

///

高校计算机教学系列教材

微机原理与汇编语言实验指导

赵梅 编著



北京航空航天大学出版社

高校计算机教学系列教材

微机原理与汇编语言 实验指导

赵 梅 编著

北京航空航天大学出版社

内容简介

本书是“微机原理与应用”和“汇编语言程序设计”课程的实验教材,分为软件和硬件两个部分。全书共分六章。第一部分是软件实验,包括1~4章,有10个实验。每个实验有两个实验内容。第一章介绍汇编语言上机过程,常用软件EDIT、DEBUG的使用及命令;第2章为初级程序实验;第3章为程序基本结构练习;第4章为输入/输出程序实验及演示程序。第二部分是硬件实验,包括5~6章,有18个实验。第5章介绍TPC-H微型机接口实验系统;第6章是硬件电路实验。全书有软硬件实验28个,从微型机的基本组成电路、指令系统到汇编语言的程序设计实验由浅入深。书末配有常用DOS系统功能调用和BIOS中断调用表,方便查询。本书适于作高等院校本、专科的实验指导。

图书在版编目(CIP)数据

微机原理与汇编语言实验指导/赵梅编著. —北京:
北京航空航天大学出版社,2003.12

ISBN 7-81077-400-X

I. 微… II. 赵… III. ①微型计算机—基础理论
—高等学校—教学参考资料②汇编语言—程序设计—高
等学校—教学参考资料 IV. ①TP36②TP313

中国版本图书馆CIP数据核字(2003)第094198号

微机原理与汇编语言实验指导

赵梅 编著

责任编辑 韩文礼

责任校对 戚爽

*

北京航空航天大学出版社出版发行

北京市海淀区学院路37号(100083) 发行部电话:(010)82317024 传真:(010)82328026

<http://www.buaapress.com.cn> E-mail: bhp@263.net

北京市松源印刷有限公司印装 各地书店经销

*

开本:787×1092 1/16 印张:9.25 字数:237千字

2003年12月第1版 2003年12月第1次印刷 印数:5000册

ISBN 7-81077-400-X 定价:14.00元



总 前 言

科教兴国,教育先行,在全国上下已形成共识。在教育改革过程中,出现了多渠道、多形式、多层次办学的局面。同时,政府逐年加大教育的投入力度。教育发展了,才能有效地提高全民族的文化、科学素质,使我们中华民族屹立于世界民族之林。

计算机科学与技术的发展日新月异,其应用领域迅速扩展,几乎无处不在。社会发展的需求,促使计算机教育生气蓬勃。从普通高校的系统性教学,到远距离的电视、网上教学;从全面讲述,到不同应用领域的、星罗棋布的培训班;从公办的到民办的;从纸介教材到电子教材等等,可以说计算机教学异彩纷呈。要进行教学,就必须有教材。

面对我们这么大的国家和教学形势,在保证国家教学基本要求的前提下,应当提倡教材多样化,才能满足各教学单位的需求,使他们形成各自的办学风格和特色。为此,我们组织北京工业大学、北京航空航天大学、北京理工大学、南开大学、天津工业大学等高校的有丰富教学经验的教师编写了计算机教学系列教材,将陆续与师生见面。

系列教材包括以下各项。

(一) **基础理论**:离散数学。

(二) **技术基础**:电路基础与模拟电子技术;数字逻辑基础;计算机组成与体系结构;计算机语言(拼盘,选择使用),包括C++程序设计基础、Visual Basic 程序设计基础、Matlab 程序设计基础、Java 程序设计基础、Delphi 语言基础、汇编语言基础等;数据结构;计算机操作系统基础;计算方法基础;微机与接口技术;数据库技术基础等。

