计算机组成

第六章　计算机的运算方法

Contents

大 纲

(一)数制与编码

1. 进位计算制及其数据之间的相互转换
2. 定点数的编码表示

(二)运算方法和运算电路

1. 基本运算部件

   加法器，算术逻辑部件（ALU）

2. 加/减法运算

   补码加减法运算器，

   标志位的生成

3. 乘除法运算的基本原理，

   乘法电路和除法电路的基本结构

(三)整数的表示和运算

1. 无符号整数的表示和运算
2. 带符号整数的表示和运算

(四) 浮点数的表示和运算

1. 浮点数的表示

   IEEE754标准

2. 浮点数的加/减运算

数制与编码

1. 进位计算制及其相 互转换

1). 基r数制：用r个基本符号（如0,1,2,…,r-1）表示数值N。r为 基数

$$N=D_{m-1}D_{m-2}\ \dots\ D_1\ D_0\ D_{-1}\ D_{-2}\ \dots\ D_{-k}$$

其中， D_i（$-k \le i \le m-1$）为基本符号，小数点位置隐含在D_0与D_{-1}之间。

2). 有权基r数制：每个D_i的单位都赋以固定权值w_i，w_i为D_i的权。

如果该数制是逢r进位，则有

$$N = \sum_{i=-k}^{m-1} D_i \times r^i$$

其中r^i为位权，称该数制为r进位数制，简称r进制，

3). 进制转换

(1) R进制数 => 十进制数

按“权”展开 (a power of R)

例1: $(10101.01)_2=1\times2^4+1\times2^2+1\times2^0+1\times2^{-2}=(21.25)_{10}$

例2: $(307.6)_8=3\times8^2+7\times8^0+6\times8^{-1}=(199.75)_{10}$

例1: $(3A.1)_{16}=3\times16^1+10\times16^0+1\times16^{-1}=(58.0625)_{10}$

(2)十进制数 => R进制数

整数部分和小数部分分别转换

① 整数(integral part)----“除基取余，上右下左”

② 小数(fractional part)----“乘基取整，上左下右”

二、十进制转换

例1:$(835.6785)_{10}=(1101000011.1011)_2$

整数----“除基取余，上右下左”

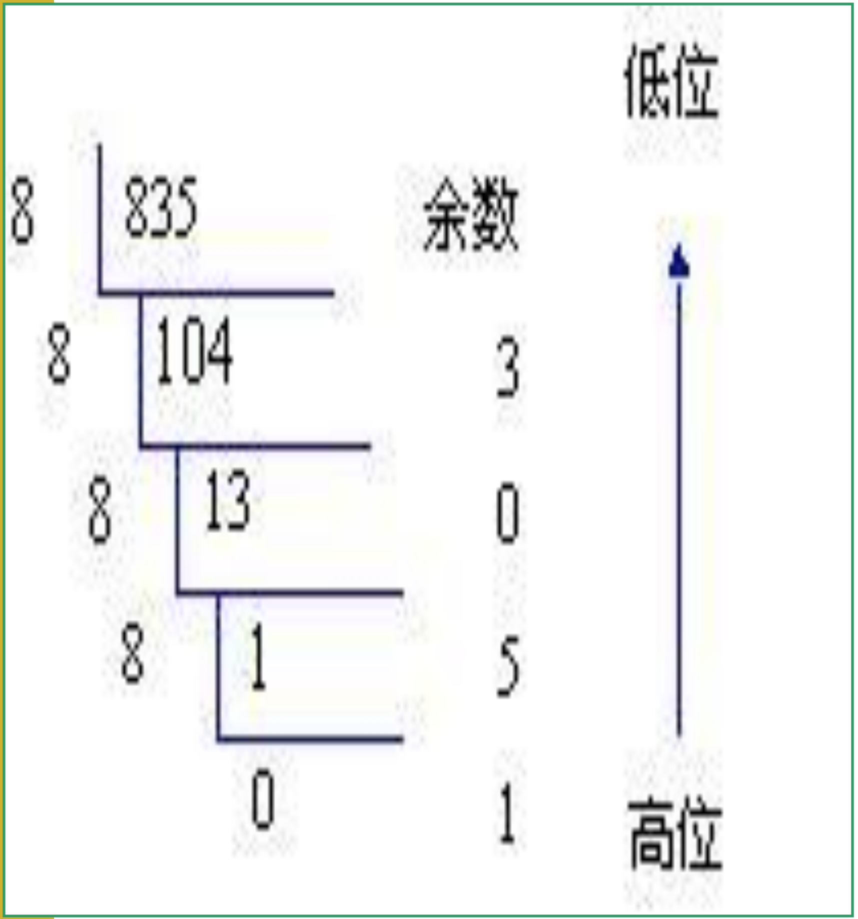
		余数	低位
2	835	1	↑
2	417	1	
2	208	0	
2	104	0	
2	52	0	
2	26	0	
2	13	1	
2	6	0	
2	3	1	
2	1	1	
	0		高位

