**【计导非课系列】 第五节 二进制 进制计算 编码**

对于计算机来说，数字只有两个——0和1。

数据对于计算机来说是相当重要的，而电路的通断两种状态决定了计算机只能通过1和0来进行一切事情的处理。所以，我们见到的计算机的一切都是通过二进制才能呈现出来的。这次就会着重介绍一下进制的计算和编码。

**博文目录**

* 什么是数据
* 数的进制
* 二进制
* 进制的转换
* 逻辑运算
* 计算机对数据的处理
* 常用编码
* 数的表示
* 符号数的机器码表示

**before we start…**

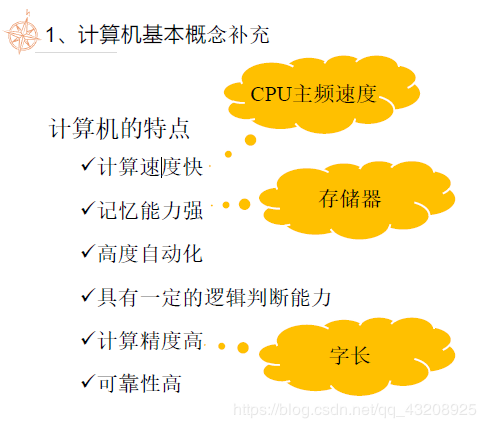
**计算机为什么采用二进制？**

计算机中，数是用物理器件的状态表示的，二进制只有两种状态，0和1，容易用电路表示。  
二进制规则简单，容易用数字逻辑电路实现。  
二进制还可以表示逻辑值，进行逻辑计算。  
To say it simply, 这本来就是电路决定的。而这样一决定之后，正好带来了许多好处：比如0代表假的，没发生过的；比如所有数字只有两种状态，便于管理，写起来程序还方便……总之，二进制就是强！

**自动计算要解决的问题**

数据的存储、表示、运算，以及自动执行的计算模型

**计算机主要技术指标**

* **字长**：计算机一次存取传递或加工的数据长度。也就是系统是多少位的。
* **主存容量**：内存（主存储器）所能存储的二进制容量，主存越大，交换越少，处理速度越快。
* **计算机指令执行速度**：（MIPS），每秒钟执行加减法有几百万次，可以通过时钟频率间接估计。
* **时钟周期（频率、主频） CPU主频速度**：CPU在单位时间内发出的脉冲数。
* **数据输入/输出最高速率**：计算机的数据吞吐量。  
  例题：下面的特点分别对应哪些技术指标？  
  

**数据**

**什么是数据**

数据：反应客观世界事物属性的原始记录。

**需要对数据做什么**

* 存储数据：数据怎样做才能被记到计算机上？用0和1。
* 组织数据：怎样存储这些0和1对我最有用？数据结构。

**存储容量单位换算**

在计算机里面，“千”不是所谓的103，而是210！也就是1024。  
**存储容量**指的是存储器有多少个**存储单元**。最基本的存储单元是位，bit，可以存放1个0或者1。  
关系如下：  
1B=1字节=8位  
1KB=1024B=210B  
1MB=1024KB=220B  
1GB=1024MB=230B  
1TB=1024GB=240B  
1PB=1024TB=250B  
1EB=1024PB=260B  
**硬件厂商和软件大小的“矛盾”**  
硬件厂商认为，Kilo当作1000……  
软件认为，Kilo当作1024……

