计算机组成原理

PRINCIPLES OF COMPUTER ORGANIZATION

第9次课：3.5.1 浮点数的加减法运算

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

燕山大學
YANSHAN UNIVERSITY

课程目标

- 掌握浮点数加减法运算操作流程；
- 熟悉规格化操作与溢出处理；
- 了解浮点数加减法运算的硬件。

浮点数的加减法运算

- 设两个浮点：$X=M_x \cdot 2^{Ex}$，$Y=M_y \cdot 2^{Ey}$
- 实现X±Y运算的法则：
  - 对阶
  - 尾数加(减)运算
  - 规格化
  - 舍入处理
  - 判溢出

小阶对大阶：

小阶码＋1，尾数右移1位，直到增大到与大阶码相同。

浮点数的加减法运算

- 设两个浮点：$X=M_x \cdot 2^{Ex}$，$Y=M_y \cdot 2^{Ey}$
- 实现X±Y运算的法则：
  - 对阶
  - 尾数加(减)运算
  - 规格化
  - 舍入处理
  - 判溢出

- **加法；**
- **减法：减数的符号取反（求补），与被减数相加。**

浮点数的加减法运算

- 设两个浮点：$X=M_x \cdot 2^{Ex}$，$Y=M_y \cdot 2^{Ey}$
- 实现X±Y运算的法则：
  - 对阶
  - 尾数加(减)运算
  - 规格化
    - 左规
    - 右规
  - 舍入处理
  - 判溢出

运算结果尾数为：

- 11.1xx…x 或 00.0xx…x：
  尾数每左移1位，阶码减1，直到使尾数成为规格化数为止。
- 阶码减1，必须同时判断是否下溢。若发生下溢，可认为结果为0。

浮点数的加减法运算

- 设两个浮点：$X=M_x \cdot 2^{Ex}$，$Y=M_y \cdot 2^{Ey}$
- 实现X±Y运算的法则：
  - 对阶
  - 尾数加(减)运算
  - 规格化
    - 左规
    - 右规
  - 舍入处理
  - 判溢出

若结果（尾数）发生溢出，即

结果出现10.XX…X 或

01.XX…X 时，

- 尾数右移1位，阶码加1。右规最多1次。
- 阶码加1，必须同时判断是否上溢。若发生上溢，可认为结果为∞。

浮点数的加减法运算

- 设两个浮点：$X=M_x \cdot 2^{Ex}$，$Y=M_y \cdot 2^{Ey}$
- 实现X±Y运算的法则：
  - 对阶
  - 尾数加(减)运算
  - 规格化
  - 舍入处理
    - 截(尾)断法
    - 末位恒置1法
    - 0舍1入法
  - 判溢出

对阶、规格化时，右移操作后，需进行舍入处理。

浮点数的加减法运算

- 设两个浮点：$X=M_x \cdot 2^{Ex}$，$Y=M_y \cdot 2^{Ey}$
- 实现X±Y运算的法则：
  - 对阶
  - 尾数加(减)运算
  - 规格化
  - 舍入处理
    - 截(尾)断法
    - 末位恒置1法
    - 0舍1入法
  - 判溢出

将需丢弃的尾数低位丢弃。

浮点数的加减法运算

- 设两个浮点：$X=M_x \cdot 2^{Ex}$，$Y=M_y \cdot 2^{Ey}$
- 实现X±Y运算的法则：
  - 对阶
  - 尾数加(减)运算
  - 规格化
  - 舍入处理
    - 截(尾)断法
    - 末位恒置1法
    - 0舍1入法
  - 判溢出

使要保留的尾数的最低位永远为1。

浮点数的加减法运算

- 设两个浮点：$X=M_x \cdot 2^{Ex}$，$Y=M_y \cdot 2^{Ey}$
- 实现X±Y运算的法则：
  - 对阶
  - 尾数加(减)运算
  - 规格化
  - 舍入处理
    - 截(尾)断法
    - 末位恒置1法
    - 0舍1入法
  - 判溢出

- 当尾数右移丢弃的是1时，要保留的尾数最末位加1；
- 当尾数右移丢弃的是0时，要保留的尾数最末位不变。
- 当遇到01.111…11这种需右规的尾数时，采用此法会再次使尾数溢出。遇到这种情况可采用截尾法。

浮点数的加减法运算

- 0舍1入法
  - 当尾数为负数，且用补码表示时，舍入规则：

1.xxx…x|0xx…x

1.xxx…x|100…0 → 舍

1.xxx…x|1xx…x → 入1

符号位　　　不全为0

浮点数的加减法运算

- 0舍1入法

【例】

$[X_1]_{补} = 11.0110\,0000$

不舍不入

$[X_2]_{补} = 11.0110\,0001$

舍

$[X_3]_{补} = 11.0110\,1000$

舍

$[X_4]_{补} = 11.0110\,1001$

入

11.0111

入 00.1001 1111 → 00.1010

入 00.1001 1000 → 00.1010

舍 00.1001 0111 → 00.1001

浮点数的加减法运算

- 溢出判断

以下情况下，可能会导致阶码溢出

– 左规（阶码 - 1）时

  - 左规时：先判断阶码是否为全0，若是，则直接置阶码下溢；否则，阶码减1后判断阶码是否为全0，若是，则阶码下溢。

– 右规（阶码 +1）时

  - 右规（+ 1）时，先判断阶码是否为全1，若是，则直接置阶码上溢；否则，阶码加1后判断阶码是否为全1，若是，则阶码上溢。

浮点数的加减法运算

例题 1：已知两个浮点数，$x=2^{+010}\times(+0.11011011)$，$y=2^{+100}\times(-0.10101100)$，求 $x+y$?

