

118

7P313

Z316

汇编语言程序设计

张秉权

王宝琴

姜玉华

王宝华

孙承福

编



A0963813

吉林科学技术出版社

【吉】新登字 03 号

汇编语言程序设计

张秉权 王宝琴

姜玉华 王宝华 孙承福 编

责任编辑：珂 丽

封面设计：杨玉中

出版
发行

吉林科学技术出版社

787×1092 毫米 32 开本 15.25 印张

336,000 字

1996 年 8 月第 1 版 1996 年 8 月第 1 次印刷

印数 1—5000 册 定价：16.80 元

印刷 吉林省委党校印刷厂

ISBN 7-5384-1674-9/TP·50

前 言

本书是根据国家机械电子工业部教育司组织的“计算机及应用专业教材编委会”所审定的教学大纲编写而成，是机械电子工业部计算机及应用专业系列教材之一。

目前，IBM 公司微型计算机系统广泛用于教学、科研和管理等各个领域。在应用过程中可选用 PASCAL 语言、BASIC 语言及 C 语言等高级语言进行程序设计。但也有时根据编程问题的需要或有利于程序设计技巧的充分发挥，可以选用汇编语言进行程序设计。本书是根据专业教材编委会的要求，结合编者多年教学实践编写的。编写时我们力求通俗易懂、系统完整、条理清楚、重点突出。在书的后面我们又增加了几篇附录，为读者用汇编语言程序设计提供了很多方便。

本书主要由张秉权、王宝琴、娄玉华、王宝华、孙承福同志共同编写。其中娄玉华副教授、张秉权教授编写了第一章至第四章部分，张秉权教授、王宝琴副教授编写了第五章、第六章部分，王宝琴副教授、孙承福副教授、张秉权教授编写了第七章、第八章部分，王宝华副教授编写了第九章及附录部分。张秉权教授、王宝琴副教授担任本书的主编。成都电子科技大学的王正智教授担任本书的主审，王教授对本书的底稿进行了认真地审查，付出了辛勤的劳动，在此对王教授给予的帮助和指导表示衷心的感谢。

由于我们对《汇编语言程序设计》这本书编写经验不足，书中难免有不妥之处，请各位专家和读者给予指正。

编 者
1993 年

目 录

第一章 绪论	(1)
§ 1.1 概述	(1)
§ 1.2 为什么要学习汇编语言	(6)
第二章 预备知识	(8)
§ 2.1 进位计数制	(8)
§ 2.2 计算机中正负数的表示.....	(16)
§ 2.3 IBM-PC 系列机中数和符号的表示	(21)
第三章 8086/8088 的结构	(28)
§ 3.1 8086/8088 的内部结构和特点	(28)
§ 3.2 存储器.....	(30)
§ 3.3 中央处理器.....	(37)
§ 3.4 指令格式.....	(44)
§ 3.5 寻址方式.....	(48)
第四章 指令系统	(64)
§ 4.1 概述.....	(64)
§ 4.2 数据传送指令.....	(65)
§ 4.3 算术运算指令.....	(82)
§ 4.4 逻辑运算指令	(102)
§ 4.5 字符串操作指令	(116)
§ 4.6 程序控制指令	(129)
§ 4.7 处理器控制指令	(133)
第五章 程序设计的基本技术	(141)
§ 5.1 概述	(141)
§ 5.2 直接程序设计	(148)

§ 5.3	分支程序设计	(152)
§ 5.4	循环程序设计	(172)
第六章	宏汇编语言	(202)
§ 6.1	宏汇编语言的基本概念	(202)
§ 6.2	常量和变量	(210)
§ 6.3	标号	(219)
§ 6.4	表达式	(222)
§ 6.5	结构和记录	(232)
§ 6.6	程序的结构	(240)
§ 6.7	汇编命令	(244)
§ 6.8	宏指令的定义和调用	(256)
第七章	子程序	(278)
§ 7.1	子程序的概念	(278)
§ 7.2	堆栈	(283)
§ 7.3	调用指令和返回指令	(284)
§ 7.4	主程序与子程序之间的参数传递	(293)
§ 7.5	子程序嵌套和实例	(314)
§ 7.6	递归子程序和实例	(320)
第八章	中断技术	(329)
§ 8.1	中断概念	(329)
§ 8.2	中断向量结构	(335)
§ 8.3	中断处理过程和中断排队	(343)
§ 8.4	中断程序举例	(349)
第九章	系统调用	(361)
§ 9.1	概述	(361)
§ 9.2	字符设备系统调用	(365)
§ 9.3	文件管理系统调用	(371)