**数的进制**

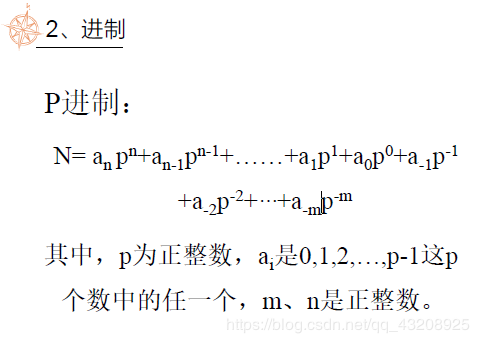
**什么是进制？**

就是，逢几进一！

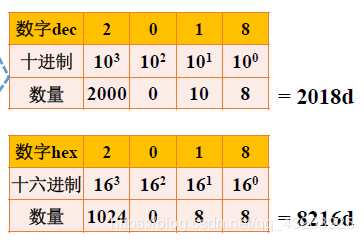
**所有进制都是这样规定的：**

基数+位权  
基数：几进制基数就是几。二进制就是2，十进制就是10！  
而且，几进制代表能出现的数字，就在0到这个数减一。  
位权：数制中某一位上1所代表的多大。所处位置的“价值”。  
啥意思？就是说这一位有多大。比如，百位，位权就是100呗~  
对于十进制243，2位权100，4位权10，3位权1。

**所有进制都是这样计算的：**



**进制的作用**

  
规定了一个数位上最多有多少种状态。

**二进制数**

**符号0是0，符号1是1，这就是二进制**

**二进制的存储问题 位 位模式**

**信息表示的最小单位：位 bit**

有两种状态的物质就构成一个位，用开关、磁极、电容器、触发器就可以物质实现。

**位模式——是一个集合：8个位构成一个字节**

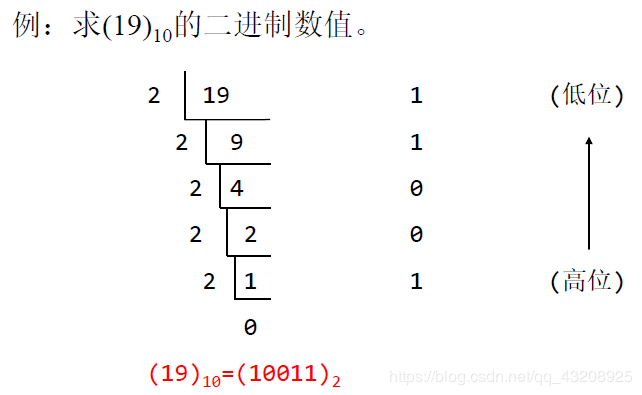
8个位构成一个字节，这就是一种位模式。  
**存储单元**=字节

**进制的转换**

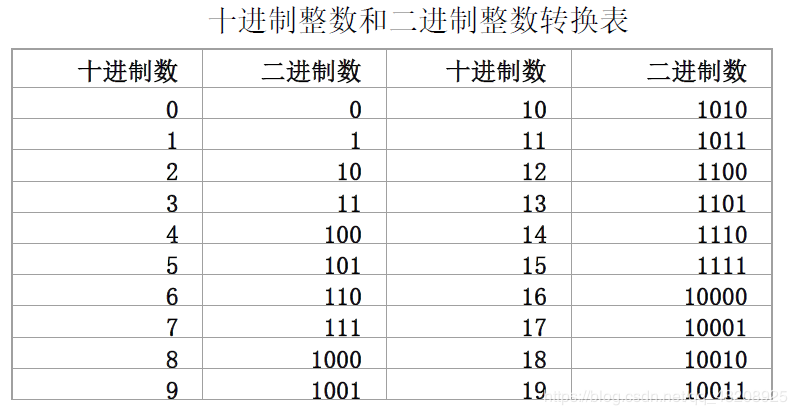
**二进制数转换为十进制数**

直接用位权乘以数字相加，第一位是0次幂，小数点往左幂次递增，往右幂次递减。  
求(1001.101)(2)的十进制数值。  
=1*20+1*23+1*2-1+1*2-3  
=8+1+0.5+0.125=9.625

