本节内容

定点数的加减运算

本节总览

- 定点数的加减运算
  - 原码的加减（了解即可）
  - 补码的加减（重要）
  - 溢出判断

注：计算机中，不会用“反码”进行加减运算，因此不作探讨

原码的加减运算

原码表示的有符号数

14	00001110
-14	+ 10001110
0	10011100

☹

加法器直接对原码进行加法运算，可能出错

$$
\begin{array}{r}
\boxed{0}0001110 \\
+\ \boxed{1}0001110 \\
\hline
\end{array}
$$

$$
\begin{array}{r}
00001110 \\
-\ 00001110 \\
\hline
00000000
\end{array}
$$

☺

用减法器实现

原码的加法运算：

正+正 →绝对值做加法，结果为正
负+负 →绝对值做加法，结果为负

可能会溢出

正+负 →绝对值大的减绝对值小的，符号同绝对值大的数
负+正 →绝对值大的减绝对值小的，符号同绝对值大的数

原码的减法运算，“减数”符号取反，转变为加法：

正-负→正+正
负-正→负+负
正-正→正+负
负-负→负+正

补码的加减运算

设机器字长为8位（含1位符号位），$A = 15$，$B = -24$，求$[A+B]_补$和$[A-B]_补$

	原码	补码
$A = +1111$ →	0,0001111 →	0,0001111
$B = -11000$ →	1,0011000 →	1,1101000

负数补→原：①数值位取反+1；②负数补码中，最右边的1及其右边同原码。最右边的1的左边同反码

$[A+B]_补 = [A]_补 + [B]_补 = 0,0001111 + 1,1101000 = 1,1110111$

原码：1,0001001　真值-9

$[A-B]_补 = [A]_补 + [-B]_补 = 0,0001111 + 0,0011000 = 0,0100111$　真值+39

$[-B]_补$：　$[B]_补$连同符号位一起取反加1

$C = 124$，求$[A+C]_补$和$[B-C]_补$

溢出

$[A+C]_补 = 0,0001111 + 0,1111100 = 1,0001011$　真值-117

$[B-C]_补 = 1,1101000 + 1,0000100 =0,1101100$　真值+108

对于补码来说，无论加法还是减法，**最后都会转变成加法**，由加法器实现运算，符号位也参与运算

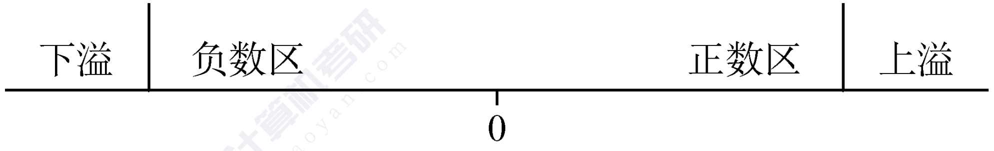
溢出判断

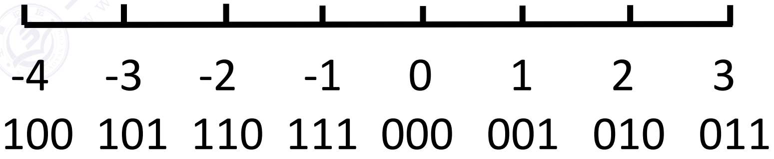
设机器字长为8位（含1位符号位），$A = 15$，$B = -24$，求$[A+B]_{补}$和$[A-B]_{补}$

$C = 124$，求$[A+C]_{补}$和$[B-C]_{补}$

$[A+C]_{补} = \boxed{0},0001111 + \boxed{0},1111100 = \boxed{1},0001011$　　　真值-117

$[B-C]_{补} = \boxed{1},1101000 + \boxed{1},0000100 = \boxed{0},1101100$　　　真值+108





只有“正数+正数 ”才会上溢 —— 正+正=负

只有“负数+负数 ”才会下溢 —— 负+负=正

溢出判断

设机器字长为8位（含1位符号位），$A = 15$，$B = -24$，求$[A+B]_{补}$和$[A-B]_{补}$

$C = 124$，求$[A+C]_{补}$和$[B-C]_{补}$

$[A+C]_{补} = \boxed{0},0001111 + \boxed{0},1111100 = \boxed{1},0001011$　　真值-117

$[B-C]_{补} = \boxed{1},1101000 + \boxed{1},0000100 = \boxed{0},1101100$　　真值+108

逻辑表达式

与：如ABC，表示A与B与C

仅当A、B、C均为1时，ABC为1

A、B、C中有一个或多个为0，则ABC为0

或：如A+B+C，表示A或B或C

仅当A、B、C均为0时，A+B+C为0

A、B、C中有一个或多个为1，则A+B+C为1

非：如 $\overline{A}$，表示A非

若A为1，则 $\overline{A}$ 为0

若A为0，则$\overline{A}$ 为1

方法一：采用一位符号位

设A的符号为A_S，B的符号为B_S，运算结果的符号为S_S，则溢出逻辑表达式为

$$V = \boxed{A_S B_S \overline{S_S}} + \boxed{\overline{A_S}\,\overline{B_S} S_S}$$

