计算机组成原理

PRINCIPLES OF COMPUTER ORGANIZATION

第5次课：3.2 定点数加减法运算

杜国栋

信息科学与工程学院计算机科学与工程系

gddu@ysu.edu.cn

燕山大学

YANSHAN UNIVERSITY

厚德 · 博学 · 求是

IEEE754浮点数标准格式

32 位和 64 位的 IEEE754 浮点数格式如下：

S(1 位)	E（8/11 位）	M(23/52 位)

S 为浮点数的符号位，正 0，负 1；

M 为尾数小数值部分，且 IEEE754 的尾数标准形式为 1.xxxxxx，所以 1.是隐藏值，故 23 位尾数可表示 24 位的值；

E 为阶码，包含 1 位符号位和 7 位阶码的数值部分。

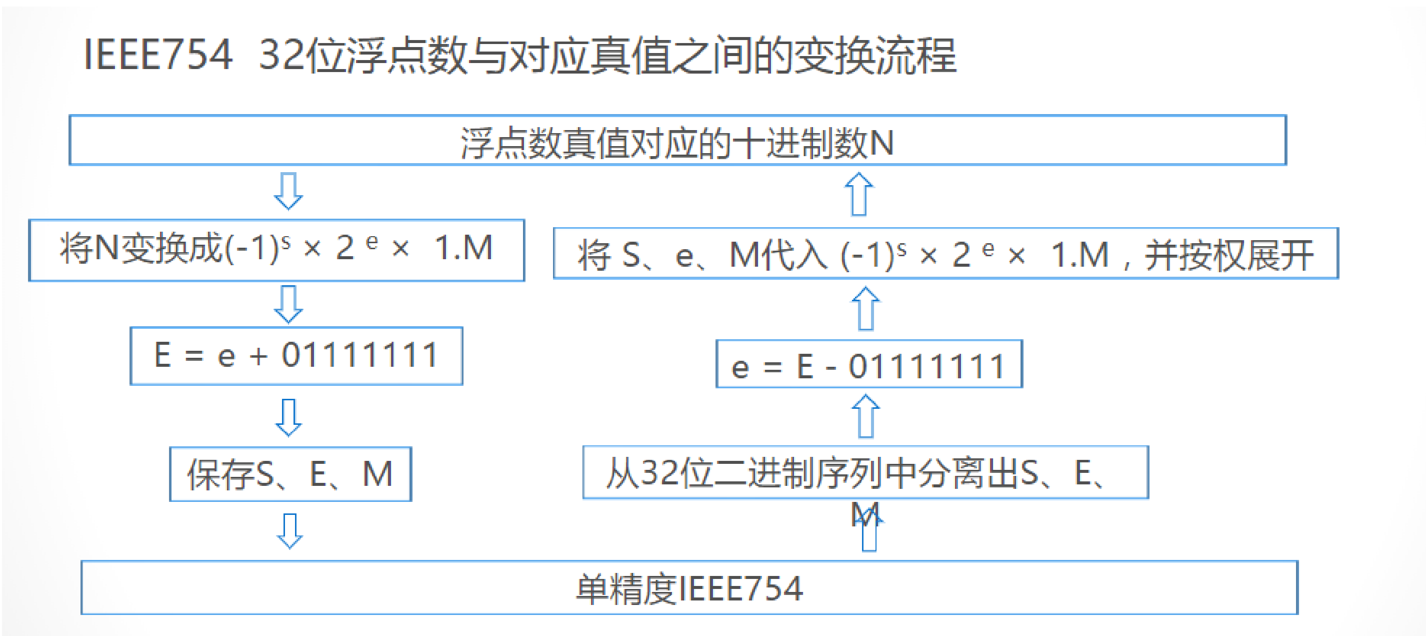
单精度（32 位）浮点数 IEEE754 的真值：

$$x = (-1)^s \times (1.M) \times 2^{E-127}$$

S	阶码（含阶符）	尾　　数

数符　　　　小数点位置

IEEE754浮点数标准格式



Float型数据通常用IEEE754单精度浮点数格式表示，若编译器将Float型变量x分配在一个32位浮点寄存器FRI中，且x=-8.25，则FRI的内容是（ ）

