

FIXING DIANZIJISUANJI JIAOCHENG

李冠英 张毅忠 郑存站 编

微
型
电
子
计
算
机
教
程



APPLE II 微型计算机 BASIC语言与磁盘操作系统

广东科技出版社

微型电子计算机教程

APPLE II 微型计算机
BASIC语言与磁盘操作系统

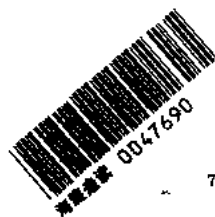
李冠英 张毅忠 郑存陆 编

广东科技出版社

下列版后之日期本館必須印出

限 100

微型电子计算机教程
APPLE II 微型计算机
BASIC语言与磁盘操作系统
寇冠英 张毅忠 郑存陆 编



广东科技出版社出版
广东省新华书店发行
广东新华印刷厂印刷
787×1092 毫米 32开本 11.875印张 223,000字
1984年11月第1版 1985年3月第2次印刷
印数 50,001—100,000 册
统一书号15182·107 定价1.60元

1008732

内 容 简 介

本书是《微型计算机教程》中的一种，供讲授微型机的操作使用方法时选用。内容包括APPLE II微型计算的上机操作方法；APPLESOFT BASIC语言程序设计；DOS3.3磁盘操作系统的使用；APPLE II机的汉字系统和绘图方法等。在本书的第十二章中给出了十七种常用计算方法的BASIC程序实例，供读者参考。此外，还给出简单管理程序的基本模式，可供读者编写工资管理、库存或合同管理等程序时借鉴。本书在取材及内容编排上尽可能结合教学和实际上机操作的特点，使初次接触微型计算机的读者容易阅读。

本书适合中等专业学校、中等技术学校及部分高级中学讲授微型计算机课时用作参考教材。也可供职工工业余大学，大专院校举办短训班时使用。

JS393/01

出版说明

《微型电子计算机教程》(以下简称教程)是根据目前中专、中技和职大的教学要求编写的。在编写过程中注意按教学和培训的需要来安排内容,力求概念简明易懂,突出使用和操作技巧,便于学习者自行复习,同时注意与实践相结合。《教程》中收入的许多程序实例都能在实践中直接运用,在学习中起举一反三的作用。

本《教程》首批安排出版的选题共有六种,以目前使用范围较广的 TRS—80、APPLE I (苹果 2 型)、R1 等机型为教学机,分别介绍各型微型机的 BASIC 语言的编程知识和上机操作方法、系统结构原理和用汇编语言编程方法等内容。这六种选题是:(1)《微型计算机基础》;(2)《TRS—80 微型计算机 BASIC I 语言与磁盘操作系统》;(3)《APPLE I 微型计算机 BASIC 语言与磁盘操作系统》;(4)《APPLE I 微型计算机系统结构》;(5)《APPLE I 微型计算机实习操作指导》;(6)《R1 微型计算机读本》。以后根据教学的需要将补充若干选题。其中,《微型计算机基础》是公共基础课教材,其余各书可由各教学单位根据所采用的教学机型号,结合课程设置和教学要求选用。

参加这套《教程》编写工作的作者是大学的教授、讲师及中专老师,他们曾为几十期培训班讲课,积累了微机教学的经验。《教程》中的六个选题就是在他们讲课的基础上经过认真修改补充后写成的。

本《教程》可供中等专业学校和中级技工学校、职工大学和业余大学开设微型计算机课时作参考教材使用,亦可供各企事业单位和大专院校开办微型机短期培训班时用作教材,还可供广大读者用作自学读本。

前 言

这本小书是为了培养微型计算机的操作使用人材而编写的。书中选择目前我国使用最多的APPLE II微型计算机作为学习机，主要介绍这种机器的操作方法；APPLESOFT BASIC语言的程序设计；磁盘操作系统、汉字系统等内容。本书在取材和内容安排上注意了循序渐进，由浅入深的教学特点，同时，又可帮助初学者解决一部分上机操作时遇到的实际问题。本书多数的章节后附有经过悉心选择和设计的习题，以帮助读者加深理解本书的内容。本书列举的一些应用实例是编者多年教学和实际应用的积累，可供读者结合具体问题编制程序时借鉴。在介绍基本语句时，我们都尽可能避开复杂的数学问题，这是为了使具有中等文化程度的读者也能读懂。

