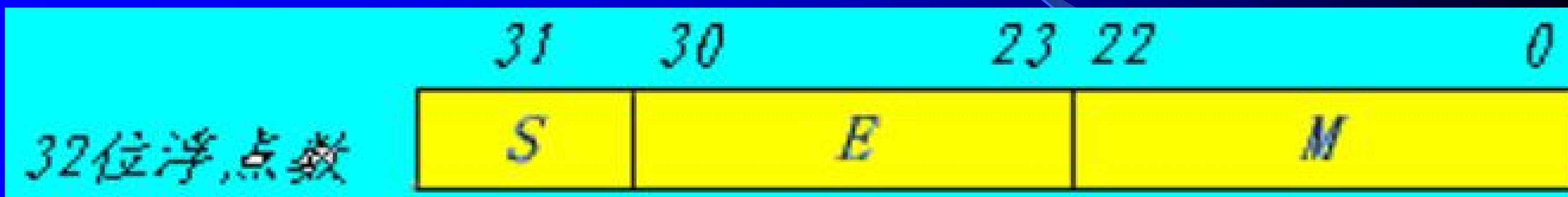
补充：浮点数IEEE754 的标准类型

32位浮点数（短实数）、64位浮点数（长实数）、80位浮点数（扩展实数）的标准格式为：



其中：

- *S*=浮点数的符号位，0表示正数，1表示负数。
- *M*=尾数，23位，用纯小数表示。
- *E*=阶码，8位，阶符采用隐含方式，即采用**移码**方式来表示正负指数，但只偏移**2^n-1**。

	63	62　　52	51　　0
64位浮点数	S	E	M

其中：

- S=浮点数的符号位，0表示正数，1表示负数。
- M=尾数，52位，用纯小数表示。
- E=阶码，11 位，阶符采用隐含方式，即采用移码

方式来表示正负指数，但只偏移2^n-1。

- <u>**几点注释**</u>：
- 为了提高数据的表示精度，当尾数的值不为 0 时，其绝对值 $|M|$应≥ 0.5。
- 浮点数所表示的范围显然远比定点数大。
- 以下两种情况计算机都把该浮点数看成<u>零值</u>，称为<u>机器零</u>。
- (1)当浮点数的尾数M为 0；（不论E为何值）
- (2)当阶码E的值＜Emin值时。（不管M为何值）
- 隐含约定尾数最高位2^0，即1。
- 浮点格式（十进制数与短浮点、长浮点、临时浮点数之间转换有两种方法）
- **方法一**：

[例1]：将十进制数20.59375转换成IEEE754的32位标准浮点数的二进制格式来存储。

- [解:]
- 1.首先分别将整数和分数部分转换成二进制数：
- $(20.59375)_{10} = (10100.10011)_2$
- 2.规格化：移动小数点，使其在第1、2位之间
- $10100.10011 = 1.\underline{010010011} \times 2^4$
- 3.求阶码：小数点被左移了4位，于是得到：$e = 4$
- 移码表示的阶码 ＝偏置值+阶码真值
- 即$E = (127 + 4)_{10} = (131)_{10} = (10000011)_2$

4. 以短浮点数格式存储该数

符号位 $S=0$ 表示该数为正

阶码 $E=$ 10000011 由3可得

尾数 $M=$010010011000000000000000 由2可得,

尾数为23位，不足在后面添14个0

所以，最后得到32位浮点数的二进制存储格式为：

0,100 0001 1,010 0100 1100 0000 0000 0000

表示为十六进制代码为：**41A4C000**

例2：将$(-18.125)_{10}$IEEE754的32位标准浮点数的二进制格式来存储。

解：1）先将$(-18.125)_{10}$ 转换成二进制数

$$(-18.125)_{10}=(-10010.001)_2$$

2）规格化二进制数$(-10010.001)_2$

$$-10010.001=-1.0010001\times 2^4$$

3）计算移码表示的阶码=偏置值+阶码真值：

$$(127+4)_{10}=(131)_{10}=(10000011)_2$$

4）以短浮点数格式存储该数

因此：符号位S=1　　　　　表示该数为负数

阶码E=10000011　　　由3）可得

尾数M=00100010000000000000000　由2）可得：

尾数为23位，不足在后面添16位0

所以，短浮点数代码为：

1;10000011;00100010000000000000000

表示为十六进制代码为：C1910000H。

方法二：公式法

1）单精度浮点计算公式:

（-1）^{S} ×1.M×2^{E-127} ⟶

- S:符号位(1为负,0为正)
- M:尾数,表示小数
- E:阶码

2）双精度浮点计算公式：

（-1）^{S} ×1.M×2^{E-1023}

例子1:IEEE754单精度浮点数:C0A00000H的十进制值是多少?

解：$(C0A00000)_{16}=(1100,0000,1010,0000,0000,0000,0000,0000)_2$

可得，符号位S是：1

阶码位E是：10000001 ==》$(10000001)_2=(129)_{10}$

尾数M是：010,0000, 0000, 0000, 0000, 0000

==》$(0.01000000000000000000000)_2=(0.25)_{10}$

因此,由公式 $(-1)^S\times1.M\times2^{E-127}$得：

$(-1)^1\times1.25\times2^{129-127}=-1.25\times4=-5$

例2：将$(-18.125)_{10}$IEEE754的32位标准浮点数的二进制格式来存储。

解：$(-18.125)_{10}=(-10010.001)_2=-1.0010001\times 2^4$

$$\Rightarrow \begin{cases} S=1 \\ M=0.0010001\text{，即}0010001\ \underline{0..0（16个0）} \\ E=127+4=131\text{，即}(131)_{10}=(10000011)_2 \end{cases}$$

$\Rightarrow \underbrace{\underline{1}}_{S};\underbrace{\underline{10000011}}_{E};\underbrace{\underline{00100010000000000000000}}_{M}$