3.5.2 浮点数的加法运算

刘 芳 副教授

国防科学技术大学计算机学院

十进制科学计数法的加法实例：

$A=1.23 \times 10^5$；$B=4.56 \times 10^2$；求 A+B

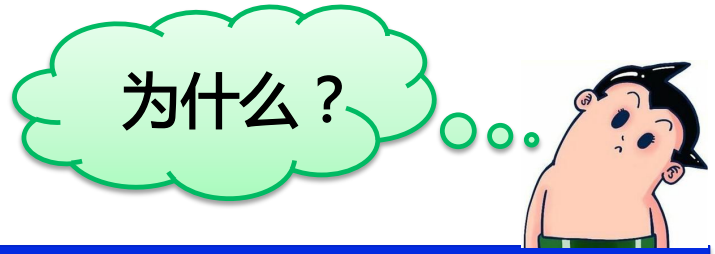
其计算过程为：

$$1.23 \times 10^5 + 4.56 \times 10^2$$
$$= 1.23 \times 10^5 + 0.00456 \times 10^5$$
$$= (1.23 + 0.00456) \times 10^5 = 1.23456 \times 10^5$$

进行尾数加法运算前，必须“对阶”！

对阶：目的是使两个操作数的阶码相等(对齐小数点)

- 规则： 小阶向大阶看齐，阶小的数的尾数右移，右移位数等于两个阶码差的绝对值。



IEEE754尾数右移时，需要注意的是什么？

1、要将隐含的“1”移到小数部分，空出位补0；

2、移出的低位保留到特定的“附加位”上

两个规格化浮点数分别为A和B

- $A=M_a \cdot 2^{Ea}$ ，$B=M_b \cdot 2^{Eb}$
- 假设$E_a \geq E_b$

浮点数加法步骤

- 求阶差　　$E_a - E_b$
- 对阶　　$Mb \cdot 2^{-(Ea-Eb)}$
- 尾数相加　　$M_a + M_b \cdot 2^{-(Ea-Eb)}$
- 结果规格化　$A+B = (M_a + M_b \cdot 2^{-(Ea-Eb)}) \cdot 2^{Ea}$

两个规格化浮点数分别为A和B

- $A=Ma \cdot 2^{Ea}$ ，$B=Mb \cdot 2^{Eb}$
- 假设Ea≥Eb

$$A \pm B = (Ma \pm Mb \cdot 2^{-(Ea-Eb)}) \cdot 2^{Ea}$$

$$A \times B = (Ma \times Mb) \cdot 2^{Ea+Eb}$$

$$A \div B = (Ma \div Mb) \cdot 2^{Ea-Eb}$$

3.5.2 浮点数加法

自然世界中的浮点数

计算机世界中的浮点数

无穷位

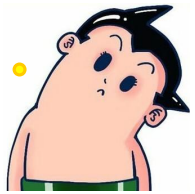
超出表示范围：溢出！

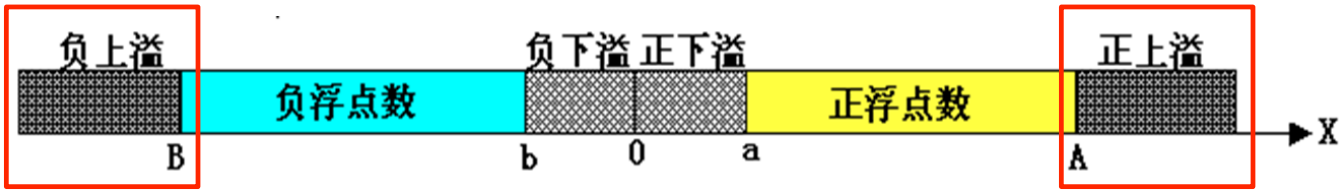
有限位

溢出

- 阶码上溢：超过了阶码可以表示的最大允许值，一般将其认为是 +∞和 -∞
- 阶码下溢
- 尾数上溢
- 尾数下溢

SP最大允许的阶码为多少？？ 127！



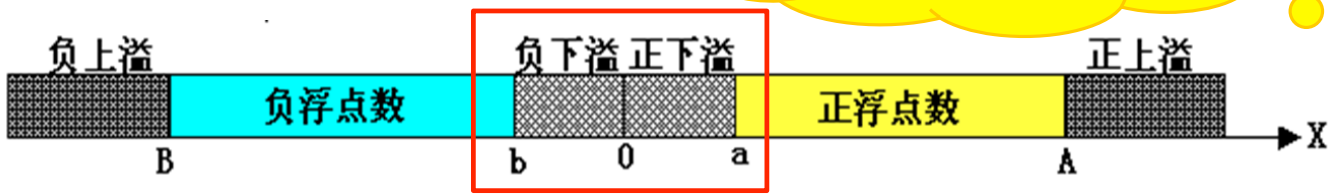


A—最大正数　　a—最小正数
B—最小负数　　b—最大负数

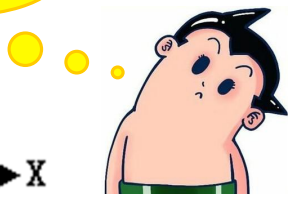
- 溢出
  - 阶码上溢
  - 阶码下溢
  - 尾数上溢
  - 尾数下溢

