

高等院校信息技术规划教材

微型计算机原理与 接口技术实验指导

陈燕俐 李爱群 周宁宁 编著

清华大学出版社



高等院校信息技术规划教材

微型计算机原理与 接口技术实验指导

陈燕俐 李爱群 周宁宁 编著

清华大学出版社
北京

内 容 简 介

本书是《微型计算机原理与接口技术》(孙力娟等编著,清华大学出版社出版)一书的配套实验教材。教材结合课程内容,针对汇编语言程序设计、微型计算机接口技术编排了内容丰富的软硬件实验项目和指导性实验例题,主要内容有汇编语言程序设计实验、微型计算机教学实验系统及系统软件介绍、微型计算机接口实验。本书的硬件实验以南京邮电大学和福州德昌电子公司共同研发的“PD-32 开放式微型计算机教学实验系统”为实验平台。

本书内容丰富,大量的实验示例和实验项目扩展了教科书的内容,可作为高等院校汇编语言程序设计、微机原理与接口技术等课程的实验教材,也可供自学者及从事计算机应用的工程技术人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

微型计算机原理与接口技术实验指导/陈燕俐,李爱群,周宁宁编著. —北京:清华大学出版社,2010.7

(高等院校信息技术规划教材)

ISBN 978-7-302-22311-5

I. ①微… II. ①陈… ②李… ③周… III. ①微型计算机—理论—实验—高等学校—教学参考资料 ②微型计算机—接口—实验—高等学校—教学参考资料 IV. ①TP36-33

中国版本图书馆 CIP 数据核字(2010)第 055721 号

责任编辑:战晓雷 李玮琪

责任校对:李建庄

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:10.25

字 数:256 千字

版 次:2010 年 7 月第 1 版

印 次:2010 年 7 月第 1 次印刷

印 数:1~3000

定 价:15.00 元

产品编号:034930-01

前言

foreword

“微型计算机原理与接口技术”是高等院校计算机专业及电类相关专业计算机硬件课程体系中的一门重要的专业基础课,是一门理论与实际结合十分紧密,实践性很强的课程。实验是微机接口教学过程中十分重要的环节,是全面提高学生素质的有效途径。

本书是与教材《微型计算机原理与接口技术》配套的实验教程,目的是使学生通过实验加深对理论课程的理解,培养学生的编程能力和实际动手能力。本书分为软件实验和硬件实验两部分,硬件实验以南京邮电大学和福州德昌电子公司共同研发的“PD-32 开放式微型计算机教学实验系统”为实验平台。本书共分为 6 章,内容如下:

第 1 章为汇编语言程序实验基础,介绍了汇编语言源程序的格式和框架、汇编语言程序的开发过程,以及汇编语言语法练习实验,为学生学习下一步的软硬件实验打下基础。第 2 章和第 3 章是汇编语言程序设计实验示例和内容,实验内容丰富,涵盖了一般汇编语言程序设计和微机原理教学中所要求的所有软件实验。第 4 章介绍了 Win32 汇编程序的框架,Win32 汇编语言程序开发过程,以及 Win32 窗口程序、字符串显示程序、消息处理程序实验。第 5 章对“PD-32 开放式微型计算机教学实验系统”的结构和功能,上位机系统软件的使用进行了介绍。第 6 章为硬件接口实验,覆盖了目前高等院校微机接口实验教学大纲的主要内容,包括保护模式程序设计。综合性实验要求学生能熟练掌握各种常用接口芯片的结构和功能,能综合运用接口芯片达到实验要求。

本书按照实验说明、实验目的和要求、实验示例和实验项目进行每一个实验的组织,每种实验的实验示例都给出了源程序清单和注释,这些程序都经过调试和运行;涉及硬件的还给出了具体的实验原理和硬件连线。本书提供了大量的实验题目,教师可以根据本校的教学特点和要求选择相应的实验内容。

