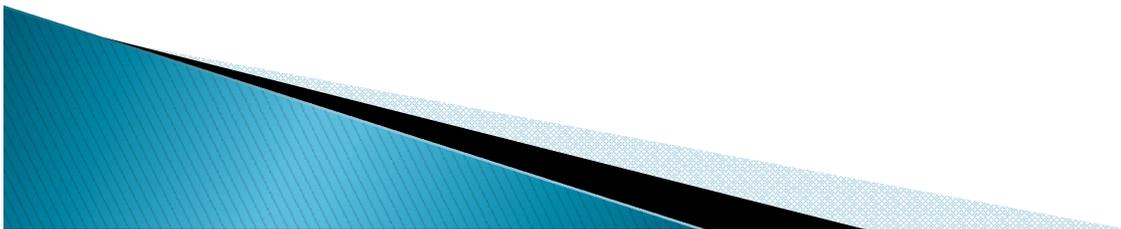


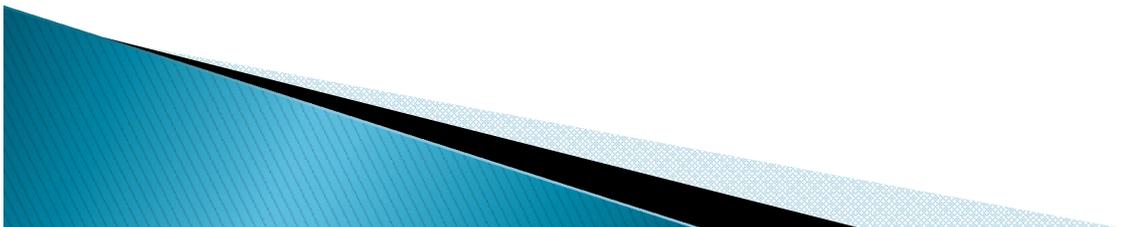
# Lecture 6

## Logic Gates and Circuits



# Goals

- ▶ Identify the basic gates and describe the behavior of each gate
- ▶ Describe how gates are implemented using transistors
- ▶ Combine basic gates into circuits
- ▶ Describe the behavior of a gate or circuit using Boolean expressions, truth tables, and logic diagrams



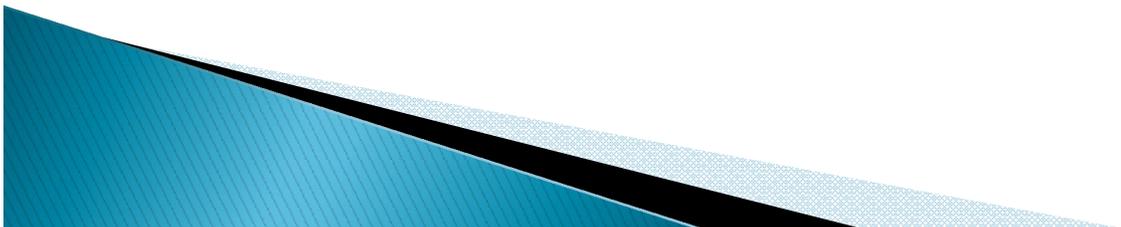
# Computers and Electricity

## Gate

A device that performs a basic operation on electrical signals.

## Circuits

Gates combined to perform more complicated tasks.



# Computers and Electricity

*How do we describe the behavior of gates and circuits?*

## Boolean expressions

Uses Boolean algebra, a mathematical notation for expressing two-valued logic

## Logic diagrams

A graphical representation of a circuit; each gate has its own symbol

## Truth tables

A table showing all possible input value and the associated output values

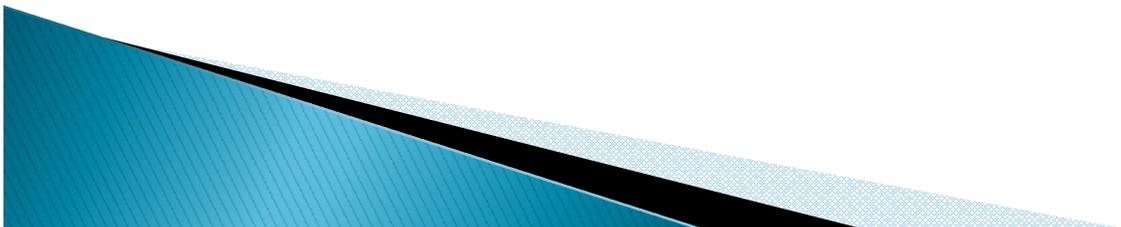
# Gates

Six types of gates

- NOT
- AND
- OR
- XOR
- NAND
- NOR

Typically, logic diagrams are black and white with gates distinguished only by their shape

