



# 第二章 数据的机器级表示与处理

数值数据的表示

非数值数据的表示

数据的存储

数据的运算

# 本章目录



## – 第一讲：数值数据的表示

- 定点数的编码表示、浮点数的表示
- C语言程序的整数类型和浮点数类型

## – 第二讲：非数值数据的表示、数据的存储

- 逻辑值、西文字符、汉字字符
- 数据宽度单位、大端/小端、对齐存放

## – 第三讲：数据的运算

- 按位运算\逻辑运算\移位运算
- 位扩展和位截断运算
- 无符号和带符号整数的加减运算
- 无符号和带符号整数的乘除运算
- 变量与常数之间的乘除运算
- 浮点数的加减乘除运算



# 第一讲：数值数据的表示

## 主要内容

- 定点数的表示
  - 进位计数制
  - 定点数的二进制编码
    - 原码、补码、移码表示
  - 定点整数的表示
    - 无符号整数、带符号整数
- 浮点数格式和表示范围
- 浮点数的规格化
- IEEE754浮点数标准
  - 单精度浮点数、双精度浮点数
  - 特殊数的表示形式
- C语言程序中的整数类型、浮点数类型
- 十进制数表示



# 数值数据表示的三要素

即：要确定一个数值数据的值必须先确定这三个要素。

例如，机器数 01011001 的值是多少？

## 进位计数制

- 十进制、二进制、十六进制、八进制数及其相互转换

## 定/浮点表示 (解决小数点问题)

- 定点整数、定点小数
- 浮点数 (可用一个定点小数和一个定点整数来表示)

## 定点数的编码 (解决正负号问题)

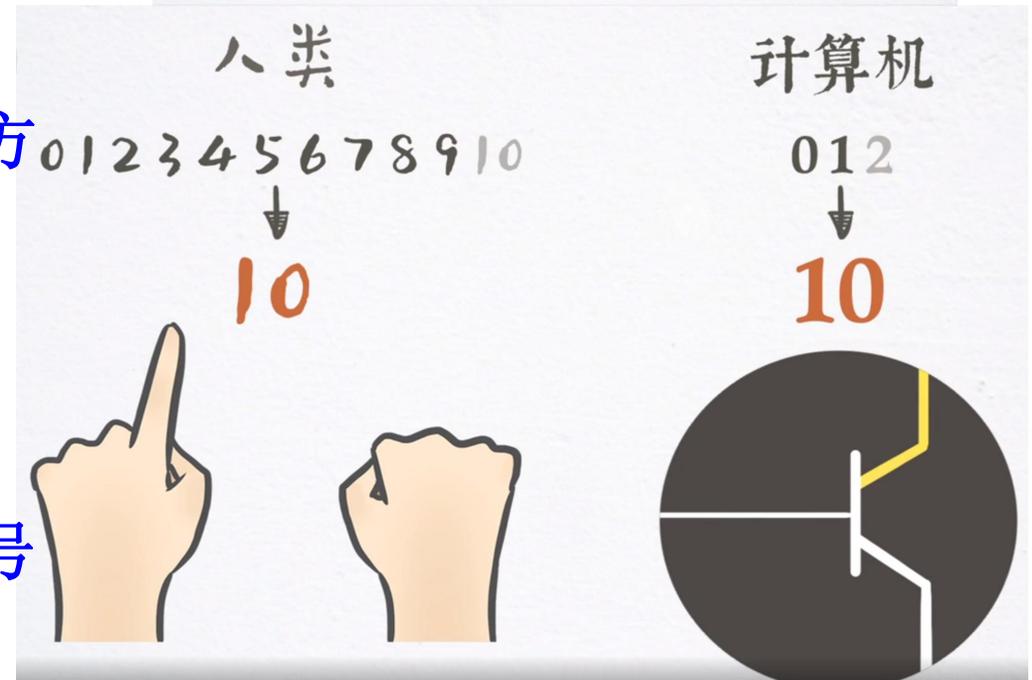
- 原码、补码、反码、移码



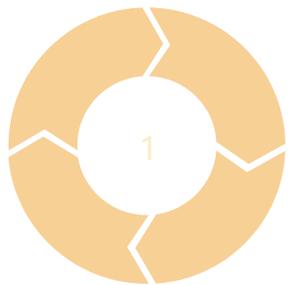
# 二进制的引入

- ◆ 计算机内部所有信息都用二进制（即：**0**和**1**）进行编码
- ◆ 用二进制编码的原因：
  - 制造二个稳定态的物理器件容易
  - 二进制编码、计数、运算规则简单
  - 正好与逻辑命题对应，便于逻辑运算，并可方便地用逻辑电路实现算术运算
- ◆ 真值和机器数
  - 机器数：用**0**和**1**编码的计算机内部的**0/1**序列
  - 真值：机器数真正的值，即：现实中带正负号的数

