数据的机器级表示与运算

浮点数的表示

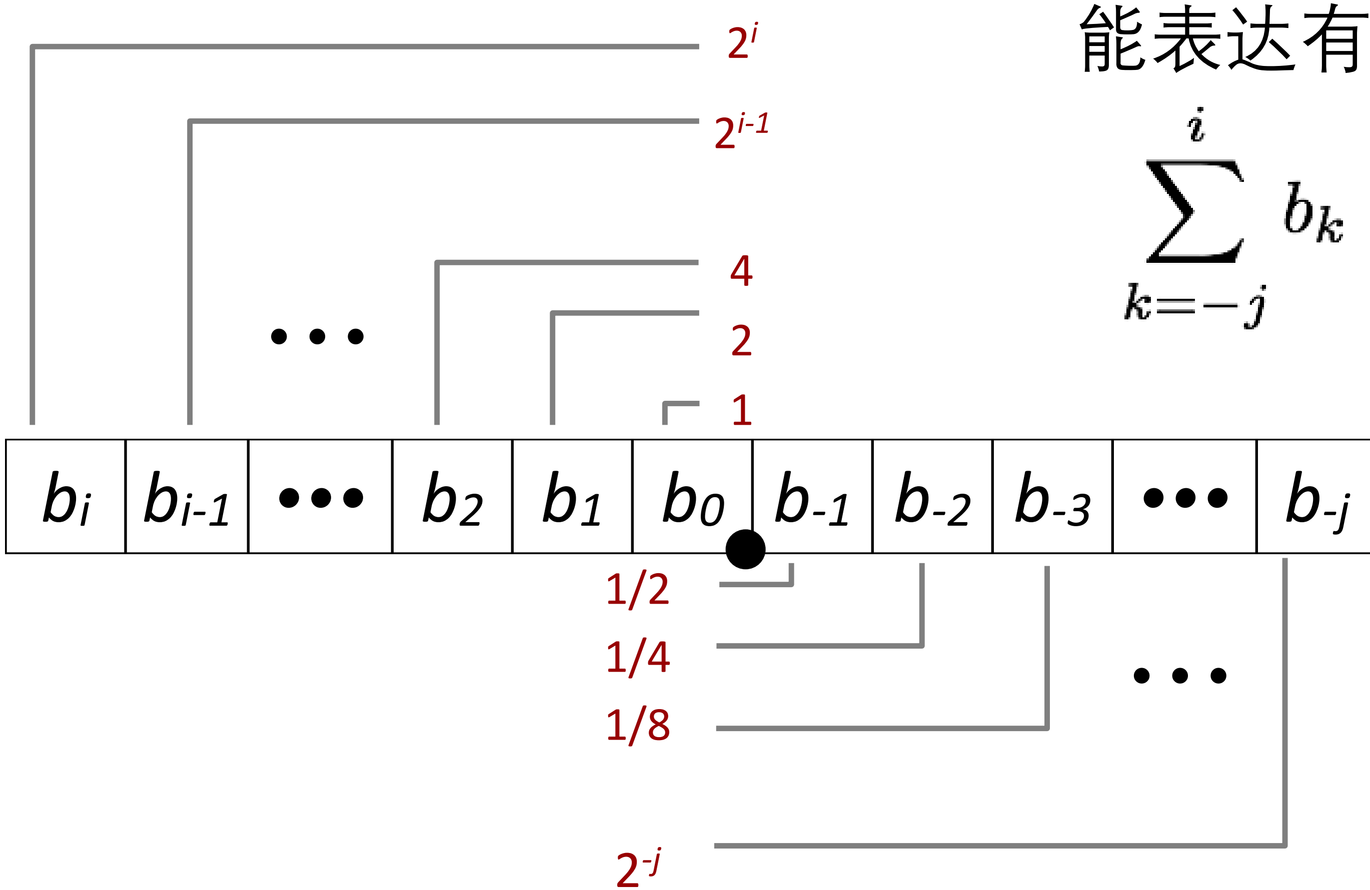
主讲人：邓倩妮

上海交通大学

本节内容

- 浮点数的定义和表示
- IEEE 754浮点数标准
- IEEE 754浮点数的特点

小数点的二进制数



能表达有理数：

$$\sum_{k=-j}^{i} b_k \times 2^k$$

二进制分数: 举例

- 值 表达

值	表达
$5\frac{3}{4}$	101.11_2
$2\frac{7}{8}$	10.111_2
$1\frac{7}{16}$	1.0111_2

- 观察:
  - $0.111111\ldots_2 \quad = 1/2 + 1/4 + 1/8 + \ldots + 1/2^i + \ldots \rightarrow 1.0$
  - 表达为 $1.0 - \varepsilon$

