计 算 机 组 成 原 理 — 数 字

刘宏伟

哈尔滨工业大学

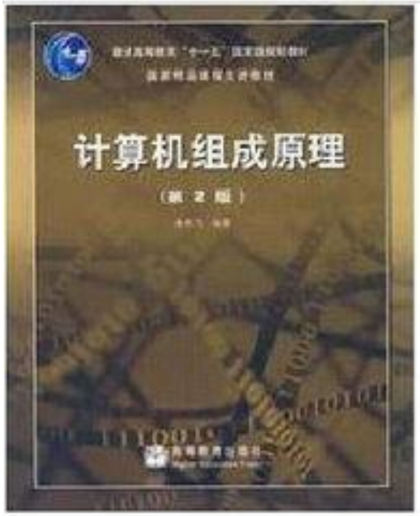
计算机科学与技术学院

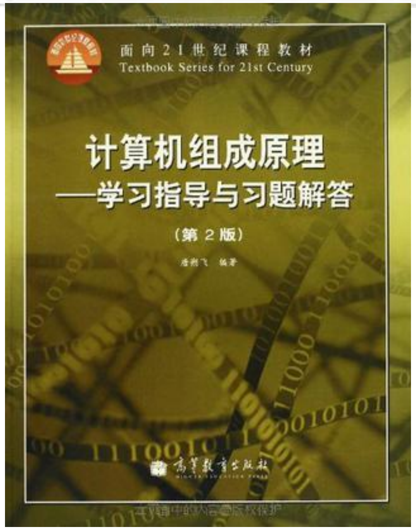
主要内容

- 1）计算机中数的表示
- 2）计算机的运算方法
- 3）运算器的设计

教材：计算机组成原理（第2版）

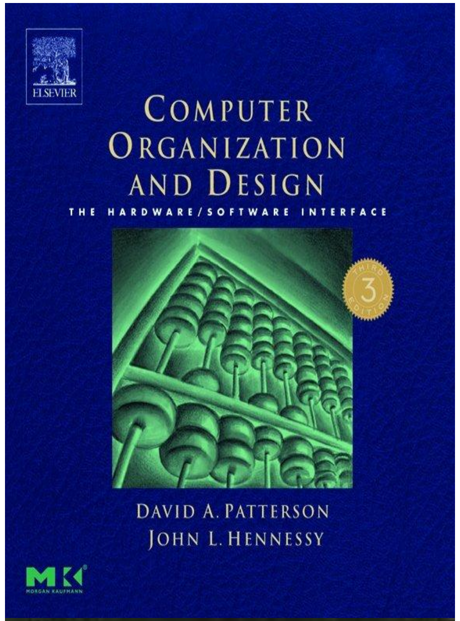
- 普通高等教育“十二五”规划教材
- 面向21世纪课程教材
- 全国普通高等学校优秀教材二等奖
- 普通高等教育精品教材





