

APPLE II 资料(二)

参 考 手 册

监 控 程 序

目 录

Apple语言系统安装和操作手册

语言系统安装在Apple机之前, 请读这本手册。不正确的安装进使系统和Apple机引起永久性毁坏。

引 言	(1)
第一章 系统要求	(1)
第二章 安装	(1)
一、怎样安装语言插件	(1)
二、怎样安装新的PROMS	(2)
第三章 使用 BASIC	(3)
一、启动 BASIC	(3)
二、从 BASIC 转换	(4)
第四章 自启动 ROM	(4)
一、自启动特点	(4)
二、复位键 (RESET)	(4)
三、BASIC 编辑特点	(4)
停止列表特点	(8)
附录 A	
自启动ROM BASIC特点摘要	(9)
附录 B	
自启动 ROM 的技术情况	(9)
加到自启动 ROM 里的子程序	(9)
自启动 ROM 的限制	(10)
附录 C	
语言插件的存贮器分布	(10)
附录 D	
语言插件控制码	(11)
附录 E	

MX—80— I 打印机接口用户手册

一、前言	(15)
二、组成	(15)
三、功能	(15)
四、安装	(15)
五、文本打印	(16)

六、位图象打印.....	(19)
七、高分辨率图形的硬拷贝.....	(25)
八、打印格式程序.....	(28)
九、其它应用.....	(30)
十、接口板的电路原理.....	(33)
十一、几点说明.....	(33)
十二、连接器信号表.....	(34)
附录 1 软开关表.....	(35)
附录 2 控制字符表.....	(36)

引 言

Apple语言系统大大地增强了Apple I型或Apple I plus计算机的功能。它使Apple I型计算机有64K字节的随机存取存储器(RAM)并可直接寻址。

Apple I和Apple I plus配有语言系统后,可使用Integer BASIC, Applesoft BASIC以及其它语言。

此外,语言插件上有2K自启动RAM(只读存储器),它便于BASIC程序的屏幕编辑,以及以接通Apple主机电源后,启动软盘驱动器,使Apple机运行一个特定程序。

语言系统还给软盘控制器加了两新的可编程的只读存储器(PROM)集成电路。新加的PROM安装在盘控制器插件上以后,就可以使用13扇区和16扇区的软磁盘。操作系统DOS3.2用13扇区软盘, DOS3.3用16扇区软盘。

语言系统包括“BASIC”软盘,可通过BASIC盘引导13扇区Integer BASIC和Applesoft BASIC以及13扇区的DOS。同样,在Apple I plus系统上可调用13扇区软盘的软和件数据。

本手册说明了怎样安装语言系统以及使用13扇区的Integer BASIC、Applesoft BASIC以及DOS。要运用其它语言或操作系统,则另有手册介绍。

第一章 系统要求

语言系统仅限于用在已有48K随机存储器(RAM)的Apple I或Apple I plus系统上。如果在Apple机上不足48KRAM,则必须扩充RAM达48K。

要使用语言系统,至少需要一个软盘驱动器和一个控制器插件,语言系统可配3个控制器插件,每个控制器插件带动一个或两个软盘驱动器。

第二章 安 装

一、怎样安装语言插件

安装语言插件是容易的,重要的是怎样正确安装,如安装不妥会损坏语言插件和Apple的其余部分,在工作之前一定首先切断电源。

要这样来打开Apple主机的盖子。向上拉主机盖子的后沿,直到拉开两面的扣闩。不要将盖子拉得很高,向后滑动盖子(从键盘方向向后看)。在拿掉或插入其它东西之前,请看下

面说明。

面对键盘,看打开盖子的Apple内部。在主板上,键盘后面的右边可以看到一个大约4英寸的白线方块,方块内为Apple机存贮器,24片RAM集成电路,每个集成电路片子上都标有很多数字。例如:MK4116, MCM4116, UPD416或HM4716。在板上标有‘RAM’,每个集成电路的一端有一个凹口朝着键盘。语言插件的小插座将替代白线方块内左后脚的集成电路E₃。

绕到线路板的后面,可以看到一行8个狭长的连接槽口,内部是镀金簧片。在槽口末端的线路板上面有槽口编号。语言插件插入0号槽口,即最左边的一个槽口。(该机不能同时使用Applesoft固定插件和语言系统)。带有语言系统的Apple所用的Applesoft与固件的Applesoft相同。

