了 $\vec{x}$ 的无符号表示。在这个编码中，每个位 x_i 都取值为 0 或 1，后一种取值意味着数值 2^i 应为数字值的一部分。我们用一个函数 $B2U_w$(Binary to Unsigned 的缩写，长度为 w)来表示：

原理：无符号数编码的定义

对向量 $\vec{x}=[x_{w-1}, x_{w-2}, \cdots, x_0]$：

$$B2U_w(\vec{x}) \doteq \sum_{i=0}^{w-1} x_i 2^i \tag{2.1}$$

在这个等式中，符号“$\doteq$”表示左边被定义为等于右边。函数 $B2U_w$ 将一个长度为 w 的 0、1 串映射到非负整数。举一个示例，图 2-11 展示的是下面几种情况下 $B2U$ 给出的从位向量到整数的映射：

$$\begin{aligned}
B2U_4([0001]) &= 0\cdot 2^3 + 0\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 = 0+0+0+1 = 1\\
B2U_4([0101]) &= 0\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 = 0+4+0+1 = 5\\
B2U_4([1011]) &= 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 8+0+2+1 = 11\\
B2U_4([1111]) &= 1\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 8+4+2+1 = 15
\end{aligned} \tag{2.2}$$

在图中，我们用长度为 2^i 的指向右侧箭头的条表示每个位的位置 i。每个位向量对应的数值就等于所有值为 1 的位对应的条的长度之和。

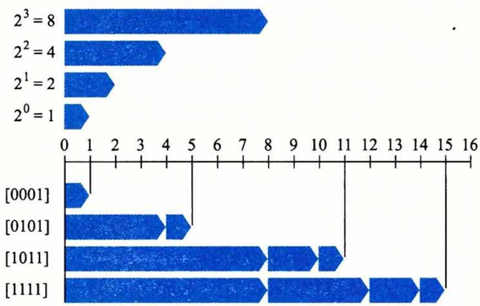

图 2-12　$w=4$ 的无符号数示例。当二进制表示中位 i 为 1，数值就会相应加上 2^i

让我们来考虑一下 w 位所能表示的值的范围。最小值是用位向量$[00\cdots0]$表示，也就是整数值 0，而最大值是用位向量$[11\cdots1]$表示，也就是整数值 $UMax_w \doteq \sum_{i=0}^{w-1} 2^i = 2^w-1$。以 4 位情况为例，$UMax_4 = B2U_4([1111]) = 2^4-1=15$。因此，函数 $B2U_w$ 能够被定义为一个映射 $B2U_w$：$\{0,1\}^w \rightarrow \{0, \cdots, 2^w-1\}$。

无符号数的二进制表示有一个很重要的属性，也就是每个介于 $0\sim2^w-1$ 之间的数都有唯一一个 w 位的值编码。例如，十进制值 11 作为无符号数，只有一个 4 位的表示，即$[1011]$。我们用数学原理来重点讲述它，先表述原理再解释。

原理：无符号数编码的唯一性

函数 $B2U_w$ 是一个双射。

数学术语双射是指一个函数 f 有两面：它将数值 x 映射为数值 y，即 $y=f(x)$，但它也可以反向操作，因为对每一个 y 而言，都有唯一一个数值 x 使得 $f(x)=y$。这可以用反函数 f^{-1}来表示，在本例中，即 $x=f^{-1}(y)$。函数 $B2U_w$ 将每一个长度为 w 的位向量都映射为 $0\sim2^w-1$ 之间的一个唯一值；反过来，我们称其为 $U2B_w$(即“无符号数到二进制”)，在 $0\sim2^w-1$ 之间的每一个整数都可以映射为一个唯一的长度为 w 的位模式。

2.2.3　补码编码

对于许多应用，我们还希望表示负数值。最常见的有符号数的计算机表示方式就是补码(two's-complement)形式。在这个定义中，将字的最高有效位解释为负权(negative weight)。我们用函数 $B2T_w$(Binary to Two's-complement 的缩写，长度为 w)来表示：