计算机组成原理与系统结构

第三章
信息编码与数据表示

http://jpkc.hdu.edu.cn/computer/zcyl/dzkjdx/

杭州電子科技大学
HANGZHOU DIANZI UNIVERSITY

第三章 信息编码与数据表示

- 3. 数值数据的表示
- 3. 数据格式
- 3.3 定点机器数的表示方法
- 3.4 浮点机器数的表示方法
- 3. 非数值数据的表示
- 3. 校验码
- 3.7 现代计算机系统的数据表示
- 本章小结

BACK

3.4 浮点机器数的表示方法

一 浮点机器数的格式

二 浮点机器数的规格化表示

三 浮点数的表示范围

四 浮点数与定点数比较

一、浮点机器数的格式

- 浮点机器数用于表示实数，其小数点的位置由其中的阶码规定，因此是浮动的。
- 浮点数 N 的构成：

$$N = M \times R^{E}$$

- 浮点数的格式：阶码的底是隐含规定的。

Ms	Es	$E_1E_2\cdots\cdots E_m$	$M_1M_2\cdots\cdots M_n$
数符	阶符	阶码数值	尾数数值

小数点位置

- 在机器中，为了方便浮点数大小的比较，通常将数符放置在浮点数的首位。

一、浮点机器数的格式

- 尾数M：为定点小数，尾数的位数决定了浮点数有效数值的精度，尾数的符号代表了浮点数的正负，因此又称为数符。尾数一般采用原码和补码表示。
- 阶码E：为定点整数，阶码的数值大小决定了该浮点数实际小数点位置与尾数的小数点位置（隐含）之间的偏移量。阶码的位数多少决定了浮点数的表示范围。阶码的符号叫阶符。阶码一般采用移码和补码表示。
- 阶码的底R：一般为2、8或16 ，且隐含规定。

IEEE 754 浮点数标准

- 根据 IEEE 754 国际标准，常用的浮点数格式有 3 种，阶码的底隐含为 2 。
- **短实数**又称为单精度浮点数，**长实数**又称为双精度浮点数，**临时实数**主要用于进行浮点数运算时保存临时的计算结果。格式：

<table>
<tr><td>M_S</td><td colspan="2">E</td><td>M</td></tr>
<tr><td>数符</td><td>阶符</td><td>阶码</td><td>尾数</td></tr>
<tr><td>0 ：正数
1 ：负数</td><td colspan="2">2^n-1 的移码</td><td>原码表示；
单、双精度的整数位“ 1” 隐藏；
临时实数无隐藏位</td></tr>
</table>

IEEE 754 浮点数标准

❖位数：

类型	总位数	尾数位数（含1位数符）	阶码位数（含1位阶符）	真值计算
短实数	32	24	8	$N=(-1)^{MS}\times(1.M_1 M_2\ldots\ldots M_n)\times2^{E-127}$
长实数	64	53	11	$N=(-1)^{MS}\times(1. M_1 M_2\ldots\ldots M_n)\times2^{E-1023}$
临时实数	80	65	15	

隐藏位，位于整数位

隐藏位，位于整数位

二、浮点机器数的规格化表示

1. **浮点数的规格化表示：** 为了充分利用尾数的二进制数位来表示更多的有效数字，将尾数的绝对值限定在某个范围之内。

- 例如：R = 2，则规格化浮点数的**尾数 M 应满足条件：绝对值最高有效位为 1**，即

$$\frac{1}{2} \le |M| \le 1$$

没有前导零

二、浮点机器数的规格化表示

2. 计算机硬件对浮点数规格化判断方法：

- 原码表示的尾数：
  - M1=1 ：规格化，即尾数为 ×.1×……× 形式；
- 补码表示的尾数：
  - MS ⊕ M1=1 ：规格化，即尾数为 0.1×……× 形式或者为 1.0×……× 形式