x 和 y 在运算器中的浮点数补码表示形式为：

$[x]_补$: 00 010; 00.11011011

$[y]_补$: 00 100; 11.01010100

例题 1：已知两个浮点数，$x=2^{+010}\times(+0.11011011)$，$y=2^{+100}\times(-0.10101100)$，求 $x+y$?

x 和 y 在运算器中的浮点数补码表示形式为：

$[x]_补$：00 010; 00.11011011

$[y]_补$：00 100; 11.01010100

(1) 对阶：

$$\Delta E=[E_x]_补-[E_y]_补=[E_x]_补+[-E_y]_补=00010+11100=11110=(-2)_{10}$$

x 的阶码较小，保留 y 的阶码，即 E=00 100;

x 的尾数右移 2 位，即 00.00110110(11)——括号中为附加位;

$[x]'_补$：00 100; 00.00110110(11)

(2) 尾数求和：

$$M=[M'_x]_补+[M_y]_补=00.00110110(11)+11.01010100=11.10001010(11)$$

(3) M 规格化：

根据(2)中尾数求和 M 的双符号位可知，需要左规（***M* 不溢出，双符号位为 00.0 或 11.1，此时可能需要多次(*K* 次)左规，才能形成 11.0xxxxx 或者 00.1xxxxx 的规格化形式**），在此题中，K 取 1，即小数点往右走一位(附加位的数值可以被利用)，故可得规格化的尾数 M'=11.00010101(1)。与此同时，阶码需要减 1，即 $E'=[E]_补+[-1]_补$=00 100+11 111=00 011。

(4) M 舍入处理：

根据(3)中 M' 的附加位最高位为 1 可知，需要进行舍入处理，按照 0 舍 1 入的原则，在 M' 的最低位加 1，可得 M''=11.00010110。

(5) 判断溢出：

由(4)中 M'' 可知，M'' 无溢出，且已规格化。与此同时，阶码 E' 为 00 011，无溢出。故 $[x+y]_补$=00 011; 11.00010110。

求两浮点数的和、差。

$X = 0.110101 \times 2^{-010}$； $Y = -0.101010 \times 2^{-001}$。

- 求两浮点数的和、差。

$X = 0.110101 \times 2^{-010}$； $Y = -0.101010 \times 2^{-001}$。

【解】

两数可表示为：

$[X]_{浮} = 11\ 110;\ 00.110101$

$[Y]_{浮} = 11\ 111;\ 11.010110$

① 对阶

② 尾数求和/差

③ 规格化

④ 舍入处理

浮点数的加减法运算

$[X]_{浮}=11\ 110;\ 00.110101$
$[Y]_{浮}=11\ 111;\ 11.010110$

- 求两浮点数的和、差。

$X=0.110101\times2^{-010}$；$Y=-0.101010\times2^{-001}$。

① 对阶

求阶差：$[-E_Y]_{补}=00\ 001$

$[\triangle E]_{补}=[E_X]_{补}+[-E_Y]_{补}=11\ 110+00\ 001=11\ 111<0$

X的阶码比Y的阶码小。

X尾数右移一位，使两者阶码相同。这时的X为：

$[X]'_{浮}=11\ 111;\ 00.011010(1)$

浮点数的加减法运算

$[X]_{浮}=11\ 110;\ 00.110101$

$[Y]_{浮}=11\ 111;\ 11.010110$

- 求两浮点数的和、差。

$X=0.110101\times 2^{-010}$； $Y=-0.101010\times 2^{-001}$。

② 尾数求和/差：

$$\begin{array}{r} 00.011010(1) \\ +11.010110 \\ \hline 11.110000(1) \end{array} \qquad \begin{array}{r} 00.011010(1) \\ +00.101010 \\ \hline 01.000100(1) \end{array}$$

