

浮点数的表示

定点数的局限性

定点数可表示的数字范围有限，但我们不能无限制地增加数据的长度

我的财富： - 8540 ¥

2B 定点整数 short 即可表示

马云的财富： +302657264526 ¥

4B 定点整数 int.....都表示不了

8B long型也
表示不了

如果换一种货币：1 人民币 ≈ 10000000000 津巴布韦币

如何在位数不变的情况下增加数据表示范围？

从科学计数法理解浮点数

普通计数法：

+302657264526

科学计数法：

+3.026 * 10¹¹

+11 +3.026

1 人民币 $\approx 10^{10}$ 津巴布韦币

阶码反映
数值大小

+21 +3.026

尾数反映
精度

阶码

尾数

J_f	$J_1 J_2 \dots J_m$	S_f	$S_1 S_2 \dots S_n$
阶符	阶码的数值部分	数符	尾数的数值部分

浮点数表示

$$r \text{ 进制: } K_n K_{n-1} \dots K_2 K_1 K_0 K_{-1} K_{-2} \dots K_{-m}$$

$$= K_n \times r^n + K_{n-1} \times r^{n-1} + \dots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0 + K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \dots + K_{-m} \times r^{-m}$$

定点数: 如纯小数0.1011和纯整数11110

浮点数:		阶码	尾数	
J _f	J ₁ J ₂ … J _m	S _f	S ₁ S ₂ … S _n	
阶符	阶码的数值部分	数符	尾数的数值部分	

浮点数的真值: $N = \boxed{r^E} \times M$

阶码的底, 通常为2

阶码E反映浮点数的表示范围及小数点的实际位置;
尾数M的数值部分的位数n反映浮点数的精度。

阶码: 常用补码或移码表示的定点整数

尾数: 常用原码或补码表示的定点小数

例: 阶码、尾数均用补码表示, 求a、b的真值

$$a = 0,01;1.1001$$

$$b = 0,10;0.01001$$

b: 阶码0,10对应真值+2

尾数0.01001对应真值 $+0.01001 = + (2^{-2} + 2^{-5})$

$$\text{b的真值} = 2^2 \times (+0.01001) = +1.001$$

相当于尾数表示的定点小数算数
左移2位, 或小数点右移2位

1B的存储空间

0 1 0 0 0 1 0 0 1



浮点数尾数的规格化

浮点数尾数的规格化

浮点数:		阶码	尾数	
J _f	J ₁ J ₂ … J _m	S _f	S ₁ S ₂ … S _n	
阶符	阶码的数值部分	数符	尾数的数值部分	
浮点数的真值: $N = \boxed{r^E} \times M$				
阶码的底, 通常为2				
阶码E反映浮点数的表示范围及小数点的实际位置; 尾数M的数值部分的位数n反映浮点数的精度。				
1B的存储空间				
0 1 0 0 0 1 0 0 1				

$+302657264526 = +3.026 * 10^{+1}$

可记为: $+11 +3.026$

也可记为: $+14 +0.003$

$10^{+1} \times 302.6$ 尾数的最高位是无效值, 会丧失精度

阶码: 常用补码或移码表示的整数

尾数: 常用原码或补码表示的小数

例: 阶码、尾数均用补码表示, 求a、b的真值

$a = 0,01;1.1001$

$b = 0,10;0.01001$

b: 阶码0,10对应真值+2
尾数0.01001对应真值 $+0.01001 = + (2^{-2} + 2^{-5})$

所以 $b = 2^2 \times (+0.01001) = +1.001$

尾数算数左移1位, 阶码减1。直到尾数最高位是有效值(左规)

王道考研/CSKAOYAN.COM

浮点数尾数的规格化

通过算数左移、
阶码减1 来规格化

规格化浮点数：规定尾数的最高数值位必须是一个有效值。

左规：当浮点数运算的结果为非规格化时要进行规格化处理，
将尾数算数左移一位，阶码减1。

通过算数右移、
阶码加1 来规格化

右规：当浮点数运算的结果尾数出现溢出（双符号位为01或10）时，
将尾数算数右移一位，阶码加1。

例： $a = 010.00.1100$, $b = 010.00.1000$, 求 $a+b$

$$\begin{aligned} a &= 2^2 \times 00.1100, b = 2^2 \times 00.1000 \\ a+b &= 2^2 \times 00.1100 + 2^2 \times 00.1000 \\ &= 2^2 \times (00.1100 + 00.1000) \\ &= 2^2 \times 01.0100 \quad \text{右规} \\ &= 2^3 \times 00.1010 \end{aligned}$$

