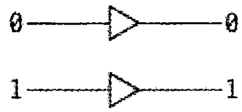
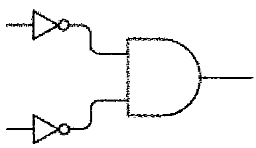


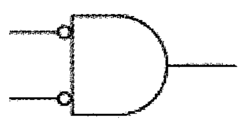
但是在输入信号很微弱的时候，缓冲器就可以派上用场。之前提到过，这也就是很多年前在电报机中使用继电器的原因。另外，缓冲器还可以用于延迟一个信号。这是因为继电器需要一点时间——几分之一秒——才会被触发。

本书从这里开始，继电器将会极少出现了。取而代之地，以后的电路会由缓冲器、反向器、四种基本逻辑门和其他由逻辑门组成的复杂电路（如 2-4 译码器）组成。当然，所有这些器件也是由继电器构成的，但我们用不着直接使用它了。

前面在建立 2-4 译码器的时候，曾出现过这样一个小电路。

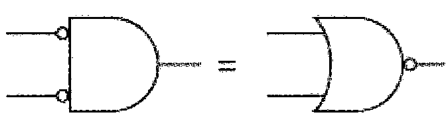


两个输入信号经过反向器后作为与门的输入。这样的组合有时可以去掉反向器而画成如下的形式。



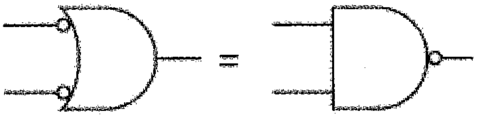
注意与门输入端的小圆圈。这些小圆圈表示信号在那一点被反转——将 0（低电平）转换为 1（高电平），反之亦然。

实际上，带有两个反向输入的与门和或非门是等价的。



只有当输入都为 0 的时候，其输出才为 1。

类似的，带有两个反向输入的或门和与非门也是等价的。



只有当输入全为 1 时，输出才为 0。

这两组等价关系就是摩根定律在电路中的实现。摩根是维多利亚女王时代的另一位数学家，比布尔大 9 岁，于 1847 年发表《形式逻辑》（*Formal Logic*）一书，与布尔的《逻辑的数学分析》（*The Mathematical Analysis of Logic*）恰好同一天出版（据说是同一天）。实际上，布尔研究逻辑的灵感正源于一场公开的争论，这个争论就发生在摩根和另一个被指剽窃的英国数学家之间（历史最后证实摩根是清白的）。摩根很早就发现了布尔的洞察力。他无私地鼓励布尔，并帮助他开展研究，而今天除了他的这个著名定律之外，他几乎已被人们所遗忘。

摩根定律可以简单地表示为如下形式：

$$\overline{A} \times \overline{B} = \overline{A + B}$$
$$\overline{A} + \overline{B} = \overline{A \times B}$$

A 和 B 是两个布尔操作数。在第一个表达式中，两个操作数先被取反，再进行与运算。这与两个操作数进行或运算后再取反（即或非）的结果是一样的。在第二个表达式中，两个操作数先取反，再进行或运算，这和两个操作数先做与运算后再取反（即与非）的结果是等价的。

摩根定律是简化布尔表达式的一种重要手段，因此也可以用来简化电路。从历史的角度来说，这正是香农的论文带给电气工程师们的真正意义。但在本书中简化电路不是重点，我们关注的是让一切有效地运转而不是以最简的形式运转。我们下面要做的就是一台加法器。

12 二进制加法器

加法是算术运算中最基本的运算，因此如果想搭建一台计算机（这也正是本书所隐含的内容），那么首先就要造出可以计算两个数的和的器件。当你真正面对它时，就会发现，原来加法计算就是计算机要做的唯一工作。如果我们可以造出加法器，同样地，就可以利用加法来实现减法、乘法和除法，计算按揭付款，引导火箭飞到火星、下棋，以及填写我们的话费账单。

