章节导学

- **进制运算的基本知识**
  - 进制运算的基础
- **二进制数据的表示方法**
  - 有符号数与无符号数
  - 二进制的反码表示法
  - 二进制的补码表示法
  - 小数的二进制补码表示
- **二进制数据的运算**
  - 定点数与浮点数
  - 浮点数的加减法运算
  - 定点数的加减法运算
  - 浮点数的乘除法运算

进制运算的基础

- 进制概述
- 二进制运算的基础

进制运算的基础

进制概述

进制的定义

常见的进制

进制运算的基础

进制概述

- 进位制是一种记数方式，亦称进位计数法或位值计数法
- 有限种数字符号来表示无限的数值
- 使用的数字符号的数目称为这种进位制的基数或底数

n=10 [0-9] 称为十进制

进制运算的基础

进制概述

八进制

二十进制

十六进制

六十进制

二进制

@咚咚呛

进制运算的基础

进制概述

玛雅文明的玛雅数字

因努伊特的因努伊特数字

二十进制

进制运算的基础

进制概述

时间、坐标、角度等量化数据

六十进制

进制运算的基础

进制概述

[0-9]和A、B、C、D、E、F

十六进制

进制运算的基础

进制概述

- 计算机喜欢二进制，但是二进制表达太长了
- 使用大进制位可以解决这个问题
- 八进制、十六进制满足2的n次方的要求

八进制&十六进制

进制运算的基础

进制概述

1024=0b10000000000

1024=0o2000

1024=0x400

八进制&十六进制

进制运算的基础

- 进制概述
- 二进制运算的基础

进制运算的基础

二进制运算的基础

- 正整数N，基数为r

$$N = d_{n-1}d_{n-2}\cdots d_1 d_0$$

$$= d_{n-1}\mathrm{r}^{n-1} + d_{n-2}\,\mathrm{r}^{n-2} + \cdots + d_1 r + d_0$$

$$N = 1024 \qquad N = 1 * 10^3 + 2 * 10^1 + 4$$

$$N = 10000000000 \qquad N = 1 * 2^{10}$$

进制运算的基础

二进制运算的基础

$$N = d_{n-1}d_{n-2}\cdots d_1d_0$$

$$= d_{n-1}\mathrm{r}^{n-1} + d_{n-2}\,\mathrm{r}^{n-2} + \cdots + d_1r + d_0$$

$$N = (01100101) = 1 * 2^6 + 1 * 2^5 + 1 * 2^2 + 1 = 101$$

$$N = (11101101) = 1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 1 * 2^3 + 1 * 2^2 + 1 = 237$$

二进制转换十进制：按权展开法

进制运算的基础

二进制运算的基础

重复除以2	得商	取余数
101/2	50	1
50/2	25	0
25/2	12	1
12/2	6	0
6/2	3	0
3/2	1	1
1/2	0	1

$$N = (01100101) = 1 * 2^6 + 1 * 2^5 + 1 * 2^2 + 1 = 101$$

（整数）十进制转换二进制：重复相除法

进制运算的基础

二进制运算的基础

重复除以2	得商	取余数
237/2	118	1
118/2	59	0
59/2	29	1
29/2	14	1
14/2	7	0
7/2	3	1
3/2	1	1
1/2	0	1

$$N = (11101101) = 1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 1 * 2^3 + 1 * 2^2 + 1 = 237$$

进制运算的基础

二进制运算的基础

$$N = d_{n-1}d_{n-2}\cdots d_1 d_0$$

$$= d_{n-1}\mathrm{r}^{n-1} + d_{n-2}\,\mathrm{r}^{n-2} + \cdots + d_1 r + d_0$$

$$N = (0.11001) = 1 * 2^{-1} + 1 * 2^{-2} + 1 * 2^{-5} = 0.78125 = \frac{25}{32}$$

$$N = (0.01011) = 1 * 2^{-2} + 1 * 2^{-4} + 1 * 2^{-5} = 0.34375 = \frac{11}{32}$$