❖ 对于非规格化浮点数，可以通过修改阶码和左右移尾数的方法来使其变为规格化浮点数，这个过程叫做规格化。

二、浮点机器数的规格化表示

- 尾数进行右移实现的规格化，则称为**右规**；
- 尾数进行左移实现的规格化，则称为**左规**。

- 使用规格化的浮点数表示数据的优点：
  - 提高了浮点数据的精度；
  - 使程序能够更方便地交换浮点数据；
  - 可以使浮点数的运算更为简化。

二、浮点机器数的规格化表示

- 例：一浮点数的阶码为 6 位（包括一位阶符），尾数为 10 位（包括一位数符），阶码与尾数均采用补码表示，阶码的底为 2，**数符在浮点数的最高位**。写出 X 与 Y 的规格化浮点数。
  - （1） X ＝－ 123.25
  - （2） Y ＝ 17/64
- （1） X ＝（－ 123.25 ）$_{10}$

  ＝（－ 1111011.01 ）$_2$

  ＝－ 0. 111101101×2^{+7}

二、浮点机器数的规格化表示

- $E_X=+7=(+00111)_2$， $M_X=-0.111101101$
- $[E_X]_{补}=000111$，　　$[M_X]_{补}=1.000010011$
- 则：$[X]_{浮}=1\ 000111\ 000010011$

❖ （2） $Y=(17/64)_{10}$

$=(0.010001)_2$

$=0.10001\times 2^{-1}$

- $E_Y=-00001$， $M_Y=0.100010000$
- $[E_Y]_{补}=111111$，　$[M_Y]_{补}=0.100010000$
- 则：$[Y]_{浮}=0\ 111111\ 100010000$

总结：

- 规格化浮点数表示方法：
- （1）写出数据的二进制真值。
- （2）转换为 $M \times 2^E$ 的形式，其中 M 为没有前导零的定点小数，E 为整数。
- （3）按照格式写出 M 和 E 的规定机器数编码。
- （4）按照格式要求排列 E 和 M 。

IEEE754 浮点数表示方法

- 例： X=17/64， 写出 X 的 IEEE754 **单精度**浮点数。
- 单精度浮点数格式

M_s	E	M
1 位	1 位+7 位	23 位

- X=（0.010001）$_2$
- X =（1. 0001）$_2 \times 2^{-2}$
- 正数， Ms=0
- M=0001000 00000000 0000000
- 【E】$_{移}$ =127-2=125， 【E】$_{移}$ =0111 1101
- 【X】$_{浮}$ = 0 0111 1101 0001000 00000000 0000000

整数位 1，隐藏

尾数数值位 23 位原码

阶码 8 位，2^n-1=127 的移码

IEEE754 浮点数表示方法

- 例： Y=00000000H … E754 的浮点数，求 Y 的十…值。
- Y=00000000H = 0000 0000 0000 0000 0000 0000 0000 0000 B
- Ms=0， 表明正数
- 【E】$_{移}$= 000 0000 0 B = 0 =12…
- 则：阶码的真值 E=0-127=-127
- M=1.000 0000 0000 0000 0000 0000B=1.0
- Y =+1.0×2^{-127}
- Y = 2^{-127}

