Part 3

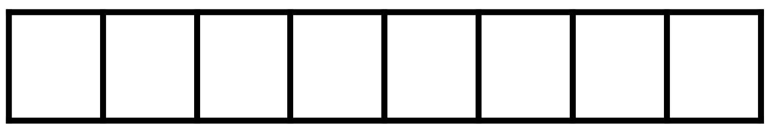
第 6 章 计算机的运算方法

6.1 无符号数和有符号数

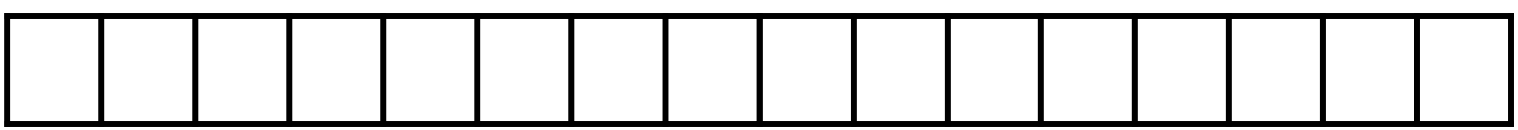
一、无符号数

寄存器的位数

反映无符号数的表示范围



8 位　　　　　　　0 ~ 255



16 位　　　　　　0 ~ 65535

二、有符号数

1. 机器数与真值

真值	机器数	
带符号的数	符号数字化的数	
+ 0.1011	0 ‖ 1011	小数点的位置
– 0.1011	1 ‖ 1011	小数点的位置
+ 1100	0 ‖ 1100	小数点的位置
– 1100	1 ‖ 1100	小数点的位置

2. 原码表示法

6.1

(1) 定义

整数

$$[x]_{原} = \begin{cases} 0, \ x & 2^n > x \geq 0 \\ 2^n - x & 0 \geq x > -2^n \end{cases}$$

x 为真值　　n 为整数的位数

如　$x = +1110$　　$[x]_{原} = 0,1110$

用 逗号 将符号位和数值部分隔开

$x = -1110$　　$[x]_{原} = 2^4 + 1110 = 1,1110$

带符号的绝对值表示

小数

$$[x]_{原} = \begin{cases} x & 1 > x \geq 0 \\ 1 - x & 0 \geq x > -1 \end{cases}$$

x 为真值

如

$x = +0.1101 \qquad [x]_{原} = 0.1101$

用 小数点 将符号位和数值部分隔开

$x = -0.1101 \qquad [x]_{原} = 1-(-0.1101) = 1.1101$

$x = +0.1000000 \qquad [x]_{原} = 0.1000000$

用 小数点 将符号位和数值部分隔开

$x = -0.1000000 \qquad [x]_{原} = 1-(-0.1000000) = 1.1000000$

(2) 举例

例 6.1 已知 $[x]_{原} = 1.0011$ 求 x -0.0011

解：由定义得

$$x = 1 - [x]_{原} = 1 - 1.0011 = -0.0011$$

例 6.2 已知 $[x]_{原} = 1,1100$ 求 x -1100

解：由定义得

$$x = 2^4 - [x]_{原} = 10000 - 1,1100 = -1100$$

例 6.3　已知 $[x]_{原} = 0.1101$　求 x

解：根据 定义 ∵ $[x]_{原} = 0.1101$

∴　$x = +0.1101$

例 6.4　求 $x = 0$ 的原码

解：设 $x = +0.0000$　　$[+0.0000]_{原} = 0.0000$

$x = -0.0000$　　$[-0.0000]_{原} = 1.0000$

同理，对于整数　　$[+0]_{原} = 0,0000$

$[-0]_{原} = 1,0000$

∴　$[+0]_{原} \neq [-0]_{原}$

原码的特点：简单、直观 6.1

但是用原码作加法时，会出现如下问题：

要求	数1	数2	实际操作	结果符号
加法	正	正	加	正
加法	正	负	减	可正可负
加法	负	正	减	可正可负
加法	负	负	加	负

能否 只作加法？

找到一个与负数等价的正数 来代替这个负数

就可使 减 ——→ 加

3. 补码表示法

(1) 补的概念

- 时钟　　逆时针　　顺时针

$$\begin{array}{r} 6 \\ -\ 3 \\ \hline 3 \end{array} \qquad \begin{array}{r} 6 \\ +\ 9 \\ \hline 15 \\ -\ 12 \\ \hline 3 \end{array}$$

