

6.1 原码、反码、补码的存储_物联网 / 嵌入式工程师 - 慕课网

“ 慕课网慕课教程 6.1 原码、反码、补码的存储涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

1. 原码、反码、补码的存储

注： 下面的讲解只涉及到基本规则使用，没有设计到具体数据类型。待学习后面数据类型后，会进行我们刚刚给上面计算的数都是我们的正数。对我们人常常来说使用的十进制，而一个十进制数是有正负之分的，那么如何在计算机中如何表示我们的正号和负号呢？这个就涉及到了，我们计算机的编码形式，在计算机中，用我们的原码、反码、补码来表示数据。在内存中的数据使用补码来存储。

- A. 数据就在内存中总是以二进制补码的形式存放。最高位为符号位。正数为 0，负数为 1（不完全）
- B. 正数的原码、反码、补码就是它本身。原码该数转换位二进制的形式
- C. 负数的反码： 符号位不变，对原码其它位依次取反。0 变 1，1 变 0

负数的补码： 符号位不变，反码 + 1

例如：

以 8bit 的数据存储为例讲解。

+10	原码：	0000 1010	8+2
	反码：	0000 1010	
	补码：	0000 1010	
			在内存中以补码存储
-10	原码：	1000 1010	
	反码：	1111 0101	
	补码：	1111 0110	
			在内存中以补码存储

•

补码转换为原码

- 方法 1: 原码 = 补码 - 1，然后符号位不变，按位取反

例如：

-10	补码：	1111 0110
-1	:	1111 0101
	取反：	1000 1010

例如：

-10	补码：	1111 0110
	取反：	1000 1001
+1	:	1000 1010

理解方法：

不考虑符号位，原码 + 反码 = 0xff

例如：

```
1111 0110
1000 1001
-----
1111 1111====>0xff
```

即原码 + 原码的反码 =0xff, 而补码 = 原码的反码 + 1 , 得出 补码 = (0xff – 原码) + 1

根据以上等式可用推导出

* 原码的反码 <====>0xff – 原码 *

补码的反码 <====>0xff – 补码

同理，改变公式。

原码 = (0xff – 补码) + 1

* 可得 : 0xff – 补码 <====> 补码的反码 *

原码 = 补码的反码 + 1 ==> 对补码再求补码可得原码。

假设数据按 8bit 存储，自己写出 – 30 的原码、反码、补码形式。

有一个负数，其补码为 1011 0011，计算它的原码

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎 beta，点击查看详细说明