在编写本书的过程中，得到华南师范大学计算机科学研究室和计算机中心的同志的支持和帮助，在此一并表示感谢。

编 者

一九八四年四月

目 录

第一章 APPLE II 微型计算机的结构	(1)
第一节 微型计算机	(1)
第二节 有关程序设计的一些概念	(3)
第三节 键盘	(9)
第四节 荧光屏显示器	(9)
第五节 主机	(13)
第六节 常用的外部设备	(14)
第七节 外部设备控制卡	(15)
第二章 APPLE II 微型计算机的使用方法	(17)
第一节 开机和关机	(17)
第二节 键盘操作方法	(19)
第三节 APPLE II 微型计算机两种 BASIC 语言系统的 转换	(23)
习 题	(24)
第三章 APPLESOFT BASIC语言概述及其基本成分	(25)
第一节 怎样使用NEW、RUN 及 LIST 命令.....	(28)
第二节 基本符号、常量及变量	(32)
第三节 函数	(35)
第四节 表达式	(45)
习 题	(46)
第四章 程序的删改	(49)
第一节 删除程序	(49)
第二节 插入程序行	(51)
第三节 修改程序	(51)

习 题	(55)
第五章 赋值语句和输入、输出语句	(56)
第一节 输出语句——PRINT	(56)
第二节 赋值语句	(65)
第三节 键盘输入语句	(67)
第四节 立即输入语句——GET.....	(70)
第五节 置常量语句和读常量语句	(71)
第六节 恢复语句——RESTORE.....	(72)
第七节 荧光屏控制语句	(74)
第八节 CLEAR、REM、END和POKE语句	(76)
第九节 打印机输出	(79)
第十节 使用卡式录音机存取程序	(83)
习 题	(85)
第六章 转移、条件转移与循环语句	(89)
第一节 无条件转移语句——GOTO.....	(89)
第二节 停止程序执行命令——CTRL-C.....	(92)
第三节 程序暂停语句——STOP	(93)
第四节 逻辑运算、条件转向语句	(94)
第五节 选择转向语句——ON...GOTO.....	(105)
第六节 循环语句——FOR...NEXT	(109)
第七节 出错转移与处理后返回语句—— ONERR GOTO与RESUME	(119)
第八节 变换字符的黑白显示语句—— INVERSE与NORMAL.....	(121)
第九节 黑底白字和白底黑字交替闪烁语句 ——FLASH	(122)
第十节 控制输出字符速度的语句——SPEED	(123)
第十一节 显示执行的语句行号——TRACE	(124)
习 题	(124)

第七章 数组	(129)
第一节 数组的概念	(129)
第二节 预留存储空间语句——DIM	(135)
第三节 二维数组	(138)
习 题	(146)
第八章 子程序及其转子语句	(148)
第一节 转子程序语句——GOSUB	(149)
第二节 结束子程序并返回主程序语句 ——RETURN	(152)
第三节 POP语句——消掉一个子程序的返回位置	(153)
第四节 子程序嵌套	(155)
第五节 选择转子程序语句——ON...GOSUB	(158)
习 题	(159)
第九章 图形与声音	(161)
第一节 低分辨率图形	(161)
第二节 高分辨率图形	(171)
第三节 APPLE II 微型机的声音	(180)
习 题	(135)
第十章 磁盘操作系统(DOS)与文件(FILES)	(186)
第一节 文件的基本概念	(186)
第二节 文件名称	(187)
第三节 磁盘操作系统(DOS)	(189)
第四节 在程序中使用DOS命令	(199)
第五节 文件结构	(201)
第六节 顺序文件	(204)
第七节 随机文件	(226)
习 题	(241)
第十一章 汉字系统	(243)
第一节 汉字系统简介	(243)

第二节 C-PLUS汉字卡的操作	(245)
第三节 APPLESOFT BASIC 语言在汉字系统中 的执行	(248)
第四节 汉字卡的子程序和软开关	(248)
第十二章 几种常用计算方法的BASIC 程序	(270)
一、求最大公因数	(270)
二、求整数的质因数	(272)
三、求多边形面积	(274)
四、求三角形	(278)
五、两个向量的分析	(282)
六、用辛浦生公式求数值积分	(287)
七、用梯形法则求数值积分	(292)
八、用高斯法则求数值积分	(294)
九、用牛顿法求多项式的实根	(299)
十、求联立线性方程组的解	(308)
十一、矩阵的加法、减法和系数乘积	(307)
十二、矩阵乘法	(312)
十三、求逆矩阵	(316)
十四、求平均数、变异数和标准差	(320)
十五、求线性相关系数	(326)
十六、线性回归	(338)
十七、多重线性回归	(333)
附 录	(340)
附录一 美国国家标准码(ASCII CODE)	(340)
附录二 APPLESOFT的保留字及其代码	(343)
附录三 APPLESOFT 摘要	(346)
附录四 错误信息	(354)
附录五 有特殊用途的PEEK和POKE工作单元	(360)
附录六 RENUMBER程序的功能及使用	(365)
附录七 APPLESOFT零页使用情况	(367)