§ 9.4 汇编语言程序上机过程	(382)
附录 A 指令系统	(420)
附录 B MASM 伪操作命令系统表	(441)
附录 C PC-DOS 2.0 版的系统调用表	(452)
附录 D 汇编语言语法错误一览表	(475)

第一章 绪 论

通过 BASIC、PASCAL、FORTRAN 等编程语言的学习后，读者已对程序设计有所了解。从计算机的出现，发展到今天计算机已成为标志科学进步程度的重要学科，程序设计在计算机科学中占有非常重要的地位，所以我们有必要了解和掌握程序设计的有关知识。

§ 1.1 概 述

当我们在谈论或使用一台计算机时，而不是只限于看得见、摸得着的计算机本体。我们所关心的是这个计算机系统环境配置如何。这里所说的计算机系统包括两大部分，即由硬件部分和软件部分组成。

硬件是指构成计算机的本体装置与设备。这就是指那些集成电路元件、多层印刷电路板、打印机、终端显示设备等。我们把这些硬件有机地组合在一起，这就构成了计算机系统的本体。

软件是指可以写出的或可以记录的各种程序与数据的总称。软件又可分为系统软件和应用软件。系统软件是使计算机能正常、高效运行、方便用户使用所配备的程序与数据的总称，如操作系统、编译程序、数据库等。应用软件是用户根据自己的任务而编写的程序与数据的总称，如用于生产自动化的监测程序，一个单位的人事管理程序，根据给定的条件计算一条弹道的程序等。

学好程序设计这门课程，对今后学习、掌握、应用计算机系统软件，开发应用软件具有重要的意义。

一、什么是程序设计

如实现交换两个变量 VAR1、VAR2 的数据，我们可以用不同的程序设计语言编写出相应的程序。

用 BASIC 语言写出的程序为：

```
100  WORK=VAR1
110  VAR1=VAR2
120  VAR2=WORK
130  END
```

用汇编语言写出的程序为：

```
MOV  AX, VAR1
MOV  WORK, AX
MOV  AX, VAR2
MOV  VAR1, AX
MOV  AX, WORK
MOV  VAR2, AX
```



图 1-1 数据传送示意图

完成两个变量 VAR1、VAR2 的数据交换过程如图 1-1 所示。请读者思考，这里为什么要引进一个中间变量 WORK？如果两个变量之间直接传送数据，是否能够实现两个变量的数据交换任务。

由此例可见，为了完成某种特定任务，按照一定的规则编写的语句序列或指令序列称为程序。编写程序的过程称为程序设计。用于实现程序设计而使用的编程语言称为程序设计语言。在计算机发展过程中，人们为了进行有效的程序设计，世界上现在已有几百种程序设计语言供软件开发人员使用。

二、程序设计语言

程序设计语言比较多，按应用角度一般分为三种类型。

1. 面向过程的程序设计语言

面向过程的程序设计语言，如 FORTRAN、PASCAL、ALGOL、COBOL 等高级编程语言，在用这些语言编写程序时，它们都独立于计算机，我们将根据编程任务的要求，使用多个语句描述解题的具体过程。这种语言接近人们的习惯表达方式，所以这种语言易学、易懂、便于使用。用这种语言编写的程序送入计算机后并不能直接运行，用这种语言编写的程序称为源程序。源程序送到计算机之后，需要把源程序翻译成与其对应的机器指令程序，这个机器指令程序称为目标程序。实现翻译功能的这个专用程序称为编译程序。

2. 面向问题的程序设计语言

如用于数理统计、报表处理等方面的高级编程语言为面向问题的程序设计语言，用这种语言编写程序时，也都是独立于计算机的，我们不需要关心问题的解法和过程的描述。在把要求的问题送进计算机之后，计算机就可以自动地选择相应的处理方法，给出最后的处理结果。这类语言的代表有报表语言和判定语言等。

