计算机组成原理

授课老师：吴炜滨

大纲

- 浮点四则运算
  - 浮点加减运算

浮点加减运算

- 浮点数的加减运算
  - 对阶
  - 尾数求和
  - 规格化
  - 舍入
  - 溢出判断

浮点加减运算

- 浮点数的一般形式： $N = S \times r^j$
  - S：尾数，j：阶码，r：尾数的基值
  - 计算机中：r 取 2、4、8、16 等；S：小数定点形式表示，绝对值小于等于1，可正可负；j：整数，可正可负
- 浮点加减运算
  - 采用补码进行加减法
  - 定点数小数点位置固定且一致，可直接按位加减
  - 浮点数尾数的小数点虽也固定且一致，但两浮点数阶码不同时，实际小数点位置不同
  - 浮点加减不能直接对尾数部分进行加减，需进行对阶，使两数的小数点位置对齐

浮点加减运算

- 浮点数的一般形式： $N = S \times r^j$
  - S：尾数，j：阶码，r：尾数的基值
  - 计算机中：r 取 2、4、8、16 等；S：小数定点形式表示，绝对值小于等于1，可正可负；j： 整数，可正可负
- 浮点加减运算
  - 对阶
    - 使两数的阶码相等
    - 计算机中采用求阶差，并调整阶码和尾数的值的方式，以在保证数据大小不发生变化的前提下，进行运算

浮点加减运算

$$x = S_x \cdot 2^{j_x} \qquad\qquad y = S_y \cdot 2^{j_y}$$

■ 对阶

- 求阶差

$$\Delta j = j_x - j_y = \begin{cases} = 0 & j_x = j_y \quad \text{已对齐} \\ > 0 & j_x > j_y \quad \begin{cases} x\text{向}y\text{看齐} & S_x \leftarrow 1,\ j_x - 1 \\ y\text{向}x\text{看齐} & \checkmark\ S_y \rightarrow 1,\ j_y + 1 \end{cases} \\ < 0 & j_x < j_y \quad \begin{cases} x\text{向}y\text{看齐} & \checkmark\ S_x \rightarrow 1,\ j_x + 1 \\ y\text{向}x\text{看齐} & S_y \leftarrow 1,\ j_y - 1 \end{cases} \end{cases}$$

- 对阶原则
  - 小阶向大阶看齐
    - 因为尾数左移可能导致其数值的高位部分的真值1被移丢，造成数据错误

浮点加减运算

- 设阶码为4位，其中阶符占了2位，尾数为6位，其中尾符占了2位，$x = 0.1101 \times 2^{01}$，$y = (-0.1010) \times 2^{11}$，求$x + y$

解：

$$[x]_{补} = 00{,}01;00.1101 \qquad [y]_{补} = 00{,}11;11.0110$$

- 对阶
  - 求阶差

$$[\Delta j]_{补} = [j_x]_{补} - [j_y]_{补} = \begin{array}{r} 00,01 \\ +\ 11,01 \\ \hline 11,10 \end{array}$$

阶差为负（-2）

$\therefore j_x + 2$，$S_x \rightarrow 2$

  - 对阶

$$[x]'_{补} = 00{,}11;\ 00.0011$$

（发生了舍入）

浮点加减运算

- 设阶码为4位，其中阶符占了2位，尾数为6位，其中尾符占了2位，$x = 0.1101 \times 2^{01}$，$y = (-0.1010) \times 2^{11}$，求$x + y$

解：　$[x]_{补} = 00{,}01; 00.1101$　　$[y]_{补} = 00{,}11; 11.0110$

- 对阶　　$[x]'_{补} = 00{,}11;\ 00.0011$
- 尾数求和

$$
\begin{array}{rl}
[S_x]'_{补} & = 00.0011 \\
+\ [S_y]_{补} & = 11.0110 \\
\hline
 & \ 11.1001
\end{array}
$$

对阶后的$[S_x]'_{补}$

$\therefore [x + y]_{补} = 00{,}11; 11.1001$　　还需进行规格化

浮点加减运算

■ 规格化

- 尽可能地提高在计算机当中浮点数的表示精度，充分利用计算机的硬件资源
- 如何判断一个数据是否是规格化的数据?
- 如何进行规格化?