(三) **应用基础**:计算机控制技术;网络技术;软件工程;多媒体技术等。

(四) **技术基础扩展**:编译原理与编译构造;知识工程——网络计算机环境下的知识处理。

(五) **应用基础扩展**:计算机辅助设计;单片机实用基础;图像处理基础;传感器与测试技术;计算机外设与接口技术。

本系列教材主要是针对计算机教学编写的,供普通高校、社会民办大学、高等职业学校、业余大学等计算机专业本科或专科选用。其中一部分教材也适合非计算机专业本科教学使用。在这些教材的内容简介或前言中对使用范围均作了说明。

本系列教材在编写时,注重以下几点:(1) 面对计算机科学与技术动态发展的现实,在内容上应具有前瞻性;(2) 面对学以致用,既有系统的基础知识,又具有应用价值的实用性;(3) 具有科学性、严谨性。另外,力求排版紧凑,使有限的版面具有最大的信息量,以使读者得到实惠。

能否实现这些愿望,只有师生在教学实践中评价。我们期望得到师生的批评和指正。

高校计算机教学系列教材编委会

AJS-259/05



高校计算机教学系列教材编委会成员

主 任:赵沁平

副 主 任(常务):陈炳和

顾 问:麦中凡

委 员(以姓氏笔划为序):

吕景瑜(北工大教授)

乔少立(社长,副教授)

麦中凡(北航教授,教育部工科计算机基础教学指导委员会副主任、中专计算机
教学指导委员会顾问)

苏开娜(北工大教授)

陈炳和(北工大教授)

张鸿宾(北工大博导)

郭玉明(北工大副教授)

金茂忠(北航博导)

赵沁平(北航博导,国务院学位办主任)



前言

一般,“微机原理与应用”和“汇编语言程序设计”是高校计算机专业或电子类专业的必修课程,都是实践性很强的课程,只有通过上机才能掌握程序设计技术;只有通过实验才能复习和验证其原理,才能使教学达到较高的水平。因此我们编写了这本实验教材,一方面是为了便于组织教学;另一方面是随着课时的压缩便于读者自学。

全书由两个部分组成。

第一部分为软件部分,分为4章,第1章介绍上机的基本方法及上机需要的软件如何使用,包括EDIT、DEBUG、MASM等。这一章以自学为主;第2章介绍初级程序实验,包括3个实验。通过初级程序实验可进一步掌握各种寻址方式,并强调DEBUG的运用;第3章介绍程序的基本结构练习,综合了顺序、循环、分支和子程序4种基本结构的编程技术;第4章介绍输入/输出程序设计及演示程序,包括程序直接控制、DOS系统功能调用、BIOS调用及中断程序设计技术等内容,提供了4个演示程序供自学参考。

第二部分为硬件部分,由第5章和第6章组成。第5章介绍教学实验的主要设备——TPC-H微型机接口实验系统的安装及基本构成电路;第6章包括18个实验,有I/O地址译码、并行接口、中断、串行接口、A/D、D/A、DMA及实际应用方面的实验,如七段数码管、竞赛抢答器、交通灯控制、集成电路测试等。这组实验对学生的训练是全面的。

在“汇编语言”课程中,安排16学时的上机训练;“微机原理与应用”课程中,安排24学时的上机与实验。加*号的实验也可作为课程设计的内容。

书末配有实验报告表和附录。前者有软件实验表和硬件实验表供学生使用;后者给学生查询提供方便。

本书的编写得到了北京航空航天大学计算机学院吴威教授的大力支持,在此表示感谢。

由于作者的实际经验和水平所限,本书难免会有错误或不足之处,欢迎读者批评指正。

赵梅 于苏州滨河路298号
2003年8月



目 录

第一部分 软件实验

第 1 章 实验的基本要求与常用软件的使用	3
1.1 实验的目的与要求	3
1.2 全屏编辑程序——EDIT	3
1.3 调试程序——DEGUG	5
1.4 宏汇编程序(MASM)及汇编语言上机过程	25
第 2 章 初级程序实验	35
2.1 实验 1:寻址方式与指令练习	35
2.2 实验 2:初级程序的编写与调试	36
2.3 实验 3:加法程序的编写与调试	37
第 3 章 程序的基本结构练习	39
3.1 实验 4:简单程序设计	39
3.2 实验 5:循环程序设计	41
3.3 实验 6:分支程序设计	43
3.4 实验 7:子程序设计	43
第 4 章 输入/输出程序设计及演示程序	46
4.1 实验 8:发声系统程序设计	46
4.2 实验 9:显示器 I/O 程序设计	47
4.2 实验 10:中断程序设计	47
4.4 演示程序	49

