

中学生课外阅读丛书

BASIC LLL

LLLLL

LLLLL

LLLLL

LLLLL 6502

北京市海淀区教师进修学校 主编

从 BASIC 语言 到 6502 汇编语言

机械工业出版社

中学生课外阅读丛书

从BASIC语言到6502汇编语言



机械工业出版社

本书是为学过BASIC语言的读者想进一步了解APPLE II的6502汇编语言而写的。本书是从汇编语言的基础知识开始介绍，写了汇编语言的操作、符号、命令和程序设计。还结合机器介绍了监控程序和特殊单元的作用。最后还提供了—些实用的汇编程序和简单介绍。本书为了便于读者自学，力求写得通俗易懂、活泼生动。总之，本书为准备制作计算机教学辅助软件的读者，提供一些工具以使自己的软件更加丰富多采。

本书适用于初、高中各年级学生。中、小学各学科教师，中学生家长和具有中等文化水平的青年职工以及自学青年。

中学生课外阅读丛书

从BASIC语言到 502 汇编语言

北京市海淀区教师进修学校 主编

★

责任编辑：鞠昌奎 版式设计：霍永明

封面设计：姚毅 责任校对：熊天荣

责任印制：郭炜

★

机械工业出版社出版（北京阜成门外百万庄阙里一号）

（北京市书刊出版业营业许可出字第117号）

机械工业出版社印刷厂印刷

机械工业出版社发行·新华书店经销

★

开本 787×1092¹/₃₂·印张 10³/₈·字数 226 千字

1988年11月北京第一版·1988年11月北京第一次印刷

印数 0,001—1,650·定价：2.45 元

★

ISBN 7-111-01017-5/G·70

序

知识的获得，能力的增长，智力的开拓和水平的提高，往往得益于课外，这是很多科学家、作家和文艺工作者的亲身体会。因为课内的讲授只能是分析、理解知识的内容和知识的结构，而要形成各种能力，则要靠大量的课外阅读。这就是本套丛书编写的目的之一。其次，这套丛书包括了从初中一年级起直至高中三年级的16个学科，它能使读者切实地掌握各学科的基础知识，培养、提高读者把握各学科的基本技能和技巧，有利于将来的工作，有利于初高中升学考试。这也是编写本套丛书的意图。中学是基础学习的阶段，如果能奠定坚实的知识基础，培养观察、想像、思维、动手等各方面的能力，对提高全民族文化素质也是有益的。这也是我们编写这套丛书的意愿。

这套丛书共16个学科，57册。其中，政治两册，初、高中语文各六册，初中数学六册，高中数学四册，初、高中英语各三册，初中物理两册、高中物理三册，初中化学一册、高中化学三册，初中中国地理、初中世界地理、高中地理各一册，中国历史、世界历史各两册，生物、动物、植物、生理卫生各一册，音乐、体育、美术各两册，计算机一册。

这套丛书充分体现了知识性、科学性和趣味性，内容充实，行文简洁，形式活泼，语言生动，读者从中可以得到爱国主义、国际主义、辩证唯物主义、历史唯物主义和美学教育。这套丛书除语文外，都是按照教学大纲和教材的要求，

以解决学习中的难点、重点为主线，介绍了本学科古今中外著名的专家学者，以及他们的故事轶闻，设计了多种形式的实验、练习以及解题的多种方法等等。语文中各种文体的文章也是按照教学大纲对每个年级每个学期的知识要求而选择的，内容丰富生动，情节曲折动人，并附有注释及分析。其中大部分文章是名家的新作，具有积极的思想内容和完美的艺术形式。

这套丛书的编写者，都是北京市海淀区有较高业务水平、有较丰富教学经验、有较强的写作能力的教师，其中大多数是中学的高级和一级教师，还有特级教师。编写班子阵容强大、实力雄厚，希望能为开辟学生的第二课堂做一些有益的工作。但限于时间和水平，书中内容有不当之处，敬请读者批评指正。