浮点加减运算

规格化数的定义

- $r = 2$，尾数最高位真值为 1

$$\frac{1}{2} \le |S| < 1$$

- $r = 4$，尾数最高 2 位真值不全为 0

$$\frac{1}{4} \le |S| < 1$$

- $r = 8$，尾数最高 3 位真值不全为 0

$$\frac{1}{8} \le |S| < 1$$

浮点加减运算

■ 规格化数的定义

$$r = 2 \qquad \frac{1}{2} \leq |S| < 1$$

■ 规格化数的硬件判断

$S > 0$	规格化形式	$S < 0$	规格化形式
真值	$0.1 \times\times \cdots \times$	真值	$-0.1 \times\times \cdots \times$
原码	$0.1 \times\times \cdots \times$	原码	$1.1 \times\times \cdots \times$
补码	$0.1 \times\times \cdots \times$	补码	$1.0 \times\times \cdots \times$

浮点加减运算

- **规格化数的硬件判断**
  - 原码：不论正数、负数，第一数位为1
  - 补码：符号位和第一数位不同
  - 硬件上以上述规则为标准来判断是否是规格化的数

$S>0$	规格化形式	$S<0$	规格化形式
真值	$0.1\times\times\cdots\times$	真值	$-0.1\times\times\cdots\times$
原码	$0.\boxed{1}\times\times\cdots\times$	原码	$1.\boxed{1}\times\times\cdots\times$
补码	$\boxed{0.1}\times\times\cdots\times$	补码	$\boxed{1.0}\times\times\cdots\times$

浮点加减运算

■ 规格化数的硬件判断

- 规格化数的定义：$r = 2$ $\qquad \frac{1}{2} \le |S| < 1$
- 规格化数的定义与硬件判断结果不同的特例（对补码而言）
  - 为了便于硬件判断，以硬件判断的结果为准

$$S = -\frac{1}{2} = -0.100 \dots 0$$

$$[S]_{原} = 1.100 \dots 0$$

$$[S]_{补} = \boxed{1.1}00 \dots 0$$

$\therefore [-1/2]_{补}$不是规格化的数

得到该结果时，需进行规格化

$$S = -1$$

$$[S]_{补} = \boxed{1\,.\,0}0\ 0\ 0$$

$\therefore [-1]_{补}$是规格化的数

浮点加减运算

- 当尾数运算结果不是硬件上所判断的规格化数时，需进行规格化
  - 对以双符号位补码来表示的尾数，只有当尾数不溢出，且符号位和第一数位不同时，才是一个硬件上所判断的规格化的数
    - $S>0$时，其补码规格化形式为：$[S]_{补}=00.1\times\times\cdots\times$
    - $S<0$时，其补码规格化形式为：$[S]_{补}=11.0\times\times\cdots\times$

浮点加减运算

- 规格化
  - 左规
    - 当尾数符号位和第一数位相同时，需左规
      - 即尾数出现$00.0\times\times\cdots\times$或$11.1\times\times\cdots\times$时
    - 尾数左移一位，阶码减1，直到尾数不溢出，且数符和第一数位不同为止
  - 右规
    - 当尾数溢出（绝对值> 1）时，需右规
      - 即尾数出现$01.\times\times\cdots\times$或$10.\times\times\cdots\times$时
    - 尾数右移一位，阶码加1，直到尾数不溢出，且数符和第一数位不同为止
    - 最高符号位为真正符号，低位符号位为运算结果溢出部分

浮点加减运算

- 规格化
  - 左规右规进行的都是算术移位
  - 对双符号位补码来说
    - 最高符号位视为真正符号，移位时需保留
    - 低位符号位可视为数值位
  - 数值位空位添补规则
    - 正数补码
      - 左移右移都添0
    - 负数补码
      - 左移 添 0
      - 右移 添 1

浮点加减运算

■ 规格化

$[x+y]_{补} = 00,11;11.1001$　　还需进行规格化

左规后：$[x+y]_{补} = 00,10;11.0010$

$\therefore x+y = (-0.1110)\times 2^{10}$

浮点加减运算

- $x = 0.1101 \times 2^{10}$　$y = 0.1011 \times 2^{01}$，求$x + y$（除阶符、数符取2位外，阶码取3位，尾数取6位）

解：　$[x]_{补} = 00{,}010; 00.110100$　　$[y]_{补} = 00{,}001; 00.101100$

对阶　$[\Delta j]_{补} = [j_x]_{补} - [j_y]_{补}$

$$
\begin{array}{rr}
= & 00,010 \\
+ & 11,111 \\
\hline
 & \boxed{1}00,001
\end{array}
$$

丢掉

阶差为+1　$\therefore j_y + 1,\quad S_y \to 1$

$\therefore [y]'_{补} = 00{,}010; 00.010110$

浮点加减运算

- $x = 0.1101 \times 2^{10}$　$y = 0.1011 \times 2^{01}$，求$x + y$（除阶符、数符取2位外，阶码取3位，尾数取6位）