超过了阶码可能表示的最小允许值，一般将其认为是0

SP最大允许的阶码为多少？？-126！



A—最大正数　a—最小正数
B—最小负数　b—最大负数



3.5.2 浮点数加法

- 溢出
  - 阶码上溢
  - 阶码下溢
  - 尾数上溢
  - 尾数下溢

两个同/异符号尾数相加/减，
最高有效位产生了进位

尾数溢出：

- 不一定浮点数溢出，即不一定会发生“异常”
- 右规：将尾数右移，阶码增1来重新对齐

$$
\begin{aligned}
&1.010\ldots0_2\times2^2\\
&+1.100\ldots0_2\times2^2\\
\hline
&=10.110\ldots0_2\times2^2\\
&=>1.0110\ldots0_2\times2^3
\end{aligned}
$$

3.5.2 浮点数加法

- 溢出
  - 阶码上溢
  - 阶码下溢
  - 尾数上溢
  - 尾数下溢

在将尾数右移时，尾数的最低有效位从尾数域右端移出去，丢失了有效信息

1、进行舍入处理

2、在运算过程中，添加保护位

浮点数加法步骤

- 求阶差
- 对阶
- 尾数相加
- 规格化并判溢出
- 舍入
- 置0

浮点数加法步骤

- 求阶差
- 对阶
- 尾数相加
- 规格化并判溢出
- 舍入
- 置0

如果尾数比规定位数长，需考虑舍入

浮点数加法步骤

- 求阶差
- 对阶
- 尾数相加
- 规格化并判溢出
- 舍入
- 置0

尾数为0说明结果也为0，根据IEEE754，阶码和尾数全为0

IEEE 754标准的四种舍入方式

就近舍入

- 舍入为最近可表示的数

例：1.110111 ~ 1.1110； 1.110101 ~ 1.1101；

1.110110 ~ 1.1110； 1.111100 ~ 1.1111

附加位为：

11：入

01：舍

10（强制结果为偶数）

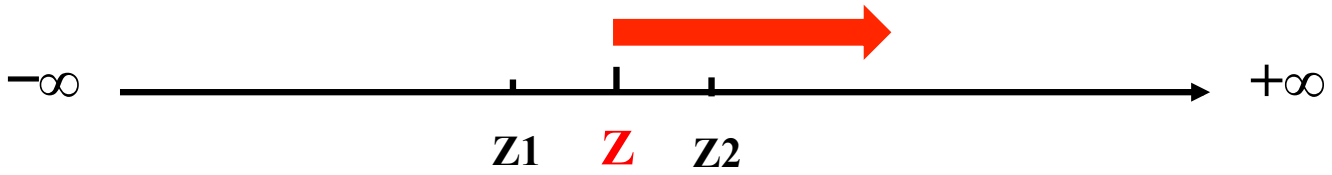
00：保持结果不变

IEEE 754标准的四种舍入方式

就近舍入

朝+∞方向舍入

- 舍入为Z2(正向舍入)



Z1和Z2分别是结果Z的最近可表示的左、右数

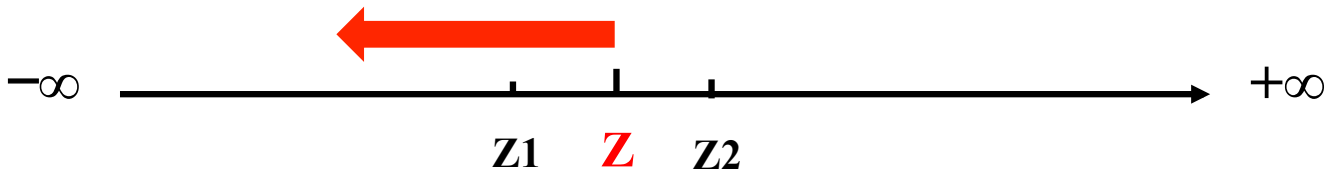
IEEE 754标准的四种舍入方式

就近舍入

朝+∞方向舍入

朝-∞方向舍入

- 舍入为Z1(负向舍入)



Z1和Z2分别是结果Z的最近可表示的左、右数

IEEE 754标准的四种舍入方式

就近舍入

朝+∞方向舍入

朝-∞方向舍入

朝0方向舍入

- 总是舍入成Z1与Z2中绝对值较小的数 (更接近于0)

