要用 16 比特的二进制数设定内存地址。JP 指令跳转的目的地为 00010000，即“LOOP:”标签所标示的语句“LD A, 0”对应的内存地址。把这个地址扩充为 16 比特就是“00000000 00010000”。要扩充到 16 位，只需要把高 8 位全部设为 0 就可以了。

还有一点希望诸位注意，在将一个 2 字节的数据存储到内存时，存储顺序是低 8 位在前、高 8 位在后（也就是逆序存储）。这样的存储顺序叫作“小端序”（Little Endian），与此相反，将数据由高位到低位顺序地存储到内存的存储顺序则叫作“大端序”（Big Endian）。根据 CPU 种类的不同，有的 CPU 使用大端序，有的 CPU 使用小端序。Z80 CPU 使用的是小端序，因此 JP LOOP 对应的机器语言为“11000011 00010000 00000000”。

```
地址          汇编语言          机器语言
00010100    JP LOOP         11000011 00010000 00000000
```

手工汇编至此就结束了。自己写的汇编语言程序，又通过自己的双手转换成了机器语言，我们应该为此感到骄傲。

3.6 尝试估算程序的执行时间

在本章的最后，介绍一下如何通过时钟周期数估算程序的执行时间。请先向前翻到表 3.2，找出执行每条汇编语言指令所需的时钟周期数。然后把代码清单 3.2 中所用到的每条指令的时钟周期数累加起来。于是可以算出到 LOOP 标签为止的 8 条指令共需要 7 + 11 + 7 + 11 + 7 + 11 + 7 + 11 = 72 个时钟周期；LOOP 标签之后的 3 条指令共需要 11 + 11 + 10 = 32 个时钟周期。因为微型计算机采用的是 2.5MHz 的晶振，也就是 1 秒可以产生 250 万个时钟周期，所以每个时钟周期是 1 秒 ÷ 250 万 = 0.0000004 秒 = 0.4 微秒。72 个时钟周期就是 72 × 0.4 = 28.8 微秒；

32 个时钟周期就是 12.8 微秒。这段程序是用 LED 的亮或灭来表示指拨开关的开关状态，所以 LOOP 标签之后所执行的操作“输入、输出、跳转”每 1 秒可以反复执行 1 秒 ÷ 12.8 微秒 / 次 = 78125 次之多，可见计算机的计算速度有多么惊人。

☆　　☆　　☆

比起 C 语言或 BASIC 等高级语言，汇编语言的语法简单、指令数少，说不定会更加容易学习，可是今天还在使用汇编语言的人却是凤毛麟角了。使用汇编语言编程时，因为要事无巨细地列出计算机的行为，所以程序会变得冗长繁复。因此诸位只在纸上体验汇编语言、机器语言以及手工汇编就足够了。只要具备了这些知识，即便是用 C 语言或 BASIC 等编程语言编程时，也一样能感受到计算机底层的工作方式，也就是说变得更加了解计算机了。

在接下来的第 4 章中，笔者将要介绍条件分支和循环等“程序的流程”，还会稍微介绍一些有关“算法”的内容。敬请期待！

第4章 程序像河水一样流动着

热身问答

在阅读本章内容前，让我们先回答下面的几个问题来热热身吧。

初级问题

Flow Chart 的中文意思是什么?

中级问题

请说出自然界中河流的三种流动方式。

高级问题

事件驱动是什么?