第二部分 硬件实验

第 5 章 教学实验的主要设备	65
5.1 TPC-H 微型机接口实验系统	65
5.2 设备安装	66
5.3 TPC-H 实验箱电路结构	66
第 6 章 PC 系列微机硬件电路实验	74
6.1 实验 1:I/O 地址译码	74
6.2 实验 2:简单并行接口	75
6.3 实验 3:可编程并行接口(8255 方式 0)	77
6.2 实验 4:七段数码管	78
6.5 实验 5:可编程定时器/计数器(8253)	80



6.6 实验 6:竞赛抢答器*	82
6.7 实验 7:交通灯控制*	83
6.8 实验 8:集成电路测试*	85
6.9 实验 9:继电器控制	87
6.10 实验 10:中断	88
6.11 实验 11:可编程并行接口(8255 方式 1)	90
6.12 实验 12:数/模转换器	92
6.13 实验 13:模/数转换器	93
6.14 实验 14:数字录音机	94
6.15 实验 15:串行通讯	97
6.16 实验 16:DMA 传送	98
6.17 实验 17:电子琴*	101
6.18 实验 18:8250 串行通讯实验	102
附 表	105
附 录	123
附录 I DOS 系统功能调用(INT 21H)	123
附录 II BIOS 功能调用	127
附录 III ASCII 码字符表	130
附录 IV 汇编程序出错信息	131
附录 V 习 题	134
参考文献	138

第一部分

软件实验



第 1 章

实验的基本要求与常用软件的使用



1.1 实验的目的与要求

1.1.1 实验目的

学习汇编语言程序设计的基本方法和技能,熟练掌握用汇编语言设计、编写、调试和运行程序的方法。为后继课程打下坚实的基础。

1.1.2 实验要求

1. 上机前要作好充分准备

包括程序框图、源程序清单、调试步骤、测试方法、运行后的结果分析等。

2. 实验报告的要求

(1) 设计说明

用来说明程序的功能、结构。包括程序名、功能、原理及算法说明和主要符号名的说明。

(2) 调试说明

调试情况说明包括:遇到的问题及解决的方法;观察到的现象及分析;程序设计的技巧的总结及分析;程序的输出结果及结果的分析;实验的心得体会;调试日期等。

(3) 使用说明

程序提供给用户使用时必须作出的说明。如:程序的使用方法,调用方式,操作步骤等;要求输入信息的类型和格式;出错信息的含义及程序的适用范围等。

(4) 程序流程图



1.2 全屏幕编辑程序——EDIT

在编写汇编语言源程序时,可通过全屏幕编辑程序 EDIT 或 PE 等建立源程序。

1.2.1 EDIT 的功能及应用范围

EDIT 是基于 DOS 平台下的文本编辑工具,具有常规的操作命令和灵活的键盘功能。其



主要功能如下。

(1) 有丰富的编辑命令:可以调入已保存的文件进行编辑,可以对字符串进行查找、替换等操作。

(2) 有区域标记定义和操作功能:可以进行区块的定义、剪切、复制、删除等操作,可以设置前景及背景颜色。

(3) 灵活的键盘功能。

(4) 方便的打印功能。

1.2.2 EDIT 的启动与退出

1. 启动

EDIT 的启动同其它 DOS 外部命令一样,只需在 DOS 状态下键入:

EDIT [文件名]

