和~扩展到位向量上。

举个例子，假设 $w=4$，参数 $a=[0110]$，$b=[1100]$。那么 4 种运算 $a\&b$、$a|b$、$a\^{}b$ 和~b 分别得到以下结果：

$$\begin{array}{r} 0110 \\ \& \ \underline{1100} \\ 0100 \end{array} \quad \begin{array}{r} 0110 \\ | \ \underline{1100} \\ 1110 \end{array} \quad \begin{array}{r} 0110 \\ \hat{} \ \underline{1100} \\ 1010 \end{array} \quad \begin{array}{r} \\ \sim \ \underline{1100} \\ 0011 \end{array}$$

练习题 2.8 填写下表，给出位向量的布尔运算的求值结果。

运算	结果
a	[01101001]
b	[01010101]
$\sim a$	______
$\sim b$	______
$a\&b$	______
$a\mid b$	______
$a\^{}b$	______

网络旁注 DATA:BOOL　关于布尔代数和布尔环的更多内容

对于任意整数 $w>0$，长度为 w 的位向量上的布尔运算 |、& 和~形成了一个布尔代数。最简单的情况是 $w=1$ 时，只有 2 个元素；但是对于更普遍的情况，有 2^w 个长度为 w 的位向量。布尔代数和整数算术运算有很多相似之处。例如，乘法对加法的分配律，写为 $a\cdot(b+c)=(a\cdot b)+(a\cdot c)$，而布尔运算 & 对 | 的分配律，写为 $a\&(b|c)=(a\&b)|(a\&c)$。此外，布尔运算 | 对 & 也有分配律，写为 $a|(b\&c)=(a|b)\&(a|c)$，但是对于整数我们不能说 $a+(b\cdot c)=(a+b)\cdot(a+c)$。

当考虑长度为 w 的位向量上的^、& 和~运算时，会得到一种不同的数学形式，我们称为布尔环(Boolean ring)。布尔环与整数运算有很多相同的属性。例如，整数运算的一个属性是每个值 x 都有一个加法逆元(additive inverse)$-x$，使得 $x+(-x)=0$。布尔环也有类似的属性，这里的“加法”运算是^，不过这时每个元素的加法逆元是它自己本身。也就是说，对于任何值 a 来说，$a\^{}a=0$，这里我们用 0 来表示全 0 的位向量。可以看到对单个位来说这是成立的，即 $0\^{}0=1\^{}1=0$，将这个扩展到位向量也是成立的。当我们重新排列组合顺序，这个属性也仍然成立，因此有 $(a\^{}b)\^{}a=b$。这个属性会引起一些很有趣的结果和聪明的技巧，在练习题 2.10 中我们会有所探讨。

位向量一个很有用的应用就是表示有限集合。我们可以用位向量 $[a_{w-1}, \cdots, a_1, a_0]$ 编码任何子集 $A\subseteq\{0, 1, \cdots, w-1\}$，其中 $a_i=1$ 当且仅当 $i\in A$。例如(记住我们是把 a_{w-1} 写在左边，而将 a_0 写在右边)，位向量 $a\doteq[01101001]$ 表示集合 $A=\{0, 3, 5, 6\}$，而 $b\doteq[01010101]$ 表示集合 $B=\{0, 2, 4, 6\}$。使用这种编码集合的方法，布尔运算 | 和 & 分别对应于集合的并和交，而~对应于于集合的补。还是用前面那个例子，运算 $a\&b$ 得到位向量[01000001]，而 $A\cap B=\{0, 6\}$。

在大量实际应用中，我们都能看到用位向量来对集合编码。例如，在第 8 章，我们会看到有很多不同的信号会中断程序执行。我们能够通过指定一个位向量掩码，有选择地使能或是屏蔽一些信号，其中某一位位置上为 1 时，表明信号 i 是有效的(使能)，而 0 表明该信号是被屏蔽的。因而，这个掩码表示的就是设置为有效信号的集合。