第 3 章　机器数的表示

本章介绍机器数的特点，以及它的表示方法和各种表示之间的相互转换。

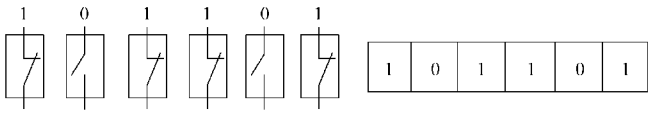
3.1　机器数的特点

学习计算机要弄清它是怎样记忆数据和传送数据，又是怎样处理数据；而要弄清这些，首先就得明白数在计算机中是怎样表示的。数在计算机中的表示称做机器数。本章介绍机器数的特点。

1. 机器数用二进制数码表示

这是因为二进制数有如下 4 个特长。

(1) 可用任何具有两个不同稳定状态的元件来表示　因为二进制只用两个数码，即 0 和 1，这也就是说，二进制数的每 1 位不是 0 便是 1。所以，可以用任何具有两个稳定状态的元件，如用开关来表示每 1 位。这只要事先规定好所用元件两个状态哪个为 1，哪个为 0 就可以了。如规定开关的通为 1，断为 0。这样，一个二进制数便可用一排开关表示出来。例如，二进制数 101101，根据以上假定，便可用对应的 6 个开关来表示。它们的对应关系如图 3.1 所示。



(a) 用开关表示的二进制数　　(b) 机器数的形象表示

图 3.1　机器数的表示

为了使机器数形象化，也为了同一般的数相区别，用图 3.1(b)所示的形式来表示机器数。

(2) 四则运算简单　大家都知道，在用十进制数做乘法时，必须要记熟九九歌，即一一得一、一二得二……九九歌有多少条呢？至少为 45 条。同样在做十进制的加法运算时，也得记住：1＋1＝2，1＋2＝3……至少也是 45 条。而二进制乘法规则和加法规则极为简单，至多有 4 条：

0＋0＝0	0×0＝0
0＋1＝1	0×1＝0
1＋0＝1	1×0＝0
1＋1＝10	1×1＝1

这就使二进制的运算比起十进制的运算简单得多，从而使实现运算的计算机运算电路也简单可靠。

这里举例说明二进制的四则运算方法。

[例 3.1]

$$\begin{array}{r} 1001 \\ +\ 0010 \\ \hline 1011 \end{array} \quad \text{验算} \quad \begin{array}{r} 9 \\ +\ 2 \\ \hline 11 \end{array}$$

[例 3.2]

$$\begin{array}{r} 1110 \\ -\ 1001 \\ \hline 0101 \end{array} \quad \text{验算} \quad \begin{array}{r} 14 \\ -\ 9 \\ \hline 5 \end{array}$$

[例 3.3]

$$\begin{array}{r} 0101 \\ \times\ 0010 \\ \hline 0000 \\ 0101 \\ 0000 \\ \hline 0000 \\ \hline 0001010 \end{array} \quad \text{验算} \quad \begin{array}{r} 5 \\ \times\ 2 \\ \hline 10 \end{array}$$

[例 3.4]

$$\begin{array}{r} 101 \\ 10\overline{)\,1010} \\ 10 \\ \hline 10 \\ 10 \\ \hline 0 \end{array} \quad \text{验算} \quad \begin{array}{r} 5 \\ 2\overline{)\,10} \\ 10 \\ \hline 0 \end{array}$$

验算说明二进制的四则运算结果同十进制的完全相同。

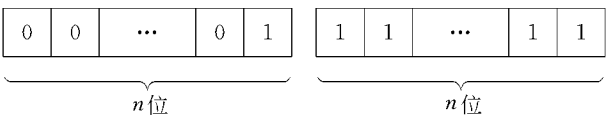
(3) 节省存储设备　如用十进制数表示 0～9 这 10 个数码就必须用 10 个设备；而用二进制数却只需要 4 个设备便可表示出 0～15 的数的范围。可见用二进制表示同样一个数要比用十进制表示节省设备。

(4) 便于逻辑分析与设计　二进制的数码 0 和 1，完全可以用来表示逻辑代数中变量的假和真这两个取值，从而使逻辑表达式的分析与设计更为简便。

2. 机器数所表示的数值范围有限

这是因为计算机字长有限的缘故。那么，字长为 n 位的计算机，它所能表示的数的范围有多大呢？下面分两种情况来讨论。

(1) 假定小数点位于机器数的最右边，即机器数表示不带符号的整数。这时，除 0 以外的机器数的最小值为 1，最大值为 2^n-1，分别如图 3.2(a)、图 3.2(b)所示。这时，机器数所能表示的范围便是 $1 \leqslant x \leqslant 2^n-1$。