若V=0，表示无溢出；

若V=1，表示有溢出。

A_S为1且B_S为1且S_S为0　或　A_S为0且B_S为0且S_S为1

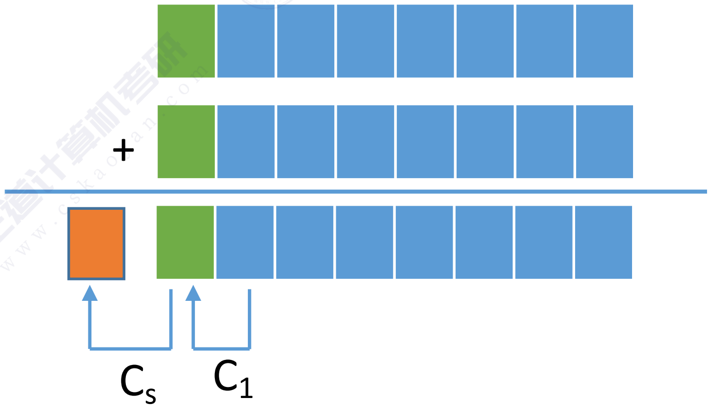
溢出判断

设机器字长为8位（含1位符号位），$A = 15$，$B = -24$，求$[A+B]_{补}$和$[A-B]_{补}$

$C = 124$，求$[A+C]_{补}$和$[B-C]_{补}$

$[A+C]_{补} = 0,0001111 + 0,1111100 = 1,0001011$　　真值-117

$[B-C]_{补} = 1,1101000 + 1,0000100 = 0,1101100$　　真值+108



方法二：采用一位符号位，根据数据位进位情况判断溢出

	符号位的进位C_S	最高数值位的进位C_1
上溢	0	1
下溢	1	0

即：C_S与C_1不同时有溢出

处理“不同”的逻辑符号：异或⊕

溢出逻辑判断表达式为$V=C_S \oplus C_1$

若$V=0$，表示无溢出；$V=1$，表示有溢出。

异或逻辑：不同为1，相同为0

$0 \oplus 0 = 0$

$0 \oplus 1 = 1$

$1 \oplus 0 = 1$

$1 \oplus 1 = 0$

溢出判断

设机器字长为8位（含1位符号位），$A = 15$，$B = -24$，求$[A+B]_补$和$[A-B]_补$

$C = 124$，求$[A+C]_补$和$[B-C]_补$

$[A+C]_补 = 0{,}0001111 + 0{,}1111100 = 1{,}0001011$　　真值-117

$[B-C]_补 = 1{,}1101000 + 1{,}0000100 = 0{,}1101100$　　真值+108

方法三：采用双符号位

正数符号为00，负数符号为11

$[A+C]_补 = 00{,}0001111 + 00{,}1111100 = 01{,}0001011$　　上溢

$[B-C]_补 = 11{,}1101000 + 11{,}0000100 = 10{,}1101100$　　下溢

记两个符号位为$S_{S1}S_{S2}$，则$V=S_{S1}\oplus S_{S2}$

若V=0，表示无溢出；若V=1，表示有溢出。

$[A+B]_补 = 00{,}0001111 + 11{,}1101000 = 11{,}1110111$

$[A-B]_补 = 00{,}0001111 + 00{,}0011000 = 00{,}0100111$

实际存储时只存储1个符号位，运算时会复制一个符号位

双符号位补码又称：模4补码

单符号位补码又称：模2补码

知识点回顾

- 定点数的加减运算
  - 原码
    - 加法
      - 同号相加
        - 数值部分 = 被加数、加数的绝对值进行相加
        - 符号位不变
      - 异号相加
        - 数值部分=被加数、加数中，绝对值更大的减绝对值更小的
        - 符号位与绝对值更大的数相同
    - 减法 —— 将减数取负，转变为加法
  - 补码
    - 符号位参与加法运算；减法要转变成等价的加法
    - 注意：已知 [Y]补码，如何求 [-Y]补码？ —— 将[Y]补码全部位取反，末位+1
  - 溢出判断（补码）
    - Key：正+正=负（上溢）；负+负=正（下溢）
    - 方法一 —— $V=A_S B_S \overline{S_S}+\overline{A_S}\,\overline{B_S} S_S$ 两个加数为正、运算结果为负时发生溢出；或者 两个加数为负、运算结果为正时溢出。
    - 方法二 —— $V=C_S \oplus C_1$ 若符号位的进位Cs与最高数值位的进位C1不同则发生溢出，否则不溢出
    - 方法三 —— $V=S_{S1} \oplus S_{S2}$ 采用双符号位补码表示数值，若运算结果的两个符号位不同则溢出，相同则未发生溢出
    - 注：上述三种逻辑表达式中，V=1时溢出，V=0时未发生溢出

模4补码、模2补码的概念