这一章中我们要创建的加法器与现代的计算器和计算机来比，将会很庞大、很笨拙、很慢，而且运转起来噪声不断。最有趣的是我们用来制作加法器的全部零件，都是像开关、灯泡、导线、电池、逻辑门、已经预先连接在各种逻辑门中的继电器等这些在之前的章节中学过的非常简单的电子器件。这个加法器所包含的所有器件早在 120 多年前就已经被发明出来了。我们并不用在房间中实际搭建出什么，相反地，可以在纸上以及我们的头脑中来搭建这个加法器。

这个加法器完全用于二进制数的计算，而且没有现代加法器那么便利。你不能使用键盘来给出需要相加的两个数，相反所要用到的是一排开关。计算结果是通过一排灯泡来显示的，而非以数字的形式来显示。

但是，这个加法器确实可以将两个数相加，其方法与计算机计算加法的方式非常相似。

计算二进制数加法与计算十进制数加法非常相似。如果你想要让 245 和 673 这两个十进制数相加，你会把这个问题分解为几个简单的步骤。每个步骤只需要将两个十进制数相加。在这个例子中，首先要将 5 和 3 相加。生活中，记住加法表能更快地解决问题。

二进制数加法与十进制数加法最大的不同就在于二进制数加法中用到了一个更为简单的加法表。

+	0	1
0	0	1
1	1	10

如果你是一头鲸鱼并且在学校中学习了这个表格，你会大声说出：

0 加 0 等于 0；
0 加 1 等于 1；
1 加 0 等于 1；
1 加 1 等于 0，进位为 1。

以上加法表可以重新写为下面带有前导零的形式，这样每个结果就都是一个 2 位的值。

+	0	1
0	00	01
1	01	10

像这样，一对二进制数相加的结果中具有两个数位，其中一位叫做加法位（sum bit），另一位则叫做进位位（carry bit，例如，1 加 1 等于 0，进位为 1）。下面我们将二进制数加法表分成两个表格，第一个是表示加法的。

+加法	0	1
0	0	1
1	1	0

第二个是表示进位的。

+进位	0	1
0	0	0
1	0	1

用这种方法来看二进制加法非常方便，因为在我们的加法器中加法与进位是分别进行的。搭建一个二进制加法器需要我们首先设计一个电路，通过该电路执行这些操作。完全用二进制数，问题将在很大程度上得以简化，因为，电路中的所有器件，像开关、灯泡、导线等都可以用来表示二进制数的位。

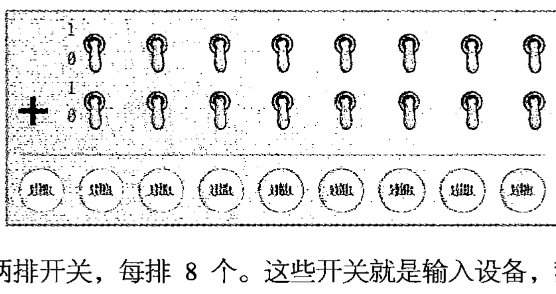
与十进制数加法一样，我们将两个二进制数字串由右向左依次逐列相加。

$$\begin{array}{r} 01100101 \\ +\ 10110110 \\ \hline 100011011 \end{array}$$

注意，在我们将从右边数的第 3 列的数相加的时候，会有一个 1 进位到下一列中。这在第 6、第 7、第 8 列也是一样的。

想要让多大的二进制数相加？由于仅仅在头脑中来搭建加法器，所我们可以搭建出一个能够让非常长的二进制数相加的加法器。但是经过理智的考虑之后，我们决定让搭建起来的二进制加法器最高能够执行的加法长度为 8 位。也就是说，我们想要相加的二进制数，其范围是从 0000-0000 到 1111-1111，即十进制数的 0 到 255。两个 8 位二进制的和最大可为 1-1111-1110，即 510。

