Chapter 4

Gates and Circuits

Chapter Goals

- Identify the basic gates and describe the behavior of each
- 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 values and the associated output values

Gates

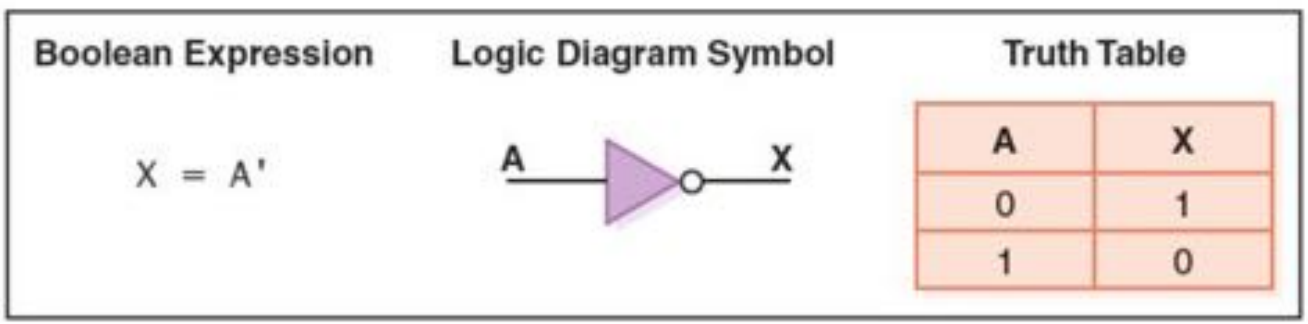
Six types of gates
- NOT
- AND
- OR
- XOR
- NAND
- NOR

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

We use color for clarity (and fun)

NOT Gate

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



A	X
0	1
1	0

FIGURE 4.1 Representations of a NOT gate

AND Gate

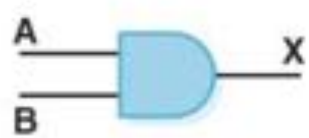
An AND gate accepts two input signals

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

Boolean Expression

$X = A \cdot B$

Logic Diagram Symbol



Truth Table

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

FIGURE 4.2 Representations of an AND gate

OR Gate

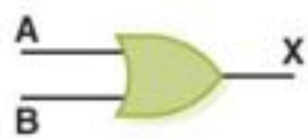
An OR gate accepts two input signals

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

Boolean Expression

$X = A + B$

Logic Diagram Symbol



Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

FIGURE 4.3 Representations of an 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

Boolean Expression

$X = A \oplus B$

Logic Diagram Symbol

Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

FIGURE 4.4 Representations of an XOR gate

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* because its output is 1 if (and only if):

- *either* one input *or* the other is 1,
- *excluding* the case that they both are

NAND Gate

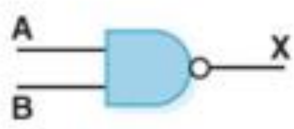
The NAND (“NOT of AND”) gate accepts two input signals

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

Boolean Expression

$X = (A \cdot B)'$

Logic Diagram Symbol



Truth Table

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

FIGURE 4.5 Representations of a NAND gate

NOR Gate

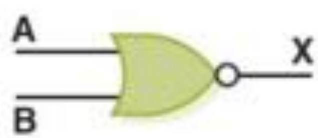
The NOR ("NOT of OR") gate accepts two inputs

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

Boolean Expression

$X = (A + B)'$

Logic Diagram Symbol



Truth Table

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

FIGURE 4.6 Representations of a NOR gate

Gates with More Inputs

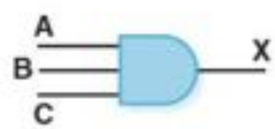
Some gates can be generalized to accept three or more input values

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

Boolean Expression

$X = A \cdot B \cdot C$

Logic Diagram Symbol



Truth Table

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

FIGURE 4.7 Representations of a three-input AND gate

Review of Gate Processing

Gate	Behavior
NOT	Inverts its single input
AND	Produces 1 if all input values are 1
OR	Produces 0 if all input values are 0
XOR	Produces 0 if both input values are the same
NAND	Produces 0 if all input values are 1
NOR	Produces 1 if all input values are 0

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

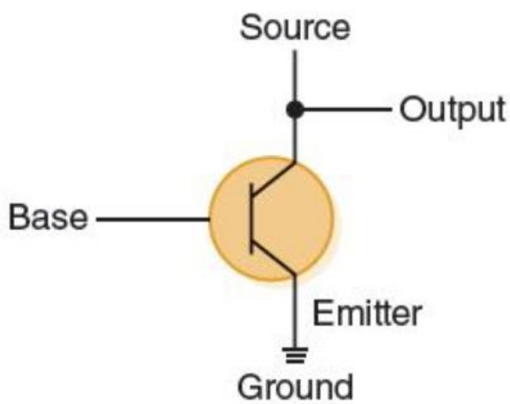


FIGURE 4.8 The connections of a transistor

Note: If an electrical signal is grounded, it is "pulled low" to zero volts

A transistor has three terminals

- A collector (or source)
- A base
- An emitter

What's the Output in Figure 4.8?

- If the Base signal is low, the transistor acts like an open switch, so the Output is the same as the Source
- If the Base signal is high, the transistor acts like a closed switch, so the Output is pulled low

What gate did we just describe?