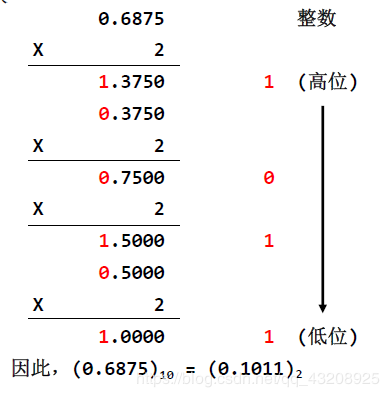
**十进制数转换为二进制数：除二取余法**

来一道题：  


**也可以记住一些常用的十进制和二进制整数转换表**

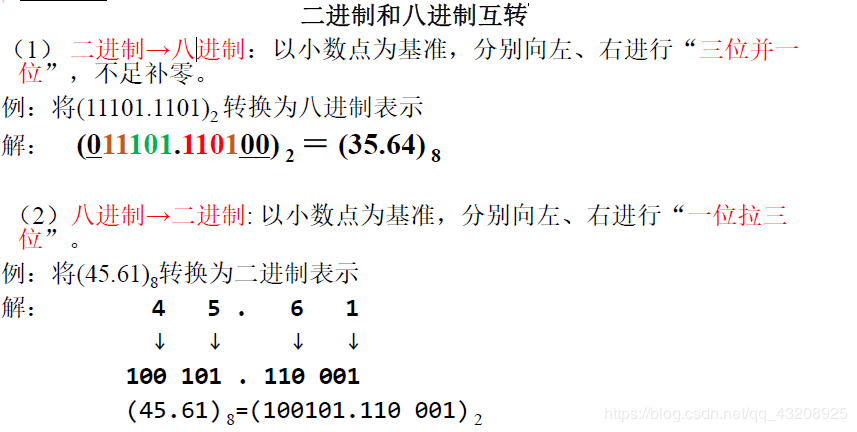
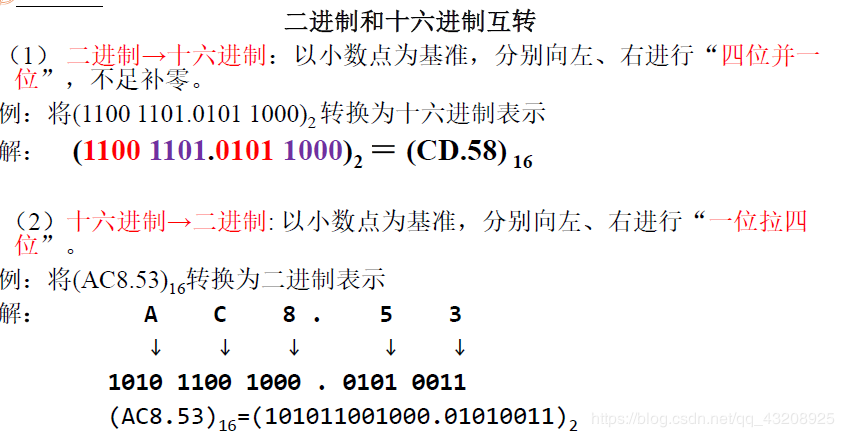
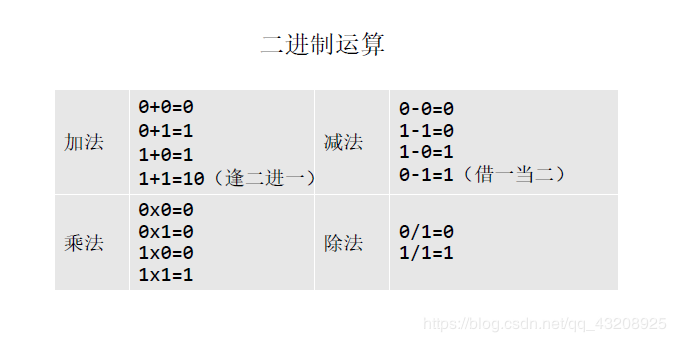


**十进制小数转化为二进制小数？乘二取余法**

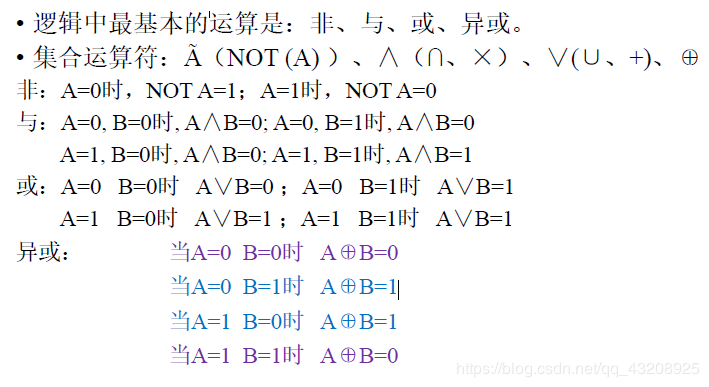
首先不断地对前次得到的积的小数部分乘2，并列出该次得到的整数数值，然后按从前向后的次序排列。该方法简称**乘2取整法**。  
  