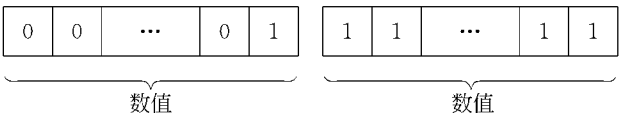


(a) 最小值为 1　　(b) 最大值为 2^n-1

图 3.2　不带符号整数的机器数表示

在这种情况下，凡是比 1 小的值都认为是机器零；如果数值大于 2^n-1 便产生溢出，溢出意味着机器无穷大。

(2) 假定小数点位于机器数的最左边，即机器数表示纯小数时，除 0 之外，它的最小值为 2^{-n}，最大值为 $1-2^{-n}$，分别如图 3.3 的(a)、(b)所示。这时，机器数所能表示的范围为 $2^{-n} \leqslant x \leqslant 1-2^{-n}$。



(a) 最小值为 2^{-n}　　(b) 最大值为 $1-2^{-n}$

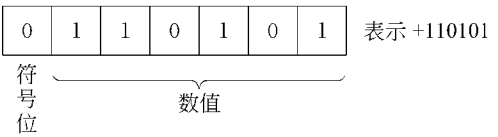
图 3.3　不带符号纯小数的机器数表示

在这种情况下，凡是小于 2^{-n} 的数都认为是机器零，而大于 $1-2^{-n}$ 的数便是机器无穷大。只要运算结果大于$(1-2^{-n})$，便产生溢出。

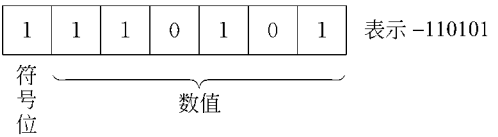
从上述两种情况可清楚地看出，溢出的根本原因在于计算机的字长有限。

3. 符号的数值化

任何能表示数的设备如开关、触发器等只有两种状态，即 0 态和 1 态。所以完全可以借用这两种状态来表示数的符号。约定俗成用 0 代表＋号，用 1 代表－号，如图 3.4 所示。



(a) 正数的机器数表示



(b) 负数的机器数表示

图 3.4　机器数的符号表示

4. 小数点的位置有一定的约定方式

在机器数中，小数点的位置有两种约定方式，即定点表示方式和浮点表示方式。

(1) 定点方式　在这种方式中，小数点的位置固定不变，但在约定上有较大的灵活性。既可以约定在符号位的右边，这时，机器数表示纯小数，如图 3.5(a)所示，也可以约定在最低位的右边，这时，机器数表示纯整数，如图 3.5(b)所示。

当然，小数点也可以认定在任何位置。注意，在机器数中，小数点实际并不占用位空间，只是约定在某位的左边或右边。

采用定点方式表示的机器数叫定点数，只能使用定点数的计算机叫定点机。

(2) 浮点方式　在这种方式中，把一个二进制数 X 表示成

$$X=2^r \cdot x$$

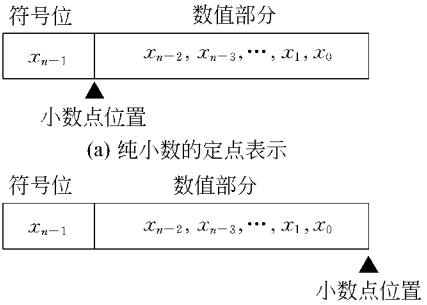
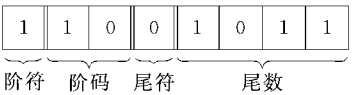