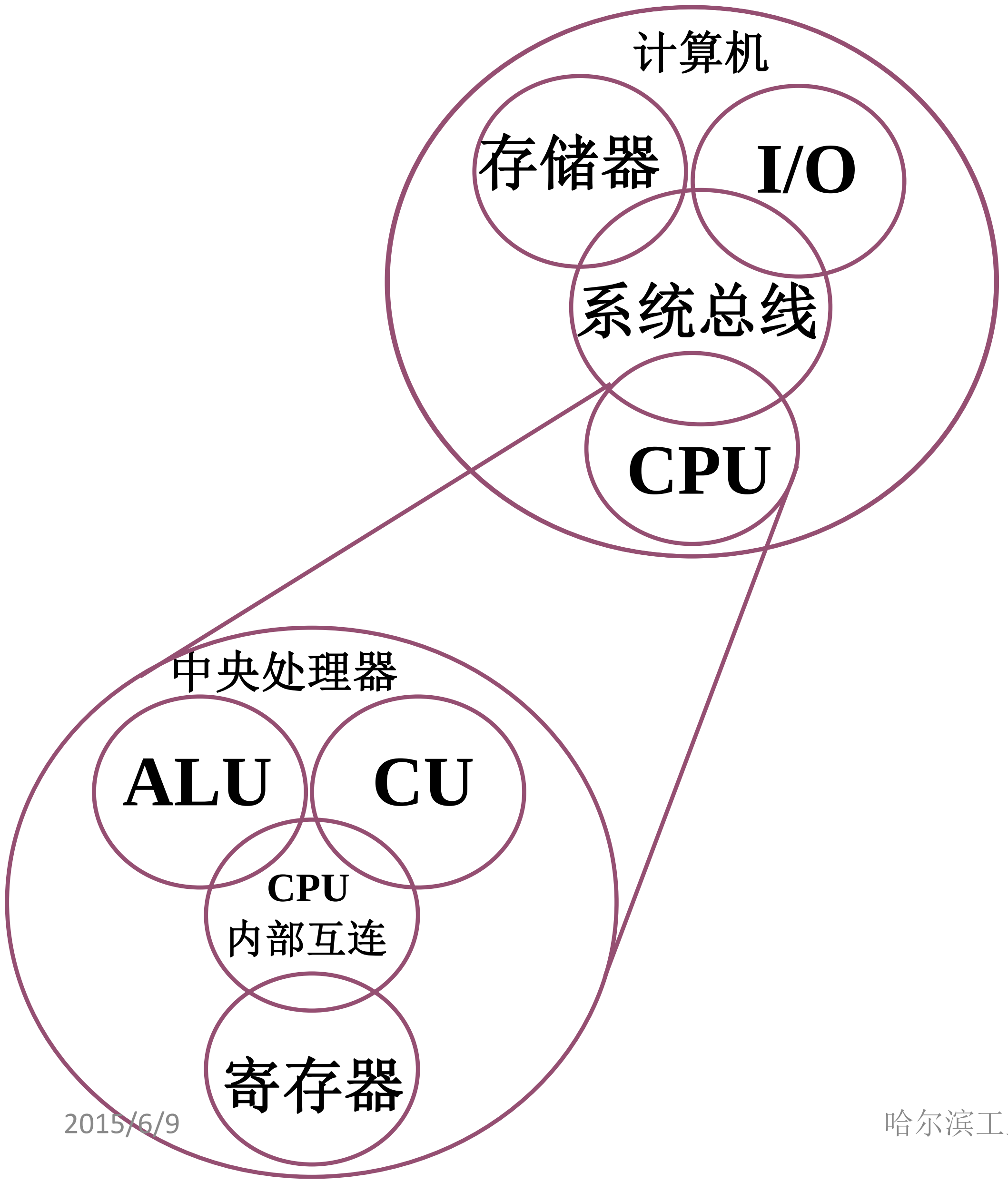
参考教材

- **David A.Patterson. John L.Hennessy. Computer Organization & Design: A Hardware/Software Interface**



- **David Harris, Sarah Harris. Digital Design and Computer Architecture. Morgan Kaufmann, 2007**

课程内容的组织



第 3 篇 CPU

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
- 二、有符号数
  - 1、机器数与真值
  - 2、原码表示法
  - 3、补码表示法
  - 4、反码表示法
  - 5、移码表示法

✓ 定义

✓ 特点

✓ 举例

➢ 机器数与真值的转换

➢ 不同机器数形式之间的转化

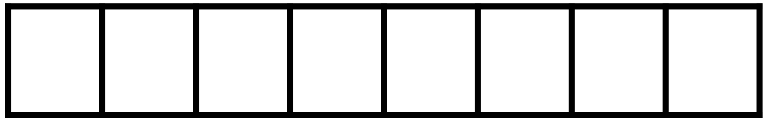
✓ 机器数表示的范围与其字长有关

6.1 无符号数和有符号数

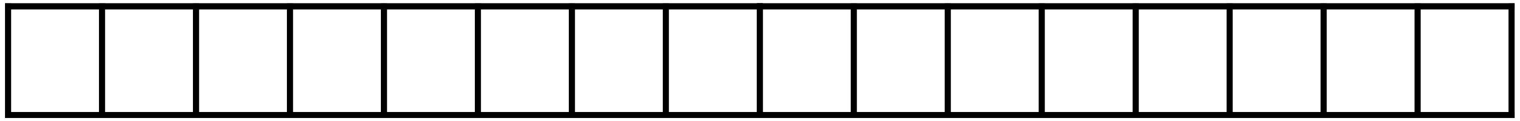
一、无符号数

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	小数定点机	0 ‖ 1011	没有硬件表明小数点的位置，都是约定位置 小数点的位置
– 0.1011		1 ‖ 1011	小数点的位置
+ 1100	整数定点机	0 ‖ 1100	小数点的位置
– 1100		1 ‖ 1100	小数点的位置

