

HOPE



适用于 Apple II 机、中华学习机及其兼容机

Apple II 汇编语言

〔美〕 Marvin L. De Jony 著

汤德芳 龚燕枝 编译



中国科学院希望

8.87121

463

中国科学院希望高级电脑技术公司

适用于Apple II机、中华学习机及其兼容机

Apple II 汇编语言

(美) MarvinL. De Jony著

汤德芳 龚燕枝 编译



中国科学院希望高级电脑技术公司

0110154

译者序

众所周知, Apple II 微型机是以价格低廉、结构灵活和有丰富的软件支持而闻名于世。该机在我国已被广泛用于教学、科研等方面。另外,我国当前正在推广使用中华学习机种,它的CPU与Apple II 完全一样,都是6502微机处理器,所以如何使更多人掌握6502汇编语言已成为大家所关心的问题。过去对Apple II 机汇编语言介绍很少,即使有少量译文,往往初学者很难看懂。

Marvin L. De Jong 先生写的“Apple II 汇编语言”一书特别适宜于初学者阅读。该书的特点是通俗、易懂和实用。並且在学习本书之前,不要求读者具有6502汇编语言知识。书中除了系统讲述Apple II 汇编语言和实用程序之外,还结合6522通用适配器的程序设计与A/D、D/A转换器应用等,着重向初学者介绍了如何把软件和硬件知识很好结合起来解决实际问题。这正是本书的精华部份。希望读者倍加注意。

本书在内容安排上严格遵守循序渐进原则。所以初学者在阅读时最好按照章节先后来学习。对十六进制等数制尚不够清楚的读者,应先看书后的附录A。另外,对于每章后的实习题目,读者一定要很好上机去做,因为作者除了巩固每章所学过内容之外,还在这里补充了一些实际应用中容易遇到的问题。

由于希望电脑公司培训资料部秦人华经理的关心和大力支持,才使本书能及早与读者见面,对此,特向她致以衷心的感谢!

序 言

欢迎你来到一个领域,在此所用的词汇仅有56个,而且都是由三个英文字母组成,如象LDA、STX、EOR 和PLA 那样。令人警奇的是:用这些有限词汇,就够达到你的目的。你可造出一台运算速度非常快的计算器,以至它在计算超越函数时,你丝毫没有等待的感觉;或者,你可以通过程序,令微型计算机做极其缓慢、乏味又为人类所厌烦的事情。人们之所以这样热衷于微型计算机,正是由于这简单的56个词汇和微型计算机具有巨大的功能和多种用途。

作者提出研究汇编语言设计,丝毫不是想贬低高级程序设计语言的用途和优点,只是使读者尽可能深入到微处理器中去,不是请读者亲自再去组装一台微处理器。通常,经过这样学习,使读者对计算机系统有更多的了解。也许某些年轻读者还会从中受到启示,去探索一些在微处理器内部无法获悉的领域,如逻辑符号、数字电子学和量子力学等。

作者希望读者把学习6502汇编语言程序设计作为一种享受。可轻松地阅读本书、跳过难懂部分。你不必在重要内容下划线、绘符、记笔记、背诵、或冥思苦想。书后也没有考试。

另外,本书也是在培养读者成为一个合格的汇编语言程序设计方面的一次重大尝试。如果这也是读者的目标,那么,为此还须具备一定的学习方法。读者须随身携带本书,多在计算机键盘上下功夫,反复操作,或是写书中所提出的程序,或是自己设计程序。仅仅阅读一本计算机书是学不会计算机编程的。读者须亲自编写、操作和调试程序。

用这种方法来学习，那么，本书就成为读者学会汇编语言程序设计的一种得力工具。当然，读者的计算机是另一种重要的手段。但是，最宝贵的源泉恰是读者的想象力和创造性。这种学习方法，未必比你躺在睡椅上阅读书籍所得的乐趣少。让你的求知欲成为学习的动力，切勿强制性学下去，这样，对学会一种新型程序设计语言来说是非常有利。

由于汇编语言程序设计师要比高级语言程序设计师更接近微型计算机硬件，因而全书有些基本接口技术和设计。万一你发现硬件讨论内容太难，就越过这些麻烦的章节。有很多饶有趣味的题目并不需要熟知集成电路，电路图和接线等。