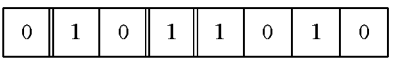
图 3.5　常用的定点方式

其中，r 称为 X 的阶码，指明小数点的位置；x 称为 X 的尾数，表示 X 的有效数字。

例如，浮点数 $N=2^{-10}\times 0.1011$ 的机器数表示形式如下：



又如浮点数 $N=-2^{10}\times 0.1010$ 的机器数表示形式为



采用浮点方式表示的机器数叫浮点数，能够使用浮点数的计算机叫浮点机。

浮点表示中有两个问题需要考虑：

① 阶码位数 r 和尾数的位数 q 的关系。在字长确定的情况下，r 增加，数的表示范围就增加，但 q 减少，故有效数字也减少。因此，r、q 的位数分配要得当。

② 浮点数通常采用规格化的表示。所谓规格化的数是指尾数的第 1 位为 1 的浮点数，例如：

$$N_1=2^{01}\times 0.1101$$

$$N_2=-2^{01}\times 0.1101$$

而尾数的第 1 位不是 1 的浮点数，就是非规格化的数，例如：

$$N_3=2^{11}\times 0.0101$$

要使浮点数规格化，只要移动小数点和调整阶码同时进行即可。具体做法是：阶码每递减一次 1，尾数的小数点便右移 1 位，直到尾数的第 1 位是 1 为止。例如：

$$N_2=2^{11}\times 0.0101=2^{10}\times 0.1010$$

最后结果已规格化了。

3.2　定点数的原码、反码和补码

机器数有原码、反码、补码和移码 4 种表示形式。本节介绍定点数的原码、反码和补码。

1. 定点数的原码表示

真值 x 的符号数值化后所表示出来的机器数就称做原码，记做 $[x]_{原}$。

(1) 正数的原码

设　　字长 $n=6, x=+11010$

则　　$[x]_{原}=[+11010]_{原}=0,11010$

即　　$[x]_{原}=x+2^n, \quad 2^{n-1}>x>0$

(2) 负数的原码

设　　字长 $n=6, x=-10011$

则　　$[x]_{原}=[-10011]_{原}=1,10011$

显然　　$[x]_{原}=2^{n-1}-x, \quad -2^{n-1}<x<0$

[例 3.5] 设字长为 8 位，$x=-1110011$，求 $[x]_{原}$。

[解]　　$[x]_{原}=2^{8-1}-(-1110011)=1,1110011$

可见，正数的符号位添上 0，负数的符号位添上 1 就是该数的机器数的原码表示。原码就是原数值部分的数码不变的意思。注意，符号位后的逗号，只是为书写整数的原码时突出符号位，实际的机器数并无此逗号。

(3) 0 的原码表示

① 正零的原码

若　　$x=+\overbrace{00\cdots0}^{(n-1)个0}$

则　　$[x]_{原}=[+0\ 0\ \cdots\ 0]_{原}=0,\overbrace{00\cdots00}^{(n-1)个0}$，　字长按 n 位

② 负零的原码

若　　$x=-\overbrace{00\cdots0}^{(n-1)个0}$

则　　$[x]_{原}=[-00\cdots0]_{原}=1,\overbrace{00\cdots00}^{(n-1)个0}$，　字长按 n 位

可见，0 在原码中有两种表示形式，换句话说，在原码表示中遇到这两种情况的机器数时都当作 0 处理。

综合(1)、(2)、(3)，可得到字长为 n 的定点整数的原码的定义：

$$[x]_{原}=\begin{cases}2^n+x, & 2^{n-1}>x\geqslant 0\\ 2^{n-1}-x, & -2^{n-1}<x\leqslant 0\end{cases} \tag{3.1a}$$

一种常用情况，即小数点在符号位之后，即 $1>x>-1$ 时，有

$$[x]_{原}=\begin{cases}2+x, & 1>x\geqslant 0\\ 2^0-x, & 0\geqslant x>-1\end{cases} \tag{3.1b}$$

[例 3.6] 设字长为 8 位，求下列两数的原码。

$$x_1=0.1101,\ x_2=-0.1101$$

[解]

$$[x_1]_{原}=0.1101000$$

$$[x_2]_{原}=1.1101000$$

同样注意，符号位后的小数点，也只是为书写纯小数的原码时突出符号位，实际的机器数并无此小数点。

2. 定点数的反码表示

关于正数和负数的反码有如下规定。

(1) 正数的反码

设　　字长 $n=6, x=+10101$

则　　$[x]_{反}=0,10101$

即　　$[x]_{反}=[x]_{原}=2^n+x, \quad 2^{n-1}>x>0$

(2) 负数的反码

设　　字长 $n=6, x=-11011$

则　　$[x]_{反}=1,00100$

即负数的反码其数值部分为真值 x 按位取反，反码也正由此而得名。不难看出，负数 x 的反码可用如下公式求出：

$$[x]_{反}=2^n-1+x, \quad -2^{n-1}<x<0$$

[例 3.7] 设字长 $n=8, x=-110011$，求 $[x]_{反}$。

解　　$[x]_{反}=2^8-1+x=11111111-110011=1,1001100$

可见，正数的反码就是该数的原码，负数的反码符号位为 1，而数值部分等于真值按位取反。

根据这一结论便得出零的反码表示如下。

(3) 零的反码

① $[+0]_{反}=0, \overbrace{00\cdots0}^{(n-1)个0}$　(字长按 n 位)