We use **color** for emphasis (and fun)



# NOT Gate

A NOT gate accepts one input signal (0 or 1) and returns the opposite signal as output

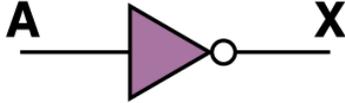
Boolean Expression	Logic Diagram Symbol	Truth Table						
$X = A'$		<table border="1"><thead><tr><th>A</th><th>X</th></tr></thead><tbody><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></tbody></table>	A	X	0	1	1	0
A	X							
0	1							
1	0							

Figure 4.1 Various representations of a NOT gate

# AND Gate

An AND gate accepts two input signals  
If both are 1, the output is 1; otherwise,  
the output is 0

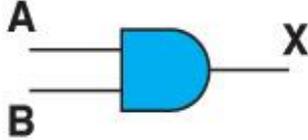
Boolean Expression	Logic Diagram Symbol	Truth Table															
$X = A \cdot B$		<table border="1"><thead><tr><th>A</th><th>B</th><th>X</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1
A	B	X															
0	0	0															
0	1	0															
1	0	0															
1	1	1															

Figure 4.2 Various representations of an AND gate

# OR Gate

An OR gate accepts two input signals  
If both are 0, the output is 0; otherwise,  
the output is 1

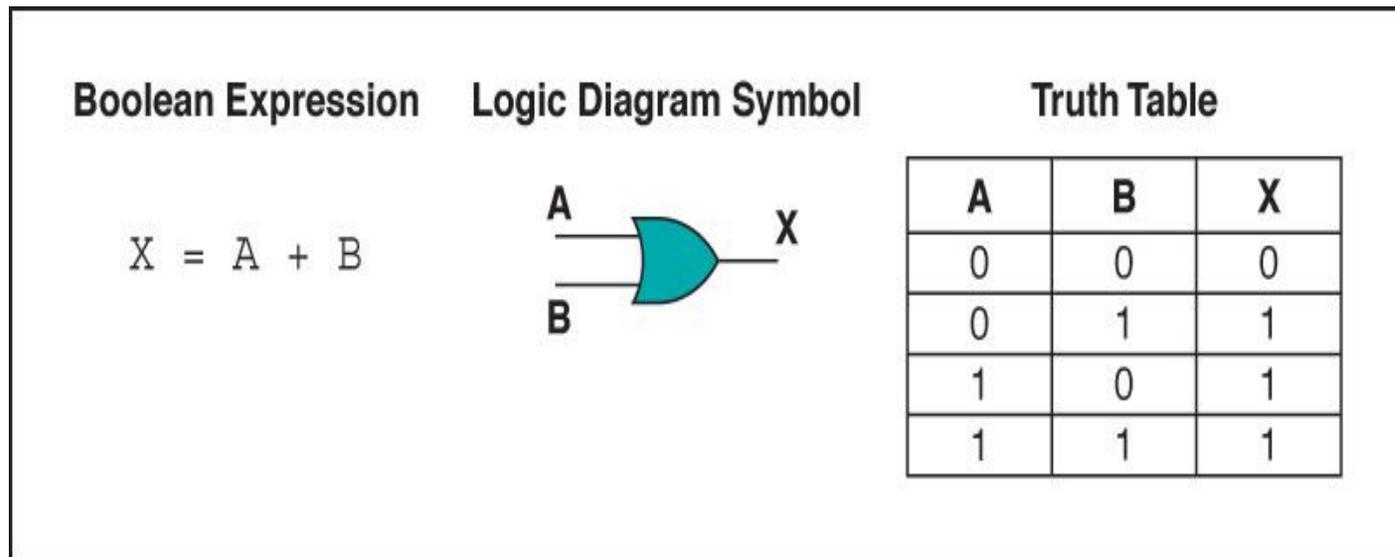


Figure 4.3 Various representations of a OR gate

# XOR Gate

An XOR gate accepts two input signals

If both are the same, the output is 0; otherwise, the output is 1

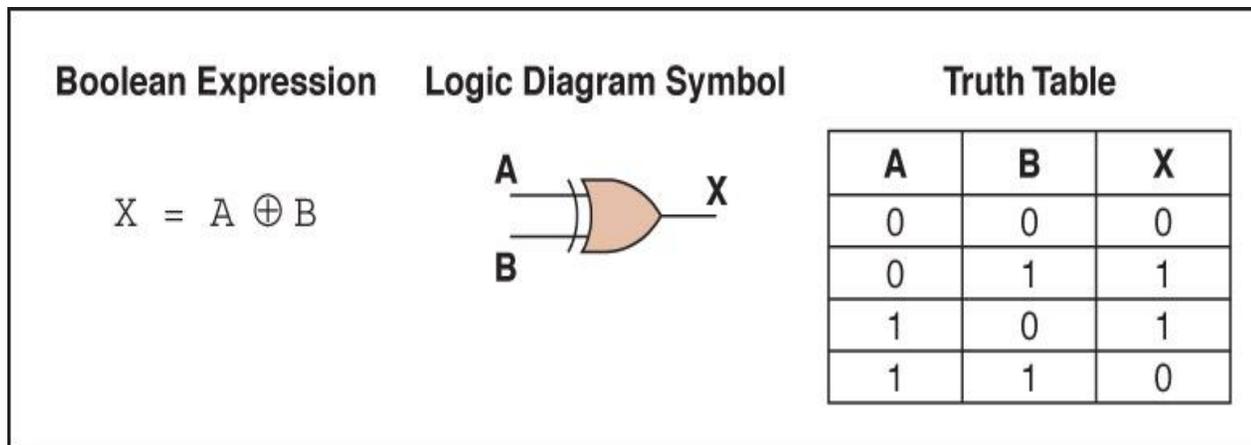


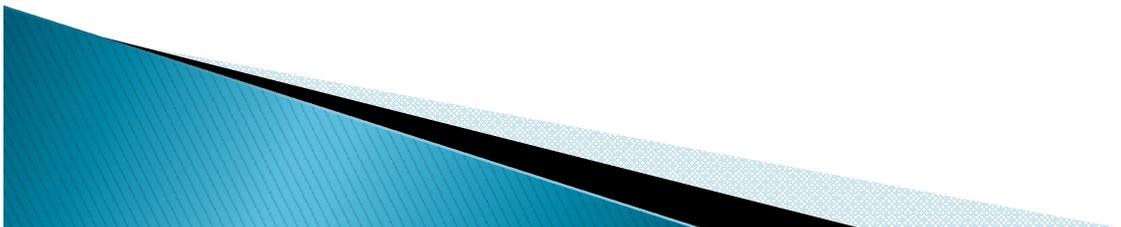
Figure 4.4

# XOR Gate

Note the difference between the XOR gate and the OR gate; they differ only in one input situation

When both input signals are 1, the OR gate produces a 1 and the XOR produces a 0

XOR is called the *exclusive OR*



# NAND Gate

The NAND gate accepts two input signals. If both are 1, the output is 0; otherwise, the output is 1.

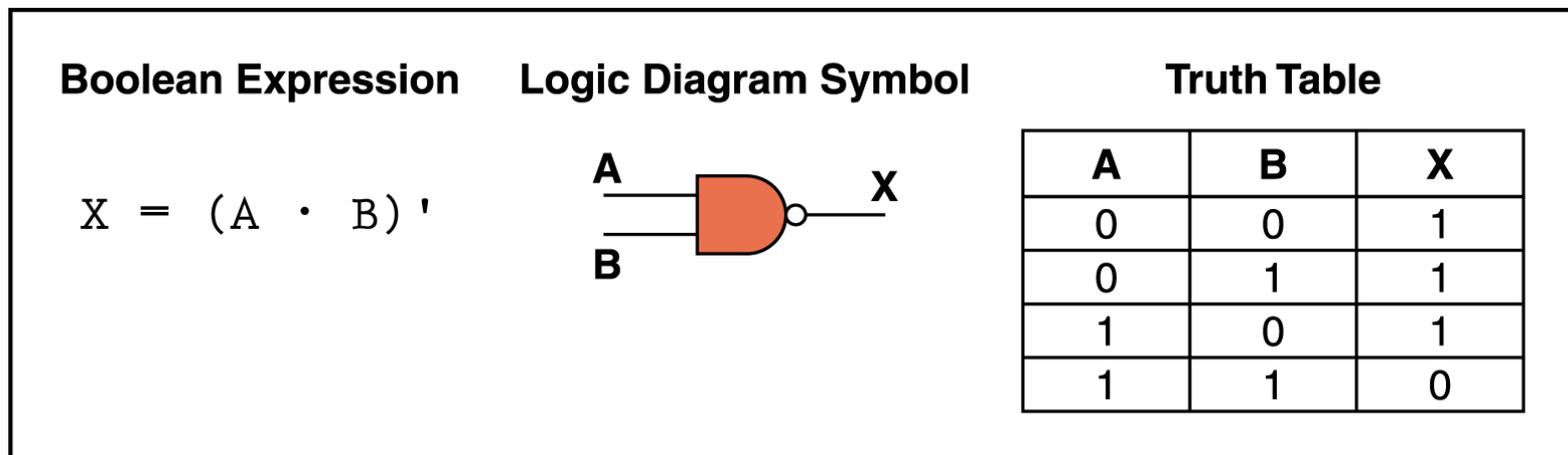


Figure 4.5 Various representations of a NAND gate

# NOR Gate

The NOR gate accepts two input signals  
If both are 0, the output is 1; otherwise,  
the output is 0