北京市海淀区教师进修学校

1988年2月

前 言

现在以 6502 芯片为中央处理器的各种中华学习机机型已经问世, 并已开始批量生产。进而需要大批的辅助教育教学的软件, 学生练习软件及游戏软件。制作这些软件仅用高级语言 BASIC 编写是不能达到预想效果的, 用 6502 汇编语言才能使软件的图形更丰富多彩, 动画速度变化更逼真。并能发出各种模拟声响, 可以唱歌, 甚至可以说话。为此, 我们写了这本书献给已经不满足用 BASIC 语言编程, 希望学习 6502 汇编语言的初中学生。当然它也是老师教授 6502 汇编语言的参考书。

6502 汇编语言有它的特点。它是一种符号语言, 指令都是英文缩写的助记符。并且要求深入了解机器的内部。指令比 BASIC 语言多, 但意思简单明确。并不十分难学难记。

本书第一章、第二章和第三章是讲解汇编语言的基本知识, 第四章与第五章是给读者介绍了监控程序和一些实际应用程序。这些程序可供读者在自己的软件中直接使用。

我们的愿望是尽量将本书写得通俗易懂、实用性强, 为发展我国的教学软件尽微薄之力。由于时间仓促, 水平有限, 书中有不当之处, 欢迎广大读者批评指正。

本书由庄燕民、贾丽娜、李彦强、刘永祚编写。

目 录

序

前言

第一章 学习6502汇编语言的初步知识	1
第一节 计算机中数的表示	1
一、数的进制	2
二、二进制数	3
三、二进制数与十进制数的互相转换	6
四、十六进制数与十进制数和二进制数之间的关系	11
五、机器数	17
六、二进制编码	24
第二节 APPLE II微型计算机的组成和简单工作过程	28
一、主机内存	29
二、6502中央处理器 (CPU)	32
三、简单工作过程	43
第三节 监控命令和小汇编的使用	46
一、监控命令简介	46
二、小汇编的使用	50
第二章 6502指令系统和寻址方式	53
第一节 6502指令的分类	54
一、数据传送指令	54
二、置标志位指令	59
三、算术和逻辑运算指令	60
四、移位指令	69



五、比较指令	72
六、转移指令	75
七、关于调用子程序指令	79
八、堆栈操作指令	80
九、中断指令及空操作指令	85
十、伪指令介绍	87
第二节 6502 的寻址方式	90
一、立即寻址	91
二、绝对寻址	92
三、零页寻址	92
四、累加器寻址	93
五、隐含寻址	93
六、间接寻址	94
七、相对寻址	95
八、使用 X 寄存器绝对变址	98
九、使用 Y 寄存器绝对变址	98
十、使用 X 寄存器零页变址	99
十一、使用 Y 寄存器零页变址	99
十二、先 X 变址后间接寻址	99
十三、先间接寻址后 Y 变址	101
第三节 编辑汇编程序命令的使用	103
一、EDASM 的启动方法	103
二、EDASM 编辑程序的使用	104
三、EDASM 汇编程序的使用	109
四、如何运行已汇编好的目标程序	110
第三章 6502 汇编语言程序设计	111
第一节 汇编语言源程序的语句格式	111
一、标号	112
二、操作码	112

三、操作数	113
四、注释	113
第二节 简单程序设计	114
第三节 循环程序设计	119
第四节 子程序	131
第五节 混合编程	143
一、机器语言的存储与调用	144
二、用机器语言子程序绘图	146
第四章 系统监控	155
第一节 监控程序的使用	156
第二节 系统监控中的子程序和入口地址	167
第三节 监控系统主程序及 21 条命令子程序入口	179
第四节 屏幕输出子程序	188
第五节 行输入子程序	201
第六节 监控系统使用的各 RAM 单元及 I/O 单元地址	213
一、监控系统使用的零页单元	213
二、监控系统使用的其他 RAM 单元	217
三、输入/输出单元 (I/O 单元)	218
第五章 应用汇编程序	222
第一节 文本汇编应用程序	222
第二节 图形汇编应用程序	259
第三节 声音汇编应用程序	295
附录	307
一、ASC I 码表	307
二、6502 汇编指令表	310
三、APPLE I 监控子程序入口地址表	314
四、APPLE I 高分辨率显示地址表	320

