



微电脑Apple II 使用方法与技巧

林卓然 莫日华 编著
中山大学出版社



APPLE II 微电脑

使用方法与技巧

(修订本)

林卓然 莫日华 编

中山大学出版社

内 容 简 介

本书从使用的角度出发,分析了APPLE II微电脑的有关系统软、硬件结构,提出了进一步开发使用该机的方法及技巧。

书中按专题形式编写,主要内容包括:系统的建立,BASIC程序、变量及磁盘文件的存储格式,BASIC程序的优化,作图及音乐程序的编制方法,OS 9的使用技巧,各种系统子程序的调用,I/O接口技术,常用加密法,汇编语言程序设计及操作方法等。本书既有一定的原理分析,又有大量的用机经验体会及实用例子。

本书可供具有一定用机实践的广大用户以及科技人员、大中专校师生阅读。

APPLE II 微电脑使用方法与技巧(修订本)

林卓然 莫日华 编

中山大学出版社出版发行

广东省新华书店经销

广州红旗印刷厂印刷

787×1092毫米 32开本 16.25印张 327千字

1990年11月第1版 (1990年11月至)次印刷

印数: 1—15000册

ISBN 7-306-00225-2

定价: 4.50元

前 言

APPLE II 微型计算机是目前国内外流行的一种微型计算机。为了更好地发挥该机的作用及提高使用效率,我们编写了这本书,供具有微型计算机基础知识及操作能力的该机用户阅读。

本书的资料主要来自作者多年的用机实践,同时吸收了国内外的先进经验。书中内容比较广泛,取材新颖,其中相当一部分是在现有资料中查不到的。

本书从 1986 年 10 月初版问世以来,受到广大读者的热情鼓励和大力支持。此次修订,精简了一部分内容,同时补充了编程技巧、调试技术、I/O 接口技术、汇编语言程序设计等内容。

本书初版由林卓然、莫日华共同编写,并由林卓然修订。

由于编者水平有限,书中一定有不少缺点、错误,敬请读者批评指正。

编者

1988 年 11 月于中山大学

目 录

第一章 系统的建立	(1)
§ 1.1 内存的分配	(1)
§ 1.2 RESET 过程	(3)
§ 1.3 建立 BASIC 系统的区域指针	(7)
第二章 BASIC 源程序及变量的存贮格式	(10)
§ 2.1 源程序的存贮格式	(10)
§ 2.2 简单变量的存贮格式	(13)
§ 2.3 数组的存贮格式	(16)
§ 2.4 字符串空间	(17)
第三章 BASIC 程序的优化	(19)
1. 提高运算速度	(19)
2. 节省存贮空间	(23)
3. 编写清晰易读的程序	(25)
第四章 程序设计技巧	(27)
§ 4.1 模块化程序设计	(27)
§ 4.2 菜单技术	(29)
§ 4.3 INT 函数的应用	(30)
§ 4.4 控制循环的方法	(32)
§ 4.5 数的一些处理方法	(35)
1. 减少数值的误差	(35)
2. 节省数据的存贮空间	(39)

3. 提高计算精度.....	(40)
§ 4.6 字符串函数的一种应用技巧.....	(43)
§ 4.7 出如移位方法.....	(44)
§ 4.8 一种好的统计方法.....	(46)
§ 4.9 汉字系统的子程序和软开关.....	(47)
第五章 程序测试.....	(51)
§ 5.1 几种常用技术.....	(51)
1. 暂停技术.....	(51)
2. 键盘运算技术.....	(51)
3. 执行过程的自动跟踪.....	(53)
4. 延迟技术.....	(54)
5. 变量的跟踪检查.....	(54)
6. 数据测试技术.....	(55)
§ 5.2 排除几种常见的程序错误及隐患.....	(56)
第六章 BASIC 操作技巧.....	(63)
§ 6.1 重编 BASIC 程序的行号.....	(63)
§ 6.2 BASIC 程序的合并.....	(64)
§ 6.3 BASIC 程序的动态链接.....	(67)
§ 6.4 把 BASIC 程序编译成目标程序.....	(69)
§ 6.5 分页打印程序清单.....	(71)
§ 6.6 检查内存及磁盘可用空间的容量.....	(72)
第七章 打印输出的技巧.....	(74)
§ 7.1 输出结果的对齐处理.....	(74)
§ 7.2 打印机的输出控制.....	(77)
§ 7.3 打印显示屏上的图形.....	(80)
§ 7.4 汉字输出.....	(86)
第八章 图形的绘制.....	(93)
§ 8.1 绘图程序例.....	(93)

