



普通高等教育“十一五”国家级规划教材

刘云玲 主编

微机原理与接口技术 实验指导

21世纪计算机科学与技术实践型教程
丛书主编 陈明

清华大学出版社





普通高等教育“十一五”国家级规划教材

刘云玲 主编

微机原理与接口技术 实验指导

纪
计
算
机
科
学
与
技
术
实
践
型
教
程

丛书主编 陈明

清华大学出版社
北 京

内 容 简 介

本书包括软件实验、硬件实验和综合实践三大部分。

软件和硬件两部分实验内容紧密结合理论教学,涵盖了微机原理和接口技术中的基本主题,包括汇编语言程序设计开发、DEBUG 软件、存储器、并口 8255A、串口 8250A、中断控制器 8259A、数/模和模/数转换器等。书中针对这些基础知识点设计了一系列趣味性的实验,加入了声音、动画元素,并提供了程序流程图和参考程序。为提高学生动手和动脑能力,软件和硬件部分都增加了拓展实验,最后提供了设计性实验和设计思路。

综合实践部分包含了 5 个设计实验:智能电脑钟、模拟电子琴、可视可听汽车自动报站器、LED 大屏幕显示器和人体健康监测器。这些实验内容来源于实际生活,有趣而生动,能最大限度地激发学生动手动脑的欲望,学生在综合实践部分可以练习如何综合运用学到的知识来解决这类身边的实际问题。书末还附有常用芯片引脚图和汇编过程错误信息表,供使用者参考。

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

微机原理与接口技术实验指导/刘云玲主编. --北京:清华大学出版社,2014

21 世纪计算机科学与技术实践型教程

ISBN 978-7-302-34978-5

I. ①微… II. ①刘… III. ①微型计算机—理论—高等学校—教学参考资料 ②微型计算机—接口技术—高等学校—教学参考资料 IV. ①TP36

中国版本图书馆 CIP 数据核字(2013)第 316720 号

责任编辑:谢 琛

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:杨 艳

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:保定市中华美凯印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:6.5

字 数:148 千字

版 次:2014 年 1 月第 1 版

印 次:2014 年 1 月第 1 次印刷

印 数:1~2500

定 价:20.00 元

产品编号:052033-01

《21 世纪计算机科学与技术实践型教程》

序

21 世纪影响世界的三大关键技术：以计算机和网络为代表的信息技术；以基因工程为代表的生命科学和生物技术；以纳米技术为代表的新型材料技术。信息技术居三大关键技术之首。国民经济的发展采取信息化带动现代化的方针，要求在所有领域中迅速推广信息技术，导致需要大量的计算机科学与技术领域的优秀人才。

计算机科学与技术的广泛应用是计算机学科发展的原动力，计算机科学是一门应用科学。因此，计算机学科的优秀人才不仅应具有坚实的科学理论基础，而且更重要的是能将理论与实践相结合，并具有解决实际问题的能力。培养计算机科学与技术的优秀人才是社会的需要、国民经济发展的需要。

制订科学的教学计划对于培养计算机科学与技术人才十分重要，而教材的选择是实施教学计划的一个重要组成部分，《21 世纪计算机科学与技术实践型教程》主要考虑了下述两方面。

一方面，高等学校的计算机科学与技术专业的学生，在学习了基本的必修课和部分选修课程之后，立刻进行计算机应用系统的软件和硬件开发与应用尚存在一些困难，而《21 世纪计算机科学与技术实践型教程》就是为了填补这部分空白。将理论与实际联系起来，使学生不仅学会了计算机科学理论，而且也学会了应用这些理论解决实际问题。

另一方面，计算机科学与技术专业的课程内容需要经过实践练习，才能深刻理解和掌握。因此，本套教材增强了实践性、应用性和可理解性，并在体例上做了改进——使用案例说明。

实践型教学占有重要的位置，不仅体现了理论和实践紧密结合的学科特征，而且对于提高学生的综合素质，培养学生的创新精神与实践能力有特殊的作用。因此，研究和撰写实践型教材是必需的，也是十分重要的任务。优秀的教材是保证高水平教学的重要因素，选择水平高、内容新、实践性强的教材可以促进课堂教学质量的快速提升。在教学中，应用实践型教材可以增强学生的认知能力、创新能力、实践能力以及团队协作和交流表达能力。

实践型教材应由教学经验丰富、实际应用经验丰富的教师撰写。此系列教材的作者不但从事多年的计算机教学，而且参加并完成了多项计算机类的科研项目，他们把积累的经验、知识、智慧、素质融于教材中，奉献给计算机科学与技术的教学。

