本节内容

IEEE 754 浮点数的表示

跟着游戏学发音

双杀 double kill——英：ˈdʌbl kɪl，美：ˈdʌbl kɪl。

三杀 triple kill——英：ˈtrɪpl kɪl，美：ˈtrɪpl kɪl。

四杀——quadra kill——（英/美）kwɒdrə kɪl。

五杀——penta kill——英：pɛntə kɪl，美：ˈpɛntə kɪl。

triple

英 ['trɪpl]　美 ['trɪpl]

adj. 三倍的；三方的

n. 三倍数；三个一组

vi. 增至三倍

vt. 使成三倍

IEEE 754 标准的浮点数格式

IEEE
Advancing Technology for Humanity

一个国际组织，制定了很多技术标准

电气电子工程师学会（Institute of Electrical and Electronics Engineers，简称IEEE）

IEEE 754 —— 由IEEE制定的二进制浮点数算术标准，规定了在计算机内部，如何使用二进制表示和运算浮点数。

注：C语言 float型（32bit，单精度浮点型）、double型（64bit，双精度浮点型）就是符合 IEEE 754 标准的浮点数格式

“浮点数”部分的408考研大纲变化

浮点数的表示和运算

1. 浮点数的表示
   - 浮点数的表示范围
   - IEEE 754 标准
2. 浮点数的加/减运算

2009年（408元年）大纲

浮点数的表示和运算

1. 浮点数的表示
   IEEE 754 标准
2. 浮点数的加/减运算

2014年大纲

浮点数的表示和运算

- 浮点数的表示：IEEE 754 标准；
- 浮点数的加/减运算

现在的大纲

C语言 float型、double型就是符合 IEEE 754 标准的浮点数格式。最常考 float

【2009年408真题_13】有史以来唯一一道不符合 IEEE 754 格式的浮点数真题

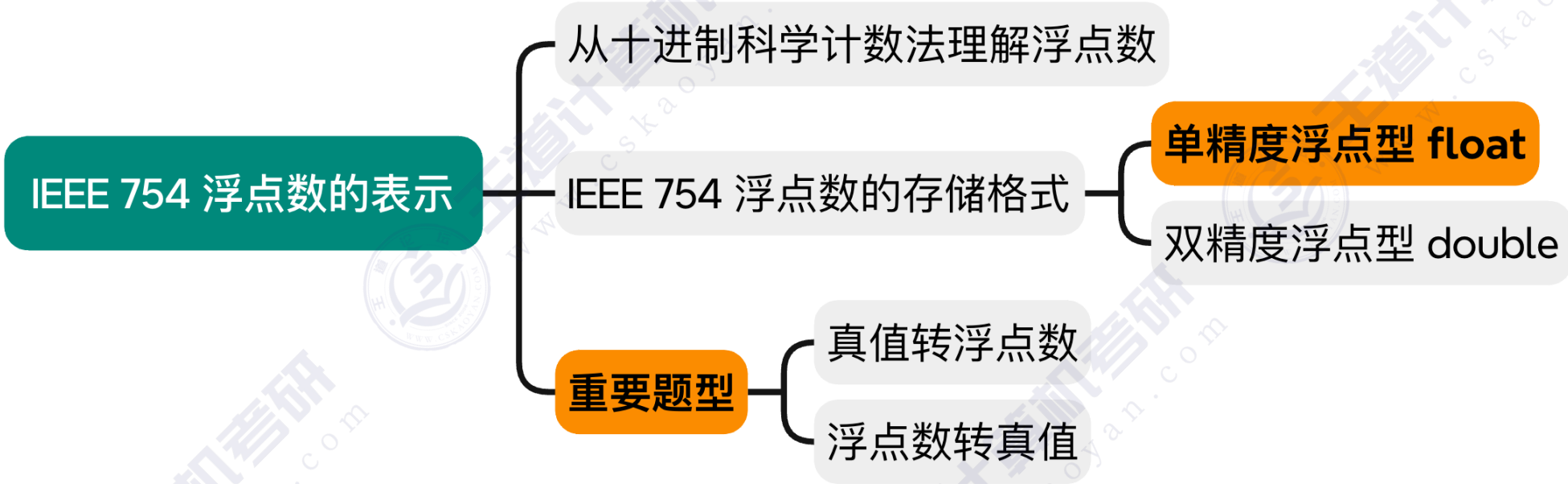
13. 浮点数加、减运算过程一般包括对阶、尾数运算、规格化、舍入和判溢出等步骤。设浮点数的阶码和尾数均采用补码表示，且位数分别为 5 位和 7 位（均含 2 位符号位）。若有两个数 $X=2^7\times29/32$，$Y=2^5\times5/8$，则用浮点加法计算 $X+Y$ 的最终结果是（　　）。