小数----“乘基取整，上左下右”

0.6875×2=1.375	整数部分=1	(高位)
0.375×2=0.75	整数部分=0	↓
0.75×2=1.5	整数部分=1	↓
0.5×2=1.0	整数部分=1	(低位)

例2:$(835.63)_{10}=(1503.50243\ldots)_8$

整数----“除基取余，上右下左”小数----“乘基取整，上左下右”



有可能乘积的小数部分总得不到0 ，此时得到一个近似值。

$0.63\times 8=5.04$　整数部分=5　(高位)

$0.04\times 8=0.32$　整数部分=0

$0.32\times 8=2.56$　整数部分=2

$0.56\times 8=4.48$　整数部分=4

$0.48\times 8=3.84$　整数部分=3　(低位)

(3) 二/八/十六进制数的相互转换

① 八进制数转换成二进制数

$(13.724)_8 = (\ 001\ 011\ .\ 111\ \ 010\ \ 100\ \ )_2 = (1011.1110101)_2$

② 十六进制数转换成二进制数

$(2B.5E)_{16} = (00101011\ .\ 01011110)_2 = (101011.0101111)_2$

③ 二进制数转换成八进制数

$(0.10101)_2 = (\ \ 000\ .\ 101\ \ 010\ \ )_2 = (\ 0.52)_8$

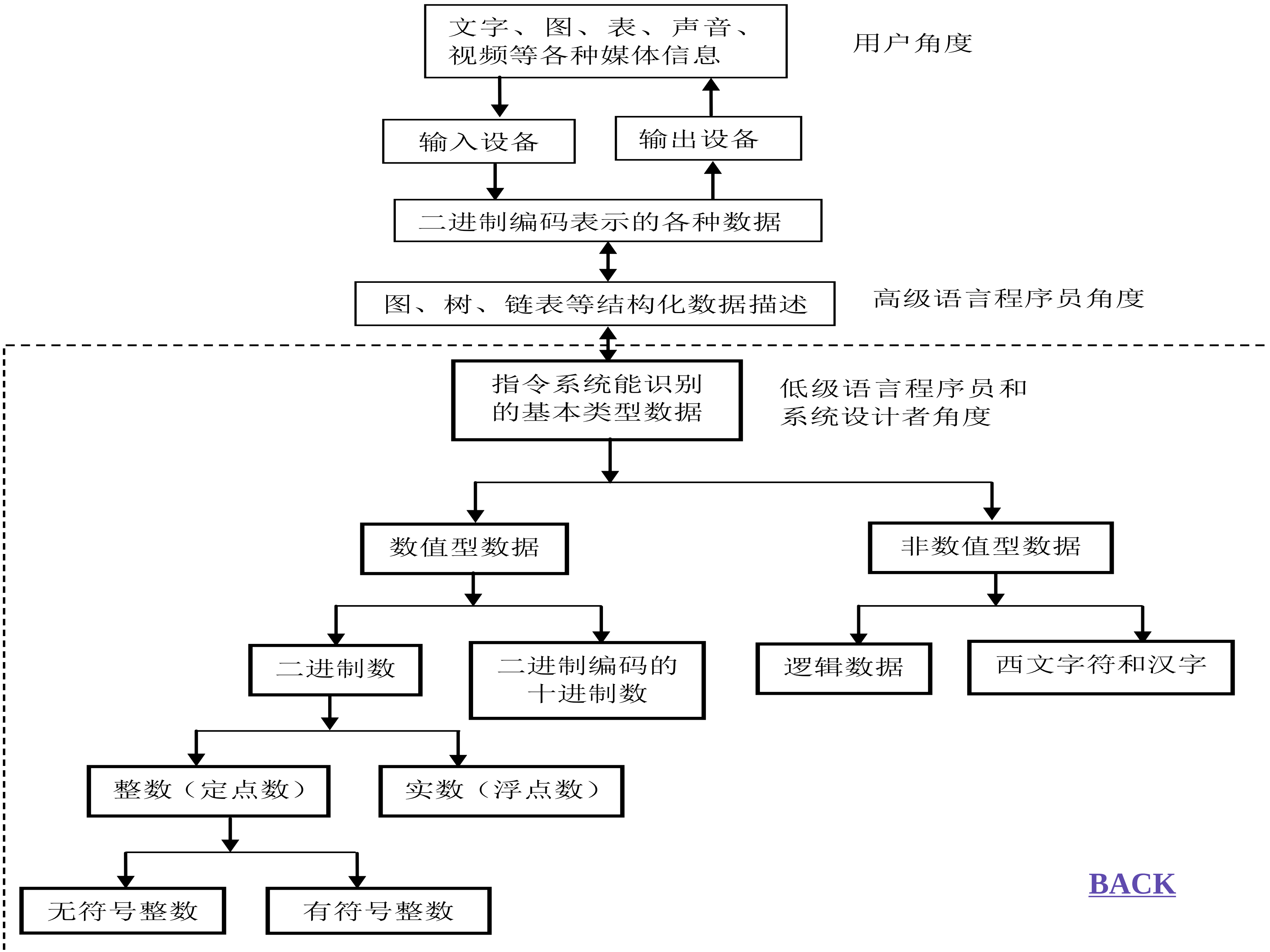
④ 二进制数转换成十六进制数

$(11001.11)_2 = (\ \ 0001\ \ 1001\ .\ 1100\ \ )_2 = (\ 19.C\ )_{16}$