可见 − 3 可用 + 9 代替　减法 ⟶ 加法

称 + 9 是 − 3 以 12 为模的 补数

时钟以12为模

记作 $-3 \equiv +9 \pmod{12}$

同理 $-4 \equiv +8 \pmod{12}$

$-5 \equiv +7 \pmod{12}$

结论

- 一个负数加上 “**模**” 即得该负数的补数
- 一个正数和一个负数互为补数时
  它们绝对值之和即为 **模** 数
  - 计数器（模 16） 1011 ⟶ 0000 ?

$$\begin{array}{r} 1011 \\ -\ 1011 \\ \hline 0000 \end{array} \qquad \begin{array}{r} 1011 \\ +\ 0101 \\ \hline \boxed{1}0000 \end{array}$$

自然去掉

可见 -1011 可用 $+0101$ 代替

记作 $-1011 \equiv +0101 \quad (\mathrm{mod}\ 2^4)$

同理 $-011 \equiv +101 \quad (\mathrm{mod}\ 2^3)$

$-0.1001 \equiv +1.0111 \quad (\mathrm{mod}\ 2)$

(2) 正数的补数即为其本身

两个互为补数的数 $\boxed{-1011 \equiv +0101}$ $(\mathrm{mod}\,2^4)$

分别加上模 $+10000$ $+10000$

结果仍互为补数 $+0101 \equiv +\boxed{1}0101$

丢掉

$\therefore +0101 \equiv +0101 \quad (\mathrm{mod}\,2^4)$

可见 $+0101 \rightarrow +0101$

? $\rightarrow -1011$

? $\boxed{0},0101 \rightarrow +0101$

? $\boxed{1},0101 \rightarrow -1011$

$2^{4+1} - 1011 = 100000 \quad (\mathrm{mod}\,2^{4+1})$

$$\begin{array}{r} 100000 \\ -\ 1011 \\ \hline 1{,}0101 \end{array}$$

用 逗号 将符号位和数值部分隔开

(3) 补码定义

整数

$$[x]_{补} = \begin{cases} 0，x & 2^n > x \geq 0 \\ 2^{n+1} + x & 0 > x \geq -2^n \ (\mathrm{mod}\ 2^{n+1}) \end{cases}$$

x 为真值　　　　n 为整数的位数

如　　$x = +1010$　　　　$x = -1011000$

$[x]_{补} = 0,1010$

用 逗号 将符号位和数值部分隔开

$$[x]_{补} = 2^{7+1} + (-1011000)$$

$$\begin{array}{r} = 100000000 \\ -\quad 1011000 \\ \hline 1,0101000 \end{array}$$

小数

$$[x]_{补} = \begin{cases} x & 1 > x \geq 0 \\ 2 + x & 0 > x \geq -1 \ (\mathrm{mod}\ 2) \end{cases}$$

x 为真值

如 $x = +0.1110$ $x = -0.1100000$

$[x]_{补} = 0.1110$

$$\begin{aligned} [x]_{补} &= 2 + (-0.1100000) \\ &= 10.0000000 \\ &\quad - \ \ 0.1100000 \\ &\quad\overline{\quad 1.0100000} \end{aligned}$$

用 小数点 将符号位

和数值部分隔开

(4) 求补码的快捷方式

设 $x = -1010$ 时

则 $[x]_{补} = 2^{4+1} - 1010 = 11111 + 1 - 1010$

$$\begin{array}{r} 100000 \\ -\ 1010 \\ \hline =\ 1,0110 \end{array} \qquad \begin{array}{r} 11111 + 1 \\ -\ 1010 \quad\ \ \\ \hline \boxed{10101} + 1 \\ = 1,0110 \end{array}$$

