

HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS



微机技术实践(修订版)

WEIJI JISHU SHIJIAN

谢瑞和 等编著

华中理工大学出版社

(鄂)新登字第 10 号

图书在版编目(CIP)数据

微机技术实践(修订版)/谢瑞和等编著. —修订版

—武汉:华中理工大学出版社,1995年5月

ISBN 7-5609-1113-7

I. 微…

I. 谢…

II. 微型计算机-计算机应用-教材

N. TP39

微机技术实践

(修订版)

谢瑞和 等编著

黄狂编辑 姜新祺

*

华中理工大学出版社出版发行

(武昌喻家山 邮编:430074)

新华书店湖北发行所经销

武汉市第二印刷厂印刷

*

开本:850×1168 1/32 印张:8.375 字数:205 000

1992年8月第1版 1995年5月第2版 1995年5月第2次印刷

印数:2001-7000

ISBN 7-5609-1113-7/TP·139

定价:6.60元

内 容 简 介

本书是以专业基础课的写作方式编写的电类专业微机技术实验课教材,初版问世后,深受读者欢迎。本次修订又根据微机技术的发展在内容上作了较多的更新和调整,使本书的内容更加丰富和新颖。它介绍了 Z80 微处理器、MCS-51 单片机及 IBM PC/XT/AT 计算机的接口设计及应用系统设计的一般方法,内容广泛;所选内容考虑了实验条件的经济性,使得一般的学校都有可能实施。因此,本书可供各级各类大专院校选作实验课教材。

本书实用性强,透明度高,有不少内容本身就可以形成产品或实际应用模块,对于学习微机技术及从事微机应用设计的工程技术人员来说,也不失为一本较好的参考书。

初 版 前 言

从 80 年代初期开始,我们华中理工大学电信系就为本、专科学生开设了有关微机技术的课程,学生们学习的积极性很高。但由于教学条件的限制,学生们实践的机会很少,只能敲一敲键盘,运行几个简单的程序。

随着微机技术突飞猛进的发展及教学条件的不断改善,到了 80 年代中期,实验室已拥有相当数量的 A/D 及 D/A 接口板、自制的接口实验板,这时,学生们不只是敲一敲键盘了,他们能利用这些接口板作部分实验,尽管仍然只能以调试有关程序为主,或者是简单地拨动几个开关,观察某些现象,但毕竟是向前跨了一大步。

相对于科学技术的发展水平而言,我们前进的步伐实在是太小了。微机新技术不断地吸引着学生,使他们的求知欲望远远得不到满足。这种形势也激励着教师们的事业心与责任感,要培养出高质量的社会主义建设人材,我们必须进行大胆的课程改革。在校系领导的全力支持下,从 1988 年开始,我们建立了开放型微机实验室,独立设置了微机技术应用实验课,编写了讲义。我们将有关接口芯片及元件发放给每个学生,让学生们自己设计、安装、调试实验电路。经过几届学生的实践证明,这种改革是成功的,深受学生的欢迎。他们掌握与应用微机技术的能力明显提高,不少学生利用第二课堂与毕业设计直接参加微机应用的项目研究,其成果直接得到应用,有的获得全国、省市级的大学生科技成果奖励。社会上的反馈信息表明,我们的毕业生微机技术应用能力较强。

本书就是在这种前提下总结多年的教学实践经验,并力图达到新的高度而编写的。编写本书的几点指导思想是:

1. 内容尽量广。考虑到我校和各兄弟院校目前教学中使用最为广泛的计算机是 Z80、MCS-51 单片机及 IBM PC/XT/AT 等三种机型,本书将三者构成一个整体,但又有各自的独立性,读者可根据具体情况进行取舍。

2. 力求实用。书中所述内容一般都能在实验室实施,而且,有不少内容本身就可形成产品或实际应用模块。因此,本书不仅适用于教学,而且对工程设计人员也有较高的参考价值。

3. 精选内容。本书是微机原理的后续教材,在选材方面尽量避免与目前几本流行的教材内容重复,并考虑了与它们的衔接,力求与之配套。

4. 透明度较高。考虑到学生的知识基础,本书各章节大部分内容都有较完整的硬件电路图与程序,它们一般都具有代表性与启迪作用。少量内容有一定的难度,可作为选做内容。每节后面都有思考题与习题,可启发学生的思维及灵活地实施自己的设计。