Constructing Gates

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

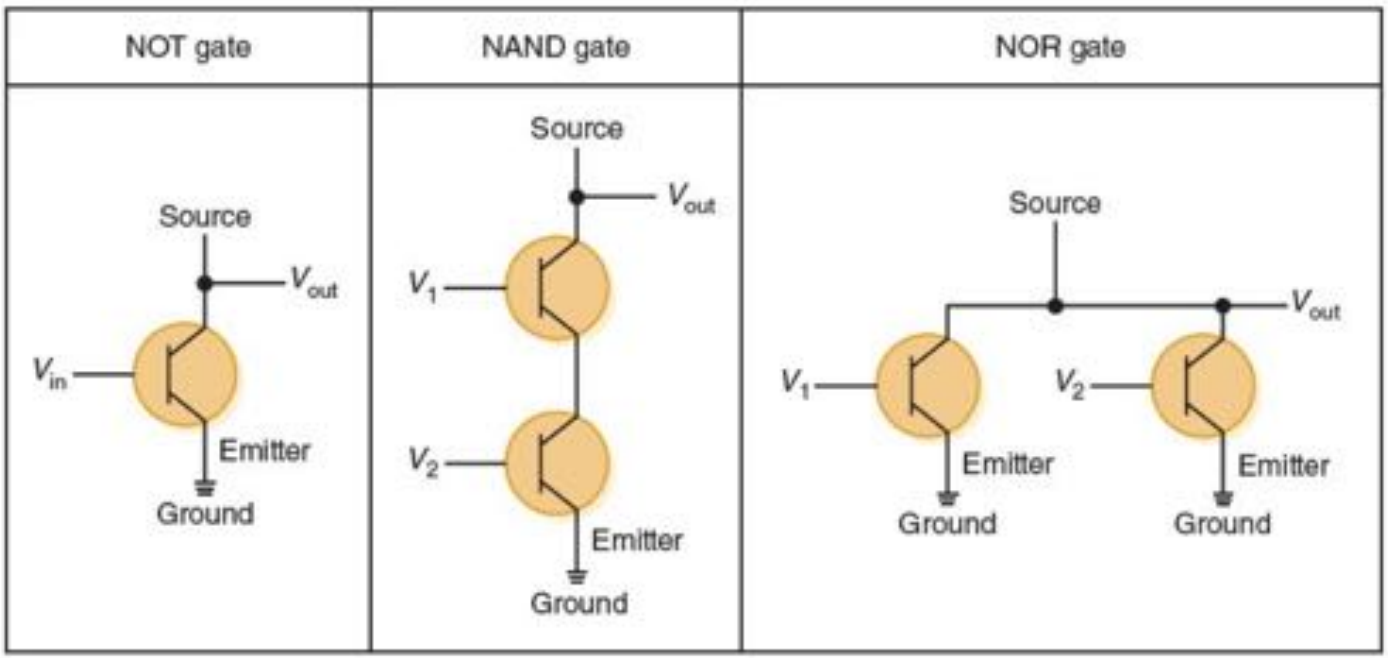


FIGURE 4.9 Constructing gates using transistors

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

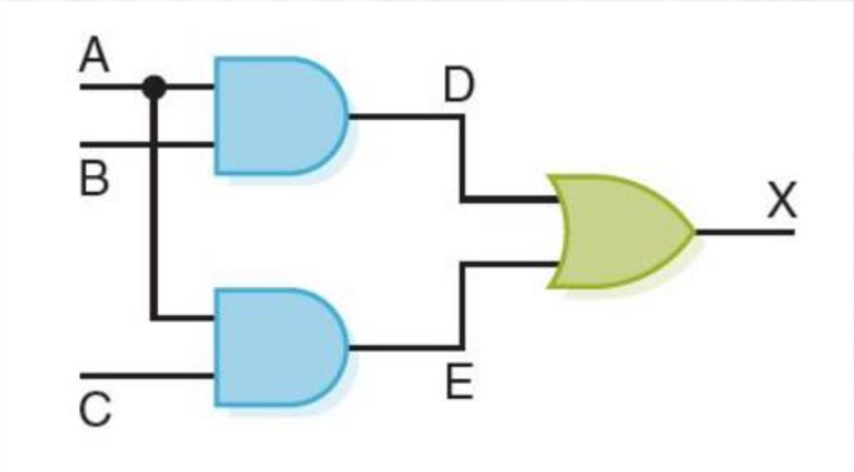
We describe the circuit operations using

- 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



This same circuit using a Boolean expression is AB + AC

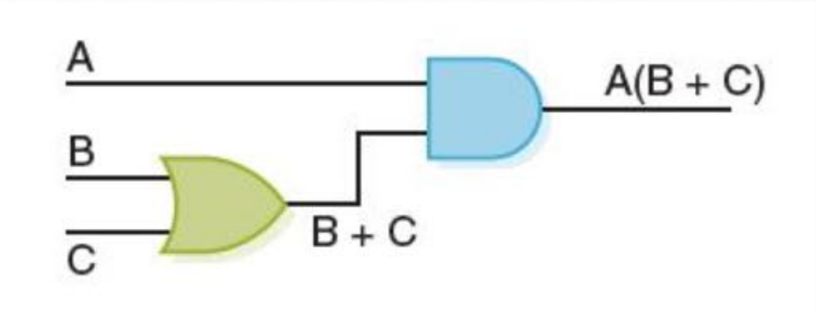
Combinational Circuits

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

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
De Morgan's law	(AB)' = A' OR B'	(A + B)' = A'B'