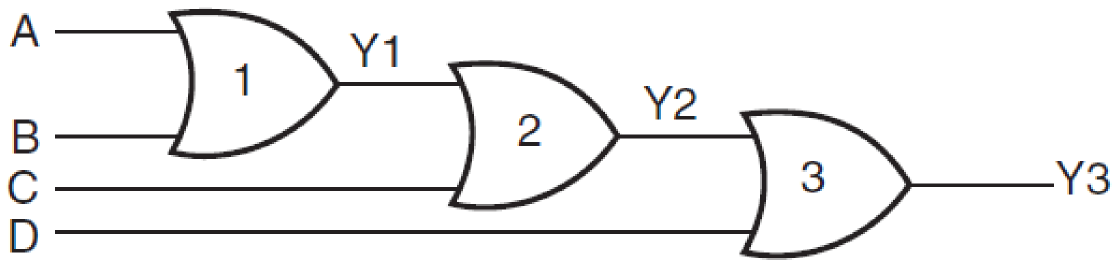

| A | B | C | Y |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

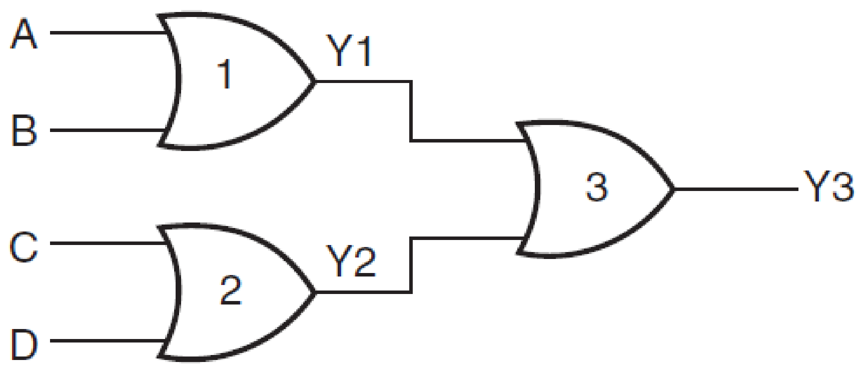
- 4-input OR gate:



# Example 21

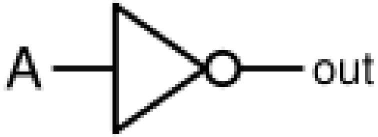
- Show the logic arrangement for implementing a four-input OR gate using two-input OR gates only.

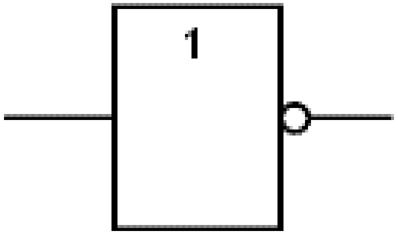


# NOT Inverter

- Functionality:
  - The output is always the complement of the input.
- A NOT inverter has only one input.
  - Symbol:



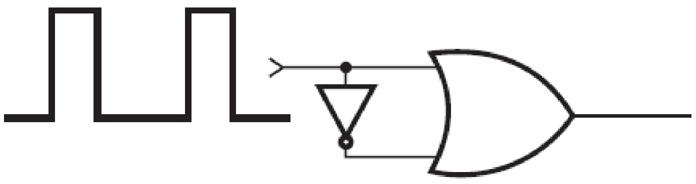

  - Truth Table:
  - Expression:

$$out = \overline{A}$$

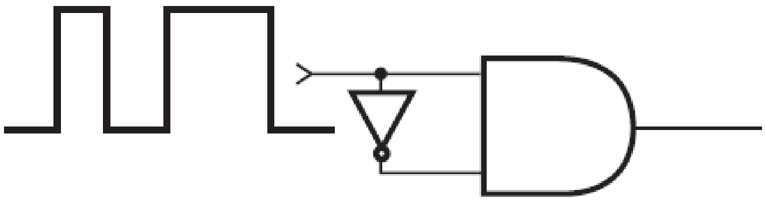
| A | out |
|---|---|
| 0 | 1 |
| 1 | 0 |

# Example 22

- Draw the output waveform:



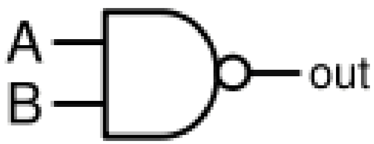
(a)

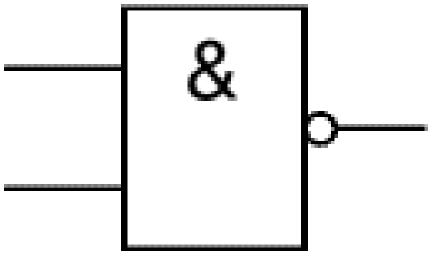


(b)

# NAND Gate

- Functionality:
  - Performing an ANDing operation followed by a NOT operation on two or more than two logic variables.
- 2-input NAND gate:
  - Symbol:



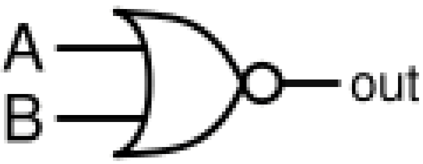

  - Truth Table:
  - Expression:

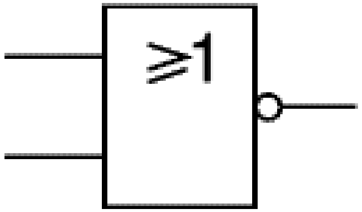
$$out = \overline{A \,.\, B}$$

| A | B | out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# NOR Gate

- Functionality:
  - Performing an ORing operation followed by a NOT operation on two or more than two logic variables.
- 2-input NOR gate:
  - Symbol:



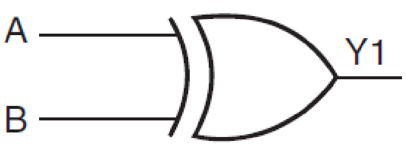

  - Truth Table:
  - Expression:

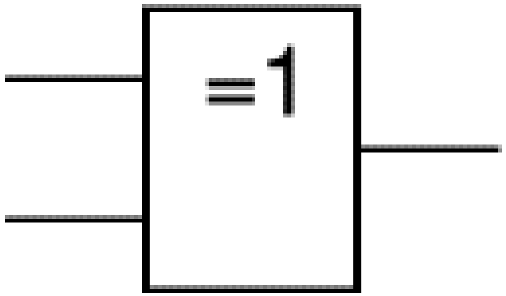
$$out = \overline{A + B}$$

| A | B | out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

# Exclusive-OR Gate (XOR Gate)

- Functionality:
  - The output of a multiple-input EX-OR logic function is a logic '1' when the number of 1s in the input sequence is odd and a logic '0' when the number of 1s in the input sequence is even, including zero.
- 2-input XOR gate:
  - Symbol:



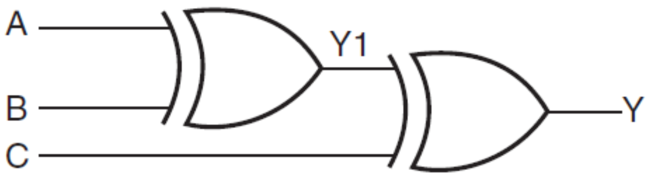

  - Truth Table:
  - Expression:

| A | B | out |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

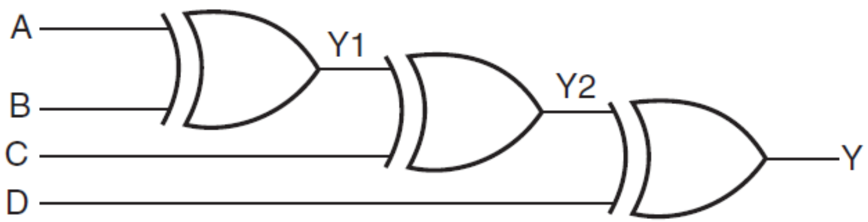
$$out = A \oplus B = \overline{A}.B + A.\overline{B}$$

# Example 23

- How do you implement three-input and four-input EX-OR logic functions with the help of two-input EX-OR gates?