全书由陈燕俐、李爱群和周宁宁编著,其中引言和第3、4、5章由陈燕俐编写,陈燕俐、李爱群合写了第1章,陈燕俐、周宁宁合写了第2章,陈燕俐、李爱群、周宁宁合写了第6章,由陈燕俐负责全书的统稿工作。本书的编写得到了南京邮电大学的孙力娟教授和洪龙教授的热情鼓励、悉心指导和积极帮助。另外南京邮电大学的张先俊老师、薛明老师、邓玉龙老师、祁正华老师、许建老师和李凌燕老师也提出了许多宝贵建议,使得本书更加完善,在此表示衷心的感谢。本书在编写过程中,还得到了福州德昌电子公司和清华大学出版社的大力支持,也参考了相关的书籍,在此一并致以诚挚的谢意。

由于编者的水平有限,书中难免有错漏之处,恳请读者批评指正。

编者

2010年5月

目录

Contents

第 1 章 汇编语言语法实验	1
1.1 汇编语言程序开发过程	1
1.2 汇编语言程序编程练习	3
1.3 汇编语言语法实验	8
第 2 章 结构化程序设计实验	10
2.1 顺序程序设计	10
2.2 分支程序设计	12
2.3 循环程序设计	16
2.4 子程序设计	22
2.5 宏指令设计	26
第 3 章 应用程序设计实验	29
3.1 数制及代码转换程序设计	29
3.2 数值计算程序设计	33
3.3 字符串操作程序设计	37
3.4 图形显示程序设计	42
3.5 磁盘文件管理程序设计	51
第 4 章 Win32 汇编程序设计实验	57
4.1 Win32 汇编语言程序开发过程	57
4.2 Win32 汇编语言程序编程练习	60
4.3 Win32 窗口程序设计	63
4.4 字符串显示程序设计	66
4.5 消息处理程序设计	70

第 5 章 PD-32 微机教学实验系统	74
5.1 PD-32 微机教学实验系统结构	74
5.2 PD-32 微机教学实验系统资源	77
5.3 上位机系统软件的使用说明	81
第 6 章 硬件接口实验	87
6.1 计数器/定时器实验	87
6.2 并行接口实验	91
6.3 中断实验	98
6.4 串行通信实验	106
6.5 小键盘扫描实验	116
6.6 D/A 转换实验	122
6.7 A/D 转换实验	125
6.8 存储器扩充实验	129
6.9 DMA 实验	136
6.10 保护模式实验	142
6.11 综合性实验	151
参考文献	155

第 1 章

chapter 1

汇编语言语法实验

1.1 汇编语言程序开发过程

汇编语言(Assembly Language)是唯一能够利用计算机所有硬件特性,并能直接控制硬件的编程语言。用汇编语言编写的程序称为汇编语言程序,汇编语言程序必须翻译成机器语言程序(即目标代码程序),才能在机器上运行。

1. 汇编语言源程序的格式

一个完整的汇编语言源程序在结构上必须做到:

- ① 用方式选择伪指令说明执行该程序的 CPU 类型;
- ② 用段定义语句定义每一个逻辑段;
- ③ 用 ASSUME 语句说明段约定;
- ④ 用汇编结束语句说明源程序结束。

实模式下汇编程序有两种编程格式:一种格式生成扩展名为 EXE 的可执行文件,称为 EXE 文件的编程格式;另一种格式可以生成扩展名为 COM 的可执行文件,称为 COM 文件的编程格式。

典型 EXE 编程格式如下:

```
.486                                ;方式选择伪指令说明执行该程序的 CPU 类型
DATA SEGMENT USE16                 ;定义作为数据段的逻辑段,段名 DATA
;定义变量
.....
DATA ENDS                          ;数据段结束
CODE SEGMENT USE16                 ;定义作为代码段的逻辑段,段名 CODE
ASSUME CS:CODE,DS:DATA             ;段约定
BEG: MOV AX,DATA                   ;数据段的段基址赋给段寄存器
MOV DS,AX
.....                               ;程序代码
MOV AH,4CH                         ;程序结束,返回 DOS
INT 21H
```

```
CODE ENDS ;代码段结束
END BEG ;源程序结束,程序的开始点为 BEG 指令
```

