Computer Science Illuminated

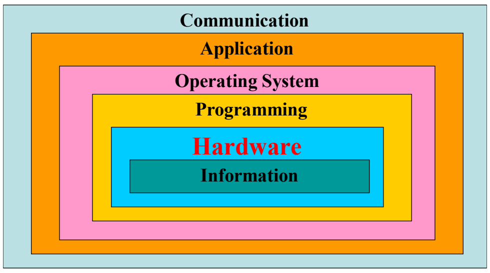
SECOND EDITION

Nell Dale • John Lewis

Chapter 4

Gates and Circuits

从这里开始



2-2

Chapter Goals

- Identify the basic gates and describe the behavior of each
- Describe the behavior of a gate or circuit using Boolean expressions, truth tables, and logic diagrams
- Compare and contrast a half adder and a full adder

4–3

Computers and Electricity

- **Gate**（门） A device that performs a basic operation on electrical signals
- **Circuits**（电路） Gates combined to perform more complicated tasks

4–4

Computers and Electricity

- There are three different, but equally powerful, notational methods for describing the behavior of gates and circuits
  - Boolean expressions
  - logic diagrams
  - truth tables

4–5

Constructing Gates

- **Transistor** A device that acts, depending on the voltage level of an input signal, either as a wire that conducts electricity or as a resistor that blocks the flow of electricity
  - 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, such as copper, nor a particularly good insulator, such as rubber

4–6

Constructing Gates

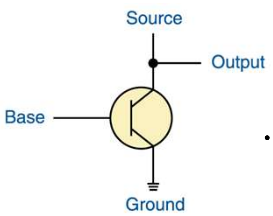


Figure 4.8 The connections of a transistor

- A transistor has three terminals
  - A source
  - A base
  - An emitter, typically connected to a ground wire
- Electrical sign and true table

Base		Output	
>1.4V	1	~Ground	0
<0.7V	0	~Source	1

4–7

Constructing Gates

- It turns out that, because the way a transistor works, the easiest gates to create are the NOT, NAND, and NOR gates

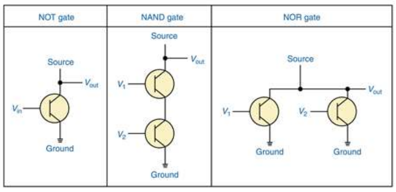


Figure 4.9 Constructing gates using transistors

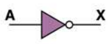
4–8

NOT（非） Gate

- A NOT gate accepts one input value and produces one output value

Boolean Expression: X = A'

Logic Diagram Symbol:



Truth Table:

A	X
0	1
1	0

Figure 4.1 Various representations of a NOT gate

4–9

NOT Gate

- By definition, if the input value for a NOT gate is 0, the output value is 1, and if the input value is 1, the output is 0
- A NOT gate is sometimes referred to as an *inverter*（反相器） because it inverts the input value

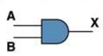
4–10

AND（与） Gate

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

Boolean Expression: X = A · B

Logic Diagram Symbol:



Truth Table:

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

Figure 4.2 Various representations of an AND gate

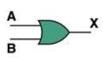
4–11

OR（或） Gate

- If the two input values are both 0, the output value 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 Various representations of a OR gate

4–12

Computers and Electricity

- **Boolean expressions（布尔表达式）** Expressions in Boolean algebra, a mathematical notation for expressing two-valued logic

  This algebraic notation are an elegant and powerful way to demonstrate the activity of electrical circuits

4–13

Computers and Electricity

- **Logic diagram（逻辑图）** A graphical representation of a circuit

  Each type of gate is represented by a specific graphical symbol

- **Truth table（真值表）** A table showing all possible input value and the associated output values

4–14

Basic Gates

- Let's examine the processing of the following six types of gates
  - NOT
  - AND
  - OR
  - XOR
  - NAND
  - NOR
- Typically, logic diagrams are black and white, and the gates are distinguished only by their shape

4–15

XOR（异或） Gate

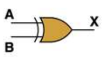
- XOR, or *exclusive* OR, gate
  - An XOR gate produces 0 if its two inputs are the same, and a 1 otherwise
  - Note the difference between the XOR gate and the OR gate; they differ only in one input situation
  - When both input signals are 1, the OR gate produces a 1 and the XOR produces a 0