A. C1040000H

B. C2420000H

C. C1840000H

D. C1C20000H

Float型数据通常用IEEE754单精度浮点数格式表示，若编译器将Float型变量x分配在一个32位浮点寄存器FRI中，且x=-8.25，则FRI的内容是（ ）

C1040000H

C2420000H

C1840000H

C1C20000H

例 3：float 型数据通常用 IEEE754 单精度浮点数格式表示，若编译器将 float 型变量 x 分配在一个 32 位浮点寄存器 FRI 中，且 x=-8.25，则 FRI 的内容是（）

A. C1040000H B. C2420000H C. C1840000H D. C1C20000H

解：$(-8.25)_{10}=(-1000.01)_2$，$-1000.01=-1.00001\times2^3$

根据 IEEE754 单精度浮点数格式,E-127=3,故 E=130,转换成二进制为 1000 0010。

根据 IEEE754 标准,最高位的“1”是被隐藏的。

IEEE754 单精度浮点数格式:数符(1 位)+阶码(8 位)十尾数(23 位)。

故,FR1 内容为 1;1000 0010;0000 10000 0000 0000 0000 000。

即,1100 0001 0000 0100 0000 0000 0000 0000=C104000H。

Float型（即IEEE754单精度浮点数格式）能表示的最大正整数是?

A. $2^{126}-2^{108}$

B. $2^{127}-2^{104}$

C. $2^{127}-2^{108}$

D. $2^{128}-2^{104}$

例 4：float 型（即 IEEE754 单精度浮点数格式）能表示的最大正整数是：

A. $2^{126}-2^{108}$ B. $2^{127}-2^{104}$ C. $2^{127}-2^{108}$ D. $2^{128}-2^{104}$

解：根据 IEEE754 单精度浮点数格式：$x=(-1)^s\times(1.M)\times2^{E-127}$ E：1~254（8 位）

正数 S=0，尾数最大 M 为 23 位全为 1，阶码最大为 254

$x=(-1)^0\times(1.111\ldots1)\times2^{254-127}=(1+1-2^{-23})\times2^{127=}2^{128}-2^{104}$

阶码和尾数各为4位（各包含1位符号位），试问浮点数的表示范围为多少?

例 5：阶码和尾数各为 4 位（各包含 1 位符号位），试问浮点数的表示范围为多少？

解：阶码的范围：

二进制补码	最小负数	最大负数		最小正数	最大正数
	1000	1111	0	0001	0111
	-8	-1		1	7

规格化尾数表示范围如下：

二进制补码	最小负数	最大负数		最小正数	最大正数
	1.000	1.011	0	0.100	0.11
	-1	-（$2^{-3}+2^{-1}$）		2^{-1}	$1-2^{-3}$

规格化的浮点数的表示范围：

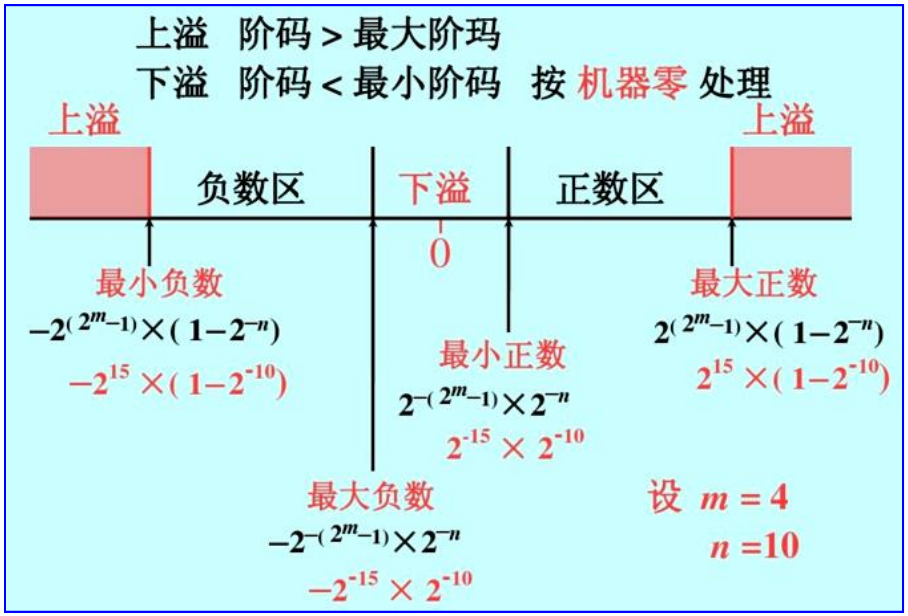
最小负数：$2^{0111}\times1.000$　　真值：$-2^{7}\times1$

