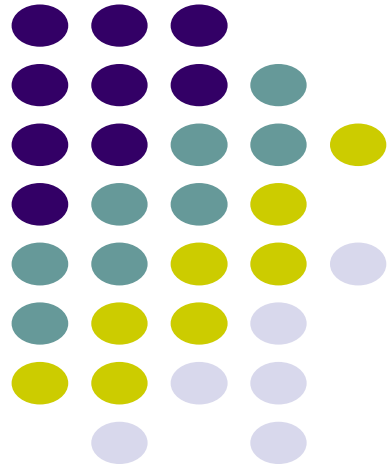
《计算机系统基础（四）：编程与调试实践》

浮点数的表示和基本运算



浮点数的表示和基本运算

IEEE 754浮点数标准

尾数的舍入处理

浮点数的基本运算

IEEE 754浮点数标准

1. IEEE 754浮点数的基本格式

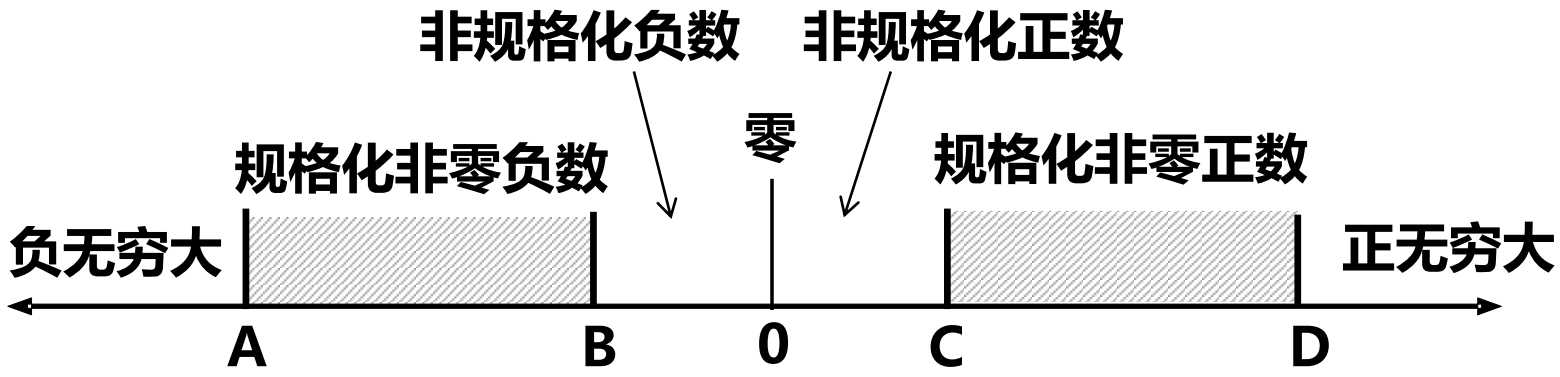
① 32位单精度浮点格式，即float格式

1位	8位	23位
符号	阶码	尾数

② 64位双精度浮点格式，即double格式

1位	11位	52位
符号	阶码	尾数

2. IEEE 754标准中的数据按值的分类



IEEE 754浮点数标准

IEEE 754浮点数32位单精度格式各类值的编码

	1位	8位	23位
正零和负零	0/1	0	0
非规格化	0/1	0	f≠0
规格化非零	0/1	1-254	f
无穷大	0/1	255	0
无定义数	0/1	255	f≠0