§ 8.2	屏幕软开关	(103)
§ 8.3	提高高分辨率的解析度	(108)
§ 8.4	制作高分辨率造型图	(111)
§ 8.5	图形信息的存取	(121)
第九章	编制音乐程序	(123)
§ 9.1	发音原理	(123)
§ 9.2	音乐程序例	(124)
1.	演奏“国歌”	(124)
2.	演奏《西游记》主题歌	(128)
3.	电子琴	(131)
第十章	机器语言子程序的使用	(135)
§ 10.1	建立机器语言子程序	(135)
§ 10.2	调用机器语言子程序	(139)
§ 10.3	调用 BASIC 程序变量	(146)
第十一章	DOS 简介	(150)
§ 11.1	磁道和扇区的划分	(150)
§ 11.2	存取过程简述	(152)
§ 11.3	DOS 的引导过程	(153)
§ 11.4	目录磁道的存贮格式	(154)
§ 11.5	各种类型文件的存贮格式	(163)
§ 11.6	使用 RWTS 子程序	(167)
1.	把数据写入指定的扇区	(169)
2.	显示指定扇区的数据	(171)
3.	磁道/扇区复制程序	(173)
第十二章	DOS 的使用方法及技巧	(176)
§ 12.1	复制磁盘文件的方法	(176)
§ 12.2	提高 DOS 装入程序的速度	(179)

§ 12.3	扩充磁盘的容量.....	(181)
1.	扩充磁道	(181)
2.	复制有 40 条磁道的磁盘.....	(182)
3.	制作无 DOS 的数据磁盘	(183)
§ 12.4	DOS 的修改	(184)
1.	修改 DOS 命令名	(184)
2.	用 CATALOG 命令自动印出自由扇区的数目 ...	(185)
3.	直接删除已加锁的文件	(187)
4.	使用缩写的文件名	(187)
5.	代替 CTRL-D 控制码.....	(188)
6.	使用 BRUN 或 EXEC 来执行 HELLO	(188)
7.	在 DOS 和 DOS 缓冲区之间建立程序区	(189)
§ 12.5	几种常见故障的排除方法.....	(190)
1.	DOS 系统不能正常工作怎么办	(190)
2.	消除掉的文件如何恢复	(191)
3.	盘中的 DOS 坏了怎么办	(192)
第十三章	在机器语言中调用系统子程序的方法.....	(193)
§ 13.1	常用的监控子程序.....	(193)
§ 13.2	调用 BASIC 运算子程序	(202)
§ 13.3	调用 DOS 系统命令子程序	(207)
§ 13.4	直接调用 DOS 命令的一种方法	(212)
第十四章	I/O 接口技术	(213)
§ 14.1	I/O 接口	(213)
1.	I/O 接口信号	(213)
2.	应用举例——控制电灯的亮熄	(213)
§ 14.2	一种实用的 I/O 接口卡.....	(225)
1.	接口卡电路原理	(225)
2.	接口卡功能调试	(229)
3.	应用举例	(230)

方波发生器	(232)
顺序控制器	(234)
§ 14.3 中断处理	(238)
1. 6502 的中断种类	(238)
2. 可屏蔽中断传送的一般程序结构	(240)
3. 应用举例	(241)
电子钟	(241)
测量频率	(246)
§ 14.4 12 位 A/D——D/A 卡简介	(250)
第十五章 常用加密法	(254)
§ 15.1 使用 LOCK-IT-UP 加密工具	(254)
§ 15.2 磁盘加密法	(258)
§ 15.3 其他加密措施	(260)
第十六章 汇编语言程序设计简介	(263)
§ 16.1 6502 内部寄存器	(263)
§ 16.2 指令的分类	(266)
§ 16.3 6502 的寻址方式	(270)
§ 16.4 汇编语言的一些规定	(275)
1. 汇编语言的格式	(275)
2. 常用伪指令	(275)
3. 常数	(277)
4. 表达式	(277)
§ 16.5 程序设计举例	(278)
第十七章 其他有关操作	(285)
§ 17.1 监控程序命令	(285)
§ 17.2 小汇编程序	(290)
§ 17.3 编辑/汇编程序	(291)
附录	(298)