如果0号槽口内有一块插件,则抓住插件的两端上下轻轻地摆动,可将插件拔出。

在主板上找到包含RAM集成电路的白线方块内,用集成电路拔出器,把E₃集成电路取出。

现在准备安装语言插件。首先进行检查,然后将插件上扁平电缆一端的插头插入白线方块内E₃位置上,再将语言插件轻轻地插在主板0号槽口上,向下压,直至插牢。安装完则可以继续下一步,安装新的PROMS。

二、怎样安装新的PROMS

语言系统需要用新的盘控制器PROMS。这样,可使盘驱动器使用16扇区和13扇区软盘。

16扇区软盘可以包含与13扇区软盘相同的数据。但在16扇区软盘上,数据格式与13扇区软盘格式不同。因此,在16扇区软盘上能存贮较多的数据。

新的PROMS代换每个盘控制器插件上的两片PROMS,查看主板后面的插了插件的插座,可找到一个或一个以上的插座标有“盘I接口插件”。切断电源,轻轻地拔出每个插件。

接口插件拿在手上,有集成电路的一面朝着自己。在插件上有四行集成电路,每行有二片集成电路。第二行左边的PROM上标有‘PROMP6’。如果PROM本身标有‘P6’或341—0010—××,这就是要用新的PROMP6A代替的集成电路。如果这片PROM标有“P6A”或“341—0028—××”,那只是已更换过的PROMP6A。

现在,找到最底下一行左边的集成电路,在板上标有“PROMP5”。如果PROM本身标有“P5”或“341—0009—××”,这就是要新的PROMP5A代替的集成电路,如果这片PROM标有“P5A”或“341—0027—××”,那已是新的PROMP5A。

注:如果在盘控制器插件上已经安装了新的PROMSP5A和P6A,那么,请跳过PROMS的安装说明这一节,继续看下一节,怎样将插件插回Apple机。

有两个以上的驱动器时,每两个驱动器需要一个控制器插件,每个插件需要两个新的PROM,其安装同上述。

现在,使每个盘控制器连接到控制器插件上,再将插件插入约定槽口,其次序在下面的表中,要从槽口6向下顺序安装。先连每个控制器的驱动器1插头,再连驱动器2的插头。这样可使驱动器处在最有效的位置,并能增加系统可靠性。程序就不会试图使用不存在的驱动器,或者忽略那些存在的驱动器。

盘 驱 动 器 的 分 布

安 装 次 序	槽 号	驱动器号
第一驱动器	6	1
第二驱动器	6	2
第三驱动器	5	1
第四驱动器	5	2
第五驱动器	4	1
第六驱动器	4	2

给每个驱动器适当地标上“驱动器 1”或“驱动器 2”。也可以用槽号给驱动器作标记。在连续操作系统时，可以根据标记找到所调用的驱动器。

Apple语言系统上，现配有一个控制器插件，通常插在槽号 6 上。每一样东西已正确安装后，可以盖上主机的盖子，继续下一步。

第三章 使用BASIC

要使用语言系统，必须用一个适当的软盘启动Apple机，这个软盘作为语言系统的一部分，将引导我们使用13扇区Integer BASIC, Applesoft BASIC和DOS，以及Pascal。要使用13扇区BASIC或DOS，就要使用BASIC盘。要使用Pascal语言或其它的语言和操作系统时，需要查阅Apple系统的参考手册。

在开始进行程序以前，让我们确定使用同一术语，统一编排。软盘驱动器 1 应插入软盘控制器插件的驱动器 1 插座，软盘驱动器 2 应插入软盘控制器插件的驱动器 2 插座，控制器插件应插入槽号 6。

Apple 机有两种：Apple I 和 Apple I plus。Apple I 机主板 ROM 里有 Integer BASIC。在Apple I plus计算机的主板上的ROM有 Applesoft BASIC，如果没有盘驱动器或附件的话，加电后，ROM里的语言就是 Apple 机要调用的语言。在这部分讨论中，假定使用13扇区的BASIC和DOS3.2.1。