我们在组织本系列教材过程中，虽然经过了详细的思考和讨论，但毕竟是初步的尝试，不完善甚至缺陷不可避免，敬请读者指正。

本系列教材主编 陈明

2005 年 1 月于北京

前 言

本实验指导包括软件实验、硬件实验和综合实践三大部分。软、硬件实验部分是在讲述完基础实验后,让学生尝试趣味性的拓展实验,提高学生的学习兴趣。综合实践部分为有趣而富于挑战性的设计实验,可以帮助激发学生的潜能,提高学生在整个学习过程中的主观能动性。

本书第1章到第4章的软件和硬件两部分实验内容紧密结合理论教学内容,涵盖了微机原理和接口技术中的基本主题,包括汇编语言程序设计开发、DEBUG 软件、存储器、并口芯片 8255、串口芯片 8250、中断控制器 8259、数/模和模/数转换器等。书中针对这些基础知识点设计了一系列有趣的实验,并提供了程序流程图和参考程序。

为提高学生动手和动脑能力,第5章的综合实践部分包含了模拟电子琴、数字录音机、智能电脑钟、可视可听汽车自动报站器、LED 大屏幕显示器和人体健康监测器等设计性实验。这些实验内容来源于实际生活,有趣而生动,能最大限度地激发学生动手动脑的欲望,学生可以综合运用学到的知识来解决这类身边的实际问题。

书末还附有汇编过程错误信息表和本书硬件实验系统的 HQFC 集成开发环境常用命令等,供使用者参考。

在使用本书进行实验教学时,老师可根据教学安排任意选做其中的一些实验,也可根据学生的具体情况灵活安排实验内容。

本书由刘云玲副教授主编,刘彤参与编写了第1章,刘云玲编写了第2、3、5章,袁钢参与编写了第4章。全书由刘云玲统稿,由吉海彦主审。

鉴于编者水平有限,不当之处敬请批评指正。

编 者

2013 年 11 月

目 录

第 1 章 汇编语言程序设计与开发	1
1.1 汇编语言程序设计开发过程	1
1.1.1 编辑过程	1
1.1.2 汇编过程	2
1.1.3 连接过程	3
1.2 运行及调试	3
1.2.1 运行可执行文件	3
1.2.2 DEBUG 软件的使用	4
1.2.3 DEBUG 调试练习	8
1.3 汇编语言程序的建立和执行例	10
1.4 思考题	14
第 2 章 软件实验	15
2.1 基础实验	15
2.1.1 实验一 清零实验	15
2.1.2 实验二 拆字和拼字实验	16
2.1.3 实验三 正负数分流实验	17
2.1.4 实验四 数据的排序实验	20
2.1.5 实验五 两个多位十进制数相加实验	21
2.1.6 实验六 数据块传送实验	24
2.1.7 实验七 字符匹配	25
2.1.8 实验八 十六进制数转换为二进制数实验	29
2.2 软件拓展实验	31
2.2.1 实验一 响铃实验	31
2.2.2 实验二 屏幕时钟实验	32
第 3 章 硬件实验平台 TPC-ZK	37
3.1 TPC-ZK 实验系统介绍	37

3.2	TPC-ZK 实验系统结构与主要电路	37
3.2.1	实验系统结构图	37
3.2.2	实验台上包含的主要电路	37
3.2.3	实验台跳线开关	43
3.2.4	用户扩展实验区	43
3.2.5	直流稳压电源	44
3.2.6	TPC-ZK 实验系统开关及跳线说明	44
3.3	与 PC 的连接——USB 接口	45
3.3.1	USB 模块结构	45
3.3.2	USB 模块功能	45
3.3.3	USB 模块的对外接口	45
3.3.4	USB 模块跳线说明	46
3.4	软件开发环境	46
第 4 章	硬件实验	47
4.1	硬件基础实验	47
4.1.1	实验一 I/O 地址译码	47
4.1.2	实验二 存储器读写实验	49
4.1.3	实验三 可编程并行接口 8255	51
4.1.4	实验四 七段数码管	52
4.1.5	实验五 可编程定时器/计数器(8253)	55
4.1.6	实验六 扩展中断控制器 8259	56
4.1.7	实验七 扩展 DMA 控制器 8237	60
4.1.8	实验八 串行通信 8251	63
4.1.9	实验九 模/数转换器 ADC0809	66
4.1.10	实验十 数/模转换器	69
4.2	硬件拓展实验	71
4.2.1	实验一 交通灯控制实验	71
4.2.2	实验二 竞赛抢答器	73
4.2.3	实验三 模拟电子琴	75
第 5 章	综合实践	79
5.1	综合性实验	79
5.1.1	乐曲设计实验	79
5.1.2	录音机实验	81
5.2	设计性实验	83
5.2.1	报时时钟	83
5.2.2	LED 大屏幕显示器	83

