计算机组成原理

数字

|  |  |
| --- | --- |
| 编 制 人： | Lizy |
| 发布日期： | 2023.3.15 |
| 页 数： |  |

更改记录

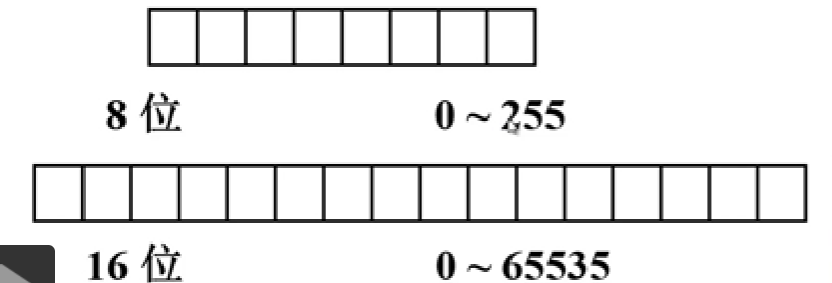
|  |  |
| --- | --- |
| **发布日期** | **更改描述** |
| 20230303 | 第一版发布 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 计算机的运算方法

## 无符号数和有符号数

### 无符号数

只有数值。保存在寄存器中，寄存器的位数反映了无符号数的表示范围。



### 有符号数

有正负号的数。包括符号和数值2部分。

#### 机器数与真值

1. 定义：

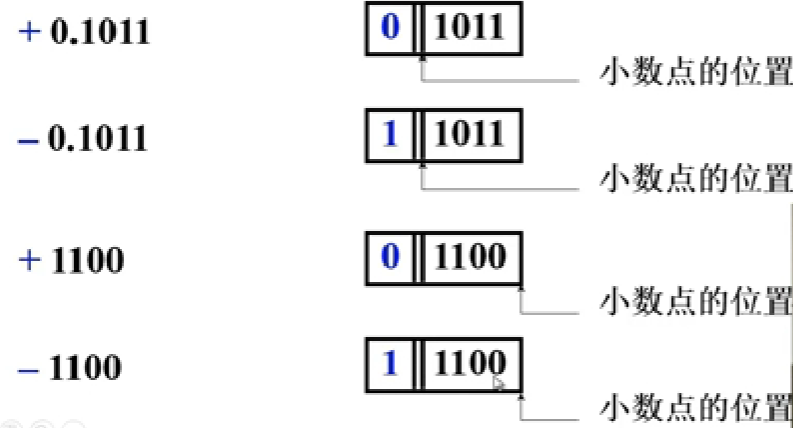
保存在计算机中数叫做**机器数**；

实际的值为**真值**；

1. 符号位：

0：正值；1：负值；

1. 小数点：以约定方式给出



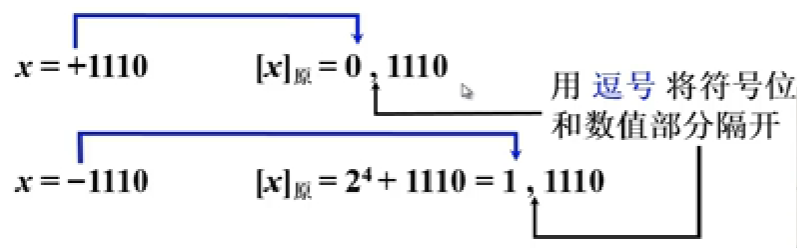
1. 定点计算机：小数点的位置是**固定**的计算机；
2. 定点计算机分为2类：

