

第二章 计算机的运算方法

第二章 计算机的运算方法

- 计算机中数的表示
- 定点运算
- 浮点运算

第二章 计算机的运算方法

- 计算机中数的表示
 - **无符号数和有符号数**
 - 定点表示和浮点表示
 - IEEE754标准
 - 算数移位与逻辑移位

(回顾) 十进制整数转换成二进制整数的经典方法

十进制整数转换为二进制整数十进制整数转换为二进制整数采用“**除 2 取余，逆序排列**”法。具体做法是：用 2 整除十进制整数，可以得到一个商和余数；再用 2 去除商，又会得到一个商和余数，如此进行，直到商为小于 1 时为止，然后把先得到的余数作为二进制数的低位有效位，后得到的余数作为二进制数的高位有效位，依次排列起来。

以下是两个例题：

The image shows two handwritten division diagrams for base conversion.

Left Diagram: Dividing 179 by 2. The quotient is 89, and the remainder is 1. The quotient 89 is then divided by 2, resulting in 44 with a remainder of 1. This process continues until a quotient of 0 is reached. The remainders are listed vertically: 1, 1, 0, 0, 1, 1, 0, 1. A red arrow points from the rightmost remainder (1) up through the sequence of remainders to the left, indicating the order of extraction.

$$\begin{array}{r} 2 \overline{)179} \\ 2 \overline{)189} \\ 2 \overline{)44} \\ 2 \overline{)22} \\ 2 \overline{)11} \\ 2 \overline{)5} \\ 2 \overline{)2} \\ 2 \overline{)1} \\ 0 \end{array}$$

Right Diagram: Dividing 206 by 2. The quotient is 103, and the remainder is 0. The quotient 103 is then divided by 2, resulting in 51 with a remainder of 1. This process continues until a quotient of 0 is reached. The remainders are listed vertically: 0, 1, 1, 1, 0, 0, 1. A red arrow points from the rightmost remainder (1) up through the sequence of remainders to the left, indicating the order of extraction.

$$\begin{array}{r} 2 \overline{)206} \\ 2 \overline{)103} \\ 2 \overline{)51} \\ 2 \overline{)25} \\ 2 \overline{)12} \\ 2 \overline{)6} \\ 2 \overline{)3} \\ 2 \overline{)1} \\ 0 \end{array}$$

$(179)_{10} = (10110011)_2$

$(206)_{10} = (11001110)_2$

无符号数

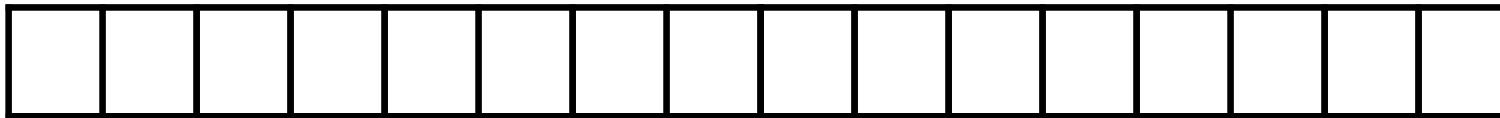
- 寄存器的位数反映无符号数的表示范围。

8 位



0 ~ 255

16 位



0 ~ 65535

求补码的快捷方式

二进制数表示补码的方法：

- 1) 正数的补码，符号位为0，然后是正常表示的二进制
- 2) 负数的补码，采用扫描法：符号位为1，然后自右向左从最先遇到的1左边开始，每位取反。

注意：扫描法只针对负数，并且最小的负数不能用扫描法

6

例题：当 $x = +1010$ 时，则 $[x]_{\text{补}} = \mathbf{0}1010$

当 $x = -1010$ 时，则 $[x]_{\text{补}} = \mathbf{1}0110$

当 $[x]_{\text{补}} = 11110$ 时，则 $x = -0010$

对于一个w位的补码，其表示的范围是 $-2^{(w-1)} \sim 2^{(w-1)-1}$