其中[文件名]是一个任选项。如果指定文件名,EDIT 就装入指定的文件以供编辑。如果不指定文件,则 EDIT 就建立一个无名文件,其后用户可在存盘时指定文件名。

2. 存盘与退出

按 Alt 键激活菜单,选择 file 菜单,再选择 save……或 save as……或 exit。分别为保存、另存为和退出。

1.2.3 编辑一个汇编语言源程序

在 DOS 状态下可以编辑一个汇编语言源程序,具体步骤如下。

(1) 在 DOS 状态下进入 EDIT 状态。

EDIT filename. ASM ✓

(2) 在编辑状态下键入用汇编语言编写的源程序:

```
DATA    SEGMENT
        S1  DB  'HOW ARE YOU !',' '$
DATA     ENDS
CODE     SEGMENT
ASSUME CS:CODE,DS:DATA
START:   MOV  AX,DATA
          MOV  DS,AX
          MOV  AH,9H
          MOV  DX,OFFSET S1
          INT  21H
          MOV  AH,4CH
          INT  21H
CODE     ENDS
```



END START

(3) 保存源文件:按 Alt 键激活菜单,选择 File 菜单,选择 Save 命令保存源文件。

(4) 退出 EDIT 状态:选择 file 菜单,选择 Exit 命令,退出 EDIT 状态。

1.3 调试程序——DEBUG

在编写和运行汇编程序的过程中,会遇到一些错误和问题,需要对程序进行分析和调试。调试程序 DEBUG 就是专为小汇编和宏汇编语言设计的一种调试工具。它在调试汇编语言程序时有很强的功能,能使程序设计者接触到机器内部,能观察和修改寄存器和存储单元内容,并能监视目标程序的执行情况,使用户真正接触到 CPU 内部,与计算机产生最紧密的工作联系。

1.3.1 DEBUG 的主要特点

1. 能够在最小环境下运行汇编程序

在 DOS 状态下运行汇编程序,必须将源程序经过 MASM(或 ASM)汇编后,还要经过 LINK 连接程序产生可执行程序,才能最终运行,比较麻烦。在 DEBUG 状态下,为用户提供了调试、控制测试的环境,可以在此环境下进行编程、调试、监督、执行用户编写的汇编程序。因此调试周期短,为用户提供了极大的方便。

2. 提供极简单的修改手段

DEBUG 提供了修改命令,可以修改内存单元内容,为调试程序、修改程序带来了方便。

3. 提供了用户与计算机内部联系的窗口

DEBUG 具有显示命令。它既可以使用户看到某内存单元或某一块单元内容,也可以看到 CPU 内部各寄存器的内容。用单步执行命令跟踪执行,每执行一步都能使用户看到各寄存器的内容的变化,以便分析和调整程序。

4. 可装入、修改或显示任何文件

当然在 DEBUG 状态下运行汇编程序也具有一定局限性。

(1) 在 DEBUG 状态下运行的程序不能使用宏汇编程序中的宏指令,大部分伪指令也不能使用,因此只能把程序分段调试。

(2) 不能调试太长的程序,只能分块进行程序设计。

(3) 在 DEBUG 状态下调试好的程序不能形成可执行文件(.EXE),因此调试好的只能记下,到编辑环境下键入调试好的程序,通过汇编程序(ASM 或 MASM),再通过连接程序(LINK)形成可执行文件(.EXE)。



1.3.2 通过 DEBUG 编写、运行汇编程序

下面通过例子使大家了解在 DEBUG 状态下编写、运行汇编语言程序的过程。

[例 1.1] 用汇编语言编写一个简单的加法。

```
MOV    AL,32H      ;将 2 的 ASCII 码送 AL 寄存器中
MOV    DL,37H      ;将 7 的 ASCII 码送 DL 寄存器中
ADD    DL,AL        ;将 2+7 结果送 DL 寄存器中
SUB    DL,30H      ;将结果进行调整,得到 9 的 ASCII 码,送 DL
MOV    AH,2        ;
INT     21H        ;输出 DL 寄存器中的字符
INT     20H        ;中断当前执行程序
```