第一章 学习6502汇编语言的初步知识

通过学习 BASIC 语言，我们已经知道机器语言是计算机能够直接识别的唯一的语言。所有的高级语言（即算法语言）、汇编语言的执行都需经过一套专门机器语言程序的“翻译”，计算机才能懂得它的意义，才能执行。所以要想深入了解一个 BASIC 程序在机器内运行的细节，或更完满地处理 BASIC 程序，以至处理一些意料之外的情况，读一读系统程序（如磁盘操作系统、BASIC 解释程序、监控程序）将会受益非浅。而这些程序都是由一条一条汇编指令写出来的。这就需要了解这个指令系统。另外对 BASIC 语言程序中的某些部分，用一些机器语言子程序加以配合就可以实现 BASIC 语言某些无法实现的功能（如，若不用机器语言制成音乐程序，单用 BASIC 语言就无法实现音乐效果）。有些机器语言子程序可以使图形执行速度大大加快，动画效果更逼真，画面变化更丰富多彩，不熟悉指令系统就无法进行这类工作。

为此，本章介绍一些初步知识。

第一节 计算机中数的表示

在学习 BASIC 语言时，程序中所见到的数都是十进制的数，为了适应人们的计数习惯，为初学者设计的 BASIC 语言一般采用这种数制。但是计算机并不认识十进制数字，只认识二进制数，为此首先看看数的进制。以下所谈的数仅

就非负数而言。

一、数的进制

在日常生活中，人们都习惯使用十进制数计数法，可是在许多计量中还是会遇到一些其他进制的计数法。如：钟表计时，就是六十进制的（1分钟 = 60秒，1小时 = 60分），而计算角度时也用六十进制。又如十二进制应用在计量时间（十二个月为一年）、长度（十二英寸为一英尺）、数量（十二个为一打）等等。

每种进制都有一个进位的规则，如：十进制就是“逢十进一”，六十进制就是“逢六十进一”，同样十二进制就是“逢十二进一”。那么推广讲K进位制就是“逢K进一”。所谓“进”就是进上一位，即一个“单位”。这个“单位”是代表了比它低一级的K个“单位”的一个整体。每个“单位”对小数点前最低“单位”的大小比称为“权”。例如：十进制计数法中最小正整数的位是1，权是1，即 10^0 ；十位是十个1位组成的一个整体，它的权是10，即 10^1 ；百位是由十个十位组成的一个整体，它的权是100，即 10^2 ；当然千位是十个百位组成的一个整体，它的权是1000，即 10^3 等等。在六十进制中也是如此。例如：具体到时间上的六十进制，秒位是最小的位，权是1，即 60^0 。分位是由六十个秒位组成的一个整体，它的权是60，即 60^1 ；小时位是由六十个分位组成的一个整体，它的权是3600，即 60^2 。由此推广K进制的各位中最低位是1，权也是1，即 K^0 。比它高的位各位权逐次是 K^1 、 K^2 、 K^3 、 K^4 、 K^5 、……等。

在每个位上都要有数字，用以表示数的大小。那么数的表示根据不同的进位制，需要的数字个数也不同，这些数字称为数基。例如：十进制就需要十个数字0、1、2、3、

4、5、6、7、8、9 就可以了，而六十进制就需要从0、1、2、3、4、…、57、58、59，六十个数字。如果把数字写到位上而构成的数的大小，就等于这个数字乘上这位上的权再求和，即写成多项式的形成。例如：十进制数 1057 按位分开并用多项式表示：

$$1057 = 1 \times 10^3 + 0 \times 10^2 + 5 \times 10^1 + 7 \times 10^0$$

再例如：六十进制的时间表示 2 小时 50 分 29 秒，按位分开并用多项式表示：

$$2^{\circ} 50' 29'' = 2 \times 60^2 + 50 \times 60^1 + 29 \times 60^0$$

那么推广看来 K 进制数所需的数字共有 K 个，即从 0 到 K - 1。它要按位分开用多项式表示，就可以写成如下式子：

$$a_1 a_2 a_3 \cdots a_{n-1} a_n = a_1 K^{n-1} + a_2 K^{n-2} + a_3 K^{n-3} + \cdots + a_{n-1} K^1 + a_n K^0 \quad (a_i \text{ 都表示数字且 } a_i \in \{0, \cdots, K-1\})$$

