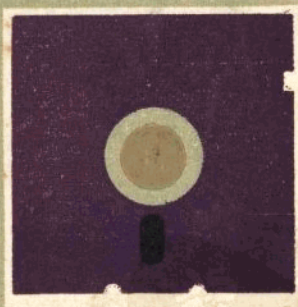


宏汇编语言程序设计

(IBM-PC〔长城 0520〕系统)

曹加恒 曾庆音 林 青 编著



武汉大学出版社

PDG

内 容 简 介

本书以16位微型计算机IBM-PC〔长城0520〕系统为背景，较全面地论述了宏汇编语言程序设计的基本原理、基本概念和程序设计的常用方法与技术，以及编辑程序和调试程序等。为了便于读者理解和掌握有关的内容，书中还有大量的程序实例及习题。

本书理论联系实际，技术先进，内容丰富。可作为高等学校计算机软、硬件专业的教材或参考书，也可以供广大从事微型计算机科研、教学和应用开发的技术人员自学或参考。

前 言

本书是根据国家教委所属高等院校计算机专业四年制教学大纲的要求编著的。

宏汇编语言程序设计是计算机软件系统的基础，是计算机软硬件专业的一门必修课程，对于训练学生掌握程序设计技术，从事软件开发和应用具有重要的作用。

宏汇编语言不仅包含一般汇编语言的功能，而且采用了高级语言使用的数据结构，是一种接近高级语言的汇编语言。例如IBM-PC [长城0520]机宏汇编语言，提供了指令中有多个变量时，程序中只需指出源操作数及目的操作数的能力；提供了记录、结构和字符串操作的使用方法；具有宏处理、条件汇编及磁盘操作系统DOS功能调用等多种功能；并支持8087浮点运算处理器进行工作，程序的开发以及调试手段也比较完善。因而宏汇编语言是一种更高级的汇编语言。

全书分8章，主要内容有：程序设计基础知识，宏汇编语言和宏汇编程序，基本程序设计，子程序设计，程序设计的其它方法与技术，输入输出和中断程序设计，程序设计综合实例，以及编辑程序和调试程序等。并通过大量的程序实例和列表说明了如何利用宏汇编语言从事软件开发。

本书特点是：吸收多年来从事宏汇编语言程序设计教学、科研和教学法研究的成果，具有新的体系、新的理论观点、新的材料和方法；采用分层叙述，由浅入深，学用结合，概念清楚。

本书作为教材（曹加恒 编著）已在武汉大学及兄弟院校、研究所等单位多次使用，受到校内外读者的一致好评。该教材被评为武汉大学（1978~1987）首届优秀自编教材并获得优秀教材奖。现经过进

一步修改和完善，奉献给广大的读者。

限于编者水平，书中如有错误和不妥之处，敬请读者不吝批评指正。

感谢为本书作过有益工作的陈远、王亚明和毛新宇等同志。

曹加恒 曾庆音 林青
武汉大学计算机科学系
1987年10月

目 录

第一章 程序设计基础知识	(1)
1.1 为什么要用宏汇编语言编写程序	(1)
1.2 数据表示	(5)
1.3 存储器 和堆栈	(9)
1.4 8088CPU的基本结构	(14)
1.5 指令格式及寻址方式	(20)
1.6 程序指令的执行过程	(32)
习题 一	(37)
第二章 宏汇编语言和宏汇编程序	(38)
2.1 概述	(38)
2.2 宏汇编语言的基本语法	(39)
2.3 8088指令系统的汇编表示	(54)
2.4 常用的汇编控制语句	(60)
2.5 DOS系统功能调用	(70)
2.6 宏汇编过程	(73)
2.7 行编辑程序EDLIN	(78)
习题 二	(86)
第三章 基本程序设计	(88)
3.1 程序设计引论	(88)
3.2 顺序程序	(93)
3.3 分支程序设计	(109)
3.4 循环程序设计	(115)
习题 三	(129)