附注：本书所有程序的机器语言版本，都是以DOS 3.3型、16扇区软盘供应，但不带文本。若你有兴趣购买这类软盘，请写信给作者。

地址为：

103 Hero Drive, Branson, MO 65616
U. S. A.

Marvin L. DE Jong

目 录

第一章 微型计算机系统	(1)
1.1 引言.....	(1)
1.2 微型计算机系统的组成.....	(1)
1.3 地址空间和存储器结构.....	(3)
1.4 读、写操作.....	(6)
1.5 从程序员角度考察6502结构	(7)
1.6 输入/输出基本概念.....	(8)
1.7 Apple机监控程序.....	(11)
1.8 Apple机监控程序操作实习.....	(11)
第二章 简易汇编语言程序的编写和执行	(13)
本章要求.....	(13)
2.1 引言.....	(14)
2.2 微型计算机指令.....	(14)
2.3 寻址方式.....	(19)
2.4 微型计算机程序.....	(20)
2.5 汇编语言程序的编写方法.....	(21)
2.6 JMP和BRK 指令.....	(24)
2.7 数据传输指令示范程序.....	(24)
2.8 实习.....	(25)
第三章 转移和循环	(28)
本章要求.....	(28)
3.1 引言.....	(29)
3.2 微处理器状态寄存器.....	(29)
3.3 转移指令.....	(31)
3.4 子程序调用.....	(33)
3.5 Apple键盘的读入.....	(34)
3.6 增量和减量指令.....	(36)
3.7 定时循环.....	(36)
3.8 有关符号问题.....	(38)
3.9 定时循环和游戏杆的一种应用.....	(38)
3.10定时循环和喇叭输出.....	(42)
3.11按钮开关状态的读入.....	(43)
3.12步进马达的控制.....	(45)
3.13实习.....	(48)
第四章 逻辑运算和移位、循环操作	(51)
本章要求.....	(51)

4.1 引言	(51)
4.2 逻辑运算	(51)
4.3 6502指令系统的逻辑指令	(53)
4.4 逻辑指令的应用	(56)
4.5 循环移位和移位指令	(60)
4.6 实习	(64)
第五章 算术运算	(68)
本章要求	(68)
5.1 引言	(69)
5.2 标记位修改指令	(69)
5.3 ADC指令	(69)
5.4 SBC指令	(73)
5.5 十进位方式计算	(74)
5.6 比较指令	(75)
5.7 四位乘法程序	(76)
5.8 八位乘法程序	(79)
5.9 除法	(79)
5.10 ASCII 码转为十六进制数	(81)
5.11 实习	(84)
第六章 寻址方式: 变址寻址	(87)
本章要求	(87)
6.1 引言	(87)
6.2 绝对变址寻址	(87)
6.3 零页变址寻址	(89)
6.4 变址寻址程序举例	(91)
6.5 间址变址寻址	(98)
6.6 变址间址寻址	(106)
6.7 实习	(108)
第七章 子程序、堆栈和中断	(111)
本章要求	(111)
7.1 引言	(111)
7.2 子程序	(112)
7.3 堆栈	(113)
7.4 堆栈操作及堆栈在存储上的应用	(115)
7.5 中断	(119)
7.6 Apple II的中断处理结构	(123)
7.7 中断控制式键盘输入	(126)
7.8 24小时时标	(130)
7.9 在BASIC程序中使用6502汇编语言	(133)

7.10 实习	(134)
第八章 程序设计专题补充	(138)
8.1 引言	(138)
8.2 6502寻址方式概述	(138)
8.3 奇偶校验位与7位字符码相加	(140)
8.4 二进制转BCD和BCD转为二进制	(141)
8.5 N字节二进制数转为BCD数例行程序	(144)
8.6 BCD数转为ASCII字符子程序	(146)
8.7 N字节BCD数转为二进制数	(149)
8.8 符号数计算	(152)
8.9 实习	(155)
第九章 6522程序设计	(155)
本章要求	(155)
9.1 引言	(155)
9.2 输入/输出原理复习	(157)
9.3 6522作简单的输入/输出操作	(158)
9.4 6522作复杂的输入/输出操作: 打印机接口	(163)
9.5 6522作定时和计数用	(171)
9.6 T_1 用于连续运行方式	(176)
9.7 T_2 用于脉冲计数	(180)
9.8 用 T_1 和 T_2 产生长时间延迟	(181)
9.9 精密定时器	(183)
9.10精密定时器程序的应用	(188)
9.11实习	(191)
第十章 应用	(197)
10.1 引言	(197)
10.2 莫尔斯电码传送程序	(198)
10.3 莫尔斯电码接收例行程序	(214)
10.4 利用A/D转换器的应用	(220)
10.5 数/模转换器接口	(230)
附录A 十进制、二进制和十六进制数字系统	(237)
本附录要求	(237)
A.1 引言	(237)
A.2 数	(237)
A.3 十进制数	(237)
A.4 二进制数	(238)
A.5 比特、字节和四位字节	(240)
A.6 十六进制数	(241)
A.7 习题	(243)

A.8 习题答案	(244)
附录B 电路和程序的补遗	(244)
B.1 单步电路和程序	(244)
B.2 计算机辅助学习 (CAL) 程序	(251)
附录C 某些集成电路管腿图	(254)