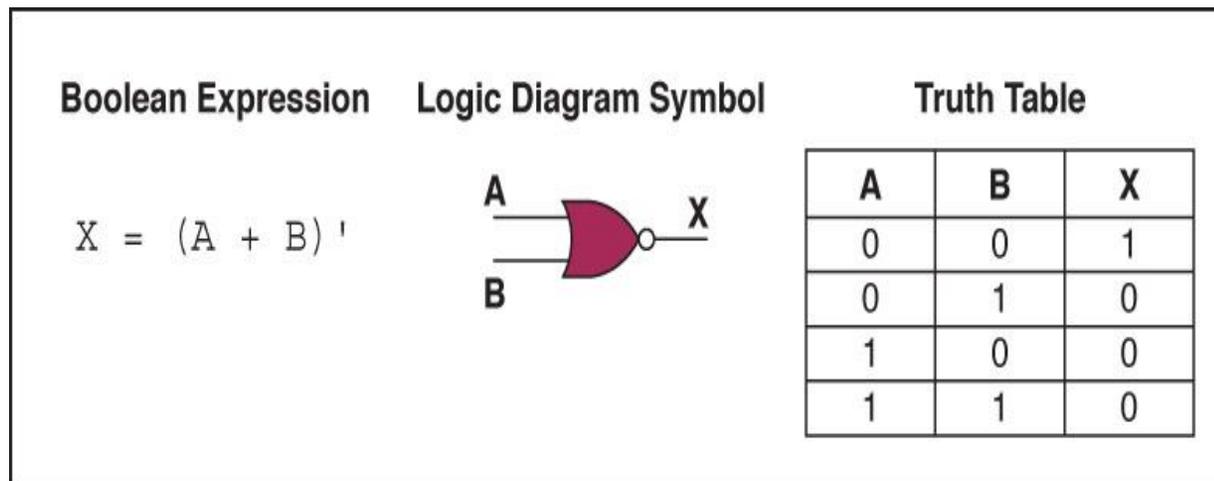


Figure 4.6 Various representations of a NOR gate

# Review of Gate Processing

A **NOT** gate **inverts** its single input

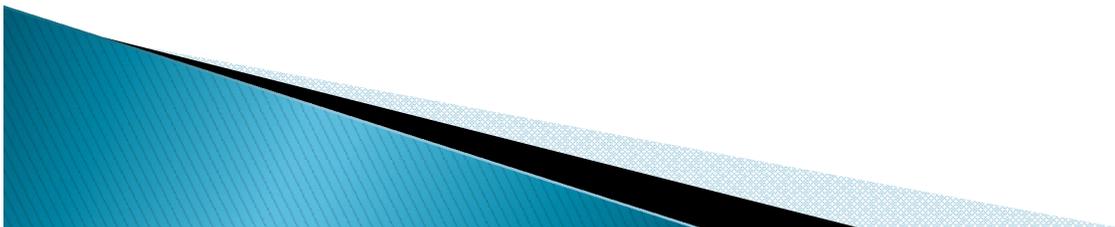
An **AND** gate produces **1** if **both** input values are **1**

An **OR** gate produces **0** if **both** input values are **0**

An **XOR** gate produces **0** if input values are the **same**

A **NAND** gate produces **0** if **both** inputs are **1**

A **NOR** gate produces a **1** if both inputs are **0**



# Gates with More Inputs

Gates can be designed to accept three or more input values

A three-input **AND** gate, for example, produces an output of **1** only if all input values are **1**