3. 面向机器的程序设计语言

我们把针对某一特定的计算机或某一类型的计算机，使用的编程语言称为面向机器的程序设计语言。

如实现交换两个变量中的数据，当用机器指令编写程序时（即用机器语言编写程序），应该确定内存分配区，把存放程序的代码段选定为 00000H 至 0FFFFH 单元，把存放数据的数据段选定为 10000H 至 1FFFFH 单元。完成交换数据的 8086/8088 机器语言程序为（十六进制数形式）：

存放地址	操作码与操作数			说明
00200	A1	10	00	(VAR1) → AX
00203	A3	14	00	(AX) ⇒ WORK
00206	A1	12	00	(VAR2) ⇒ AX
00209	A3	10	00	(AX) → VAR1
0020C	A1	14	00	(WORK) ⇒ AX
0020E	A3	12	00	(AX) ⇒ VAR2

程序与数据在机器内存分配情况如图 1-2 所示。

由上面可见，用机器语言编写程序是一项比较困难和复杂的工作。编程人员必须熟悉机器语言，编写出的程序不易阅读，也难于修改和调试，编程效率也低。为了克服机器语言编程的这些不足，现在已多用汇编语言进行程序设计。

汇编语言也是面向机器的程序设计语言。在编写程序时，指令的三要素（内存分配地址、操作码及操作数）用记忆符号表示的面向机器的程序设计语言称为汇编语言。如交换两个变量的 8086/8088 机器汇编语言程序可写为：

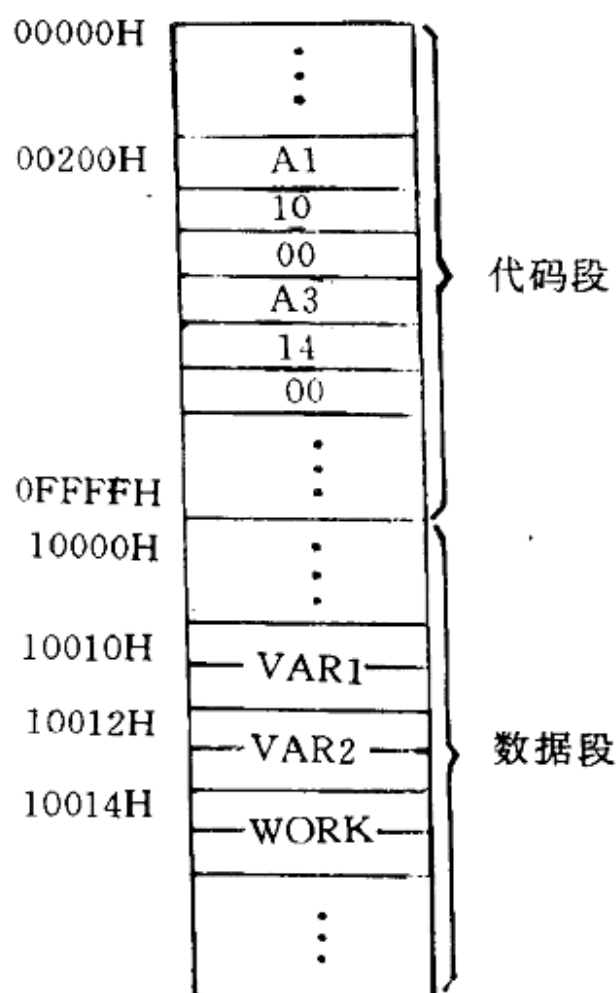


图 1 2 内存分配示意图

分 配 地 址	操 作 码	操 作 数
PROG:	ORG	0200H
	MOV	AX, VAR1
	MOV	WORK, AX
	MOV	AX, VAR2
	MOV	VAR1, AX
	MOV	AX, WORK
	MOV	VAR2, AX
	HLT	