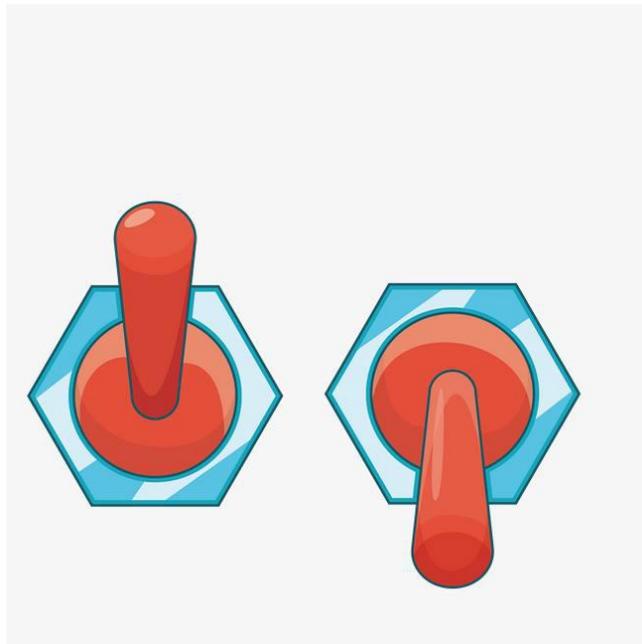
## 二进制



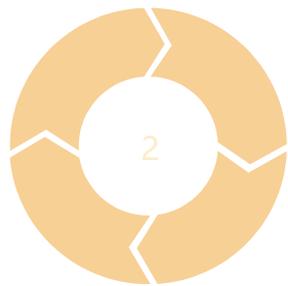
# 二进制的好处之一



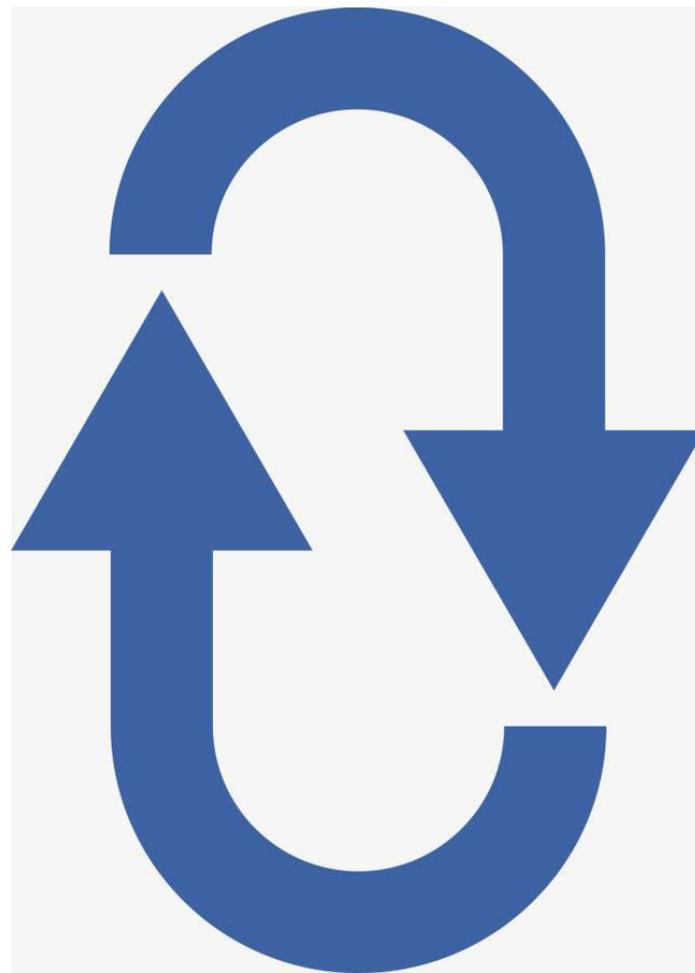
技术实现简单  
简化运算规则



## 二进制的好处之二



适合逻辑运算  
易于进行转换



## 二进制的好处之三



抗干扰能力强  
可靠性高优点





# Sign and Magnitude (原码的表示)

Decimal	Binary	Decimal	Binary
0	0000	-0	1000
1	0001	-1	1001
2	0010	-2	1010
3	0011	-3	1011
4	0100	-4	1100
5	0101	-5	1101
6	0110	-6	1110
7	0111	-7	1111

- ◆ 容易理解, 但是:
  - ✓ 0 的表示不唯一, 故不利于程序员编程
  - ✓ 加、减运算方式不统一
  - ✓ 需额外对符号位进行处理, 故不利于硬件设计
  - ✓ 特别当  $a < b$  时, 实现  $a - b$  比较困难