可能会一直很多步，出不来结果，甚至循环！  
此时要由转换精度决定。

**数字游戏——二八十六之间的转换**

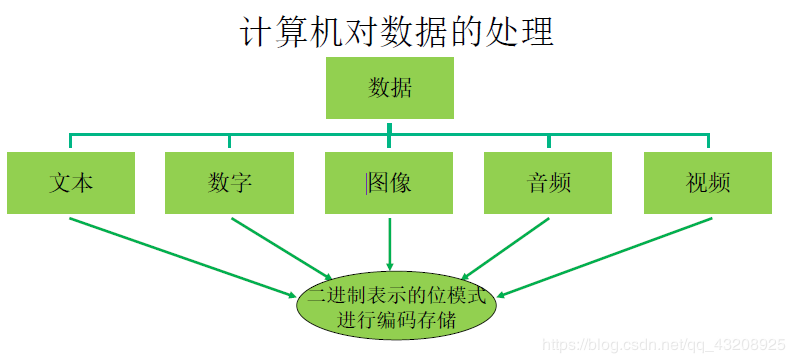
**二进制是最基本的，八进制是三位二进制，十六进制是四位二进制**

直接上例题，一看就会！  
  
  
  
加法减法乘法除法列竖式和十进制完全一样

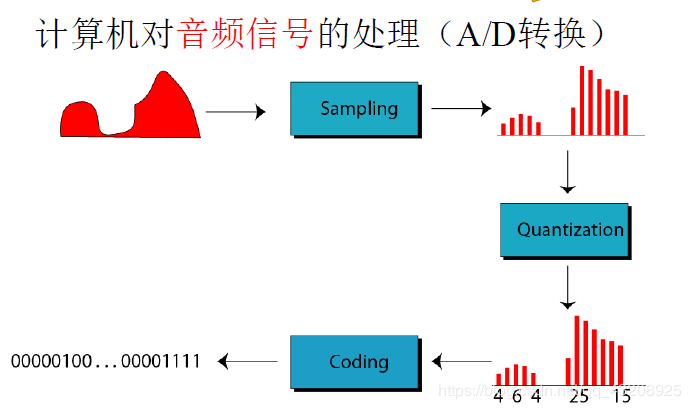
**逻辑运算**

非、与、或、异或  
  
异或XOR：就是不一样的时候，结果才是真。  
或非NOR：一样的时候，结果才是真。

**计算机对数据的处理**



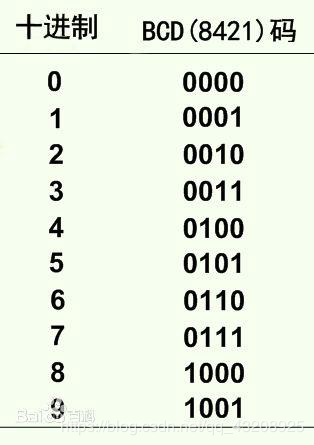
**典例：AD转换 Analog/Digital**

  
Sampling采样 Quantization数字化 Coding编码  
对图像信号，可以使用矩阵，1黑0白；也可以使用RGB。  
对视频信号，使用帧frame。