解：　$[x]_{补} = 00,010; 00.110100$　　$[y]_{补} = 00,001; 00.101100$

对阶　$[y]'_{补} = 00,010; 00.010110$

尾数求和

$$
\begin{array}{rl}
 & [S_x]_{补} = 00.110100 \\
+ & [S_y]'_{补} = 00.010110 \\
\hline
 & 01.001010
\end{array}
$$

对阶后的$[S_y]'_{补}$

尾数溢出需右规

浮点加减运算

- $x = 0.1101 \times 2^{10}$　　$y = 0.1011 \times 2^{01}$，求$x + y$（除阶符、数符外，阶码取3位，尾数取6位）

$$[x + y]_{补} = 00,010; 01.001010$$

尾数溢出需右规

尾数右移一位，阶码加1

$$[x + y]_{补} = 00,011; 00.100101$$

$$\therefore x + y = 0.100101 \times 2^{11}$$

浮点加减运算

舍入

- 在对阶和右规过程中，可能出现尾数末位丢失
  - 引起误差，需考虑舍入来提高尾数的精度
- 截断法
  - 直接将多余位全部舍去
- 0舍1入法
  - 被移去的最高数值位为0，直接舍去
  - 被移去的最高数值位为1，在尾数的末位加1
  - 可能导致尾数又溢出，需再做一次右规
- 恒置“1”法
  - 不管丢掉的数值是多少，都使右移后的尾数末位置为1
    - 如00.1000右移两位：00.0011

浮点加减运算

- $x=\left(-\frac{5}{8}\right)\times 2^{-5}$ $y=\left(\frac{7}{8}\right)\times 2^{-4}$，求$x-y$（除阶符、数符取2位外，阶码取3位，尾数取6位)

解：

$$x=(-0.101000)\times 2^{-101} \qquad y=(0.111000)\times 2^{-100}$$

$$[x]_{补}=11,011;11.011000 \qquad [y]_{补}=11,100;00.111000$$

浮点加减运算

- $x = \left(-\frac{5}{8}\right) \times 2^{-5} \quad y = \left(\frac{7}{8}\right) \times 2^{-4}$，求$x - y$（除阶符、数符取2位外，阶码取3位，尾数取6位)

解：$\quad [x]_{补} = 11{,}011; 11.011000 \qquad [y]_{补} = 11{,}100; 00.111000$

对阶 $\quad [\Delta j]_{补} = [j_x]_{补} - [j_y\ ]_{补} \quad = \ 11, 011$

$$
\begin{array}{r}
11, 011 \\
+\ 00, 100 \\
\hline
11, 111
\end{array}
$$

阶差为-1 $\qquad \therefore j_x + 1, \quad S_x \to 1$

$\therefore [x]'_{补} = 11{,}100; 11.101100$

浮点加减运算

- $x = \left(-\frac{5}{8}\right) \times 2^{-5}$　$y = \left(\frac{7}{8}\right) \times 2^{-4}$，求$x - y$（除阶符、数符取2位外，阶码取3位，尾数取6位）

$[x]_{补} = 11,011; 11.011000$　　$[y]_{补} = 11,100; 00.111000$

$[x]'_{补} = 11,100; 11.101100$

尾数求和

$$
\begin{array}{rr}
 & [S_x]'_{补} = 11.101100 \\
+ & [-S_y]_{补} = 11.001000 \\
\hline
 & \boxed{1}10.110100
\end{array}
$$

