

2.3.3 补码的非

可以看到范围在 $TMin_w \leq x \leq TMax_w$ 中的每个数字 x 都有 $+_w^1$ 下的加法逆元, 我们将 $-_w^1 x$ 表示如下。

原理: 补码的非


对满足 $TMin_w \leq x \leq TMax_w$ 的 x , 其补码的非 $-_w^1 x$ 由下式给出

$$-_w^1 x = \begin{cases} TMin_w, & x = TMin_w \\ -x, & x > TMin_w \end{cases} \quad (2.15)$$

也就是说, 对 w 位的补码加法来说, $TMin_w$ 是自己的加法的逆, 而对其他任何数值 x 都有一 $-x$ 作为其加法的逆。

推导: 补码的非

观察发现 $TMin_w + TMin_w = -2^{w-1} + (-2^{w-1}) = -2^w$ 。这将导致负溢出, 因此 $TMin_w + _w^1 TMin_w = -2^w + 2^w = 0$ 。对满足 $x > TMin_w$ 的 x , 数值 $-x$ 可以表示为一个 w 位的补码, 它们的和 $-x + x = 0$ 。 ■

 **练习题 2.33** 我们可以用一个十六进制数字来表示长度 $w=4$ 的位模式。根据这些数字的补码的解释, 填写下表, 确定所示数字的加法逆元。

x		$-_w^1 x$	
十六进制	十进制	十进制	十六进制
0			
5			
8			
D			
F			

对于补码和无符号(练习题 2.28)非(negation)产生的位模式, 你观察到什么?

网络旁注 DATA:TNEG 补码非的位级表示

计算一个位级表示的值的补码非有几种聪明的方法。这些技术很有用(例如当你在调试程序的时候遇到值 `0xffffffffa`), 同时它们也能够让你更了解补码表示的本质。

执行位级补码非的第一种方法是对每一位求补, 再对结果加 1。在 C 语言中, 我们可以说, 对于任意整数 x , 计算表达式 $-x$ 和 $\sim x + 1$ 得到的结果完全一样。

下面是一些示例, 字长为 4:

\vec{x}		$\sim\vec{x}$		$incr(\sim\vec{x})$	
[0101]	5	[1010]	-6	[1011]	-5
[0111]	7	[1000]	-8	[1001]	-7
[1100]	-4	[0011]	3	[0100]	4
[0000]	0	[1111]	-1	[0000]	0
[1000]	-8	[0111]	7	[1000]	-8

从前面的例子我们知道 `0xf` 的补是 `0x0`, 而 `0xa` 的补是 `0x5`, 因而 `0xffffffffa` 是 `-6` 的补码表示。

计算一个数 x 的补码非的第二种方法是建立在将位向量分为两部分的基础之上的。假设 k 是最右边的 1 的位置, 因而 x 的位级表示形如 $[x_{w-1}, x_{w-2}, \dots, x_{k+1}, 1, 0, \dots, 0]$ 。(只要 $x \neq 0$ 就能够找到这样的 k 。)这个值的非写成二进制格式就是 $[\sim x_{w-1}, \sim x_{w-2}, \dots,$