最大负数：$2^{1000}\times1.011$　　真值：$-2^{-8}\times(2^{-3}+2^{-1})$

最小正数：$2^{1000}\times0.100$　　真值：$2^{-8}\times2^{-1}$

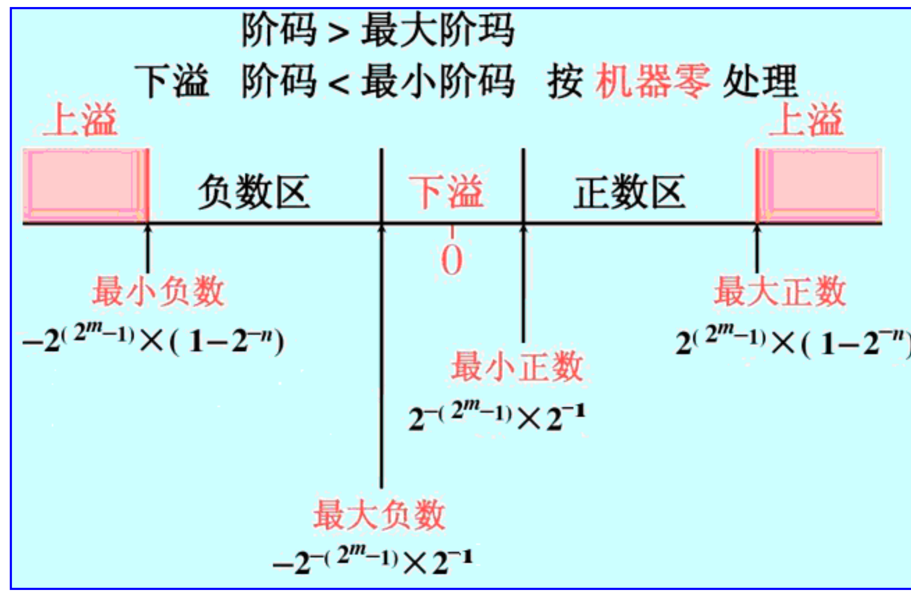
最大正数：$2^{0111}\times0.111$　　真值：$2^{7}\times(1-2^{-3})$

浮点数的范围



← 未规格化

规格化→



课程目标

- 掌握定点数的运算方法；
- 熟悉运算方法和溢出判断；
- 了解定点数运算方法的硬件实现。

补码运算基础

➢ 补码加法所依据的关系式

$$[x]_{补}+[y]_{补}=[x+y]_{补} \qquad (\mathrm{mod}\ 2^{n+1}/2)$$

➢ 补码减法所依据的关系式

$$[x-y]_{补}=[x+(-y)]_{补}=[x]_{补}+[-y]_{补}=[x]_{补}-[y]_{补} \qquad (\mathrm{mod}\ 2^{n+1}/2)$$