怎么样？被这么一问，是不是发现有一些问题无法简单地解释清楚呢？下面，笔者就公布答案并解释。

答案

初级问题：流程图。

中级问题：向着一个方向流淌；流着流着产生支流；卷成漩涡。

高级问题：用户的操作等产生事件后，由事件决定程序的流程。

解释

初级问题：流程图（Flow Chart）是指用图的形式表示程序的流程。

中级问题：与河流的流动方式一样，程序的流程也分为三种。在程序中，把犹如水流向着一个方向流淌的流程称作“顺序执行”；把犹如水流流着流着产生了支流的流程称作“条件分支”；把犹如水流卷成漩涡的流程称作“循环”。

高级问题：Windows 应用程序的运行就是由事件驱动的。例如，选择“打开文件”菜单项就能打开一个窗口，在里面可以指定要打开文件的名称和存储位置。之所以能够这样是因为一旦触发了“选中了菜单项”这个事件，程序的流程就相应地流转到了处理打开窗口的那部分。

本章重点

本章的主题是程序的流程。程序员一般都是先考虑程序的流程再开始编写程序的。只有编写过程序的人才能体会到“程序是流动着的”。一个人编写的程序如果不能按照预期运行，就说明他还没有很好地掌握“程序是流动着的”这一概念。

为什么说“程序是流动着的”呢？因为作为计算机大脑的 CPU 在同一时刻基本上只能够解释、执行一条指令。把指令和作为指令操作对象的数据排列起来就形成了程序。请想象把若干条指令一条挨一条地依次排列到一条长长的纸带上。然后把这条纸带展开抻平，从顶端开始依次解释并执行上面的每条指令，这样看起来程序就好像流动起来了。这就是程序的流程。但是程序的流程并不是只有这一种。那么下面笔者就先介绍一下程序流程的种类吧。

4.1 程序的流程分为三种

诸位读到此处，应该能够从硬件上想像出计算机的运作方式了吧。计算机的硬件系统由 CPU、I/O 和内存三部分构成。内存中存储着程序，也就是指令和数据。CPU 配合着由时钟发生器发出的滴答滴答的时钟信号，从内存中读出指令，然后再依次对其进行解释和执行。

CPU 中有各种各样的各司其职的寄存器。其中有一个被称为 PC（Program Counter，程序计数器）的寄存器，负责存储内存地址，该地址指向下一条即将执行的指令。每解释执行完一条指令，PC 寄存器的值就会自动被更新为下一条指令的地址。

PC 寄存器的值在大多数情况下只会增加。下面假设 PC 寄存器正指向内存中一个从 10 号地址开始的 3 字节指令。CPU 解释执行完这条

指令后，PC 寄存器中的值就变成 10 + 3 = 13 了。也就是说，程序基本上是从内存中的低地址（编号较小的地址）开始，向着高地址（编号较大的地址）流下去的 。我们把程序的这种流动称为“顺序执行”（如图 4.1 所示）。

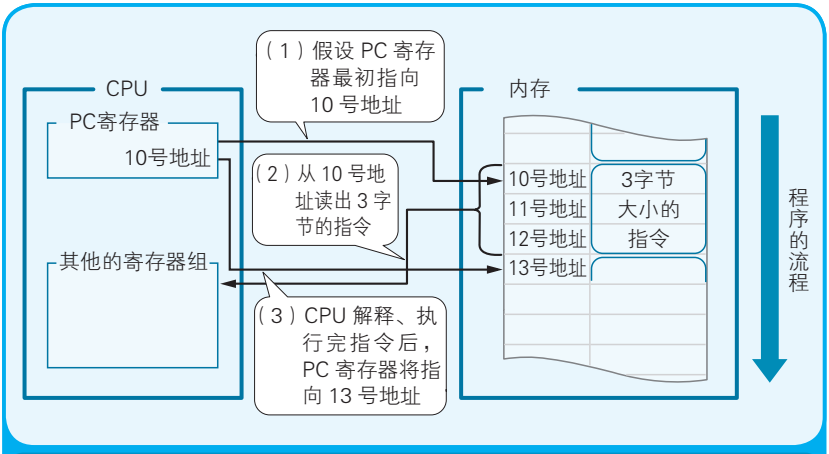

图 4.1　硬件上的程序流程（顺序执行）

程序的流程总共有三种。除了顺序执行以外，还有“条件分支”和“循环”。因为只有这三种，记忆起来还是很轻松的吧。

正如上文所述，顺序执行是按照指令记录在内存中的先后顺序依次执行的一种流程。而循环则是在程序的特定范围内反复执行若干次的一种流程。条件分支是根据若干个条件的成立与否，在程序的流程中产生若干个分支的一种流程。无论规模多么大多么复杂的程序，都是通过把以上三种流程组合起来实现的。

