Mahidol University
Wisdom of the Land

Logic and Computer Design Fundamentals

Chapter 2 – Combinational Logic Circuits

Part 2 – Additional Gates and Circuits

Asst.Prof.Dr. Preecha Tangworakitthaworn
Semester 2/2023

Overview

- **Part 1 – Review Basic Logic Gates and Behavior**
- **Part 2 – Additional Gates and Circuits**
  - **Other Gate Types**
  - **Exclusive-OR Operator and Gates**

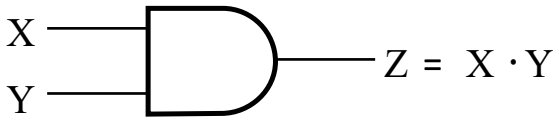
Part 1 – Review Basic Logic Gates

What is the logic gate?

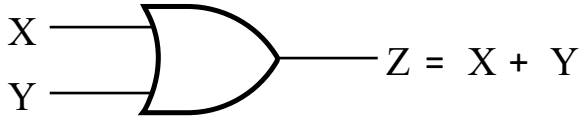
A logic gate is a device performing a Boolean Logic Operation on one or more binary inputs and then give a single binary output.

Part 1 – Review Basic Logic Gates

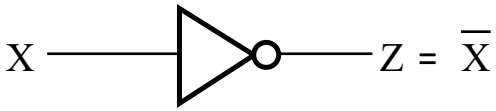
Three basic symbols for Logic gates:



AND gate

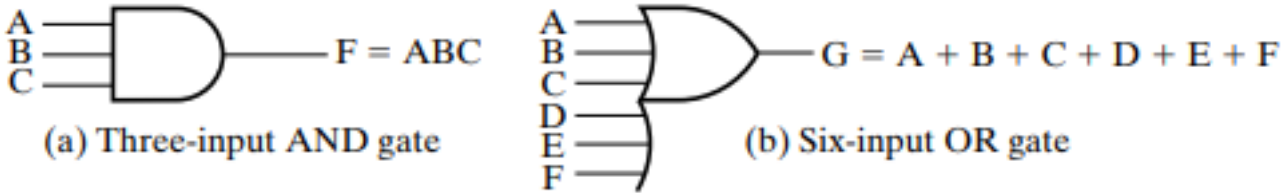


OR gate



NOT gate or
inverter

Basic Logic Gates formation



(a) Three-input AND gate (b) Six-input OR gate

A single-output Boolean function is a mapping from each of the possible combinations of input values 0 and 1 on the function variables to output value 0 or 1.

Behavior and Representation

- **Truth table** is a representation of gates' behavior described its basic logic operations:

AND		
X	Y	Z = X·Y
0	0	0
0	1	0
1	0	0
1	1	1

OR		
X	Y	Z = X+Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
X	$Z = \overline{X}$
0	1
1	0

Behavior and Representation

- **Truth table** for more variables can be listed the combination as follows:

2 Variables= 2^2

	A	B
0	0	0
1	0	1
2	1	0
3	1	1

4 combinations

3 Variables= 2^3

	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

8 combinations

4 Variables= 2^4

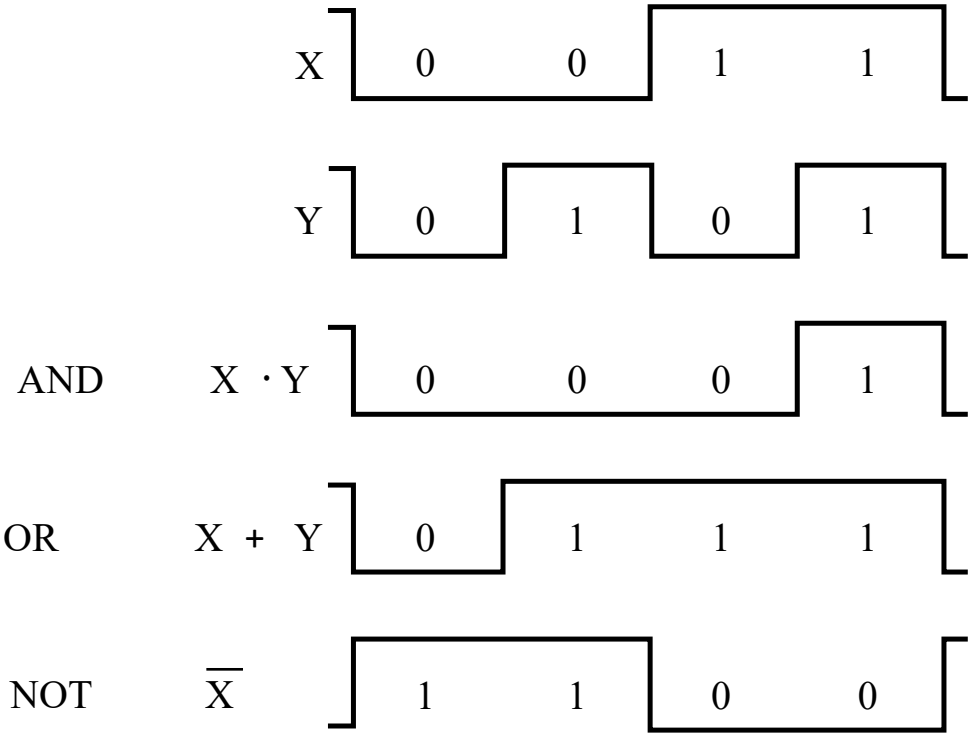
	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
.	0	0	1	1
.	0	1	0	0
	0	1	0	1
	0	1	1	0
	0	1	1	1

A	B	C	D	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	14
1	1	1	1	15

16 combinations

Behavior and Representation

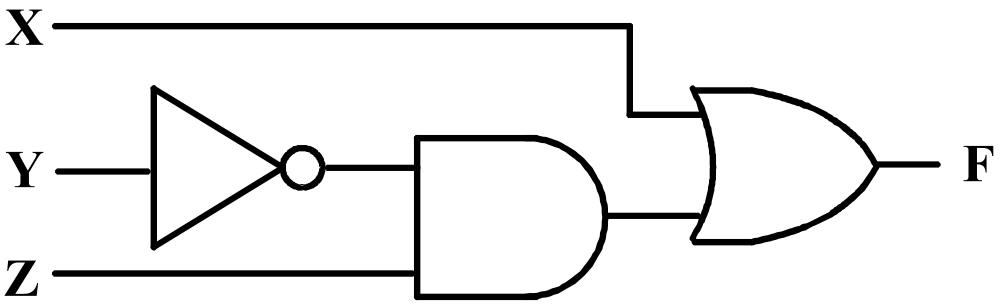
- **Timing Diagram** is a representation of gates' behavior described its logic operations based on Time:



Behavior and Representation

- **Logic Diagram (or Circuit Diagram)** consists of gates and symbols that can directly replace an expression in Boolean function or Boolean equation:

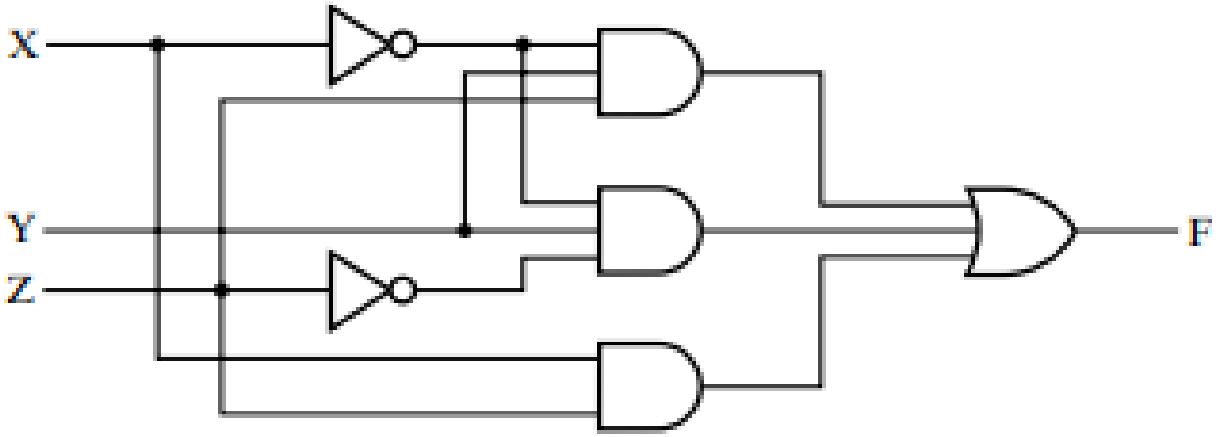
Logic Diagram



Boolean Equation

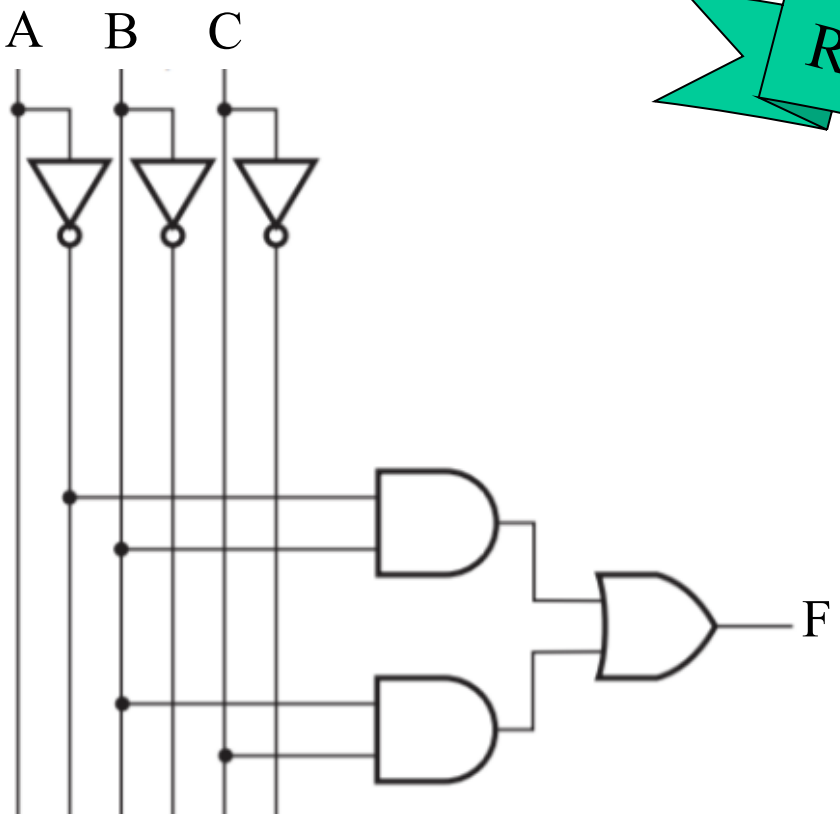
$$F = X + \overline{Y}\,Z$$

Example of Circuit Diagram



$$F(X,Y,Z) = \bar{X}YZ + \bar{X}Y\bar{Z} + XZ$$

Example of Circuit Diagram



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

Software for Drawing [online]

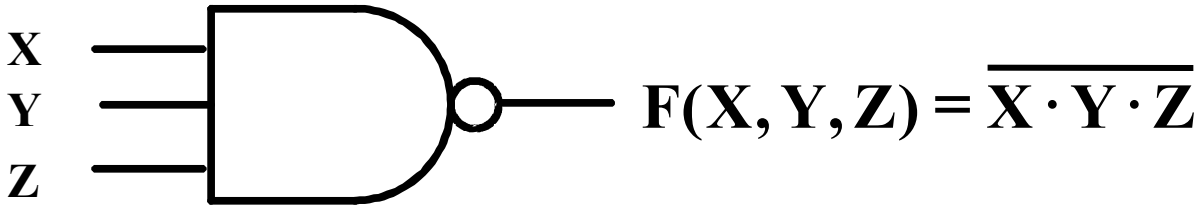
- **https://www.circuitlab.com/**
- **https://cloud.smartdraw.com/**
- **https://www.digikey.it/schemeit/project/**

Part 2 – Other Gate Types

- Why?
  - Implementation feasibility and low cost
  - Power in implementing Boolean functions
  - Convenient conceptual representation
- Gate classifications
  - Primitive (or basic) gate - a gate that can be described using a single primitive operation type (AND or OR) plus an optional inversion (NOT).
  - Complex gate - a gate that requires more than one primitive operation type for its description.