二进制转换十进制：按权展开法

进制运算的基础

二进制运算的基础

重复乘以2	得积	取1
25/32	50/32=1+9/16	1
9/16	18/16=1+1/8	1
1/8	1/4=0+1/4	0
1/4	1/2=0+1/2	0
1/2	1=1+0	1

$$N=(0.11001)=1*2^{-1}+1*2^{-2}+1*2^{-5}=0.78125=\frac{25}{32}$$

（小数）十进制转换二进制：重复相乘法

进制运算的基础

二进制运算的基础

重复乘以2	得积	取1
11/32	11/16=0+11/16	0
11/16	11/8=1+3/8	1
3/8	3/4=0+3/4	0
3/4	3/2=1+1/2	1
1/2	1/1=1+0	1

$$N = (0.01011) = 1 * 2^{-2} + 1 * 2^{-4} + 1 * 2^{-5} = 0.34375 = \frac{11}{32}$$

进制运算的基础

- 进制概述
- 二进制运算的基础

有符号数与无符号数

负数怎么办?

+表示正数，-表示负数

$$N = (11101101) = 1 * 2^7 + 1 * 2^6 + 1 * 2^5 + 1 * 2^3 + 1 * 2^2 + 1 = 237$$

有符号数与无符号数

+237=011101101

-237=111101101

怎么判断他是数字位还是符号位呢?

使用0表示正数，使用1表示负数

有符号数与无符号数

符号位

0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	1

1	0	0	0	0	0	0	0	1	1	1	0	1	1	0	1

+237=011101101

-237=111101101

原码表示法

有符号数与无符号数

原码表示法

- 使用0表示正数、1表示负数
- 规定符号位位于数值第一位
- 表达简单明了，是人类最容易理解的表示法

有符号数与无符号数

原码表示法

0有两种表示方法：00、10

原码进行运算非常复杂，特别是两个操作数符号不同的时候

- 判断两个操作数绝对值大小
- 使用绝对值大的数减去绝对值小的数
- 对于符号值，以绝对值大的为准

有符号数与无符号数

原码表示法

希望找到不同符号操作数更加简单的运算方法

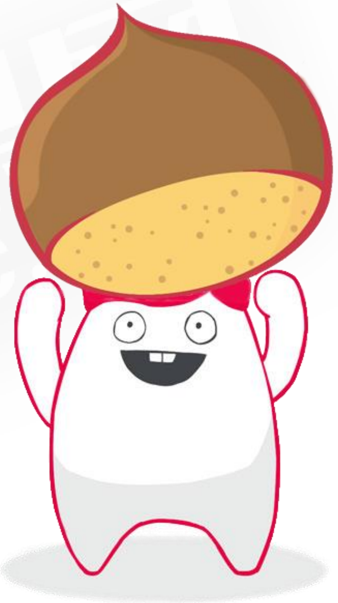
希望找到使用正数代替负数的方法

使用加法操作代替减法操作，从而消除减法

二进制的补码表示法

$$x = \begin{cases} x & 2^n > x \geq 0 \\ 2^{n+1} + x & 0 > x \geq -2^n \end{cases}$$

补码的定义



二进制的补码表示法

例子1：n=4，x=13，计算x的二进制原码和补码

原码：x=0,1101

补码：x=0,1101

$$x = \begin{cases} x & 2^n > x \geq 0 \\ 2^{n+1} + x & 0 > x \geq -2^n \end{cases}$$

二进制的补码表示法

例子2：x=-13，计算x的二进制原码和补码

原码：x=1,1101

$$x = \begin{cases} x & 2^n > x \geq 0 \\ 2^{n+1} + x & 0 > x \geq -2^n \end{cases}$$

补码：$2^{n+1} + x = 2^{4+1} - 13 = 100000 - 1101 = 10011$

补码：x=1,0011

符号位

二进制的补码表示法

例子3：x=-7，计算x的二进制原码和补码

原码：x=1,0111

$$x = \begin{cases} x & 2^n > x \geq 0 \\ 2^{n+1} + x & 0 > x \geq -2^n \end{cases}$$

补码：$2^{n+1} + x = 2^{4+1} - 7 = 100000 - 0111 = 11001$

补码：x=1,1001

符号位

二进制的补码表示法

例子4：x=-1，计算x的二进制原码和补码

原码：x=1,0001

$$x = \begin{cases} x & 2^n > x \geq 0 \\ 2^{n+1} + x & 0 > x \geq -2^n \end{cases}$$

补码：$2^{n+1} + x = 2^{4+1} - 1 = 100000 - 0001 = 11111$

补码：x=1,1111

二进制的补码表示法

$$x = \begin{cases} x & 2^n > x \geq 0 \\ 2^{n+1} + x & 0 > x \geq -2^n \end{cases}$$