第一章 微型计算机系统

本章要求

学完本章后，读者应掌握以下几点：

- 能辨认微型计算机系统主要部件，并能叙述它们的功能。
- 懂得地址空间概念和存储器结构。
- 熟悉读、写操作。
- 了解基本输入/输出概念。
- 会用Apple机监控程序来检查和修改存储单元的内容。
- 会用Apple机监控程序列出机器语言和汇编语言程序。
- 能辨认6502微处理器的各个寄存器。

1.1 引言

对于BASIC或PASCAL这类高级语言来说，即使用户对微型计算机内的硬件毫无所知，或者对微型计算机工作方式一窍不通，他仍然可成为一名优秀的计算机程序员。但若要用6502汇编语言来写该计算机的程序时，则上述说法就不能成立。对于后者，用户必须对微型计算机系统的结构和微处理机的构造有所了解。这种知识既不复杂也易掌握。本章目的即在这方面介绍一些必要的概念，以便迅速转到汇编语言程序的编写上。

1.2 微型计算机系统的组成

在计算机文献中，微型计算机和微处理器这两个术语经常可通用。在此我们想谈一下它们的主要差别。微型计算机是指整个计算机系统，多半按装在桌面上。作者所用的微型计算机系统如图1—1所示。而微处理器仅是微型计算机中一个部件。通常微处理

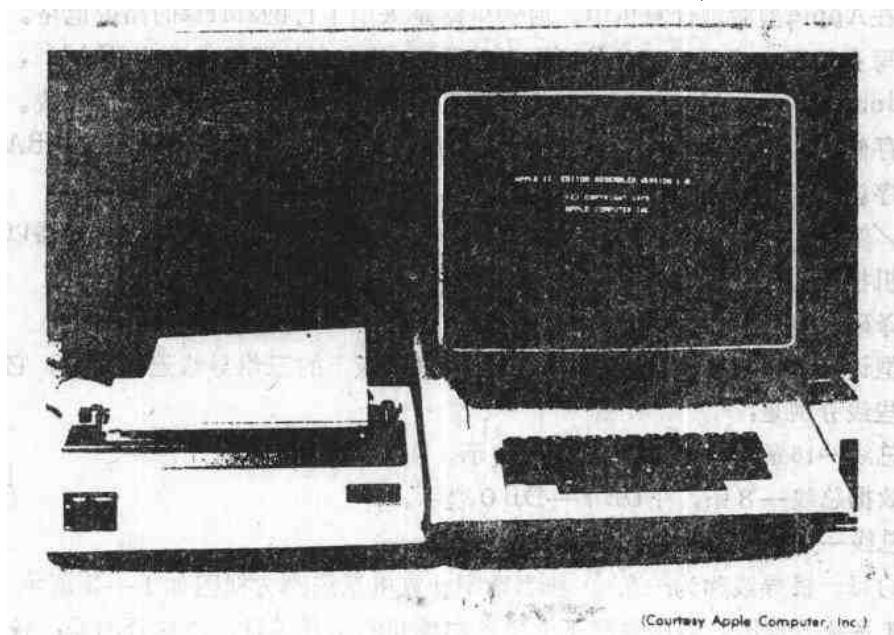


图1.1 Apple 微型计算机

器是一块有40个管腿的集成电路，其中最普及的有6502、6809、8085、1802、Z80、8080A和8086这一类微处理器。6502微处理器的照片如图1—2所示。若你注视一下Apple微型计算机的内部，就会看到许多集成电路，但很容易查到6502微型处理器。微处理器是微型计算机的大脑中枢。