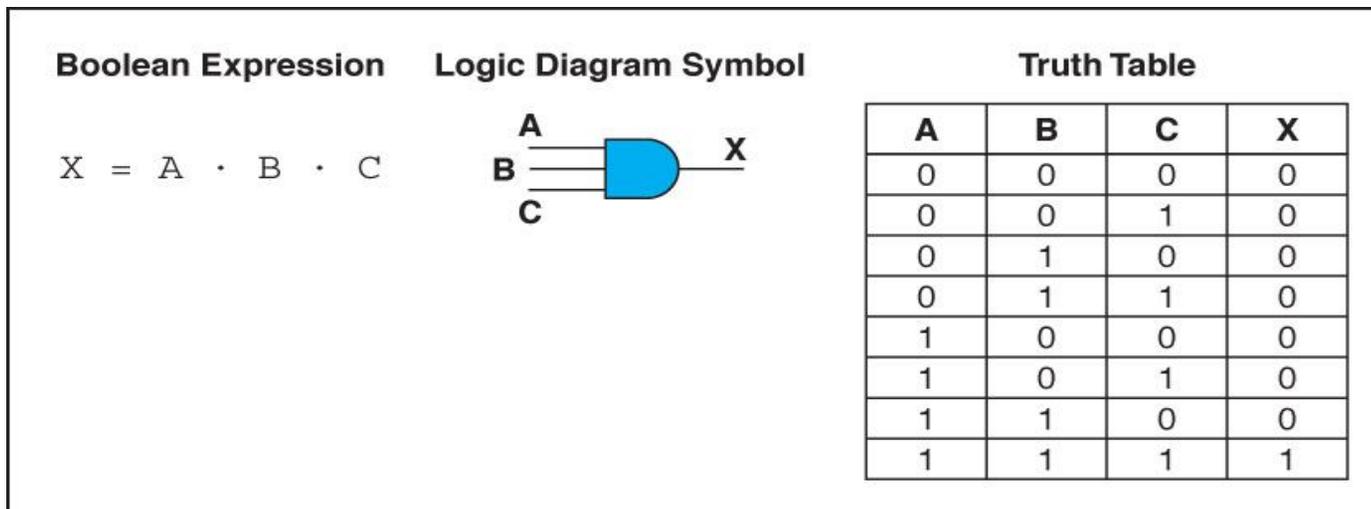


Figure 4.7 Various representations of a three-input AND gate

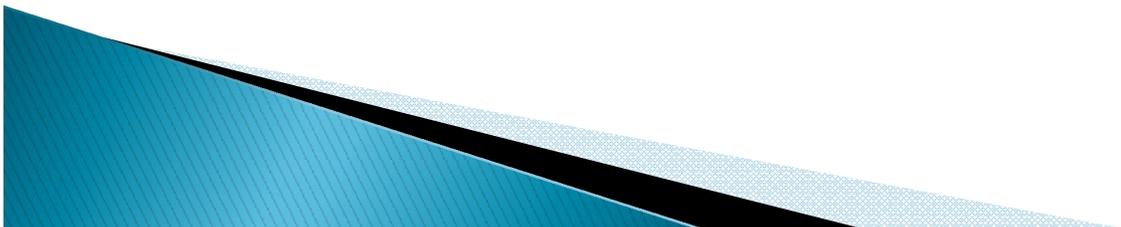
# Constructing Gates

## Transistor

A device that acts either as a wire that conducts electricity or as a resistor that blocks the flow of electricity, depending on the voltage level of an input signal

A transistor has no moving parts, yet acts like a switch

It is made of a **semiconductor** material, which is neither a particularly good conductor of electricity nor a particularly good insulator



# Constructing Gates

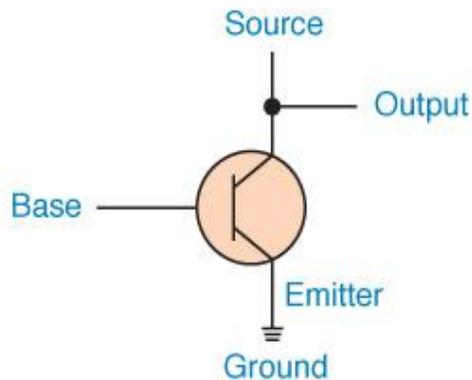


Figure 4.8 The connections of a transistor

A transistor has three terminals

A source

A base

An emitter, typically connected to a ground wire

If the electrical signal is grounded, it is allowed to flow through an alternative route to the ground (literally) where it can do no harm