A. 00111 1100010　　B. 00111 0100010　　C. 01000 0010001　　D. 发生溢出

复习建议：

- 对于不符合 IEEE 754 标准的浮点数格式，可以少花时间，当作思维扩展即可。
- 浮点数只需关注加减运算，不关注乘除运算

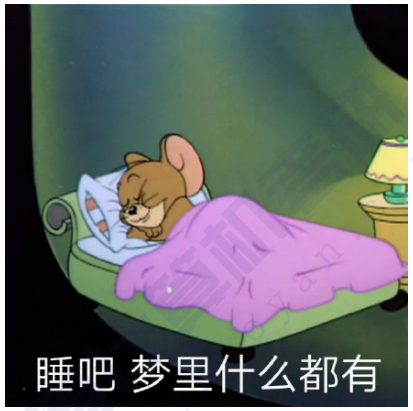
本节总览



定点数的局限性

我的实际财富：- 8540 ¥

2B 定点整数 short 即可表示



我梦里的财富：+302657264526 ¥

超出 4B 定点整数 int 的表示范围

计算机的机器字长位数有限，我们不能无限制地增加数据的长度

从科学计数法理解浮点数

普通记法　　科学计数法

我梦里的财富：+302657264526￥　　$+3.026 \times 10^{11}$

符号　尾数　基数　阶码

电子的质量：

0.00000000000000000000000000910938356 克　　$+9.10938536 \times 10^{-28}$ 克

规格化　规格化

$+0.000910938356 \times 10^{-24}$ 克　　$+910.938536 \times 10^{-30}$ 克

符号：决定数值的正负性
尾数：影响数值的精度。尾数的位数越多，精度越高
阶码：反映小数点的实际位置
基数：K进制通常默认基数为K
规格化：确保尾数的最高位非0数位刚好在小数点之前

从科学计数法理解浮点数

二进制普通记法

真值：-6.75 → -110.11

二进制科学计数法

规格化浮点数：-1.1011×2^{2}

符号：-　尾数：1.1011　基数：2　阶码：2

规格化

非规格化浮点数：-0.011011×2^{4}

尾数“规格化”的好处：便于在计算机内部存储

符号：决定数值的正负性
尾数：影响数值的精度。尾数的位数越多，精度越高
阶码：反映小数点的实际位置
基数：K进制通常默认基数为K
规格化：确保尾数的最高位非0数位刚好在小数点之前

IEEE 754 标准定义的浮点数格式

二进制普通记法　　二进制科学计数法

真值：-6.75　　-110.11　　规格化浮点数　　-1.1011×2^{2}

符号　尾数　基数　阶码

C语言 float 型

1 bit	8 bit	23 bit
符号	阶码	尾数

float：32位单精度浮点数

IEEE
Advancing Technology for Humanity

C语言 double 型

1 bit	11 bit	52 bit
符号	阶码	尾数

double：64位双精度浮点数

注：IEEE 754 也定义了 80bit 扩展精度浮点型（C语言 long double）、16bit半精度浮点型、128bit四倍精度浮点型

float 单精度浮点型的存储

二进制普通记法　　二进制科学计数法

真值：-6.75　-110.11　规格化浮点数　-1.1011×2^{2}

阶码　符号　尾数　基数

如何将十进制真值转换为偏置值为M的移码？

① 将十进制真值 + 偏置值
② 按“无符号整数”规则转换为指定位数

例： 2+127=129 → 10000001

8bit 移码（偏置值为127）

1 bit	8 bit	23 bit
符号	阶码	尾数

1	10000001	.10110000000000000000000

有什么好处：增加尾数的实际精度

符号的存储：0 正 1 负
尾数的存储：规定小数点位置在23bit之前。默认存储规格化尾数，小数点前的 1 省略（隐含）
阶码的存储：用**移码**表示，规定偏置值为 127
基数：基数不用专门存储，规定基数为2即可

double 双精度浮点型的存储

二进制普通记法　　二进制科学计数法

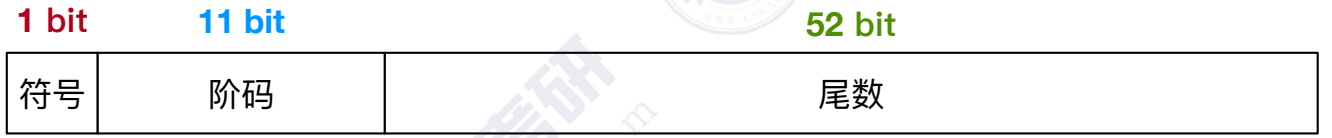
真值：-6.75　-110.11　规格化浮点数　-1.1011×2^2

阶码　符号　尾数　基数

如何将十进制真值转换为偏置值为M的移码？

① 将十进制真值 + 偏置值
② 按“无符号整数”规则转换为指定位数

例： 2+1023=1025 → 10000000001



11bit 移码（偏置值为1023）

1	10000000001	.10110000000000000000000000000000000000000000000000

符号的存储：0 正 1 负
尾数的存储：规定小数点位置在52bit之前。默认存储规格化尾数，小数点前的 1 省略（隐含）
阶码的存储：用移码表示，规定偏置值为 1023
基数：基数不用专门存储，规定基数为2即可