信息的二进制编码

- 计算机的外部信息与内部机器级数据
- 机器级数据分两大类：
  - 数值数据：无符号整数、带符号整数、浮点数（实数）、十进制数
  - 非数值数据：逻辑数（包括位串）、西文字符和汉字
- 计算机内部所有信息都用二进制（即：0和1）进行编码
- 用二进制编码的原因：
  - 制造二个稳定态的物理器件容易
  - 二进制编码、计数、运算规则简单
  - 正好与逻辑命题对应，便于逻辑运算，并可方便地用逻辑电路实现算术运算

数值数据的表示


文字、图、表、声音、
视频等各种媒体信息
用户角度
输入设备
输出设备
二进制编码表示的各种数据
图、树、链表等结构化数据描述
高级语言程序员角度
指令系统能识别
的基本类型数据
低级语言程序员和
系统设计者角度
数值型数据
非数值型数据
二进制数
二进制编码的
十进制数
逻辑数据
西文字符和汉字
整数（定点数）
实数（浮点数）
无符号整数
有符号整数

BACK

数值数据的表示

- 数值数据表示的三要素
  - 进位计数制
  - 定、浮点表示
  - 如何用二进制编码

  即：要确定一个数值数据的值必须先确定这三个要素。

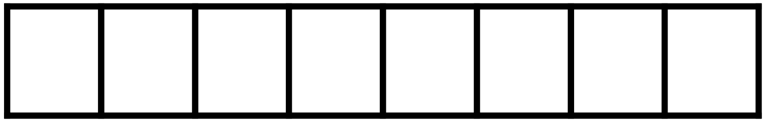
  例如，机器数 01011001的值是多少？ 答案是：不知道！
- 进位计数制
  - 十进制、二进制、十六进制、八进制数及其相互转换
- 定/浮点表示（解决小数点问题）
  - 定点整数、定点小数
  - 浮点数（可用一个定点小数和一个定点整数来表示）
- 定点数的编码（解决正负号问题）
  - 原码、补码、反码、移码 （反码很少用）