第四章 子程序设计	(132)
4.1 子程序概念	(132)
4.2 调用、返回和过程定义	(133)
4.3 子程序的一般设计方法	(137)
4.4 递归子程序	(147)
4.5 阅读程序	(151)
4.6 调试程序DEBUG	(161)
习题 四	(171)
第五章 宏功能程序设计	(174)
5.1 宏指令程序设计	(174)
5.2 重复块语句的程序设计	(184)
5.3 条件汇编和宏列表控制语句	(190)
习题 五	(199)
第六章 程序设计的其它方法与技术	(202)
6.1 字符串操作程序设计	(202)
6.2 记录和结构的使用方法	(216)
6.3 与位置无关程序和动态程序再定位	(224)
6.4 程序的连接技术	(228)
习题 六	(238)
第七章 输入输出和中断程序设计	(241)
7.1 基本概念	(241)
7.2 输入输出指令和数据传送的方式	(244)
7.3 中断和中断处理程序设计	(249)
7.4 ROM BIOS中断调用	(264)
7.5 磁盘文件管理程序设计	(266)
习题 七	(280)
第八章 程序设计综合实例	(282)
8.1 加密和解密	(282)

8.2	开窗口程序	(286)
8.3	声音和图形	(291)
8.4	宏汇编语言程序与高级语言程序的连接	(300)
8.5	异步通讯	(306)
习题 八		(309)
附录		(311)
附录一	宏汇编语言8088机器指令表	(311)
附录二	错误信息	(316)
附录三	常用BIOS子程序的功能 及其调用的参数	(321)
附录四	DOS的软中断与系统功能调用	(325)
附录五	IBM - PC / AT计算机简单介绍	(332)

第一章 程序设计基础知识

宏汇编语言程序设计与具体计算机的结构有关。要想编制好的程序，必须了解机器的有关特性。本章以IBM-PC [长城0520]为背景机，讨论数据在计算机中的表示，存储器 and 堆栈，8088的CPU结构，指令格式及寻址方式和程序在机器中的执行过程。通过这些特性的学习，掌握程序设计的基础知识。

1.1 为什么要用宏汇编语言编写程序

从高层次观点（面向问题）看用高级语言编写程序对解决大量实际问题很方便，从低层次观点（面向机器）看用宏汇编语言写程序却能充分发挥计算机特性的威力，并编出高效率的程序。

1.1.1 什么是宏汇编语言

例如 计算 $3+2=?$ 让计算机求两数之和，要执行传送数、相加等有关的命令。每一种命令是计算机的一种基本操作，每一种基本操作通常对应一条指令。

· 指令 指令是指指挥计算机完成某一基本操作的命令，一般包含有操作码和操作数两部分（有些操作码甚至隐含操作数）。操作码指出机器执行什么样的操作，操作数表示参加的数本身或操作数所在的位置。

· 程序 程序是指令的集合。二进制编码的指令序列是机器语言程序。例如在8088微处理器的情况下，计算 $3+2$ 的程序用十六进制书写的指令序列是：

操作码	操作数	说 明
B0	03	传送指令：将3送到累加器AL中
04	02	加法指令：将2加到AL的内容上

显然，这种机器指令难记忆，编程不便，程序难读。如果用符号MOV AL代替上述传送指令操作码B0，3代替03，类似地，用符号ADD AL代替加法指令操作码04，2代替02，注解前加上分号“；”表示说明，则符号指令序列（符号程序）是：

操作符	目的操作数，源操作数	； 注 解
MOV	AL, 3	； 3→AL
ADD	AL, 2	； (AL)+2→AL

其中，操作符用的一些记忆符是指令功能的英文词或英文词的缩写（如MOV是传统指令助记符MOVE的缩写，加法用ADD等）。源操作数指数据的来源位置，目的操作数指存放结果的目的地位置，目的操作数有时也可以兼作另一个参加操作的数据位置。注解栏中的箭头→表示左边的数或者结果送到右边的存放位置。注意：(AL)+2表示累加器AL的内容加2，也可以写成AL+2。

· 汇编语言 能用记忆符代替指令的操作码，用数字或符号代替指令中操作数或操作数存放位置的一种程序设计语言，称为汇编语言。这种汇编语言实质上是机器语言的一种符号表示，程序不必为代码及数据存放在机器中的实际位置进行具体的安排。汇编程序（Assembler）把用汇编语言编写的程序（称为汇编语言源程序）汇编成机器语言程序（即目标代码或目标程序）。汇编语言虽然较之机器语言在阅读记忆或编写方面前进了一大步，但是对描述任务、编程设计仍感不便，于是产生了具有机器语言优点，而又较好地面向问题的语言，即宏汇编语言。

· 宏汇编语言 宏汇编语言不仅包含一般汇编语言的功能，而且采用了高级语言使用的数据结构，是一种接近高级语言的汇编语言。例如美国MICROSOFT公司开发的PC机宏汇编语言，提供了指令中