最终，都是要回归到位模式存储的。

**常用编码**

**BCD码 Binary-Coded Decimal 二-十进制编码**

使用4位表示一个十进制数  
8421BCD编码 就是位权分别为8、4、2、1  
表示0~9绰绰有余了。  


**压缩BCD码：一个字节分成两半，每半表示一个十进制数**

**非压缩BCD码：不拆开，一个字节表示一个十进制数**

比如，89表示为： (00001000 00001001)BCD

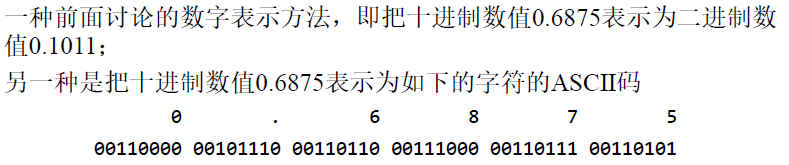
**ASCII码 美国信息交换标准代码**

The American Standard Code for Information Interchange

**使用7位或8位表示特定的128或256种字符**

**编码方案：高三位和低四位**

**计算机对数字的位模式存储处理：二进制 VS ASCII**

  
前一种：**二进制文件**  
后一种：**ASCII码文件**或文本文件

辨析：文本文件指的是？ASCII码文件！  
ASCII码文件占用空间更大

十六进制表示？只不过是把二进制多转换了一下。

**汉字常见编码体系**

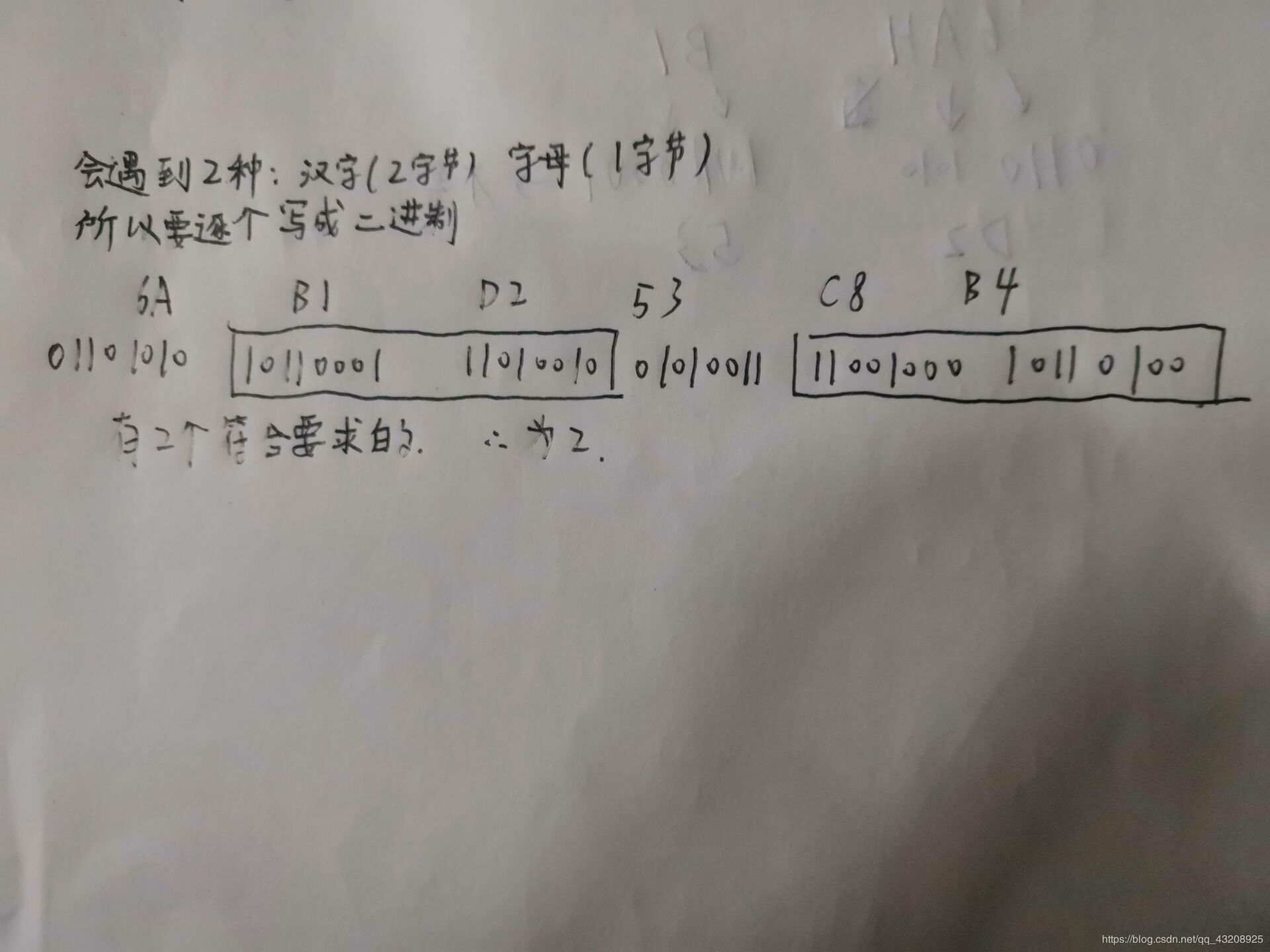
**汉字输入码 （外码）： 输入设备产生的，你怎么把这个汉字输进去**

比如，区位码，国标码、拼音码、新全拼、新双拼、五笔字型码、简码、表形码、自然码、智能ABC汉字输入码。

**汉字机内码：用于计算机内部存储和处理的汉字编码**

用于**计算机内部** **存储**和**处理**的汉字编码，通常由国标码的两个字节（最高位置“1”形成）  
**下面是我的一点理解**  
要点：什么叫汉字机内码？就是机器内部用的编码！不管你从鼠标或者别的什么用的什么方法输入的这些汉字，在我计算机里面，我就用这样的编码来记下来你到底输入的是什么玩意，在需要处理的时候，我也用我的这个编码来管理。  
机内，计算机内部。干什么用？第一，从外部读入之后存下来：存储。第二，需要使用这些汉字，处理。  
