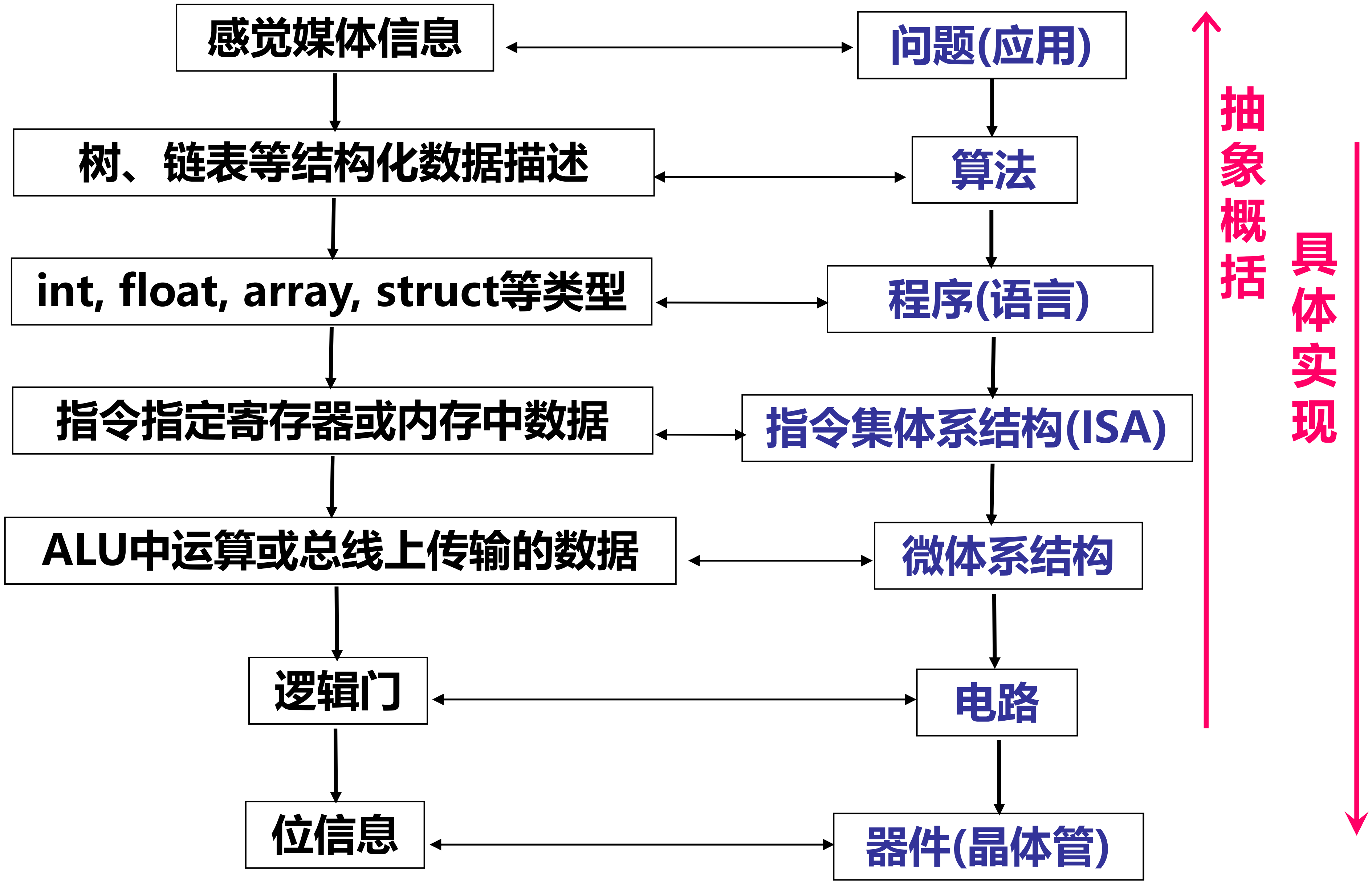