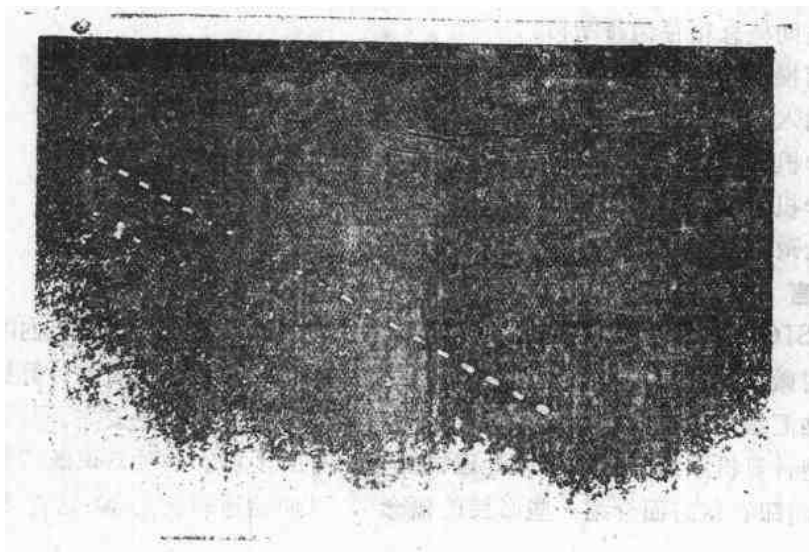


图1—2 6502微处理器

典型微型计算机包含下列部件：

- 一块微处理器，例如6502。
- 一块时钟电路，由它来管理微处理器的运行，它对处理器所执行的全部操作予以协调。在Apple型微型计算机中，时钟电路是采用了1.023MHz的晶振电路。
- 读/写存储器，有时缩写为R/W存储器。这类存储器通常称为RAM，它由英文Random Access Memory（随机存取存储器）这三个词的首字母所组成。
- 只读存储器（ROM），在此装有须计算机长期保存的程序。监控程序和BASIC的解释程序就是这类程序的二个实例。
- 输入/输出（I/O）装置，例如键盘、视频监视器、游戏杆、软盘驱动器以及其它由计算机控制的装置。
- 地址译码电路。

微型计算机系统各部件由三组线即印刷电路板上的三组导线连接而成，它们称为总线。这些线分别是：

- 地址总线—16条，用AD15—AD0表示。
- 双向数据总线—8条，用DB7—DB0表示。
- 控制总线—一条的数目不定。

总线中的每一根导线称为一条。典型微型计算机系统的方框图如1—3所示，它也是Apple II系统方框图。该图说明了上述各部件间的相互关系。须解释的是：各存储器件也许只译码其中一部分地址线，而高位地址线由地址译码器来译码。