最关键的就是，计算机内部，存储，处理。  
**外码和内码关联**：外码多种多样，内码往往统一。  
**汉字机内码每个字节第一位都是1**：为了与ASCII码区别开来。所以，汉字机内码会是这个样子：  
1。。。。。。。1。。。。。。。  
**通过这一点，来一个例题**：

在内存中，若汉字以GB2312的内码表示，已知存储了6个字节的字符串。其十六进制内容依次为：6AH、B1H、D2H、53H、C8H、B4H，这个字符串中有\_\_\_\_\_\_\_个汉字。

解题思路：汉字和ASCII怎么区分开的？**每个字节第一位都是1**的才是汉字。所以，这就很简单了。  


**汉字字形码：用于打印、输出，确定汉字字形点阵**

确定一个汉字字形点阵的编码，用于汉字显示和打印输出。  
保留在存储介质中的全部汉字字形码称为**字库**。

用16×16点阵存储2个汉字的字形码，需要用多少个字节？

How to? 16×16÷8×2=64

**数据校验码**

奇偶校验码 1的个数为奇数  
海明校验码

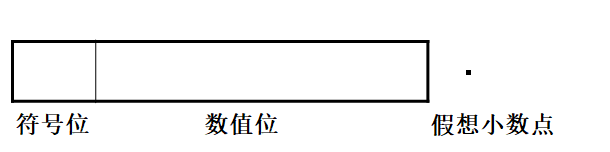
**数的表示**

计算机存储数据分为定点数和浮点数两种方式。

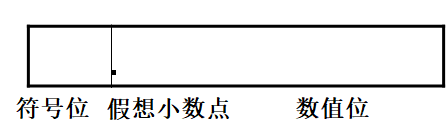
**定点数**

小数点位置固定不变，计算机设计者在机器的结构中指定一个不变的位置。  
常用的定点数：定点整数、定点小数

**定点整数**

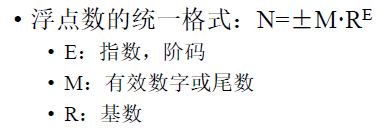
小数点固定在最低数值位之后，有一个符号位。  
如果参加运算的是小数，需要乘以一个比例因子放大为整数，然后送入计算机。  


**定点小数**

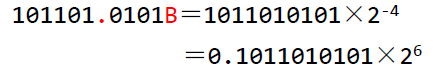
表示的都是小数，小数点在最左边符号位和数值位之间，只能保存不到1的数。  
如果参加运算的数是大于等于1的数，在送入计算机以前，除以一个比例因子，将其缩小为小于1的数。  
  