在计算机中所能使用的数是二进制数，而在计算机所显示出的数又是十六进制的数。在深入研究中有些编码还采取了八进制数等。

下面先介绍二进制数。

二、二进制数

从前面的介绍中，可以看到 K 进制数的进位规则是“逢 K 进一”，由前面分析知道 K 进制数写成多项式形式。即：

$$a_1 a_2 \cdots a_{n-1} a_n = a_1 K^{n-1} + a_2 K^{n-2} + \cdots + a_{n-1} K^1 + a_n K^0$$

当 K = 2 时，就是二进制数。它的进位规则是“逢二进一”。它所使用的数基只有 0 和 1 两个数字。一个二进制数写成多项式的形式，各位的权就是以 2 为底的整数幂。写为：

$$a_1 a_2 \cdots a_n = a_1 2^{n-1} + a_2 2^{n-2} + \cdots + a_{n-1} 2^1 + a_n 2^0 \quad (a_i \in 0, 1)$$

如： $101001 = 2^5 + 2^3 + 2^0$

二进制数整数表示是这样的，小数表示也是如此，只不过小数点后面各位上的权都是 2 的负整指数幂。如小数点后面第一位的权是 2^{-1} ，即 $\frac{1}{2}$ 或 0.5。第二位是 2^{-2} ，即 $\frac{1}{4}$ 或 0.25。第三位是 2^{-3} ，即 $\frac{1}{8}$ 或 0.125 等等。用多项式形式表示：

$$0.a_1a_2\cdots a_m = a_12^{-1} + a_22^{-2} + \cdots + a_m2^{-m}$$

如 $0.101001 = 2^{-1} + 2^{-3} + 2^{-6}$

一个带小数的二进制数的多项式表示如下：

$$1101.1001 = 2^3 + 2^2 + 2^0 + 2^{-1} + 2^{-4}$$

注意整数部分各位权的幂指数，从小数点向左数位的个数少 1，而小数部分各位权的幂指数的绝对值与小数点后数位的个数相同。如整数部分从小数点向左数第四位，它的权的幂指数是 3，而小数部分从小数点向右数第四位，该位权的幂指数的绝对值是 4。

二进制数在读法上请注意 100 不能读“壹佰”，而只能读“壹零零”。从上面介绍可知它的各位的位权不同于十进制数的，100 在十进制数中 1 表示在百位上。而 100 在二进制数中 1 表示在 2^2 位，应该念“壹肆”，但不方便故念“壹零零”。为了加强对二进制数的概念，读法一定要正确。在写法上，要几种进制写在一起时应用括号括起，外边右下角用小字标明是几进位制的数，如： $(100)_2$ ， $(100)_{10}$ 等表示不同数制，尽管数的写法一样。

二进制数的运算法则基本同十进制的最基本的加减乘除，但二进制是逢二进一，而十进制是逢十进一。

二进制数各位运算法则。

加法: $1 + 1 = 10$ 乘法: $1 \times 1 = 1$

$1 + 0 = 1$ $1 \times 0 = 0$

$0 + 1 = 1$ $0 \times 1 = 0$

$0 + 0 = 0$ $0 \times 0 = 0$

例如: 加法 1011 减法 1001

$$\begin{array}{r} + 1001 \\ \hline 10100 \end{array}$$

$$\begin{array}{r} - 100 \\ \hline 101 \end{array}$$

乘法 110

除法 111

$$\begin{array}{r} \times 11 \\ \hline 110 \\ + 110 \\ \hline 10010 \end{array}$$

$$\begin{array}{r} 11 \overline{) 10101} \\ \underline{- 11} \\ 100 \\ \underline{- 11} \\ 11 \\ \underline{- 11} \\ 0 \end{array}$$

其它的一些运算不过是从这最基本的运算发展出来的。这样就构成了一个二进制数的完整系统。

计算机为什么要使用二进制数呢? 这是因为计算机中的主要部件是电气元件, 电气元件中有二种状态最容易实现, 如电路的通、断, 电位(电压)的高、低, 电容的充电与放电, 晶体管的导通与截止等等。而且这两种状态可以保持很稳定, 并且很容易实现对电路本身的控制。