Z1和Z2分别是结果Z的最近可表示的左、右数

浮点数加法举例

用IEEE 754单精度形式，求出浮点数 $X=0.5_{10}$ 与 $Y=-0.4375_{10}$ 之和

解：　$0.5_{10}=1/2_{10}=0.1_2=(1.00\ldots0_2)\times2^{-1}$

$-0.4375_{10}=-7/16_{10}=-0.0111_2=-1.110_2\times2^{-2}$

$[0.5]_{浮}=0\ 01111110\ 000\ldots0$，$[-0.4375]_{浮}=1\ 01111101\ 110\ldots0$

浮点数加法举例

用IEEE 754单精度形式，求出浮点数 $X=0.5_{10}$ 与 $Y=-0.4375_{10}$ 之和

解： $0.5_{10}=1/2_{10}=0.1_2=(1.00\ldots0_2)\times2^{-1}$

$-0.4375_{10}=-7/16_{10}=-0.0111_2=-1.110_2\times2^{-2}$

$[0.5]_{浮}=0\ 01111110\ 000\ldots0$，$[-0.4375]_{浮}=1\ 01111101\ 110\ldots0$

对阶（求阶差）：

$$[\Delta E]_{补}=[Ex]_{移}+[-[Ey]_{移}]_{补}\quad(\text{mod }256)$$

$[\Delta E]_{补}=0111\ 1110+1000\ 0011=0000\ 0001$，$\Delta E=1$

所以：对Y进行对阶，$[Y]_{浮}=1\ \underline{0111\ 1110}\ \ \underline{1110\ldots0}$

浮点数加法举例

用IEEE 754单精度形式，求出浮点数 $X=0.5_{10}$ 与 $Y=-0.4375_{10}$ 之和

解： $0.5_{10}=1/2_{10}=0.1_2=(1.00\ldots0_2)\times2^{-1}$

$-0.4375_{10}=-7/16_{10}=-0.0111_2=-1.110_2\times2^{-2}$

$[0.5]_{浮}=0\ 01111110\ 000\ldots0$，$[-0.4375]_{浮}=1\ 01111101\ 110\ldots0$

对阶（求阶差）：$[Y]_{浮}=1\ \underline{0111\ 1110}\ \underline{1110\ldots0}$

尾数相加：01.0000…0+(10.1110…0)=00.00100…0

两个异号的原码数加法
（尾数为原码表示，最左边一位为符号位）

浮点数加法举例

用IEEE 754单精度形式，求出浮点数 $X=0.5_{10}$ 与 $Y=-0.4375_{10}$ 之和

解： $0.5_{10}=1/2_{10}=0.1_2=(1.00\ldots0_2)\times2^{-1}$

$-0.4375_{10}=-7/16_{10}=-0.0111_2=-1.110_2\times2^{-2}$

$[0.5]_{浮}=0\ 01111110\ 000\ldots0$，$[-0.4375]_{浮}=1\ 01111101\ 110\ldots0$

对阶（求阶差）：$[Y]_{浮}=1\ \underline{0111\ 1110}\ \underline{1110\ldots0}$

尾数相加：$01.0000\ldots0+(10.1110\ldots0)=00.00100\ldots0$

规格化和判溢出：$+(0.00100\ldots0)_2\times2^{-1}=+(1.00\ldots0)_2\times2^{-4}$ (阶码减3)