5.2.3 可视可听汽车自动报站器	83
附录 A 汇编错误信息说明	85
附录 B DEBUG 命令一览表	89
附录 C HQFC 开发环境常用调试命令	91
参考文献	93

第 1 章 汇编语言程序设计与开发

1.1 汇编语言程序设计开发过程

要建立和运行汇编语言程序,系统盘上应该首先有如下文件:

EDIT. COM(或其他全屏幕编辑软件)	编辑程序
MASM. EXE	汇编程序
LINK. EXE	链接程序
DEBUG. COM	调试程序

建立和运行汇编程序的基本步骤如图 1-1 所示。

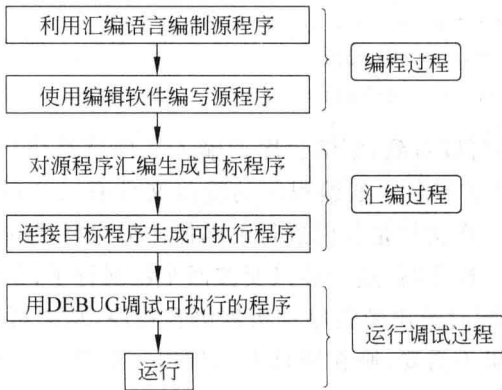


图 1-1 建立和运行汇编程序的基本步骤

下面以建立和执行用户程序 ABC. EXE 为例来说明上机过程。

1.1.1 编辑过程

源程序就是用户用汇编语言的语句编写的程序,它不能被机器识别。源程序一般用 EDIT 或者记事本来建立,而且汇编语言源程序必须以 ASM 为文件后缀名。

首先进入 Windows 操作系统的命令提示符状态,命令提示符状态下键入 DOS 命令:

C:\>EDIT ABC. ASM(回车)

此时用户可以编写源程序 ABC. ASM,一定要严格按照汇编语言书写风格及段结构的格式来编写。EDIT 的使用方法可以查阅相关手册。打开记事本,录入源文件,将其保存为

ABC. ASM。

1.1.2 汇编过程

源程序建立以后,就可以用汇编程序 MASM. EXE 进行汇编。所谓汇编,实际上就是把以 ASM 为后缀名的源文件转换成用二进制代码表示的目标文件,目标文件以 OBJ 为后缀名。汇编过程中,汇编程序对源文件进行两次扫描,如果源程序中有语法错误,则汇编过程结束后,汇编程序会指出源程序中的错误,这时,用户可以在编辑环境中修改源程序中的错误,然后再次进行汇编,最后得到没有语法错误的 OBJ 文件。

例如,对 ABC. ASM 的汇编过程如下:

```
C:\>MASM ABC.ASM(回车)
```

如果源程序没有语法错误,此时,汇编程序会给出如下回答:

```
Microsoft(R)Macro Assembler Version 5.00
Copyright(C)Microsoft Corp 1981-1985, 1987. All rights reserved.
Object filename [abc. OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
51526-402346 Bytes symbol space free
0 Warning Errors
0 Severe Errors
```

从上面的操作过程中可以看到,汇编程序的输入文件就是用户编写的汇编语言源程序,它必须以 ASM 为文件扩展名。汇编程序的输出文件有三个,第一个是目标文件,它以 OBJ 为扩展名。产生 OBJ 文件是我们进行汇编操作的主要目的,所以这个文件是一定要产生,也一定会产生的。操作时,这一步只要按回车键就行了;第二个是列表文件,它以 LST 为扩展名,列表文件同时给出源程序和机器语言程序,从而可以使调试变得方便,列表文件是可有可无的,如果不需要,则在屏幕上出现提示信息“[NUL. LST]:”时,按回车键即可,如果需要,则输入文件名后按回车键;第三个是交叉符号表,以 CRF 为扩展名。此表给出了用户定义的所有符号,对每个符号都列出了定义它的所在行号和引用它的所在行号,并在定义行号上加上“#”号。同列表文件一样,交叉符号表也是为了便于调试而设置的,对于一些规模较大的程序,交叉符号表为调试工作带来很大方便。当然,交叉符号表也是可有可无的,如果不需要,那么在屏幕上出现提示信息“[NUL. CRF]:”时,按回车键即可。