4–16

XOR Gate

Boolean Expression: X = A ⊕ B

Logic Diagram Symbol:



Truth Table

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

Figure 4.4 Various representations of an XOR gate

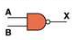
4–17

NAND and NOR Gates

- The NAND and NOR gates are essentially the opposite of the AND and OR gates, respectively

Boolean Expression: X = (A · B)'

Logic Diagram Symbol:



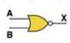
Truth Table

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

Figure 4.5 Various representations of a NAND gate

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 Various representations of a NOR gate

Review of Gate Processing

- A NOT gate inverts its single input value
- An AND gate produces 1 if both input values are 1
- An OR gate produces 1 if one or the other or both input values are 1

4–19

Review of Gate Processing

- An XOR gate produces 1 if one or the other (but not both) input values are 1
- A NAND gate produces the opposite results of an AND gate
- A NOR gate produces the opposite results of an OR gate

4–20

课堂练习：用门电路计算补码

- 补码的计算？……

C	I	I'	O	Cnext
0	1	0	0	0
0	0	1	1	0
1	1	0	1	0
1	0	1	0	1

让我们观察：
（1）I'和I是什么关系？
（2）O和C，I'是什么关系？
（3）Cnext和C，I'是什么关系？

4–21

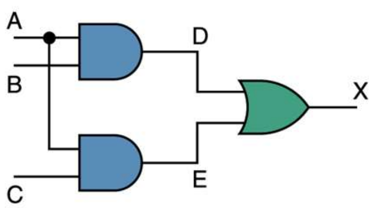
Circuits

- Two general categories
  - In a **combinational circuit**, the input values explicitly determine the output
  - In a **sequential circuit**, the output is a function of the input values as well as the existing state of the circuit
- As with gates, we can describe the operations of entire circuits using three notations
  - Boolean expressions
  - logic diagrams
  - truth tables

4–22

Combinational Circuits

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



4–23

Combinational Circuits

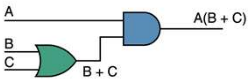
A	B	C	D	E	X
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

- Because there are three inputs to this circuit, eight rows are required to describe all possible input combinations
- This same circuit using Boolean algebra is (AB + AC)

4–24

Now let's go the other way; let's take a Boolean expression and draw

- 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

- Now compare the final result column in this truth table to the truth table for the previous example
  - **They are identical**

4–25

课堂练习

- 给定布尔表达式：O＝AB＋AC
- （1）画出逻辑图
- （2）画出真值表

4–26

Now let's go the other way; let's take a Boolean expression and draw

- We have therefore just demonstrated **circuit equivalence**
  – That is, both circuits produce the exact same output for each input value combination
- Boolean algebra allows us to apply provable mathematical principles to help us design logical circuits

4–27

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' OR B'	(A + B)' = A'B'

4–28

课堂练习：证明等价

- 给定布尔表达式 A＋A’ = 1
- （1）写出A+A’的真值表。

4–29

Adders

- At the digital logic level, addition is performed in binary
- Addition operations are carried out by special circuits called, appropriately, **adders**

4–30

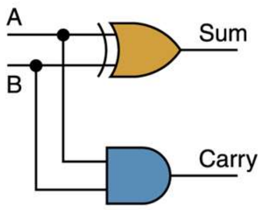
Adders

- The result of adding two binary digits could produce a *carry value*
- Recall that 1 + 1 = 10 in base two
- A circuit that computes the sum of two bits and produces the correct carry bit is called a **half adder**

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

4–31

Adders



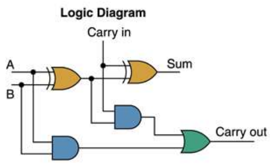
- Circuit diagram representing a half adder
- Two Boolean expressions:

$$\text{sum} = A \oplus B$$
$$\text{carry} = AB$$

4–32

Adders

- A circuit called a **full adder** takes the carry-in value into account



Truth Table

A	B	Carry-in	Sum	Carry-out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 4.10 A full adder