丢掉

浮点加减运算

$$[x-y]_{补} = 11,100; 10.110100$$

尾数溢出需右规

尾数右移一位，阶码加1

右规后 $[x-y]_{补} = 11,101; 11.011010$

$$\therefore x-y = (-0.100110) \times 2^{-11} = (-\frac{19}{32}) \times 2^{-3}$$

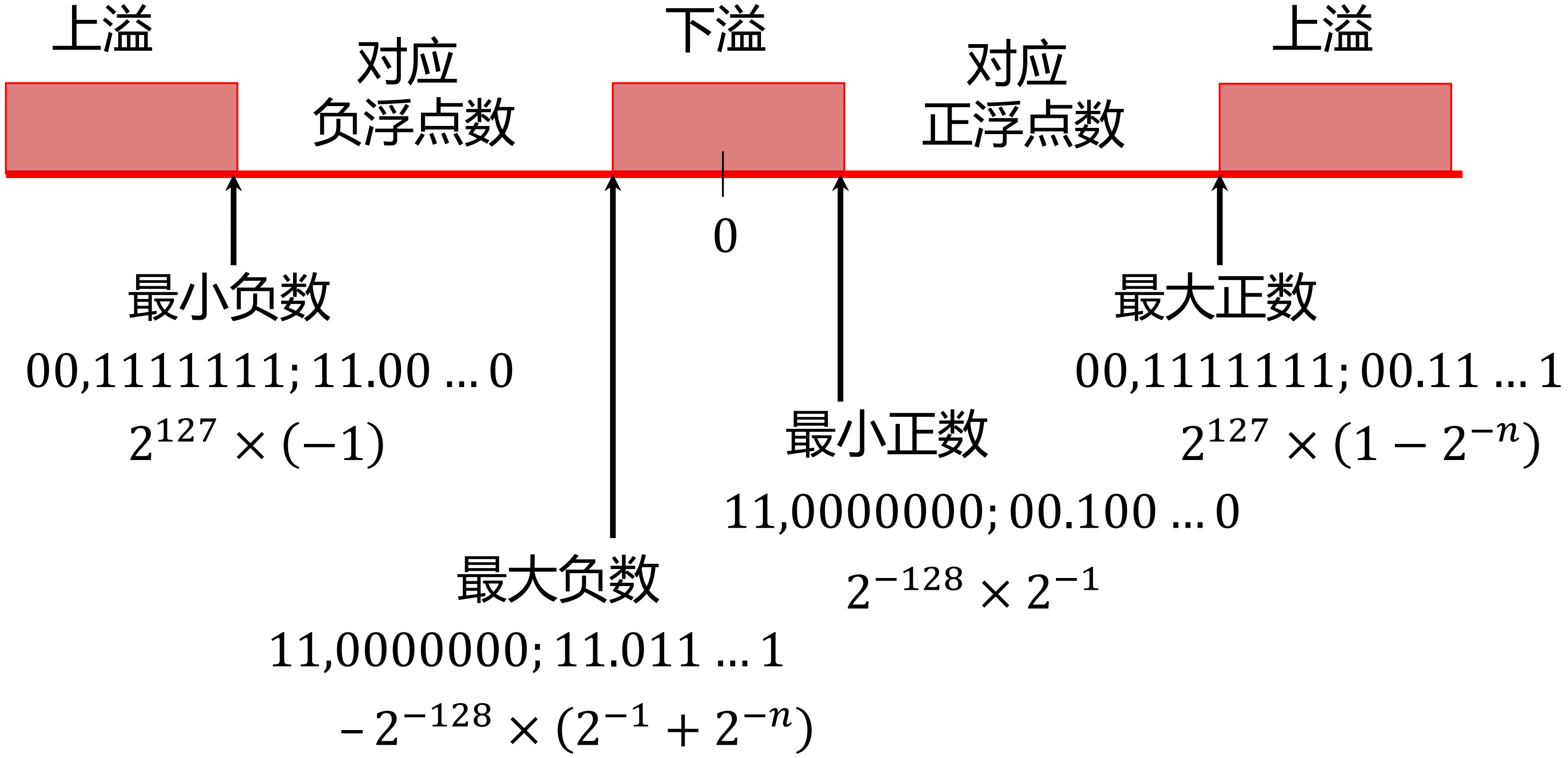
浮点加减运算

- 溢出判断
  - 整个浮点数的值超出了浮点机的表示范围，即发生了溢出
    - 尾数计算结果出现溢出时，并不一定表示发生溢出
    - 需先将尾数规格化，再根据阶码判断浮点运算结果是否溢出

浮点加减运算

■ 溢出判断

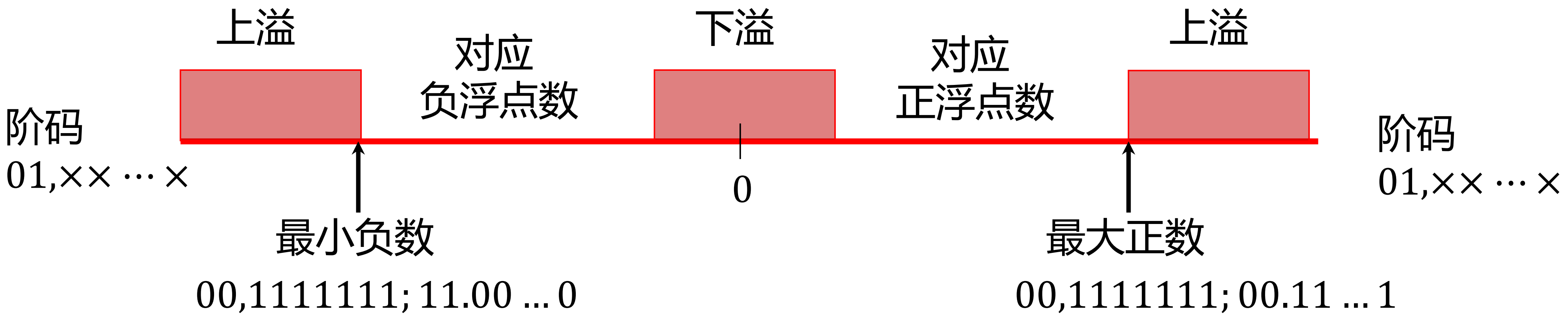
- 设机器数为补码，尾数为规格化形式，并假设阶符取2位，阶码的数值部分取7位，数符取2位，尾数取n位，则该补码在数轴上的表示范围为