典型 COM 编程格式如下:

```
.486 ;方式选择伪指令说明执行该程序的 CPU 类型
CODE SEGMENT USE16 ;定义作为代码段的逻辑段,段名 CODE
ASSUME CS:CODE ;段约定
ORG 100H ;偏移地址为 100H 的单元必须是程序的启动指令
BEG: JMP START ;跳过数据区
..... ;定义程序使用的数据,也可设置在代码段的末尾
START: ;程序代码
.....
MOV AH,4CH ;程序结束,返回 DOS
INT 21H
CODE ENDS ;代码段结束
END BEG ;源程序结束,程序的开始点为 BEG 指令
```

2. 汇编语言的开发过程

汇编语言程序设计的实验环境对计算机的配置要求比较低,普通的个人计算机一般都可以满足。常用的汇编语言开发工具有 Borland 公司的 MASM 和 Microsoft 公司的 TASM,另外集编辑、汇编链接、调试为一体的 16 位 TASM 集成环境——“未来汇编”使用也很方便。使用者可在硬盘某个分区上建立一个子目录,例如 C:\TASM,将某个开发工具的相关文件复制到此目录下,如 Borland Turbo Assembler 5.0 所对应的文件是: TASM.EXE、TLINK.EXE、RTM.EXE、DPMI16BI.OVL、TD.EXE 和 TDHELP.TDH。此外还可将某个编辑程序也复制到该目录下,如 EDIT.EXE,这样就在该子目录下构成了一个集编辑、汇编、链接和调试为一体的开发环境。

汇编语言程序的开发过程如图 1.1 所示。这个过程主要由编辑、汇编、链接和调试几个步骤构成。

(1) 源程序的编辑

编辑就是调用编辑程序编辑源程序,生成一个扩展名为 ASM 的文本源文件。DOS 提供的 EDIT.EXE 或其他屏幕编辑软件都能完成编辑任务。

(2) 源程序的汇编

为了使用汇编语言编写的程序能在机器上运行,必须利用汇编程序(Assembly Program,如 Microsoft 公司的 MASM 或 Borland 公司的 TASM)对源程序进行翻译,生成扩展名为 OBJ 的目标文件。

汇编语言源程序包含指令性语句(即符号指令)和指示性语句(即伪指令)两类语句。符号指令和机器指令具有一一对应的关系,伪指令是为汇编程序提供汇编信息,为链接程序提供链接信息,在汇编后并不产生目标代码。

在汇编过程中,汇编程序如检查到源程序中有语法错误,则不生成目标代码文件,并给出错误信息。根据用户需要,汇编程序还可生成列表文件(LST 文件)和交叉参考文件(XRF 文件)。