NAND Gate

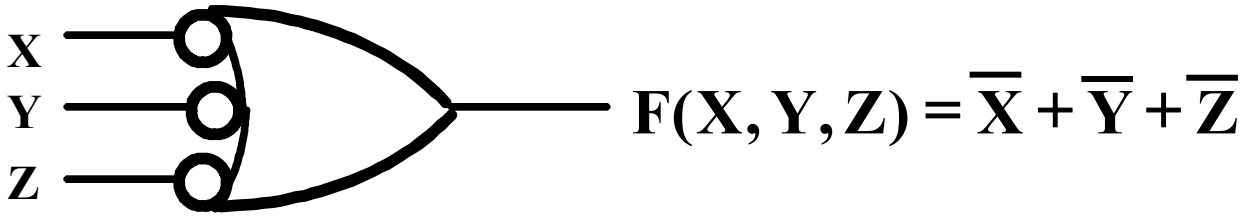
- The basic NAND gate has the following symbol, illustrated for three inputs:
  - AND-Invert (NAND)



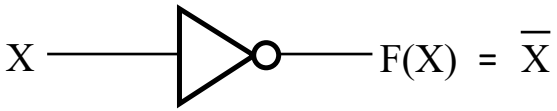
- NAND represents NOT AND, i. e., the AND function with a NOT applied. The symbol shown is an AND-Invert. The small circle ("bubble") represents the invert function.

NAND Gate (continued)

- Applying DeMorgan's Law gives Invert-OR (NAND)

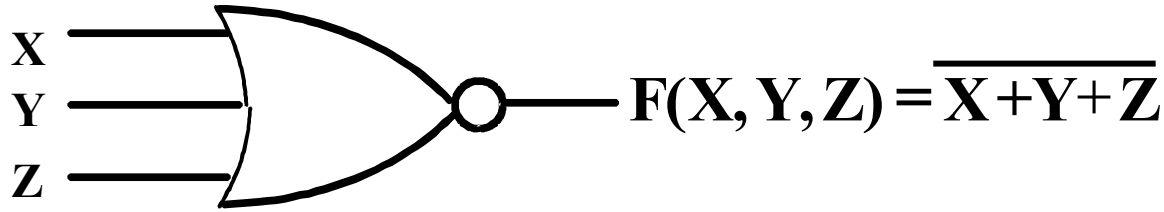


- This NAND symbol is called Invert-OR, since inputs are inverted and then operate OR together.
- AND-Invert and Invert-OR both represent the NAND gate. Having both makes visualization of circuit function easier.
- A NAND gate with one input degenerates to an inverter.



NOR Gate

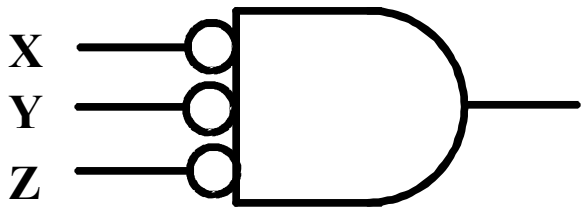
- The basic NOR gate has the following symbol, illustrated for three inputs:
  - OR-Invert (NOR)



- NOR represents NOT - OR, i. e., the OR function with a NOT applied. The symbol shown is an OR-Invert. The small circle ("bubble") represents the invert function.

NOR Gate (continued)

- Applying DeMorgan's Law gives Invert-AND (NOR)



- This NOR symbol is called Invert-AND, since inputs are inverted and then operate AND together.
- OR-Invert and Invert-AND both represent the NOR gate. Having both makes visualization of circuit function easier.
- A NOR gate with one input degenerates to an inverter.

Exclusive OR/ Exclusive NOR

- The *eXclusive OR* (*XOR*) function is an important Boolean function used extensively in logic circuits.
- The *eXclusive NOR (XNOR)* function is the complement of the XOR function.
- By definition, XOR and XNOR gates are complex gates.