(a)



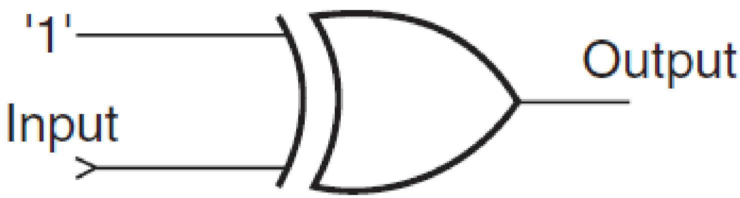

(b)

3-input XOR gate

| A | B | C | Output |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

# Example 24

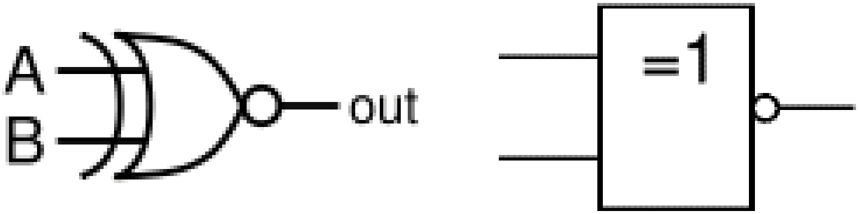
- How can you implement a NOT circuit using a two-input EX-OR gate?




| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# XNOR Gate

- Functionality:
  - Complementing the output of an EX-OR gate.
- 2-input XNOR gate:
  - Symbol:



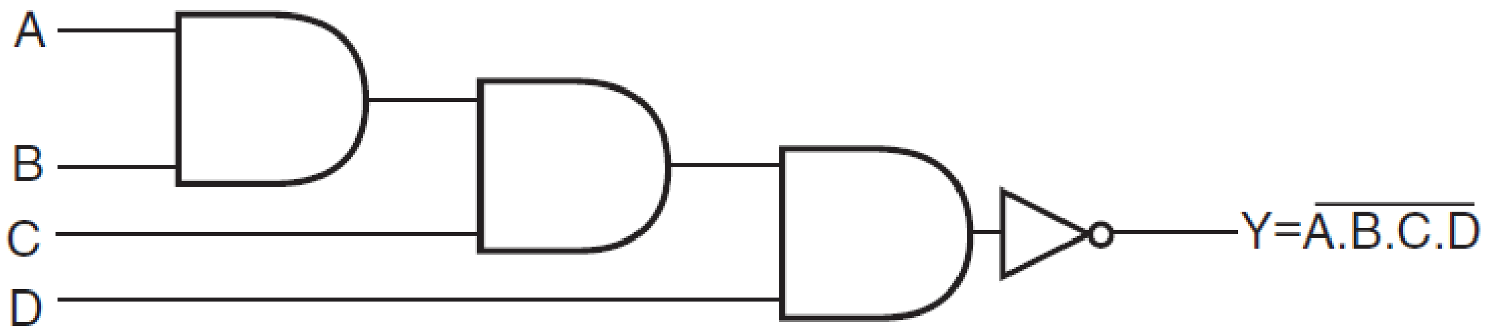

  - Truth Table:
  - Expression:

$$out = \overline{A \oplus B} = A.B + \overline{A}.\overline{B}$$

| A | B | out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

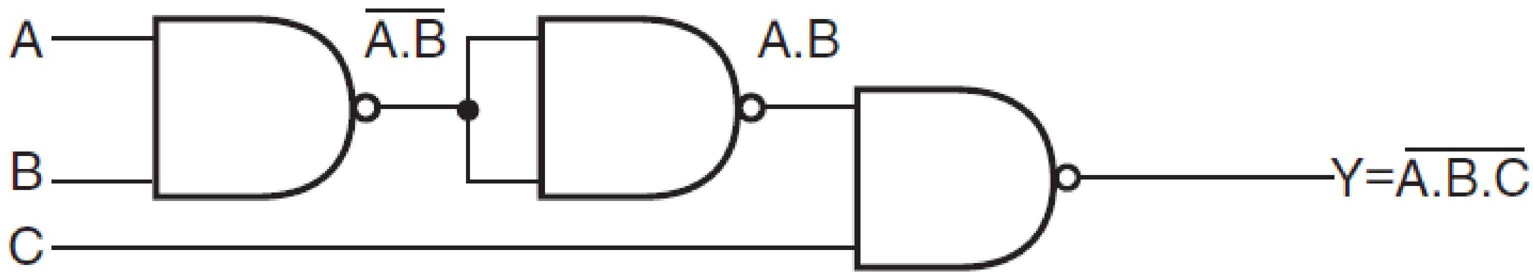
# Example 25

- How do you implement a three-input EX-NOR function using only two-input EX-NOR gates?



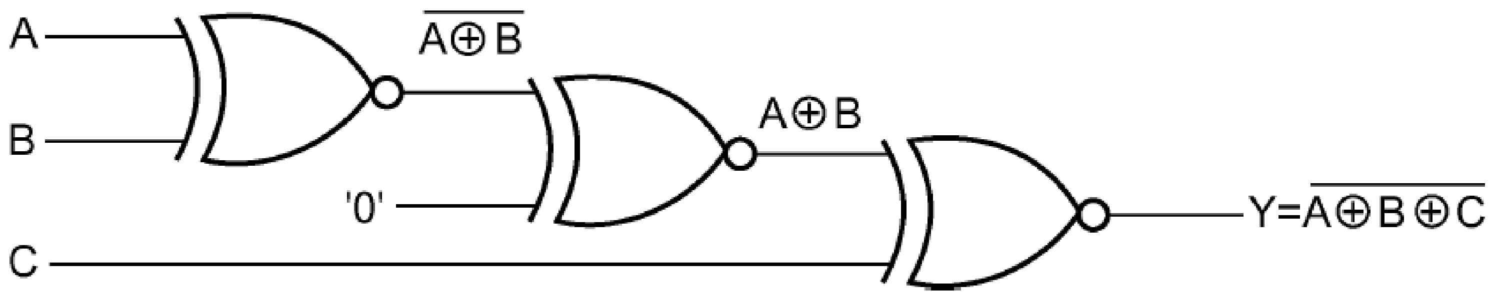
# Example 26

- Show the logic arrangements for implementing a three-input NAND gate using two-input NAND gates.



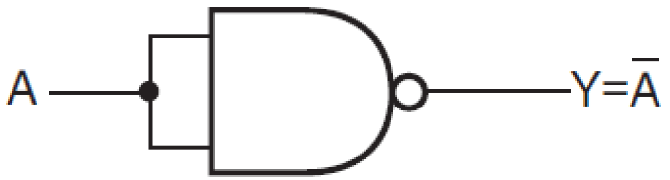
# Example 27

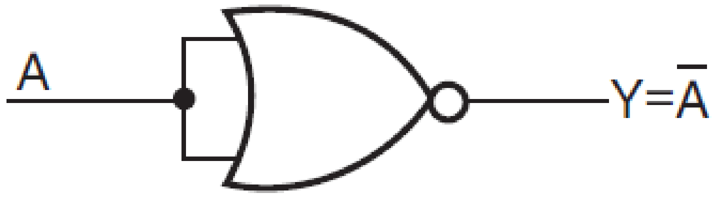
- Show the logic arrangements for implementing a three-input XNOR gate using two-input XNOR gates.

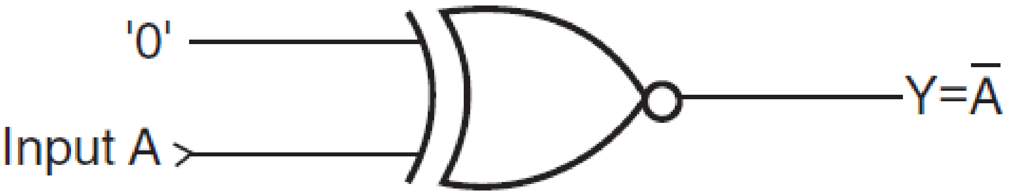


# Example 28

- Show the logic arrangements for implementing a NOT gate using:
  - 2-input NAND gates
  - 2-input NOR gates
  - 2-input XNOR gates



# Exercise 1

a) Show the logic arrangements for implementing a 8-input NAND gate using 2-input AND gates and 2-input NAND gates.

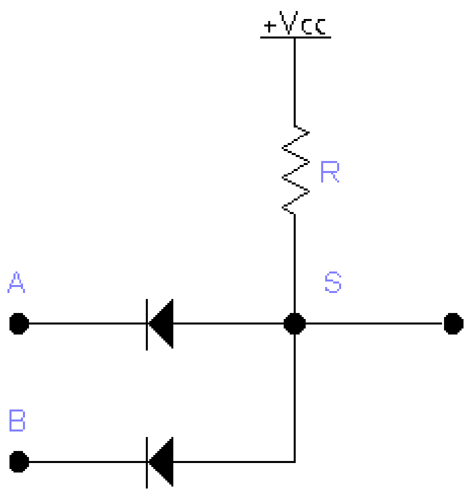
b) Show the logic arrangements for implementing a 8-input XNOR gate using a minimum number of 2-input logic gates.