又 $[x]_{原} = \boxed{1,1010}$

当真值为 **负** 时，**补码** 可用 **原码除符号位外每位取反，末位加 1 求得**

(5) 举例

例 6.5　已知 $[x]_{补} = 0.0001$

求 x

解：由定义得　$x = +0.0001$

例 6.6　已知 $[x]_{补} = 1.0001$　　$[x]_{补} \xrightarrow{?} [x]_{原}$

求 x　　$[x]_{原} = 1.1111$

解：由定义得　　$\therefore x = -0.1111$

$$x = [x]_{补} - 2$$

$$= 1.0001 - 10.0000$$

$$= -0.1111$$

例 6.7　已知 $[x]_{补} = 1,1110$ 　　　　6.1

求 x

解：　由定义得　　　$[x]_{补} \xrightarrow{?} [x]_{原}$

$x = [x]_{补} - 2^{4+1}$　　　$[x]_{原} = 1,0010$

$= 1,1110 - 100000$　　　$\therefore\ x = -0010$

$= -0010$

当真值为负时，原码可用补码除符号位外每位取反，末位加 1 求得

练习 求下列真值的补码 6.1

真值		$[x]_{补}$	$[x]_{原}$
$x = +70$	$= 1000110$	0, 1000110	0,1000110
$x = -70$	$= -1000110$	1, 0111010	1,1000110
$x = 0.1110$		0.1110	0.1110
$x = -0.1110$		1.0010	1.1110
$x = \boxed{0.0000}$	$[+0]_{补} = [-0]_{补}$	$\boxed{0.0000}$	0.0000
$x = \boxed{-0.0000}$		$\boxed{0.0000}$	1.0000
$x = -1.0000$		1.0000	不能表示

由小数补码定义 $[x]_{补} = \begin{cases} x & 1 > x \geq 0 \\ 2 + x & 0 > x \geq -1 \quad (\mathrm{mod}\ 2) \end{cases}$

$[-1]_{补} = 2 + x = 10.0000 - 1.0000 = 1.0000$

4. 反码表示法

(1) 定义

整数

$$[x]_{反} = \begin{cases} 0,\ x & 2^n > x \geq 0 \\ (2^{n+1} - 1) + x & 0 \geq x > -2^n\ (\mathrm{mod}\ 2^{n+1} - 1) \end{cases}$$

x 为真值　　　　n 为整数的位数

如　$x = +1101$　　　　$x = -1101$

$[x]_{反} = 0,1101$　　　　$[x]_{反} = (2^{4+1} - 1) - 1101$

$= 11111 - 1101$

$= 1,0010$

用 逗号 将符号位

和数值部分隔开

小数

$$[x]_{反} = \begin{cases} x & 1 > x \geq 0 \\ (2 - 2^{-n}) + x & 0 \geq x > -1 \ (\mathrm{mod}\ 2 - 2^{-n}) \end{cases}$$

x 为真值　　n 为小数的位数

如

$x = +0.1101$　　　　$x = -0.1010$

$[x]_{反} = 0.1101$　　　　$[x]_{反} = (2 - 2^{-4}) - 0.1010$

$= 1.1111 - 0.1010$

$= 1.0101$

用 小数点 将符号位
和数值部分隔开

(2) 举例

例6.8　已知 $[x]_{反} = 0,1110$　求 x

解：　由定义得　$x = +1110$

例6.9　已知 $[x]_{反} = 1,1110$　求 x

解：　由定义得　$x = [x]_{反} - (2^{4+1} - 1)$

$= 1,1110 - 11111$

$= -0001$

例 6.10　求 0 的反码

解：　设 $x = +0.0000$　$[+0.0000]_{反} = 0.0000$

$x = -0.0000$　$[-0.0000]_{反} = 1.1111$

同理，对于整数　$[+0]_{反} = 0,0000$　$[-0]_{反} = 1,1111$

$\therefore\ [+0]_{反} \neq [-0]_{反}$

三种机器数的小结

- 最高位为符号位，书写上用“,”（整数）或“.”（小数）将数值部分和符号位隔开
- 对于正数，原码 = 补码 = 反码
- 对于负数，符号位为 1，其数值部分
  原码除符号位外每位取反末位加 1 ⟶ 补码
  原码除符号位外每位取反 ⟶ 反码