程序的三种流程正像是河流本身。从高山的泉眼中涌出的清泉形成了河流的源头（程序执行的起点）。水流从山中缓缓流下，有时向着

一个方向流淌（顺序执行），有时中途分出了支流（条件分支），还有时由于地势卷起了漩涡（循环）。难道诸位不认为程序的流程也很美吗？完全就像是裱在画轴上的山水画一样（如图 4.2 所示）。还有一种称作“无条件分支”的流程，它就仿佛是大雨瓢泼引发的泥石流，突然就向着某处流去了，可以认为这是一种特殊的条件分支。

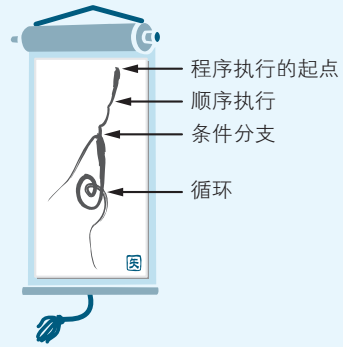

图 4.2 程序的三种流程

虽然可能不如山水画那样优美，但是我还是要给诸位展示一段简单的程序。代码清单 4.1 中列出了用 VBScript（Visual Basic Scripting Edition）编写的“石头剪刀布游戏”的代码，VBScript 是 BASIC 语言的一个版本。该程序可以在 Windows 98/Me/2000/XP 操作系统上运行[①]。玩家和计算机可以进行五轮石头剪刀布比赛，比完后会显示玩家获胜的次数。

请诸位用记事本等文本编辑器编写这个程序，并存储到扩展名为

① 用于执行 VBScript 程序的 WSH（Windows Script Host）已作为标准组件，被集成进 Windows 98/Me/2000/XP 操作系统。