第一章 APPLE II 微型计算机的结构

第一节 微型计算机

自从1946年美国宾夕法尼亚大学的莫希莱(John Mauchly)和埃克特(J. Presper Eckert)等人设计成第一台数字电子计算机以来,电子计算机经历了电子管、晶体管、小规模集成电路和大规模集成电路四代。从第一代到第四代,计算机的功能不断地增强,体积显著地缩小,价格也大幅度地下降。使计算机在各个领域得到广泛的应用。由此,又反过来促使集成电路的不断发展,而微型计算机就是一种应用大规模集成电路的产物。一九七一年,英国英特尔(INTEL)公司制成一种INTEL 4004的微处理器,这种微处理器包括了能同时对四位二进制数进行运算和控制处理的功能,称为四位微处理器。一九七四年以后,英特尔、莫托洛拉(MOTOROLA)等公司又研制成 INTEL 8080、M6800 等八位微处理器和INTEL 8086、M68000 等十六位微处理器。目前,市场出售的微处理器种类繁多,各有各的特点。人们用这种用微处理器作为计算机的中心处理单元(又称为中央处理器,简称CPU)装制成的电子计算机,称为微型计算机。APPLE II(苹果 II 型)微型计算机就是其中的一种。

APPLE II 微型计算机采用积木式结构,其基本部分包括键盘(输入设备),荧光屏显示器(输出设备)和主机。此外,还可以根据使用的需要增加盒式录音机、磁盘机、打印机、绘图板等外部设备,其外形如图1-1所示。

从整机系统来说,还有一种叫APPLE II PLUS 的计

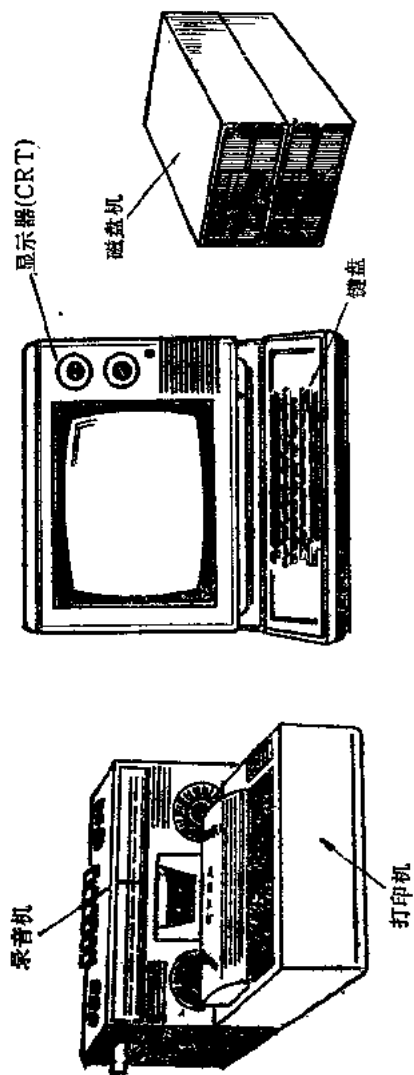


图1-1 APPLE II 微型计算机外型

算机，它与APPLE I不同之处在于使用了不同的 BASIC 语言解释程序，APPLE II 使用的是整数BASIC解释程序，而APPLE II PLUS使用的是APPLESOFT BASIC（实数 BASIC）解释程序，目前多数人都采用 APPLESOFT BASIC，故本书介绍的APPLE II 微型机是使用APPLESOFT BASIC解释程序的。

第二节 有关程序设计的一些概念

一、字节、字长

（一）二进制位(bit)

bit是binary digit的缩写，它是计算机度量信息的单位，也是存储资料的最小单位，一个二进制位能存贮一位二进制数“0”或“1”。

（二）字节(byte)

字节是用作处理一串二进制数的单位，通常是以八位二进制数组成一个字节。因为八位二进制数可以组成00000000，00000001，……，11111111等256 ($2^8 = 256$)种不同的状态，它足以代表计算机所使用的数字（0，1，2，3，…，9）、字母（A，B，C，…，Z）和运算符号（+、-、…）等。

（三）字长（word length）

在电子计算机中，字是计算机处理信息的单位。通常，一个字由若干个字节组成。一个字所包含的二进制数的个数，叫做字长，各种计算机的字长可以相同，也可以不同。如IBM870的字长是82位，即由四个字节组成。国产100系列计算机的字长为16位，由二个字节组成。APPLE II 微型计算机的字长是8位，即由一个字节组成。一般来说，字长较