② $[-0]_{反}=1, \overbrace{11\cdots1}^{(n-1)个1}$　(字长按 n 位)

可见，机器数在反码表示中，0 的表示也有两种形式，遇到这两种情况的机器数都当作 0 处理。

综合(1)、(2)、(3)，可得到字长为 n 的定点整数的反码的定义：

$$[x]_{反}=\begin{cases}2^n+x, & 2^{n-1}>x\geqslant 0\\ 2^n-1+x, & 0\geqslant x>-2^{n-1}\end{cases} \tag{3.2a}$$

一种常用的情况，即 $-1<x<1$ 时字长为 n 的定点数反码定义：

$$[x]_{反}=\begin{cases}2+x, & 1>x\geqslant 0\\ 2-2^{-(n-1)}+x, & 0\geqslant x>-1\end{cases} \tag{3.2b}$$

[例 3.8] 设字长为 8 位，求下列两数的反码。

$$x_1=0.1101, \quad x_2=-0.1101$$

[解]

$$[x_1]_{反}=0.1101000$$

$$[x_2]_{反}=1.0010111$$

3. 定点数的补码表示

(1) 互补数　对于两个整数 a、b，如果用某一个正整数 k 去除所得的余数相同，则称 a、b 对于模 k 来说是同余数，也称做互补。

当 a、b 对于模 k 互补时，就可以说 a 和 b 在模 k 的意义下是相等的，记做

$$a=b \pmod{k}$$

例如，$a=13, b=25, k$ 取 12，则余数均为 1，13 和 25 就是同余数，记做

$$13=25 \pmod{12}$$

(2) 补码的概念　利用互补的概念，便有

$$\begin{aligned} a+k &= a \pmod{k} \\ a+2k &= a \pmod{k} \\ &\vdots \\ a+nk &= a \pmod{k} \end{aligned} \tag{3.3}$$

其中，式 $a+k=a \pmod{k}$ 可记做

$$[a]_{\text{补码}} = k+a \tag{3.4}$$

$[a]_{\text{补码}}$ 称做 a 对模 k 的补码，简称 a 的补码。

例如，$a=-5$，$k=12$，根据式(3.3)，则有

$$-5+12=-5 \pmod{12}$$

即

$$7=-5 \pmod{12}$$

根据式(3.4)可记做

$$[-5]_{\text{补}} = 12+(-5) = 7$$

即 -5 对于模 12 的补码是 7。这说明在模 12 的意义下，-5 相当于 $+7$。这样，正负数之间的相加，便可以转化为正数间的相加，例如：

$$4+(-5) = 4+7 \pmod{12}$$

这个式子是有实际意义的。实际意义就是钟表的走时问题，即从 4:00 开始再过 7h 是 11:00，而这 11:00 也等于从 4:00 开始退回 5h 的结果。