# Basic Logic Gates

- Concepts
- Implementation of AND, OR gates using diodes
- Implementation of a NOT gate using transistors
- Integrated circuits (ICs)

# Implementing a 2-Input AND Gate using Diodes

- Standard TTL
- Input A = 5V, Input B = 5V → Output S ≈ 5V
- Input A = 0V, Input B = 5V → Output S ≈ 0V
- Input A = 5V, Input B = 0V → Output S ≈ 0V
- Input A = 0V, Input B = 0V → Output S ≈ 0V




**S = A.B**

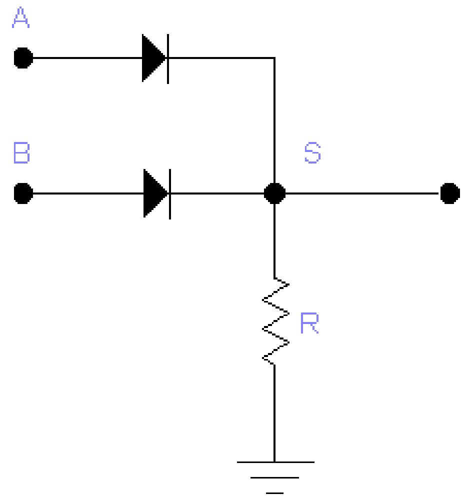
| $U_A$ | $U_B$ | $U_S$ | |
|---|---|---|---|
| 0 | 0 | 0 | $D_A$, $D_B$ thông |
| 0 | 5 | 0 | $D_A$ thông, $D_B$ tắt |
| 5 | 0 | 0 | $D_A$ tắt, $D_B$ thông |
| 5 | 5 | 5 | $D_A$, $D_B$ tắt |

→

| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# Implementing a 2-Input OR Gate using Diodes

- Standard TTL
- Input A = 5V, Input B = 5V → Output S ≈ 4.3V
- Input A = 0V, Input B = 5V → Output S ≈ 4.3V
- Input A = 5V, Input B = 0V → Output S ≈ 4.3V
- Input A = 0V, Input B = 0V → Output S ≈ 0V




**S = A+B**

| $U_A$ | $U_B$ | $U_S$ | |
|---|---|---|---|
| 0 | 0 | 0 | $D_A$, $D_B$ tắt |
| 0 | 5 | 5 | $D_A$ tắt, $D_B$ thông |
| 5 | 0 | 5 | $D_A$ thông, $D_B$ tắt |
| 5 | 5 | 5 | $D_A$, $D_B$ thông |

→

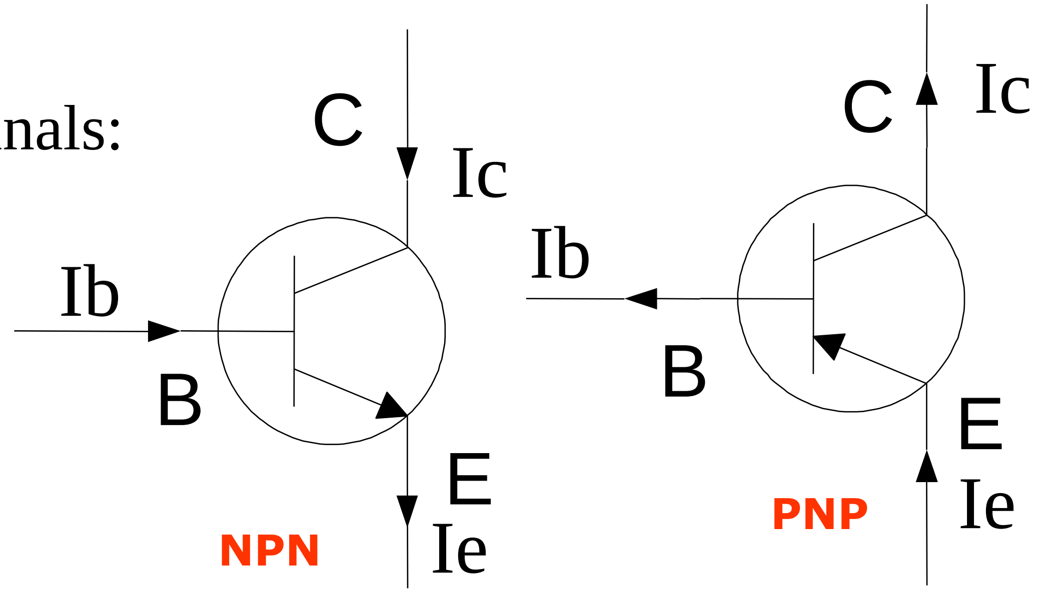
| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

# Basic Logic Gates

- Concepts
- Implementation of AND, OR gates using diodes
- Implementation of a NOT gate using transistors
- Integrated circuits (ICs)

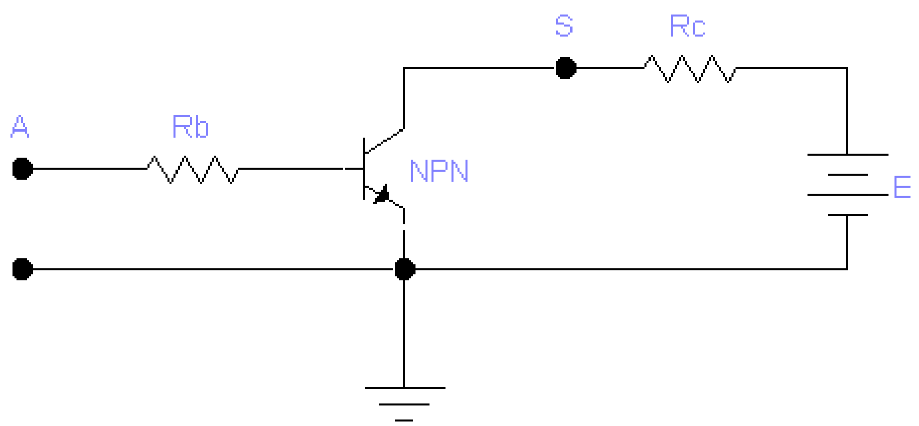
# Implementing a NOT Gate using Transistors

- BJT: NPN & PNP
  - A BJT has 3 terminals:
    - B: Base
    - C: Collector
    - E: Emitter




  - Functionality: controlling $I_B$ to get amplified $I_C$
  - Operation:
    - $I_B = 0$, the BJT is in cut-off mode, $I_C = 0$
    - $I_B > 0$, the BJT is in forward-active mode, $I_C = \beta.I_B$, where $\beta$ is the common-emitter current gain.

# Implementing a NOT Gate using Transistors




- Standard TTL, small $R_b$
- Input A = 0V → the BJT off → Output S ≈ 5V
- Input A = 5V → the BJT on → Output S ≈ 0V

| $U_A$ | $U_S$ |
|---|---|
| 0 | 5 T tắt |
| 5 | 0 T thông |

→

| A | S |
|---|---|
| 0 | 1 |
| 1 | 0 |

$$S = \overline{A}$$

# Basic Logic Gates

- Concepts
- Implementation of AND, OR gates using diodes
- Implementation of a NOT gate using transistors
- Integrated circuits (ICs)

# Integrated Circuits (ICs)

- An IC is a set of electronic circuits on one small flat piece (or "chip") of semiconductor material, usually silicon. [*Wiki*]
  - **Advantage:** cost and performance.
  - **Disadvantage:** high cost of designing ICs and fabricating the required photomasks.
- 2 types of integrated circuits:
  - Analog IC: to handle continuous signals such as audio signals
  - Digital IC: to handle discrete signals such as binary values

