### 2.4.2　IEEE 浮点表示

前一节中谈到的定点表示法不能很有效地表示非常大的数字。例如，表达式 $5\times2^{100}$ 是用 101 后面跟随 100 个零的位模式来表示。相反，我们希望通过给定 $x$ 和 $y$ 的值，来表示形如 $x\times2^y$ 的数。

IEEE 浮点标准用 $V=(-1)^s\times M\times2^E$ 的形式来表示一个数：

- 符号(sign)　$s$ 决定这数是负数($s=1$)还是正数($s=0$)，而对于数值 0 的符号位解释作为特殊情况处理。
- 尾数(significand)　$M$ 是一个二进制小数，它的范围是 1～$2-\varepsilon$，或者是 0～$1-\varepsilon$。
- 阶码(exponent)　$E$ 的作用是对浮点数加权，这个权重是 2 的 $E$ 次幂(可能是负数)。

将浮点数的位表示划分为三个字段，分别对这些值进行编码：

- 一个单独的符号位 $s$ 直接编码符号 $s$。
- $k$ 位的阶码字段 exp$=e_{k-1}\cdots e_1e_0$ 编码阶码 $E$。
- $n$ 位小数字段 frac$=f_{n-1}\cdots f_1f_0$ 编码尾数 $M$，但是编码出来的值也依赖于阶码字段的值是否等于 0。

图 2-32 给出了将这三个字段装进字中两种最常见的格式。在单精度浮点格式(C 语言中的 float)中，s、exp 和 frac 字段分别为 1 位、$k=8$ 位和 $n=23$ 位，得到一个 32 位的表示。在双精度浮点格式(C 语言中的 double)中，s、exp 和 frac 字段分别为 1 位、$k=11$ 位和 $n=52$ 位，得到一个 64 位的表示。

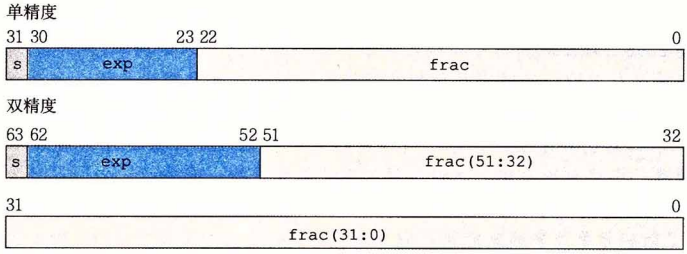


图 2-32　标准浮点格式(浮点数由 3 个字段表示。两种最常见的格式是它们被封装到 32 位(单精度)和 64 位(双精度)的字中)

给定位表示，根据 exp 的值，被编码的值可以分成三种不同的情况(最后一种情况有两个变种)。图 2-33 说明了对单精度格式的情况。

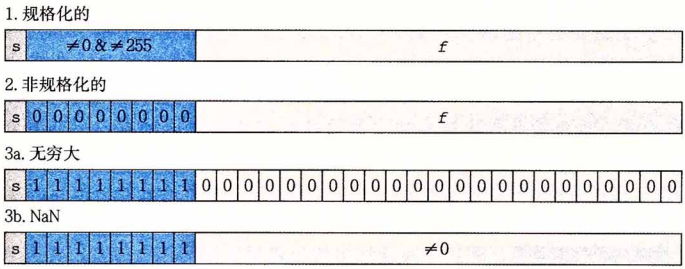


图 2-33　单精度浮点数值的分类(阶码的值决定了这个数是规格化的、非规格化的或特殊值)