一、启动BASIC

1. 既使用Integer BASIC，又使用Applesoft BASIC，请放BASIC盘到驱动器里，然后接通电源约 5 秒钟后，屏幕显示，放BASIC盘，并且按return键。此时，主板ROM里有一个BASIC程序，在语言插件的RAM里有另一个BASIC程序。

2. 放在任一13扇区的DOS/BASIC软盘，并且按return键，Apple机将象DOS手册中描述的那样运行。除了这一操作方法以外，还有一种引导方法。主机加电后，BASIC盘在驱动器中，打入PR、#6、IN #6或打入6CTRL—P，当屏幕出现提示符时，再放入你想调用的DOS/BASIC盘。如果系统处于BASIC之中，且又打入了PR #6或IN #6，那就必须放入BASIC盘，然后放入要调用的盘。

在Apple系统中，可反复调用DOS，而不必切断电源。DOS运行后，通过使用INT或FP

命令可以从一种BASIC转到另一种BASIC。从屏幕上的提示符可知是处于那种BASIC。提示符>为Integer BASIC,]为Applesoft BASIC。

二、从BASIC转换

从13扇区的BASIC转换到语言系统的另一种语言之前, 首先把现行工作程序贮存起来, 因为系统调用另一种语言时, 现行程序进会被擦除。调用其它语言或操作系统的方式, 在介绍具体的语言或系统部分中说明。

第四章 自启动ROM

在 Apple 机的语言系统里, 附加了16K可写保护的RAM、2K 自启动 ROM。自启动 ROM是语言插件上最大的一片集成电路。它自动接替主板上ROM里的监控程序即主板上一个标有“ROM—F8”的插座, 这个自启动ROM, 控制着Apple 系统中最基本的功能, 机器语言程序和高级语言都利用了这些特点。

一、自启动

顾名思义, 自启动ROM自启动Apple 机, 其作用象一个总控制键一样, 也就是说, 当主机接通电源后, 立即执行一个专用程序。这样, 不需要用户有专门的知识, 就可以操作机器, 执行程序。

接通电源后, 引导槽号6, 驱动器1里的软盘BASIC, 或其它地区的引导盘。如果是13扇区DOS/BASIC盘, 应先运行BASIC盘, 接着就可以调用DOS/BASIC 盘。(另外, 可从BASIC盘上加载Applesoft BASIC)

通常, 按上述方法做, 系统会运行起来。

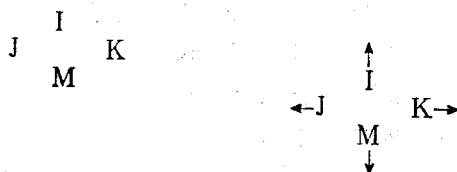
二、RESET键

去电以后, 再加电, 会启动Apple系统, 主机去电后, 会清除存贮器中的程序。在BASIC中按 RESET 键, 不管主机正在运行什么程序, 它都会停机显示所用的 BASIC提示符, 不丢失存贮器里的程序。

在BASIC中, 采用自启动控制时, RESET 键有类似于 CTRL—C 的功能。RESET 和CTRL—C两者都可停止程序的运行, 且可重新运行程序。另外RESET能使计算机恢复到程序首次运行前的状态。CTRL—C不能使存贮器中的数学恢复成他们的初始值。因此, 再次运行程序时, 结果可能不同。

三、BASIC—编辑特点

Apple 机语言系统的自启动 ROM, 可以使我们容易地编辑 BASIC 程序。采用自启动 ROM 的控制方式, 在屏幕上移动光标非常简单方便。请看键盘上的 I、J、K、M 键。它们的排列是这样的:



假设用四个箭头标在上面:

要按箭头法向移动光标，首先按下，再释放“ESC”键。（“ESC”键在键盘左边）按了“ESC”键，立即使主机进入“编辑方式”。机器在编辑方式可以使用四种光标移动键。打入I，光标向上，打入M，光标向下移动，打入J，光标向左移动，打入K，光标向右移动。