# Classification of ICs

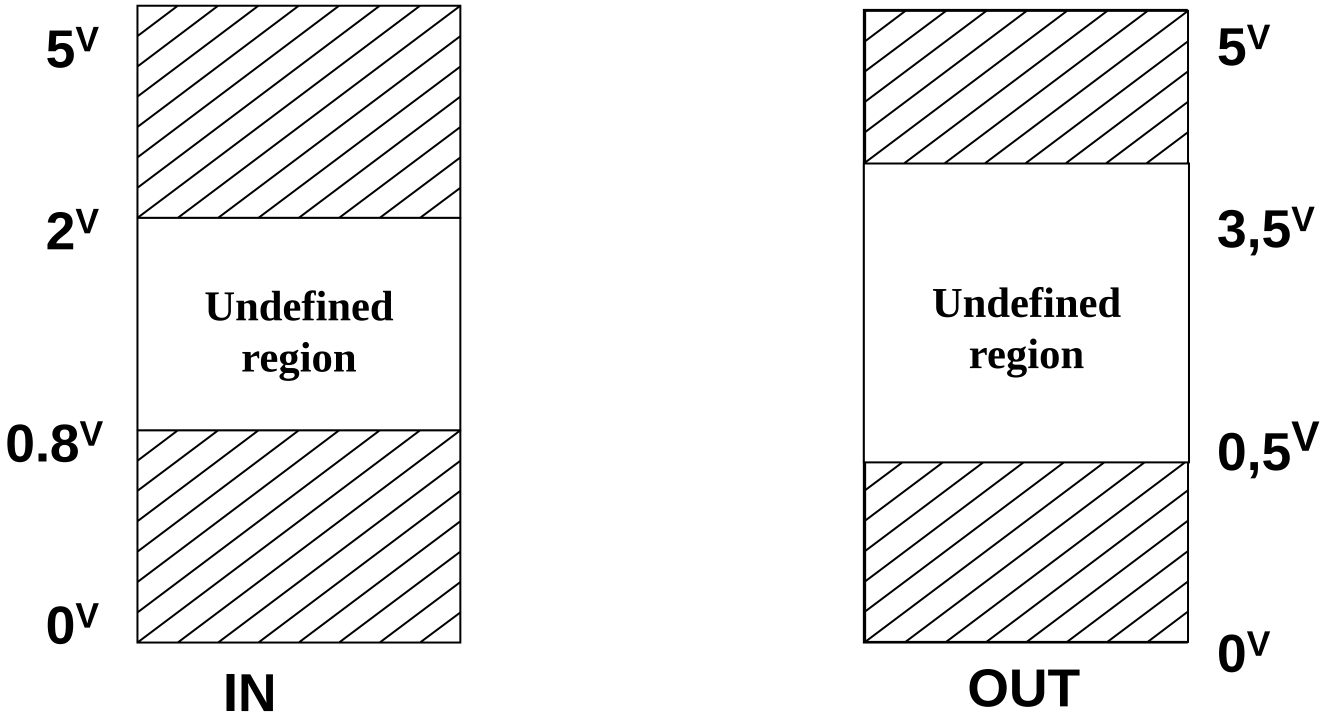
- Based on the chip size:
  - SSI - Small Scale Integration: <10 gates/chip, each gate consists of 2~10 transistors
  - MSI - Medium Scale Integration: 10~100 gates/chip
  - LSI - Large Scale Integration: 100~1000 gates/chip
  - VLSI - Very Large Scale Integration: $10^3$~$10^6$ gates/chip
  - ULSI - Ultra Large Scale Integration: >$10^6$ gates/chip

# Classification of ICs

- Based on the logic family:
  - Using BJT:
    - RTL – Resistor Transistor Logic
    - DTL – Diode Transistor Logic
    - TTL – Transistor Transistor Logic
    - ECL – Emitter Coupled Logic
  - Using FET (Field Effect Transistor)
    - MOS – Metal Oxide Semiconductor
    - CMOS – Complementary MOS

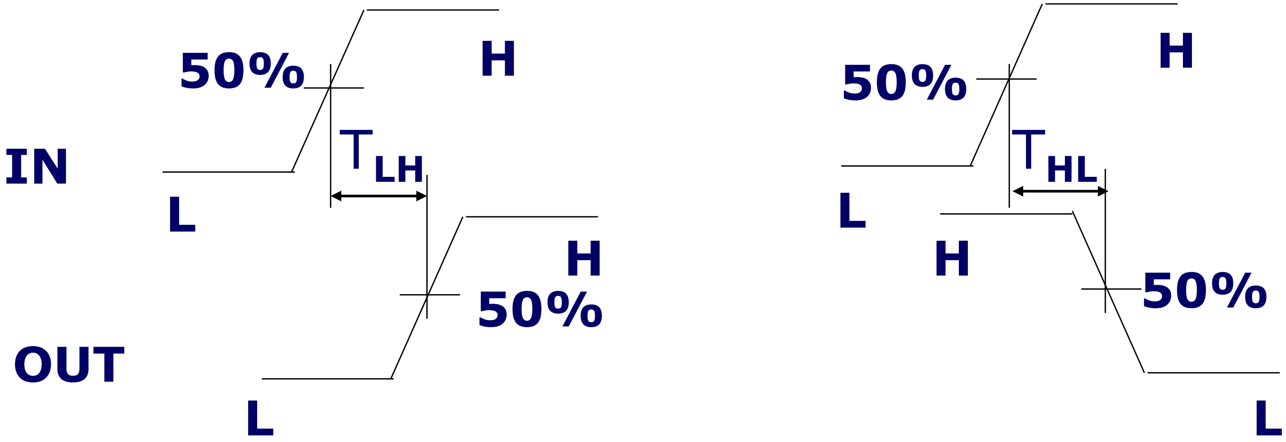
# Electrical Characteristics of ICs

- Logic levels are associated directly with voltage ranges.
  - **Example:** a TTL input signal is defined as “low” between 0V and 0.8V with respect to ground, and "high" when between 2V and 5V (Vcc).



# Electrical Characteristics of ICs

- The propagation delay is the time taken to respond when there is change on its inputs.

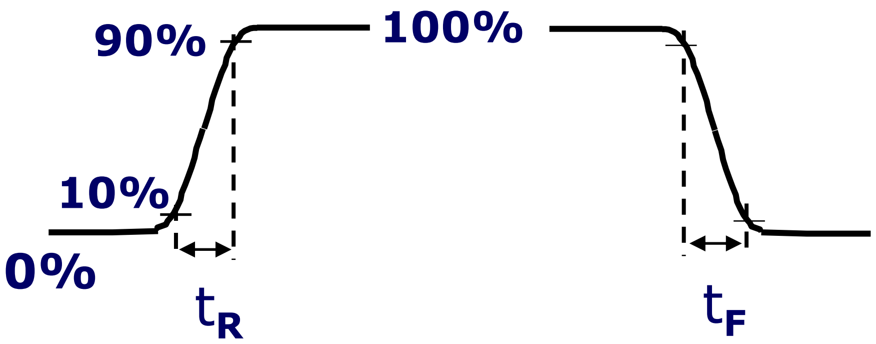



- Defined as the average of these two times:

$$T_{avg} = (T_{LH} + T_{HL})/2$$

# Electrical Characteristics of ICs

- The rise and fall times are the time taken by a signal to change its logic states from 0 to 1 or vice versa.
  - Ideally, the rise and fall times of a signal are 0.
  - In fact, the rise and fall times of a signal are typically measured from the 10% level to the 90% level (or vice versa) of its voltage.

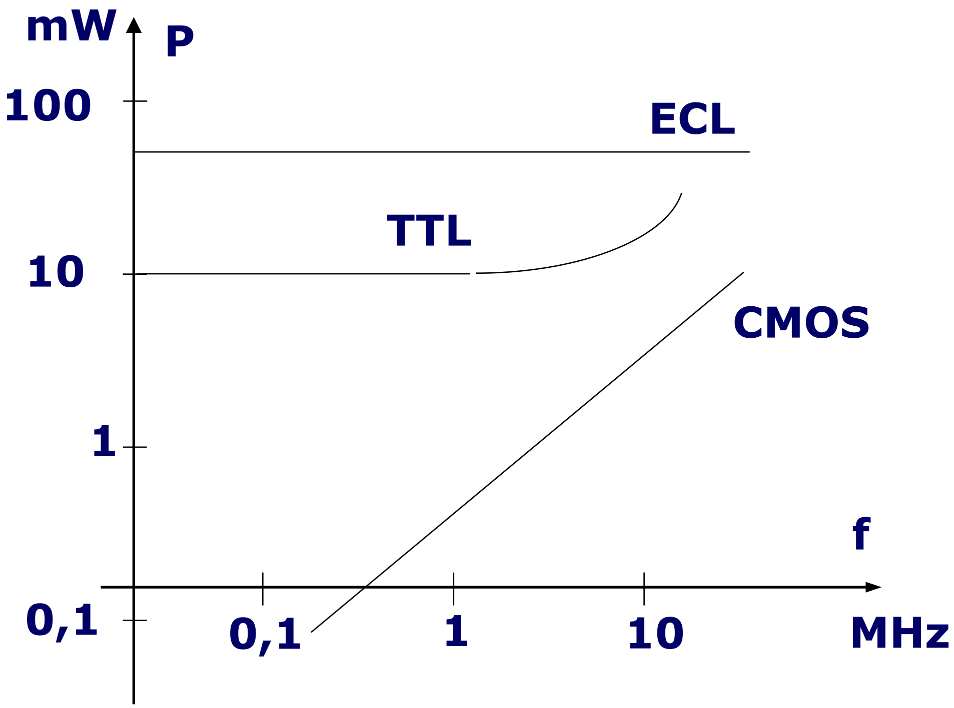



$t_R$: rise time
$t_F$: fall time

# Electrical Characteristics of ICs

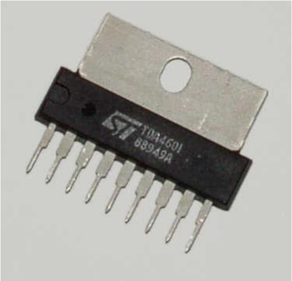
- Power dissipation: static power dissipation vs dynamic power dissipation.
- The dynamic power dissipation of an IC depends on the frequency of switching and logic families.