例6.11 设机器数字长为 8 位（其中 1 位为符号位）对于整数，当其分别代表无符号数、原码、补码和反码时，对应的真值范围各为多少？

二进制代码	无符号数对应的真值	原码对应的真值	补码对应的真值	反码对应的真值
00000000	0	+0	±0	+0
00000001	1	+1	+1	+1
00000010	2	+2	+2	+2
⋮	⋮	⋮	⋮	⋮
01111111	127	+127	+127	+127
10000000	128	-0	-128	-127
10000001	129	-1	-127	-126
⋮	⋮	⋮	⋮	⋮
11111101	253	-125	-3	-2
11111110	254	-126	-2	-1
11111111	255	-127	-1	-0

例6.12　已知 $[y]_{补}$　　求 $[-y]_{补}$

解：设 $[y]_{补} = y_0 . y_1 y_2 \cdots y_n$

<I>　$\boxed{[y]_{补} = 0. y_1 y_2 \cdots y_n}$

$[y]_{补}$连同符号位在内，每位取反，末位加 1

即得$[-y]_{补}$

$$\boxed{[-y]_{补} = 1.\overline{y_1}\,\overline{y_2} \cdots \overline{y_n} + 2^{-n}}$$

<II>　$\boxed{[y]_{补} = 1. y_1 y_2 \cdots y_n}$

$[y]_{补}$连同符号位在内，每位取反，末位加 1

即得$[-y]_{补}$

$$\boxed{[-y]_{补} = 0.\overline{y_1}\,\overline{y_2} \cdots \overline{y_n} + 2^{-n}}$$

5. 移码表示法

补码表示很难直接判断其真值大小

如 十进制	二进制	补码	
$x = +21$	$+10101$	$0,10101$	错
$x = -21$	-10101	$1,01011$	大
$x = +31$	$+11111$	$0,11111$	错
$x = -31$	-11111	$1,00001$	大