二进制分数表达的数

- 限制 #1
  - 能准确表达的数的形式 ： $x/2^k$
  - 其他有理数，只能 循环小数位近似表达
    - Value　　Representation
      - 1/3　　**0.0101010101[01]**…2
      - 1/5　　**0.001100110011[0011]**…2
      - 1/10　　**0.0001100110011[0011]**…2
- 限制 #2
  - 可表达的二进制位是有限的，例如:w（数据宽度） bits；
  - 可表达的数的个数和范围也是有限的

浮点数如何表示？

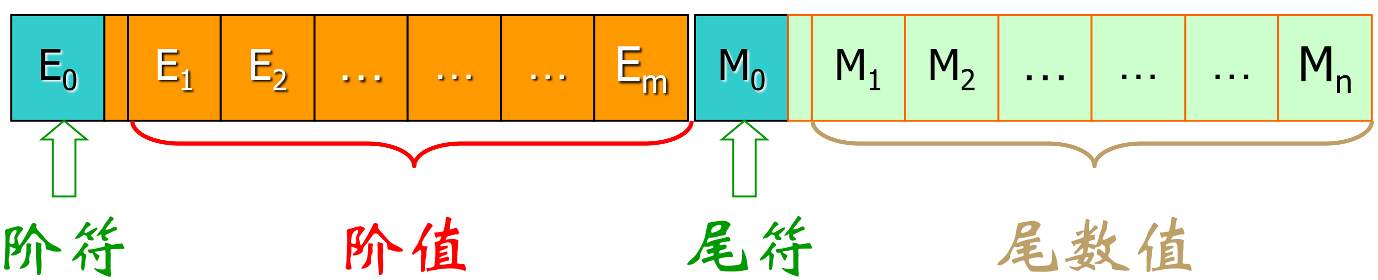
- 参与运算的数据通常既包括整数也包括小数部分。
- 例如： 2.5， 13.45（十进制）， 1001.001 （二进制）
- 如何在机器中表示？

浮点数如何表示？

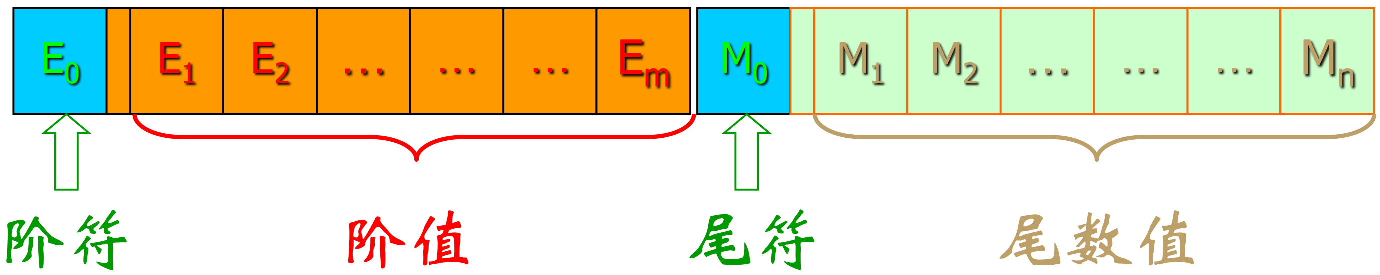
- 将数据按照一定比例因子缩小成定点小数来表示和运算
- 运算完毕后再根据比例因子还原成实际数值
- 例如：科学记数法(十进制)
- $2.5 \times 10^{33} g = 0.25 \times 10^{34}$ （比例因子为10）
- 例如： **101.11_2 = 0.10111** $\times 2^{11}$ （比例因子为2）

浮点数的表示： $N=M \times R^{e}$

- M 称为尾数，是一个纯小数
- R为基数(比例因子)，计算机采用二进制表示浮点数： R=2
- e是比例因子的阶数，称为浮点数的指数，是一个整数，
- $N= 2^{E}\times M \quad =2^{\pm e}\times (\pm m)$



范围 & 精度



- 机器字长一定时，阶码越长，表示范围越大，精度越低
- 浮点数表示范围比定点数大，精度高

举例

- 例1： 8位定点小数可表示的范围
  - 0.0000001 --- 0.1111111
  - 1/128 --- 127/128 （准确表达127个有理数）