浮点加减运算

溢出判断

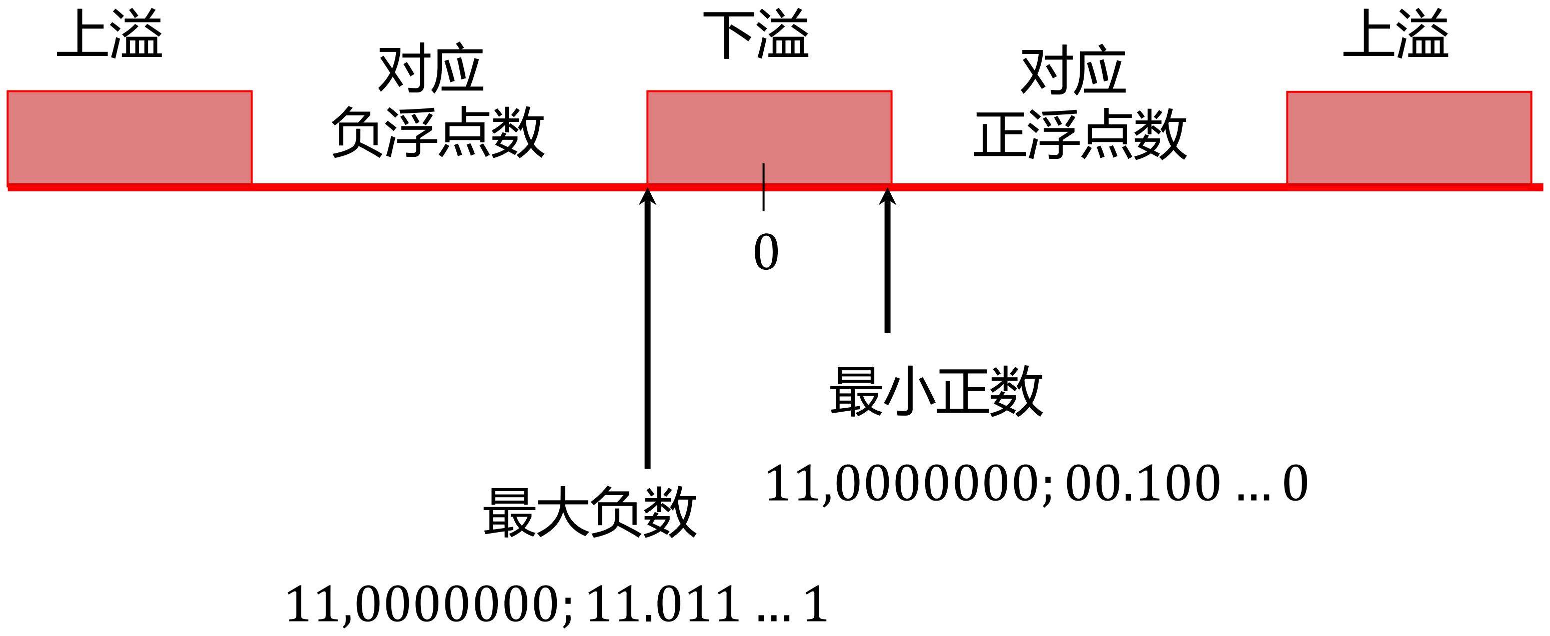
- 上溢
  - 阶码为01,×× ⋯ × 时：发生上溢，为真正溢出，需进行溢出中断处理



浮点加减运算

溢出判断

- 下溢
  - 阶码为10,×× ⋯ ×时：下溢，按机器零处理



浮点加减运算

- 溢出判断
  - 将尾数规格化后，由阶符判断是否溢出
    - 阶符：01
      - 上溢，浮点数真正溢出，需做溢出中断处理
    - 阶符：10
      - 下溢，按机器零处理

谢谢！