计算机组成原理

第 3 讲

左德承

哈尔滨工业大学计算学部

容错与移动计算研究中心

原码的特点：简单、直观

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 $\longrightarrow$ 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 $-1011 \equiv +0101 \quad (\bmod 2^4)$

同理 $-011 \equiv +101 \quad (\bmod 2^3)$

$-0.1001 \equiv +1.0111 \quad (\bmod 2)$

(2) 正数的补数即为其本身

2.1

两个互为补数的数 $\boxed{-1011 \equiv +0101}$ （mod2^4）

分别加上模 $+10000$ $\quad$ $+10000$

结果仍互为补数 $+0101 \equiv +\boxed{1}0101$

丢掉

$\therefore +0101 \equiv +0101$ （mod2^4）

可见 $+0101 \rightarrow +0101$

? $\rightarrow -1011$

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

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

$2^{4+1} - 1011 = 100000$ （mod2^{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

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

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

$$[x]_补 = 2^{7+1} + (-1011000) = 100000000 - 1011000 = 1,0101000$$

小数

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

x 为真值

如　　$x = +0.1110$　　　　$x = -0.1100000$

$[x]_{补} = 0.1110$　　　　$[x]_{补} = 2 + (-0.1100000)$

$$\begin{array}{r} = 10.0000000 \\ -\ \ 0.1100000 \\ \hline 1.0100000 \end{array}$$

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

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

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 $[x]_{补} = 1.0001$

求 x

解：由定义得

$$x = [x]_{补} - 2$$
$$= 1.0001 - 10.0000$$
$$= -0.1111$$

$[x]_{补} \xrightarrow{?} [x]_{原}$

$[x]_{原} = 1.1111$

$\therefore x = -0.1111$

例 6.7　已知 $[x]_{补} = 1,1110$

求 x

解：由定义得

$$x = [x]_{补} - 2^{4+1}$$

$$= 1,1110 - 100000$$

$$= -0010$$

$$[x]_{补} \xrightarrow{?} [x]_{原}$$

$$[x]_{原} = 1,0010$$

$$\therefore x = -0010$$

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　2.1

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

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

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

4. 反码表示法

(1) 定义

整数

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

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

正确

$-10101 + 100000 = 001011$

$+11111 + 100000 = 111111$ 大

正确

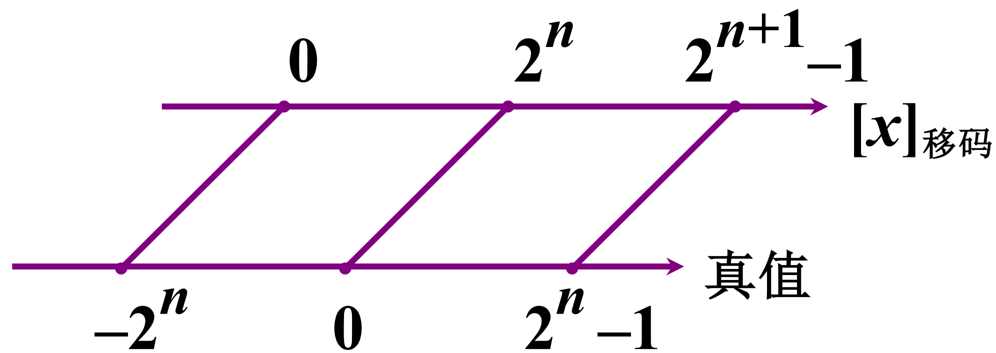
$-11111 + 100000 = 000001$

(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

2.2 数的定点表示和浮点表示

2.2 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

S_f	$S_1S_2\cdots S_n$
数符	数值部分

小数点位置

或

S_f	$S_1S_2\cdots S_n$
数符	数值部分

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

$N = S \times r^{j}$　　　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

$= 1.10101 \times 2^{1}$

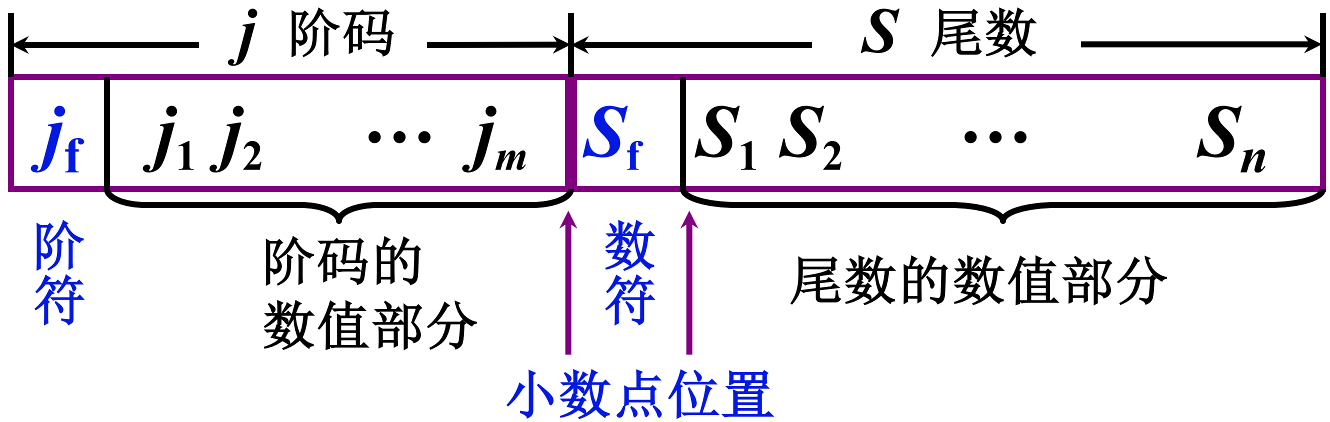
$= 1101.01 \times 2^{-10}$

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

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