* 小数定点器：小数点在符号位后面；
* 整数定点器：小数点在整数后面；

#### 原码表示法

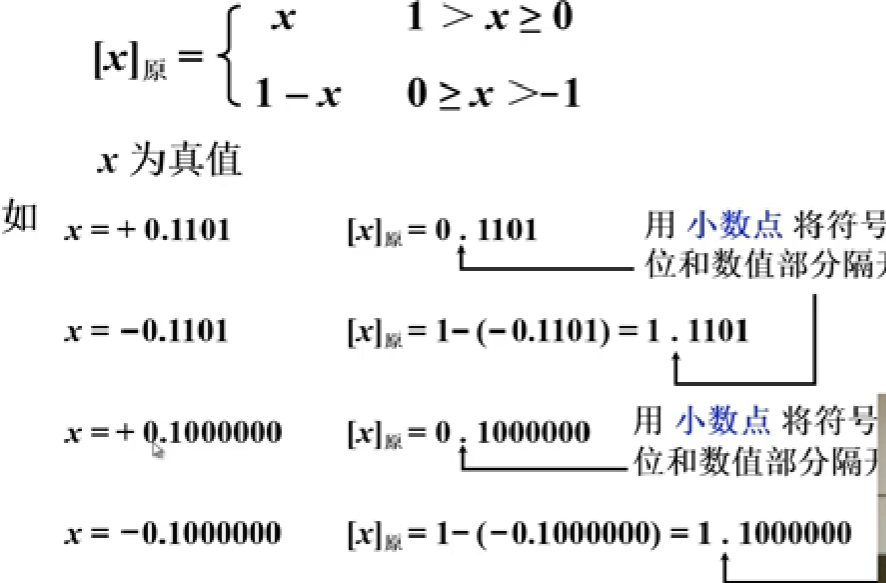
1. 定义：符号位数字化；**带符号的绝对值**；

* 整数：用“**，**”将符号位和数值隔开



* 小数：用“**.**”将符号位和数值隔开

真值范围:**（0.a ~ (-0.a)）**

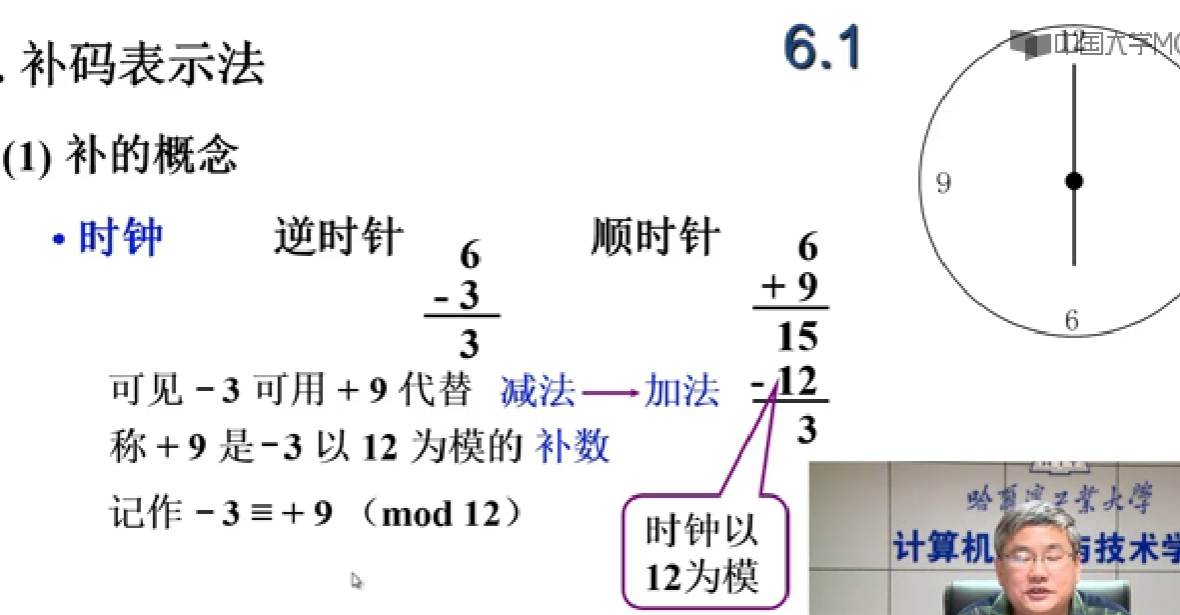


1. 特点：简单、直观，但是做加法时也要用减法操作；

#### 补码表示法

1. 补的概念：

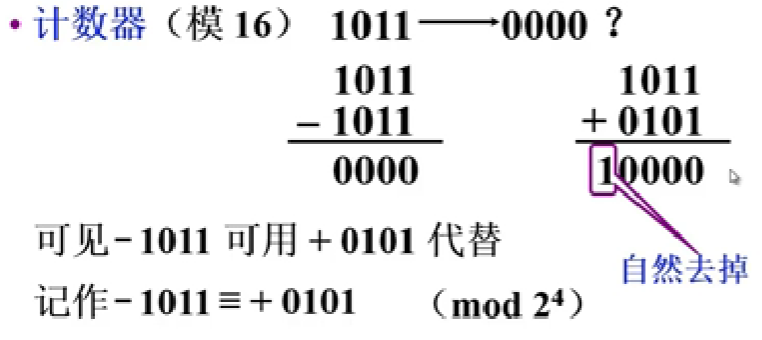