因为127≥-4 ≥ -126，没有溢出

浮点数加法举例

用IEEE 754单精度形式，求出浮点数 $X=0.5_{10}$ 与 $Y=-0.4375_{10}$ 之和

解： $0.5_{10}=1/2_{10}=0.1_2=(1.00\ldots0_2)\times2^{-1}$

$-0.4375_{10}=-7/16_{10}=-0.0111_2=-1.110_2\times2^{-2}$

$[0.5]_{浮}=0\ 01111110\ 000\ldots0$，$[-0.4375]_{浮}=1\ 01111101\ 110\ldots0$

对阶（求阶差）：$[Y]_{浮}=1\ \underline{0111\ 1110}\ \underline{1110\ldots0}$

尾数相加：$01.0000\ldots0+(10.1110\ldots0)=00.00100\ldots0$

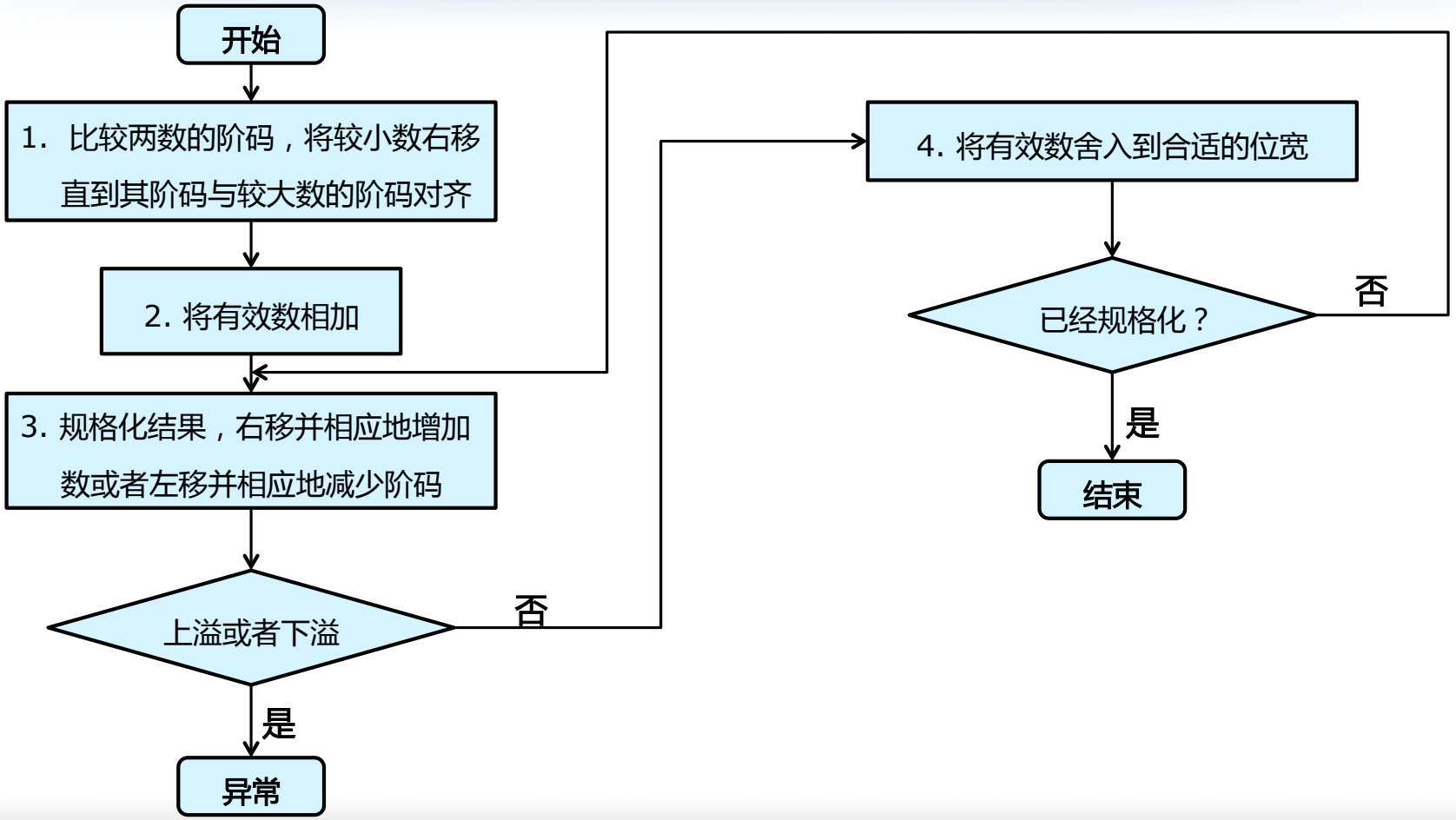
规格化和判溢出：$+(0.00100\ldots0)_2\times2^{-1}=+(1.00\ldots0)_2\times2^{-4}$ (阶码减3)

因为127≥-4 ≥ -126，没有溢出

舍入：无需舍入

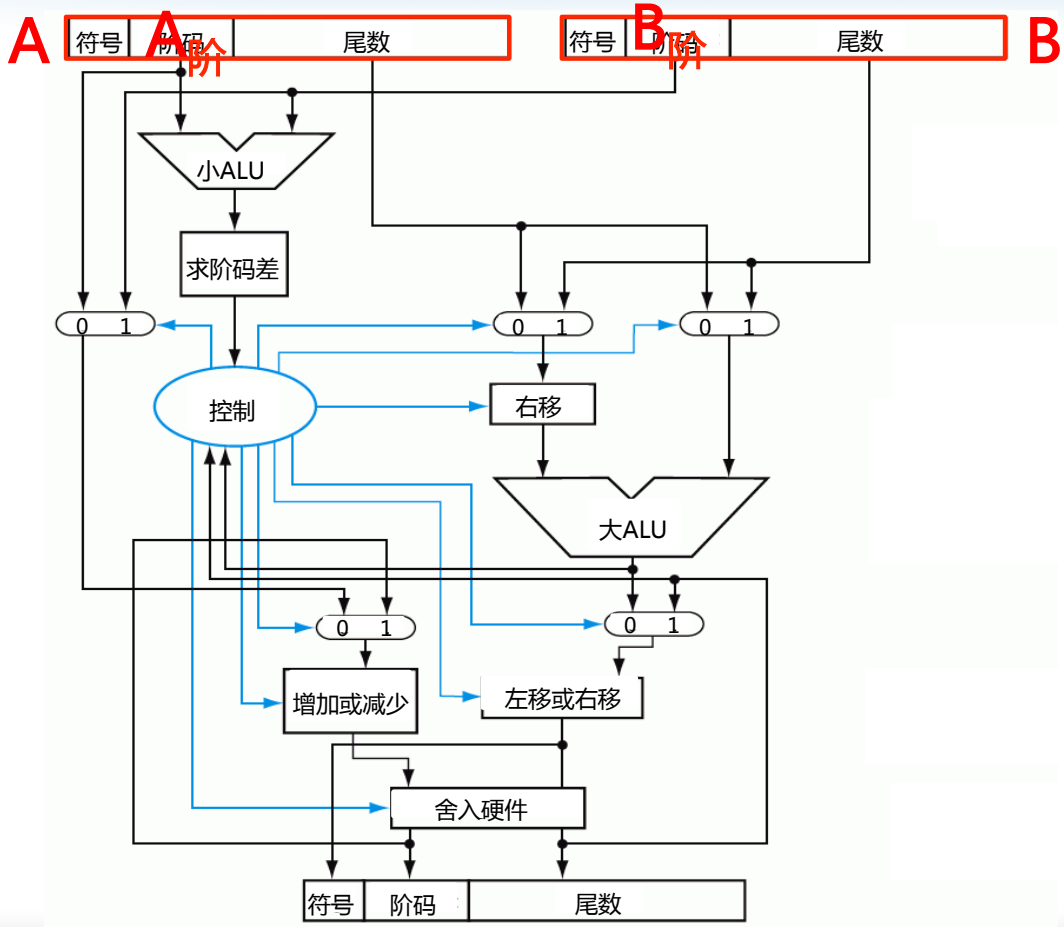
结果为：$(1.00\ldots0)_2\times2^{-4}$ 即 1/16=0.0625