希望找到使用正数代替负数的方法

使用加法操作代替减法操作，从而消除减法

在计算补码的过程中，还是使用了减法！！

二进制的补码表示法

例子3：x=-7，计算x的二进制原码和补码

原码：x=1,0111

$$x = \begin{cases} x & 2^n > x \geq 0 \\ 2^{n+1} + x & 0 > x \geq -2^n \end{cases}$$

补码：$2^{n+1} + x = 2^{4+1} - 7 = \boxed{100000 - 0111} = 11001$

补码：x=1,1001

在计算补码的过程中，还是使用了减法！！

二进制的反码表示法

引进补码的目的

- 减法运算复杂，希望找到使用正数替代负数的方法
- 使用加法代替减法操作，从而消除减法

在计算补码的过程中，还是使用了减法！！

二进制的反码表示法

例子2：x=-13，计算x的二进制原码和补码

原码：x=1,1101

$$x = \begin{cases} x & 2^n > x \geq 0 \\ 2^{n+1} + x & 0 > x \geq -2^n \end{cases}$$

补码：$2^{n+1} + x = 2^{4+1} - \boxed{13 = 100000 - 1101} = 10011$

补码：x=1,0011

二进制的反码表示法

例子3：x=-7，计算x的二进制原码和补码

原码：x=1,0111

$$x = \begin{cases} x & 2^n > x \geq 0 \\ 2^{n+1}+x & 0 > x \geq -2^n \end{cases}$$

补码：$2^{n+1}+x = 2^{4+1}-7 = \boxed{100000 - 0111} = 11001$

补码：x=1,1001

二进制的反码表示法

引进补码的目的

- 减法运算复杂，希望找到使用正数替代负数的方法
- 使用加法代替减法操作，从而消除减法

在计算补码的过程中，还是使用了减法！！

反码的目的是找出原码和补码之间的规律，消除转换过程中的减法

二进制的反码表示法

$$x = \begin{cases} x & 2^n > x \geq 0 \\ (2^{n+1} - 1) + x & 0 > x \geq -2^n \end{cases}$$

$$x = \begin{cases} x & 2^n > x \geq 0 \\ 2^{n+1} + x & 0 > x \geq -2^n \end{cases}$$

反码的定义

二进制的反码表示法

例子1：x=-13，计算x的二进制原码和反码

原码：x=1,1101

$$x = \begin{cases} x & 2^n > x \geq 0 \\ (2^{n+1}-1)+x & 0 > x \geq -2^n \end{cases}$$

反码：$(2^{n+1}-1)+x=(2^{4+1}-1)-13=011111-1101=10010$

符号位

反码：x=1,0010

二进制的反码表示法

例子2：x=-7，计算x的二进制原码和反码

原码：x=1,0111

$$x = \begin{cases} x & 2^n > x \geq 0 \\ (2^{n+1} - 1) + x & 0 > x \geq -2^n \end{cases}$$

反码：$(2^{n+1} - 1) + x = (2^{4+1} - 1) - 7 = 011111 - 0111 = \underset{\uparrow\ \text{符号位}}{1}1000$

反码：x=1,1000

二进制的反码表示法

十进制	原码	补码	反码
13	0,1101	0,1101	0,1101
-13	1,1101	1,0011	1,0010
-7	1,0111	1,1001	1,1000
-1	1,0001	1,1111	1,1110