一、ASCII 码表	(298)
二、APPLESOFT 各专用词的代码及解释执行入口地址	(301)
三、显示器地址分配图	(304)
四、一些零页地址的使用情况... ..	(305)
五、DOS 的特殊地址(\$ 3D0— \$ 3FF)	(307)
六、6502 指令系统	(308)

第一章 系统的建立

§ 1.1 内存的分配

APPLE II 微电脑的寻址范围为 64K，其地址分配为如下三大部分（以下 \$ 表示十六进制数）：

\$ 0000 ~ \$ BFFF——随机存储器 RAM（48K 字节）。

\$ C000 ~ \$ CFFF——外部设备码（4K 字节）。

\$ D000 ~ \$ FFFF——只读存储器 ROM（12K 字节）。

APPLE II 微电脑中 ROM 主要用于存放系统监控程序和 BASIC 解释程序。RAM 主要用于暂存由外存储器（如磁盘）调入的操作系统、各种高级语言的源程序、变量、中间结果及图象信息等等。RAM 空间的分配如下：

\$ 0000 ~ \$ 00FF——即 0 页，为系统所用，主要用于设置一些指针及一些工作单元。

\$ 0100 ~ \$ 01FF——即 1 页，为系统堆栈区。

\$ 0200 ~ \$ 02FF——即 2 页，为键盘输入缓冲区。

\$ 0300 ~ \$ 03FF——即 3 页，该页区除了高地址约 30 个字节被 DOS 系统占用外，多数可以由使用者自由使用。这个区，非经用户指定，系统是不会访问它们的，所以较安全。对于不太长的机器语言程序，可以存放在本页区。

\$ 0400 ~ \$ 07FF——为文本和低分辨率图形第一页的显示存储区。其地址分配见附录三。

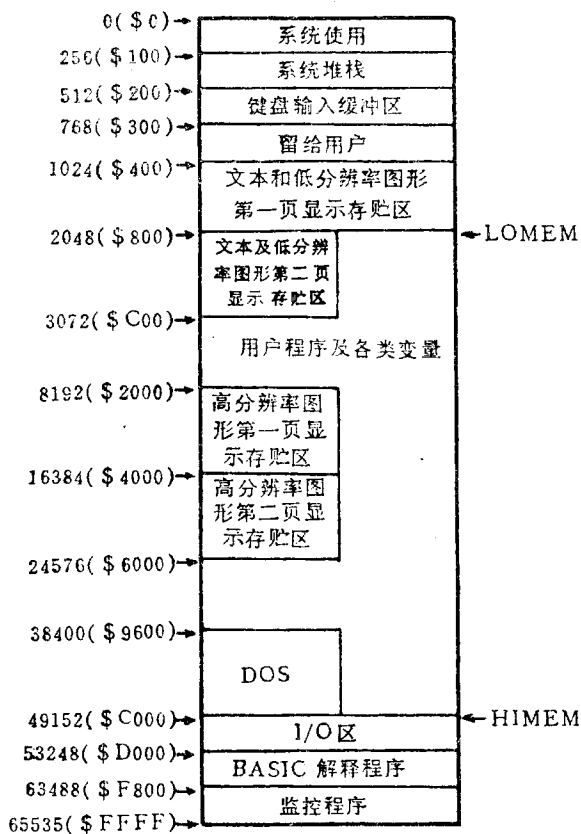


图 1.1 APPLE II 内存的分配

从 \$0800 开始至最高可用地址为用户区，其中还划出 \$2000~\$3FFF 和 \$4000~\$5FFF 两个区域分别为高分辨率图形的第一页显示存贮区和第二页显示存贮区。