$x + 2^5$

$+10101 + 100000 = 110101$ 大

$-10101 + 100000 = 001011$ 正确

$+11111 + 100000 = 111111$ 大

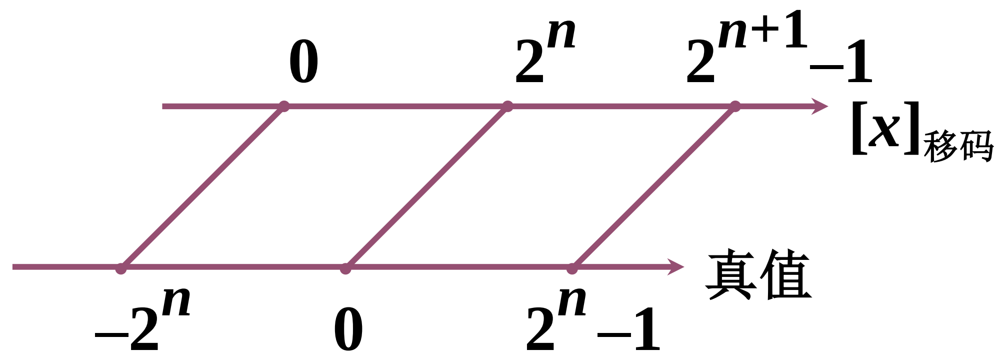
$-11111 + 100000 = 000001$ 正确

(1) 移码定义

$$[x]_{移} = 2^n + x \quad (2^n > x \geq -2^n)$$

x 为真值，n 为 整数的位数

移码在数轴上的表示



如 $x = 10100$

$$[x]_{移} = 2^5 + 10100 = 1,10100$$

$$x = -10100$$

$$[x]_{移} = 2^5 - 10100 = 0,01100$$

用 逗号 将符号位和数值部分隔开

(2) 移码和补码的比较

设 $x = +1100100$

$[x]_{移} = 2^7 + 1100100 = 1,1100100$

$[x]_{补} = 0,1100100$

设 $x = -1100100$

$[x]_{移} = 2^7 - 1100100 = 0,0011100$

$[x]_{补} = 1,0011100$

补码与移码只差一个符号位

(3) 真值、补码和移码的对照表

真值 x (n=5)	$[x]_补$	$[x]_移$	$[x]_移$对应的十进制整数
- 1 0 0 0 0 0	1 0 0 0 0 0	0 0 0 0 0 0	0
- 1 1 1 1 1	1 0 0 0 0 1	0 0 0 0 0 1	1
- 1 1 1 1 0	1 0 0 0 1 0	0 0 0 0 1 0	2
⋮	⋮	⋮	⋮
- 0 0 0 0 1	1 1 1 1 1 1	0 1 1 1 1 1	31
± 0 0 0 0 0	0 0 0 0 0 0	1 0 0 0 0 0	32
+ 0 0 0 0 1	0 0 0 0 0 1	1 0 0 0 0 1	33
+ 0 0 0 1 0	0 0 0 0 1 0	1 0 0 0 1 0	34
⋮	⋮	⋮	⋮
+ 1 1 1 1 0	0 1 1 1 1 0	1 1 1 1 1 0	62
+ 1 1 1 1 1	0 1 1 1 1 1	1 1 1 1 1 1	63

(4) 移码的特点

- 当 $x = 0$ 时 $[+0]_{移} = 2^5 + 0 = 1,00000$

  $[-0]_{移} = 2^5 - 0 = 1,00000$

  $\therefore [+0]_{移} = [-0]_{移}$

- 当 $n = 5$ 时 最小的真值为 $-2^5 = -100000$

  $[-100000]_{移} = 2^5 - 100000 = 000000$

  可见，最小真值的移码为全 0

  用移码表示浮点数的阶码

  能方便地判断浮点数的阶码大小

6.2 数的定点表示和浮点表示

小数点按约定方式标出

一、定点表示

$\underset{\text{数符}}{S_f}\ \underbrace{S_1S_2\cdots S_n}_{\text{数值部分}}$　　或　　$\underset{\text{数符}}{S_f}\ \underbrace{S_1S_2\cdots S_n}_{\text{数值部分}}$

小数点位置　　　　　　　　　　　　小数点位置

定点机	小数定点机	整数定点机
原码	$-(1-2^{-n}) \sim +(1-2^{-n})$	$-(2^n-1) \sim +(2^n-1)$
补码	$-1 \sim +(1-2^{-n})$	$-2^n \sim +(2^n-1)$
反码	$-(1-2^{-n}) \sim +(1-2^{-n})$	$-(2^n-1) \sim +(2^n-1)$