注：采用“双符号位”，当溢出发生时，可以挽救。更高的符号位是正确的符号位

王道考研/CSKAOYAN.COM

规格化浮点数的特点

规格化浮点数的特点

规格化的原码尾数，最
高数值位一
定是1

1. 用原码表示的尾数进行规格化：
正数为 $0.1 \times \dots \times$ 的形式，其最大值表示为 $0.11\dots1$ ；最小值表示为 $0.10\dots0$ 。
尾数的表示范围为 $1/2 \leq M \leq (1-2^{-n})$ 。
负数为 $1.1 \times \dots \times$ 的形式，其最大值表示为 $1.10\dots0$ ；最小值表示为 $1.11\dots1$ 。
尾数的表示范围为 $-(1-2^{-n}) \leq M \leq -1/2$ 。

规格化的补
码尾数，符
号位与最高
数值位一定
相反

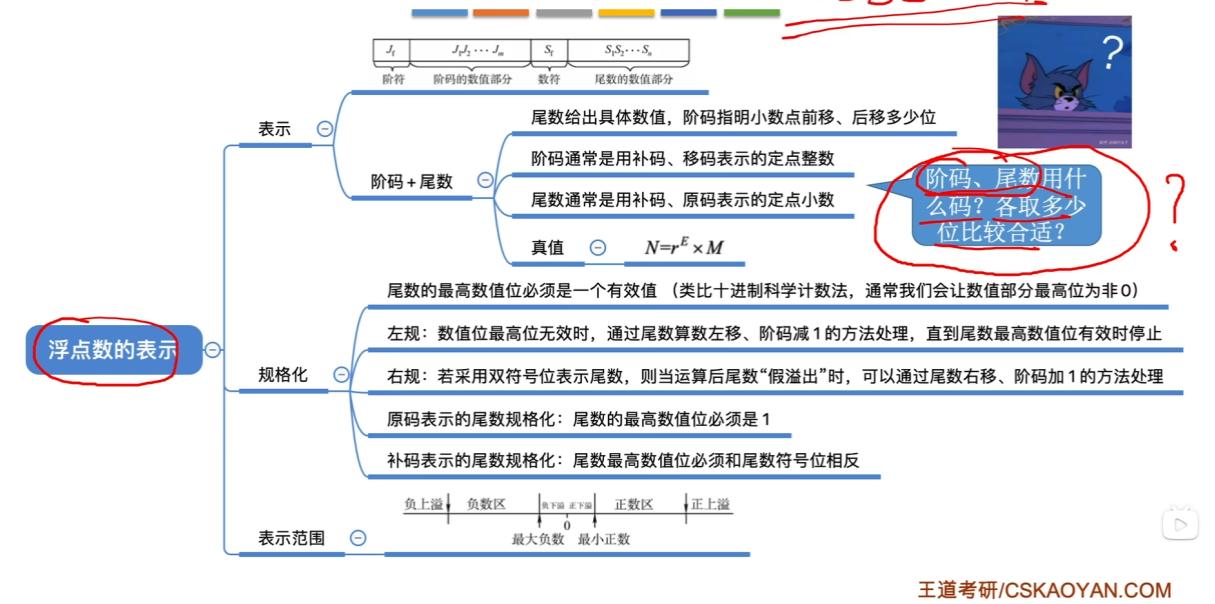
2. 用补码表示的尾数进行规格化：
正数为 $0.1 \times \dots \times$ 的形式，其最大值表示为 $0.11\dots1$ ；最小值表示为 $0.10\dots0$ 。
尾数的表示范围为 $1/2 \leq M \leq (1-2^{-n})$ 。
负数为 $1.0 \times \dots \times$ 的形式，其最大值表示为 $1.01\dots1$ ；最小值表示为 $1.00\dots0$ 。
尾数的表示范围为 $-1 \leq M \leq -(1/2+2^{-n})$ 。