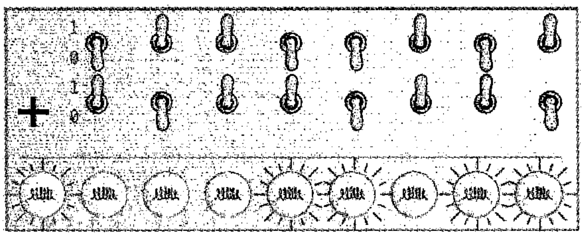
加法器的控制面板如下图所示。



在这个面板中，有两排开关，每排 8 个。这些开关就是输入设备，我们将用它们来“输入”两个 8 位二进制数。在这套输入设备中，开关断开（关）即表示 0，闭合（开）表示 1，这与你房间墙上的开关是一样的。面板底部的输出设备是一排灯泡，共 9 个。这些灯泡用来显示结果。不发光的灯泡表示 0，发光的灯泡表示 1。这里有 9 个灯泡，因为两个

8 位二进制数的相加结果可能是一个 9 位的二进制数。

加法器的其他部分是以各种形式连接起来的逻辑门。开关将触发逻辑门中的继电器来点亮相应的灯泡。例如，如果我们要将 0110-0101 和 1011-0110（之前例子中的两个数）相加，则要将相应的开关置于下图所示位置。



灯泡发光显示结果为：1-0001-1011（然而，这只是一个希望的结果，因为我们还没有将加法器真正搭建出来！）

在上一章中提到过，本书中我们将会用到许多继电器。这个 8 位二进制加法器中所用到的继电器不少于 144 个，其中我们用来相加的 8 对二进制位，每对都需要 18 个继电器。如果将全部电路展示出来，你一定会崩溃的。没有人能看懂以各种方式连接起来的 144 个继电器所表达的意义。相反地，我们要利用逻辑门来分阶段地处理这个问题。当看到进位（两个 1 相加就会产生一个进位）结果表的时候，或许你已经看出来逻辑门和二进制加法的一些相关性了。