+9是-3以12为模的补数；



1. 结论：

一个负数加上“模”即得该负数的补数；

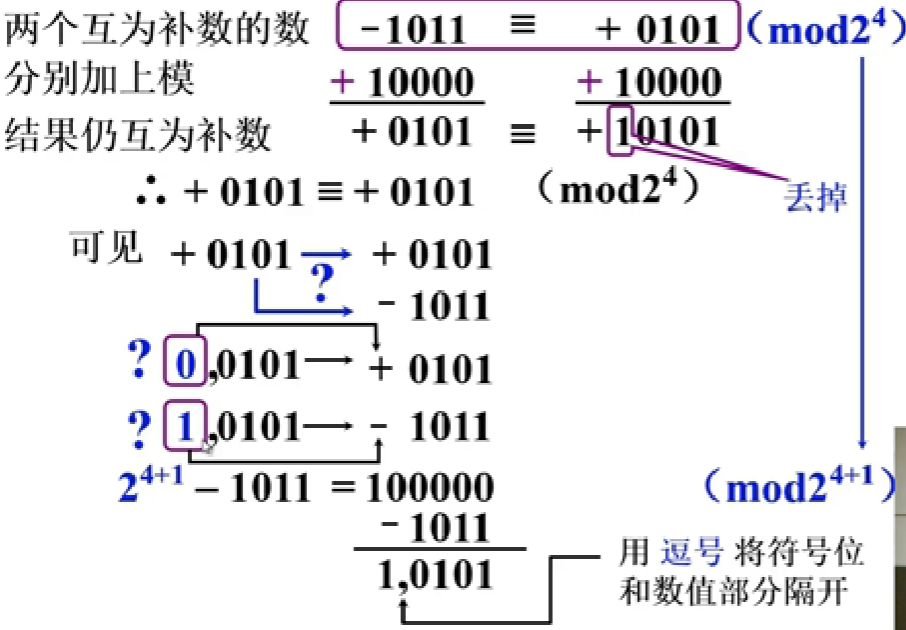
一个正数和一个负数互为补数时，他们的绝对值之和即为“模数”；



1011 + 0101 = 10000；

寄存器为4位，则其模为24（16即10000）；

1. 正数的补数即为其本身：

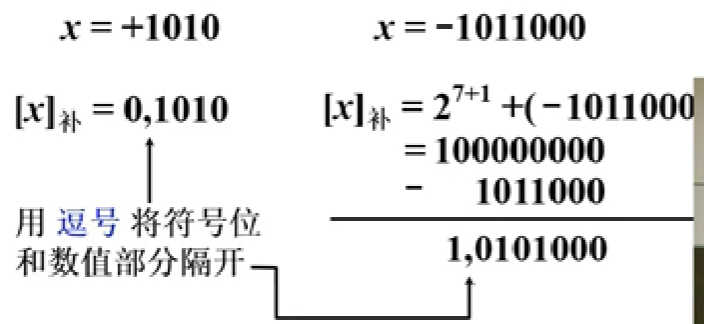


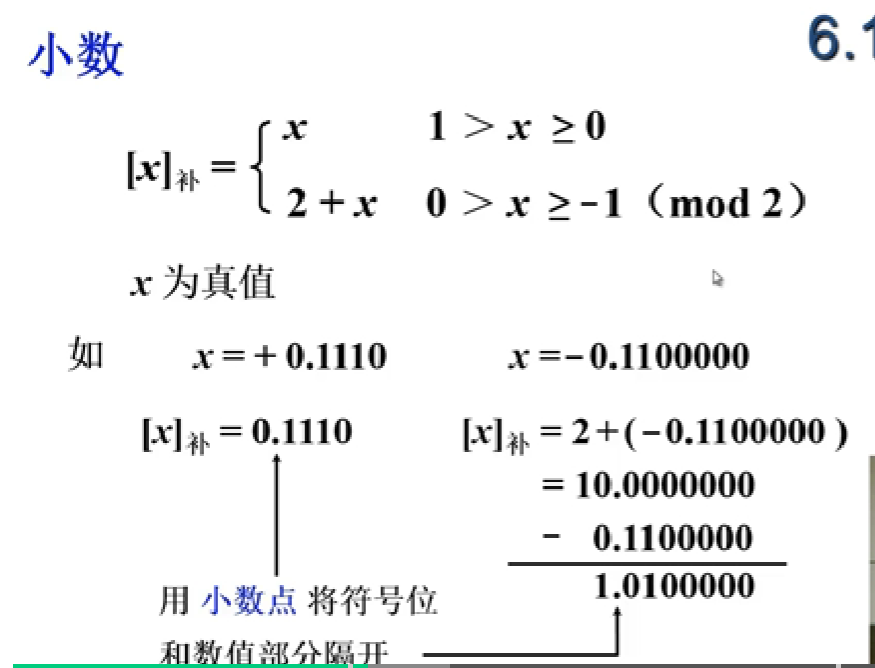
1. 补码定义：

正数的补码就是其本身；