eg: 若某浮点数的阶码、尾数用补码表示，共4+8位：
0.110; 1.1110100 如何规格化？

注：补码算数左移，低位补0；补码算数右移，高位补1。

王道考研/CSKAOYAN.COM



IEEE 754

移码：补码的基础上将符号位取反。注意：移码只能用于表示整数

$$x = +19D$$

$$[x]_{\text{原}} = 0,0010011$$

$$[x]_{\text{反}} = 0,0010011$$

$$[x]_{\text{补}} = 0,0010011$$

$$[x]_{\text{移}} = 1,0010011$$

$$x = -19D$$

$$[x]_{\text{原}} = 1,0010011$$

$$[x]_{\text{反}} = 1,1101100$$

$$[x]_{\text{补}} = 1,1101101$$

$$[x]_{\text{移}} = 0,1101101$$

定点整数
的表示

移码的定义：移码=真值+偏置值

移码

真值(十进制)	补码	移码
-128	1000 0000	0000 0000
-127	1000 0001	0000 0001
-126	1000 0010	0000 0010
...
-3	1111 1101	0111 1101
-2	1111 1110	0111 1110
-1	1111 1111	0111 1111
0	0000 0000	1000 0000
1	0000 0001	1000 0001
2	0000 0010	1000 0010
3	0000 0011	1000 0011
...
124	0111 1100	1111 1100
125	0111 1101	1111 1101
126	0111 1110	1111 1110
127	0111 1111	1111 1111

移码的定义: 移码=真值+偏置值

偏置值一般取 2^{n-1} , 此时移码=补码
符号位取反

此处8位移码的偏置值=128D=1000 0000B, 即 2^{n-1}

真值 -127 = -1111 111B

移码 = -1111 111 + 1000 0000 = 0000 0001

真值 -3 = -11B

移码 = -11 + 1000 0000 = 0111 1101

真值 +0 = +0

移码 = +0 + 1000 0000 = 1000 0000

真值 +3 = +11B

移码 = +11 + 1000 0000 = 1000 0011

真值 +127 = +1111 111B

移码 = +1111 111 + 1000 0000 = 1111 1111

真值(十进制)	补码	移码	移码
-128	1000 0000	0000 0000	1111 1111
-127	1000 0001	0000 0001	0000 0000
-126	1000 0010	0000 0010	0000 0001
...
-3	1111 1101	0111 1101	0111 1100
-2	1111 1110	0111 1110	0111 1101
-1	1111 1111	0111 1111	0111 1110
0	0000 0000	1000 0000	0111 1111
1	0000 0001	1000 0001	1000 0000
2	0000 0010	1000 0010	1000 0001
3	0000 0011	1000 0011	1000 0010
...
124	0111 1100	1111 1100	1111 1011
125	0111 1101	1111 1101	1111 1100
126	0111 1110	1111 1110	1111 1101
127	0111 1111	1111 1111	1111 1110

移码的定义: 移码=真值+偏置值

偏置值可以取其他值

令偏置值=127D=0111 1111B, 即 $2^{n-1}-1$

真值 -128 = -1000 0000B

移码 = -1000 0000 + 0111 1111 = 1111 1111

真值 -127 = -111 1111B

移码 = -111 1111 + 0111 1111 = 0000 0000

真值 -126 = -111 1110B

移码 = -111 1110 + 0111 1111 = 0000 0001

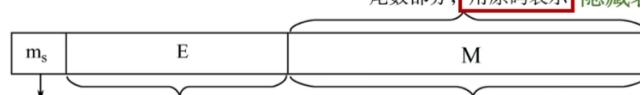
真值 +0 = +0

移码 = +0 + 0111 1111 = 0111 1111

真值 +127 = +1111 111B

移码 = +111 1111 + 0111 1111 = 1111 1110

阶码全1、全0
用作特殊用途



表示尾数1.M

真值正常范围:

-126~127

偏置值= $2^{n-1}-1$

类型	数 符	阶 码	尾 数 数 值	总 位 数	十六 进 制	十 进 制
短浮点数	1	8	23	32	7FH	127
长浮点数	1	11	52	64	3FFH	1023
临时浮点数	1	15	64	80	3FFFH	16383

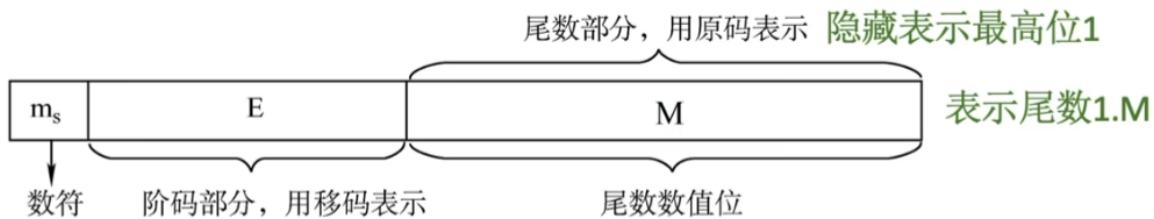
float 1000 0001 1000 1010 0101 0000 1000 0000

double 1000 0001 1100 0010 0101 0000 1000 0000 0000 0000 0001 1111 0000 0000 0000 0000