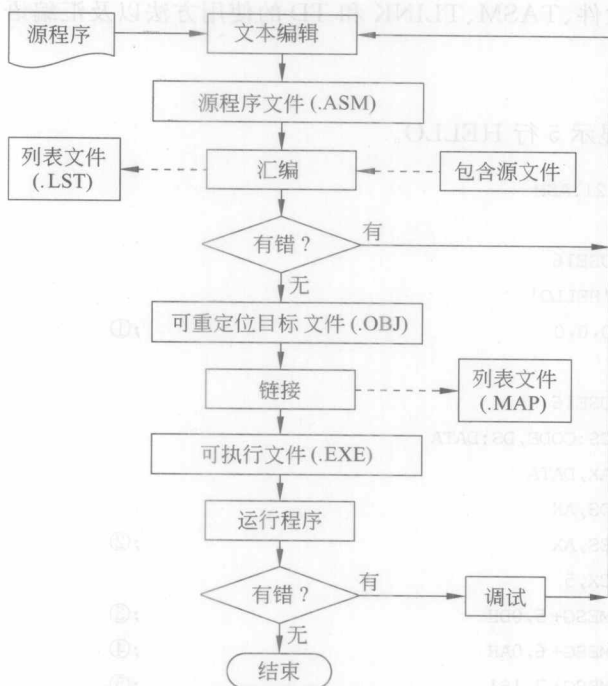


图 1.1 汇编语言程序的开发过程

(3) 目标程序的链接

链接就是利用链接程序(如 Microsoft 公司的 LINK 或 Borland 公司的 TLINK)将用户目标程序和库文件进行链接和定位,生成扩展名为 EXE 的可执行文件。链接时,如果在目标文件或库中找不到所需的链接信息,则链接程序会发出错误信息,而不生成可执行文件。根据用户需要,链接程序还可生成内存映射文件(MAP 文件)。

(4) 动态调试

有时用户生成的 EXE 文件运行后,并没有按照设计的意图运行,这就需要对程序进行调试。根据具体情况,调试的过程也不尽相同。一般地,可利用调试工具(各版本 DOS 所带的 DEBUG 或 Borland 公司的 Turbo Debugger)对生成的可执行文件进行调试,并找出错误。再对源程序进行修改……,即重复进行编辑、汇编、链接、调试,直到生成完全正确的可执行文件。

1.2 汇编语言程序编程练习

1. 实验说明

在 1.1 节的基础上掌握汇编语言程序设计过程。

2. 实验目的和要求

掌握汇编语言源程序的编辑、汇编、目标文件的链接和可执行文件的调试执行全过

程;掌握文本编辑软件、TASM、TLINK 和 TD 的使用方法以及汇编语言的语法规则。

3. 实验示例

【例 1.2.1】显示 5 行 HELLO。

```
;FILENAME: EXA121.ASM
.486
DATA SEGMENT USE16
MSG DB 'HELLO'
    DB 0,0,0
DATA ENDS
CODE SEGMENT USE16
    ASSUME CS:CODE,DS:DATA
BEG: MOV AX,DATA
    MOV DS,AX
    MOV ES,AX
    MOV CX,5
LL1: MOV MSG+5,0DH
    MOV MSG+6,0AH
    MOV MSG+7,'$'
    CALL DISP
    MOV MSG+5,0
    MOV MSG+6,0
    MOV MSG+7,0
    LOOP LL1
    MOV AH,4CH
    INT 21H
DISP PROC
    MOV AH,9
    MOV DX,OFFSET MSG
    INT 21H
    RET
DISP ENDP
CODE ENDS
END BEG
```

以上是该程序的源文件,执行后,在屏幕上显示 5 行 HELLO,语句①~⑧是为了演示 Turbo Debugger 而设置的。下面以此例来介绍汇编语言源程序的开发过程。

(1) 启动 DOS 命令窗口

如果机器安装的是 Windows 操作系统,则用户先用以下两种方法启动 DOS 命令窗口。

方法 1: 在 Windows“开始”菜单中单击“运行”命令,在弹出的“运行”对话框中输入“cmd”,单击“确定”按钮启动 DOS 命令窗口,如图 1.2 所示。

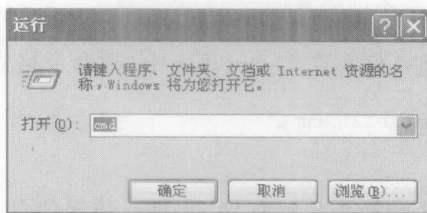


图 1.2 Windows 系统的“运行”对话框

方法 2: 在 Windows“开始”菜单中单击“程序”|“附件”|“命令提示符”选项,也能够启动 DOS 命令窗口。

用户进入 DOS 命令窗口后,应键入“进入子目录”命令进入当前汇编目录(即开发工具的相关文件已复制到此目录下),如:

```
>c:↵          (↵表示按回车键)
>cd tasm↵
```