IEEE 754浮点数标准

规格化数的真值与机器数的对应关系：

真值： $+/-1.xxxxxxxxxx * 2^E$

机器数：

例如

真值： $5.0=1.01B*2^2$ 2+127=1000 0001B

机器数： 0 1000 0001 010 0000 0000 0000 0000 0000B

=40a0 0000H

机器数：

真值： $(-1)^S \times (1 + Significand) * 2^{(Exponent-127)}$

例如

机器数： 40a0 0000H

=0 1000 0001 010 0000 0000 0000 0000 0000B

真值： $1.01B*2^2$ 1000 0001B-127=2

IEEE 754浮点数标准

非规格化数的真值与机器数的对应关系：

真值： $+/-0.xxxxxxxxxxx * 2^{-126}$

机器数：

例如

真值： $1e\text{-}40=10^{-40}\approx0.000\ 0001\ 0001\ 0110\ 1100\ 0010B*2^{-126}$

机器数： 0 0000 0000 000 0001 0001 0110 1100 0010B

=0001 16C2H

机器数：

真值： $(-1)^S$ x Significand $* 2^{-126}$

例如

机器数： 0001 16C2H

=0 0000 0000 000 0001 0001 0110 1100 0010B

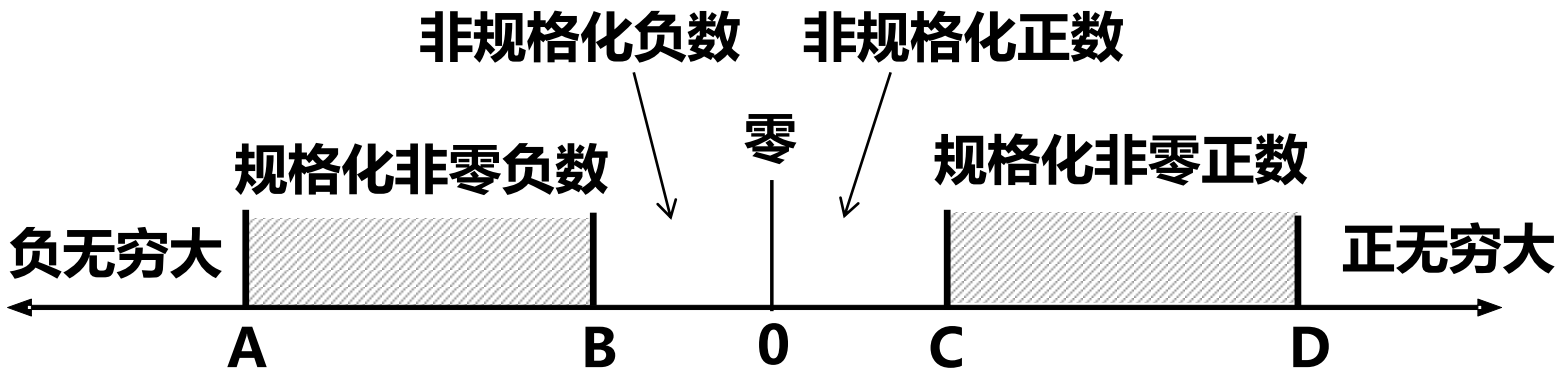
真值： $0.000\ 0001\ 0001\ 0110\ 1100\ 0010B*2^{-126}$