二、浮点表示

6.2

$N = S \times r^{j}$　　浮点数的一般形式

S 尾数　j 阶码　r 基数（基值）

计算机中 r 取 2、4、8、16 等

当 $r = 2$　　$N = 11.0101$

二进制表示

$\checkmark = 0.110101 \times 2^{\boxed{10}}$　规格化数

$= 1.10101 \times 2^{1}$

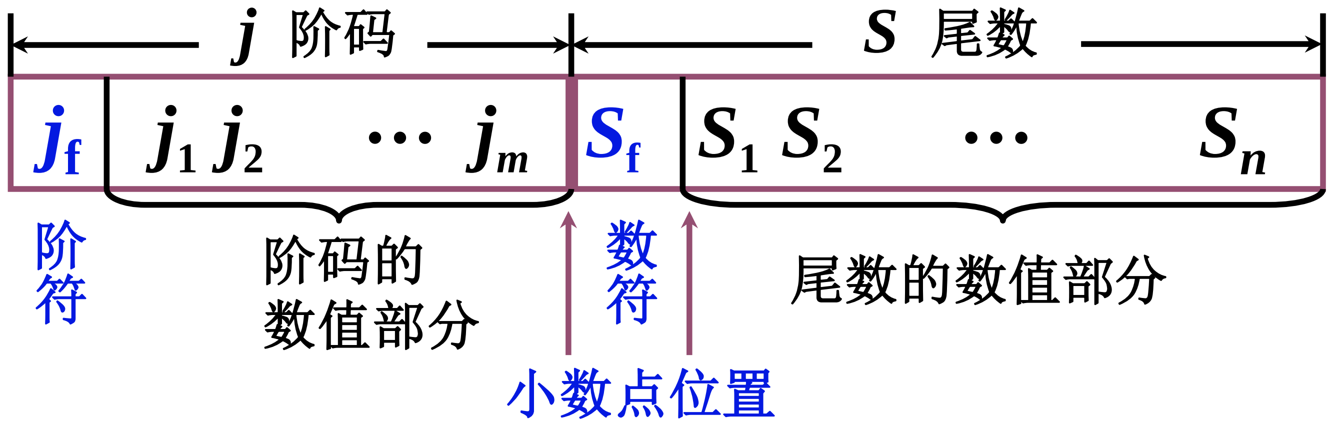
$= 1101.01 \times 2^{-10}$

$\checkmark = 0.00110101 \times 2^{100}$

计算机中　S　小数、可正可负

　　　　　j　整数、可正可负

1. 浮点数的表示形式



S_f 代表浮点数的符号

n 其位数反映浮点数的精度

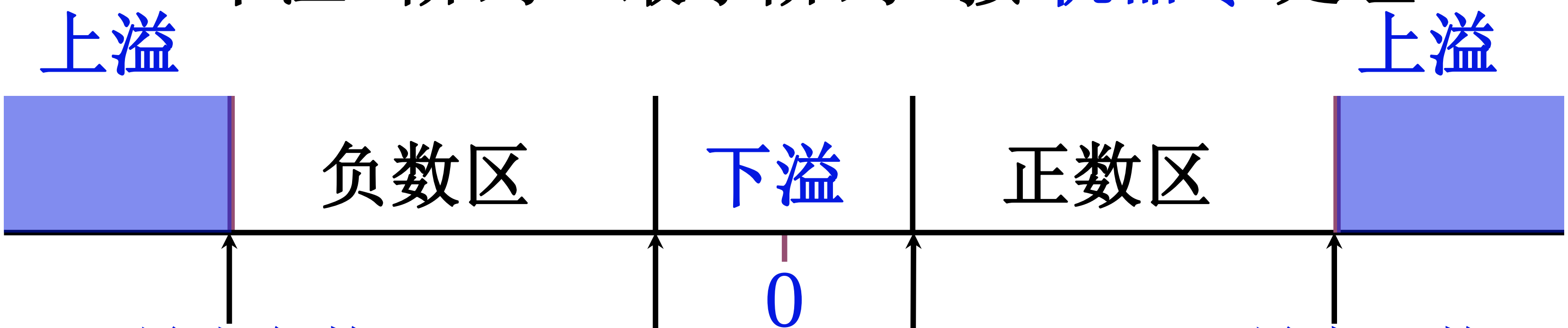
m 其位数反映浮点数的表示范围

j_f 和 m 共同表示小数点的实际位置

2. 浮点数的表示范围 (以原码为例)　　6.2

上溢　阶码 > 最大阶码

下溢　阶码 < 最小阶码　按 机器零 处理



最小负数

$-2^{(2^m-1)} \times (1-2^{-n})$

$-2^{15} \times (1-2^{-10})$

最大负数

$-2^{-(2^m-1)} \times 2^{-n}$

$-2^{-15} \times 2^{-10}$

最小正数

$2^{-(2^m-1)} \times 2^{-n}$

$2^{-15} \times 2^{-10}$

最大正数

$2^{(2^m-1)} \times (1-2^{-n})$

$2^{15} \times (1-2^{-10})$

设 $m = 4$

$n = 10$

练习 6.2

设机器数字长为 24 位，欲表示±3万的十进制数，试问在保证数的最大精度的前提下，除阶符、数符各取1 位外，阶码、尾数各取几位？

解： $\because \quad 2^{14} = 16384 \qquad 2^{15} = 32768$

$\therefore$ 如果是定点数15 位二进制数可反映 ±3 万之间的十进制数