“.vbs”的文件中，比如 ShitouJiandaoBu.vbs。只要双击保存后的文件，程序就可以执行了（如图 4.3 所示）。

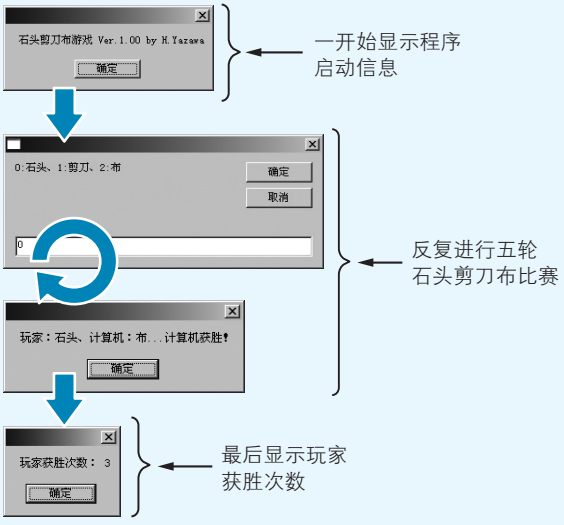

图 4.3　代码清单 4.1 执行后的结果

代码清单 4.1　用 VBScript 编写的“石头剪刀布游戏”

```
' 初始化表示手势的变量
Dim gesture(2)
gesture(0) = "石头"
gesture(1) = "剪刀"
gesture(2) = "布"

' 初始化对玩家获胜次数计数的变量
wins = 0

' 初始化随机数种子
Randomize

' 显示程序启动信息
MsgBox  "石头剪刀布游戏 Ver.1.00 by H.Yazawa"
```

```
' 进行五轮比试
For i = 1 To 5
    ' 输入玩家的手势
    user = CInt(InputBox("0：石头、1：剪刀、2：布"))

    ' 用随机数决定计算机的手势
    computer = CInt(Rnd * 2)

    ' 生成提示双方出的手势的字符串
    s = "玩家：" & gesture(user) & "、计算机：" & gesture(computer)

    ' 判定胜负，显示结果
    If user = computer Then
        MsgBox s & "...平局！"
    ElseIf computer = (user + 1) Mod 3 Then
        MsgBox s & "...玩家获胜！"
        wins = wins + 1
    Else
        MsgBox s & "...计算机获胜！"
    End If
Next

' 显示玩家的获胜次数
MsgBox "玩家获胜次数：" & wins
```

4.2 用流程图表示程序的流程

代码清单 4.1 所示的“石头剪刀布游戏”的程序是由顺序执行、条件分支和循环三种流程组成的。对于没有学过 VBScript 的人来说，也许会觉得程序代码就好像是魔法的咒语一样。因此就需要用一种无论是谁都能明白的方法来表示代码清单 4.1 中的程序。为此所使用的图表，就是诸位都已经知道的“流程图”。

所谓流程图，正如其名，就是表示程序流程（Flow）的图（Chart）。有很多专业的程序员，他们在编写程序前，都会通过画流程图或是类似的图来思考程序的流程（如图 4.4 所示）。

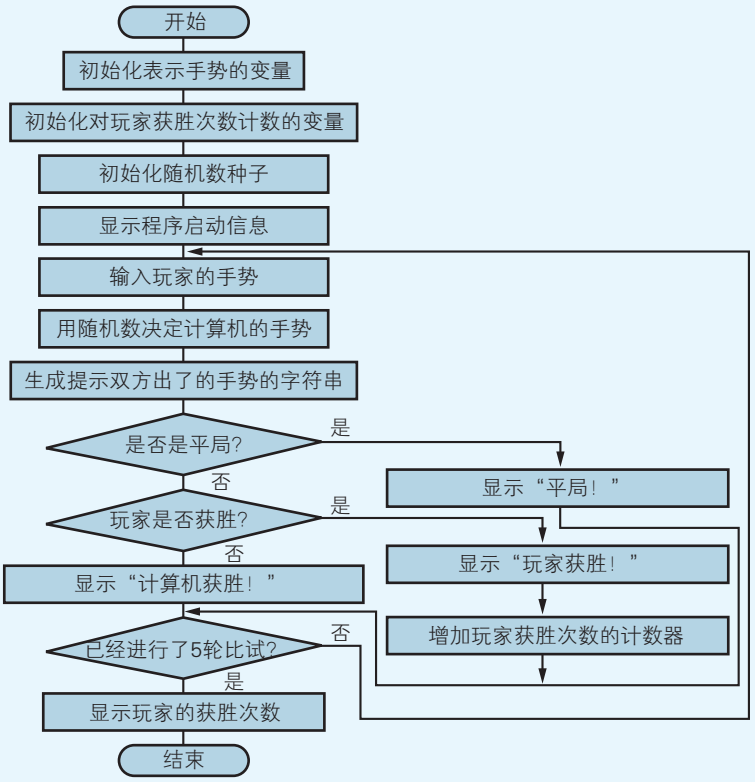

图 4.4　用流程图表示的“石头剪刀布”游戏

流程图的方便之处在于它并不依赖于特定的编程语言。图 4.4 的流程图所表示的流程不仅能转换成 VBScript 程序，还可以转换成用其他语言编写的程序，比如 C 语言或 Java 语言。可以认为编程语言只不过是将流程图上的流程用文字（程序）重现出来罢了。各种编程语言的差异正如一种自然语言中各地方言的差异一样。只要给出了详细的流程图，就可以编写出基本相同的程序。笔者也曾有过这样的经历，画流