该程序是将两个十进制数 2 和 7 的 ASCII 码送入寄存器 AL 和 DL。当 ASCII 码进行相加时,必须通过调整后才能得到正确结果。如例中:32H+37H 等于 69H,69H-30H=39H,39H 才是十进制 9 的 ASCII 码。将结果在显示器上输出时,该程序段采用的是 DOS 系统功能调用(INT 21H)中的 2 号功能,即 DL 寄存器中的字符(ASCII 码)显示在显示器上,INT 20H 是中断正常结束程序。

运行步骤如下。

1. 进入 DEBUG 状态

在 DOS 状态下,进入装有 DEBUG 程序的磁盘目录,如:D 盘 MASM 目录下装有 DEBUG 调试工具。

```
D:\>CD MASM ✓
D:\MASM\>DEBUG ✓
```

屏幕显示:

—

“—”为进入 DEBUG 状态。在该提示符下可键入 DEBUG 命令。下划线部分为用户键入的字符或命令。

2. 键入程序并汇编

用 DEBUG 的 A 命令送入程序:

```
—A100 ✓
0A47:0100  MOV AL,32 ✓
0A47:0102  MOV DL,37 ✓
0A47:0104  ADD DL,AL ✓
0A47:0106  SUB DL,30 ✓
0A47:0109  MOV AH,2 ✓
0A47:010B  INT 21 ✓
0A47:010D  INT 20 ✓
```



0A47:010F ↵

当键入 A 命令时,会自动产生程序所送内存单元的段地址和偏移地址,通过偏移地址可以看到每条指令占内存单元多少个字节,MOV AL,32 占两个字节,SUB DL,30 占 3 个字节。当程序段送完时,只键入回车键↵,就退出汇编状态(A 状态)回到 DEBUG 状态“—”。其中送入数据为十六进制数,DEBUG 状态下程序中的数据均按十六进制处理,不需要键入 H 来表示数据为十六进制数。

3. 执行程序

用 DEBUG 的 G 命令执行刚刚汇编的程序:

—G↵
9

Program terminated normally

4. 反汇编

可以用反汇编 U 命令将键入的程序调出,并且可以得到每条汇编指令的机器码。

—U 100 10D↵

0A47:0100 B032	MOV	AL,32
0A47:0102 B237	MOV	DL,37
0A47:0104 00C2	ADD	DL,AL
0A47:0106 80EA30	SUB	DL,30
0A47:0109 B402	MOV	AH,02
0A47:010B CD21	INT	21
0A47:010D CD20	INT	20

其中命令 U 后面的地址为要反汇编程序的起始偏移地址和终止偏移地址。

5. 退出 DEBUG 返回 DOS 状态

—Q↵

D:\MASM>

1.3.3 DEBUG 的进入

1. DEBUG 的启动

在操作系统(DOS)状态下,直接调入 DEBUG 程序,键入命令的格式如下:

DEBUG[d;][Path][Filename[.exe]][Parm1][Parm2]

其中[]的内容为可选项。

[d:]为驱动器号,是指要调入 DEBUG 状态的可调试文件在哪个驱动器中,如 A:、C:、



D:等。

[Path]为路径,是指要调入 DEBUG 状态的可调试文件是在哪个目录下或子目录下。

[filename[.exe]],是指要调入 DEBUG 状态的可调试文件的文件名。该文件是通过编辑、汇编、连接后产生的可执行文件,也可以是在 DEBUG 状态下汇编的程序段,通过写盘命令 W 写入磁盘的文件。

[Parm1][Parm2]为任选参数,是给定文件的说明参数。

在启动 DEBUG 时,如果输入了 filename(文件名),则 DEBUG 程序把指定文件装入内存。用户可以通过 DEBUG 的命令对指定文件进行修改、显示或执行。如果没有文件名,则是以当前内存的内容工作,或者用命名命令或装入命令把需要的文件装入内存,然后再通过 DEBUG 命令进行修改、显示或执行。