从 50 年代开始, 整数都采用补码来表示  
但浮点数的尾数用原码定点小数表示



# 补码特性 - 模运算 (modular运算)

重要概念：在一个模运算系统中，一个数与它除以“模”后的余数等价。

时钟是一种模12系统

假定钟表时针指向10点，要将它拨向6点，则有两种拨法：

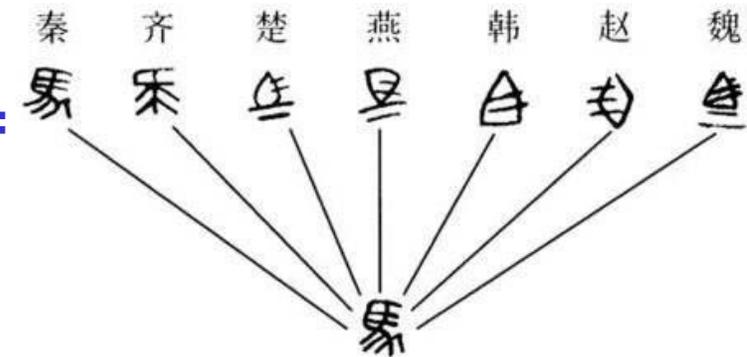
- ① 倒拨4格： $10 - 4 = 6$
  - ② 顺拨8格： $10 + 8 = 18 \equiv 6 \pmod{12}$
- 模12系统中： $10 - 4 \equiv 10 + 8 \pmod{12}$   
 $-4 \equiv 8 \pmod{12}$

则，称8是-4对模12的补码（即：-4的模12补码等于8）。

同样有  $-3 \equiv 9 \pmod{12}$   
 $-5 \equiv 7 \pmod{12}$  等

- 结论1：一个负数的补码等于模减该负数的绝对值。
- 结论2：对于某一确定的模，某数减去小于模的另一数，总可以用该数加上另一数负数的补码来代替。

补码 (modular运算)：+ 和- 的统一



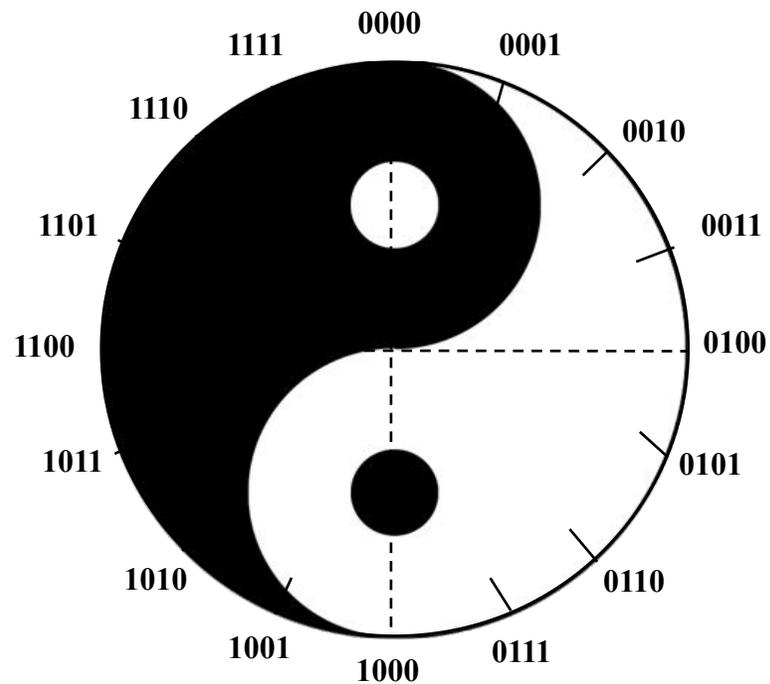
张石画社

# 编码的意义

补码的定义 假定补码有n位, 则:

$$[X]_{\text{补}} = 2^n + X \quad (-2^n \leq X < 2^n, \text{ mod } 2^n)$$

**X是真值,  $[x]_{\text{补}}$ 是机器数**



**真值和机器数的含义是什么?**



# 浮里看画



00	10	11	00	11	10	01	01	11	10
10	10	10	01	10	00	01	01	10	10
10	10	00	11	00	11	01	00	00	10
01	01	11	10	01	11	11	01	11	01
01	01	11	01	01	01	11	01	00	01
01	01	10	01	01	01	10	01	01	01
01	00	00	01	01	00	00	01	00	00
11	01	11	11	11	01	11	11	01	01
00	00	10	00	11	10	00	00	10	00
00	10	01	00	10	10	01	00	11	10
11	10	00	11	00	00	01	11	00	10
10	00	10	10	11	11	01	10	00	00
11	10	00	01	11	01	01	11	10	10