$$2^{\boxed{15}} \times 0.\underbrace{\times\times\times \;\cdots\; \times\times\times}_{n\text{位}}$$

$$m = 4，\ 5，\ 6，\cdots$$

满足 最大精度 可取 $m = 4$，$n = 18$

3. 浮点数的规格化形式

6.2

$r = 2$ 尾数最高位为 1

$r = 4$ 尾数最高 2 位不全为 0

$r = 8$ 尾数最高 3 位不全为 0

基数不同，浮点数的规格化形式不同

4. 浮点数的规格化

$r = 2$	左规	尾数左移 1 位，阶码减 1
	右规	尾数右移 1 位，阶码加 1
$r = 4$	左规	尾数左移 2 位，阶码减 1
	右规	尾数右移 2 位，阶码加 1
$r = 8$	左规	尾数左移 3 位，阶码减 1
	右规	尾数右移 3 位，阶码加 1

基数 r 越大，可表示的浮点数的范围越大

基数 r 越大，浮点数的精度降低

例如：设 $m = 4$，$n = 10$，$r = 2$

尾数规格化后的浮点数表示范围

最大正数 $2^{+1111} \times 0.\underbrace{1111111111}_{10个1} = 2^{15} \times (1-2^{-10})$

最小正数 $2^{-1111} \times 0.1\underbrace{000000000}_{9个0} = 2^{-15} \times 2^{-1} = 2^{-16}$

最大负数 $2^{-1111} \times (-0.1\underbrace{000000000}_{9个0}) = -2^{-15} \times 2^{-1} = -2^{-16}$

最小负数 $2^{+1111} \times (-0.\underbrace{1111111111}_{10个1}) = -2^{15} \times (1-2^{-10})$

三、举例

例 4.13　将 $+\frac{19}{128}$ 写成二进制定点数、浮点数及在定点机和浮点机中的机器数形式。其中数值部分均取 10 位，数符取 1 位，浮点数阶码取 5 位（含1位阶符）。