由[y]补求[-y]补的方法：

①当0≦y<1时

②当-1<y<0时，

补码运算基础

- 不管y的真值为正或者为负，已知[y]补，求[-y]补的方法就是：

  将[y]补**连同符号位一起取反，然后末位加1**（在定点小数中这个1是2^{-n}）。

- 由[y]补求[-y]补的过程称为对[y]补“变补”或“求补”。
- 变补是连同符号位 一起变反，末尾加“1”，目的是变减为加。

补码运算的基本法则

- 均用补码表示
- 数符作为数的最高位参与运算
- 将减数变补，用求和代替求差
- 运算结果用补码表示

设$[X]_{补}=0.1011$，$[Y]_{补}=1.1110$，求$[X+Y]_{补}$和$[X-Y]_{补}$的值。

设$[X]_{补}=0.1011$，$[Y]_{补}=1.1110$，求$[X+Y]_{补}$和$[X-Y]_{补}$的值。

$[X]_{补} =0.1011$
$[Y]_{补} = 1.1110$
$[-Y]_{补}= 0.0010$

$[X+Y]_{补} = [X]_{补}+ [Y]_{补}= 0.1011+1.1110= 0.1001$

$[X-Y]_{补}= [X]_{补}+ [-Y]_{补}= 0.1011+0.0010= 0.1101$

溢出判断

➢ 已知机器数$X_4,X_3X_2X_1X_0$，请使用补码的加减法公式，计算如下各算式：

$9+5=14$

$$\begin{array}{r} 01001 \\ +\ 00101 \\ \hline 01110 \end{array}$$

$(-9)+(-5)=-14$

$$\begin{array}{r} 10111 \\ +\ 11011 \\ \hline \underline{1}10010 \end{array}$$

$14-1=13$

$$\begin{array}{r} 01110 \\ +\ 11111 \\ \hline \underline{1}01101 \end{array}$$

$-14+1=-13$

$$\begin{array}{r} 10010 \\ +\ 00001 \\ \hline 10011 \end{array}$$

$12+7=19$（溢出）

$$\begin{array}{r} 01100 \\ +\ 00111 \\ \hline 10011 \end{array}$$

$(-12)+(-7)=-19$（溢出）

$$\begin{array}{r} 10100 \\ +\ 11001 \\ \hline \underline{1}01101 \end{array}$$

变形补码

- 符号相同的两个数（补码形式）相加时，如果结果的符号与加数/被加数的符号位不相同，则溢出。
- 当以单符号位两数相加（补码形式）时，如果数值高位的进位≠符号位的进位时，则溢出。
- 采用双符号（fs1和fs2）位（变形补码或模4补码）时，正数的双符号位为00，负数的双符号位为11。符号位参与运算，当结果的两个符号位不相同时（10或01），则溢出。

$9+5=14$

```
   0 0 1 0 0 1
+  0 0 0 1 0 1
--------------
   0 0 1 1 1 0
```

$f_{S1}=f_{S2}$ 不溢出

$12+7=19$

```
   0 0 1 1 0 0
+  0 0 0 1 1 1
--------------
   0 1 0 0 1 1
```

$f_{S1}\neq f_{S2}$ 溢出

$x_4, x_3x_2x_1x_0$

$\Downarrow$

$f_{s2}\,f_{s1}, x_3x_2x_1x_0$

$(x_5, x_4x_3x_2x_1x_0)$

X = 0.1001，Y=0.1100，用变形补码求X+Y的值，并判断结果是否溢出。

X = 0.1001，Y=0.1100，用变形补码求X+Y的值，并判断结果是否溢出。

$$[X+Y]_{补} = [X]_{补} + [Y]_{补} = 00.1001$$

$$+\ 00.1100$$

$$= 01.0101$$ 溢出

移位

- 算术移位

保持该数的符号不变，而数量上则发生变化，实现$\times 2^n$和$\div 2^n$的操作。

（1）原码表示的正负数的移位规则：符号位保持不变，空位补0。

①左移：符号位不变，尾数最低位补0

②右移：符号位不变，最高有效位补0

（2）补码表示的正数的移位规则同上

补码表示的负数的移位规则：

①左移同原码

②右移，符号位不变，最高位有效位补"1"

移位

- 逻辑移位只有数码位置的变化而无数量概念的变化

$$[x+y]_{移}=[x]_{移}+[\mathrm{y}]_{补}$$

$$[x-y]_{移}=[x]_{移}-[\mathrm{y}]_{补}=[x]_{移}+[-\mathrm{y}]_{补}$$

总结

$$[x]_{补}+[y]_{补}=[x+y]_{补} \qquad (\mathrm{mod}\ 2^{n+1}/2)$$

$$[x-y]_{补}=[x+(-y)]_{补}=[x]_{补}+[-y]_{补}=[x]_{补}-[y]_{补} \qquad (\mathrm{mod}\ 2^{n+1}/2)$$

变补是连同符号位 一起变反，末尾加“1”，目的是变减为加。

溢出判断---通过变形补码，用双符号位判断

$$[x+y]_{移}=[x]_{移}+[\mathrm{y}]_{补}$$

$$[x-y]_{移}=[x]_{移}-[\mathrm{y}]_{补}=[x]_{移}+[-\mathrm{y}]_{补}$$

课后习题：P68 3.14 3.15 3.16

有问题欢迎随时跟我讨论

办公地点：西校区信息馆423

邮　　箱：gddu@ysu.edu.cn