要重复移动光标，就要不断地打入光标移动键。要快速移动光标，按下光标移动键(I、J、K或M)，同时按下标有“RPE T”的键，于是光标就迅速向上、下、左或右移动。

如果光标到达屏幕的顶部，这就停止。如果光标达到屏幕的底部，它也会停止且语句行开始向上移动。如果光标到达右边，它将消失，重现在下一行的左边。如果它到达左边，光标将重现在上一行右边。要退出编辑方式，按空格键。现在主机进行正常方式。

当按了“ESC”后，可使光标在编辑方式下移动。而打入除了I、J、K和M以外的任何一个键，可使Apple机退出编辑方式。在开始正常的打键输入之前，必须按一次空格键退出编辑方式。

在监控方式中，可以通过按“ESC”键进入编辑方式。然后打入A（向右移动），或打入（向左移动），或打入C（向下移动），或打入D（向上移动）。在移动其中任何一个键后，机器立即返回到正常方式，在再次移动光标箭以前，必须再按“ESC”键。

怎样在屏幕上从一行中清除一个或多个字符；在编辑方式下，移动光标到要修改的第一个字符上。现在，光标复盖在这个字符上。接着按下一空格键，使机器退出编辑方式，再对要擦除的每个字符再次按空格，每个字符将由空格代替。

怎样清除屏幕上的一行至末尾；在编辑方式中，把光标移至要擦除的第一个字符上，然后按E，被光标复盖的字符至该行的字符全部消失，机器返回到正常方式。

怎样清除屏幕到末尾；在编辑方式中，把光标移到你要擦除的第一个字符上然后按F。屏幕上闪烁光标的字符，以及其后的字符全部消失。机器返回到正常方式。

怎样清除全部屏幕；在编辑方式中打入@，屏幕上的字符全部消失，光标返回左上角，且返回到正常方式。要获得BASIC提示符(>或[)请按return。如不是在编辑方式而要清除屏幕，首先按“ESC”键，然后打入@。

在编辑方式中按I、J、K或M后，机器仍在编辑方式（因为可以继续移动光标）。而如果按A、B、C、D、E、F或@，在执行了其中一条命令后，机器立即退出编辑方式。按空格也使机器返回到正常方式。

所有这些特征只影响屏幕上所显示的内容，而不影响内存贮所存贮的内容。另外，有两个键可以改变存贮器内容，这两个键在编辑程序时是很有用的。这两个键是左向箭头键和右向箭头键。也叫做删除键和重打键。即→，←。

这两个键只影响正在打入的那一行字符。当按下删除箭（左向箭头键）时，光标向左移动一格，擦除刚刚打入的字符。如果刚刚打入的字符在屏幕上消失了，这个字符就被删除了。如果刚刚打入了一个非打印的控制字符，则犹如什么未打入。

不改变屏幕上的任何显示，如刚刚打入的字符是纯光标移动（在编辑方式下按I、J、K、或M）。那么，这种光标移动不输入字符，也决不会改变该行的部分字符。

当按重打键时（右向箭头键）光标向前（右）移动，复制光标所越过的字符。如果用重打键复制了一行字符，然后按return键，其结果就象又重打了这一行字一样。

可以按下重打键的同时，按下REPT键作快速重打。机器将重打光标越过的字符和空白，直掉放掉其中一个或两个键。

可以使用上述的这些特点，修改、插入或删除1行程序或一部分程序。

让我们看一个Applesoft BASIC的简单程序。

```
10PRINT "JABBERWOCK"
```

要用“JIBBER”代换“JABBER”，可以这样做：

1)、列程序在屏幕上。

```
[ LIST 10
```

```
10PRINT "JABBERWOCK"。
```

```
]■
```

2)、按“ESC”键，然后使用J和I键，移动光标到第10行第一个字符（数字1）。

```
10 PRINT "JABBERWOCK"
```

3)、按下一空格退出编辑方式，然后使用向右箭头键和REPT键，快速重打到“JABBERWOCK”的“J”，闪烁的光标将复盖在“A”上。

```
10PRINT "JA BBERWOCK"
```