有多个变量时，程序中只需指出源操作数及目的操作数的能力；提供了记录、结构和字符串操作；具有宏处理、条件汇编及磁盘操作系统DOS功能调用等多种功能；并支持8087浮点运算处理器进行工作，程序的开发以及调试手段也比较完善。因而宏汇编语言是一种更高级的汇编语言。

1. 1. 2 宏汇编语言的用途

由于宏汇编语言仍属于汇编语言的范畴，因此语言中基本指令和机器指令是一一对应的，它随机器的不同而不同。一般情况下，不同型号的计算机其汇编语言是不一样的。Z-80型号微型机的汇编语言在8086/8088型号的微型机上就不能被执行，反之亦然。因而要学会一种汇编语言，就要了解该机器的指令系统、硬件结构、寻址方式、汇编语言的语法规则等等，学习比较困难。另外用汇编语言编写程序也比较复杂，编程速度慢。这些都是汇编语言的缺点。

高级语言可以避免汇编语言的这些缺点，常用高级语言有BASIC、FORTRAN、COBOL、PASCAL、C语言等。这些高级语言都是面向过程的，用它们来编写程序同我们通常的解题过程很相似，并且高级语言不依赖于具体的计算机结构和指令系统，因而比较容易学习和使用，描述任务较为方便，编程效率较高，编写的程序具有可移植性。

高级语言虽然可以克服汇编语言的缺点，节省软件的开发时间，但是用高级语言编写的程序执行速度较慢，控制也不灵活。而汇编语言却具有内存容量省、执行速度快、程序运行效率高、以及可直接使用计算机的全部资源等优点，这些都是高级语言所望尘莫及的长处。

宏汇编语言允许程序设计员直接使用计算机内部集成电路芯片的许多特性。例如：寄存器、CPU的标志位和一些特殊的指令等，可以编写十分精炼、灵活而又高效率的程序。而且由于微型计算机的特点，它大量地应用于实时控制上，这时如果用高级语言编程序，就不能充分利用微机接口技术和按位处理的功能，汇编语言则可以充分利用这

些技术。在磁盘数据存取方面，高级语言是以文件管理的形式存取，宏汇编语言则可直接存取磁盘设备上某一扇区的信息，而且可以编写磁盘数据加密和解密程序。在机器人的控制级里，用汇编语言编写程序让计算机直接控制机器人的控制机。可见，掌握了机器和汇编语言技术，就能驾驭机器，充分发挥机器的潜在威力。

当前大多数计算机（特别是微型机）的用户程序使用汇编语言，许多用户需要用宏汇编语言编写程序，因而宏汇编语言在系统软件、应用软件和数据库管理软件〔例如常用的标准子程序、代码转换程序、系统程序（尤其是核心部分）、硬件诊断程序、检查程序、数据库管理程序等〕得到广泛的应用。

由于宏汇编语言兼有机器语言和高级语言的一些主要优点，因而在软件开发中十分有用。宏汇编语言的优点如此突出，特别是在许多场合下，高级语言还不能代替汇编语言，所以宏汇编语言始终是程序设计的主要手段之一。

1.1.3 一个简单的宏汇编语言程序实例

宏汇编语言程序一般由几个段构成。每个段都以SEGMENT语句开始，以ENDS语句结束。本例程序共有三个段，第一段称为DATA数据段，存放显示输出使用的汉字字符串数据（如语句2~3所示）；第二段为CODE代码段，语句6~8为初始化，语句9~11显示字符串输出，语句12~13自动返回操作系统DOS；第三段为STACK堆栈段（语句15~17）。至于这些段在计算机中的具体存贮位置，将由宏汇编程序（翻译程序）和其它实用程序负责安排，这个程序只有在汉字操作系统CC-DOS下运行才能显示汉字。

这个例子仅供初学者见识一个真实的可上机通过的宏汇编语言程序，只要从功能上有所了解就可以了。

程序如下：

```

DATA    SEGMENT                                     ; 数据段
HANGW DB    '亲爱的读者，欢迎您学习'
        DB    '宏汇编语言程序设计！'，'$'
DATA    ENDS
CODE    SEGMENT                                     ; 代码段
        ASSUME    CS, CODE, DS, DATA
START:  MOV     AX, DATA                           ; 初始化
        MOV     DS, AX
        MOV     DX, OFFSET HANGW                   ; 显示字符串
        MOV     AH, 9
        INT     21H
        MOV     AX, 4C00H                           ; 自动返回 DOS
        INT     21H
CODE    ENDS
STACK   SEGMENT STACK                               ; 堆栈段
        DB      256 DUP ( ? )
STACK   ENDS
        END      START                             ; 程序块终止