尾数的舍入处理

32位单精度浮点格式，即float格式

1位	8位	23位
符号	阶码	尾数

24位精度

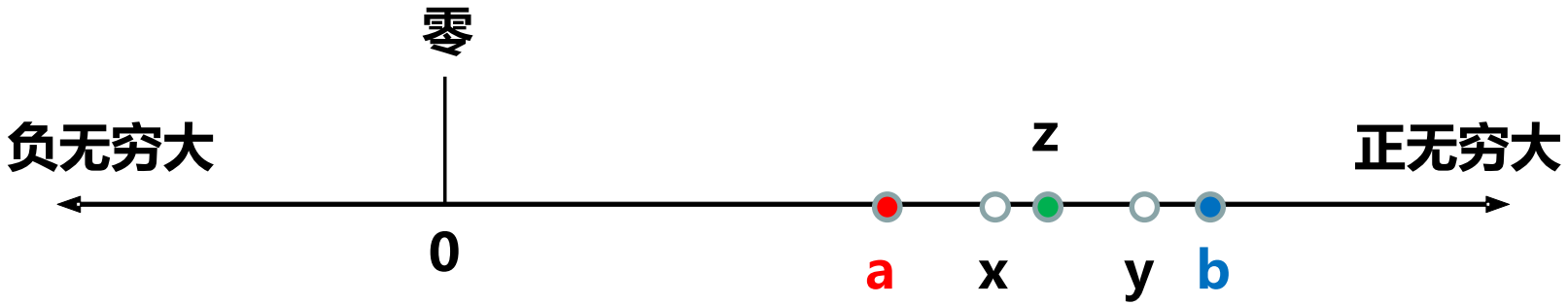


0.1=0.000 1100 1100 1100 1100 1100 1100 1100 ……B

尾数的舍入处理

IEEE754标准提供四种舍入模式：

① 就近舍入（中间值舍入到偶数）	x≈a、y≈b、z≈?
② 朝+∞方向舍入	x≈b、y≈b、z≈b
③ 朝-∞方向舍入	x≈a、y≈a、z≈a
④ 朝0方向舍入	x≈a、y≈a、z≈a



a、 b： 两个连续的浮点格式可表示的数据

x、y、z：需要浮点编码的数据，z=(a+b)/2

尾数的舍入处理

就近舍入方法（以32位单精度浮点格式为例）：

真值的尾数：

$1.x_1x_2\cdots\cdots x_{n-1}x_nB$　　假设 n>23

$=\ 1.\ x_1x_2\cdots x_{23}x_{24}x_{25}x_{26}\cdots x_nB$　　粘位处理

$=\begin{cases} 1.\ x_1x_2\cdots x_{23}x_{24}x_{25}0\ B & \text{若}x_{26}\cdots x_n\text{全为0} \\ 1.\ x_1x_2\cdots x_{23}x_{24}x_{25}1\ B & \text{若}x_{26}\cdots x_n\text{不全为0} \end{cases}$

需要截断的位只有3位了，有8种编码000~111，把100看成是中间值

① 000~011　小于100，舍，$1.\ x_1x_2\cdots x_{23}$

② 101~111　大于100，入，$1.\ x_1x_2\cdots x_{23}$ +0.0…01　（最低位加1）

③ 100 $\begin{cases} \text{若}x_{23}=0\text{，则　舍，}1.\ x_1x_2\cdots x_{23} \\ \text{若}x_{23}=1\text{，则　入，}1.\ x_1x_2\cdots x_{23}\ +0.0\cdots01 \quad \text{（最低位加1）} \end{cases}$

尾数的舍入处理

就近舍入方法（以32位单精度浮点格式为例）：

数据	尾数	“粘位”处理	舍入规则	新尾数
8000000H	1.…00 0000B	1.…00 000B	就近舍	1.…00 B
8000001H	1.…00 0001B	1.…00 001B	就近舍	1.…00 B
8000014H	1.…01 0100B	1.…01 010B	就近舍	1.…01 B
8000017H	1.…01 0111B	1.…01 011B	就近舍	1.…01 B
8000008H	1.…00 1000B	1.…00 100B	中间数，舍	1.…00 B
8000018H	1.…01 1000B	1.…01 100B	中间数，入	1.…10 B
8000019H	1.…01 1001B	1.…01 101B	就近入	1.…10 B
800000CH	1.…00 1100B	1.…00 110B	就近入	1.…01 B
800000DH	1.…00 1101B	1.…00 111B	就近入	1.…01 B

浮点数的基本运算

1. 设两个规格化浮点数分别为 $A=M_a \cdot 2^{Ea}$　$B=M_b \cdot 2^{Eb}$,则:

$A \pm B = (M_a \pm M_b \cdot 2^{-(Ea-Eb)}) \cdot 2^{Ea}$　(假设Ea>=Eb)