4)、打入“I”，然后快速重打这一行的其余部分。

```
10PRINT "JIBBERWOK" ■
```

5)、按return，告诉机器替换存贮器中的该行。

```
10PRINT "JIBBERWOCK"
```

```
]■
```

6)、列程序清单，看看你已经做了些什么：

```
10PRINT "JIBBERWOCK"
```

```
]■
```

要在“JIBBER”后插入“JABBER”可这样做：

1)、给程序列表

```
ILIST JLIST
```

```
10PRINT "JIBBERWOCK"
```

```
]■
```

2)、首先按“ESC”，再按I，上移光标，再按J，使光标移到10行的数字1上，再按空格键，退出编辑方式，再使用向右箭头键，使光标越过“JIBBER”。

```
10PRINT "JIBBERW OCK"。
```

3)、按“ESC”再按“J”，左移光标到“J”。

```
10PRINT "J IBBERWOCK"。
```

4)、按空格键，退出编辑方式，在“JIBBER”的地方打入“JABBER”。需要的话可用快速重打，在屏幕上“JABBER”代换了“JIBBER”。现在“JIBBER”，“JABBER”都打入到现行这一行里，而屏幕上却显示为：

```
10PRINT "JABBERW OK"。
```

5)、快速重打该行其余的字符，再按“return”。

```
10PRINT "JABBERWOCK".
```

```
]■
```

6)、给程序列表，检查已经做了什么，

```
]LIST
```

```
10PRINT "JIBBER JABBERWOCK"
```

```
]■
```

从现行的行删除“JABBER”这样作：

1)、给程序列表

```
]LIST
```

```
10PRINT "JIBBER JABBERWOCK"
```

```
]■
```

2)、按“ESC”，使用I、J键移动光标到1，再按空格键，退出编辑方式，再使用向右箭头键，快速重打，并越高“JIBBER”。

```
10PRINT "JIBBER J ABBERWOCK".
```

3)、按“ESC”再按K键使光标越过“JABBER”落在W上。虽然“JABBER”仍然显示在屏幕上，但它可避免进“JABBER”重新打入现行行。

4)、按空格键，退出编辑方式，然后重打该行中其余字符。再按“return”键。

```
10PRINT "JIBBER JABBERMOCK".
```

```
]■
```

5)、给程序列表，检查已经做了什么，

```
LIST
```

```
10PRINT "JIBBERWOCK"
```

```
]■
```

请试这些方法，仔细观察屏幕上所发生的变化，然后列表，检查存储器有什么变化。通过实践，会更熟悉这两个过程之间的关系。

这儿有一个编辑程序的技巧。如果利用快速重打特点，打入占据屏幕上一行以上的程序，我们会发现屏幕上有多余的空格。可处理掉这些空格，且在屏幕上以窗口形式列表。即为避免把多余的空格重打到现行行，这样做：

1)、按“ESC”的同时再按“@”清除屏幕。

2)、打入POKE33, 33。

3)、按“return”键。

4)、给程序列表。

这使得屏幕显示的文本窗口为33个字符的宽度，语句超过33个字符时另下一行。因而，快速重打一个语句时，若屏幕显示的字符超过33个字符则转下一行的初始位置。

快速打完所有程序后，再打入TEXT，然后按“return”。使文本窗口恢复正常方式。

注：

在宽度约为33文本窗口内打入的字符不过40，否则将冲掉Applesoft BASIC 和程序。

(Applesoft BASIC参考手册中详细地说明了这个问题)。

停止列表特点:

Apple系统中,语言插件的自启动ROM还有另一个有用的特点:即随意停止和启动程序表列。要中断程序列表时,打入CTRL—S,要继续列表时,再次打入CTRL—S。

(CTRL—S通过按下CTRL键的同时,按下S产生)。让我们来试一试。

1)、从磁盘上或盒式磁带上输入一段20行或20行以上的程序。或者在键盘上打入一段新程序。

2)、打入LIST,不要按“return”。

3)、按“CTRL”键的同时按致S”键,停止列表。

4)、再次打入CTRL—S(或除了SHIFT, RESET或CTRL—C以外的其它键),按复列表。

5)、按“RESET”键,完全停止列表。

这种方法很少用。因为程序列表很快,很容易越过所要看的行,这就是为什么要先按下“CTRL”,等待着你要看的那一行程序。

停止列表特点可以暂停执行屏幕上所显示的文本中的程序。让我们用一个简单的例子来试一试

```
10 I=I+1
20 PRINT I
30 GOTO 10
```