4–33

练习

- Draw the Truth Table
  - $((A+B) \oplus C)'$
- Use truth tables to show that the three-variable form of DeMorgan's Law is true; that is, the following equation holds:

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

4–34

Circuits as Memory

- Digital circuits can be used to store information
- These circuits form a sequential circuit, because the output of the circuit is also used as input to the circuit
- More about latch

4–35

Circuits as Memory

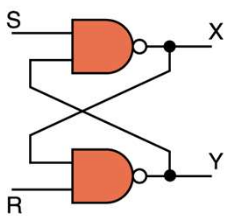


Figure 4.12 An S-R latch

- An S-R latch stores a single binary digit (1 or 0)
- There are several ways an S-R latch circuit could be designed using various kinds of gates

4–36

Circuits as Memory

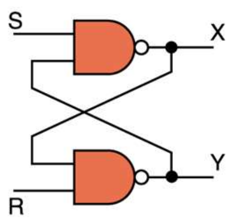


Figure 4.12 An S-R latch

- The design of this circuit guarantees that the two outputs X and Y are always complements of each other
- The value of X at any point in time is considered to be the current state of the circuit
- Therefore, if X is 1, the circuit is storing a 1; if X is 0, the circuit is storing a 0

4–37

Integrated Circuits

- **Integrated circuit** (also called a *chip*) A piece of silicon on which multiple gates have been embedded

  These 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

4–38

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

4–39

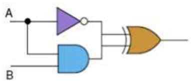
Knowledge : Intel CPUs

years	bits	type	width	clock rate	Transistors	package
1971	4	4004	10μm	740kHz	2,300	
1972	8	8008	10μm	500kHz	3,500	
1974	8/16	8080	3μm	2MHz	6,000	
1975	8/16	8085	3μm	3MHz	6,500	
1978	16/20	8086	3μm	8MHz	29,000	
1982	16/24	80286	1.5μm	16MHz	134,000	
1985	32	80386DX	1μm	33MHz	275,000	
1989	32	80486DX	0,8μm	50MHz	1.2m	
1993	32	Pentium	0,8μm	66MHz	3.1m	273 PGA
1997	32	Pentium II	0,35μm	266MHz	7.5m	241 Slot1
1999	32	Pentium III	0,25μm	533MHz	28.1m	

4–40

作业（part 1 of 2）

1) Give the three representations of an AND gate and say in your words what AND means.
2) Give the three representations of an XOR gate and say in your words what XOR means.
3) Draw a circuit diagram corresponding to the following Boolean expression: (A + B)(B + C)
4) Show the behavior of the following circuit with a truth table:



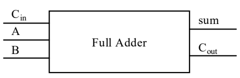
5) What is circuit equivalence? Use truth table to prove the following formula.

(AB)’ = A’ + B’

4–41

作业（part 2 of 2）

6) There are eight 1bit full adder integrated circuits. Combine them to 8bit adder circuit using the following box diagram.



7) Logical binary operations can be used to modify bit pattern. Such as

$$(X_8X_7X_6X_5X_4X_3X_2X_1)_2 \text{ and } (00001111)_2 = (0000X_4X_3X_2X_1)_2$$

We called that $(00001111)_2$ is a mask which only makes low 4 bits to work.
Fill the follow expression

(1) $(X_8X_7X_6X_5X_4X_3X_2X_1)_2$ or $(00001111)_2 = (\quad\quad)_2$
(2) $(X_8X_7X_6X_5X_4X_3X_2X_1)_2$ xor $(00001111)_2 = (\quad\quad)_2$
(3) $((X_8X_7X_6X_5X_4X_3X_2X_1)_2$ and $(11110000)_2)$
or (not $(X_8X_7X_6X_5X_4X_3X_2X_1)_2$ and $(00001111)_2) = (\quad\quad)_2$

4–42

作业（part 3 of 3）

使用维基百科，解释以下概念。
1)Logic gate
2)Boolean algebra

自学存储电路。维基百科：“Flip-flop”，选择中文：
1)Flip-flop 中文翻译是？
2)How many bits information does a SR latch store?

2-43