浮点数的加减法运算

$[X]_{浮}=11\ 110;\ 00.110101$

$[Y]_{浮}=11\ 111;\ 11.010110$

- 求两浮点数的和、差。

$X=0.110101\times 2^{-010}$； $Y=-0.101010\times 2^{-001}$。

$$\begin{array}{r} 00.011010(1) \\ +11.010110 \\ \hline 11.110000(1) \end{array} \qquad \begin{array}{r} 00.011010(1) \\ +00.101010 \\ \hline 01.000100(1) \end{array}$$

$$\begin{array}{r} 1111 \\ +1110 \\ \hline 1101 \end{array}$$

左规：

尾数左移2位，阶码减2，

$\therefore\ [X+Y]_{浮}$

$=1101;\ 11.000010$

右规：

尾数右移1位，阶码加1，

$\therefore\ [X-Y]_{浮}$

$=0000;\ 00.100010(01)$

浮点数的加减法运算

$[X]_{浮}=11\ 110;\ 00.110101$

$[Y]_{浮}=11\ 111;\ 11.010110$

➢ 求两浮点数的和、差。

$X=0.110101\times 2^{-010}$； $Y=-0.101010\times 2^{-001}$。

④ 舍入处理：

$$\begin{array}{r} 00.011010(1) \\ +11.010110 \\ \hline 11.110000(1) \end{array} \qquad \begin{array}{r} 00.011010(1) \\ +00.101010 \\ \hline 01.000100(1) \end{array}$$

$$\begin{array}{r} 11111 \\ +11110 \\ \hline 11101 \end{array}$$

左规：
尾数左移2位，阶码减2，
$\therefore [X+Y]_{浮}$
$=1101;\ 11.000010$

右规：
尾数右移1位，阶码加1，
$\therefore [X-Y]_{浮}$
$=0000;\ 00.100010\cancel{(01)}$

舍

⑤判溢出：

$X+Y=-0.000010\times 2^{-011}$ $X-Y=0.100010\times 2^{000}$

浮点数乘法

- 设 X、Y、Z 为浮点数，

$X = M_x \cdot 2^{Ex}$

$Y = M_y \cdot 2^{Ey}$

$Z = X \times Y = (M_x \cdot M_y) 2^{Ex+Ey}$

- 两浮点数相乘之积的
  - 阶码为两乘数阶码之和
  - 尾数为两乘数尾数之积

浮点数乘法

浮点乘法的运算过程：

① 两乘数一定是规格化数。
若有一个乘数为0，则乘积必为0。

② 求乘积的阶码： $E_z = E_x + E_y$；
判断积的阶码是否溢出：上溢、下溢。

③ 求乘积的尾数：两乘数的尾数相乘。

④ 规格化乘积的尾数。

- 若尾数为*n*位补码（含1位符号），则
规格化正数范围：$+1/2 \sim +(1-2^{-(n-1)})$；
规格化负数范围：$-1 \sim -(1/2+2^{-(n-1)})$。
- ∵ |两规格化数之积|≥1/4,
∴ 积的尾数若需左规，只需1次左移。
- 积有可能为＋1，需右规。
只需1次右移，并采用某种舍入算法。

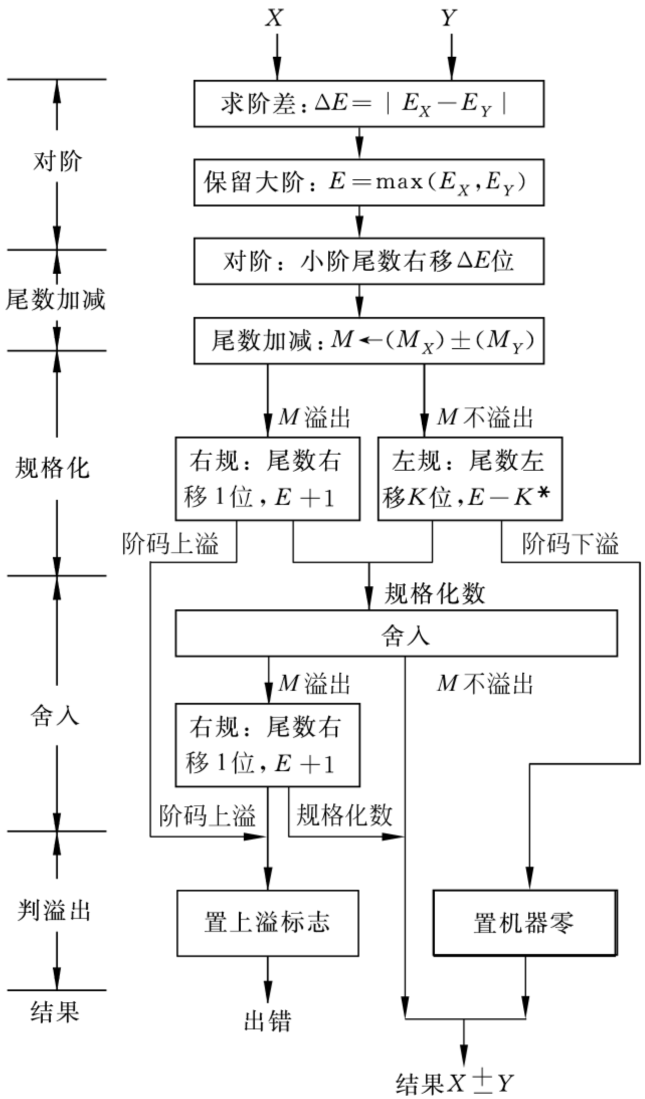
0操作数检查

阶码加

尾数乘

结果规格化、舍入

总结



*如果已为规格化数，则K＝0，不移位。

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