Exclusive OR/ Exclusive NOR

Recommend

- Definitions
  - The XOR function is:

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

  - The eXclusive NOR (XNOR) function, otherwise known as *equivalence* is:

$$\overline{X \oplus Y} = XY + \overline{X}\,\overline{Y}$$

Truth Tables for XOR/XNOR

- Operator Rules: XOR

X	Y	X⊕Y
0	0	0
0	1	1
1	0	1
1	1	0

XNOR

X	Y	$\overline{(X \oplus Y)}$
0	0	1
0	1	0
1	0	0
1	1	1

- The XOR function means:

  X OR Y, but NOT BOTH

XOR/XNOR

The XOR function can be extended to 3 or more variables. Processing step by step will help you to not confuse!

A	B	A ⊕ B	C	(A⊕B) ⊕ C
0	0	0	0	0
0	0	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1

XOR/XNOR (Continued)

Recommend

- The XOR function can be extended to 3 or more variables. For more than 2 variables, it is called an *odd function* or *modulo 2 sum* (*Mod 2 sum*), not an XOR:

$$X \oplus Y \oplus Z = \overline{X}\,\overline{Y}\,Z + \overline{X}\,Y\,\overline{Z} + X\,\overline{Y}\,\overline{Z} + X\,Y\,Z$$

- The complement of the odd function is the even function.
- The XOR identities:

$$X \oplus 0 = X \qquad X \oplus 1 = \overline{X}$$

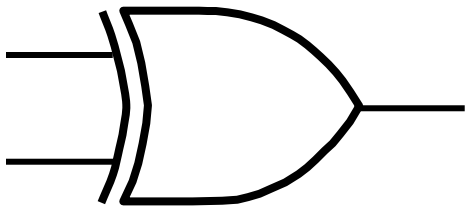
$$X \oplus X = 0 \qquad X \oplus \overline{X} = 1$$

$$X \oplus Y = Y \oplus X$$

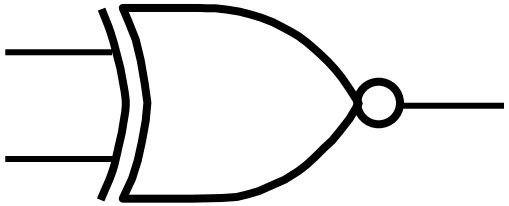
$$(X \oplus Y) \oplus Z = X \oplus (Y \oplus Z) = X \oplus Y \oplus Z$$

Symbols For XOR and XNOR

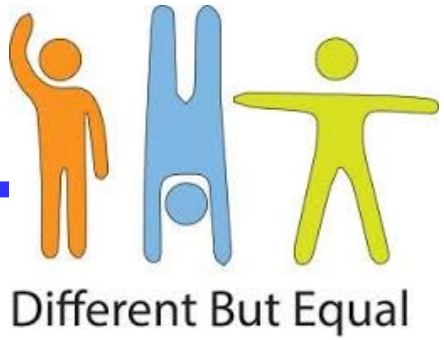
- **XOR symbol:**



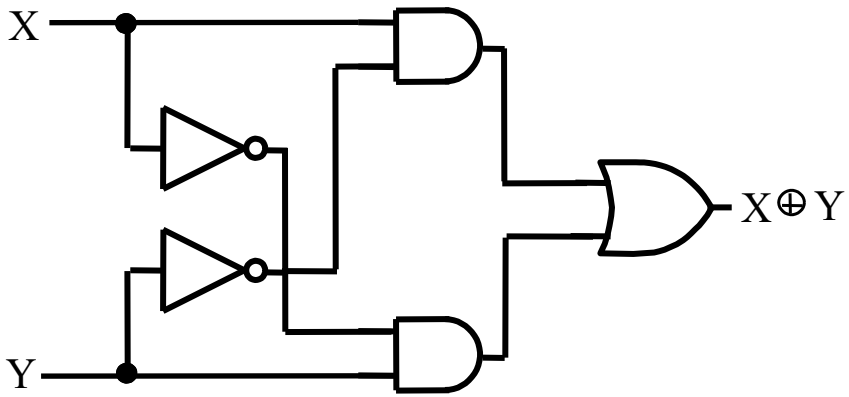
- **XNOR symbol:**



XOR Implementations



The simple implementation uses the following structure:



which is equivalent to.. XOR gate implementation: