

微型计算机在结构 工程中的应用

[英]W·H·莫斯利 W·J·斯潘塞 著
毕少林 孙世魁 魏庆礼 译

中国建筑工业出版社

六、七章探讨了微机在结构分析和设计的专门领域中更具体的应用。虽然在第六、七章中参考的是《英国实用规范》，但依据这些规范的有关内容都清楚地标出了，因此可以容易地进行相应的改动，以适应其它有关的规范。

最后一章简述了计算机图形学的一些基本原理。从工程的角度来说，计算机图形学可以进一步改进我们已经熟知的一些计算机在结构分析和设计中的应用。例如，在某种程度上，可以直接以图形的形式进行框架分析数据的输入和输出。如果仔细地研究过框架分析问题，就能很好地理解这一点。

存放本书中所列出的一些用于Apple型微机的程序的磁盘，可以直接从出版者那里得到。

希望本书有益于从事结构工程的实习工程师和大学生。

W·H·莫斯利

W·J·斯潘塞

本书前一部分简明地介绍了微型计算机系统，BASIC 语言和个人计算机系统的结构。后一部分介绍程序编制方法及建筑设计程序的编制技巧，并列举了平面框架、格架、钢结构、钢筋混凝土结构等一系列实用程序。每个程序均由程序框图、程序步骤、程序清单及变量说明、例子等几部分组成。书中所给程序可以直接在 Apple II 机上运行。书的最后介绍了微机绘图。

本书适用于土建结构设计人员、科研人员、大专院校有关师生。

Microcomputer Applications in Structural Engineering
W.H.Mosley W.J.Spencer MACMILLAN PRESS,
LONDON, 1984

* * *

微型计算机在结构工程中的应用

毕少林 孙世魁 魏庆礼 译

中国建筑工业出版社出版(北京西郊百万庄)
新华书店北京发行所发行 各地新华书店经售
北京市平谷县大华山印刷厂印刷

开本: 850×1168毫米1/32 印张: 8³/₄ 字数: 223千字

1988年3月第一版 1988年3月第一次印刷

印数: 1—7,980册 定价: 2.70元

ISBN 7-112-00059-0/TU·2

统一书号: 15040·5370

前 言

本书从实用上介绍了如何用微型计算机（后面简称微机）来进行结构分析和设计，特别着重于微机作为个人机的作用，并对这方面的内容进行了详细的讨论。

虽然这本书在内容上看起来分为明显的两部分，但它绝不是微机系统或结构工程的专业著作。与此相反，我们试图把传统的结构工程和基本的计算机技术结合起来，从而使工程师们对微机有所了解。如果需要深入了解这方面的工作，还可以参考其它的教科书（本书提供了进一步阅读的书目）。有许多很好的关于微机、结构分析和计算机辅助设计的书，但通常的情况是，要把这三方面结合起来并获得预期的结果，工程师们都不具备必要的基础知识和经验。

本书比较详细地介绍了BASIC程序语言，给出的一些例子也都是用BASIC语言写成。这是目前在微机上应用最广泛的语言，不过其他的语言也可以在微机上使用，因此我们还试图尽可能地说明基本的原理和概念。

第一章对微机系统进行了简单的介绍，并介绍了用机器码编程的方法。第二章介绍了BASIC程序语言，虽然也谈到了一些更一般的编程技术，但实例都由BASIC语言给出。在第三章中讨论了个人计算机系统的构成和编程中文件的使用问题。读完了第一、二、三章之后，读者对微机系统和程序语言的使用就会有一个相当的了解。随后的章节把前面的知识应用到专门的结构工程问题中去。

第四章用来给出一系列与结构分析和设计有关的简单例子。挑选这些例子是由于它们简单并容易在微机系统上实现。第五、

目 录

前言

1	微型计算机介绍	1
1.1	导言	1
1.2	微机系统	2
1.2.1	微处理器	3
1.2.2	计算机存储器	3
1.3	微机中的程序和变量存储	7
1.3.1	脱机存储	8
1.4	编微机程序	9
1.4.1	用机器码编程	9
1.4.2	监控程序的使用	16
1.5	如何挑选一台微机	18
	阅读材料	20
2	在微机上用BASIC语言编程	21
2.1	用BASIC编程的基本点	23
2.1.1	一个BASIC程序	23
2.1.2	操作系统简介	24
2.2	变量、运算符和赋值语句	25
2.2.1	数字变量和数字常量	25
2.2.2	整数型变量	26
2.2.3	字符串变量和字符串常量	26
2.2.4	运算符	27
2.2.5	赋值语句	27
2.3	数据的输入和输出	28
2.3.1	READ-DATA语句	28
2.3.2	INPUT语句	28
2.3.3	PRINT语句	29

2.4	控制语句	30
2.4.1	GOTO和ON GOTO语句	30
2.4.2	IF语句	30
2.4.3	FOR-NEXT循环	31
2.4.4	子程序和GOSUB, RETURN语句	32
2.5	函数	33
2.5.1	算术函数和三角函数	34
2.5.2	字符串函数	34
2.5.3	用户定义函数	35
2.6	数组和其他一些特性	36
2.6.1	下标变量和DIMension语句	36
2.6.2	下标变量应用的例子	36
2.7	编程序的一些概念	41
2.7.1	编程序的原则	41
2.7.2	程序的编辑、检查和调试	42
2.7.3	模块式程序要点	45
2.7.4	提高存储效率和速度	52
2.7.5	BASIC的其他特性	54
	阅读材料	56
3	个人计算机系统的结构	57
3.1	文件的概念和文件的存储	57
3.1.1	文件类型	58
3.1.2	磁盘操作系统和磁盘初始化	58
3.1.3	数据文件	59
3.2	其它的文件处理命令	66
3.3	文件在微机系统中的应用	68
3.4	程序资料化	71
3.4.1	程序资料	72
3.4.2	用微机存储资料文件	74
3.5	个人计算机系统的维护	78
3.5.1	备用文件	78
3.5.2	测试程序和内存检查	80

3.5.3	使用工作日记	80
3.6	扩展微机系统的应用	81
3.6.1	替换操作系统	81
3.6.2	输入/输出外部设备	82
	阅读材料	82
4	简单程序设计在结构分析和设计中的应用	83
4.1	单跨梁分析程序	83
4.1.1	梁分析程序清单	88
4.1.2	梁分析程序应用实例	91
4.2	计算变截面杆件刚度及传递系数程序	92
4.2.1	刚度和传递系数程序清单	95
4.2.2	刚度和传递系数程序应用举例	97
4.3	一般形式截面特性计算程序	98
4.3.1	截面特性计算程序清单	100
4.3.2	截面特性计算程序应用举例	102
4.4	线性方程组求解程序	103
4.4.1	线性方程组求解程序清单	104
4.4.2	线性方程组求解程序应用举例	105
4.5	群桩分析程序	106
4.5.1	群桩分析程序清单	108
4.5.2	群桩分析程序应用举例	110
	阅读材料	111
5	结构分析程序	112
5.1	刚度法说明	112
5.2	平面框架分析	115
5.2.1	平面框架程序的步骤	115
5.2.2	平面框架程序符号规则	122
5.2.3	平面框架程序清单和所使用的变量	122
5.2.4	平面框架程序应用举例说明	131
5.2.5	平面框架程序的进一步开发	133
5.3	平面格架分析	134
5.3.1	平面框架程序修改为平面格架程序	135

5.3.2	平面格架程序修改行清单	136
5.3.3	平面格架程序应用例子说明	138
5.4	变截面杆件	140
	阅读材料.....	142
6	钢结构设计程序.....	143
6.1	钢柱设计	143
6.1.1	钢柱设计框图	144
6.1.2	钢柱设计程序主要步骤	144
6.1.3	钢柱设计程序的附加选择	153
6.1.4	钢柱设计程序及变量清单	153
6.1.5	程序用于钢柱设计例子的说明	159
6.2	钢梁设计	161
6.2.1	钢梁设计框图	163
6.2.2	钢梁设计程序步骤	163
6.2.3	钢梁设计程序清单及所使用的变量	168
6.2.4	举例说明钢柱设计程序的使用	174
	阅读材料.....	175
7	钢筋混凝土结构设计程序.....	176
7.1	结构分析	176
7.1.1	等代框架分析程序框图	180
7.1.2	等代框架分析程序步骤说明	182
7.1.3	等代框架分析程序的进一步改进	188
7.1.4	等代框架分析程序及其变量清单	188
7.1.5	等代框架分析程序应用举例说明	195
7.2	钢筋混凝土梁设计	200
7.2.1	长方形梁截面抗弯钢筋设计	201
7.2.2	带翼缘截面梁抗弯钢筋设计	206
7.2.3	钢筋混凝土梁设计程序及其变量清单	209
7.3	选择钢筋直径	211
7.3.1	钢筋选择程序清单	212
7.4	钢筋混凝土柱设计	213
7.5	柱钢筋的对称布置	214

7.5.1	对称配筋柱设计框图	216
7.5.2	对称配筋柱设计程序步骤说明	217
7.5.3	对称配筋柱设计程序及其变量清单	218
7.6	柱钢筋的非对称布置	221
7.6.1	非对称配筋柱设计程序清单	222
	阅读材料	223
8	微型计算机图形	224
8.1	图形屏幕	225
8.2	视觉显示器件	226
8.3	图形屏幕上的线段生成	227
8.3.1	图形模式	228
8.3.2	色彩选择	228
8.3.3	画线段命令	229
8.4	图形软件	229
8.4.1	基本的图形操作	230
8.4.2	画一个两维物体——一个图形软件的例子	232
8.4.3	文字与图形	236
8.5	用于图形的外部设备	237
8.5.1	硬拷贝设备和打印机	237
8.5.2	绘图设备	238
8.5.3	图形输入板和数字化仪	238
8.5.4	其他的输入设备	240
8.6	计算机图形的组织	240
8.7	计算机图形程序举例	242
8.7.1	一个通用绘图程序	242
8.7.2	绘制弯矩图	248
8.8	计算机图形系统	249
	阅读材料	250
	附录: 十进制、十六进制和二进制	251

1 微型计算机介绍

1.1 引言

微机作为个人用计算机的这一发展，掀起了计算机技术的第三次浪潮，它也波及到了工程师们。同时，这一进展也对使用者提出了新的要求：需要对计算机系统有更深入的了解，以及由于操作的个人性质，使用者本人要承担更多的责任。对于大型机和小型机两者来说，无论是在机房内使用还是享用远距离分时终端，使用者一般不必操心计算机的管理和组织方面的工作。然而，照看微机本身和它的运算过程，现在对一个使用者来说却是极其普通的事情。这种新出现的灵活性——如果是这样认为，也要求使用者本人对机器更了解、更负责。由于微机价格低廉和购买方便，使一些小公司甚至个人都可以获得这样的一台机器。大公司也认为用许多个人机来补充其它的计算设备是有利的。总之，为了得到这种投资的最大效益，工程师们应该对微机的构造及运行方式有一个基本的了解，这一点是很重要的。本章的目的就是要提供这种了解所需要的基础知识，以适应实习工程师和工程专业学生们的要求。迄今为止他们或许更关心计算机的应用，而不是计算机系统或计算机硬件。

书中参考了两种特定的微机，它们都是8位机。所谓8位机就是在一个操作周期中有8位数据同时被传送或操作。按此可以很方便地对微机进行分类，现有的8位机种类很多，例如Apple II，Commodore PET，Tandy TRS-80等等。在介绍微机运行的概念时，应该清楚所指的都是8位机。不过当需要时，可以容易地把这些概念扩展到其他的机型，但是在许多情形下微机的

分类与这些概念是毫不相关的。16位微机的技术也有了足够的进展,有几种16位机即将商品化,更加强有力的机器也在开发之中。由于16位微机能够在一次操作中传送16位数据,所以它要比8位机运行快得多。另外,16位机通常设计有更大的存储空间,在1.2.2节中将会谈到这一点。

1.2 微机系统

一个可以在工程中实用的最小系统包括:
中央处理器。

读/写存储器(随机存储器或RAM)。

只读存储器(ROM)。

视觉显示单元和键盘。

盒式磁带录音机,和/或磁盘驱动器。

打印机。

图1.1所示就是这样的一个系统。下面将对系统中每一单元的功能进行简短的讨论,尽管这些功能已广为人们所熟知。有些关于存储容量的数字比较难于解释。例如,有16K字节的读/写存储或1.5兆字节的磁盘意味着什么?对这些问题的回答依赖于期望微机去做什么,对此后面将给出进一步的说明。现在先让我们看一下微机的核心并由此讲起。

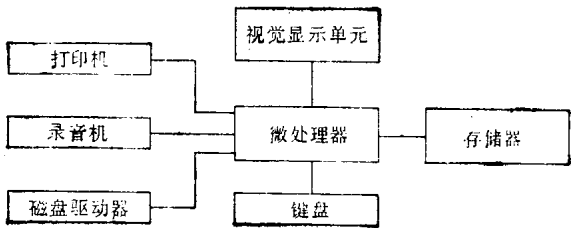


图 1.1 微机系统

1.2.1 微处理器

微处理器是微机的中央处理单元。它是一个超大规模的集成电路，执行实际的运算并控制微机的其他部件。微处理器包括一系列的寄存器和累加器，用来寄存和处理一定位数的信息，每一位信息代表一位二进制的数字。根据给定的指令，可以对存储区存取信息，也可以在微处理器的不同寄存器和累加器之间传送信息，指令本身也要进行类似的处理。数字计算机在二进制算术的水平上运行，0和1这两个数字可以用“开”和“关”或“正”和“负”等类似的条件来表示。给微机的任何指令都要由微处理器分解成这种表示。

如果是定型大批量生产，制造商们现在能够生产出造价很低的微处理器，因此在几种不同的微机内或其他的装置内使用同一种微处理器是不足为奇的。看一下一些微机的制造说明书会使你受到启发，其中微处理器的详情用类型名称和运行频率给出。例如，许多8位的微机使用一种称为6502的微处理器，它的运行频率是1.8兆赫兹或0.56微秒/周。微机第二个最基本的部分是存储器，或者是一些地点，由微处理器输出的或准备输送给微处理器的一些二进制数就存放在其中。

1.2.2 计算机存储器

存储器可以被看作是一些地点，它们的地址都是已知的，8位的信息段——称为字节，可以存放在里面。这些信息包括指令和数据，在微处理器的控制下，信息在存储器和微处理器之间进行着频繁的交流，图1.2是一幅概念性的微机结构图，重点在于表示数据的传输，另外还有一些传输数据时进行控制和提供地址用的连线。在图1.2中，“数据总线”是表示信息传输通道的术语，“8位并行”指的是同时传输8位信息。某些指令必须要永久地存放在内存中（另一种办法是每一次要用计算机时再读进去），当需要微机执行相应的操作时再调用。这些指令占据的存储器称作只读存储器，或称ROM。一条“把微处理器中某个寄存器的内容存放到某个地址中去”这样的简单指令，可能由3个

或 4 个字节组成。如果在称为机器码的低级水平上来编写，200 条这样的机器码指令可能需要 400 到 600 个 ROM 字节。因为用机器码编程既乏味又困难，所以通常需要有某种形式的翻译程序，这样就可以使用高级的程序指令了。翻译程序将用户程序翻译成机器码，然后微机再按机器码运行。因此如果发现至少需要 4000 个字节，即大约 4 K 字节的只读存储器，就不会感到奇怪了。

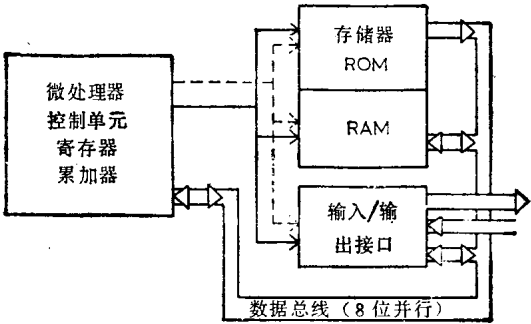


图 1.2 微机结构示意图

通常用 1 K 字节代表 1024 个字节。考虑到存储器的地址具有 8 位的字节，采用上述数字的原因就清楚了。重要的是要记住有两件事情与存储器中的每一个字节有关：其一是每一字节固定不变的地址，其二是每一字节中的 8 位二进制数以二进制形式所表示的内容。因为地址要由微处理器来使用，它也必须以二进制的形式来表达。通常用两个字节来确定一个地址。由于一个字节有 8 位，所以两字节的地址意味着从 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 到 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 整个范围内的二进制数都可以用来作为地址。当然，用二进制数来表示地址是不方便的，即使这要由微处理器来完成。更方便和通常的做法是使用十六进制数，因为一个 16 位的地址可以写成 4 位十六进制的数。

十六进制的基数是 16，也就是说它基于 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 这 16 个数字。表

1.1是一些数在十进制、十六进制和二进制之间的转换，关于十进制、十六进制和二进制的进一步说明在附录中给出。十六进制的数前通常都冠以一个\$，本书也将使用这一习惯表示。

某些十进制-十六进制-二进制的转换

表 1.1

十进制	十六进制	二进制
0	00	0000 0000
1	01	0000 0001
2	02	0000 0010
3	03	0000 0011
4	04	0000 0100
5	05	0000 0101
6	06	0000 0110
7	07	0000 0111
8	08	0000 1000
9	09	0000 1001
10	0A	0000 1010
11	0B	0000 1011
12	0C	0000 1100
13	0D	0000 1101
14	0E	0000 1110
15	0F	0000 1111
16	10	0001 0000
249	F9	1111 1001
250	FA	1111 1010
251	FB	1111 1011
252	FC	1111 1100
253	FD	1111 1101
254	FE	1111 1110
255	FF	1111 1111

由表中可见，最大的地址是\$FFFF，它代表十进制的65535，当然也对应于二进制的1111111111111111。因为地址\$0000也是可用的，所以总共有65536个可寻址的8位数字节存储单元。如果1K字节定义为1024个字节，那末65536个地址代表64K字节。当然比这小的存储量实际是存在的，但一个具有16位地址的微处理器不可能有大于64K字节的存储。

对于16位的机器，8位数字节存储的概念不变，但机器的结构允许地址长度大于16。比如说地址的长度是20位，机器将可以寻址1024K字节的存储。一个24位的地址可以寻址16兆字节的存储。

除只读存储器外，还要在存储区中存放用户建立的程序指令和数据。作此用途的那一部分存储器叫作读/写存储器（因为用户可能对那部分区域写入或读出），或称随机存储器或RAM。存储区中的其他部分有更专门的功能，例如输入/输出指令或屏幕（视觉显示单元）存储区。存储区的使用在设计微机时就已经规划好了，因此每一台计算机都有一个存储区分配图，它描述了整个存储区中各部分的功能。图1.3给出了一个简化的典型存储区分配图。在该情况下，微机可以使用BASIC语言。图1.3中左边的数字表示每一部分的十进制起始和终止地址，右边的数字是用十六进制表示的同样信息。