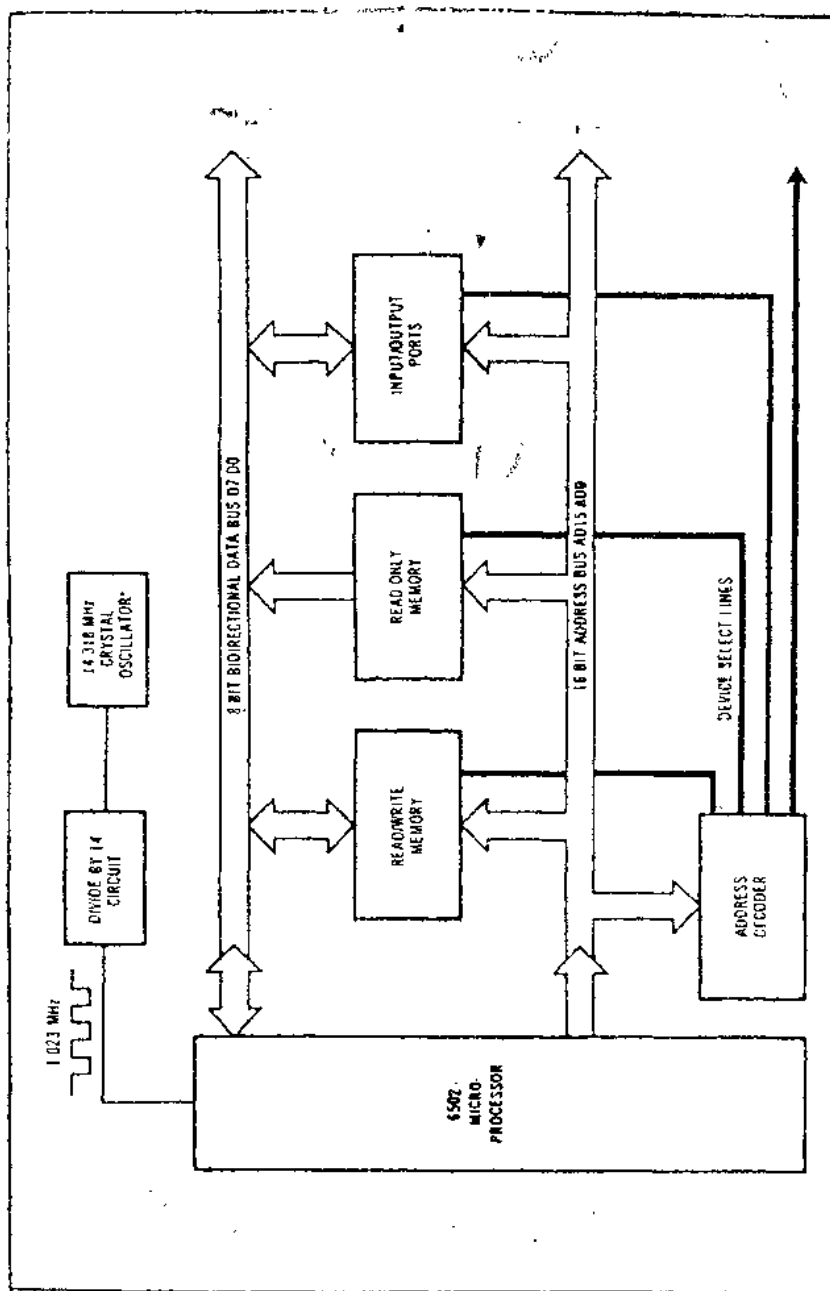


图 1—3 用6502组成的微型计算机系统方框图

1.3 地址空间和存储器结构

6502微处理器上有16条管腿是用于寻址的，也就是说，它们控制了16条地址线（即地址总线）上两种都可能会出现的逻辑电平。虽然它们在电参量上是0伏和5伏的电压值，但在逻辑电平上却称为“0”和“1”。地址总线AD15、AD14、AD13……AD0上所有0或1逻辑电平的组合就构成了一个16位的二进制数，我们称它为一个存储单元的地址。根据地址便能确定（或辨认）和管理所有存储单元。

由于存在 2^{16} 个互相不同的16位二进制数，于是6502就能寻址 65,536 个存储单元（凡对二进制、十六进制数、“位”、“字节”、“四位字节”等术语不熟悉的读者，在阅读以下内容之前，最好先学习一下附录A）。通常对16位地址是分两个字节来讨论的：

- 高位字节，即“高位地址”，简称为ADH，其值由地址的高8位（AD15—AD8）决定。
- 低位字节，即“低位地址”，简称为ADL，其值由地址的低8位（AD7—AD0）决定。

上述的每个字节都可由两个十六进制数（0—9和A—F）来表示。于是，整个地址就由四个十六进制数表示。本书在所有十六进制数之前都冠以前缀“\$”。例如 \$F9A6就是一个地址，其中 \$F9是ADH，\$A6是ADL。每个存储单元对应一个16位地址，于是地址空间就是微处理器能寻址到的全部存储单元的总和。

宜将整个地址空间划分成若干个块。块的最小尺寸叫做页，它有256（ 2^8 ）个存储单元。表1—1说明了地址空间是如何划分成页的。该表说明ADH是页数，而ADL是在该页内的存储单元。我们将会看到：零页是由地址为 \$0000—\$00FF的存储单元组成，而一页是由地址为 \$0100—\$01FF的存储单元组成，它们在汇编语言程序设计中是有特定的意义。

Table 1-1. Dividing the Address Space Into Pages

Address High ADH	Address Low ADL	Address
(Binary)		(Hexadecimal)
00000000	00000000	\$0000
00000000	00000001	\$0001
⋮	⋮	⋮
⋮	⋮	⋮
00000000	11111111	\$00FF
00000001	00000000	\$0100
00000001	00000001	\$0101
⋮	⋮	⋮
⋮	⋮	⋮
00000001	11111111	\$01FF
⋮	⋮	⋮
⋮	⋮	⋮
11111111	00000000	\$FF00
11111111	00000001	\$FF01
⋮	⋮	⋮
⋮	⋮	⋮
11111111	11111111	\$FFFF

表 1—1 地址空间划分成页

比页容量再大些的块即为1024（ 2^{10} ）个存储单元，一般称它为1K存储器。由于

$2^{10}=1024$ ，为了在1K块内单一确定每个存储单元就需占用10个地址位。这样还剩下6个地址位AD15—AD10。此6个地址位又能确定 $2^6(64)$ 个块，于是，在地址空间内就有64个1K存储器。通常半导体存储器（无论是RAM还是ROM）是以“K”为销售单位的。例如，买得16K RAM，把其装到容量为32K的 Apple II 机上，就使它成为48K的机器了。图1—4说明了如何用4K位或16K位集成电路片按装到 Apple 机上，组成各种容量的读/写存储器。