```

程序正确执行的结果是：

亲爱的读者，欢迎您学习宏汇编语言程序设计！

1.2 数据表示

数在计算机中是用二进制表示的，一串二进制位可表示数字字母、标点符号和其它有用的信息。数据的四种形式如下：

1. 二进制数（或整数）
2. 浮点数（或实数）
3. 二进制编码的十进制数（BCD）（或十进制数）
4. 字符编码（或字符串）

本节除浮点数外，将讨论数制和字符编码。

1. 2. 1 二进制

数是抽象存在的，它可以用种种规则和模型来表示。通常非负整数由挑选的基数 X 和一串数字如 $a_n a_{n-1} \dots a_1 a_0$ 来表示。即

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

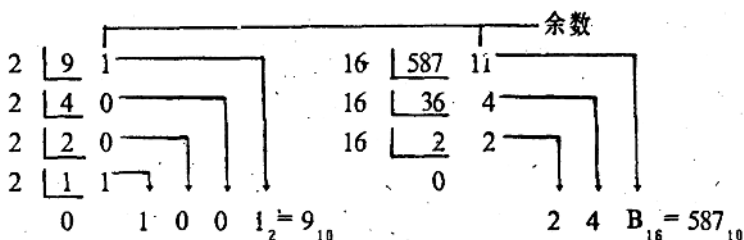
例如二进制数 $10110_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2 + 0 = 22_{10}$

而十六进制数 $51A_{16} = 5 \times 16^2 + 1 \times 16 + 10 = 1306_{10}$

这些转换也能用Horner's规则，即

$$\begin{aligned} & a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \\ &= ((\dots (a_n x + a_{n-1}) x \dots) x + a_1) x + a_0 \end{aligned}$$

应用Horner's规则，并使用 X 作除数，一个十进制数能根据有关形式转换成基数为 X 的数。



十六进制与二进制互换按四个二进制位为一组进行，类似地二进制和八进制互换按三个二进制位为一组实现转换。由于十六进制数比二进制数书写简洁，所以本书的数多用十六进制数且带后缀H表示。

二进制数在计算机中进行算术运算的时候，负数一般用补码表示。一个 n 位二进制数能表示数的范围是 -2^{n-1} 到 $2^{n-1}-1$ 。对于 $n=16$ ，整数的范围从 -32768 到 32767 ，也可以写成：

$$1000000000000000 = 8000$$

$$1111111111111111 = FFFF$$

$$0111111111111111 = 7FFF$$

对于 $n=8$ 时，数的范围在 -2^7 到 2^7-1 之间（ $-128\sim 127$ ）。

1. 2. 2 BCD形式

BCD是二进制编码的十进制数，它的每一位用四位二进制编码表示。BCD有二种基本类型：压缩BCD和未压缩BCD。压缩BCD的一串十进制数字按4个二进制位为一组的顺序存放，例如508的压缩BCD形式是： 0101 0000 1000

未压缩BCD的每个数字在每8个二进制位（字节）的低四位，而高四位的内容是什么并不重要（这里用U表示），采用这种形式508的表示是： UUUU 0101 UUUU 0000 UUUU 1000

BCD和二进制负整数的补码不同，在系统硬件没有特别指出的情况下，带有BCD符号数是由程序员挑选相应的转换来决定的。负整数能使用符号量形式或10的补码形式存放数据，如果选用符号量形式，则正负号由两个未使用的4个二进制位组合表示，即1100指示正，1101指示负，符号可以放在数字串之前或之后。

1. 2. 3 字符编码

计算机使用的美国标准信息交换码（ASCII）和扩展二进制编码的十进制交换码（EBCDIC）是两种主要的字符编码。ASCII码表如表1.1所示。

字符编码是计算机外部设备实现输入输出的基本手段。当按下终端的一个键时，就产生相应的ASCII码并被送到计算机。反之，如果计算机送一个ASCII码位串到终端，则终端一定翻译这个位串并相应作出反应。注意：不是所有ASCII字符都能被打印。有些字符指示终端空格（20H）、退格BS、换行LF（0AH或10）、回车CR（0DH或13）等等。另外有些ASCII字符包括几个打印和控制字符，如正文结束（ETX），传输结束（EOT），它们是在传送和存贮数据时作为标记使用。