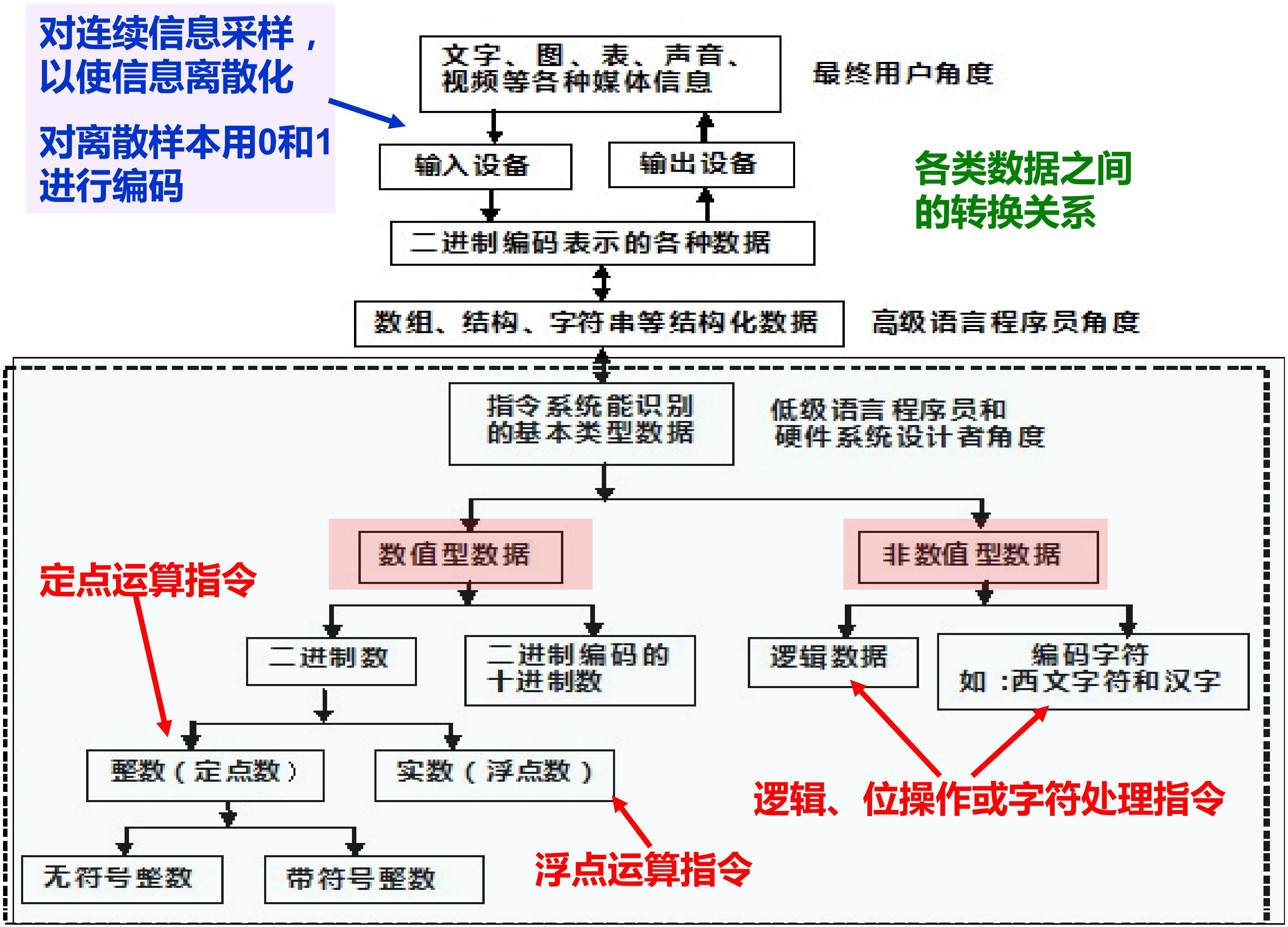
数制与编码