# Constructing Gates

The easiest gates to create are the **NOT**, **NAND**, and **NOR** gates

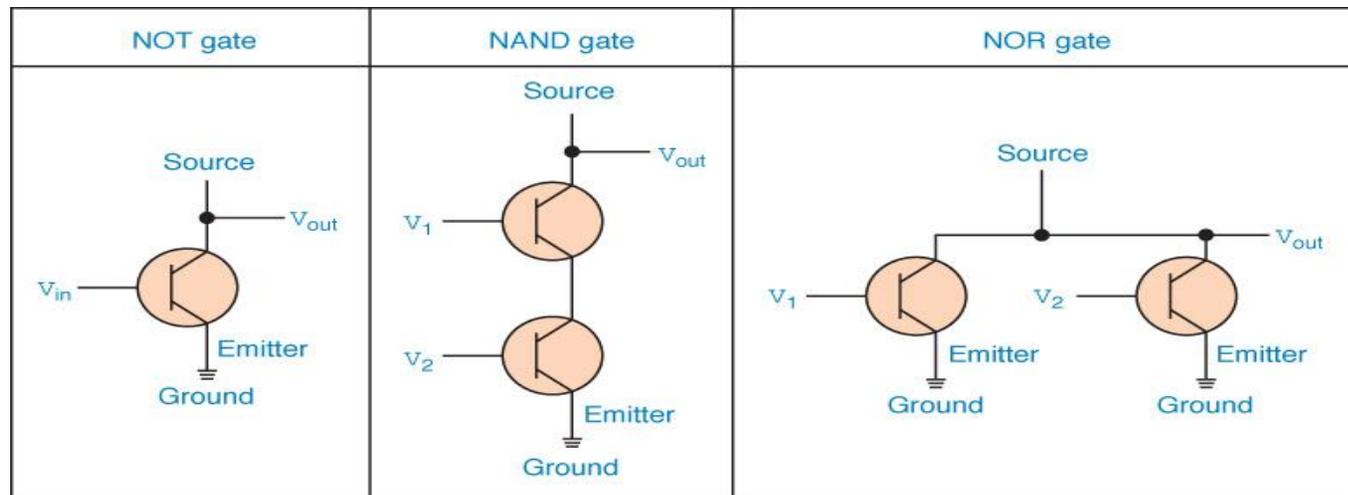


Figure 4.9 Constructing gates using transistors

# Circuits

## Combinational circuit

The input values explicitly determine the output

## Sequential circuit

The output is a function of the input values and the existing state of the circuit

We describe the circuit operations using

Boolean expressions

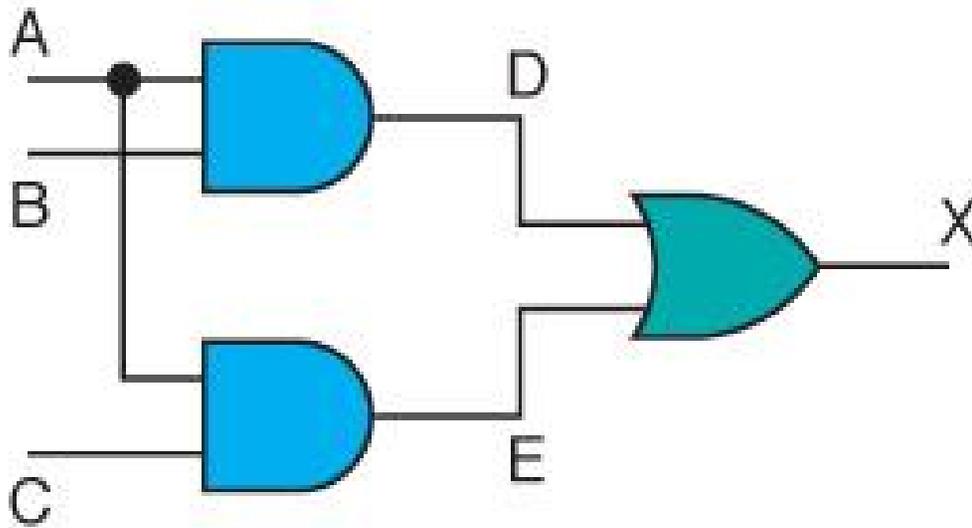
Logic diagrams

Truth tables

*Are you surprised?*

# Combinational Circuits

Gates are combined into circuits by using the output of one gate as the input for another



# Combinational Circuits

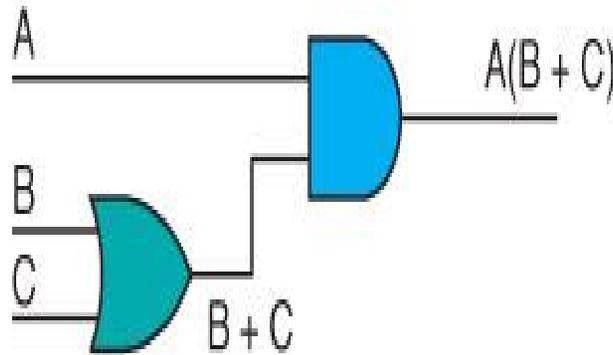
A	B	C	D	E	X
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

Three inputs require eight rows to describe all possible input combinations

This same circuit using a Boolean expression is  $(AB + AC)$

# Combinational Circuits

Consider the following Boolean expression  $A(B + C)$



A	B	C	$B + C$	$A(B + C)$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

*Does this truth table look familiar?*

*Compare it with previous table*

# Combinational Circuits

## Circuit equivalence

Two circuits that produce the same output for identical input

Boolean algebra allows us to apply provable mathematical principles to help design circuits