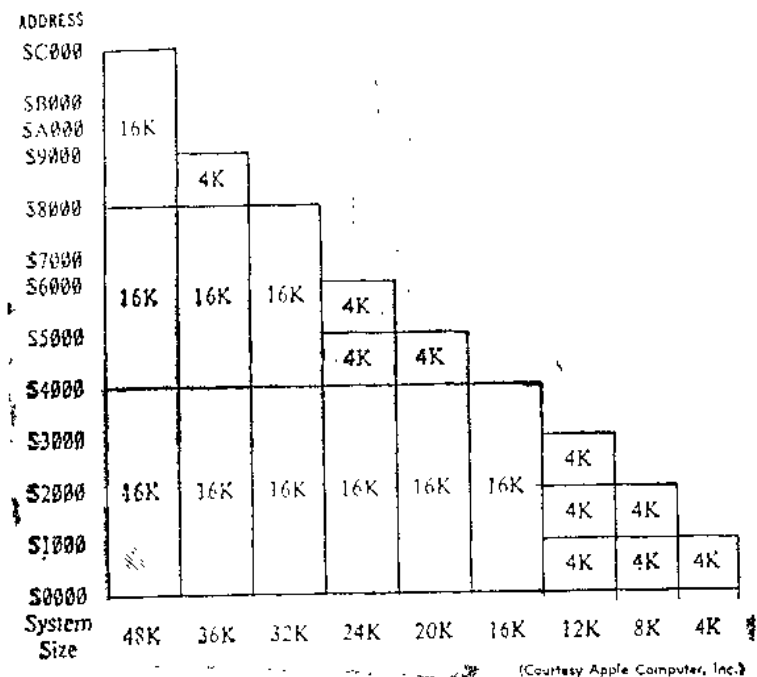


图1—4 Apple 机读/写存储器的配置

只读存储器（ROM）也是以“K”为单位进行封装、销售和装配的。例如 Apple 机的监控程序就装在2K ROM集成电路内，其存储单元地址为\$F800~\$FFFF。此外，在Apple电路板上还能在ROM插座上再插5个2KROM，例如在这些ROM上可装有解释程序。

像Apple II这些用6502微处理器组成的微型计算机，通常都是这样来安排地址空间的：即把读/写存储器放在地址空间的“低端”（地址从\$0000开始往上升）；ROM位于地址空间的“高端”（地址从\$FFFF开始向下降）；而将输入/输出装置的地址排在地址空间中段某处。我们把各存储功能部件在微型计算机地址空间上的分配图叫做存储区映象图。Apple的存储区映象图如图1—5所示。并不是每一个 Apple 都装满48K读/写存储器（在图1—5上称RAM），也不见得像 Apple 存储区结构那样一定要有2K（2048）个输入/输出（I/O）存储单元。总之，当你对本书将要讨论的6502某些特性有所了解时，你就会搞清Apple机和以6502为基础的其它系统在存储区结

构上所隐含的奥妙。

System Memory Map		
Page Number:		
Decimal	Hex	
0	S00	RAM (48K)
1	S01	
2	S02	
.	.	
.	.	
190	SBE	
191	SBF	I/O (2K)
192	SC0	
193	SC1	
.	.	
.	.	
198	SC6	
199	SC7	I/O ROM (2K)
200	SC8	
201	SC9	
.	.	
.	.	
206	SC E	
207	SC F	ROM (12K)
208	SD0	
209	SD1	
.	.	
.	.	
254	SFE	
255	SFF	

图 1—5 Apple 机存储区映象图

1.4 读、写操作

为了阐明读、写操作，我们必须对数据总线、控制总线的 $R/\bar{W}$ 线和存储单元有所了解。6502上有8个管腿是连到数据总线上，将6502上读/写($R/\bar{W}$)管腿连到一条控制总线上，即所谓 $R/\bar{W}$ 线上。一个读/写存储单元是八组双稳态半导体器件组成，每组器件能存储一位二进制数。八位数据总线和一个存储单元上的八个数据位都按以下顺序排列：D7, D6, D5……, D0，从最高有效位到最低有效位。最好是把一个存储单元所存入的八位信息当作一个8位码，但现对这样信息有多种称呼：如称为一个数、一个字或一个数据字节。根据上下文情况，本书中将会用到上述各种名称。

现在我们准备下二个主要定义：

- 所谓读操作是指一个待传输的8位码（或数），由地址总线上地址所指定的存储单元出发，经数据总线到达微处理器上某个8位寄存器这一过程。
- 所谓写操作是指一个待传输的8位码，由微处理器中某个寄存器出发，经数据总线到达地址总线上地址所指定的存储单元内这一过程。

有时装入、储存两词可与读、写两词同义使用。

由于读（装入）操作是使数在一个方向移动，而写（储存）操作是使数在另一个方向移动，所以说数据线是双向的。此外，由于传输和储存信息是8位的，故称6502是一种8位微处理器，因而Apple机是一种8位微型计算机。

地址译码电路的功能就是在65536个可选存储单元中，仅使一个存储单元被赋能（激活），也就是只有一个存储单元在写操作时能接受数据，在读操作时能送出数据。在Apple II机中利用地址空间是划分成块的原理，通过专门集成电路如74LS138和其它芯片来完成部分译码功能。余下的译码由存储器芯片自己来完成。通常地址译码器对地址总线的高位线（AD15，AD14，……）进行译码得到片选位，通过它再来激活一个特定的存储方块。存储器芯片译出低位地址线（……，AD1，AD0），从而使所选方块上的特定存储单元被赋能。图1—3对上述很多概念进行了说明。

顾名思义，ROM存储器仅能做读操作，写操作没有意义。同样，微处理器仅能向输出口写，从输入口读。对ROM、I/O口操作与对读/写存储器操作一样都涉及到8位码的传输。