2. 启动 DEBUG 后对寄存器和标志位的初始化

当启动 DEBUG 程序后,屏幕上出现“—”,说明系统已进入 DEBUG 状态,可以调用 DEBUG 的命令,寄存器和标志位置成下面状态:

(1) 段寄存器(CS,DS,ES 和 SS)被置到自由存储空间的底部,即第一段位于 DEBUG 程序的末尾处。

(2) 指令指针(IP)置为 0100H(或 0000H)。

(3) 堆栈指针(SP)置为段的尾部或装入程序的暂存区域的底部。

(4) 寄存器(AX,BX,CX,DX,BP,SI 和 DI)置为“0”。若启动 DEBUG 程序时指定了文件,则 CX 寄存器内装入文件长度(字节数)。如果文件长度大于 64KB,则文件长度位于 BX 和 CX 中(高位在 BX 中)。

(5) 标志位置为清除值。

由此可见,所有可利用的内存空间都作了安排,因此不能用装入的程序去分配内存。如装入程序扩展名为 .EXE 的文件,进入 DEBUG 后由 DEBUG 进行分配,把段寄存器、堆栈指针置成程序中所规定的值。

1.3.4 DEBUG 的主要命令

1. DEBUG 命令的有关规定

(1) DEBUG 命令都是一个字母,后面跟着一个或多个有关参数。多个操作参数之间用“,”或空格隔开。

(2) DEBUG 命令必须接着按 ENTER 键命令才有效。

(3) 参数中不论是地址还是数据,均用十六进制数表示,但十六进制数据后面不要用“H”。

(4) 可以用 Ctrl_Break 键来停止一个命令的执行,返回到 DEBUG 的提示符“—”下。

(5) 用 Ctrl_Num Lock 键中止正在上卷的输出行,再通过按任意键继续输出信息。



2. DEBUG 命令

(1) 汇编命令 A

格式:① A[段寄存器名]:[偏移地址]

② A[段地址]:[偏移地址]

③ A[偏移地址]

④ A

功能:用该命令可以将汇编语言程序直接汇编进入内存。

当键入 A 命令后,显示段地址和偏移地址等待用户键入汇编指令。每键入一条汇编指令回车后,自动显示下一条指令的段地址和偏移地址,再键入下一条汇编指令,直到汇编语言程序全部键入,又显示下一条地址时可直接键入回车返回到提示符“—”为止。

其中①的段地址在段地址寄存器中,因此在使用该命令时必须将段地址寄存器送入段地址;③的段地址在 CS 中;④的段地址在 CS 中,偏移地址为 100H。

例如:用汇编语言编写一个程序段,将十六进制 0、1、2……F 的 ASC II 码送入偏移地址为 100H 单元开始的存储单元中,并将该十六进制数据从 100H 单元开始的存储区传送到以 200H 为起始地址的存储区中。

将编好的程序段通过命令 A 送入内存并汇编,用块传送指令 MOVSB 将数据串进行传送。

```
D:\MASM>DEBUG ✓
-A ✓
14DE:0100 DB 30,31,32,33,34,35,36,37,38,39,41,42,43,44,45,46 ✓
14DE:0110 MOV SI,100 ✓
14DE:0113 MOV DI,200 ✓
14DE:0116 MOV CX,10 ✓
14DE:0119 REP MOVSB ✓
14DE:011B HIT ✓
14DE:011C ✓
—
```

其中,SI 为源串地址寄存器;DI 为目的串地址寄存器;计数寄存器 CX 存放计数初值 10H。REP MOVSB 为重复传送;以字节为传送单位。每传送一个字节,CX 中计数值减“1”,直到 CX 为 0 为止。

(2) 显示内存命令 D

格式:① D [地址]

② D [地址范围]

③ D

功能:显示指定内存范围的内容。