规格化的短浮点数的真值为: $(-1)^s \times 1.M \times 2^{E-127}$

规格化长浮点数的真值为: $(-1)^s \times 1.M \times 2^{E-1023}$

阶码真值=移码-偏移量



例：将十进制数 -0.75 转换为 IEEE 754 的单精度浮点数格式表示。

$$(-0.75)_{10} = (-0.11)_2 = (-1.1)_2 \times 2^{-1}$$

数符 = 1

尾数部分 = .1000000.... (隐含最高位1)

阶码真值 = -1

单精度浮点型偏移量 = 127D

移码 = 阶码真值+偏移量 = -1 + 111 1111 = 0111 1110 (凑足8位)

→ 1 01111110 10000000000000000000000000

例：IEEE 754 的单精度浮点数 C0 A0 00 00 H 的值时多少。

C0 A0 00 00 H → 1100 0000 1010 0000 0000 0000 0000 0000

数符 = 1 → 是个负数

尾数部分 = .0100.... (隐含最高位1) → 尾数真值 = (1.01)₂

移码 = 10000001, 若看作无符号数= 129D

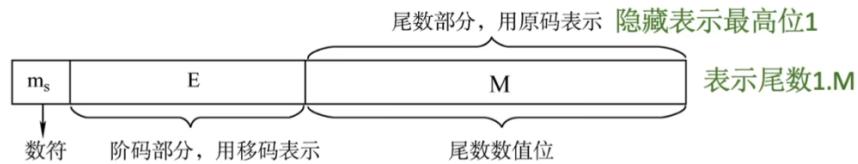
单精度浮点型偏移量 = 127D

阶码真值= 移码 - 偏移量 = 1000 0001 - 111 1111 = (0000 0010)₂ = (2)₁₀

→ 浮点数真值 = (-1.01)₂ × 2² = -1.25 × 2² = -5.0

IEEE 754 标准

天然地完成了“规格化”



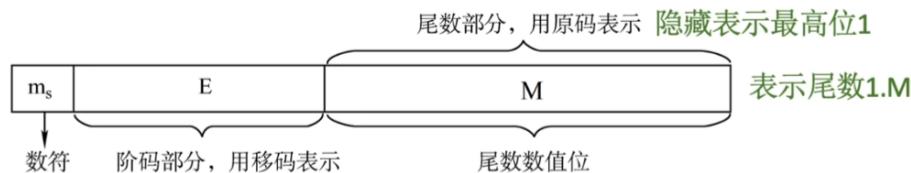
IEEE 754 单精度浮点型能表示的最小绝对值、最大绝对值是多少？

最小绝对值：尾数全为0，阶码真值最小-126，对应移码机器数 0000 0001
此时整体的真值为 $(1.0)_2 \times 2^{-126}$

最大绝对值：尾数全为1，阶码真值最大 127，对应移码机器数 1111 1110
此时整体的真值为 $(1.11\dots1)_2 \times 2^{127}$

格 式	规格化的最小绝对值	规格化的最大绝对值
单精度	$E=1, M=0: 1.0 \times 2^{1-127} = 2^{-126}$	$E=254, M=.11\dots1: 1.11\dots1 \times 2^{254-127} = 2^{127} \times (2 - 2^{-23})$
双精度	$E=1, M=0: 1.0 \times 2^{1-1023} = 2^{-1022}$	$E=2046, M=.11\dots1: 1.11\dots1 \times 2^{2046-1023} = 2^{1023} \times (2 - 2^{-52})$

阶码全1、全0
用作特殊用途



类 型	数 符	阶 码	尾 数 数 值	总 位 数	偏 置 值	
					十 六 进 制	十 进 制
短浮点数	1	8	23	32	7FH	127
长浮点数	1	11	52	64	3FFH	1023
临时浮点数	1	15	64	80	3FFFH	16383

由浮点数确定真值（阶码不是全0、也不是全1）：

- 根据“某浮点数”确定数符、阶码、尾数的分布
- 确定尾数 1.M （注意补充最高的隐含位1）
- 确定阶码的真值 = 移码 - 偏置值 （可将移码看作无符号数，用无符号数的值减去偏置值）
- $(-1)^s \times 1.M \times 2^{E-\text{偏置值}}$