6.1 无符号数和有符号数

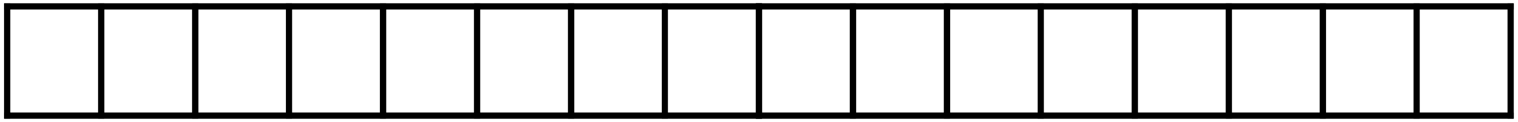
一、无符号数

寄存器的位数（机器字长）

反映无符号数的表示范围



8 位　　　　0 ~ 255



16 位　　　　0 ~ 65535

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

(1) 定义

整数

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

x 为真值　　n 为整数的位数

如　$x = +1110$　　$[x]_{原} = 0,1110$

用 逗号 将符号位和数值位隔开

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

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

$x = +0.1101$　　$[x]_{原} = 0.1101$

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

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

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

(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

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

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

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

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
- 两个互为补数的数 它们绝对值之和即为 模 数
  - 计数器（模 16） 1011 ⟶ 0000 ?

11

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

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 6.1

两个互为补数的数 $-1011 \equiv +0101 \pmod{2^4}$

分别加上模 $+10000$ $+10000$

结果仍互为补数 $+0101 \equiv +\boxed{1}0101$

$\therefore +0101 \equiv +0101 \pmod{2^4}$

丢掉

可见 $+0101 \rightarrow +0101$

? $\rightarrow -1011$

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

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

$2^{4+1} - 1011 = 100000 - 1011 = 1,0101 \pmod{2^{4+1}}$

用 **逗号** 将符号位和数值位隔开

(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** 将符号位和数值位隔开

$$\begin{aligned}[x]_{补} &= 2^{7+1} + (-1011000) \\ &= 100000000 \\ &\quad\ -1011000 \\ \hline &\quad\ 1,0101000\end{aligned}$$

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 $x = +0.1110$ $\qquad$ $x = -0.1100000$

$[x]_{补} = 0.1110$

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

用 小数点 将符号位

和数值位隔开

补码说明

- 补码最高一位是符号位，符号 0 正 1 负
- 补码表示为：2×符号位 + 数的真值
- 零的补码只有一个，故补码能表示 -1
- 补码能很好地用于加减运算，运算时，符号位与数值位一样参加运算。

求特殊数的补码

假定机器数有n位

① $[-2^{n-1}]_{补} = 2^n - 2^{n-1} = 10\ldots0$（n-1个0）　（mod 2^n）

② $[-1]_{补} = 2^n - 0\ldots01 = 11\ldots1$（n个1）　（mod 2^n）　整数补码

③ $[-1.0]_{补} = 2 - 1.0 = 1.00\ldots0$（n-1个0）　（mod 2）　小数补码

④ $[+0]_{补} = [-0]_{补} = 00\ldots0$（n个0）

特殊数的补码（续）

- 当补码的位数为n位时，其模为2^n，所以

$[-2^{n-1}]_{补} = 2^n - 2^{n-1} = 10\ldots0$（n-1个0）　（mod 2^n）

- 当补码的位数为n+1位时，其模为2^{n+1}，所以

$[-2^{n-1}]_{补} = 2^{n+1} - 2^{n-1} = 2^n + 2^{n-1} = 1\ 10\ldots0$（n-1个0）(mod 2^{n+1})

这说明同一个真值在不同位数的补码表示中，对应的机器数不同，因此给定编码表示时，一定要明确编码的位数，在机器内部编码的位数就是机器中运算部件的位数。

(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 $[x]_补 = 1.0001$

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

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

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　　$x = +1101$　　　　$x = -1101$

$[x]_{反} = 0,1101$

用 逗号 将符号位和数值位隔开

$[x]_{反} = (2^{4+1} - 1) - 1101$

$= 11111 - 1101$

$= 1,0010$

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

x 为真值

如

$x = +0.1101$

$[x]_{反} = 0.1101$

$x = -0.1010$

$[x]_{反} = (2-2^{-4}) - 0.1010$

$= 1.1111 - 0.1010$

$= 1.0101$

用 小数点 将符号位

和数值位隔开

(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（其中一位为符号位）

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