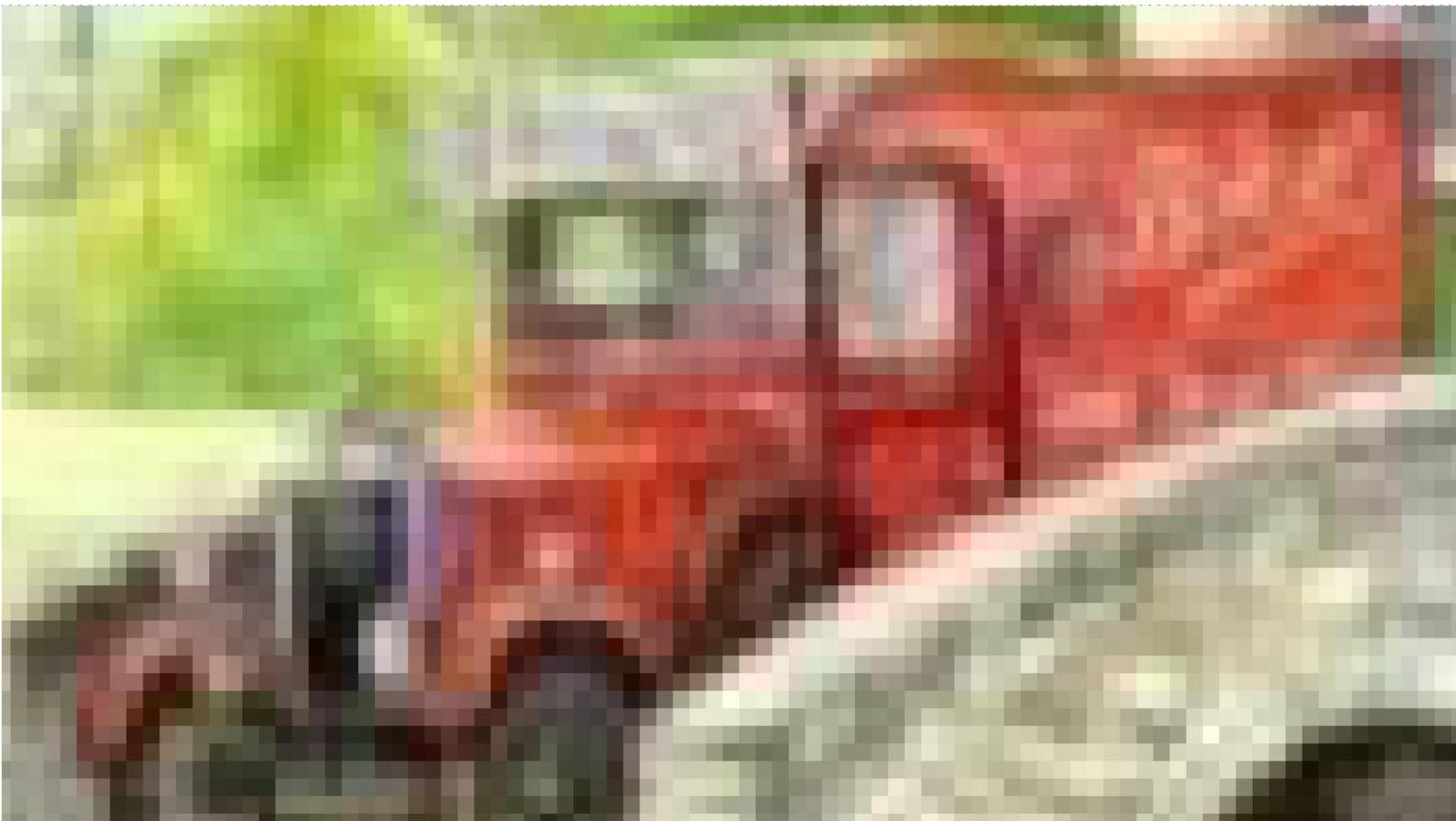
# 3





# 2





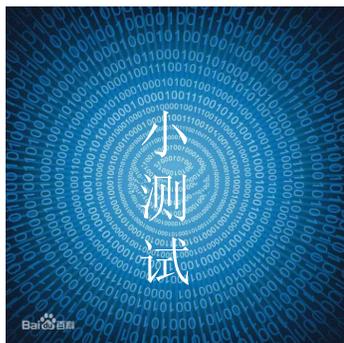


<http://www.jigsawsuk.co.uk/product/shop-by-theme/house-of-puzzles-river-cottage-ray-cresswell-250piece/>

# 计算机中的数的表达图谱

不管我们看到的任何数据最终的表现形式是什么，皆是虚像，0和1才是它本来的本来面目。

在计算机的世界里常常有11种人，1种人喜欢二进制，1种人喜欢十进制，1种人喜欢十六进制。除此之外，没有第100种人。



你懂我的意思吧？

嗯我懂你的意思