$A(B + C) = AB + BC$  (distributive law) so circuits must be equivalent



# Properties of Boolean Algebra

Property	AND	OR
Commutative	$AB = BA$	$A + B = B + A$
Associative	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive	$A(B + C) = (AB) + (AC)$	$A + (BC) = (A + B)(A + C)$
Identity	$A1 = A$	$A + 0 = A$
Complement	$A(A') = 0$	$A + (A') = 1$
DeMorgan's law	$(AB)' = A' \text{ OR } B'$	$(A + B)' = A'B'$

# Integrated Circuits

Integrated circuits (IC) are classified by the number of gates contained in them

Abbreviation	Name	Number of Gates
SSI	Small-Scale Integration	1 to 10
MSI	Medium-Scale Integration	10 to 100
LSI	Large-Scale Integration	100 to 100,000
VLSI	Very-Large-Scale Integration	more than 100,000

# Integrated Circuits

**Integrated circuit** (also called a *chip*)

A piece of silicon on which multiple gates have been embedded

Silicon pieces are mounted on a plastic or ceramic package with pins along the edges that can be soldered onto circuit boards or inserted into appropriate sockets



# Integrated Circuits

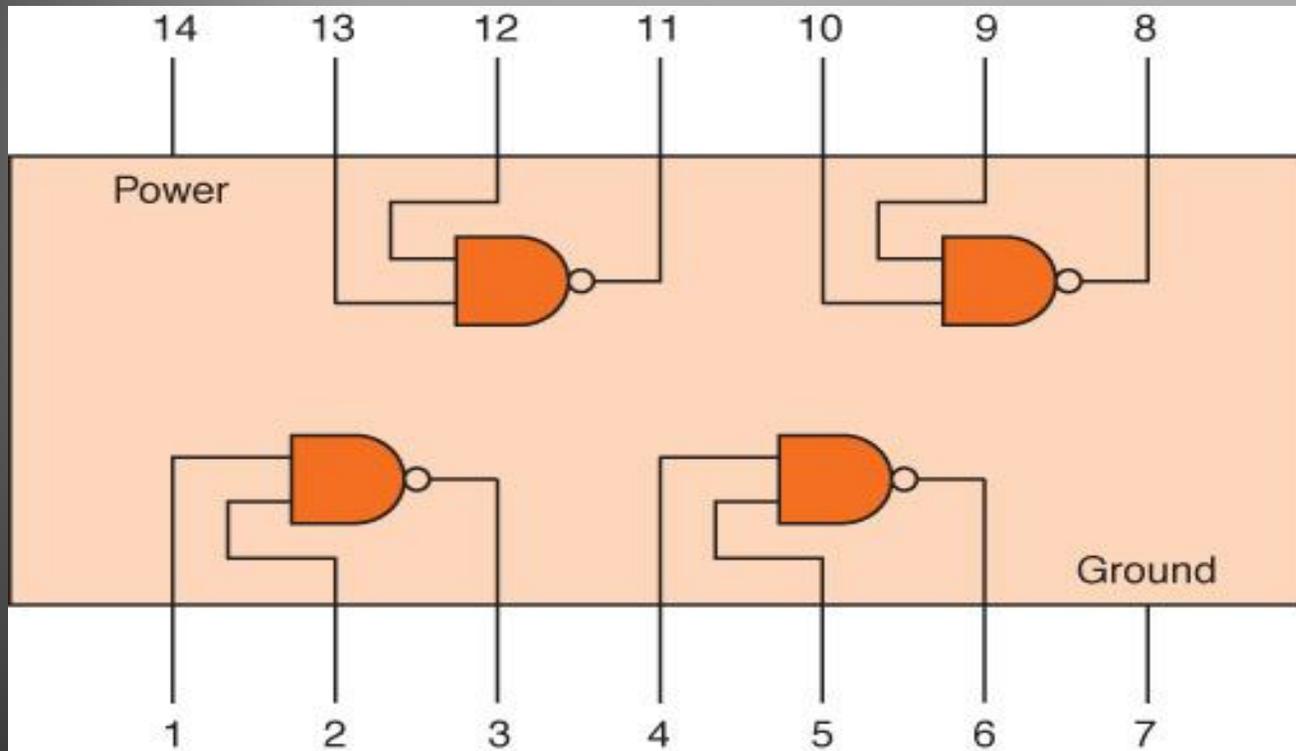
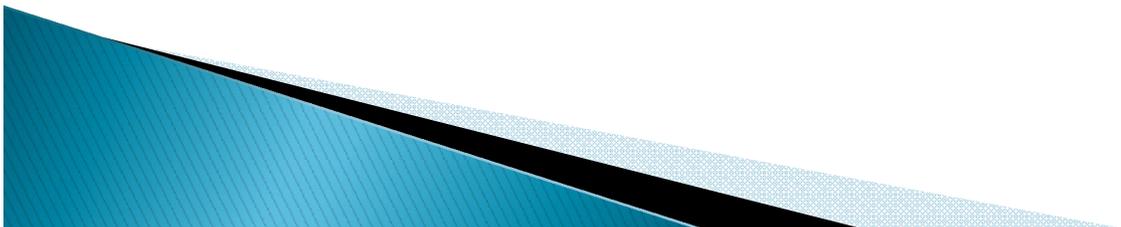