为了全面理解读，写操作，应说明的是：在微处理器中，一个寄存器就是一个8位储存单元。寄存器是用来储存微处理器准备进行的操作的数据，或者用寄存来控制微处理器本身的操作。在6502中并不是所有寄存器都与汇编程序设计员有关，所以凡对6502内部操作有兴趣的读者请参阅6500硬件手册。本节仅对汇编语言编程时有用的一些寄存器作些说明和介绍。

1.5 从程序员角度考察6502结构

图1—6是另一种以6502为基础的微型计算机框图。该图侧重于6502程序设计中要用到的五个8位寄存器和一个16位寄存器。由于全面探讨这些寄存器需要些程序设计经验，故在此仅对它作些扼要介绍。

- 累加器（A）的用途是与存储器交换数据。在所有算术和逻辑运算中，都是把参加运算的两个数之一放在累加器中。
- 两个变址寄存器（X和Y）的作用与累加器类同，可作为数据传送用。当用变址寻址方式（请予阅第六章）时，它们可作为基址的一个8位变址（下标）。由于这类寄存器内的数每次可递增或递减一个单位，故也作循环计算器用。
- 微处理器状态寄存器（P寄存器）含有标志位或状态码，随着程序每一指令的执行，状态码也许会变化。细节请参阅第三章。
- 程序计数器（pc）是一个16位寄存器，它指出了被运行着程序的下个字节放在存储器中的存储单元地址。宜将程序计数器分成两个8位字节，把它的高位字节称为pcH，低位字节叫做pcL，在程序运行期间，如果你能查看程序计算器的话，它应显示出读数，要不是程序产生了转移或跳跃，这时应在计数器读数上显示是正向还是负向跳跃。
- 堆栈指示器（S）是一个8位寄存器，它是指向存储器第一页（地址为\$0100—\$01FF）上的某个单元。当访问堆栈的读/写操作产生时，地址线上的高位字节是\$01，这时，可以把堆栈指示器当作一个16位寄存器，其高位字节被固定在\$01，而它的低位字节就是堆栈指示器的数值。堆栈是用来储存正确处理子程序调用和中断所需的信息。更详尽的细节请参阅第七节。

