就像前面所看到的，这个 16 位的位模式就是－12 345 的补码表示。当我们把它强制类型转换回 int 时，符号扩展把高 16 位设置为 1，从而生成－12 345 的 32 位补码表示。

当将一个 w 位的数 $\vec{x}=[x_{w-1}, x_{w-2}, \cdots, x_0]$ 截断为一个 k 位数字时，我们会丢弃高 $w-k$ 位，得到一个位向量 $\vec{x}'=[x_{k-1}, x_{k-2}, \cdots, x_0]$。截断一个数字可能会改变它的值——溢出的一种形式。对于一个无符号数，我们可以很容易得出其数值结果。

原理：截断无符号数

令 $\vec{x}$ 等于位向量 $[x_{w-1}, x_{w-2}, \cdots, x_0]$，而 $\vec{x}'$ 是将其截断为 k 位的结果：$\vec{x}'=[x_{k-1}, x_{k-2}, \cdots, x_0]$。令 $x=B2U_w(\vec{x})$，$x'=B2U_k(\vec{x}')$。则 $x'=x \bmod 2^k$。

该原理背后的直觉就是所有被截去的位其权重形式都为 2^i，其中 $i \geqslant k$，因此，每一个权在取模操作下结果都为零。可用如下推导表示：

推导：截断无符号数

通过对等式(2.1)应用取模运算就可以看到：

$$\begin{aligned} B2U_w([x_{w-1}, x_{w-2}, \cdots, x_0]) \bmod 2^k &= \left[\sum_{i=0}^{w-1} x_i 2^i\right] \bmod 2^k \\ &= \left[\sum_{i=0}^{k-1} x_i 2^i\right] \bmod 2^k \\ &= \sum_{i=0}^{k-1} x_i 2^i \\ &= B2U_k([x_{k-1}, x_{k-2}, \cdots, x_0]) \end{aligned}$$

在这段推导中，我们利用了属性：对于任何 $i \geqslant k$，$2^i \bmod 2^k = 0$。 ■

补码截断也具有相似的属性，只不过要将最高位转换为符号位：

原理：截断补码数值

令 $\vec{x}$ 等于位向量 $[x_{w-1}, x_{w-2}, \cdots, x_0]$，而 $\vec{x}'$ 是将其截断为 k 位的结果：$\vec{x}'=[x_{k-1}, x_{k-2}, \cdots, x_0]$。令 $x=B2U_w(\vec{x})$，$x'=B2T_k(\vec{x}')$。则 $x'=U2T_k(x \bmod 2^k)$。

在这个公式中，$x \bmod 2^k$ 将是 0 到 2^k-1 之间的一个数。对其应用函数 $U2T_k$ 产生的效果是把最高有效位 x_{k-1} 的权重从 2^{k-1} 转变为 -2^{k-1}。举例来看，将数值 $x=53\,191$ 从 int 转换为 short。由于 $2^{16}=65\,536 \geqslant x$，我们有 $x \bmod 2^{16}=x$。但是，当我们把这个数转换为 16 位的补码时，我们得到 $x'=53\,191-65\,536=-12\,345$。

推导：截断补码数值

使用与无符号数截断相同的参数，则有

$$B2U_w([x_{w-1}, x_{w-2}, \cdots, x_0]) \bmod 2^k = B2U_k[x_{k-1}, x_{k-2}, \cdots, x_0]$$

也就是，$x \bmod 2^k$ 能够被一个位级表示为 $[x_{k-1}, x_{k-2}, \cdots, x_0]$ 的无符号数表示。将其转换为补码数则有 $x'=U2T_k(x \bmod 2^k)$。 ■

总而言之，无符号数的截断结果是：

$$B2U_k[x_{k-1}, x_{k-2}, \cdots, x_0] = B2U_w([x_{w-1}, x_{w-2}, \cdots, x_0]) \bmod 2^k \tag{2.9}$$

而补码数字的截断结果是：

$$B2T_k[x_{k-1}, x_{k-2}, \cdots, x_0] = U2T_k(B2U_w([x_{w-1}, x_{w-2}, \cdots, x_0]) \bmod 2^k) \tag{2.10}$$

练习题 2.24　假设将一个 4 位数值(用十六进制数字 0～F 表示)截断到一个 3 位数值(用十六进制数字 0～7 表示)。填写下表，根据那些位模式的无符号和补码解释，说明这种截断对某些情况的结果。