优点：计算简单方便；缺点：但需要对参加运算的数进行比例因子的计算，增加了额外的计算量。

**浮点数**

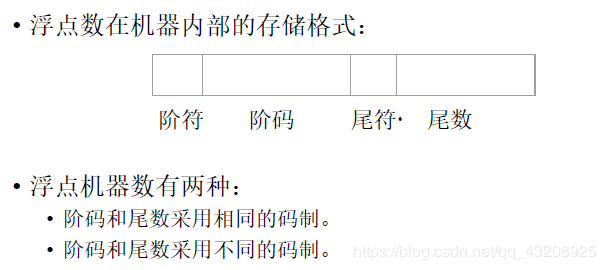
**浮点数的统一格式**

  
我的理解：  
可以看出，分为4部分。  
一是符号，整数还是负数。  
二是有效数字，也叫尾数，也就是一长串小数。  
三是基数，也就是用的几进制。  
四是阶码，也就是小数点偏离了几位，才让有效数字变得很中规中矩。

**规格化浮点数**

二进制，  
M是纯小数，且小数点后面是1。  
阶码E是正或负整数  
举例：  
  
这两个都是规格化浮点数。前面是定点数。

**浮点机器数**



**计算机中数的表示**

二进制的四则运算，乘法和除法可以通过加减法实现，因此可以没有乘除法装置。

**符号数的机器码表示**

**机器数**：一个有符号的数在机器中的二进制表示形式。最高位是符号数，1代表负数，0代表正数。  
**真值**：将带符号位的机器数转化为真正数值的形式，就是机器数的真值。  
比如：机器数：10000011  
真值：-0000011  
**无符号数**：整个机器字长全部二进制位均表示数值位，相当于数的绝对值。  
符号位只是第一位，其余位都是数值位。

**三种表示法**

* 原码表示法
* 反码表示法
* 补码表示法

**前提：正数原码、反码、补码都是一样的**

**原码表示法：符号+真值的绝对值**

如，5这个数：  
+5：0 0000101  
-5： 1 0000101

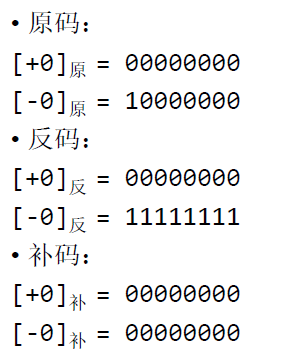
**反码表示法：负数符号位不动，其余按位取反**

如：5这个数  
反码  
+5：0 0000101  
-5：1 1111010

**补码表示法：负数补码加一**

如：5这个数：  
补码  
+5：0 0000101  
-5： 1 1111011

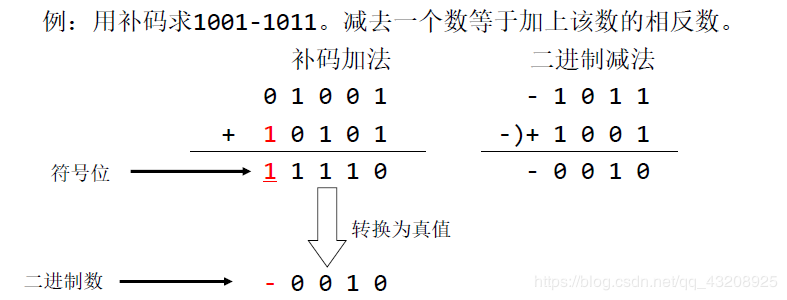
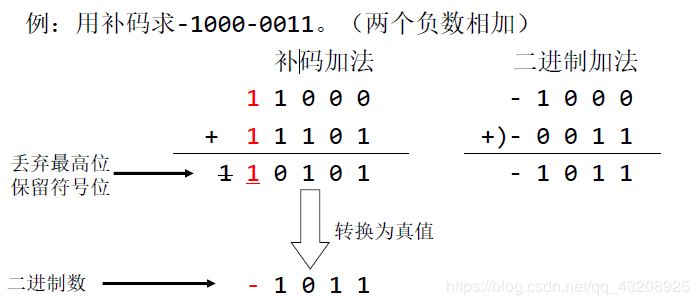
**不得不说的0**

