

请注意补码(Two's complement)和反码(Ones' complement)中撇号的位置是不同的。术语补码来源于这样一个情况，对于非负数  $x$ ，我们用  $2^w - x$  (这里只有一个 2) 来计算  $-x$  的  $w$  位表示。术语反码来源于这样一个属性，我们用  $[111\cdots 1] - x$  (这里有很多个 1) 来计算  $-x$  的反码表示。


为了更好地理解补码表示，考虑下面的代码：

```
1      short x = 12345;
2      short mx = -x;
3
4      show_bytes((byte_pointer) &x, sizeof(short));
5      show_bytes((byte_pointer) &mx, sizeof(short));
```

当在大端法机器上运行时，这段代码的输出为 30 39 和 cf c7，指明  $x$  的十六进制表示为 0x3039，而  $mx$  的十六进制表示为 0xcfc7。将它们展开为二进制，我们得到  $x$  的位模式为  $[0011000000111001]$ ，而  $mx$  的位模式为  $[1100111111000111]$ 。如图 2-15 所示，等式(2.3)对这两个位模式生成的值为 12 345 和 -12 345。

权	12 345		-12 345		53 191	
	位	值	位	值	位	值
1	1	1	1	1	1	1
2	0	0	1	2	1	2
4	0	0	1	4	1	4
8	1	8	0	0	0	0
16	1	16	0	0	0	0
32	1	32	0	0	0	0
64	0	0	1	64	1	64
128	0	0	1	128	1	128
256	0	0	1	256	1	256
512	0	0	1	512	1	512
1 024	0	0	1	1 024	1	1 024
2 048	0	0	1	2 048	1	2 048
4 096	1	4096	0	0	0	0
8 192	1	8192	0	0	0	0
16 384	0	0	1	16 384	1	16 384
±32 768	0	0	1	-32 768	1	32 768
总计	12 345		-12 345		53 191	

图 2-15 12 345 和 -12 345 的补码表示，以及 53 191 的无符号表示。注意后面两个数有相同的位表示

 **练习题 2.18** 在第 3 章中，我们将看到由反汇编器生成的列表，反汇编器是一种将可执行程序文件转换回可读性更好的 ASCII 码形式的程序。这些文件包含许多十六进制数字，都是用典型的补码形式来表示这些值。能够认识这些数字并理解它们的意义（例如它们是正数还是负数），是一项重要的技巧。

在下面的列表中，对于标号为 A~I(标记在右边)的那些行，将指令名(sub、mov 和 add)右边显示的(32 位补码形式表示的)十六进制值转换为等价的十进制值。

```
4004d0: 48 81 ec e0 02 00 00    sub    $0x2e0,%rsp      A.
4004d7: 48 8b 44 24 a8          mov    -0x58(%rsp),%rax  B.
4004dc: 48 03 47 28            add    0x28(%rdi),%rax   C.
4004e0: 48 89 44 24 d0          mov    %rax,-0x30(%rsp)  D.
```