数符，+

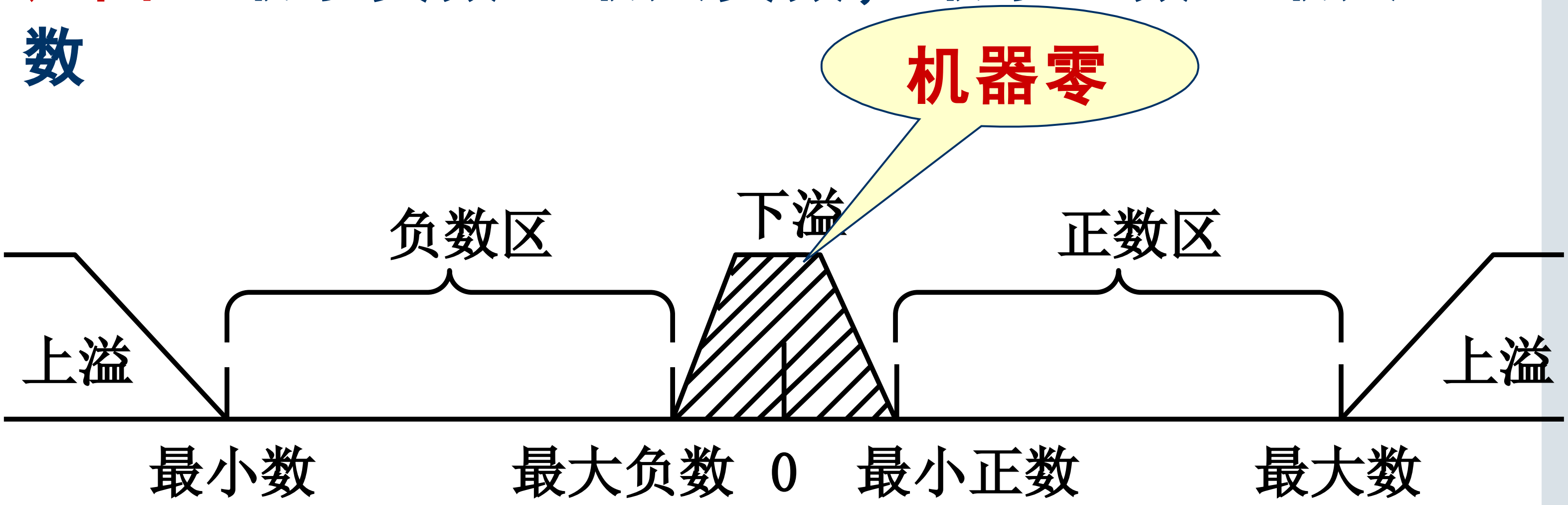
阶码 8 位， 2^n-1 的移码

尾数数值位 23 位原码

添加隐藏位 1

三、浮点数的表示范围

- 浮点数的表示范围通常由4个点界定：最小（负）数、最大负数、最小正数、最大（正）数。
- **范围：**最小负数～最大负数，最小正数～最大正数



三、浮点数的表示范围

- 下溢：位于最大负数和最小正数之间的数据（除0 外），机器无法表示。
  - 处理：计算机直接将其视为机器零。
- 上溢：当一个数据大于最大（正）数，或者小于最小（负）数时，机器也无法表示，称为上溢，上溢又称溢出。
  - 处理：计算机置溢出标志位，或者报警。
- 机器零：有两种情况
  - （1）若浮点数的尾数为零，无论阶码为何值；
  - （2）当阶码的值遇到比它能表示的最小值还要小时（阶码负溢出），无论其尾数为何值

正溢出

负溢出

三、浮点数的表示范围

$$N = M \times R^E$$

浮点数	尾数 M	阶码 E
最小数	最小数	最大数
最大负数	最大负数	最小数
最小正数	最小正数	最小数
最大数	最大数	最大数

三、浮点数的表示范围

- 一浮点数的阶码为 6 位，尾数为 10 位，阶码与尾数均采用补码表示，阶码的底为 2，写出浮点数格式的规格化和非规格化表示范围。
- 解：（1）规格化表示范围：尾数必须规格化

	真值	浮点数表示
最小数	$-1\times 2^{31} = -2^{31}$	1，11111 1.000000000
最大负数	$-(2^{-1}+2^{-9})\times 2^{-32}$	0，00000 1.011111111
最小正数	$2^{-1}\times 2^{-32} = 2^{-33}$	0，00000 0.100000000
最大数	$(1-2^{-9})\times 2^{31}$	1，11111 0.111111111

三、浮点数的表示范围

❖（2）非规格化表示范围：

	真值	浮点数表示
最小数	$-1 \times 2^{31} = -2^{31}$	1，11111 1.000000000
最大负数	$-2^{-9} \times 2^{-32} = -2^{-41}$	0，00000 1.111111111
最小正数	$2^{-9} \times 2^{-32} = 2^{-41}$	0，00000 0.000000001
最大数	$(1-2^{-9}) \times 2^{31}$	1，11111 0.111111111

四、定点数与浮点数比较

- 相同点：
  - 1、计算机所能表示的数据都是一系列离散的点
    - 问题：存在于两个点之间的数据？
    - 解决：计算机通常采用合适的舍入操作，选取最近似的值来替替代。
  - 2、计算机硬件的字长是有限的
    - 问题：数据超出了机器数所能表示的最大界限？
    - 解决：计算机必须能够产生“溢出”的异常报告。

四、定点数与浮点数比较

- **不同点：**
  - 定点数可表示的点在数轴上是均匀的，距离是等长的1（定点整数）或者 2^{-n}（定点小数）；
  - 浮点数则是分布不均匀、距离不相等的。

The End !