解：　设 $x = +\frac{19}{128}$

二进制形式　$x = 0.0010011$

定点表示　$x = 0.0010011\,000$

浮点规格化形式　$x = 0.1001100000 \times 2^{-10}$

定点机中　$[x]_{原} = [x]_{补} = [x]_{反} = 0.0010011000$

浮点机中　$[x]_{原} = 1, 0010;\ \ 0.1001100000$

$[x]_{补} = 1, 1110;\ \ 0.1001100000$

$[x]_{反} = 1, 1101;\ \ 0.1001100000$

例 4.14 将 –58 表示成二进制定点数和浮点数，并写出它在定点机和浮点机中的三种机器数及阶码为移码、尾数为补码的形式（其他要求同上例）。

解：　设 $x = -58$

二进制形式　$x = -111010$

定点表示　$x = -0000\,111010$

浮点规格化形式　$x = -(0.1110100000) \times 2^{110}$

定点机中

$[x]_{原} = 1, 0000111010$

$[x]_{补} = 1, 1111000110$

$[x]_{反} = 1, 1111000101$

浮点机中

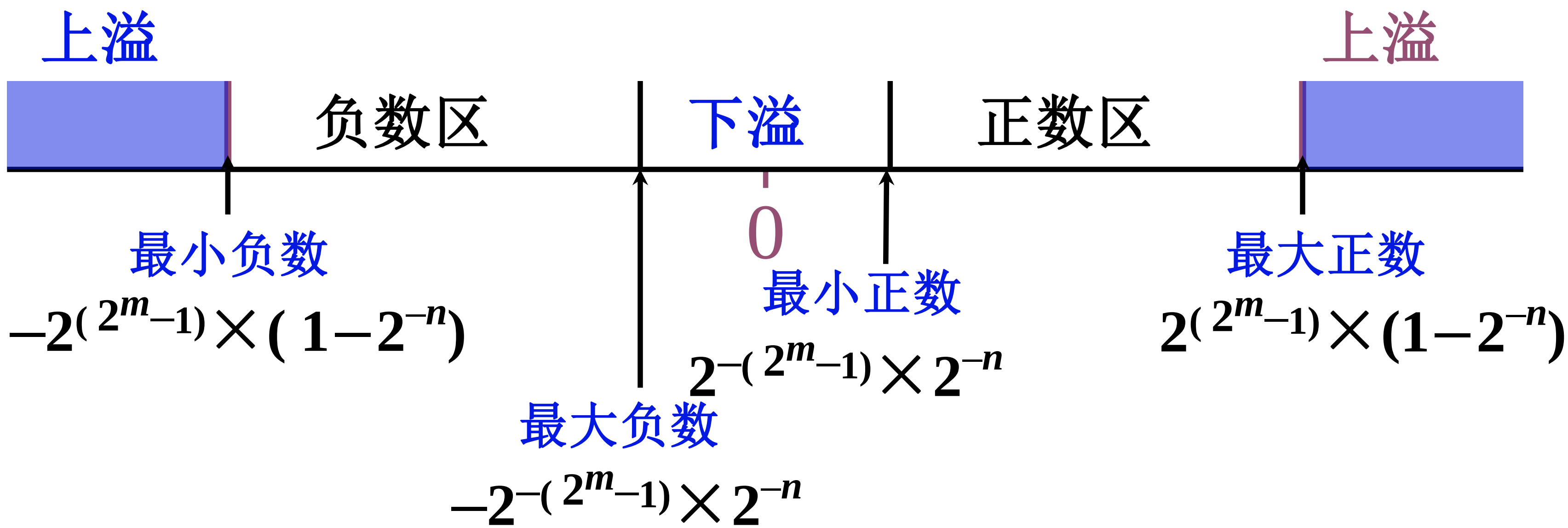
$[x]_{原} = 0, 0110; 1.\,1110100000$

$[x]_{补} = 0, 0110; 1.\,0001100000$

$[x]_{反} = 0, 0110; 1.\,0001011111$

$[x]_{阶移、尾补} = 1, 0110; 1.\,0001100000$

例4.15 写出对应下图所示的浮点数的补码形式。设 $n = 10$，$m = 4$，阶符、数符各取1位。 6.2



解：

	真值	补码
最大正数	$2^{15}\times(1-2^{-10})$	0,1111; 0.1111111111
最小正数	$2^{-15}\times 2^{-10}$	1,0001; 0.0000000001
最大负数	$-2^{-15}\times 2^{-10}$	1,0001; 1.1111111111
最小负数	$-2^{15}\times(1-2^{-10})$	0,1111; 1.0000000001

机器零

6.2

- 当浮点数 尾数为 0 时，不论其阶码为何值按机器零处理
- 当浮点数 阶码等于或小于它所表示的最小数 时，不论尾数为何值，按机器零处理

如 $m = 4$ $n = 10$

当阶码和尾数都用补码表示时，机器零为

×, × × × ×； 0. 0 0 ··· 0

（阶码 = −16） 1, 0 0 0 0； ×.× × ··· ×

当阶码用移码，尾数用补码表示时，机器零为

0, 0 0 0 0； 0. 0 0 ··· 0

有利于机器中“判 0”电路的实现

四、IEEE 754 标准

6.2

S	阶码（含阶符）	尾　　数

数符　　小数点位置

尾数为规格化表示

非“0”的有效位最高位为“1”（隐含）

	符号位 *S*	阶码	尾数	总位数
短实数	1	8	23	32
长实数	1	11	52	64
临时实数	1	15	64	80