南京大学
计算机科学与技术系
袁春风
email: cfyuan@nju.edu.cn
2015.6

“转换”的概念在数据表示中的反映





对连续信息采样，以使信息离散化
对离散样本用0和1进行编码
文字、图、表、声音、视频等各种媒体信息
最终用户角度
输入设备
输出设备
各类数据之间的转换关系
二进制编码表示的各种数据
数组、结构、字符串等结构化数据
高级语言程序员角度
指令系统能识别的基本类型数据
低级语言程序员和硬件系统设计者角度
数值型数据
非数值型数据
定点运算指令
二进制数
二进制编码的十进制数
逻辑数据
编码字符 如 :西文字符和汉字
整数（定点数）
实数（浮点数）
逻辑、位操作或字符处理指令
无符号整数
带符号整数
浮点运算指令

信息的二进制编码

- 机器级数据分两大类
  - 数值数据：无符号整数、带符号整数、浮点数（实数）
  - 非数值数据：逻辑数（包括位串）、西文字符和汉字
- 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(电位高/低，脉冲有/无，正/负极)
  - 二进制编码、计数、运算规则简单
  - 正好与逻辑命题真/假对应，便于逻辑运算
  - 可方便地用逻辑电路实现算术运算
- 真值和机器数（非常重要的概念！）
  - 机器数：用0和1编码的计算机内部的0/1序列
  - 真值：真正的值，即：现实中带正负号的数

例：unsigned short型变量x的真值是127，其机器数是多少？

$127=2^7-1$，其机器数为0000 0000 0111 1111

数值数据的表示

- **数值数据表示的三要素**
  - 进位计数制
  - 定、浮点表示
  - 如何用二进制编码

  即：要确定一个数值数据的值必须先确定这三个要素。

  例如，20137564的值是多少？　　答案是：不知道！
- **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 （反码很少用）

十进制（Decimal）计数制

- **十进制数**，每个数位可用**十个不同符号**0,1,2,…,9来表示，每个符号处在十进制数中不同位置时，所代表的数值不一样。

  例如，2585.62代表的值是：

  $$2585.62 = 2\times10^3+5\times10^2+8\times10^1+5\times10^0+6\times10^{-1}+2\times10^{-2}$$

- 一般地，任意一个十进制数

  $D=d_n d_{n-1} \ldots d_1 d_0 . d_{-1} d_{-2} \ldots d_{-m}$ （m,n为正整数)

- 其值可表示为如下形式：

  $$V(D) = d_n\times10^n + d_{n-1}\times10^{n-1} + \ldots+ d_1\times10^1 + d_0 \times10^0 + d_{-1} \times10^{-1} + d_{-2} \times10^{-2}+\ldots+d_{-m} \times10^{-m}$$

  其中，d_i（$i=n,n-1, \ldots ,1,0, -1, -2, \ldots -m$）可以是0,1,2,3,4,5,6,7,8,9这10个数字符号中的任何一个；

  “10”称为基数（base），它代表每个数位上可以使用的不同数字符号个数。**10^i 称为第i位上的权**。

  运算时，“**逢十进一**”。

二进制（Binary）计数制

- **二进制数**，每个数位可用**两个不同符号**0和1来表示，每个符号处在不同位置时，所代表的数值不一样。

  例如，100101.01代表的值是：

  $(100101.01)_2 = 1\times2^5 + 0\times2^4 + 0\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} = 37.25$

- 一般地，任意一个二进制数

  $B=b_n b_{n-1} \ldots b_1 b_0 . b_{-1} b_{-2} \ldots b_{-m}$ （m,n为正整数)

- 其值可表示为如下形式：

  $V(B) = b_n\times2^n + b_{n-1}\times2^{n-1} + \ldots + b_1\times2^1 + b_0\times2^0 + b_{-1}\times2^{-1} + b_{-2}\times2^{-2} + \ldots + b_{-m}\times2^{-m}$

  其中，b_i（$i=n,n-1, \ldots ,1,0, -1, -2, \ldots -m$）可以是0或1

  “2”称为基数（base），它代表每个数位上可以使用的不同数字符号个数。**2^i 称为第i位上的权。**

  运算时，**“逢二进一”**。

后缀“B”表示二进制数，如01011010B

R进位计数制

- 在R进制数字系统中，应采用R个基本符号（0，1，2，．．．．，R-1）表示各位上的数字，采用“逢R进一”的运算规则，对于每一个数位i，该位上的权为R^i。R被称为该数字系统的基。