(3) 计算机所用的补码　对于字长为 n 的计算机来说，数 2^n 已溢出，故有

$$x+2^n = x \pmod{2^n}$$

根据式(3.4)便有

$$[x]_{\text{补}} = 2^n+x \tag{3.5a}$$

利用同余的概念，式(3.5a)可缩写为

$$[x]_{\text{补}} = x \pmod{2^n} \tag{3.5b}$$

下面，根据式(3.5a)来讨论正数、负数和零的补码。

① 正数的补码

设　　字长 $n=6$，　$x=+11010$

则　　$[x]_{\text{补}}=2^6+11010=0,11010$

② 负数的补码

设　　字长 $n=6$，　$x=-11010$

则　　$[x]_{\text{补}}=2^6+(-11010)=1,00110$

③ 零的补码

设　　字长 $n=6$

则　　$[+0]_{\text{补}}=2^6+00000=0,00000$

$[-0]_{\text{补}}=2^6+(-00000)=0,00000$

由①、②、③可见，正数的补码就是该数的原码；负数的补码符号位为 1，而数值部分等

于真值按位取反加 1；0 的补码只有一种形式，就是 n 个 0，这称做零元素的唯一性。

④ 当 $x=-2^{n-1}$ 时，

$$[x]_{补}=2^n+(-2^{n-1})=2^{n-1}(2-1)=2^{n-1}$$

这表明，-2^{n-1} 的补码是存在的且等于 2^{n-1}。

综合①、②、③、④，可得字长为 n 的定点数的补码定义：

$$[x]_{补}=2^n+x,\quad 2^{n-1}>x\geqslant -2^{n-1} \tag{3.6a}$$

一种常用情况，即对于小数点位于符号位之后（$-1\leqslant x<1$）的定点数的补码定义：

$$[x]_{补}=2+x,\quad 1>x\geqslant -1 \tag{3.6b}$$

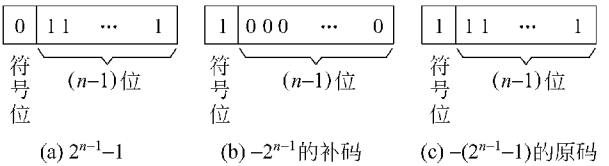
[例 3.9] 设字长 $n=8$，求下列两数的补码：$x_1=0.1101$，$x_2=-0.1101$。

[解] $[x_1]_{补}=0.1101000$

$[x_2]_{补}=2^1+x=10-0.1101=1.0011000$

[例 3.10] 定长为 n 的定点数，假定第一位为符号位，小数点位于最低位的后边，问该数所能表示的最大值和最小值各是多少？

[解] 这种情况下的最大值、最小值的机器数表示形式分别如图 3.6 的(a)、(b)所示。因为最大值只能出现在符号为正，各位全为 1 时，所以如图 3.6(a)所示，最大值为 $2^{n-1}-1$；那么最小值是不是就是 $-(2^{n-1}-1)$，即图 3.6(c)所示的呢？不是的！因为机器数一般用补码表示，而 -2^{n-1} 的补码是存在的，那就是如图 3.6(b)所示的，故最小值是 -2^{n-1}。



(a) $2^{n-1}-1$　(b) -2^{n-1}的补码　(c) $-(2^{n-1}-1)$的原码

图 3.6　字长为 n 的定点数的最大值和最小值

3.3　变形码、移码和浮点数表示

本节介绍机器数的变形码、移码和浮点数的表示方法。

1. 变形码

(1) 什么是变形码　如果将符号位改为多位，用每位都是 1 表示负，用每位都是 0 表示正，那么，这样所得到的机器数的表示形式就称做变形码。例如，符号位取两位，若 $X>0$，符号位为 00；而 $X<0$，符号位为 11。

(2) 变形码的表示　变形码也有变形原码、变形反码和变形补码 3 种表示形式。

[例 3.11] 假定机器数的字长为 8 位，符号占 2 位，请写出下列数的变形原码、变形反码和变形补码。

$$[+1011]_{变原}=00,001011$$

$$[+1011]_{变反}=00,001011$$

$$[+1011]_{变补}=00,001011$$

$$[-1011]_{变原}=11,001011$$

$$[-1011]_{变反}=11,110100$$
$$[-1011]_{变补}=11,110101$$
$$[-0.1011]_{变原}=11.101100$$
$$[-0.1011]_{变反}=11.010011$$
$$[-0.1011]_{变补}=11.010100$$

人们常把符号位为 2 位的变形码称做模 4 码，而把符号位为 1 位的变形码称做模 2 码。

2. 定点数的移码

移码又称增码、余码，或偏码，常用于表示浮点数中的阶码。

对于字长为 n 的机器数，取 1 位符号位，这时，真值 x 所对应的移码，定义如下：

$$[x]_{移}=2^{n-1}+x,\quad -2^{n-1}\leqslant x<2^{n-1} \tag{3.7a}$$

该定义也可记做

$$[x]_{移}=x,\quad \bmod 2^{n-1} \tag{3.7b}$$

对于纯小数来说，其移码定义为

$$[x]_{移}=1+x,\quad -1\leqslant x<+1 \tag{3.7c}$$

下面讨论几个问题。

(1) 符号问题　当 $x>0$ 时，真值 x 最高位加 1，即$[x]_{移}$，符号位为 1；当 $x<0$ 时，2^{n-1}减去真值 x 的绝对值，即$[x]_{移}$，符号位为 0。可见，同一个真值的移码与原码、反码、补码符号位的值正好相反。