**Dynamic power dissipation of several logic families**

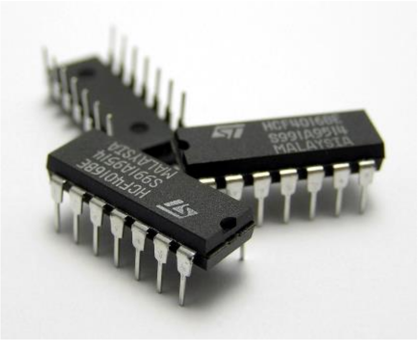


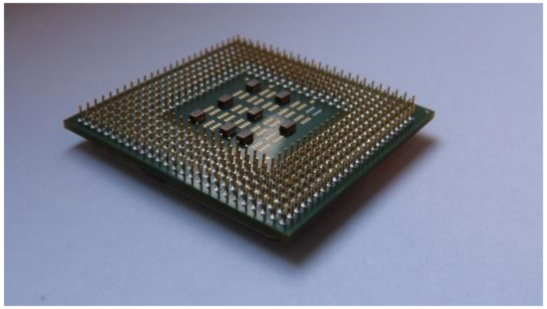
# IC Packaging Types

- SIP (Single Inline Package) or SIPP (Single In-line Pin Package)



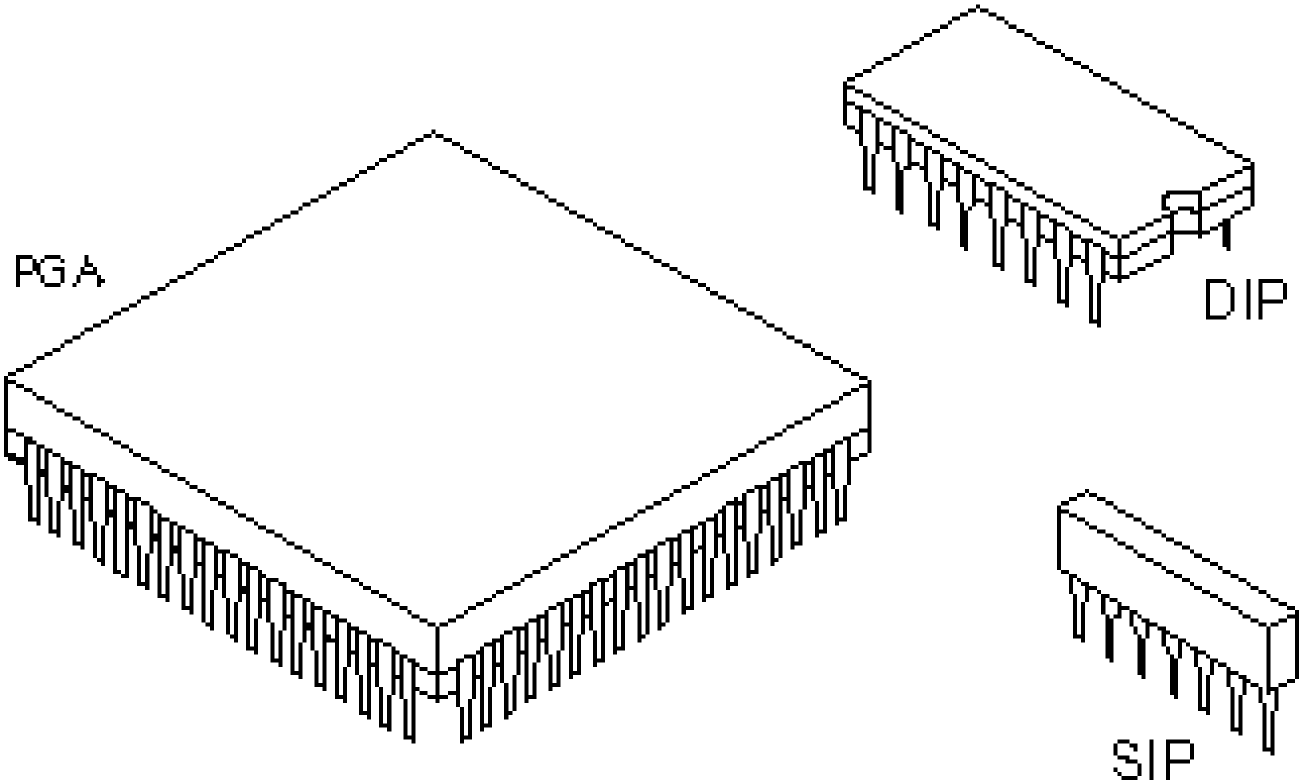
- DIP (Dual Inline Package)





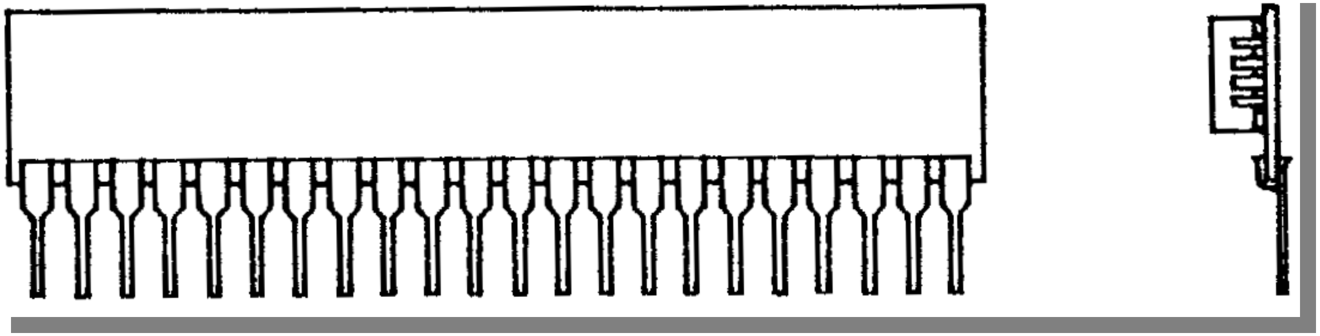
- PGA (Pin Grid Array)
- …

# IC Packaging Types



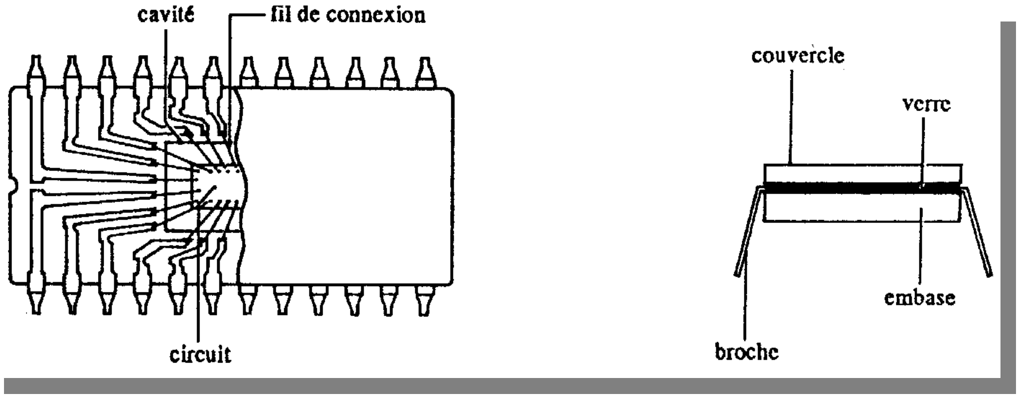
# IC Packaging Types

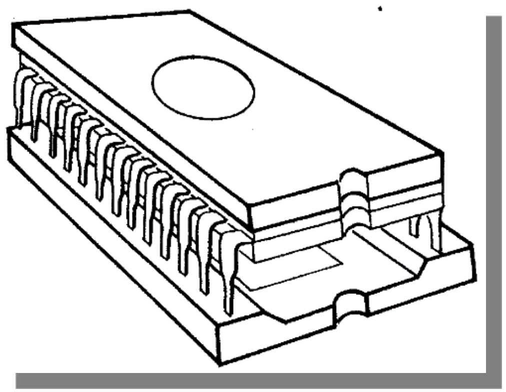
- SIL (Single In Line)