5. 讲究经济性。本书是一本实践性很强的教材,就一个系来说,每届学生都有几十人、几百人,假如开销太大,开放实验就不可能维持下来。本书涉及的大部分元器件都是市场上的通用器件,实践内容的安排都考虑了不需要特殊的仪器设备,使得各大专院校在现有的条件下一般都能实施。

最后,考虑到本书既能作为本、专科学生的实验指导书,又能供读者作为学习微机技术或应用设计的参考书,我们打破了传统的实验指导书格式,以专业基础课的形式与要求组织编写,希望这种尝试能获得较好的效果。

本书叙述的内容多数是作者多年来积累的教学与科研第一线资料,全书语言简洁精练。

本书由谢瑞和、马爱梅负责编写。参加编写工作的还有朱定华同志。黄铁侠教授对本书的出版给予了极力的推荐,陈耀奎副教授及电子与信息工程系微机技术教研室的同志们给予了热情的支持,谨在此一并致谢。

由于水平所限,书中定有错误与不妥之处,敬请读者批评指正。

编著者

1992年1月于华中理工大学

修订版序言

《微机技术实践》问世不到两年,在广大读者中已留下了深刻的印象,以致供不应求,并于1993年被评为中南地区大学出版社优秀教材二等奖。这使我们感到欣慰。然而计算机的发展日新月异,以致于这方面的教材无法跟上它的步伐。《微机技术实践》尽管受到众多读者的喜爱,但与当今的微机技术发展仍有很大的差距。为了尽量缩短这一差距,我们在原书的基础上又重新编写了这一修订版。

编写修订版的宗旨仍立足于加强高等院校理工科电子信息类专业本、专科学生微机应用基本技术的训练,但本版与初版相比,在内容上作了较多的更新与调整。它以IBM PC/XT/AT系列计算机及MCS-51单片机为主,以适应我校与其它许多兄弟院校当前微机技术实验教学的需要。此外,为了扩展学生视野,增长系统设计的实际知识,同时为第二课堂的开展提供些材料,启发思路,本版增加了若干节选读内容。这些内容对于工程技术人员也具有重要的参考价值。

本书由谢瑞和、朱定华、马爱梅负责编写,李群芳等微机教研室的同志们也做了许多工作,校系有关领导给予了全力支持,在此一并致谢。

编著者

1994年7月于华中理工大学

目 录

修订版序言

初版前言

第一章 IBM PC/XT/AT 计算机程序设计 (1)

§ 1.1 汇编语言程序的汇编调试与运行 (1)

§ 1.2 汇编语言程序设计(一)..... (14)

§ 1.3 汇编语言程序设计(二)..... (31)

§ 1.4 高级语言与汇编语言的接口..... (39)

第二章 IBM PC 系列计算机接口设计 (47)

§ 2.1 IBM PC 总线及其总线接口设计 (47)

§ 2.2 并行接口芯片 8255A (56)

§ 2.3 计数器/定时器电路 8253/8254 (66)

§ 2.4 A/D 转换接口设计 (75)

§ 2.5 D/A 转换接口设计 (82)

§ 2.6 12 位 A/D 的接口设计 (85)

§ 2.7 12 位 D/A 的接口设计 (93)

§ 2.8 电子显示屏的设计..... (96)

§ 2.9 DMA 传送 (102)

§ 2.10 串行通信..... (107)

§ 2.11 远程数据采集..... (118)

第三章 MCS-51 单片机应用设计 (126)

§ 3.1 TP801-8031 单片单板机的设计与使用..... (126)

§ 3.2 实践举例 (133)

§ 3.3 MCS-51 模拟仿真软件 (146)

§ 3.4 MICE-EZ 8051 仿真器 (149)