+进位	0	1
0	0	0
1	0	1

你可能意识到了，这和上一章中与门的输出结果是一样的。

AND	0	1
0	0	0
1	0	1

因此，利用与门可以计算两个二进制数加法的进位。

到此，我们着实取得了一些进展。下面我们要做的就是利用继电器来实现下表。

+加法	0	1
0	0	1
1	1	0

这是在作两个二进制数加法时需要解决的另一个问题。加法位的情况并不像进位位那样简单，但是我们即将实现它。

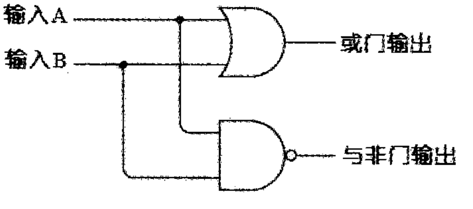
首先我们要知道，或门和我们想要的结果很相似，除了右下角的结果。

OR	0	1
0	0	1
1	1	1

与非门同样和我们想要的结果很相似，除了左上角的结果。

NAND	0	1
0	1	1
1	1	0

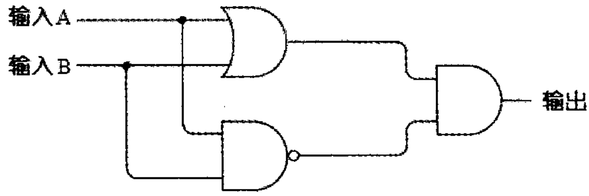
下面我们将或门和与非门连接到相同的输入上，如下图所示。



下表总结了或门和与非门的输出，并将其与我们想要的结果进行了对比。

输入 A	输入 B	或门输出	与非门输出	想要的结果
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

注意，我们想要的是 1，那么这种情况只有在或门和与非门的输出都为 1 时才会出现。这表明两个输出端可以通过一个与门连接到一起。

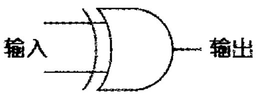


这就是我们想要的结果。

注意，在整个电路中仍然有两个输入和一个输出。两个输入同时作为或门和与非门的输入。或门和与非门的输出又分别作为一个与门的输入，最后得出了我们想要的结果。

输入 A	输入 B	或门输出	与非门输出	与门输出
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

实际上这个电路有个专门的名称，叫做异或门，简写为 XOR。之所以称为异或门是因为若想其输出结果为 1，要么仅让输入 A 为 1，要么仅让输入 B 为 1，两输入端都为 1 则输出为 0。为了不把或门、与非门和与门都画出来，我们可以使用一个电气工程师所采用的特定电气符号来表示异或门。



异或门在输入端比或门多出了一条曲线，除此之外它看上去和或门非常相像。异或门的特征如下表所示。

XOR	0	1
0	0	1
1	1	0

异或门是本书中详细介绍的最后一个逻辑门（第 6 个门有时会在电气工程中介绍到。它称做同或门，因为只有当两个输入相同的时候，其输出才为 1。同或门所给出的输出刚好与异或门相反，因此同或门的符号和异或门相同，但在输出端多了个小圆圈）。

让我们回顾一下到目前为止所了解的内容。将两个二进制数相加将产生一个加法位和一个进位位。

+加和	0	1
0	0	1
1	1	0

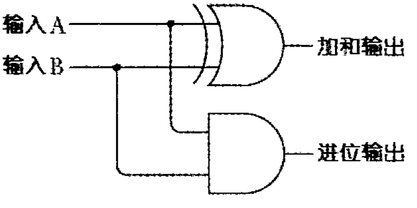
+进位	0	1
0	0	0
1	0	1

可以利用下面这两个逻辑门来实现这些结果。

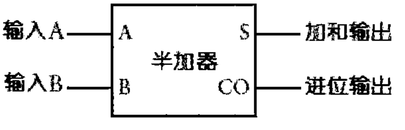
XOR	0	1
0	0	1
1	1	0

AND	0	1
0	0	0
1	0	1

两个二进制数相加的结果是由异或门的输出给出的，而进位位是由与门的输出给出的。因此我们可以将与门和异或门连在一起来计算两个二进制数（即 A 和 B）的和。



为了避免重复画与门和异或门，你可以采用如下这种简单的表示方式。

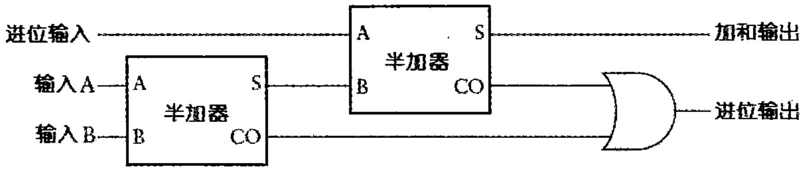


这个符号被称为半加器（Half Adder）。之所以叫半加器是有原因的，它将两个二进制数相加，得出一个加法位和一个进位位。但是绝大多数二进制数是多于 1 位的。半加器没有做到的是将之前一次的加法可能产生的进位位纳入下一次运算。例如，假设我们要将如下两个二进制数相加。

$$\begin{array}{r} 1111 \\ +\ 1111 \\ \hline 11110 \end{array}$$

我们只能将半加器用于最右面一列的相加：1 加 1 等于 0，进位 1。对于从右面算起的第二列，由于进位位的存在，实际上需要将三个二进制数相加，而随后每一列的加法都是这样的。随后的每一列二进制数相加都需要将进位位算进来。

