

本章重点

要想彻底掌握计算机的工作原理，最好的方法就是自己搜集零件，试着组装一台微型计算机。微型计算机 (MicroCom) 是 Micro Computer 的缩写，字面含义是微小的计算机，但一般也可用于指代 IC 元件外露的、用于控制的计算机。因为要制作一台真正的微型计算机既花时间又花金钱，所以本章就在纸上体验一下微型计算机的制作过程吧。需要让诸位准备的只有如图 2.1 所示的电路图和一根红铅笔。将电路图复印下来后，请诸位一边想象着元件之间传输的信号的作用，一边用红铅笔描画出笔者所介绍的电路，以此来代替实际的布线环节。当所有的电路都描红了，微型计算机也就完成了。

别看只是描了描线，却一样能学到很多知识，甚至可以说是不费吹灰之力就能了解计算机的工作原理。从此之后不但消除了对硬件的恐惧感，而且还会感到和计算机更加亲近了。请诸位一定要借此机会体验微型计算机的制作过程。

2.1 制作微型计算机所必需的元件

首先让我们来收集元件吧。制作微型计算机所需的基础元件只有 3 个，CPU、内存和 I/O，每种元件都是作为一块独立的 IC 在市场上出售的。CPU 是计算机的大脑，负责解释、执行程序。内存负责存储程序和数据。I/O 是 Input/Output (输入 / 输出) 的缩写，负责将计算机和外部设备 (周边设备) 连接在一起。

这里我们使用 Z80 CPU 作为微型计算机的 CPU、TC5517 作为内

1. 为了易于理解, 有些引脚的名称或有改动, 不同于厂商原始说明中的名称
2. 对于CPU、内存、IO也可以使用兼容元件

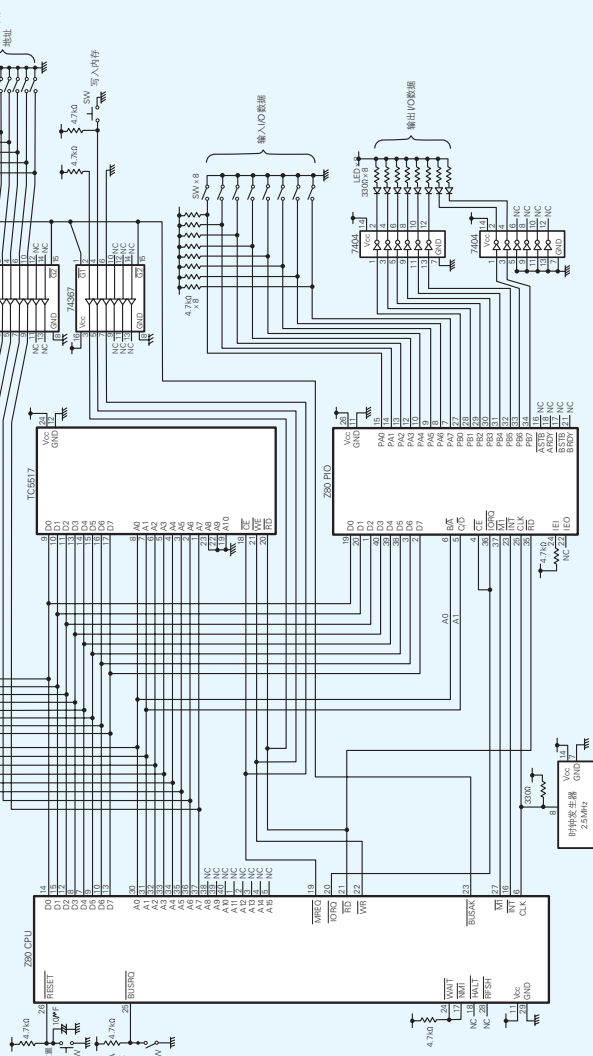


图 2.1 Z80 微型计算机的电路图 (本书末尾附有更大的电路图)

存、Z80 PIO 作为 I/O。Z80 CPU 是一款古老的 CPU，在 NEC 的 PC-8801、SHARP 的 MZ-80 等 8 比特计算机广泛应用的时代，曾以爆炸般的速度普及过。TC5517 是可以存储 2K 的 8 比特数据的内存。在计算机的世界里，K 表示 $2^{10} = 1024$ 。TC5517 的容量是 8 比特 $\times 2 \times 1024 = 16384$ 比特，即 2K 字节。虽然这点容量与诸位所使用的个人计算机比起来相差悬殊，但是对于用于学习的微型计算机来说是绰绰有余了。Z80 PIO 作为 I/O，经常与 Z80 CPU 一起使用。正如其名，PIO（Parallel I/O，并行输入/输出）可以在微型计算机和外部设备之间并行地（一排一排地）输入输出 8 比特的数据。在计算机爱好者们沉浸在制作微型计算机的那个年代，这些元件都是常见的 IC。这里要先跟诸位事打声招呼，这里制作的微型计算机终归只是用于学习的模型，并没有什么实用的价值。

为了制作微型计算机，除了 CPU、内存和 I/O，还需要若干辅助元件。

为了驱动 CPU 运转，称为“时钟信号”的电信号必不可少。这种电信号就好像带有一个时钟，滴答滴答地每隔一定时间就变换一次电压的高低（如图 2.2 所示）。输出时钟信号的元件叫作“时钟发生器”。时钟发生器中带有晶振，根据其自身的频率（振动的次数）产生时钟信号。时钟信号的频率可以衡量 CPU 的运转速度。这里使用的是 2.5MHz（兆赫兹）的时钟发生器。

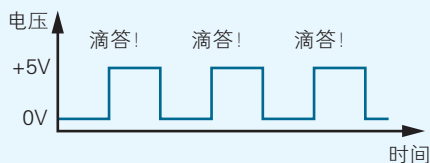


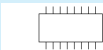
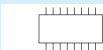
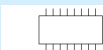




图 2.2 时钟信号的波形图

用于输入程序的装置也是必不可少的。在这里我们通过拨动指拨开关来输入程序，指拨开关是一种由 8 个开关并联连在一起构成的元件（如照片 2.1(a) 所示）。输出程序执行结果的装置是 8 个 LED（发光二极管）。到此为止，主要的元件就都备齐了。









剩下的就都是些细碎的元件了。表 2.1 是所需元件的一览表，里面也包含了之前介绍过的元件。请诸位粗略地浏览一遍。所需元件表中的 74367 和 7404 也是 IC，用于提高连接外部设备时的稳定性。

电阻是用于阻碍电流流动、降低电压值的元件。为了省去布线的麻烦，这里也会使用将 8 个电阻集成到 1 个元件中的集成电阻（如照片 2.1(b) 所示）。电阻的单位是 Ω （欧姆）。电容是存储电荷的元件，衡量存储电荷能力的单位是 F（法拉）。要让微型计算机运转起来，5V（伏特）的直流电源是必不可少的。于是还需要使用一个叫作“开关式稳压电源”的装置，将 220V 的交流电变成 5V 的直流电。

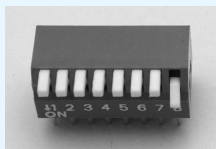
表 2.1 本次用到的制作微型计算机的元件

元件名称	数量	电路图符号	说明
Z80 CPU	1		CPU（8 比特 CPU）
TC5517	1		内存（8 比特 × 2K）
Z80 PIO	1		I/O（8 比特 × 2 个并口 I/O）
74367	4		三态总线缓冲器
7404	2		六反相器
时钟发生器	1		2.5MHz
指拨开关（DIP switch）	3		用于切换开 / 关状态（8 比特）

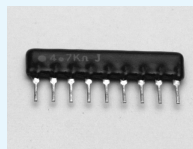
(续)

元件名称	数量	电路图符号	说明
按键开关 (Push switch)	2		平时处于关的状态, 按下后电路连通, 手指离开后由内部的弹簧弹回关的状态
快动开关 (Snap switch)	2		用于切换开 / 关状态
集成电阻	3		$4.7\text{k}\Omega \times 8$ 个 ($1/4\text{W}$)
集成电阻	1		$330\Omega \times 8$ 个 ($1/4\text{W}$)
电阻	6		$4.7\text{k}\Omega$ ($1/4\text{W}$)
电阻	1		330Ω ($1/4\text{W}$)
电容	1		$10\mu\text{F}$ (25V)
LED	8		颜色任意
开关式稳压电源	1	在电路图中省略了该元件	用于将 220V 的交流电转换为 5V 的直流电
用于连接各元件的导线	适量	用直线表示	AWG 30 号线很好用

(a) 指拨开关



(b) 集成电阻



照片 2.1 指拨开关和集成电阻

2.2 电路图的读法

在开始布线之前, 先来介绍一下电路图的读法。在电路图中, 用连接着各种元件符号的直线表示如何布线。电路中有些地方有交叉,

但若只是交叉在一起的话，并不表示电路在交叉处构成通路。只有在交叉处再画上一个小黑点才表示构成通路。

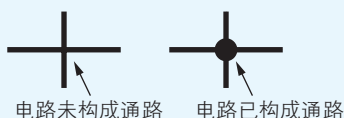


图 2.3 判断电路交叉时是否构成通路

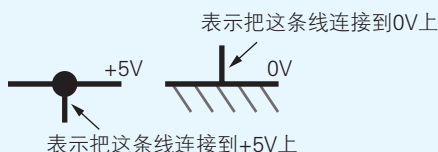


图 2.4 电源的表示方法

本次制作的微型计算机工作在 +5V 的直流电下。虽然在实际的电路中要把 +5V 和 0V 连接到各个元件的各个引脚上，但是如果在电路图中也把这些地方都一一标示出来的话，就会因为到处都是 +5V 和 0V 的布线而显得混乱不堪了。所以要使用如图 2.4 所示的两种电路图符号来分别表示电路连接到 +5V 和连接到 0V 的情况。

IC 的引脚（所谓引脚就是 IC 边缘露出的像蜈蚣腿一样的部分）按照逆时针方向依次带有一个从 1 开始递增的序号。数引脚序号时，要先把表示正方向的标志，比如半圆形的缺口，朝向左侧。举例来说，带有 14 个引脚的 7404，其引脚序号就如图 2.5 所示。

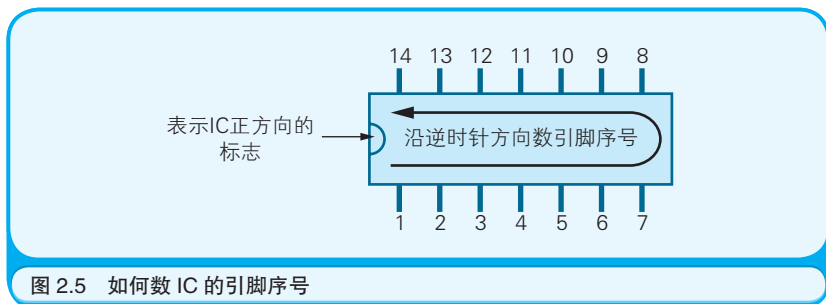


图 2.5 如何数 IC 的引脚序号

如果按照引脚序号的排列顺序来画 IC 的电路图符号，那么标示如何布线时就会很不方便。所以通常所绘制的电路图都不受引脚实际分布的限制^①。画图时，在引脚的旁边写上引脚的序号，在表示 IC 的矩形符号中写上表明该引脚作用的代号。代号就是像 RD（Read）表示执行读取操作，WR（Write）表示执行写入操作这样的代表了某种操作的符号。各个代号的含义等到为引脚布线时再一一说明^②。

2.3 连接电源、数据和地址总线

下面就开始布线吧。请假想自己正在制作微型计算机，并按照如下的说明用红铅笔在电路图中描画相应的电路。

首先连接电源。IC 与普通的电器一样，只有接通了电源才能工作。Z80 CPU、TC5517 和 Z80 PIO 上都分别带有 Vcc 引脚和 GND 引脚。Vcc 和 GND 这一对儿引脚用于为 IC 供电。下面请先将 +5V 电源连接

- ① 有时也会遵循引脚序号的顺序绘制电路图，这样的电路图叫作实物布线图。
- ② 写在引脚旁边的代号，其含义会写在 IC 生产厂商发布的资料中，但在这里为了保持文章的通俗易懂，改变了一部分代号的写法，这一点还望诸位谅解。例如，在厂商的资料中 TC5517 的第 20 个引脚的代号是 OE（Output Enable，输出使能），在这里则改为了含义相同的 RD（Read，读取）。

到各个 IC 的 Vcc 引脚上，然后将 0V 电源连接到各个 IC 的 GND 引脚上。接下来还需要将 +5V 和 0V 连接到时钟发生器上。接通电源后这些 IC 和时钟发生器就可以工作了。

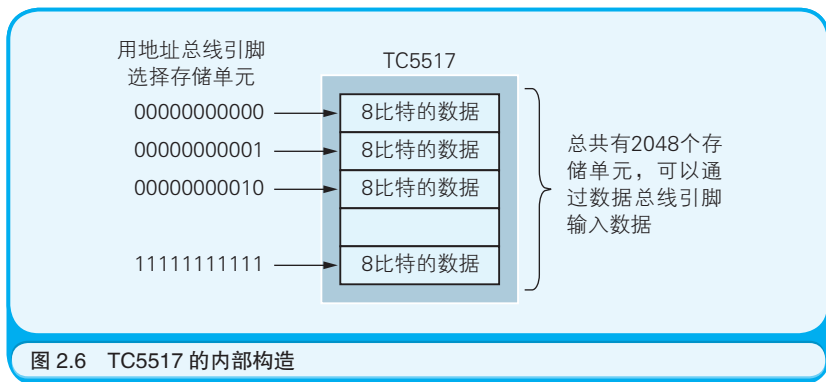
微型计算机所使用的 IC 属于数字 IC。在数字 IC 中，每个引脚上的电压要么是 0V、要么是 +5V，通过这两个电压与其他的 IC 进行电信号的收发。用于给 IC 供电的 Vcc 引脚和 GND 引脚上的电压是恒定不变的 +5V 和 0V，但是其他引脚上的电压，会随着计算机的操作在 +5V 和 0V 之间不断地变化。

稍微说一点题外话，只要想成 0V 表示数字 0、+5V 表示数字 1，那么数字 IC 就是在用二进制数的形式收发信息。也正因为如此，二进制数在计算机当中才如此重要。有关二进制的内容，本书并不会详细介绍，但是请先记住以下知识点：通常将 1 个二进制数（也就是数字 IC 上 1 个引脚所能表示的 0 或者 1）所表示的信息称作“1 比特”，将 8 个二进制数（也就是 8 比特）称作“1 字节”。比特是信息的最小单位，字节是信息的基本单位。这里制作的微型计算机是一台 8 比特微型计算机，因此是以 8 比特为一个单位收发信息的。

下面回到正题。计算机以 CPU 为中心运转。CPU 可以与内存或 I/O 进行数据的输入输出。为了指定输入输出数据时的源头或目的地，CPU 上备有“地址总线引脚”。Z80 CPU 的地址总线引脚共有 16 个，用代号 A0~A15 表示，其中的 A 表示 Address（地址）。后面的数字 0~15 表示一个 16 位的二进制数中各个数字的位置，0 对应最后一位、15 对应第一位。16 个地址总线引脚所能指定的地址共有 65536 个，用二进制数表示的话就是 0000000000000000~1111111111111111。因此 Z80 CPU 可以指定 65536 个数据存取单元（内存存储单元或 I/O 地址），进行信息的输入输出。

一旦指定了存取数据的地址，就可以使用数据总线引脚进行数据的输入输出了。Z80 CPU 的数据总线引脚共有 8 个，用代号 D0~D7 表示。其中的 D 表示 Data（数据），后面的数字 0~7 与地址总线引脚代号的规则相同，也表示二进制数中各个数字的位置。Z80 CPU 可以一次性地输入输出 8 比特的数据，这就意味着如果想要输入输出位数（比特数）大于 8 比特的数据，就要以 8 比特为单位切分这个数据。

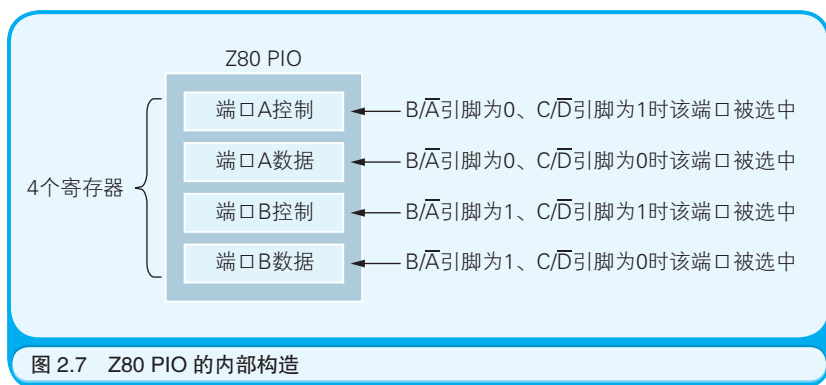
作为内存的 TC5517 上也有地址总线引脚（A0~A10）和数据总线引脚（D0~D7）。这些引脚需要同 Z80 CPU 上带有相同代号的引脚相连。一块 TC5517 上可以存储 2048 个 8 比特的数据（如图 2.6 所示）。可是由于用于输入程序的指拨开关是以 8 比特为一个单位指定内存地址的，所以我们只使用 TC5517 上的 A0~A7 这 8 个引脚，并把剩余的 A8~A10 引脚连接到 0V 上（这些引脚上的值永远是 0）。虽然总共有 2048 个存储单元，最终却只能使用其中的 256 个，稍微有些浪费。下面就请诸位用红铅笔把 Z80 CPU 和 TC5517 的 D0~D7 以及 A0~A7 引脚分别连接起来。



2.4 连接 I/O

下面开始连接 I/O。只有了解了作为 I/O 的 Z80 PIO 的结构，才能理解为什么要这样布线。诸位都知道“寄存器”这个词吗？寄存器是位于 CPU 和 I/O 中的数据存储器。Z80 PIO 上共有 4 个寄存器。2 个用于设定 PIO 本身的功能，2 个用于存储与外部设备进行输入输出的数据。

这 4 个寄存器分别叫作端口 A 控制、端口 A 数据、端口 B 控制和端口 B 数据。所谓端口就是 I/O 与外部设备之间输入输出数据的场所，可以把端口（Port）想象成是轮船装卸货物的港口。Z80 PIO 有 2 个端口，端口 A 和端口 B，最多可以连接 2 个用于输入输出 8 比特数据的外部设备（如图 2.7 所示）。



既然已经大体上了解了 Z80 PIO 的结构，下面就开始布线吧。因为 Z80 PIO 上也有 D0~D7 的数据总线引脚，所以先把它和 Z80 CPU 中带有同样代号的引脚连接起来。这样 CPU 和 PIO 就能使用这 8 个引脚交换数据了。