ASCII字符串送入计算机的例子

DE,

JP -5

它与下述用十六进制给出位组合串是一致的。

D	E	,	回车	换行	J	P	空格	-	5
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
44	45	2C	0D	0A	4A	50	20	2D	35

字母“E”——→终端——→计算机

——→ 0 1 0 0 0 1 0 1 ←——

发送“E”的ASCII码并存放计算机中 第一个传送

表 1.1 ASCII码表

十六进制	十	六	进	制	高	位		
低位	0	1	2	3	4	5	6	7
0	NUL	DEL	空	0	0	P	\	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	*	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	:	K	[	k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

字符编码需要的二进制位数叫做字符长度，长度为 n 的编码可定义 2^n 种字符。如表1.1，7位编码的ASCII码能表示128种字符，每个字符用十六进制数的高位和低位组合编码。另外在每一个字符码上还附加一位奇偶校验位，产生8位传送，因此字符内部按8个二进制位存贮。

字符在编码中的数字集合称为编码集。数字编码是在数字上加十六进数30构成ASCII码。例如数字0的ASCII码是30，9的ASCII码是39等等，它允许机器直接按数字ASCII码进行运算。

当计算机接收数字不加修改而存放的时候，数字在机器内用未压缩BCD形式存放。如果机器舍去数字ASCII码的高4位，将每位数字的低四位按顺序拼在一起存放，则这个数是压缩BCD形式。或者计算机将ASCII数字转换成二进制存放。数究竟按何种形式存贮，依赖于程序的执行情况。未压缩BCD数不需要转换可直接进行输入输出操作，但是占用较多的存贮空间。无符号整数3986以未压缩BCD、压缩BCD、二进制形式存放，分别需要32、16、13个二进制位。另外，计算机按二进制数运算比用BCD数进行算术运算花的时间要少些。

1.3 存贮器和堆栈

计算机系统包括两大部分：硬件和软件。硬件是指电路、机壳、电缆等电子器件和机械装置，软件是计算机直接执行任务的程序集合。现在以微机系统为例（同其它任何计算机系统一样）简要说明计算机的基本结构和特性，重点叙述堆栈和存贮器。

1.3.1 什么是计算机结构

计算机结构是指计算机主要部件的一般设计布局、基本特性和部件之间如何相互连接。图1.1描述了一个典型的微型计算机系统结构。图中的部件有中央处理机（CPU）、主存贮器、输入输出（I/O）系统，总线控制部件和系统总线。

从信息（包括数据和程序）处理角度来看，信息由输入设备进入计算机，再经CPU加工处理，最后由输出设备送出处理结果。中间经过系统总线和总线控制部件流入或流出信息，或临时保存在主存贮器中。微型计算机的CPU（Central Processing Unit）是一个微处理器部件，用来有步骤地控制和执行指令，指挥机器工作。它包括算术逻辑部件（ALU）、通用寄存器、其它寄存器及控制部件等。总线控制部件可以和CPU合在一起，也可以分开，它是系统总线和CPU的接口部件。

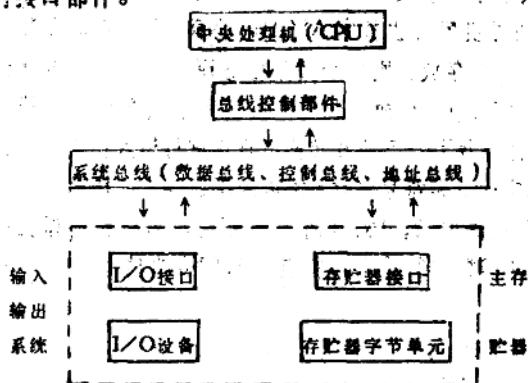


图1.1 微处理机基本结构

所谓接口是指连接系统总线到设备的电路。输入输出系统包括输入输出设备和I/O接口。I/O设备（外部设备或输入输出装置）是计算机与外界（包括人）进行通讯的桥梁和通路。I/O接口连接外部设备到系统总线上，控制计算机与I/O设备通讯。设立I/O接口的目的是适应各种不相同的外部设备在系统总线上转换标准信号的需要。

系统总线连接CPU到存贮器和I/O设备，是计算机各主要部件共享的连接线。它包括传送信息的数据线，信息从什么地方来或者到什么地方去的地址线，以及调整总线操作的控制线。这样，五个部件互连就组成了计算机系统的结构。