负数的反码等于原码除符号位外按位取反

负数的补码等于反码+1

二进制的反码表示法

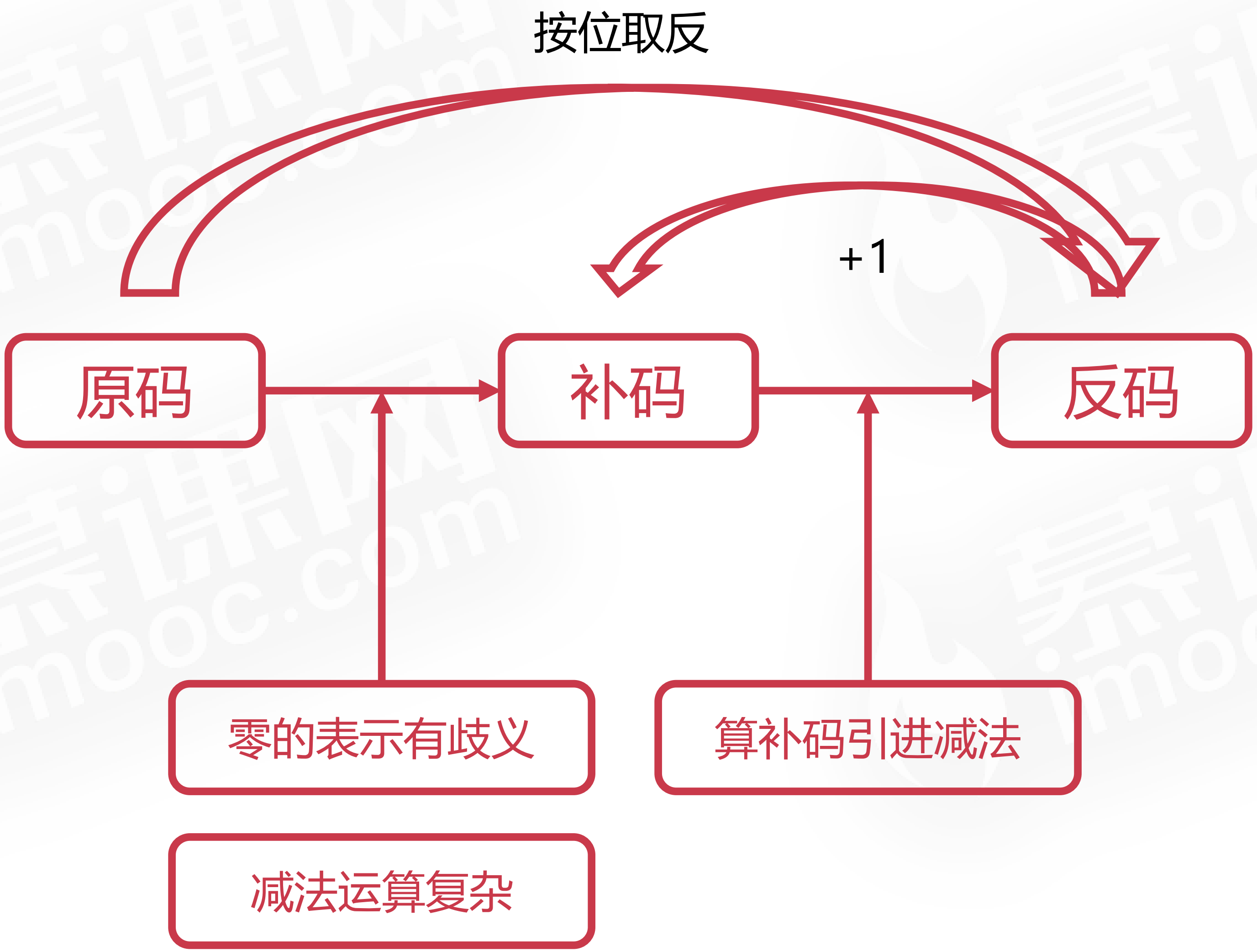
例子3：x=-7，计算x的二进制原码和反码和补码

原码：x=1,0111　　反码：x=1,1000　　补码：x=1,1001

例子4：x=-9，计算x的二进制原码和反码和补码

原码：x=1,1001　　反码：x=1,0110　　补码：x=1,0111

二进制的反码表示法



小数的补码

$$x = \begin{cases} x & 2^n > x \geq 0 \\ 2^{n+1} + x & 0 > x \geq -2^n \end{cases}$$

二进制整数的补码

$$x = \begin{cases} x & 1 > x \geq 0 \\ 2 + x & 0 > x \geq -1 \ (mod\ 2) \end{cases}$$

二进制小数的补码

小数的补码

十进制	原码	补码	反码
13	0,1101	0,1110	0,1101
-13	1,1101	1,0011	1,0010
-7	1,0111	1,1001	1,1000
-1	1,0001	1,1111	1,1110

负数的反码等于原码除符号位外按位取反

负数的补码等于反码+1

小数的补码

例子1：$x=\frac{9}{16}$，计算x的二进制原码和反码和补码

原码：x=0,0.1001　　反码：x= 0,0.1001　　补码：x= 0,0.1001

例子2：$x=-\frac{11}{32}$，计算x的二进制原码和反码和补码

原码：x=1,0.01011　　反码：x=1,1.10100　　补码：x=1,1.10101