浮点数加法算法


开始
1. 比较两数的阶码，将较小数右移直到其阶码与较大数的阶码对齐
2. 将有效数相加
3. 规格化结果，右移并相应地增加数或者左移并相应地减少阶码
上溢或者下溢
否
是
异常
4. 将有效数舍入到合适的位宽
已经规格化？
否
是
结束

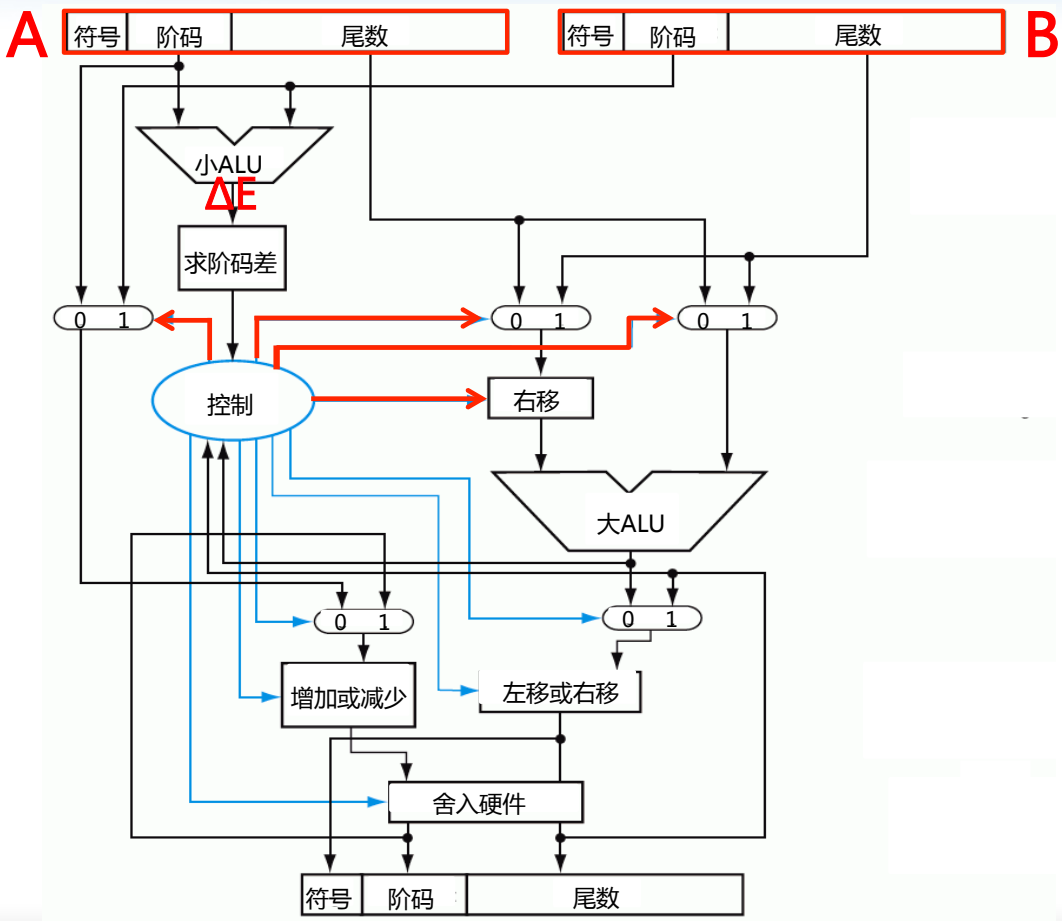
浮点数加法运算的硬件逻辑结构图

比较阶码

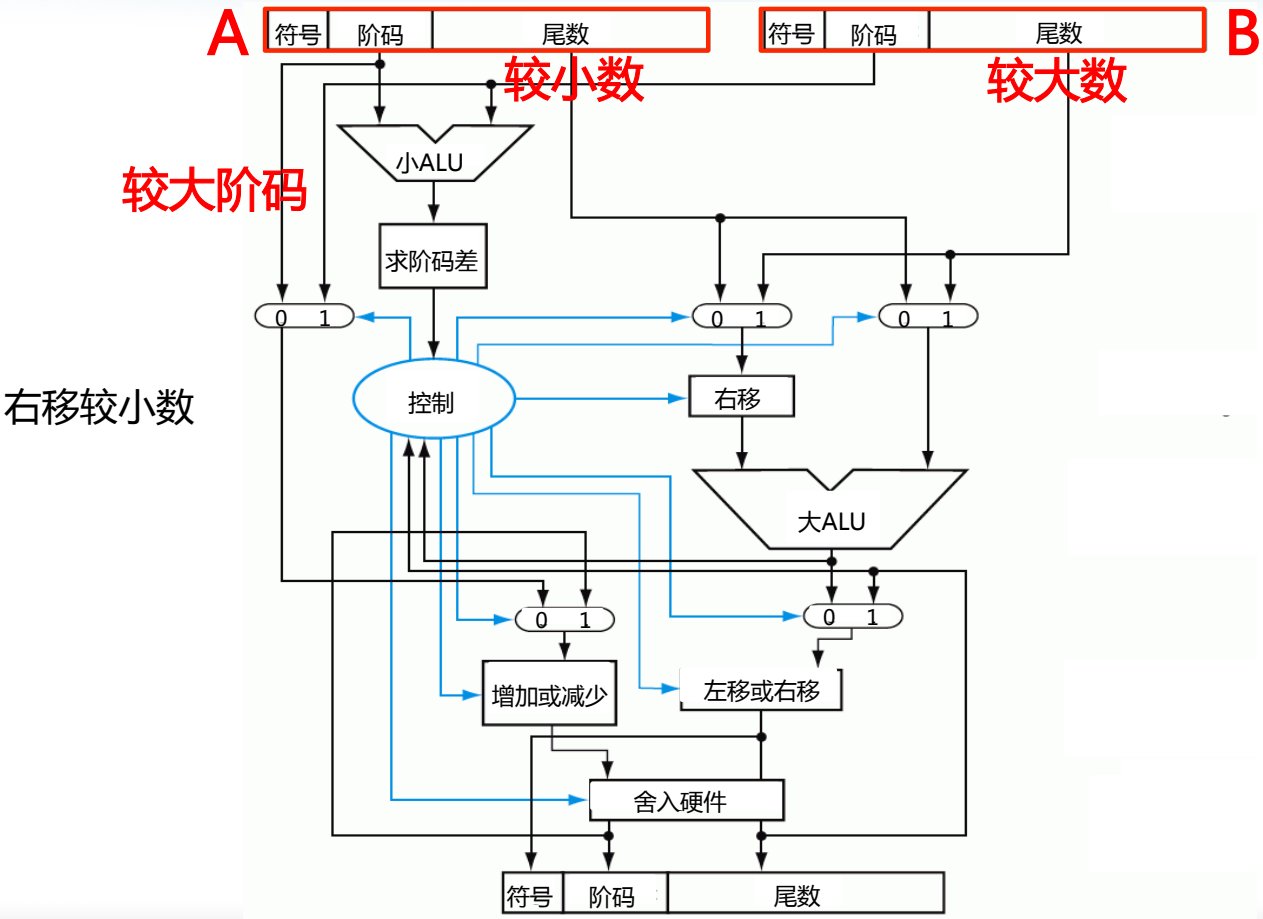


浮点数加法运算的硬件逻辑结构图

右移较小数