为了对三个二进制数进行加法运算，我们需要将两个半加器和一个或门做如下连接。



要理解它的工作原理，首先从最左边第一个半加器的输入 A 和输入 B 开始，其输出是一个加和及相应的进位。这个和必须与前一列的进位输入相加，然后再把它们输入到第二个半加器中。第二个半加器的输出和是最后的结果。两个半加器的进位输出又被输入到一个或门中。你可能会觉得，这里还需要一个半加法器，这当然是可行的。但是如果你了解了所有的可能性之后，你会发现，两个半加法器的进位输出是不会同时为 1 的。或门在这里已经足够，因为或门除了在输入都为 1 的时候以外，其他情况下结果和异或门结果相同。

为了避免重复地画上面的那个图，我们用以下形式来替代上图中的一堆符号，它称为全加器（Full Adder）。

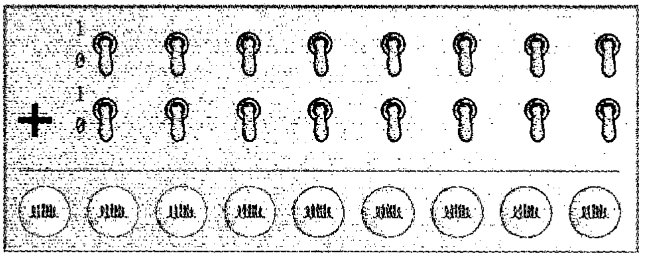
以下表格总结了全加法器所有可能的输入组合以及对应的输出结果。

输入 A	输入 B	进位输入	加和输出	进位输出
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

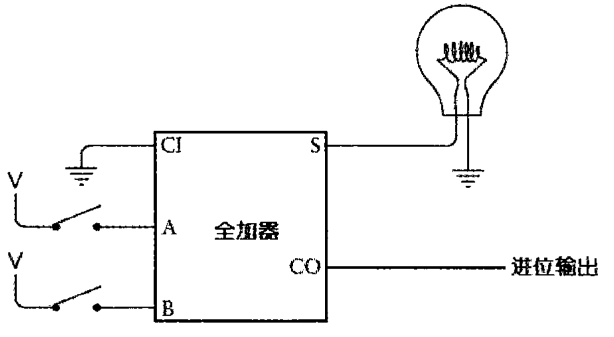
在本章前面曾经提到过，我们的加法器需要 144 个继电器。下面就来解释一下这个数目

是如何得到的。每个与门、或门和与非门都需要两个继电器，因此一个异或门中就包含 6 个继电器。一个半加器是由一个异或门和一个与门组成的，因此一个半加器就需要 8 个继电器。每个全加器由两个半加器和一个或门组成，所以它要 18 个继电器。我们需要 8 个全加器来制作 8 位二进制加法器。因而总共需要 144 个继电器。