Figure 4.13 An SSI chip contains independent NAND gates

# CPU Chips

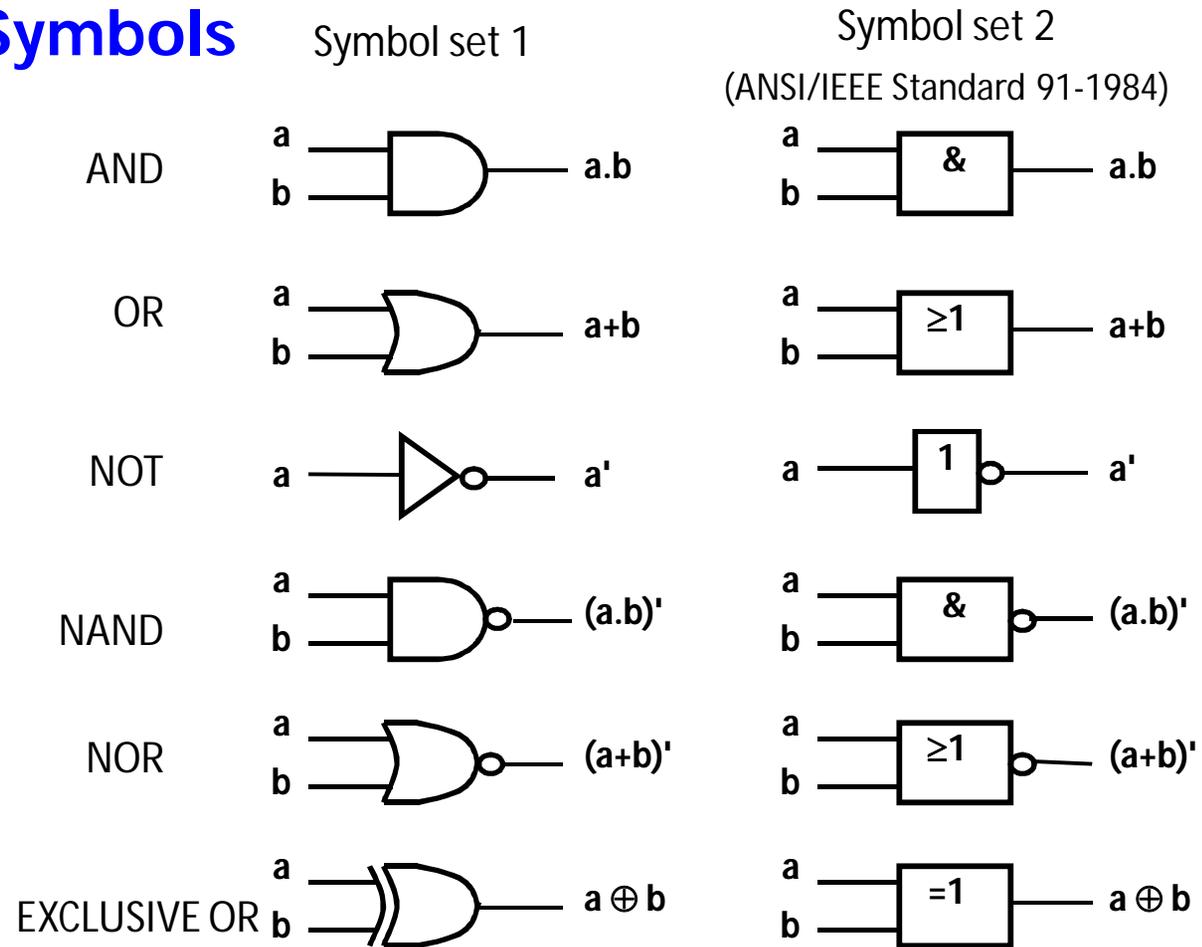
The most important integrated circuit in any computer is the **Central Processing Unit**, or CPU

Each CPU chip has a large number of pins through which essentially all communication in a computer system occurs



# Logic Gates

## Gate Symbols



# Assignment-6

- ▶ Why NAND and NOR are known as Universal Gates ?