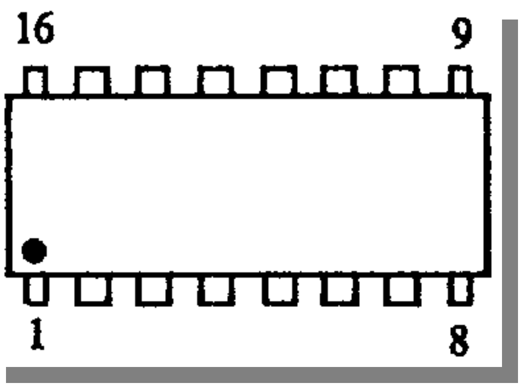


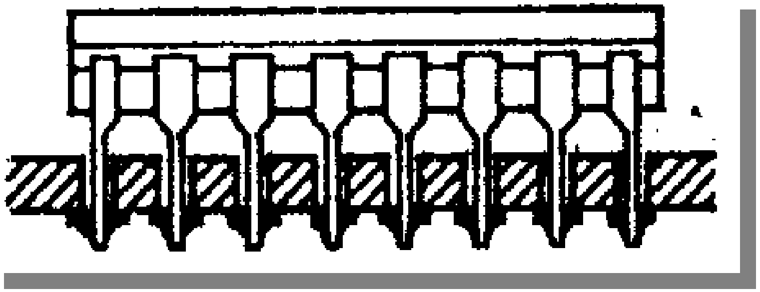
# IC Packaging Types

- DIL (Dual In Line): 8 to 64 pins



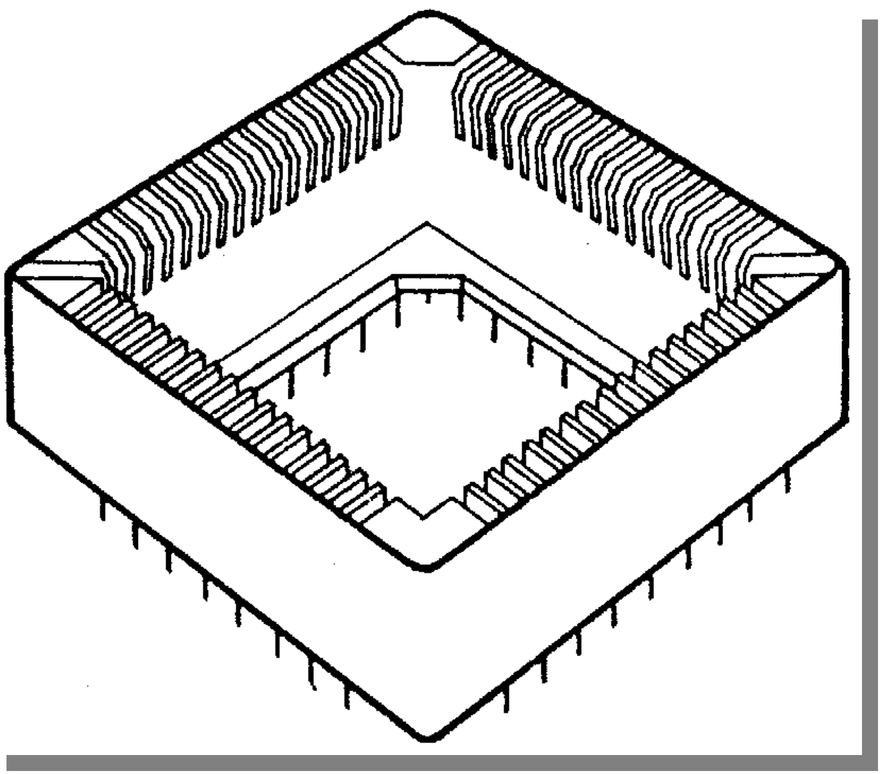



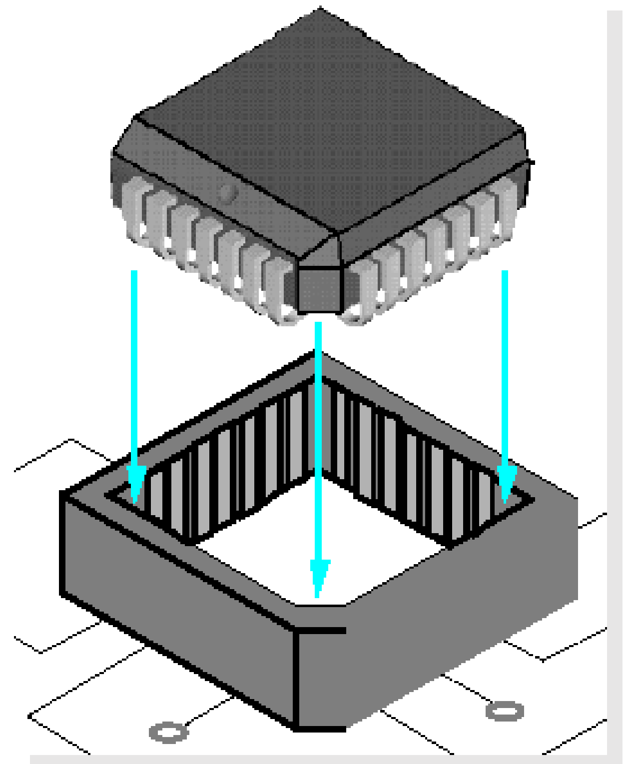


# IC Packaging Types

- PGA (Pin-Grid Array)

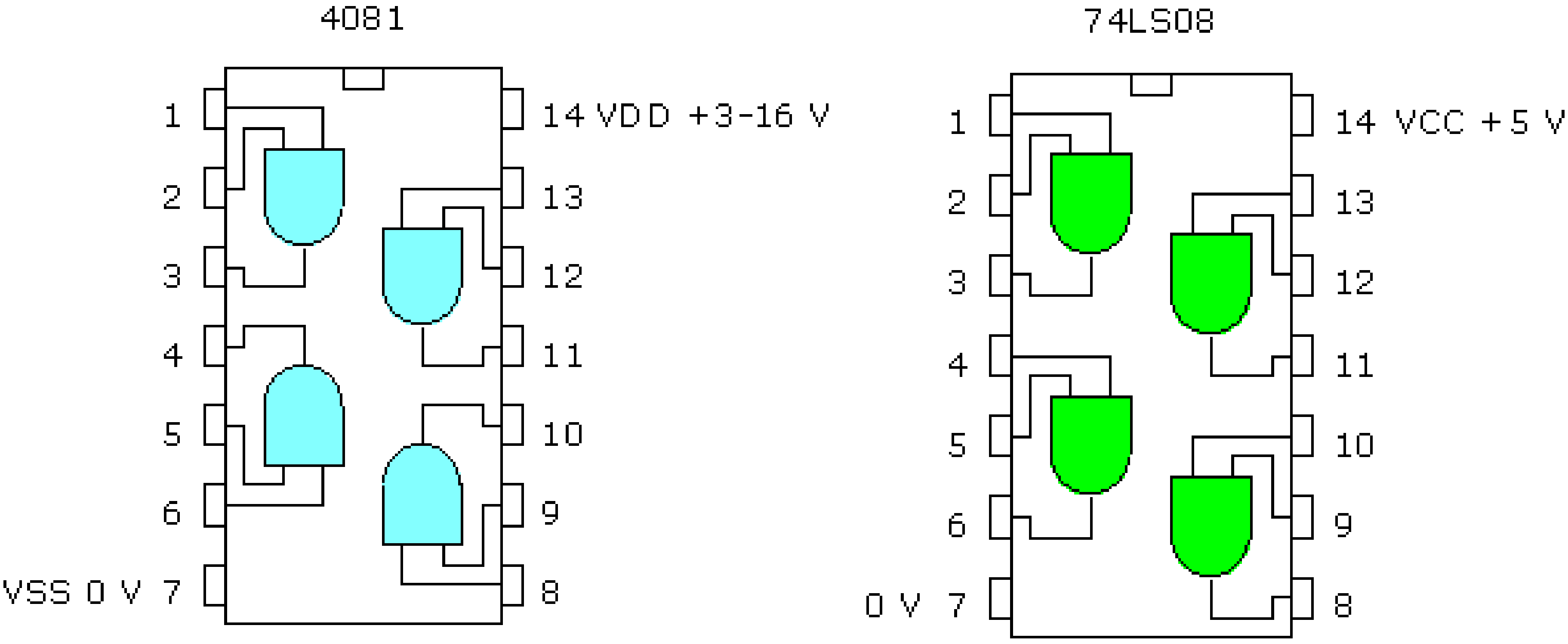




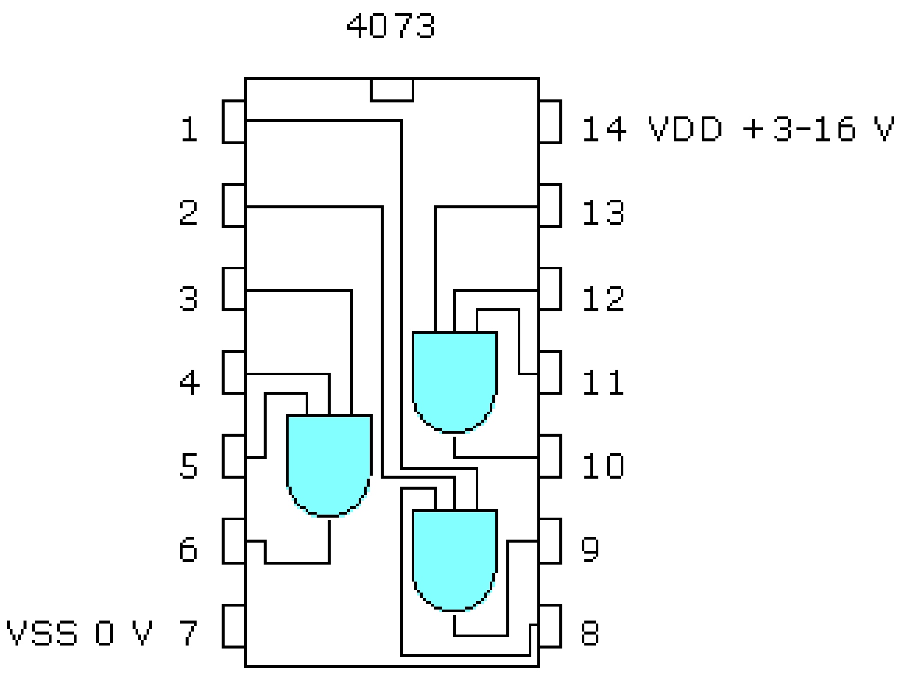
# Thermal Characteristics

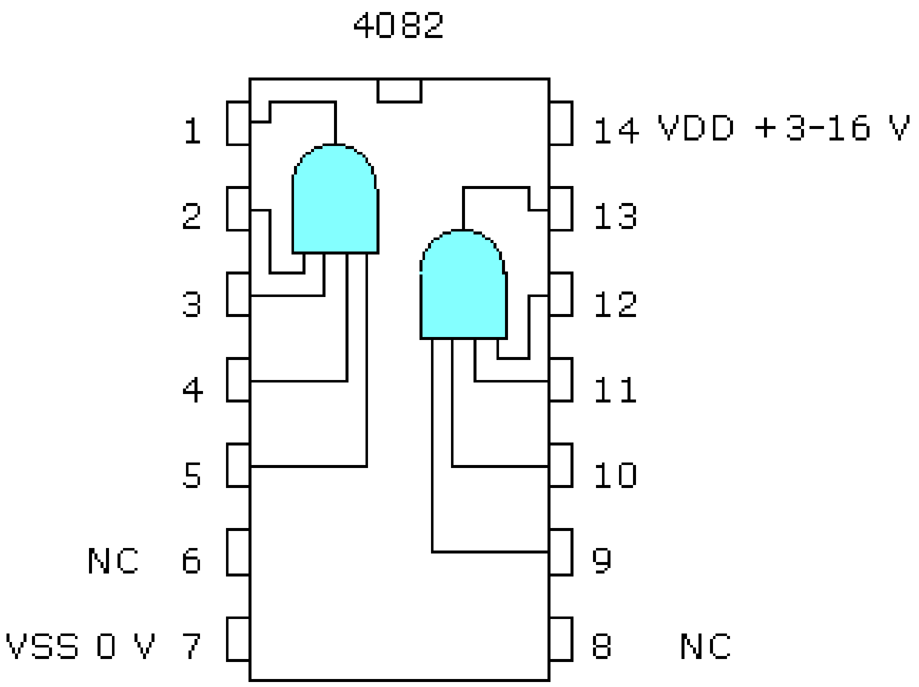
- The thermal resistance of an IC package is the measure of the package's ability to transfer heat generated by the IC (die) to the circuit board or the ambient.
- Operating Temperature: the allowable temperature range of the local ambient environment at which ICs operate.
  - Commercial: 0 °C to 70 °C
  - Industrial: −40 °C to 85 °C
  - Military: −55 °C to 125 °C

# IC of AND Gates

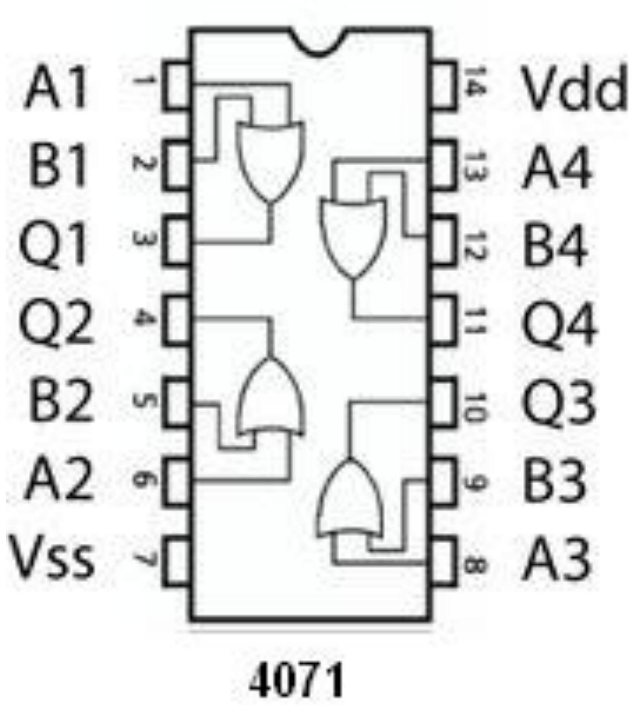


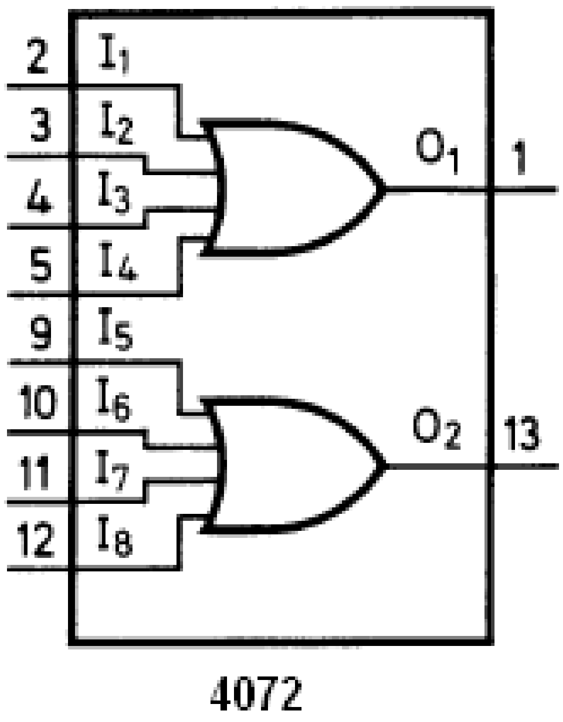
# IC of AND Gates



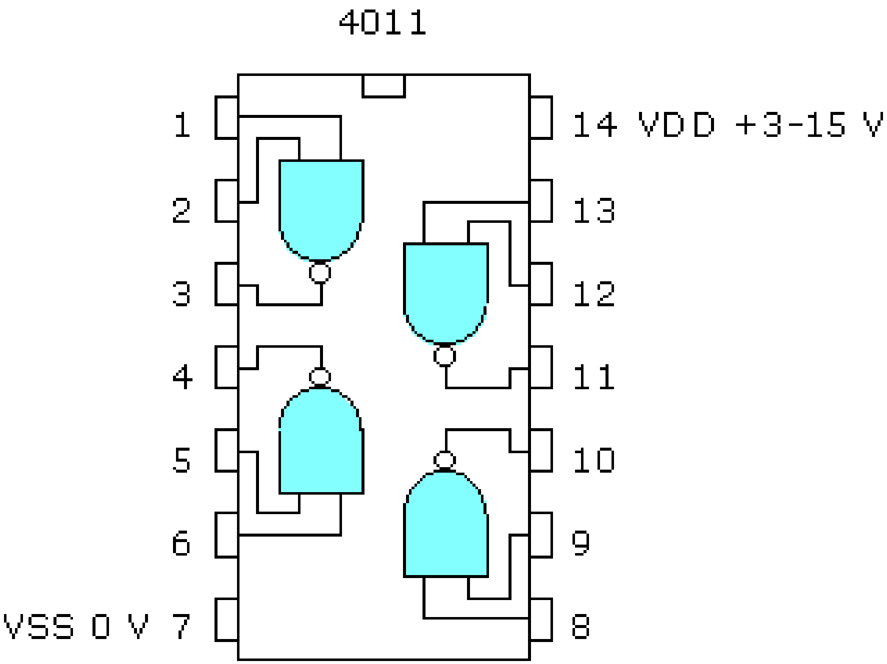



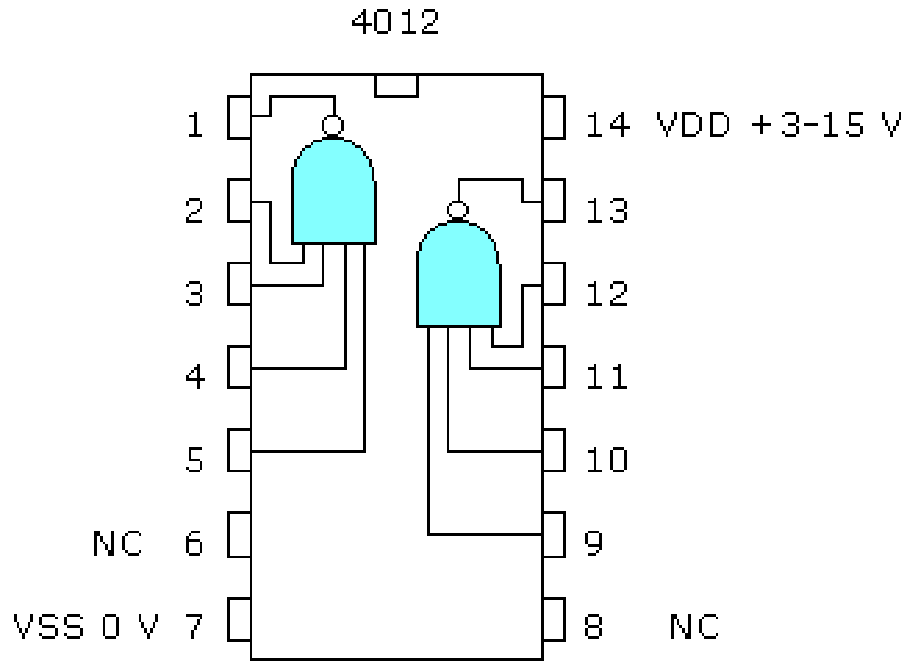
# IC of OR Gates



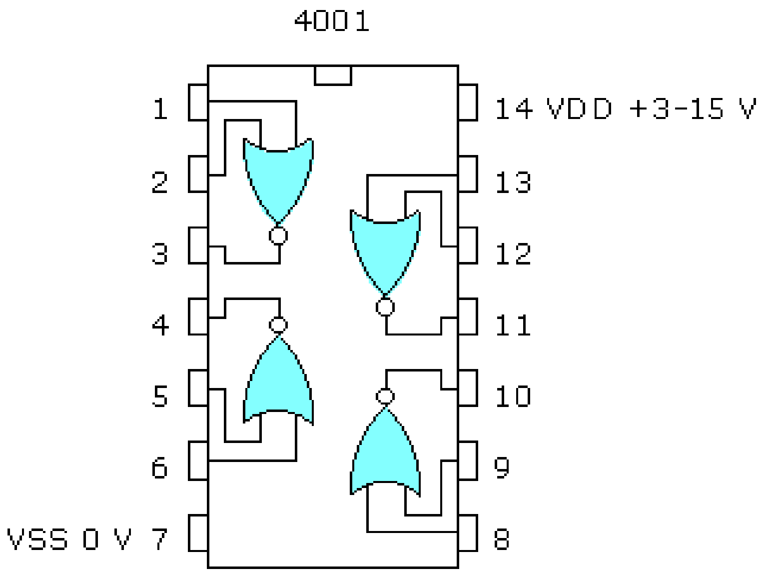



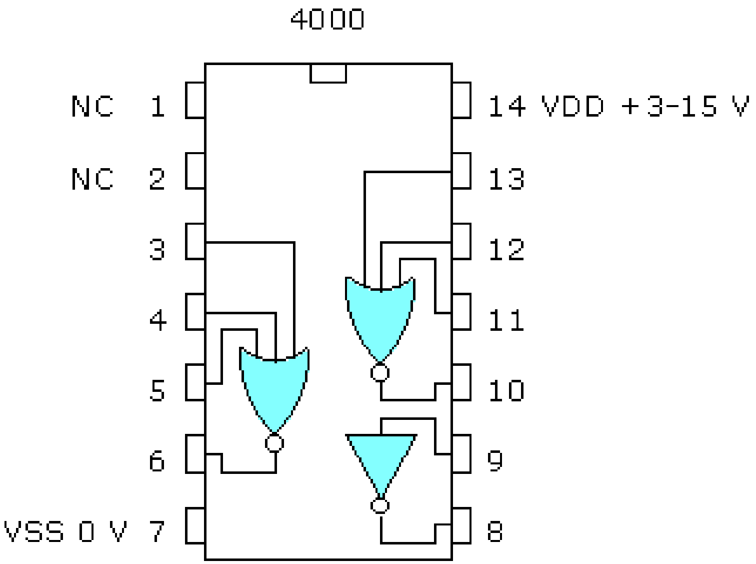
# IC of NAND Gates

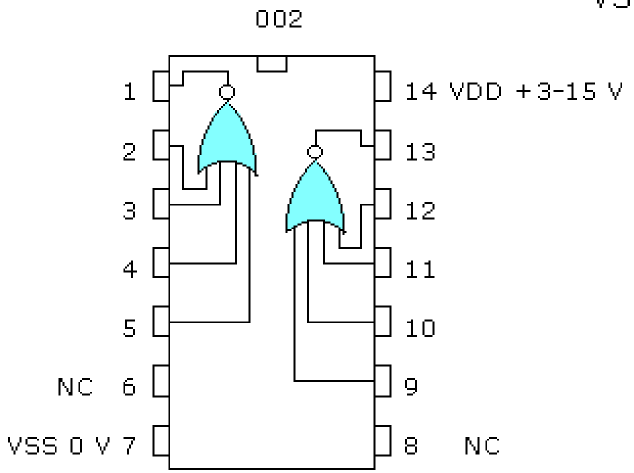


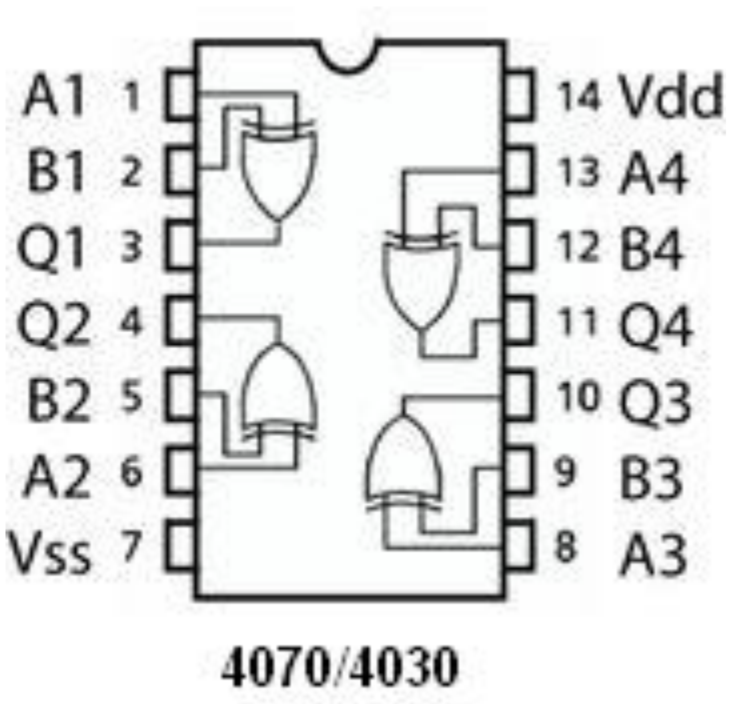
# IC of NOR Gates

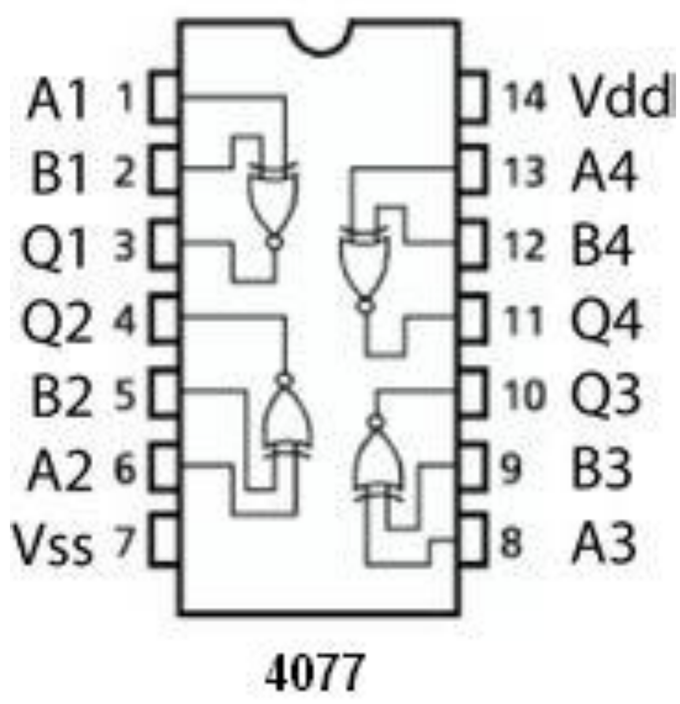


# IC of XOR/XNOR Gates



# List of Logic Gate ICs

- AND: 74LS08
- OR: 74LS32
- NOT: 74LS04/05
- NAND: 74LS00
- NOR: 74LS02
- XOR: 74LS136
- NXOR: 74LS266

Read more: **MultiSim**