$A*B = (M_a * M_b) \cdot 2^{Ea+Eb}$

$A/B = (M_a / M_b) \cdot 2^{Ea-Eb}$

2. 浮点运算部件

早期：浮点协处理器芯片（FPU）：8087、80287

　　CPU：8086/8088、80286/80386

现在：CPU　定点运算部件

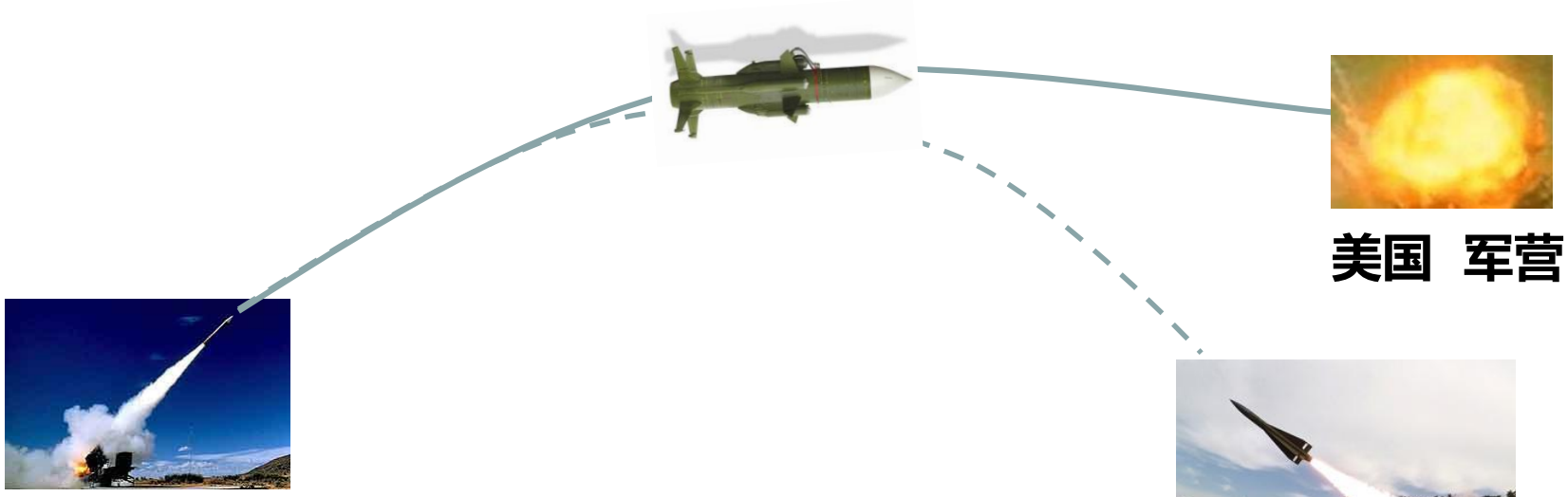
　　浮点运算部件

3. 浮点数的运算中有对阶、舍入、溢出等问题，导致运算结果会出现大数吃小数、精度误差、结果异常等问题。

4. 爱国者导弹定位错误的案例分析：差之毫厘，失之千里

浮点数的基本运算

(1) 事故：爱国者导弹定位错误。



伊拉克 飞毛腿导弹

美国 军营

美国 爱国者导弹

(2) 原因：0.1的计算机表示误差

0.1的误差很小，但运算后的累计误差就大了。

浮点数的基本运算

(3) 数据：

爱国者导弹系统的内置时钟，每隔0.1秒计数一次；

爱国者已经连续工作100小时；

飞毛腿导弹的飞行速度约为2000米/秒；

(4)爱国者系统时钟的误差导致计算的距离偏差是多少?

(5) 分析：

0.1=0.000 1100 1100 1100 1100 1100 [1100]•••B

程序：x=0.000 1100 1100 1100 1100 1100B　24位定点小数表示0.1

时间误差：(0.1-x)*100*60*60*10≈0.3433秒

距离误差：2000*0.3433=686.6米

(6) 讨论：

对 0.1采用不同的表示方式，计算的距离误差分别是多少?

float格式、32位定点小数、就近舍入后的24位定点小数

谢谢！