§ 3.5	显示器的设计	(162)
§ 3.6	键盘的设计	(165)
§ 3.7	并行打印机接口设计	(168)
§ 3.8	积分式 A/D 接口设计	(173)
§ 3.9	自动频率计的设计	(177)
§ 3.10	步进电机控制器	(184)
§ 3.11	阻抗测量仪	(188)
§ 3.12	快速面积测量仪	(192)
第四章	Z80 微处理器应用设计	(196)
§ 4.1	程序的运行实践	(196)
§ 4.2	程序设计	(200)
§ 4.3	存储器与 I/O 端口及其中断	(214)
§ 4.4	并行接口芯片 Z80 PIO	(218)
§ 4.5	并行接口芯片 8255A	(223)
§ 4.6	计数器/定时器芯片 Z80 CTC	(229)
§ 4.7	用 8253/8254 测量转速或频率	(232)
§ 4.8	积分式 A/D 的接口设计	(235)
§ 4.9	IBM PC—Z80 交叉汇编	(241)
附录 1	TP801 单板机键盘操作说明	(245)
附录 2	常用集成电路引脚图	(249)
	主要参考文献	(257)

第一章 IBM PC/XT/AT 计算机 程序设计

本章学习 IBM PC/XT/AT 等计算机的应用程序设计技术。目前国内教学用机中,PC/XT/AT 机主板上 CPU 为 8088/8086/80286 的机型占主流,CPU 为 80386/80486 的 32 位高档微机也占有较大比例,并在逐步取代前者。一般说来,无论是何种机型,对于初学者而言,一般应用程序都是兼容的。为简述起见,本书除特别说明外,统称这些机器为 PC 机。

§ 1.1 汇编语言程序的汇编调试与运行

1. 运行汇编语言源程序的过程

①用行编辑 EDLIN 或全屏幕编辑 PE 或多功能窗口软件 SK 或 TURBO PASCAL 编辑器等建立或修改扩展名为 ASM 的源程序文件。

②用汇编程序 MASM 或 TASM 将 ASM 文件汇编生成扩展名为 OBJ 的目标文件。

③用连接程序 LINK 将 OBJ 文件连接成可执行的 EXE 文件。

④用调试程序 DEBUG 调试 EXE 文件。

经过上述前三步生成的 EXE 文件,在 DOS 状态下直接键入文件名,就可将 EXE 文件从磁盘调入内存并立即执行。然而,一个较复杂的汇编语言源程序一次通过的可能性是很小的,免不了会出现一些意想不到的错误。通过汇编和连接的程序仅是一个无语

法错误的程序,不一定是正确的程序。因此,必须利用调试程序 DEBUG 对 EXE 文件进行调试。一旦程序中存在错误,就必须从修改源程序开始重复上述过程,直到程序完全正确为止。

2. 多功能窗口软件 SK (SIDE KICK)

SK 的编辑窗口是一个全屏幕文本编辑窗口。SK 与其编辑文本长驻内存,用它编辑、修改源程序十分方便。进入 SK 编辑窗口的步骤如下:

①在 DOS 状态下键入 SK $\swarrow$ 。键入后 SK 程序被装入内存,且长驻内存,此后便可随时按下面的步骤调用而不必再键入 SK $\swarrow$ 。

②键入<Alt>+<Ctrl>,进入主选择窗口。

③键入<F2>或<Alt>+N 或将主选择窗口的指示线移至 F2 Notepad 下键入 $\swarrow$,即可进入编辑窗口。

进入编辑窗口后,<Alt>+<Ctrl>就是个互锁键,即可退出或直接进入编辑窗口,而不需再途经主选择窗口。初次进入编辑窗口时,会在其上部出现一个显示文件目录的小窗口,移动光标至欲编辑的文件名处键入 $\swarrow$,该文件便被调入。如不编辑当前文件,键入<ESC>便可退出显示目录的小窗口,建立一个新文件;或键入路径名*.* ,让小窗口显示该路径中的文件目录。若文件太多,查找不便,也可在退出小窗口后键入文件名,将该文件调入。