再来看看之前提到的由灯泡和开关所组成的控制面板。现在我们可以开始将开关和灯泡连接到全加器了。

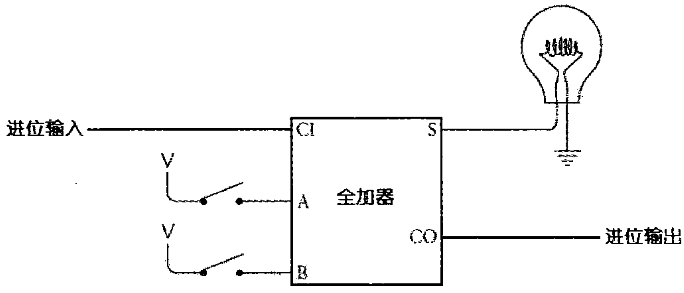


首先将最右端的两个开关和最右端的一个灯泡连接到一个全加器上。



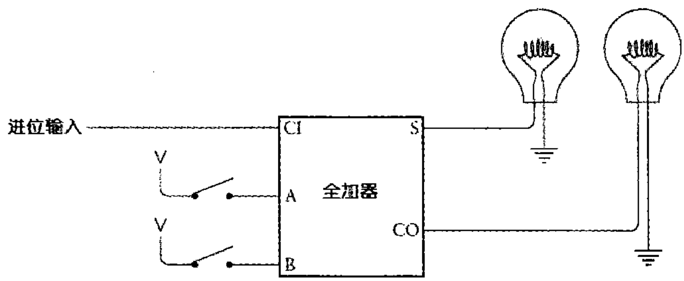
当把两个二进制数相加时，第 1 列的处理方式与其他列有所不同。因为后面的几列可能包括来自前面加法的进位，而第 1 列不会，所以全加器的进位输入端是接地的，这表示第 1 列的进位输入是一个 0。第 1 列二进制数相加后很可能会产生一个进位输出，这个进位输出是下一列加法的输入。

对于接下来的两个二进制位和灯泡，可以按如下办法来连接全加器。



第一个全加器的进位输出就是第二个全加器的进位输入。随后的每列二进制数都以同样的方式连接。每一列进位输出都是下一列的进位输入。

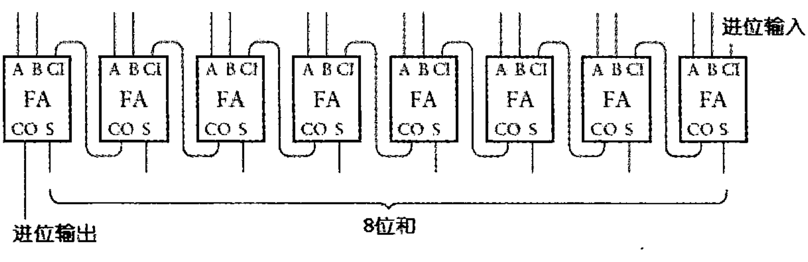
最终，第 8 个灯泡和最后一对开关将以如下方式连接到全加法器上。



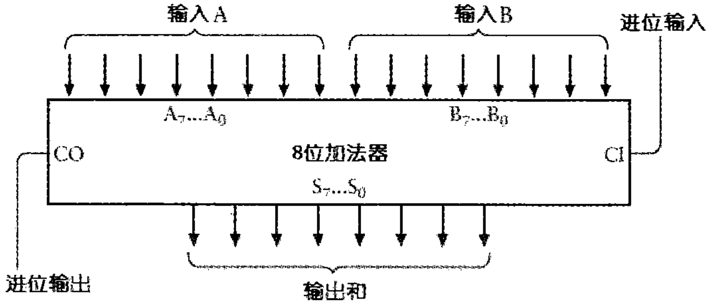
这里，最后一个进位输出将被连接到第 9 个灯泡上。

至此，我们就大功告成了。

还可以用另一种方式来看这 8 个全加器的连接，每个全加器的进位输出都作为下一个全加器的进位输入。



下面是画成一个盒子的完整的 8 位二进制加法器，输入标记为 $A_0 \sim A_7$ 和 $B_0 \sim B_7$，输出标记为 $S_0 \sim S_7$。



这就是表示多位数字中各位数字的常用方法。A_0、B_0和 S_0是最低有效位，或者说是最右边的一位。A_7、B_7和 S_7是最高有效位，或者说是最左边的一位。例如，下面演示了这样一列带有下标的字母是如何用来表示一个二进制数 0110-1001 的。

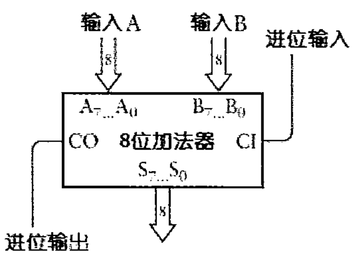
A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	1	1	0	1	0	0	1