负数的补码等于：符号位 |（2n+1 + 负数）；其中n为数值位数

这里“模”是**n+1**次方





小数的“模”是**2**；

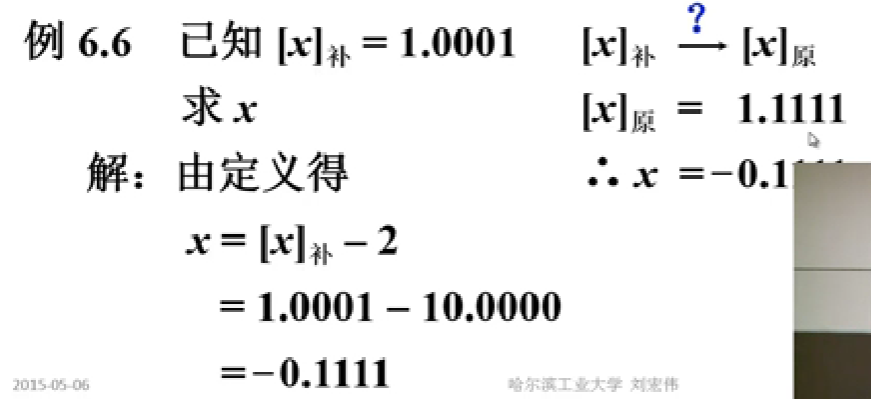
负数的补码：**2 + 负数；**

1. 求补码的快捷方式

**负数**的补码：原码符号位不变，数值位**取反+1**；小数也同样适用；

设x = -1010

* 数值取反+1 ： 0101 + 1 = 0110
* 添加符号位：10110



1. 真值为**负数**时：**补码 --> 原码：**符号位不变，数值取反 + 1

#### 反码表示法

## 数的定点