内存的分配情况，见图 1.1 所示。

§ 1.2 RESET 过程

当用户开启 APPLE II 微电脑电源时，由于机内线路的作用，就给 6502 微处理器产生一个被称为 $\overline{RES}$ 的复位信号。这一复位信号自动将 \$FFFC 单元中的内容送入程序计数器低位 PC_L ，将 \$FFFD 单元中的内容送入程序计数器高位 PC_H 。因为 \$FFFC 和 \$FFFD 两单元的内容是固化的（处于监控程序中），一接通电源便可读出。这两单元的内容分别是 \$62 和 \$FA，于是程序计数器最先得到的地址是 \$FA62。这样机器就从 \$FA62 开始执行监控程序，我们把这段程序称为开工程序，或称为 RESET 程序。

$\overline{RES}$ 复位信号也可由 RESET 键产生。不论在何种语言状态下工作，当用户按下 CTRL-RESET 时，计算机就会中止当前的程序而执行与上述接通电源后同样的过程。我们把这两种执行同样操作的启动过程称为“重新启动过程”或“RESET”过程。

RESET 过程的任务是对 APPLE II 微电脑系统初始化，它包括设置屏幕显示器及键盘为系统的输出输入设备，置屏幕为 24 行 40 列的全屏幕正显示方式（黑底白字），关闭键盘选通，驱动扬声器以 1 千赫频率发出持续时间为 0.1 的响声等等。

RESET 过程的粗框图，如图 1.2 所示。

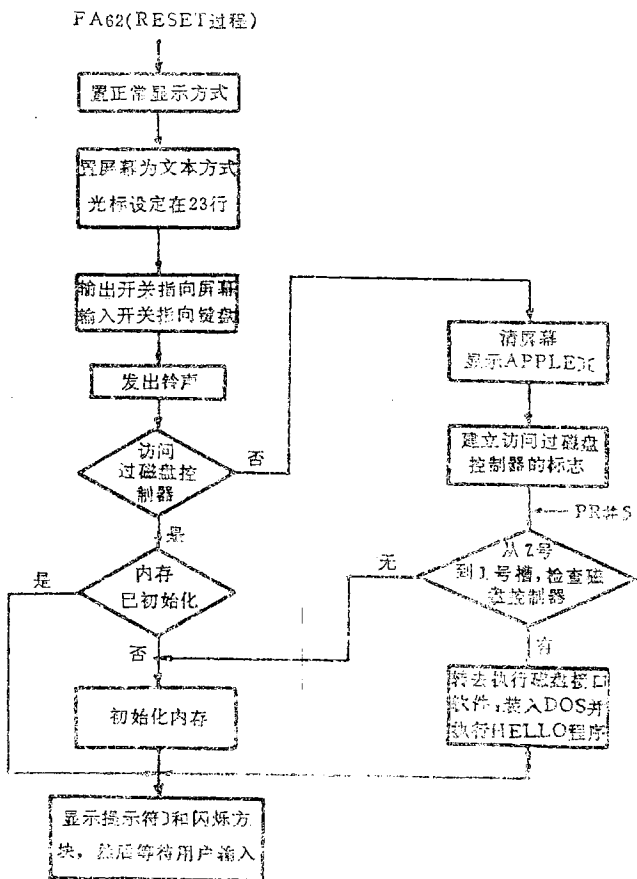


图 1.2 RESET 过程

从框图 1.2 中, 我们不难看出 RESET 过程有以下几个

特点：

1. 如果系统未配置磁盘驱动器，那么从 7 号到 1 号槽口上将找不到磁盘控制器，此时只能建立无 DOS 的 BASIC 系统。

2. 如果系统配置有磁盘驱动器，RESET 过程将会把 DOS 装入内存，并运行磁盘上的“欢迎”程序 (HELLO)。此后，内存中的程序就可以和磁盘交换信息了。

3. 如果虽有磁盘驱动器，但却没有把磁盘插入，驱动器就会没完没了地转下去，这时按下 CTRL-RESET 就可以使驱动器停止工作，并建立了无 DOS 的 BASIC 状态。

用户调试程序中往往会提出这样的要求：当程序调试过程中遇到麻烦时，希望能通过人机对话，例如按下 CTRL-RESET 时，使计算机系统回到初始化状态，但同时要求计算机不要破坏原来使用的语言状态，并保存原先使用的程序及变量值。显然，合上电源后所执行的启动过程是无法满足这种要求的。为此 RESET 过程对“合上电源”及“按 CTRL-RESET 键”两种启动过程进行了区别。通常把合上电源后所执行的 RESET 过程称为“冷启动”，而将程序执行过程中，由操作者按下 CTRL-RESET 键所引起的 RESET 过程称为“暖启动”。