汇编过程结束时,会给出源程序中的警告性错误“Warning Errors”和严重错误“Severe Errors”,前者指一般性错误,后者指语法性错误。当存在这两类错误时,屏幕上除指出错误个数外,还给出错误信息代号(汇编过程错误信息说明见附录 A),程序员可以通过查找手册进一步弄清错误的性质。如果汇编过程中发现有错误,则程序员应该重新用编辑命令修改错误,再进行汇编,最终直到汇编正确通过。要指出的是在汇编过程中只能指出源程序中的语法错误,并不能指出算法错误和其他错误。

1.1.3 连接过程

汇编过程是根据源程序产生出二进制的目标文件(Obj 文件),但 Obj 文件用的是浮动地址,它不能直接上机执行,所以还必须使用链接程序(LINK)将 Obj 文件转换成可执行的 EXE 文件。LINK 命令还可以将某一个目标文件和其他多个模块(这些模块可以由用户编写的,也可以是某个程序库中存在的)链接起来。

具体操作如下(以对 ABC. OBJ 进行链接为例):

```
C:\>LINK ABC(回车)
```

此时,在屏幕上见到如下回答信息:

```
IBM Personal Computer link
```

```
Version 2.00 (C) Copyright IBM Corp 1981,1982,1983
```

```
Run File [ABC.EXE]:
```

```
List File [NUL.MAP]:
```

```
Libraries [.LIB]:
```

```
Warning:NO STACK Segment
```

LINK 命令有一个输入文件,即由汇编程序产生的 Obj 文件。有时用户程序用到库函数,此时对于提示信息“Libraries[. LIB]”,要输入库名。

LINK 过程产生两个输出文件,一个是扩展名为 EXE 的执行文件,产生此文件当然是 LINK 过程的主要目的;另一个是扩展名为 MAP 的列表分配文件,有人也称它为映像文件,MAP 文件也是可有可无的,它给出每个段在内存中的分配情况。比如某一个列表分配文件为如下内容:

```
Warning : No Stack Segment
```

Start	Stop	Length	NameClass
0000H	0015H	0016H	CODE
0020H	0045H	0026H	DATA
0050H	0061H	0012H	EXTRA
Origin	Group		
Program	entry	Point	at 0000 : 0000

从 LINK 过程的提示信息中,可看到最后给出了一个“无堆栈段”的警告性错误,这并不影响程序的执行。当然,如果源程序中设置了堆栈段,则无此提示信息。

1.2 运行及调试

1.2.1 运行可执行文件

有了 EXE 文件后,就可以执行程序了,此时,只要打入文件名即可。仍以 ABC

为例:

```
C:\>ABC(回车)
```

1.2.2 DEBUG 软件的使用

实际上,大部分程序必须经过调试阶段才能纠正程序设计中的错误,从而得到正确的结果。所谓调试阶段,就是用调试程序(DEBUG 程序)发现错误,再经过编辑、汇编、链接纠正错误。DEBUG.COM 是专为 8086/8088 汇编语言设计的一种调试工具,是汇编语言程序设计者必须掌握的工具。具有显示、修改、动态跟踪等功能。使用它可以对 *.COM、*.EXE 文件进行动态跟踪。关于 DEBUG 程序中的各种命令,可参阅 DOS 手册,本实验指导书附录 B 有一些常用命令的介绍,下面给出最常用的几个命令的用法。

首先进入 DEBUG 程序并装入要调试的程序 ABC.EXE,操作命令如下:

```
C:\>DEBUG ABC.EXE(回车); 进入 DEBUG,并装配 ABC.EXE 文件
```

此时,屏幕上出现一个短划线“-”,表明已进入 DEBUG 状态。以下是一些常用 DEBUG 命令介绍。

1. 程序输入汇编命令(A 命令)

在输入程序应注意以下几点:

- (1) 输入的数字必须是十六进制数,并且不输入 H。
- (2) 分号后面的内容(注释)不要输入。
- (3) 不输入标号,条件转移指令中的标号用偏移地址代替。方法是:先用一个离本转移指令的 IP 不远的但不一定准确的偏移地址代替标号,待程序输入且完成反汇编后获得标号所在处的准确的偏移地址时再修改。