用汇编语言编写的程序称为汇编语言的源程序。源程序送入计算机之后,必须把它翻译成与其对应的机器语言程序,

这个机器语言程序称为目标程序。完成翻译功能的程序称为汇编程序。汇编程序运行时完成翻译的过程称为汇编过程，工作过程如图 1-3 所示。

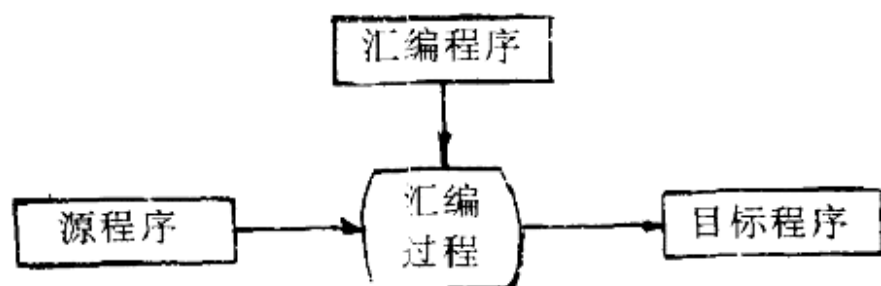


图 1-3 汇编过程示意图

§ 1.2 为什么要学习汇编语言

通过以前和上节学习之后，读者会感到用高级语言编写程序，易学而且也比较简单。但是，由于计算机不能直接执行源程序，需要翻译成相应的机器语言程序，执行目标程序之后才给出处理结果。在翻译过程中，生成的目标程序往往都比较大，占用机器内存空间比较大，执行这个程序的时间比较长。也就是说，生成的目标程序质量比较差，一般来说这种目标程序的质量相当于中等程度手编程序员编写出的程序水平。用汇编语言编写的程序，它与机器语言程序彼此对应，不会生成一些多余指令，这样程序占用机器内存空间少，执行这个程序的速度也快。

用汇编语言编写程序，要求我们对计算机硬件系统、计算机指令系统比较熟悉，这样可以充分发挥整个计算机系统的功能，提高编程质量。比较好地利用多个寄存器、标志测试及一些特殊指令，这样可以充分发挥程序设计的各种技巧，编出高水平的程序。

我们常用的一些微机系统，具有比较多的软件开发程序供用户使用，如监控程序、操作系统、编辑程序、汇编程序、调试程序及汇编语言程序库等。有这些软件支持使汇编语言编程工作变得比较方便和灵活，使原来比较复杂和困难的工作变得比较简单和容易得多了。

在计算机软件研究、开发工作中，有时还经常应用汇编语言编程。如要求程序质量较高的操作系统、编译程序、汇编程序、程序库等系统软件的编程工作。还有一些应用软件，如要求处理实时性较强的数据采集程序、测试程序、控制程序、数据传送程序等，也多用汇编语言进行编程。

学习汇编语言程序设计，为读者以后学习和应用计算机打下一定的基础。在计算机系统中，硬件是软件的基础，软件是硬件的应用，这门课程有机地把软硬件知识结合起来，还可以帮助我们加深理解计算机的组成和工作原理。这是因为汇编语言要涉及到计算机中央处理器（CPU）、存储器和指令系统等有关内容，这样对计算机的内部结构、以及各部分之间的联系，读者会理解、掌握得更深入和全面。

第二章 预备知识

在用汇编语言进行程序设计时,我们要经常使用数据,比如有数据的存放与数据的运算等。计算机是由能够表示各种不同状态的电磁元器件组成,那么数据在计算机中是如何表示的呢?

在这章里,我们将从大家非常熟悉的十进制数开始,进而引入介绍以后用到的数制与码制。

§ 2.1 进位计数制

我们经常使用的十进制数,它有 0、1、2、3、4、5、6、7、8、9 十个数字,计算时按“逢十进一”的法则处理。如四位数 1245 可表示为:

$$1245 = 1 \times 10^3 + 2 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$$

对于任意一个多位正数我们可以表示为:

$$\begin{aligned} S &= k_n \times 10^n + k_{n-1} \times 10^{n-1} + \cdots + k_1 \times 10^1 + k_0 \times 10^0 \\ &\quad + \cdots + k_{-m} \times 10^{-m} \\ &= \sum_{i=-m}^n k_i \times 10^i \end{aligned}$$

(在这里的 m, n 为正整数, 10 为基数, k_i 为数字 0~9 中的一个。)

在十进制数中,它的基数为十,逢十进一。基数就是某种进位制数中可能用到的数字个数。在我们日常生活中也有非十进制的计数制。比如时间六十秒为一分,六十分为一小

时，这是六十进制。一周为七天，这是七进制。下面我们将介绍计算机中常用的数制与码制。

一、二进制数

在计算机中，我们现在不使用十进制数，而使用二进制数。与十进制数相比，二进制数的数字只有0、1两个，基数为2，逢二进一。一个二进制数10101表示为：

$$S_2 = (10101)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

对于任意一个多位正数我们可以表示为：

$$S = k_n \times 2^n + k_{n-1} \times 2^{n-1} + \cdots + k_0 \times 2^0 + k_{-1} \times 2^{-1} + \cdots + k_{-m} \times 2^{-m}$$