负0的反码全部是1，补码全部是0。  


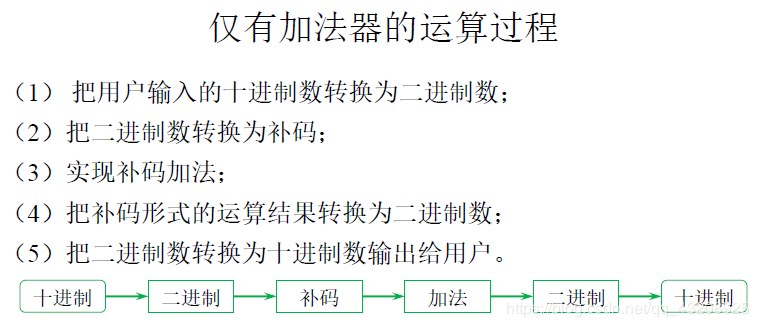
**举例理解**

求-1011的补码，并放入8位的存储单元中。  
存储单元：字节  
原码 1000 1011  
反码 1111 0100  
补码 1111 0101

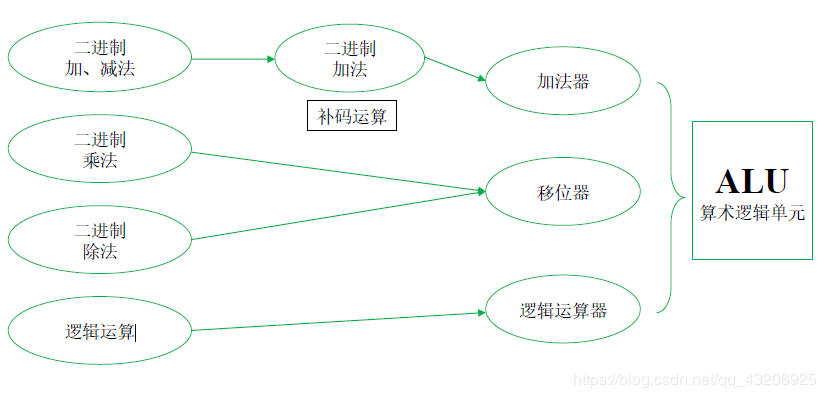
**划重点：补码计算原数——补码当原码，再次求补码**

已知1111 0101是一个补码，它是负数，反码就是1111 0100，原码就是1000 1011。  
或者对1111 0101再求补码：  
原码 1111 0101  
反码 1000 1010  
补码 1000 1011  
两种方法都可以求出来。  
  
  
丢弃最高位，保留符号位，这里涉及到了溢出的问题。  
机器负数计算方法：补码连同符号位一起取反，末尾加一。  
比如：  
0000 1011的相反数：1000 1011  
它的补码：1111 0101  
全部取反：0000 1010  
末尾加一：0000 1011  
这样得到了正数。  
原理：先求原码，也就是求补码，包括除了符号位以外其他位置按位取反，和末尾加一。第二步是符号位取反，两步合起来就是这样了。  
我觉得不用去记，知道原理就可以游刃有余。

**补码特点**

* 0的机器码表示唯一
* 二进制加减法都变成了补码加法
* 符号位直接参加运算  
  这样，减法器也可以不要了。  
  

**怎样实现的只有一个加法器，实现所有功能？**

  
加减法通过直接补码加法，乘除法需要进行移位，移位器实现乘除法。逻辑运算需要逻辑运算器。

**表示范围**

假设有n位。  
  
原码：n-1位，每一位都可以表示2n-1个数字，再加上+0和-0，所以一共有-(2n-1-1)~+(2n-1-1)  
反码也是这么多位，因为原码反码互相转换很方便。  
补码不太一样，没有+0和-0的区别，显然多了一个。10000000，就是它，-128。既当成符号位又当成数值位。

**总结**

二进制是计算机采用的基本方式，在0和1构成8字节位模式中实现所有计算机要做的事情。所有！而原码反码补码等二进制的实现形式，汉字机内码BCD码等码制，这是计算机能够实现如此复杂功能的一个重要方面。了解二进制，才能真正接近计算机。