1)、RUN程序

2)、打入CTRL—S

当程序中送出一个“RETURN”到屏幕后,为了开始一行新的程序,机器需要查询键盘。如果查到是CTRL—S,机器就暂停程序的执行,等待新指令。

3)、再次打入CTRL—S(或除了SHIFT, RESET, CTRL—C的任一其它键)将恢复程序的执行。

4)、打入CTRL—C,停止程序,出现BASIC提示符:“>”。

对有些程序按CTRL—S将不能暂停程序。例如:

```
10 I=I+1
20 PRINT I
30 GOTO 10
```

这一程序与刚才的程序不一样,是因为在20行末尾加了一个分号。如果打入CTRL—S,机器不作响应。因为该程序送出了一连串的字符到屏幕上(由于分号的原因)而不是一个字符一行。它不送“RETURN”到屏幕上。因此,机器不查询键盘。

附录A 具有BASIC特点的自启动ROM摘要

命令	作用
插入BASIC盘并且接通电源	将不在ROM中的 BASIC 加载到语言插件RAM中
插入DOS3.2盘并且接通电源按“RETURN”	引导DOS并且运行向候程序
按“reset”	停止并重新启动，保存的程序
按“ESC”	进入编辑方式
按空格键	退出编辑方式
按I、J、KM	在编辑方式中移动光标，继续处于编辑方式中
按左向箭头键	删除刚刚打印的字符
按右向箭头键	重打光标越过的所有字符
按“ESC”再按“E”	从光标所在位置开始清除到一行末尾
按“ESC”再按“F”	从光标所在位置开始清除到屏幕最后
按“ESC”再按“@”	清除整个屏幕
打印CTRL—S	停止列表或在下一个RETURN
打印CTR—S	停止列表
	恢复（继续）列表

附录B 自启动的ROM的技术情况

本附录中介绍了自启动ROM和监控ROM之间的差异。

加到自启动ROM里的子程序。

名称: Apple I (\$FB60) CALL—1184

作用: 这个子程序清除屏幕并使字符串“APPLE I”写进文本缓冲器的第一行，因此，RESET的同时，屏幕显示“APPLE I”。

送入: 在调用这个子程序以前，要确保滚卷窗口参数值设置在极限值。要做到这一点可以使TEXT命令。

或 10 POKE 32,0

20 PODE 33,40

30 POKE 34,0

40 POKE 35,24

送出: A = \$ C1, Y = 0;

X = 送入值

状态: N = 0

Z = 1, C = X, V = X

名称: SETPWRC (\$FB6F) CALL—1169

作用: 在按“reset”后, 使机器进入监控方式。它设置\$3F4的内容为\$3F3地址的内容和\$A5半加。

送入: 无条件, \$3F3(POKE1011)地址应设置为对这个调用有意义的复位矢量的高位地址。

送出: $A = EOR \# \$A5$, \$3F3, V = 送入值, X = 送入值。

状态: $N = X$, $Z = X$, C = 送入值, V = 送入值

注解: 要使Appte机靠RESET进入监控方式, 做一下CALL—155。这将使机器进入监控方式。在monifor, 提示符(*)出现后打入3F2: 66FF5A, 按RETURN。

在做了这些以后, 无论何时按RESET, 机器将进入监控方式。(监控方式在Apple参考手册中说明)。这个特性使之在按“RESEF”后, 通过设置复位矢量指出子程序的地址, 使机器执行机器语言子程序。例如, 假定我们有一个子程序的入口点为\$803。若我们在该点设置一个断点, 那么程序转向该点时, 计算机将停在\$805。(计算机总是停在比断点大2个字节地址)。首先打入CALL—151按“return”。进入监控方式。出现“*”后, 打入803: 0按“return”。让我们设置2个字节复位矢量指向\$803, 在“*”出现后打入。

3F2: 3 6按“return”首先打入低位字节, 然后再打入高位字节。

现在, 调用一个子程序, 这个子程序防止机器在按“RESET”时, 再次进行引导。可打入FB5FG按“return”。当按“RESET”后, 执行\$803的子程序, 并显示:

\$805 - A = 03, X = 38, Y = 1B, P = 35, S = EB。

第三页存贮器的用法:

地址	用法	约定值
\$3F0, 3F1	中断矢量	\$59\$FA
\$3F2	复位矢量低位	\$03*
\$3F3	复位矢量高位	\$E0*
\$3F4	电源接通	\$45*

自启动ROM的限制

1、不能使用ROM里16位整数乘法子程序。因为这些都不在自启动ROM中。

2、不能使用任一部分单部模拟代码子程序, 原因同上。

* 在电源接通, 无磁盘系统情况下。

附录 C 语言插件的存贮器分配图

语言插件RAM语言插件ROM

系统ROM地址空间

\$ FFFF				
\$ F800				
\$ F7FF				
\$ E000				
\$ DFFF				
\$ D000	第二个4 K 库	第一个4 K 库		

附录D语言插件控制码

RAM选择 对\$ D000—\$ DFFF		RAM/ROM选择以及RAM写保护选择
第二个4 K 库	第一个4 K 库	
\$ C080 —16256	\$ C088 —16248	选择RAM读写保护RAM
\$ C081 —16255	\$ C089 —16247	不选择RAM读(可选择ROR读)二次或二次以上读该地址可 写RAM
\$ C802 —16254	\$ C08A —16246	不选择RAM读(可选择ROM读)写保护RAM
\$ C083 —16253	\$ C08B —16245	选择RAM读二次或二次以上读该地址可写RAM

这些控制码单独做三件事：写保护或写语言插件的RAM，选择或不选择RAM读，以及对地址空间\$ D000~\$ DFFF选择第一个4 K库或第二个4 K库。因为地址空间\$ C000—\$ CFFF已被输入输出设备和子程序占用，仅在12KRAM的地址空间(\$ D000~\$ CFFF)，有用的地址16KRAM在地址空间\$ D000~\$ DFFF设置2个4 K库。4 K库之间的转换由控制码控制。这样，8K的存储器就能用这4K地址空间寻址了。语言插件上余下的8 KRAM地址\$ E000~\$ FFFF无存储体转换。

控制码(专用地址)可用16进制和10进制两种形式表示。16进制形式(用\$表示)用于汇编语言程序，10进制形式用于BASIC程序。控制码用\$ C8X的形式表示，其中X是16进制数字。

控制码\$ C084~\$ C087和\$ C08C~\$ C08F与控制码\$ C080~\$ C083和\$ C088~\$ C08B的作用相同，每个地址的第2位和被忽略。

控制码的第3位选择哪个4 K库的RAM，定位在地址空间\$ D000~\$ DFFF里，如果第3位=0，第一个4 K库就划入\$ D000~\$ DFFF里；如第3位=1，第二个4 K库RAM划入到\$ D000~\$ DFFF。