(2) 编辑

采用文本编辑软件编辑汇编语言源程序,注意保存时,文件的扩展名必须是.asm。尤其如果采用的是 Windows 环境下的如“记事本”等编辑工具,保存时的“保存类型”选项必须选择“所有文件”,如图 1.3 所示。

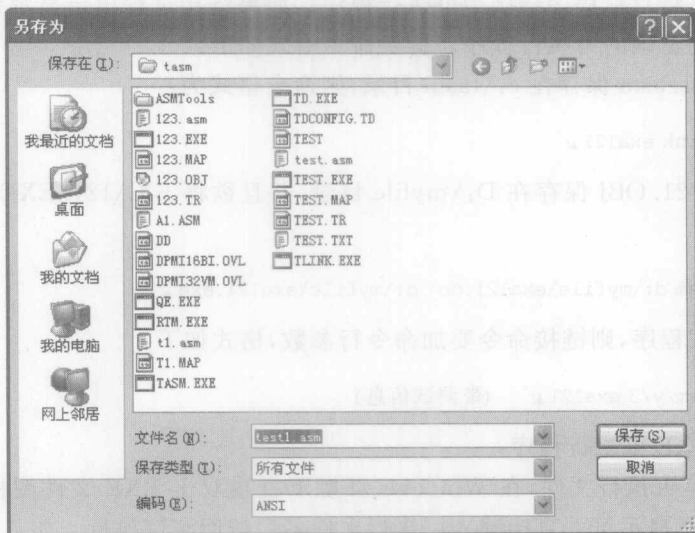


图 1.3 记事本的“另存为”对话框

源程序以及汇编、链接后的目标程序和可执行程序,可以存放在开发工具所在的目录中,如 C:\tasm,也可以集中存放在用户建立的一个文件夹中,例如 D:\myfile 中。这时,所有涉及这些文件的路径前缀就不能省略。

如果 EXA121.ASM 就保存在 C:\tasm 目录中,则命令格式为:

```
C:\tasm>edit exa121.asm
```

如果欲将 EXA121. ASM 保存在 D:\myfile 中,则命令格式为:

```
C:\tasm>edit d:\myfile\exa121.asm
```

(3) 汇编

汇编操作能够将源程序转换为目标程序,并显示错误信息。

如果 EXA121. ASM 保存在 C:\tasm 目录,则命令格式为:

```
C:\tasm>tasm exa121
```

如果 EXA121. ASM 保存在 D:\myfile 目录,并且欲将 EXA121. OBJ 也保存在此目录中,则命令格式为:

```
C:\tasm>tasm d:\myfile\exa121.asm d:\myfile\exa121.obj
```

如果要调试程序,则汇编命令要加命令行参数,格式如下:

```
C:\tasm>tasm/zi exa121 (带调试信息)
```

如果系统给出源程序中的错误信息(错误原因和错误行号),则需要采用编辑软件修改源程序中的错误,直到汇编正确为止。

(4) 链接

链接操作是将目标程序链接为可执行程序。如果链接过程出错显示错误信息,也要修正后才能得到正确的可执行程序。链接命令格式如下:

如果 exa121. asm 保存在 c:\tasm 目录,则命令格式为:

```
C:\tasm>tlink exa121
```

如果 EXA121. OBJ 保存在 D:\myfile 目录,并且欲将 EXA121. EXE 也保存在此目录,格式如下:

```
C:\tasm>tasm d:\myfile\exa121.obj d:\myfile\exa121.exe
```

如果要调试程序,则链接命令要加命令行参数,格式如下:

```
C:\tasm>tasm/v/3 exa121 (带调试信息)
```

(5) 运行 EXE 可执行程序

EXE 文件是可执行文件,在 Windows 环境下直接双击 EXE 文件图标就可执行,也可在 DOS 命令行提示符下直接键入可执行文件名后按回车键执行。如:

```
C:\tasm>exa121
```