$2^3=8$，对应3位二进制

$2^4=16$，对应4位二进制

二进制：R=2，基本符号为0和1

八进制：R=8，基本符号为0,1,2,3,4,5,6,7

十六进制：R=16，基本符号为0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

十进制：R=10，基本符号为0,1,2,3,4,5,6,7,8,9

二进制	八进制	十进制	十六进制	二进制	八进制	十进制	十六进制
0000	0	0	0	1000	10	8	8
0001	1	1	1	1001	11	9	9
0010	2	2	2	1010	12	10	A
0011	3	3	3	1011	13	11	B
0100	4	4	4	1100	14	12	C
0101	5	5	5	1101	15	13	D
0110	6	6	6	1110	16	14	E
0111	7	7	7	1111	17	15	F

八进制和十六进制

日常生活中用十进制表示数值，计算机中用二进制表示所有信息！
那为什么还要引入 八进制 / 十六进制呢？

八进制 / 十六进制是二进制的简便表示。便于阅读和书写！

它们之间对应简单，转换容易。

在机器内部用二进制表示，在屏幕或其他设备上表示时，转换为八进制/十六进制数，可缩短长度。

八进制：Octal （用后缀“O”表示）
十六进制：Hexadecimal （用后缀“H” ，或前缀“0x”表示）

例：1010 1100 0100 0101 0001 0000 1000 1101B可写成
0xac45108d 0xAC45108D 或 ac45108dH AC45108DH
或 8进制：25421210215O
010 101 100 010 001 010 001 000 010 001 101

现代计算机系统多用十六进制表示机器数

十进制数与R进制数之间的转换

(1) R进制数 => 十进制数

按“权”展开

例1: $(10101.01)_2=1x2^4+1x\ 2^2+1x2^0+1x2^{-2}=(21.25)_{10}$

例2: $(307.6)_8=3x8^2+7x8^0+6x8^{-1}=(199.75)_{10}$

例1: $(3A.\ 1)_{16}=3x16^1+10x16^0+1x16^{-1}=(58.0625)_{10}$

(2)十进制数 => 二进制数，再将二进制转换为16或8进制

整数部分和小数部分分别转换

① 整数----“除基取余，上右下左”

② 小数----“乘基取整，上左下右”

理论上的做法

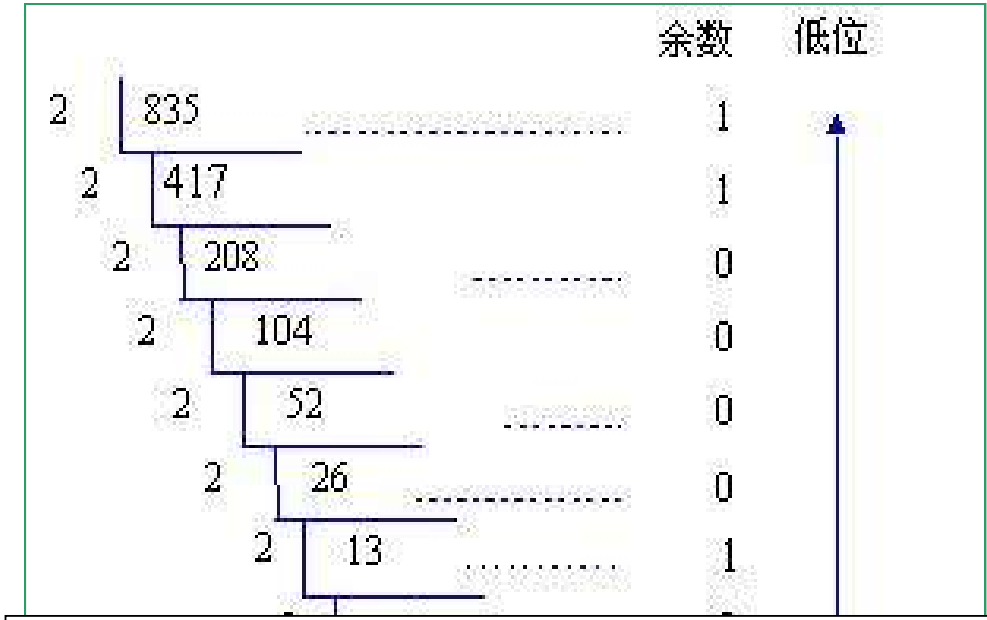
实际上，记住1、2、4、8、16、32、64、128、256、512、1024、2048、4096、8192、16384、32768、65536，.....就可简单进行整数部分的转换