- 例2： 设阶码2位， 尾数4位
  - 可表示2^{-11}*0.0001 --- 2^{11}*0.1111
  - 0.0000001 --- 111.1

规格化问题（normalization）

- 例如：$0.05\times10^{1} = 50\times10^{-2} = 5\times10^{-1}$
- 例如：$0.01\times2^{1} = 10\times2^{-2} = 1\times2^{-1}$
- 为了在尾数中表示最多的有效数据位
- 为了数据表示的唯一性
  - 尾数最高有效位为1的数 称为规格化数。
- 两种规格化数
  - 1.XXXXX
  - 0.1XXXXX
- 机器零：全部为0，特殊的数据编码

本节内容

- 浮点数的定义和表示
- IEEE 754浮点数标准
- IEEE 754浮点数的特点

浮点数标准之父

IEEE Standard 754 for Binary Floating-Point Arithmetic.

1989 ACM Turing Award Winner!

Prof. Kahan

www.cs.berkeley.edu/~wkahan/

…/ieee754status/754story.html

浮点数标准 IEEE754

- 单精度 Single precision: 32 bits

s	exp	frac
1	8-bits	23-bits

- 双精度 Double precision: 64 bits

s	exp	frac
1	11-bits	52-bits

- 扩展精度 Extended precision: 80 bits (Intel only)

s	exp	frac
1	15-bits	63 or 64-bits

单精度浮点数编码格式

S(1bit)	E(23~30共8bit)	M(0~22共23bit)

符号位S，阶码E，尾数M

符号位	阶码	尾数	表示
0/1	255	非零1xxxx	*NaN* *Not a Number*
0/1	255	非零0xxxx	*sNaN* *Signaling NaN*
0	255	0	$+\infty$
1	255	0	$-\infty$
0/1	1~254	*f*	$(-1)^S \times (1.f) \times 2^{(e-127)}$
0/1	0	*f*(非零)	$(-1)^S \times (0.f) \times 2^{(-126)}$
0/1	0	0	*+0/-0*

单精度浮点数标准 IEEE754…

- 规格化数(Normal)：

S(1bit)	E(23~30共8bit)	M(0~22共23bit)

代表数值： $(-1)^{s}\times 1.m\times 2^{e-127}$

- 尾数部分采用原码表示，阶为：E减去偏移量 127
- $e_{min}=1,\ e_{max}=254$
- 表达范围： -126 ~ +127
- 规格化数 的最高数字位总是1，该标准将这个1缺省存储(隐藏位implicit)，使得尾数表示范围比实际存储多一位

规格化数举例

$$V = (-1)^s \, M \, 2^E$$
$$E = Exp - Bias$$

- 数值 F = 15213.0;
  - $15213_{10} = 11101101101101_2$
    $= 1.1101101101101_2 \times 2^{13}$
- 尾数 Significand

  $M = \mathtt{1.\underline{1101101101101}}_2$

  $\mathtt{Frac} = \mathtt{\underline{1101101101101}0000000000}_2$

- 阶 Exponent

  $E = 13$

  $Bias = 127$

  $Exp = 140 = \mathtt{10001100}_2$

- 结果 Result:

s	exp	frac
0	10001100	11011011011010000000000

IEEE754 规格化浮点数表示范围

格式	最小值	最大值
单精度	$E_{min}=1, M=0,$ $1.0\times2^{1-127} = 2^{-126}$	$E_{max}=254,$ $f=1.1111\cdots, 1.111\cdots1\times2^{254-127}$ $= 2^{127}\times(2-2^{-23})$
双精度	$E_{min}=1, M=0,$ $1.0\times2^{1-1023} =2^{-1022}$	$E_{max}=2046,$ $f=1.1111\cdots,1.111\cdots1\times2^{2046-1023}$ $=2^{1023}\times(2-2^{-52})$

