1001011

原码 —— 机器

编码 —— 人

符号位.

$[-0.10110]_{原} = 1.10110$

$[+1011]_{原} = 01011$

$[0]_{原} = 00 = 10$

原码表达的负数无法满足 减法 = + 负数

-0.1011

⇒ 1 . 0100　按位取反

⇒ 1. 0101　最后一位+1 (若为1,则进位)

-0111

⇒11000

⇒11001

(4) 特点

补码最高一位为符号位，0正1负；

补码零有唯一编码；

补码能很好用于加减运算。

补码满足 $[-x]_{补}+[x]_{补}=0$　$[+7]_{补}=00111$　$[-7]_{补}=11001$

最高位参与演算，与其它位一样对待。

扩展方便。5位的补码扩展为8位 00111 ⟶ 00000111

11001 ⟶ 11111001

算术移位。假设 $[x]_{补}= x0.\ x1\ x2 \cdots xn$,

$[x/2]_{补}= x0.\ x0\ x1\ x2 \cdots xn-1$

原符号位不变，符号位与数值位均右移一位，

[X]补 =10010　则　[X/2]补 =11001

27

最大的优点就是将减法运算转换成加法运算。

$[X+Y]_{补}= [X]_{补} + [Y]_{补}$

$[X-Y]_{补}= [X]_{补} + [-Y]_{补}$

例如：$X=(11)_{10}=(1011)_2$，$Y=(5)_{10}=(0101)_2$，已知字长n=5位，则

$[X]_{补}+[-Y]_{补}= 01011 + 11011 = 100110 = 00110 = (6)_{10}$

注：最高1位已经超过字长故应丢掉。系统自动舍弃

$[X-Y]_{补}= [0110]_{补}=00110$

0.11010011

1.00110101

0.00001000

0.00101100

1.11001010

1.11110110

0.

0.10000001

0.00101111

0.10110000

0.01111111

0.11010001

1.01010000

1.01001111

0.11010011

1.11001011

1 0.10011110

0.00110101

0.11010011