下标从 0 开始，并且向着高有效位的方向递增，因为它们和 2 的乘方数（幂）相对应。

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	1	0	1	0	0	1

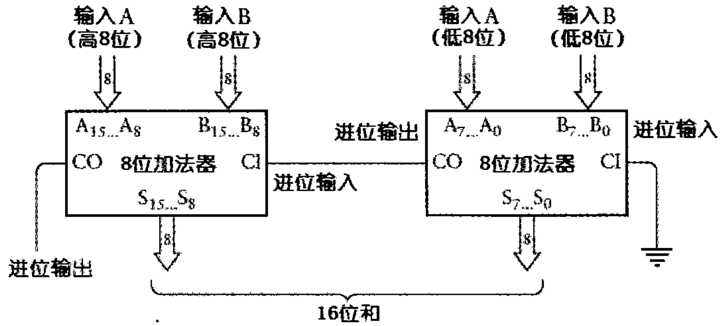
如果把下面一排的每个二进制位和其对应的 2 的幂相乘再依次相加，你就会得到 0110-1001 的十进制数表示 64 + 32 + 8 + 1，即 105。

另一种 8 位二进制加法器可用下图表示。



双线箭头包含了 8 个输入端，代表一组 8 个独立的信号。它们被标识为 $A_7...A_0$、$B_7...B_0$、$S_7...S_0$，同样也表示一个 8 位二进制数。

一旦你搭建起了 8 位二进制加法器，你就可以再搭建另外一个加法器。把它们级联起来就可以很容易地扩展出一个 16 位加法器。



右边加法器的进位输出被连接到了左边加法器的进位输入上。左边加法器的输入包含了两个加数的高 8 位，而得到的结果也是最终加和的高 8 位。

你可能会问："这真的就是计算机进行加法运算时所采用的方式么？"

基本上来说，是的。但也并不完全是。

首先，可以制作一个比这个算得更快的加法器。如果你看一下这个电路是如何工作的，最低有效位的一对数字相加所得出的一个进位输出，将要参与接下来的一对数字的加法运算，由此得到的一个进位输出又要参与再下一对数字的加法运算，依此类推。加法器的总体速度等于数字的位数乘以全加器器件的速度，这被称做行波进位（ripple carry，或脉冲进位）。更快的加法器运用了一个被称为"前置进位"的电路来提高运算的速度。

其次，也是最重要的，计算机已经不再使用继电器了！尽管它曾经被使用过。第一台数字计算机在 20 世纪 30 年代被建造完成，当时所使用的就是继电器，后来也使用过真空管。今天的计算机使用的是晶体管。在被用到计算机中时，晶体管的工作方式与继电器基本相同，但是正如我们即将了解到的，晶体管要比继电器计算速度更快、体积更小，而且噪声更弱、耗能也更低，而且更便宜。搭建一个 8 位加法器依然需要 144 个晶体管（如果你用前置进位法代替行波进位，将会用到更多的晶体管），但是电路却是极小的。

13

如何实现减法

当你确信继电器连接到一起真的可以实现二进制数加法的时候，你可能会问：“那么如何实现减法呢？”本章后续的内容会帮你解答这个问题，因此提出这样的问题并不是说你在没事找事，而实际上这表明你是相当有察觉力的。加法和减法在某些方面相互补充，但在机制方面这两个运算则是不同的。加法是始终从两个加数的最右列向最左列进行计算的。每一列的进位加到下一列中。在减法中没有进位，而是有借位——一种与加法存在本质区别的麻烦机制。

例如，我们来看一个典型的借位减法的题目：

$$\begin{array}{r} 253 \\ -\ 176 \\ \hline ??? \end{array}$$

要解决这个问题，首先从最右列着手。我们看到，6 是大于 3 的，因此从 5 上借 1，再用 13 减去 6，得到结果为 7。由于我们已经在 5 上借了 1，因此，现在实际上那一位是 4，而 4 是小于 7 的，因此继续从 2 上借 1，14 减 7 结果为 7。而由于在 2 上借了 1，实际上这一位是 1，从中减去 1，结果为 0。因此，最后的结果应为 77：