2. 原码表示法

(1) 定义

整数

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

最高位为1，多了2^n

在0处出现两种表示

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

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

小数形式的0也可能有两种表示

x 为真值

表示的是符号

如 $x = +0.1101$　　$[x]_{原} = 0\,.\,1101$

用 小数点 将符号位和数值部分隔开

$x = -0.1101$　　$[x]_{原} = 1-(-0.1101) = 1\,.\,1101$

$x = +0.1000000$　　$[x]_{原} = 0\,.\,1000000$

用 小数点 将符号位和数值部分隔开

$x = -0.1000000$　　$[x]_{原} = 1-(-0.1000000) = 1\,.\,1000000$

(2) 举例

6.1

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

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

6.1

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

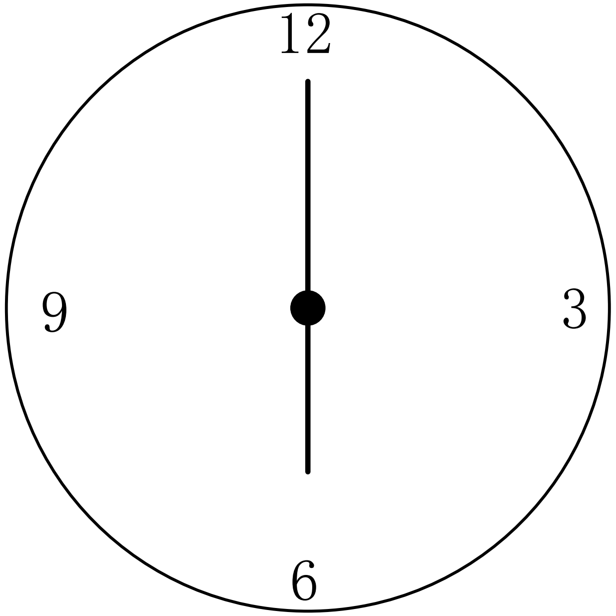
能否 只作加法？

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

6.1



(1) 补的概念

- 时钟

逆时针

$$\begin{array}{r} 6 \\ -\ 3 \\ \hline 3 \end{array}$$

顺时针

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

时钟以12为模

可见 -3 可用 $+9$ 代替 减法 ⟶ 加法

称 $+9$ 是 -3 以 12 为模的 补数

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） $1011 \longrightarrow 0000$ ？

4位计数器

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

自然去掉

可见 -1011 可用 $+0101$ 代替

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

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

为了扔掉最高位

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

(2) 正数的补数即为其本身

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

分别加上模 $+10000$ $\quad$ $+10000$

结果仍互为补数 $+0101 \equiv +\boxed{1}0101$

丢掉

$\therefore +0101 \equiv +0101 \quad (\bmod 2^4)$

可见 $+0101 \rightarrow +0101$

? $\rightarrow -1011$

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

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

$2^{4+1} - 1011 = 100000$

-1011

$1,0101$

$(\bmod 2^{4+1})$

用 逗号 将符号位和数值部分隔开

(3) 补码定义

整数

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

0的补码表示只有一种

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

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

mod2，小数点前1位；
mod4，小数点前2位；
mod8，小数点前3位；

x 为真值

考虑计算机中定点小数的位数问题

如 $x = +0.1110$ $x = -0.1100000$

$[x]_{补} = 0.1110$

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

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

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

又 $[x]_{原} = \boxed{1,1010}$

当真值为 负 时，补码 可用 原码除符号位外

符号位不变！！！

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

$x = [x]_{补} - 2^{4+1}$

$= 1,1110 - 100000$

$= -0010$

$[x]_{补} \xrightarrow{?} [x]_{原}$

$[x]_{原} = 1,0010$

$\therefore x = -0010$

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 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

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

补码表示可以把减法转化为加法

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

4. 反码表示法 末位不+1了

6.1

(1) 定义

整数

正数的时候不变

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

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 6.1

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

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）**6.1**

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$[-y]_{补}$　　6.1

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

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

$[y]_{补}$连同符号位在内，　每位取反，　末位加 1

即得$[-y]_{补}$

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

< 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

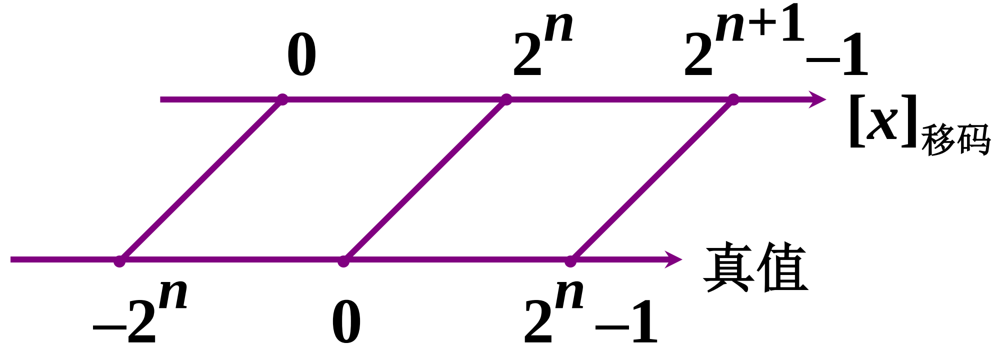
$-11111 + 100000 = 000001$ 正确

(1) 移码定义 只有整数的定义，没有小数的定义

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

x 为真值，n 为 **整数的位数**

移码在数轴上的表示



如 $x = 10100$

$[x]_{移} = 2^5 + 10100 = 1,10100$

1代表正数，0代表负数

用 **逗号** 将符号位和数值部分隔开

$x = -10100$

$[x]_{移} = 2^5 - 10100 = 0,01100$

(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 $[+0]_{移} = 2^5 + 0 \quad = 1,00000$

  $[-0]_{移} = 2^5 - 0 \quad = 1,00000$

  $\therefore\ [+0]_{移} = [-0]_{移}$

- 当 $n = 5$ 时 最小的真值为 $-2^5 = -100000$

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

  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