

# Lec-2,3-数据的编号系统

## Lec-2,3-数据的编号系统

基础概念

各种进制的转换

二进制 --> 十进制

Binary coded decimal (BCD)二进制编码的十进制

二进制负数的表示方式 (Important!!!)

Conversion between lengths

二进制数字的计算

加减法

乘法

二进制小数相关 (Important!!!)

## 基础概念

Decimal ---- 十进制

Binary ---- 二进制

Hexadecimal ---- 十六进制

Octal number ---- 八进制数

Bit ---- 位

Byte ---- 字节

Word ---- 字 (处理器一次可以处理 (读/写) 的位数)

## 各种进制的转换

### 二进制 --> 十进制

### Binary coded decimal (BCD)二进制编码的十进制

- Complicated to do binary conversion to display numbers
- Unsigned integer
- Replace each denary digit by its 4-bit binary representation

Decimal	Binary	BCD	
68	0100 0100	0110 1000	
99	0110 0011	1001 1001	largest 8-bit BCD
255	1111 1111	0010 0101 0101	largest 8-bit binary

## 二进制负数的表示方式 (Important!!!)

N = 4	Number Represented			
	Unsigned	Signed Mag	1's Comp	2's Comp
0000	0	0	0	0
0001	1	1	1	1
0010	2	2	2	2
0011	3	3	3	3
0100	4	4	4	4
0101	5	5	5	5
0110	6	6	6	6
0111	7	7	7	7
1000	8	-0	-7	-8
1001	9	-1	-6	-7
1010	10	-2	-5	-6
1011	11	-3	-4	-5
1100	12	-4	-3	-4
1101	13	-5	-2	-3
1110	14	-6	-1	-2
1111	15	-7	-0	-1

- Sign-and-magnitude (“符号-数值”表示法)
  - 0 means positive
  - 1 means negative
  - (问题是加减的时候不好使用)
- 1's-complement (反码)
  - Flip every bit to represent negative
  - (问题同样是加减的时候不好使用)
- 2's-complement (most common) (补码)
  1. Flipping bits
  2. Add one to the 1's complement

## Conversion between lengths

In general, pack with most-significant bits (MSB)

+18 = 0001 0010

+18 = 0000 0000 0001 0010

-18 = 1110 1110

-18 = 111111111110 1110

## 二进制数字的计算

### 加减法

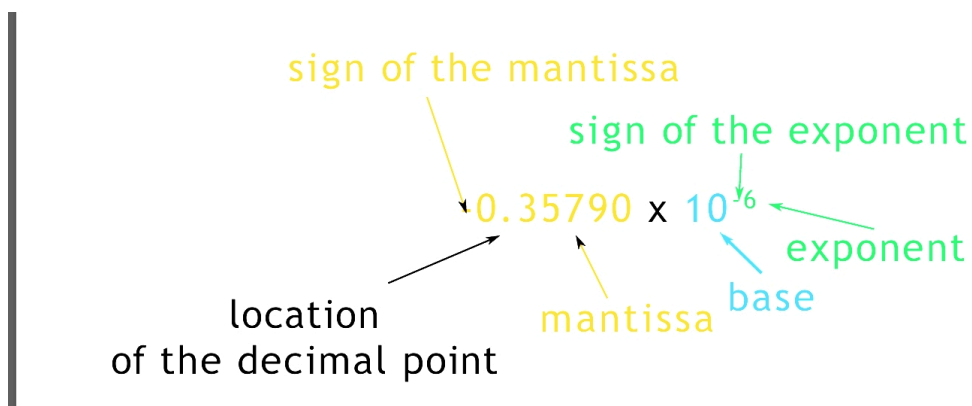
二进制中，使用补码进行加减计算的时候，算出来的数字看开头的数字，是0就是正数，当 $A+(-B)$ 中若 $|A| > |B|$ 则计算出来就是正数，但这种情况都是会存在进位的情况，使得溢出，有效数字最高位变成进位之后的0？这里的溢出是需要注意的

### 乘法

Q：为什么要取最低的有效位，如： $15 \times 15$ 。仅使用4位表示不了

## 二进制小数相关 (Important!!!)

Binary Floating-Point Numbers(二进制浮点数)



# Binary Floating-Point Numbers

IEEE 754 Floating-Point Standard (32-bits):

$$\text{sign} * \text{fraction} * 2^{\text{exp}}$$



Exponent is represented in 127-bias notation

1. **符号位 (Sign bit)** : 1位, 表示数值的正负, 0表示正数, 1表示负数。
2. **指数位 (Exponent bits)** : 8位, 用于存储指数部分。指数位使用的是偏移量 (bias) 表示法, 对于32位浮点数, 偏移量是127。实际指数值需要从存储的指数值中减去偏移量来获得。

Bias 127 means that we use the numbers 0 to 255 in order to represent the numbers -127 to 128

**使用的时候直接加上-127就好**

3. **尾数位 (Mantissa bits)** : 23位, 也称为有效数字位, 用于存储小数部分。在IEEE 754标准中, 尾数位的第一位是隐含的, 即默认为1 (除非是特殊情况, 如0), 所以实际存储的尾数位只有23位, 但表示的数值范围是24位

二进制浮点数的乘法

1. 将指数相加
2. 尾数相乘
3. 对结果进行标准化