在计算机里, 一般以电位的高低表示数, 高电位(或高电平)代表 1, 低电位(或低电平)代表 0。这样就为使用二进制数提供了硬件基础。如果要在计算机中使用十进制数就要找出十种不同的稳定状态, 这在硬件的实现上存在难以想象的困难。另外在计算机上进行运算, 二进制数各位算术运算加法法则和乘法法则各 4 条, 而十进制数各位加法达 100

条；乘法不算乘零还有 81 条。这也显示了计算机使用了二进制数的优越性。

三、二进制数与十进制数的互相转换

计算机对实现二进制数是非常有利的，但是人们对它是很不习惯的。人们希望将一个二进制数能折算成十进制数来读，或把一个十进制数变成一个二进制数来写。即实现二进制数与十进制数之间的转换。

简单的方法是可以查二进制数与十进制数之间的对照表，如下表：

十 进 制 数	二 进 制 数
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000

根据前面有关数制的介绍，和二进制数用各位以 2 的整数幂为权的多项式表示方法就可以很容易地实现从二进制数到十进制之间的转换，即把各位数字与各位位权相乘求和，就可得到这个二进制数的十进制数表示。

$$\begin{aligned} \text{例如：} (10110111)_2 &= 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^4 \\ &\quad + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \end{aligned}$$

$$\begin{aligned}
 &= 128 + 32 + 16 + 4 + 2 + 1 \\
 &= (183)_{10}
 \end{aligned}$$

二进制数 10110111 就这样实现了转化成十进制数 183, 同样用于小数也是如此。

$$\begin{aligned}
 \text{例如: } (101.1011)_2 &= 1 \times 2^2 + 1 \times 2^0 + 1 \times 2^{-1} \\
 &\quad + 1 \times 2^{-3} + 1 \times 2^{-4} \\
 &= 4 + 1 + \frac{1}{2} + \frac{1}{8} + \frac{1}{16} \\
 &= 5 + 0.5 + 0.125 + 0.0625 \\
 &= 5.6875
 \end{aligned}$$

反过来要把一个十进制数转化成二进制数该怎样做呢?
对于十进制整数我们可以用除 2 求余的办法。

例如: $(157)_{10} = (?)_2$

2	157	
	2 78 ... 1	↑
	2 39 ... 0	
	2 19 ... 1	
	2 9 ... 1	
	2 4 ... 1	
	2 2 ... 0	
	1 ... 0	

$$(157)_{10} = (10011101)_2$$

方法是短除法, 即用 2 连续去除要转化的这个十进制数, 不断求得一个商数和一个余数, 每次除得的余数不是 0 就是 1, 直到商数为 1 才算完。然后再从最后一个数值为 1 的商开始倒着往上记录下余数, 所得的这个二进制数即为所求。其实这个方法与各位数字与位权乘积求和写成多项式的形式是统一的。

$$\begin{aligned}
 \text{如上例: } (157)_{10} &= 78 \times 2 + 1 \\
 &= 39 \times 2 \times 2 + 1 \\
 &= (19 \times 2 + 1) \times 2 \times 2 + 1 \\
 &= ((9 \times 2 + 1) \times 2 + 1) \times 2 \times 2 \\
 &\quad + 1 \\
 &= (((4 \times 2 + 1) \times 2 + 1) \times 2 \\
 &\quad + 1) \times 2 \times 2 + 1 \\
 &= (((((2 \times 2 \times 2 + 1) \times 2 + 1) \\
 &\quad \times 2 + 1) \times 2 \times 2 + 1
 \end{aligned}$$

由此可见短除 2 取余数相当求二进制数各位上的数字。然后将此式写成以 2 为底指数幂的形式。取各位的数字即可得到该数二进制的表示。

$$\begin{aligned}
 &= 1 \times 2^7 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \\
 &= 1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \\
 &\quad \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
 &= (10011101)_2
 \end{aligned}$$

由此可以看出数的各种数制之间，以各位上的数字与该位位权相乘求和的多项式形式是互相转化的纽带。当然在十进制数不大时可以用逐次减权法逐位求之，因为二进制各位权是以 2 为底的整数指数的幂。即：

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	$2^{13}...$
1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192...

这样当遇到一个不太大的十进制数时就可以凭着经验和记忆用此数与某个位数比较，若数大于此位权且两数之差小于此位权，则在此位处写 1。所剩的差再与这个位权的下一个位