单精度：（有效尾数24位，相当于7位十进制有效位数）
双精度：（有效尾数53位，相当于17位十进制有效位数）

单精度IEEE754…非规格化数

- 非规格化数(Subnormal) (e=0)

$$(-1)^s \times 0.m \times 2^{-126}$$

- 阶的值：$E = 1 - 127$（而不是 $E = 0 - \mathbf{127}$）（why?）
- 尾数的编码 （with implied leading 0）：
- $M = 0.xxx\cdots x_2$ （有效尾数24位，相当于7位十进制有效位数）
  - **`xxx…x`**: bits of **`frac`**

s	exp	frac
1	8-bits	23-bits

举例

- Case 1：
  - **exp** = **000…0**, **frac** = **000…0**
    - 表达： 0
    - 注意不同的值 ：+0 and –0 (why?)
- **Case 2:**
  - **exp** = **000…0**, **frac** ≠ **000…0**
    - 数值最接近 0.0

s	exp	frac
1	8-bits	23-bits

特殊的值： ∞ (infinity)

exp = **111…1**

- Case 1: **`exp`** = **111…1**, **`frac`** = **000…0**
  - 表示 ∞ (infinity)
  - 一般是 溢出 （ overflows ） 后得到的结果
  - Both positive and negative
  - E.g., $1.0/0.0 = -1.0/-0.0 = +\infty$, $1.0/-0.0 = -\infty$

操作: $5 / 0 = +\infty$, $-5 / 0 = -\infty$

$5+(+\infty) = +\infty$, $(+\infty)+(+\infty) = +\infty$

$5 - (+\infty) = -\infty$, $(-\infty) - (+\infty) = -\infty$ 等等

特殊的值 续(NaN)

exp = **111…1**

- Case 2: **exp = 111…1**, **frac ≠ 000…0**
  - 不是一个数 Not-a-Number (NaN)
  - 表达当数值无法确定时， E.g., sqrt(–1), $\infty - \infty$, $\infty \times 0$

操作：

sqrt (-4.0) = NaN

op (NaN,x) = NaN

+∞- (+∞) = NaN

0/0 = NaN

+∞+(-∞) = NaN

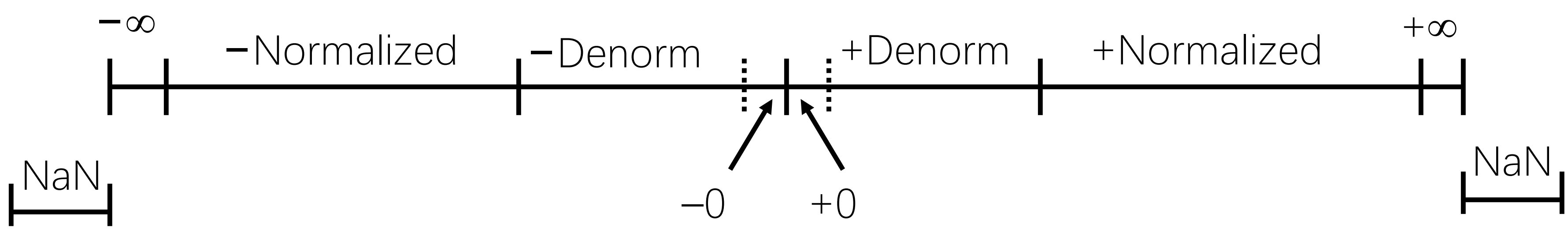
∞/∞ = NaN

等等.