(6) 调试程序

如果程序运行错误,则可启动调试软件 TD 对程序进行调试,并找出错误原因。

```
C:\tasm>td exa121 (exa121.asm 保存在 C:\tasm 目录)
```

或

C:\tasm>td d:\myfile\exa121

(exa121.asm 保存在 D:\myfile 目录)

成功启动 TD 之后,TD 进入 MOUDLE 显示模式,屏幕上显示 EXA121. ASM 源程序,一个三角形符号指示出标号为 BEG 的那条指令是启动指令。TD 动态调试程序主要操作命令如下。

① 连续执行程序。按 F9 键(即单击 Run|Run 命令),即可从 BEG 开始连续执行程序。

② 查看执行结果。按 Alt+F5 组合键(即单击 Window|User screen 命令)临时切换到 DOS 屏幕可查看程序的执行结果,即 5 行 HELLO。

③ 使光标重新指向启动指令。当程序运行结束(三角形光标消失),按 Ctrl+F2 组合键(单击 Run|Program rest 命令)可以重新装入程序并使光标指向启动指令。

④ 程序的单步执行。单步操作一次仅执行一条指令,它有三种单步操作命令,它们的区别仅在于“跟踪”的情况不同。

按 F8 键(单击 Run|Step over 命令)单步操作。执行 CALL 和 INT n 指令的时候,“不跟踪”相关的子程序。“不跟踪”的含义是进入相关子程序后,自动地连续执行子程序指令直到返回,所以宏观上看不到跟踪的效果。

按 F7 键(单击 Run|Trace into 命令)单步操作,执行 CALL 指令能够跟进子程序,随即暂停,等待用户的下一步操作,但是遇到 INT n 指令时和按 F8 键一样,进入服务程序后立即自动地连续执行服务程序直到返回。

按 Alt+F7 组合键(单击 Run|Instruction trace 命令)单步操作,是真正意义上的单步操作,执行 CALL 和 INT n 指令进入相关子程序后,立即停止等待用户的下一步操作。

⑤ 断点的设置和取消。按 ↑、↓ 键,使光标指向欲设置断点的指令,再按 F2 键(单击 Breakpoints|Toggle 键),则红色光条覆盖光标处的指令,表示断点设置成功,重复上述操作可以设置多个断点。将光标移到红色光条覆盖的指令,再次按 F2 键,红色光条消失,表明此处断点撤销。设置断点的目的是使程序执行到断点指令时暂停,以便检查执行结果。

⑥ 检查单步执行结果。指令执行后一定会使目标寄存器和状态标志发生变化,如何查看执行效果呢?

在 CPU 窗口调试时,因为界面中有寄存器和标志寄存器显示窗口,所以当前指令执行后,可以从相关的显示窗口中看到结果。

在 MODULE 窗口调试时,单击 View|Registers 命令可弹出寄存器和标志寄存器显示窗口,也可以查看执行结果。

⑦ 检查内存数据区的内容。以上题为例,说明这一操作的实现过程。

如果调试是在 MODULE 窗口进行,则首先单击 View|Dump 命令,弹出内存数据显示窗口,接着再按 Ctrl+S 组合键弹出一个对话框,用户键入“HELLO”这一字符串是例 1.2.1 用户数据段中设置的内容,TD 根据用户的提示立即找出并显示用户程序数据区。

做好以上准备工作之后,按 F8 键单步执行指令,就可看到用户数据段内容的变化。
例 1.2.1 中的语句①~⑧就是为此项调试而设置的。

4. 实验项目

【实验 1.2.1】 汇编语言编程过程的练习。

请将例 1.2.1 的源程序通过一个编辑软件输入计算机并加以保存,命名为: EXA121.ASM。然后调用 TASM 和 TLINK 完成编译和链接,生成可执行文件 EXA121.EXE。试着在当前目录下运行程序 EXA121.EXE。最后用 TD 将 EXA121.EXE 调入 TD 的调试界面,掌握调试过程。