大的计算机处理能力较强，因此，计算机的字长常常被用来说明计算机功能的标志之一。

二、地址、指令

在计算机中，地址用来标识寄存器、存储器或外部设备的编号和名字。在指令中，它规定操作数所在的位置。对于存储单元的地址编号，一般采用八进制数或十六进制数。例如APPLE I 微型机的地址分配是（用十六进制）：

0000~BFFF 随机存储器RAM 48K字节

C000~CFFF 外部设备码 4 K字节

D000~FFFF 固定存储器ROM 12K字节

注：为了避免与英文字母中的“O”相混淆，计算机通常把数字“0”写成“ ϕ ”，本书由于排版上的原因，对于数字“0”仍用0表示。

所谓指令是指要计算机执行操作的命令。电子计算机进行各种操作，是按不同的指令进行的，不同的计算机具有不同的指令系统。每一条指令都是由一定位数的二进数码组合而成，分成操作码和地址码二部分，即

操 作 码	地 址 码
-------	-------

一般，操作码放在指令前部，地址码在指令的后部。每条指令只有一个操作码，但可以有几个地址码。有二个地址码的指令叫二地址指令；有三个地址码的指令叫三地址指令。以后结合具体问题的讨论，我们可以进一步了解指令的意义。

三、程序设计语言

人们使用计算机，首先要把想做的事，用人和机器都能识别的语言编出程序，计算机执行了这个程序，才能完成预

定的目的。程序设计语言是指人们用来编制计算机程序的语言，目前，使用的程序设计语言很多，但从构成来说大致可分为机器语言、汇编语言和高级语言三类。

（一）机器语言

机器语言是直接使用机器的指令作为程序设计的语言，机器的指令是由计算机的设计者制定的。一部计算机的所有指令就组成这部机器的指令系统，各种计算机有不同的指令系统。例如，如果选定二进制码

0000 代表加的操作码

0001 代表减的操作码

0010 代表乘的操作码

0011 代表除的操作码

则计算机遇到操作码为0000时就做加法，遇到操作码为0001就做减法……。这种依赖于计算机本身的指令系统的语言，就称为机器语言。假如用二进制码表示的操作码和操作数是

“+”的操作码 0000

B 单元的地址为 0100

C 单元的地址为 0101

则将 B 单元的数据加上 C 单元的数据且存入 B 单元的指令格式为

操 作 码	第一地址	第二地址
-------	------	------

机器指令可写成：

0000 0100 0101

计算机遇到这样的一串二进制码，就知道要进行 $B = B + C$ 的操作，即把存在地址为 0100 内的数据与存在地址为 0101 内的数据相加，再把结果存入 0100 内。

由此可见，用机器语言编写程序不但极其繁琐，且容易发生错误。不同的机器所使用的机器语言也不同，因此在使用上非常不便。

（二）汇编语言

汇编语言不是用一串二进制数代表指令，而是用英文字母及符号代表指令。例如假定表示上述 $B = B + C$ 操作的一串二进制数码

0000 0100 0101

汇编语言可写为： `ADD B, C`

其意义就是将 B 单元的数据加 C 单元的数据，结果存入 B 单元中。这样，使用起来就会方便得多。

（三）高级语言

高级语言比较接近人们的阅读习惯，例如计算 $A = B + C + D + E$ ，在 BASIC 语言（一种高级语言）中，可以使用如下的语句：

$A = B + C + D + E$

因此，高级语言较易为人们接受，而且通用性较强，即不同的机器，可以使用相同的高级语言。目前常用的高级语言有 BASIC、COBOL、FORTRAN 和 ALGOL 等。

BASIC 语言，称为初学者通用的符号指令代码语言，它的特点是简单、易学和使用方便，并可进行人机对话。目前大部分微型计算机都可以使用这种语言。

COBOL 语言，即面向商业的公共语言，它是为商业数据处理而设计的计算机语言，主要用于数据处理。

FORTRAN 语言及 ALGOL 语言，它们都适用于描述数值计算过程，是在科学、工程技术设计上，使用最为广泛的两种计算机语言。

四、流程图

流程图是解决问题的步骤的图形表示，它借助各种不同的图形和箭头来表示解决问题的顺序，每一种图形代表一种作业功能，而箭头则为流程的方向，它可帮助我们设计各种较为复杂的程序，也有助于程序的阅读和检查。根据我国 GB 1526—79 标准规定的信息处理流程图的图形符号和使用规则，常用的图形符号如下。

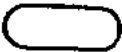
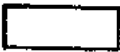
图形	功能
	起始或终止
	一般处理
	输入或输出
	由键盘输入
	打印输出
	测试判断
	调用子程序
	连接
	流程方向

图1-2 常用的图形符号