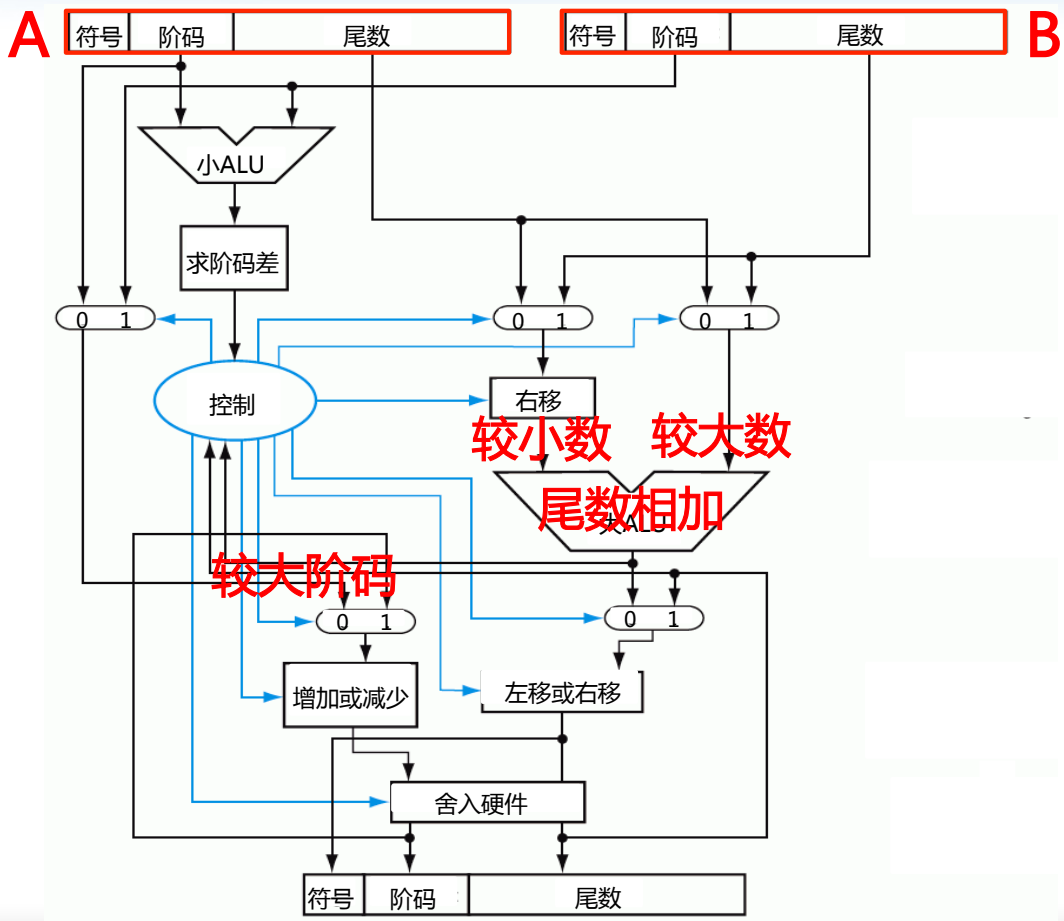


浮点数加法运算的硬件逻辑结构图

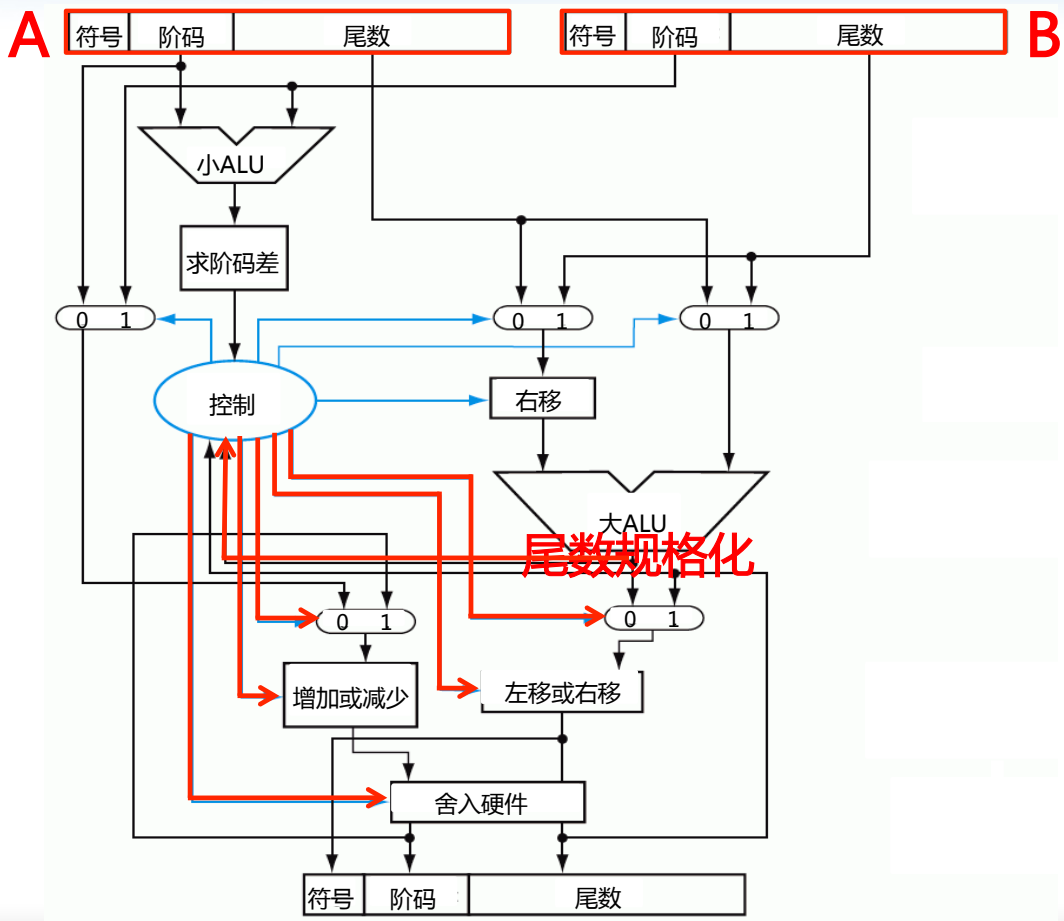


浮点数加法运算的硬件逻辑结构图

尾数相加