SK 的命令和功能键较多,可随时键入功能键<F1>查询,本书不作详细介绍。

用 SK 编辑下面的源程序 EXAM·ASM,以熟悉运行汇编语言源程序的全过程。

```
STACK    SEGMENT PARA STACK 'STACK'
          DW 32 DUP(?)
STACK    ENDS
DATA1    SEGMENT
A         DW      M
```

```

BUF      DB      'AB',0DH,0AH,'$'
CON1     EQU      3
CON2     EQU      12
B        DW      75,-5,100
C        DB      6 DUP(?)
D        DD      N
M        DB      2 DUP(1),2 DUP(2,'B'),'123',1,2,3
DATA1    ENDS
DATA2    SEGMENT
N        DB 12 DUP(?)
DATA2    ENDS
CODE     SEGMENT
BEGIN    PROC FAR
        ASSUME SS:STACK,CS:CODE,DS:DATA1,ES:DATA2
        PUSH    DS
        SUB     AX,AX
        PUSH    AX
        MOV     AX,DATA1
        MOV     DS,AX
        LEA     DX,BUF
        MOV     AH,9
        INT     21H
        MOV     BX,OFFSET B
        MOV     BP,OFFSET C
        MOV     CX,CON1
LOPA:    MOV     AX,[BX]
        MOV     WORD PTR DS,[BP],AX
        INC     BX
        INC     BX
        INC     BP
        INC     BP
        LOOP    LOPA

```

```

MOV    SI,A
LES    DI,D
MOV    CX,CON2
CLD
REP     MOVSB
RET
BEGIN  ENDP
CODE   ENDS
END     BEGIN

```

3. 汇编程序

(1)MACRO ASSEMBLER(MASM)

汇编语言源程序经 MASM 汇编后,在磁盘上可以建立3个文件:一个是 OBJ 目标程序文件,在此文件中,符号指令变为机器码,但地址操作数只是可浮动的相对地址,而不是内存的绝对地址;另一个是 LST 列表文件,它把源程序和目标程序都制成表,可打印出来供检查和调试用;第三个是可用来对符号进行前后对照的 CRF 文件。汇编时 MASM 对要不要建立这些文件进行提问。

汇编 EXAM. ASM 源程序,在 DOS 状态下调用 MASM 的汇编操作过程如下:

G:\>MASM_✓

Microsoft (R) Macro Assembler Version 4.00_✓

Copyright (C) Microsoft Corp 1981,1983,1984,1985.

All rights reserved.

Source filename [.ASM]:EXAM_✓

Object filename [EXAM.OBJ]:_✓

Source listing [NUL.LST]:EXAM_✓

Cross-reference [NUL.CRF]:EXAM_✓

48460 Bytes symbol space free

0 Warning Errors

0 Severe Errors

上述操作也可以用命令行的形式来完成:

```
G:\>MASM EXAM, EXAM, EXAM, EXAM
```

若只生成 OBJ 文件,可用符号“,”表示以后的参数选用缺省值。这时只需键入命令

```
G:\>MASM EXAM,
```

汇编过程中若发现源程序有语法错误,则计算机可自动列出错误所在的行号和错误性质的编号,并指出错误的类型,最后列出错误的总数。此时应分析错误,修改源程序后再汇编,直至汇编后无错误。

汇编后建立的 LST 文件可用于打印,以了解汇编后的情况。汇编后建立的 CRF 文件是不能单独使用的,若需要了解源程序中的符号在定义和引用时的情况,还要调用 CREF 程序对 CRF 文件进行处理,生成 REF 文件。其过程如下:

```
G:\>CREF
```

```
·Cref filename [CRF] EXAM
```

```
List filename [EXAM.REF]:
```

生成的 REF 文件可以打印出来。

(2) TURBO ASSEMBLER (TASM)

TASM 是 16 位 Turbo 汇编程序,和 MASM 相比,其差别主要是前者使用了以下简化的段定义伪指令:

- STACK n 定义 n 个字节堆栈段。
- CODE 定义代码段,标志代码段的开始。
- DATA 定义数据段,标志数据段的开始。

用户必须将 @DATA 的内容装入 DS 寄存器。ES 段的定义、使用与装入同 DS。

·DOSSEG 规定程序中段的次序。该伪指令使源程序中的段按照 Microsoft 段排序约定进行分组。

·MODEL 为使用简化段定义伪指令的汇编模块指定内存模式,即如何在内存中为代码、数据、堆栈分配空间。可用的内存模式有:

①tiny——微模式 程序代码和数据都在同一个64K 的段中。其中,代码和数据指近调用代码和近调用数据。

②small——小模式 代码段在单一64K 的段中,数据在一个不同的64K 的段中。其中,代码指近调用代码,数据指近调用数据。

③medium——中模式 程序代码可多于64K,但数据只能在单一的64K 的段中。其中,代码指远调用代码,数据指近调用数据。

④compact——紧凑模式 代码段在单一的64K 的段中,数据可以多于64K。其中,代码指近调用代码,数据指远调用数据,但数组不可大于64K。

⑤large——巨模式 代码和数据都可多于64K,数组也可多于64K。其中,代码和数据均为远调用代码和数据,指向数组元素的指针是远调用指针。

实际应用中很少有程序需要多于64K 的代码段和数据段,所以大多数程序选用 small 模式已足够了,而且只要有可能,用户应该尽量使用 small 模式,因为远调用代码使程序的执行速度降低,远调用数据在汇编时管理相当困难。

值得特别指出的是,使用简化的段定义伪指令时一定要用 .MODEL,否则 TASM 无法得知如何建立 .CODE 和 .DATA 所定义的段,而且 .MODEL 务必在所有段定义之前使用。

使用简化段定义伪指令的源程序格式如下:

```
DOSSEG
.MODEL    SMALL
.STACK    40H
.DATA
.
.
.
.CODE
.
.
.
MOV      AX,@DATA
```

```

MOV    DS,AX
.
.
.
MOV    AH,4CH
INT     21H
END

```

4. 连接程序(LINK)

由汇编程序生成的目标文件必须经过连接以后才能成为可执行文件。连接程序并不是专为汇编语言设计的。利用它可把若干个模块连接在一起,这些模块可以是汇编程序产生的目标文件,也可以是其它高级语言编译程序产生的目标文件。在连接过程中,将产生几个新文件。连接过程如下:

G:\)LINK

Microsoft (R) 8086 Object Linker Version 3.05

Copyright (C) Microsoft Corp 1983, 1984, 1985. All rights reserved.

Object Modules [.OBJ]:EXAM

Run File [EXAM.EXE]:

List File [NUL:MAP],EXAM

Libraries [.LIB]:

同 MASM 一样,连接程序可以使用命令行形式。汇编语言没有库文件,若不建立内存分配文件 MAP,仅生成可执行文件,则只需键入命令:

G:\)LINK EXAM

如果由多个目标文件连成一个 EXE 文件,应一次键入各目标文件,用“+”号相互连接,例如:

G:\)LINK EXAM1+EXAM2+EXAM3

5. 调试程序(DEBUG)

DEBUG 是专为汇编语言设计的一种调试工具,它具有显示、

修改内存和寄存器或指令的功能,还可以跟踪程序的执行。

(1)DEBUG 的启动

在 DOS 状态下键入 DEBUG [文件引用名]。其中文件引用名是可选的,若有,其扩展名只能是 EXE 或 COM,其它扩展名的文件不能调试。若在 DEBUG 启动时规定了文件引用名,则在 DOS 下将 DEBUG 调入内存后,计算机还可自动将被调试的程序装入内存,用户可根据需要对装入的程序进行显示、修改或执行。若未规定文件引用名,则可用命名命令 N 和装入命令 L 将需要调试的文件装入内存,然后再对内存中的程序进行修改、显示或执行。对于简短的程序,还可以用汇编命令 A 直接编辑,免除前述的汇编、连接等步骤。

程序装入后,出现提示符“-”,这时系统已在 DEBUG 程序的管理之下,所有的 DEBUG 命令只有在出现此提示符后才有效。此时各寄存器和标志位置成以下状态:

4个段寄存器的值相同,置于自由存贮空间的底部,也就是 DEBUG 程序后的第一个段。

指令指示器 IP 置为 0100H。

堆栈指示器 SP 置到段的结尾处,或者是装入程序的临时底部,哪个更低就指向哪个。除 SP 外其余 7 个通用寄存器置为 0。

标志位均清除。

但是,若启动 DEBUG 时规定了文件引用名,则 BX 和 CX 置为以字节表示的文件长度,且当被装入的调试程序为 EXE 文件时,DEBUG 必须进行再分配,将段寄存器、IP 和 SP 置为程序所规定的值。

(2)有关 DEBUG 命令的一些共同信息

DEBUG 命令都是一个字母,后跟一个或多个参数。命令和参数之间用定界符逗号或空格分隔。两个相邻的十六进制数之间必须用定界符分隔。

命令中的参数以及程序中的数据都规定为十六进制数,由 1~