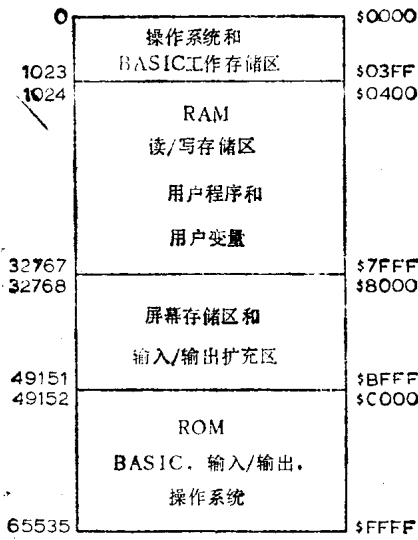


图 1.3 简化的微机存储区分配图

从图1.3中可以看到，在整个64K字节的存储区中，用户可以使用的大约为32K字节。其余的由微机用来实现执行程序所必需的操作。

应该注意到，在图中开始那部分的地址用户是不能使用的。在这种情况下，假定使用了一种特殊的寻址方式，使得微处理器

在寻址这部分存储区时比其他部分要快，而对于其他部分的地址则需要更进一步的说明。控制微机总体运行的操作系统使用了存储区的起始部分，把它作为一个工作空间来保存要频繁用到的关键信息。也可以很方便地按其它的形式来划分存储区，如64K字节可以由16个（0到15）4096字节的块来表示，而每一块又可进一步划分为16页，每一页有256个字节。各种微机存储区分配图的细节都在制造商的手册中给出。

1.3 微机中的程序和变量存储

具有诸如32K字节的读/写存储可供用户使用这一点的意义，只有对存储用户程序及相应的变量所占据的空间有所考虑时，才能够理解。大多数微机都具有可查询程序大小的功能，因此用户一开始就可查看一下有多大的存储区可供使用。这是除了实际存储本身之外的几种功能之一。当键入或装入一个程序之后，用户还可以查询一下仍可使用的存储区的大小。这时用掉的存储区只是被程序指令所占据。当程序开始运行后，各种变量（单变量，数组，串变量等）还要使用存储区，而且可能比仅仅存储程序本身所使用的空间要大得多。

要进行概括是不太容易的，下面的数字说明一种基于6502微处理器、ROM中带有BASIC翻译程序的微机的情况。粗略地说，BASIC程序中的每一条语句平均占8个字节的存储空间，每个简单变量（非整数）占7个字节。至于数组，整型数组的每个元素占2个字节，非整型数组的每个元素占5个字节。另外，还要在数组的前面存放数组说明，对于一个二维数组这需要9个字节的长度。例如，一个联立方程求解程序，假定包括80条BASIC语句，可能只需要640个字节（约0.6K字节）来存储这些指令，但要用它来求解含有20个未知数的60个方程式时，可能还需要6000个字节（约6K字节）。32K字节的读/写存储是一个有意义的数字，许多有价值的工程程序都能够以比它小的存储空间运行，但要用少于8到16K字节的读/写存储来做什么实际的工作是难

想象的。

1.3.1 脱机存储

读/写存储器（不象ROM）只有在开机之后才能保持用户的程序和数据，因此很明显某种永久性的程序和数据存储是必要的。这种存储常常被称为脱机存储或海量存储，它通常由盒式磁带系统或使用小软盘的磁盘驱动器来实现，也可能两者兼备。微机的操作系统为用户提供了可以把存储区中的程序写到磁带或磁盘上的指令，因此能够留待以后使用。比起磁盘系统来，磁带系统要简单而且便宜，但速度也慢且易于出错。许多微机使用一般的录音盒式磁带，以正常的放音速度运行，通常每秒钟传输50个字节。这样一盘30分钟的磁带一面就可以存放45000个字节（约40K字节）。许多磁盘系统使用标准的5 $\frac{1}{4}$ 英寸软盘，有时也叫作小盘。磁盘和磁盘系统有多种的技术指标，包括诸如双面或单面及双密度或单密度等，表示了不同的存储能力。一个单面单密度的磁盘大约可以存放170000个字节或约166K字节。一个4500字节的程序可以在不到20秒的时间内进行传输，由此可以得到一点运行速度的概念。表1.2是一个有关单位和速度情况的小结。

单 位 和 速 度 小 结 表 1.2

1 字节 = 8 位信息	
1024 字节 = 1 K字节	
典型的程序单位	
每个语句	8 字节
每个非下标变量	7 字节
每个整型数组元素	2 字节
每个非整型数组元素	5 字节
磁带系统——一般盒式磁带	
数据传输速率	每秒50字节
30分钟磁带的存储量	每面45000字节
磁盘系统——标准5 $\frac{1}{4}$ 英寸小盘	
数据传输速率	每秒约2250字节
存储量	每盘690块，每块256字节