1.3 汇编语言语法实验

1. 实验说明

在 1.2 节的基础上进一步掌握汇编语言程序开发过程。

2. 实验目的和要求

进一步学习汇编语言源程序的编辑、汇编、目标文件的链接和可执行文件的执行全过程;掌握编辑软件、TASM、TLINK 和 TD 的使用方法;掌握汇编语言的语法规则。

3. 实验项目

【实验 1.3.1】 排除语法错误。

下面给出的是一个通过比较法完成 8 位二进制数转换成十进制数送屏幕显示功能的汇编语言源程序,但有很多语法错误。要求实验者按照原样对源程序进行编辑,汇编后,根据 TASM 给出的错误信息对源程序进行修改,直到没有语法错误为止。然后进行链接,并执行相应的可执行文件。正确的执行结果是在屏幕上显示: $25+9=34$ 。

【程序清单】

```
;FILENAME: EXA131.ASM
.486
DATA    SEGMENT    USE16
SUM      DB        ?,?,
MSG      DB        '25+9='
          DB        0,0
N1       DB        9,F0H
N2       DW        25
DATA    ENDS
CODE    SEGMENT    USE16
          ASSUME    CS: CODE,DS: DATA
BEG:    MOV        AX,DATA
```

```

MOV     DS,AX
MOV     BX,OFFSET SUM
MOV     AH,N1
MOV     AL,N2
ADD     AH,AL
MOV     [BX],AH
CALL    CHANG
MOV     AH,9
MOV     DX,OFFSEG MEST
INT     21H
MOV     AH,4CH
INT     21H

CHANG:  PROC
LAST:   CMP     [BX],10
JC      NEXT
SUB     [BX],10
INC     [BX+7]
JMP     LAST
NEXT:    ADD     [BX+8],SUM
        ADD     [BX+7],30H
        ADD     [BX+8],30H
        RET
CHANG:  ENDP
CODE    ENDS
END      BEG

```



图 2.1.1 例 2.1.1 程序流程图

图 2.1.1 例 2.1.1 程序流程图

384

第2章

chapter 2

结构化程序设计实验

结构化程序设计是指具有结构性的编程方法。采用结构化程序设计方法编程,旨在提高所编程序的质量。自顶向下、逐步精化方法有利于在每一抽象级别上尽可能保证所编程序的正确性;按模块组装方法编程以及所编程序只含顺序、分支和循环三种程序,可使程序结构良好、易读、易理解和易维护,并易于保证及验证程序的正确性。任一流程图均可利用循环和嵌套等价地改写成只含顺序、分支和递归的程序,并且每种程序只有一个入口和一个出口。汇编语言程序设计的主要方法,包括顺序、分支、循环、子程序和宏指令的设计等。

2.1 顺序程序设计

1. 实验说明

顺序程序是一种最简单也是最基本的结构形式,是最简单的序列结构,程序上没有用到分支和循环,没有控制转移类指令,它的执行流程与指令的排列顺序完全一致,顺序程序设计是所有程序设计的基础。

2. 实验目的和要求

掌握顺序程序的编程方法。

3. 实验示例

【例 2.1.1】 采用顺序编程方法实现在屏幕上显示字符串“Enjoy programming in TASM”。

【程序流程图】

程序流程图如图 2.1 所示。

【程序清单】

```
;FILENAME: EXA211.ASM  
.486
```

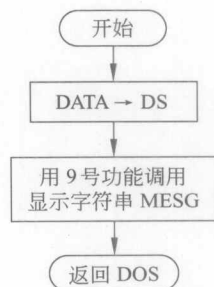


图 2.1 例 2.1.1 程序流程图