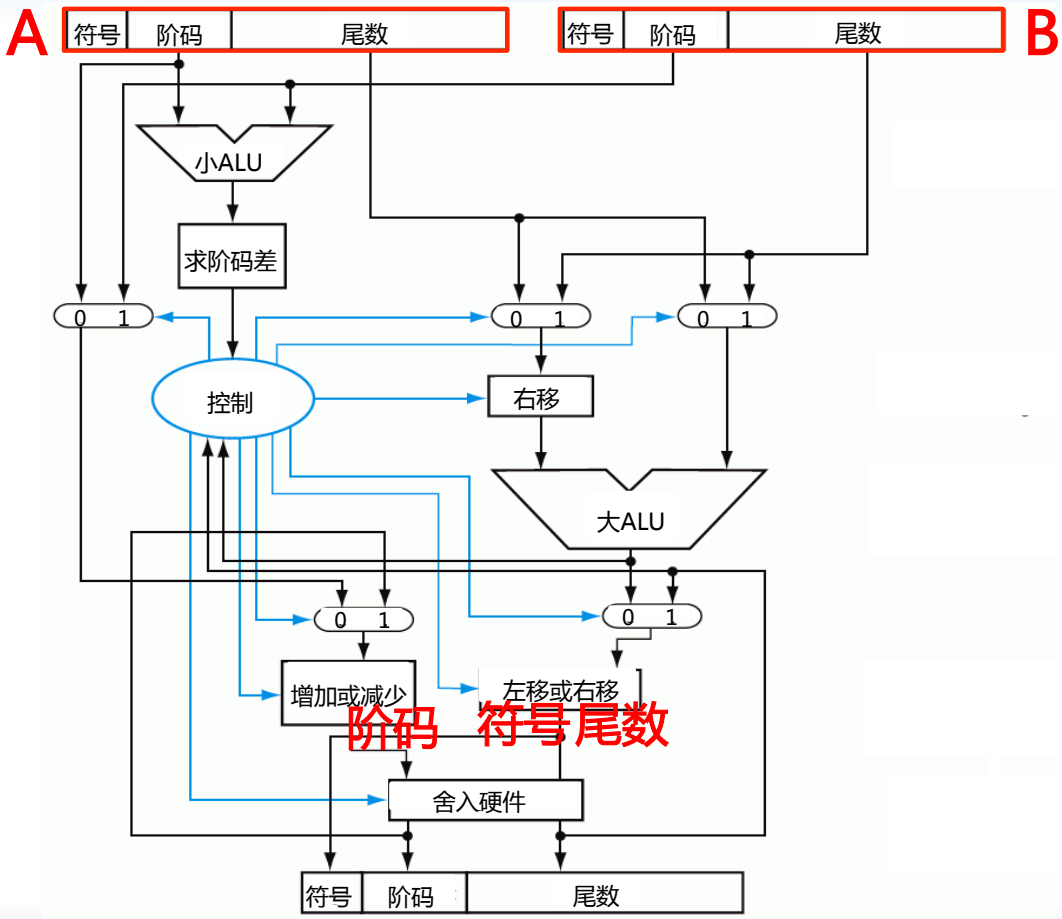


浮点数加法运算的硬件逻辑结构图



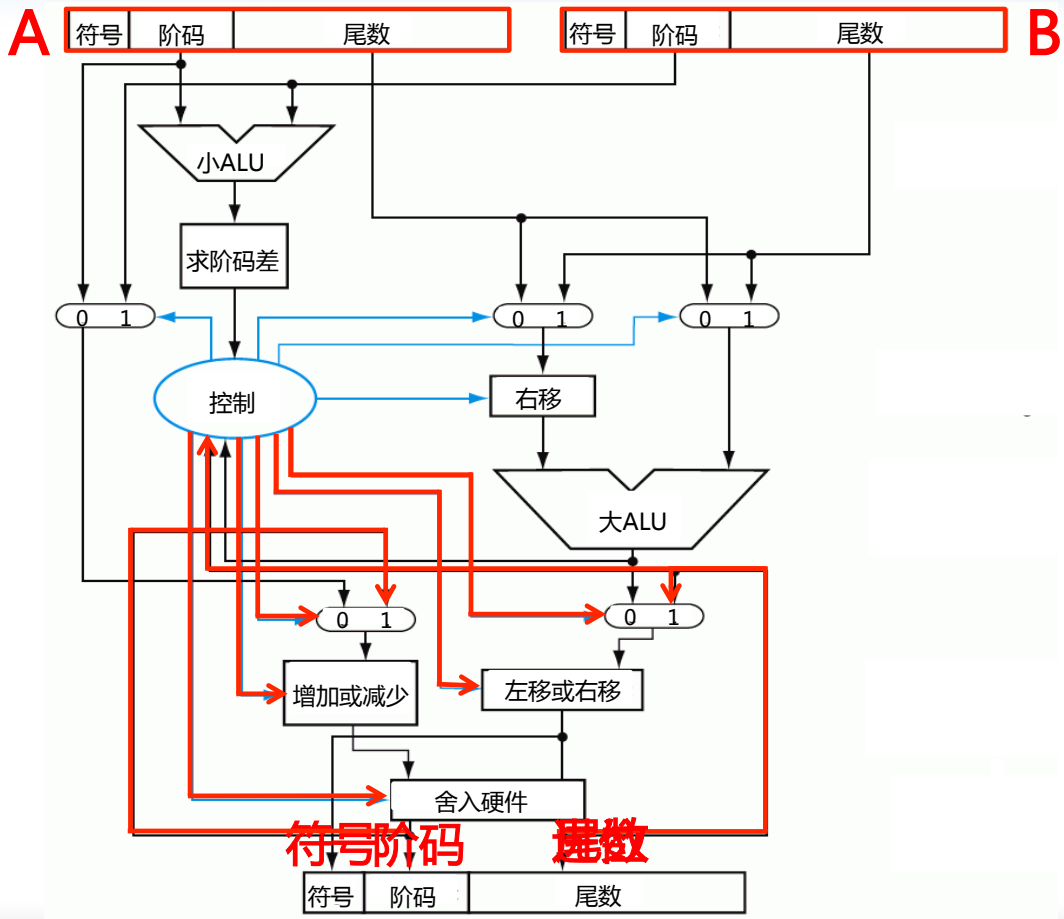
规格化

浮点数加法运算的硬件逻辑结构图



舍入

浮点数加法运算的硬件逻辑结构图



舍入