记住0.5、0.25、0.125、0.0625、...... 就可简单进行小数部分的转换

十进制数与二进制数之间的转换

例1: $(835.6875)_{10}=(11\ 0100\ 0011.1011)_2$

整数----“除基取余，上右下左”



小数----“乘基取整，上左下右”

0.6875×2=1.375	整数部分=1	(高位)
0.375×2=0.75	整数部分=0	↓
0.75×2=1.5	整数部分=1	↓
0.5×2=1.0	整数部分=1	(低位)

简便方法：835=512+256+64+2+1，故结果为 11 0100 0011

0.6875=0.5+0.125+0.0625，故结果为 0.1011

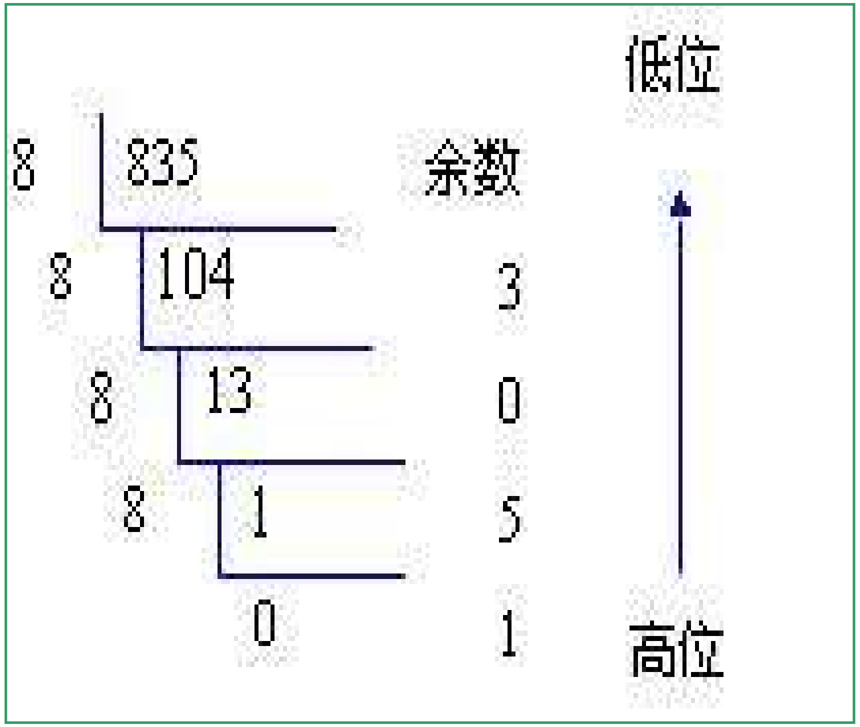
结果为 11 0100 0011.1011

这里有一个问题：小数点在计算机中如何表示？

十进制数与8进制数之间的转换

例2:$(835.63)_{10}=(1503.50243\ldots)_8$

整数----“除基取余，上右下左” **小数----“乘基取整，上左下右”**



0.63×8=5.04 整数部分=5 (高位)

0.04×8=0.32 整数部分=0

0.32×8=2.56 整数部分=2

0.56×8=4.48 整数部分=4

0.48×8=3.84 整数部分=3 (低位)

可能小数部分总得不到0，此时得到一个近似值

说明：现实中的精确值可能在机器内部无法用0和1精确表示！

定点数和浮点数

- 计算机中只有0和1，数值数据中的小数点怎么表示呢？
  - 计算机中只能通过约定小数点的位置来表示
    - 小数点位置约定在固定位置的数称为定点数
    - 小数点位置约定为可浮动的数称为浮点数
- 定点小数用来表示浮点数的尾数部分
- 定点整数用来表示整数，分带符号整数和无符号整数
- 任何实数：$X=(-1)^S \times M \times R^E$

结论：要解决数值数据的表示问题，只要解决定点数的编码问题！

其中，S取值为0或1，用来决定数X的符号；M是一个二进制定点小数，称为数X的尾数（mantissa）；E是一个二进制定点整数，称为数X的阶或指数（exponent）；R是基数（radix、base），可以为2、4和16等。 计算机中只要表示S、M和E三个信息，就能确定X的值，这称为浮点数

S	E	M