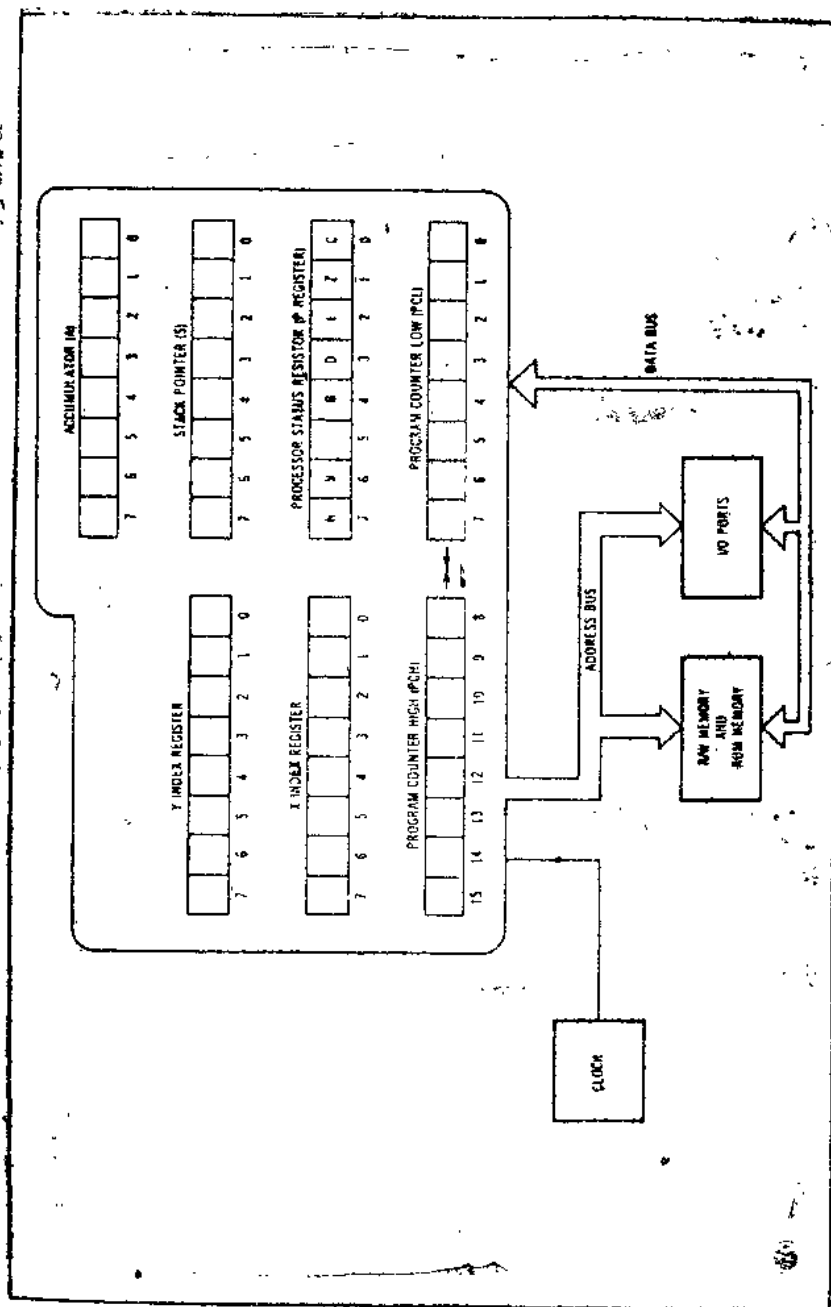


图 1—6 6502 微型计算机模型

到此为止，我们已初步讨论了6502的结构，和介绍了微型计算机中所遇到绝大多数硬件。

1.6 输入/输出基本概念

微型计算机的输入/输出操作是其各种应用的基础。Apple II最典型的输入操作是键盘输入口的读入。最代表性的输出操作是视频监视器上字符的写入。更灵巧的是用游戏I/O口所产生的那些I/O操作。

输入口的作用是向计算机提供外界信息。用一台外部设备（例如键盘）控制一块集

成电路上 8 个管腿电平的方法就可实现上述功能。它把 5 伏左右电压变换成逻辑 1，将近于 0 伏电压转化为逻辑 0。把功能为输入口的集成电路连到微型计算机的总线上。根据图 1—7 连接 Apple 机的键盘输入口。当 6502 把该输入口的地址放到地址总线上时，上述集成电路就控制了数据总线的逻辑电平，于是 6502 便读到了此输入口用电平表示的二进制数。例如用 Apple 机键盘敲入的数就是在地址为 \$C000 处读得的。本书将经常用到此输入口。

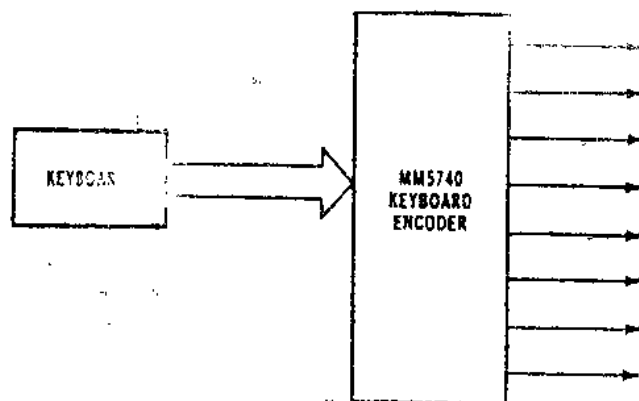


图 1—7 地址为 \$C000 的键盘输入口方框图

上述工作模式称为存储映象输入，其意思为：只有当输入口被地址总线访问到时，才向 6502 输送数据，从这意义来说，输入口与其它存储器的功能类同。以 6502 而论，输入口仅是其地址空间的另一种存储单元。从图 1—3 还可看到：I/O 口与存储器器件一样，也连到地址和数据总线上。

Apple 机键盘上每个键都连到标号为 MM5740 的集成电路上，它是一个键盘编码器。对于键盘上每个键，编码器都会产生一个与其对应的 8 位码。例如按下 A 键时，集成编码器输出端便出现 \$C1 码。请参阅图 1—7 方框图。现把各字母数字键相应的十六进制码汇集于表 1—2。当读到 \$C000 存储单元处的键盘口时，上述表中某个 8 位码就被传送到微处理器上。

Table 1-2. Keyboard Code Table (ASCII)

Key	Code	Key	Code	Key	Code
A	\$C1	N	\$D5	0	\$B0
B	\$C2	O	\$D6	1	\$B1
C	\$C3	P	\$D7	2	\$B2
D	\$C4	Q	\$D8	3	\$B3
E	\$C5	R	\$D9	4	\$B4
F	\$C6	S	\$DA	5	\$B5
G	\$C7	T	\$DB	6	\$B6
H	\$C8	U	\$DC	7	\$B7
I	\$C9	V	\$DD	8	\$B8
J	\$CA	W	\$DE	9	\$B9
K	\$CB	X	\$DF	SPACE	\$A0
L	\$CC	Y	\$E0	RETURN	\$BD
M	\$CD	Z	\$E1	ESC	\$B6

表 1—2 键盘码表 (ASCII)