(2) 0 的移码　0 可以看做是+0，也可以看做是−0，根据移码的定义，于是有

$$[+0]_{移}=2^{n-1}+0=2^{n-1}$$
$$[-0]_{移}=2^{n-1}-0=2^{n-1}$$

可见，0 的移码也具有零元素的唯一性。

(3) -2^{n-1}的移码　根据移码的定义，则有

$$[-2^{n-1}]_{移}=2^{n-1}+(-2^{n-1})=0$$

可见，-2^{n-1}的移码其绝对值最小。

(4) 移码与补码的关系　根据补码的定义，真值 x 的补码为

$$[x]_{补}=2^{n}+x=2\cdot 2^{n-1}+x=2^{n-1}+2^{n-1}+x$$
$$=2^{n-1}+[x]_{移}$$

从上面的推导结果可以看出，移码的符号位加 1，就是该数的补码。换句话说，同一个真值的补码与移码的异同是，数值部分完全相同，而符号正好相反。因此，补码和移码之间的变换只要符号位取反即可。

[例 3.12]　写出下列数的补码和移码。假定字长为 8 位，1 位符号位。

$$[+1011]_{补}=0,0001011$$
$$[+1011]_{移}=1,0001011$$
$$[-1011]_{补}=1,1110101$$
$$[-1011]_{移}=0,1110101$$

[例 3.13]　从供选答案中，选出正确答案。

字长为 8 位的机器数的移码 2^{7}、0 和 255 的真值分别为[A]、[B]和[C]。

供选择答案(A、B、C)：

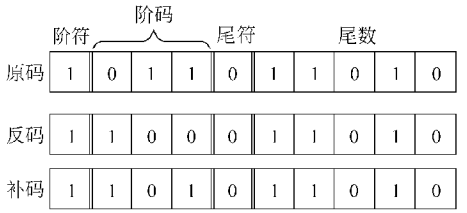
① +127　② +128　③ +255　④ 0

⑤ −128　⑥ −127　⑦ −255　⑧ +1

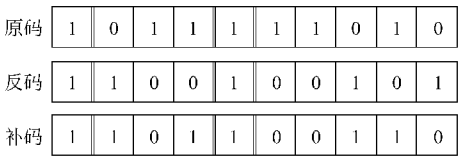
答案：A 为④；B 为⑤；C 为①。

3. 浮点数的表示

(1) 浮点数的原码、反码、补码　把原码、反码和补码的定义分别应用到浮点数的阶码和尾数上，即得到浮点数的原码、反码和补码。例如，浮点数 $x=2^{-11}\times 0.11010$ 的原码、反码和补码分别是



又如，浮点数 $x=-2^{-11}\times 0.11010$ 的原码、反码和补码分别是



[例 3.14]　设某计算机的字长为 16 位，其中 3 位用来表示阶码，11 位用来表示尾数，阶符和尾符各占一位。求该浮点数用补码表示时的最大值、最小值和最小绝对值(0 除外)。

[解]　该浮点数的最大值、最小值和最小绝对值的补码表示如图 3.7 的(a)、(b)和(c)所示。

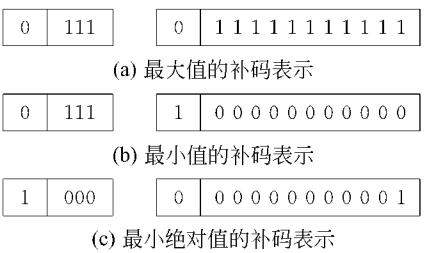


图 3.7　浮点数的最大值、最小值和最小绝对值

可见，该浮点数的最大值为 $2^{2^3-1}\times(1-2^{-11})=2^7\times(1-2^{-11})=127.9375$；最小值为 $2^{2^3-1}\times(-1)=-2^7=-128$；最小绝对值为 $2^{-2^3}\times(0.00000000001)=2^{-8}\times 2^{-11}=2^{-19}$。

(2) 规格化浮点数的尾数其模 4 补码的形式　设浮点数的尾数值为 M，其规格化的值，对于正数，应满足 $\frac{1}{2}\leqslant M<1$，模 4 补码形式为 00.10…00～00.11…11，即 00.1ϕ…$\phi\phi$；对于负数，除 $-\frac{1}{2}$ 之外，应满足 $-1\leqslant M<-\frac{1}{2}$，模 4 补码形式为 11.00…00～11.01…11，即