在计算机中使用二进制数有很多优点：

①二进制数只有两个数字0、1，计算机用具有两个稳定状态的元件表示比较容易，比如用晶体管的导通与截止，用电位高与低分别表示“1”与“0”。这样就大大简化了计算机的设计与制造的复杂性，可以节省大量的设备。

②二进制数算术运算比较简单。如果是十进制数运算这需要用五十多个计算公式，这样计算机设备就会庞大，控制线路也很复杂。二进制数相加求和，相乘求积比较容易，即有：

相加为 $0+0=0$

$$1+0=0+1=1$$

$$1+1=10$$

相乘为 $0 \times 0 = 0$

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

$$1 \times 1 = 1$$

③二进制数可以很方便地进行逻辑代数运算，这就为计

计算机的逻辑运算提供了便利条件，如有：

与运算为 $0 \wedge 0 = 0$ （用 $\wedge$ 表示与运算符）

$$0 \wedge 1 = 1 \wedge 0 = 0$$

$$1 \wedge 1 = 1$$

或运算为 $0 \vee 0 = 0$ （用 $\vee$ 表示或运算符）

$$0 \vee 1 = 1 \vee 0 = 1$$

$$1 \vee 1 = 1$$

因为二进制数有这些非常重要的优点，所以在计算机中现在正被广泛地使用着。由于二进制数书写和口读不方便，所以有时也用八进制数、十六进制数和 BCD 码等其他形式的数制，来弥补二进制数的不足之处。

二、八进制数

八进制数有八个不同的数字 0、1、2、3、4、5、6、7，基数为八，逢八进一。

因为有 $2^3 = 8$ ，所以一位八进制数相当于 3 位二进制数，这样为八进制数与二进制数之间的转换带来很多方便。对一个八进制数来说，整数部分由右向左（由低位向高位），每一个八进制数都用 3 位二进制数写出来，小数部分由左向右（由高位向低位），每一个八进制数都用 3 位二进制数写出来。如有八进制数 135.246 转换为二进制数有：

$$\begin{array}{ccc} 1 & 3 & 5 \\ \downarrow & \downarrow & \downarrow \\ 001 & 011 & 101 \end{array} . \quad \begin{array}{ccc} 2 & 4 & 6 \\ \downarrow & \downarrow & \downarrow \\ 010 & 100 & 110 \end{array}$$

$$\begin{aligned} (135.246)_8 &= (001011101.010100110)_2 \\ &= (1011101.01010011)_2 \end{aligned}$$

二进制数转换为八进制数，对一个二进制数来说，整数部分由右向左，每 3 位二进制数用一位八进制数写出来，小数部分由左向右，每 3 位二进制数，用一位八进制数写出来。如有二进制数 $(11001.0101)_2$ 转换为八进制数：

$$\begin{array}{ccccccc} \underline{11001} . \underline{0101} & _2 = & \underline{011} & \underline{001} . & \underline{010} & \underline{100} & _2 \\ & & \downarrow & \downarrow & \downarrow & \downarrow & \\ & & (3 & 1 & . & 2 & 4)_8 \end{array}$$

所以有

$$(11001.0101)_2 = (31.24)_8$$

由上例可见，用八进制比用二进制书写要简短，口读时也较方便。

三、二进制数之间转换

在不同数制的两个数之间转换中，它们之间转换具有一定的共同规律。我们想通过介绍二、十进制数之间转换原理和方法，使读者学会和掌握其他任意数制之间的转换方法。

(一) 十进制数转换为二进制数

在计算机中常用的数，可以有整数，也可以有小数。在数制转换时，对它们的处理方法有所不同，所以我们将分别给予介绍。只要学会了对整数和小数的转换方法，对给定的任意一个数的转换就迎刃而解了。

1. 十进制整数转换为二进制整数

不同数制的两个数进行转换，它们的数值应该相等。如把一个十进制数 201 转换为二进制数时有：

$$(201)_{10} = (k_n k_{n-1} \cdots k_1 k_0)_2$$

(这里 $k_n, k_{n-1}, \cdots, k_1, k_0$ 是二进制数字 0、1 其中之一。)