本节内容

- 浮点数的定义和表示
- IEEE 754浮点数标准
- IEEE 754浮点数的特点

IEEE754 浮点数编码可以表达的数

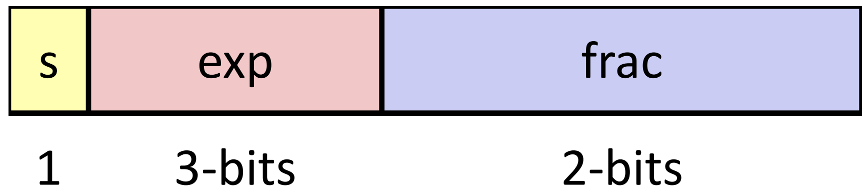


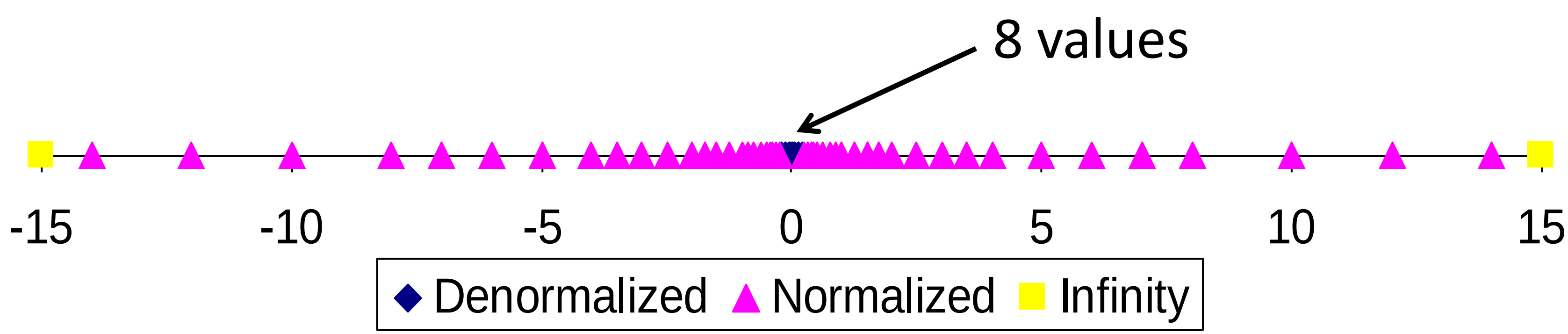
引入非规格化数的原因?

非规格化数、特殊的数 编码的规律？

值的分布

- 6-bit IEEE-like format
  - e = 3 exponent bits
  - f = 2 fraction bits
  - Bias is $2^{3-1}-1 = 3$

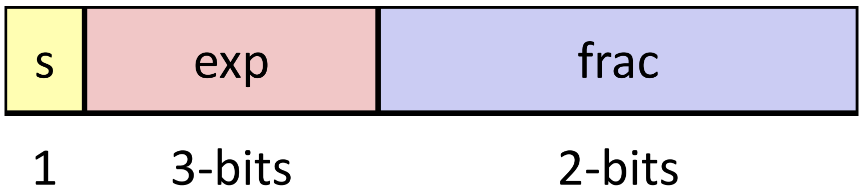


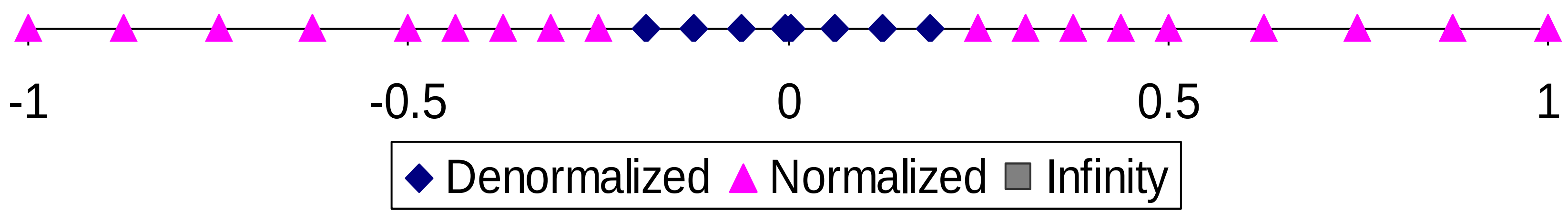


- 注意：可表示的数不是均匀分布的；
- 数轴上越趋向于0越密集.

值的分布 (close-up view)

- 6-bit IEEE-like format
  - e = 3 exponent bits
  - f = 2 fraction bits
  - Bias is 3





IEEE 754的特殊属性

- FP Zero 形式和 Integer Zero 相同
  - All bits = 0
- 几乎可以用Unsigned Integer 比较器直接比较大小，除了：
  - 先比较符号位
  - 必须考虑 -0 = 0
  - NaNs 比任何其他数值都大
  - 其余部分 OK
    - Denorm vs. normalized
    - Normalized vs. infinity

IEEE754浮点数转换：举例

例： 将十进制数-0.75表示成单精度的IEEE754标准代码。

解：$-0.75=-3/4=-0.11_2=-1.1\times2^{-1}$

$=(-1)^1\times(1 + 0.1000\ 0000\ 0000\ 0000\ 0000\ 000)\times2^{-1}$

$=(-1)^1\times(1 + 0.1000\ 0000\ 0000\ 0000\ 0000\ 000)\times2^{126-127}$

$s=1$， $e=126_{10}=01111110_2$， $f = 1000 \ldots 000$

1 01111110 10000000000000000000000

2014考题

14. float型数据据常用IEEE754单精度浮点格式表示。假设两个float型变量x和y分别存放在32位寄存器f1和f2中，若(f1)=CC90 0000H，(f2)=B0C0 0000H，则x和y之间的关系为：

A．x<y且符号相同 B．x<y且符号不同

C．x>y且符号相同 D．x>y且符号不同

(A)

谢谢！