从硬件上看，冷启动和暖启动都是通过产生 $\overline{RES}$ 信号，并由 CPU6502 的第 40 脚来触发的，而线路上是不能判别属何种启动的。APPLE II 采用软件判断方法来区分冷启动和暖启动。RESET 程序规定了三个 0 页单元来分别存放启动入口地址及启动标志，入口地址用 \$3F2 和 \$3F3 两单元存放，启动标志用 \$3F4 单元存放。上述三个单元属 RAM 区，当开启电源时，若不对它们置数，其内容是随机的，如果此时取 \$3F3

单元的内容与立即数 \$A5 进行“异或”操作，其结果等于 \$3F4 单元的内容的概率几乎为 0。这两步操作可以验证程序是否曾对上述两单元置过约定数，也是区分冷启动和暖启动的条件。若判定 \$3F3 和 \$3F4 两单元存的是刚开机时所产生的乱数，则进行冷启动。冷启动过程中会把 BASIC 的起始地址（通常是 \$E003）存入单元 \$3F2 和 \$3F3 中，并算出 \$3F3 的内容与 # \$A5 的异或值（即 # \$45）和存入 \$3F4 单元中。此后凡遇到 RESET 过程，因为 \$3F4 单元的内容已是合适的数值，就不再执行冷启动而代之以暖启动了。

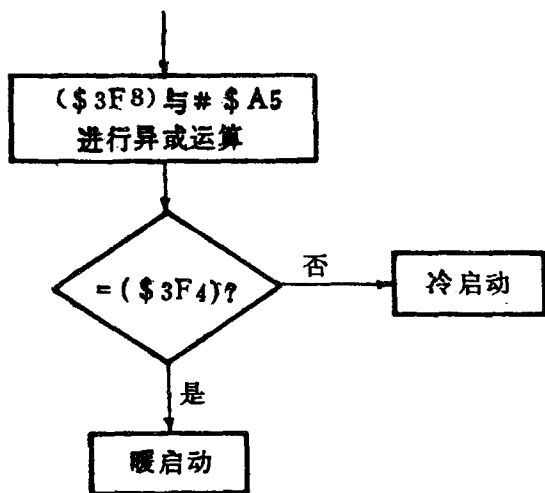


图 1.3

图 1.3 表示了区分冷启动和暖启动的流程。

§ 1.3 建立 BASIC 系统的区域指针

为了在执行程序中存贮各种信息，BASIC 系统在用户区里划分出如图 1.4 所示的几个区域。

其中，简单变量区紧跟在源程序区之后，数组区又紧跟在简单变量区之后，它们都随着上面区域的变动而变化。例如，当补充程序行时，变量区和数组区会自动下移（实际上是区域指针变化，而区域里的内容是不会移动的），同样，当取消程序行时，变量区和数组区会自动上移。

BASIC 系统将不同的信息分块存贮后，就必须记住每一区域的首地址，这样才能访问它们，这些首地址都存放在零页固定单元里，每两个单元存放一个区域的首地址。这些单元也称为区域的指针。例如，程序区首址指针为（\$67，\$68），它表示当程序区首址为 \$801 时，\$68 单元存放 \$801 的高 8 位 \$08，而 \$67 单元存放低 8 位 \$01。表 1.1 列出了 BASIC 各区域指针及其初始值（假设系统配有 48kRAM）。

HIMEM 表示用户可用的最高地址（实际上比它小 1）。在 DOS（本书指的是 DOS3.3）状态下，HIMEM 值是由 DOS 确定的，下面列出在各种 RAM 容量不同时，DOS 引导时的 HIMEM 值。

在无 DOS 的 BASIC 状态下，HIMEM 值是通过检测内存得到的。检测方法如下：从 \$800 单元开始检测，先从中读出内容，全字节取“反”（0 变 1，1 变 0）后，再存入该单元，然后读出比较。如果该单元原来的数和现在的数不同，说明它是 RAM 的一部分，再如此检测下一页（即以 \$900 开始的 \$100 字节），每页只检测页首一个字节。这样一页一页地检