当不选择RAM时，就选择语言插件上的2KROM地址(F800~FFFF)和主板上的ROM地址空间从\$ D000~\$ FFFF。

当不选择RAM时，它仍可写而不可读。

加电后按“RESET”键，初始化ROM为读方式，RAM为写方式，和选择第二个4 K库划入\$ D000~\$ DFFF。

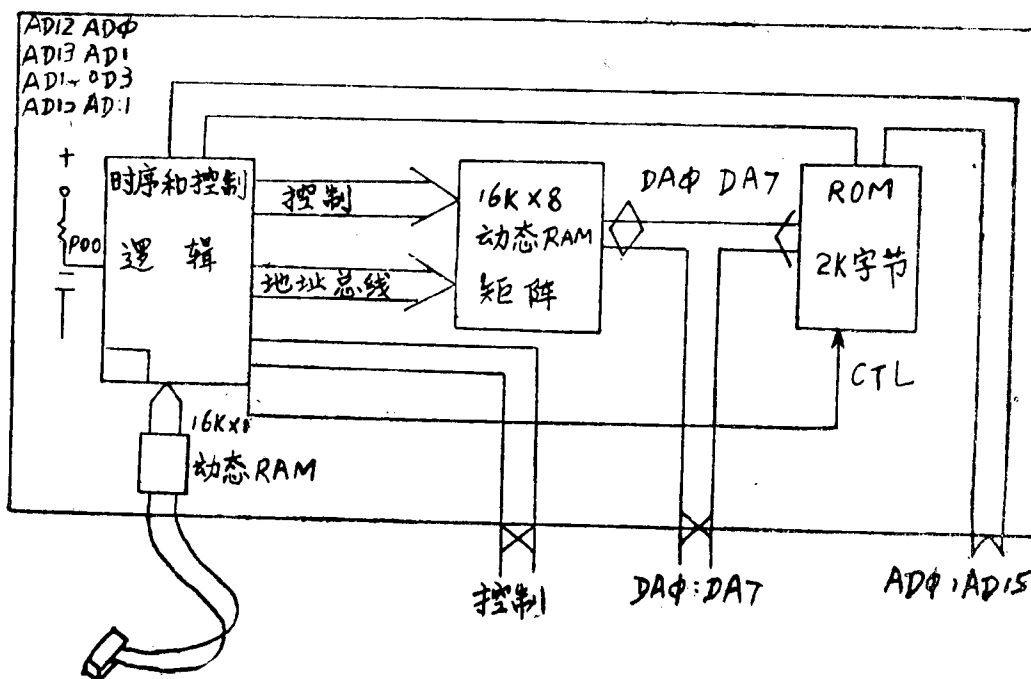
作为使用这种代码的例子，请看下面一部分汇编语言程序：

```
AD    83    C0    LDA $    C083           选择第二个4 K库和读/写。
```

AD	83	C0	LDA \$ C\$83	连续2次读
A9	D0		LDA# \$D0	* SET UP 建立
85	01		STA BEGIN	* NEW 清除内存
A9	FF		LDA# \$FFFF	* 主存
85	02		STA END	* 指针
20	97	C9	JSR RAMTST	* 对于12K存贮器进行测试
AD	8B	C0	LDA \$ C08B	* 选择第1个4K存贮体
20	97	C9	JSR RAMTST	* 使用上述指针
AD	88	C0	LDA \$ C088	* 选择第一个4K库写保护
A9	80		LDA# \$80	
E6	10		INC TSTNUM	
20	58	C9	JSR WPTSINIT	
AD	80	C0	LDA \$ C080	* 选择第2个4K库和写保护
E6	10		INC TSTNUM	
A9	01		LDA# PAT12K	
20	58	C9	JSR WPTSINIT	
AD	8B	C0	LDA \$ C08B	* 选择第一个K4库和读写
AD	8B	C0	LDA \$ C08B	* 连续二次读
E6	0E		INC RWMODE	* 标志RAM在读/写
E6	10		INC TSTNUM	
A9	08		DA# PAT4K	
20	58	C9	JSR WPTSTNIT	

LDA指令对指定的存贮单元进行一次读出。不使用累加器的数据。这个程序解释了两
次连续执行LDA指令的特殊用法。

附录E: 语言插件的功能方框图



16引脚双立直插到系统RAM插座

一、前言

本手册，着重介绍 Apple II 微型机系统中的MX系列打印机接口部件的使用，并对硬件作简要说明。MX系列打印机接口，用于连接主机和MX系列点阵打印机，它可使MX系列打印机打印文本、图象，对高分辨率屏幕进行硬拷贝，还能使打印机执行文本格式操作。

用MX系列打印机接口部件连接主机和 MX—80—II 打印机之前，希望用户阅读一下本手册。了解正确的联机步骤，以防操作不当，损坏接口部件或主机，并掌握使用打印机的一般方法。

二、组成

图 1 是Apple II 的MX系列打印机接口部件，它由接口板（8132）和接口电缆（8231）组成。

图 1 略

三、功能

三、功能

1、文本打印

- 设置打印行宽（40~132字符）/行。
- 打印小写字母

2、位图象打印

该接口可以使具有位图象打印功能的MX系列打印机产生以下的CRT屏幕的硬拷贝。高分辨率（H1—RES）屏幕显示硬拷贝：

2.1、标准打印

2.2、H1—RES第一页和H1—RES第二页之间的逻辑操作打印

2.3、反相打印

2.4、行打印

2.5、放大打印

2.6、倍密度打印

3、打印格式

APPLE II BASIC不提供格式功能，但MX系列打印机接口部件通过固化在ROM 中的软件提供格式功能。

- 整数表示
- 定点表示
- 指数表示

四、安装

1、切断主机、MX—80—II 打印机电源。