在 DEBUG 提示符下键入 A 即进入汇编输入状态:

```
-A(回车)
```

键入下面的汇编程序

```
13F2:0100    MOV     AX,2000
13F2:0103    MOV     DS,AX
13F2:0105    MOV     SS,AX
13F2:0107    MOV     BX,2050
13F2:010A    MOV     SI,BX
13F2:010C    MOV     DI,3050
13F2:010F    MOV     SI,DI
13F2:0111    MOV     SP,5FFF
13F2:0114    MOV     CL,19
13F2:0116    MOV     BL,CL
13F2:0118    MOV     AH,F0
13F2:011A    MOV     CH,AH
```

```

13F2:011C      MOV     BYTE PTR [DI],40
13F2:011F      MOV     WORD PTR [SI],0100
13F2:0123      MOV     DL,[SI+01]
13F2:0126      MOV     DH,[SI+01]
13F2:0129      MOV     AL,[SI+01]
13F2:012C      MOV     WORD PTR [BX+SI],0022
13F2:0130      MOV     [BX+SI+02],BX
13F2:0133      MOV     [BX+SI+02],BX
13F2:0136      MOV     BP,[BX+DI+02]
13F2:0139      MOV     [BX+DI],BX
13F2:013B      MOV     AX,[BP+DI]
13F2:013D      MOV     BL,AL
13F2:013F      MOV     ES,BX
13F2:0141      PUSH    BX
13F2:0142      PUSH    DI
13F2:0143      PUSH    DI
13F2:0144      POP     CX
13F2:0145      POP     DX
13F2:0146      XCHG    BP,AX
13F2:0147      XCHG    DH,BL
13F2:0149      SAHF
13F2:014A      CMC
13F2:014B      LAHF
13F2:014C      STD
13F2:014D      CLI
13F2:014E      PUSHF
13F2:014F      CLD
13F2:0150      POPF
13F2:0151      INT     21
13F2:0153      ^C;      "C"代表"Ctrl+C",表示汇编命令结束,重新回到 DEBUG 提示符"-
           状态

```

2. 反汇编命令(U 命令)

-U 0100 0153(回车)

则屏幕显示如下信息:

```

13F2:0100 B80020      MOV     AX,2000
13F2:0103 8ED8        MOV     DS,AX
13F2:0105 8ED0        MOV     SS,AX
13F2:0107 BB5020      MOV     BX,2050
13F2:010A 89DE        MOV     SI,BX
13F2:010C BF5030      MOV     DI,3050
13F2:010F B119        MOV     SI,DI

```

```

      :
13F2:014E 9C      PUSHF
13F2:014F FC      CLD
13F2:0150 9D      POPF
13F2:0151 CD21     INT 21

```

3. 写文件于磁盘中

在程序键入后,应及时存盘,以免发生错误操作时,丢失程序。写文件前先用 N 命令给文件命名。若文件名定为 LXA.COM,则具体操作如下:

-N LXA.COM(回车)

定义好名称后,再在 CX 中存放文件长度即字节数:

文件长度=终止地址-起始地址+1

具体操作如下:

```

-R CX(回车)      ;利用 R 命令显示 CX 寄存器当前内容
      CX 0000      ;屏幕显示 CX 的现存内容
:0053(回车)      ;输入文件长度
-W(回车)         ;利用 W 命令将文件写入
      Writing 0053 bytes ;屏幕显示写入完成

```

若文件长度>65536,则用 BX 和 CX 存放长度。返回 DOS 后用 DIR 命令可以见到 LXA.COM 文件已经存在。

4. 程序跟踪调试

调试前可先装入已存好的程序。对已存在的 COM 文件(或 EXE 文件)可以这样装入文件:键入

C:\>DEBUG LXA.COM(回车)

出现提示符“-”。此时可以用 R 命令查看 DEBUG 对 .COM 文件(或 .EXE 文件)的段寄存器、IP 及 F 寄存器的初始化值。

```

-R(回车)
AX=0000  BX=0000  CX=0000  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=13F2  ES=13F2  SS=13F2  CS=13F2  IP=0100  NV UP DI PL NZ NA PO NC
13F2:0100  B80020      MOV     AX, 2000

```

可以用 T 命令来进行单步和多步指令的运行,运行指令后将显示各寄存器内容和下一条将要执行的指令:

-T=100(回车) ;从 CS:0100H 开始单步执行,显示寄存器值

```

AX=2000  BX=0000  CX=0153  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=13F2  ES=13F2  SS=13F2  CS=13F2  IP=0103  NV UP DI PL NZ NA PO NC

```

```
13F2:0103 8ED8 MOV DS,AX
```

-T(回车) ;从当前 IP 开始执行一条指令

```
AX=2000 BX=0000 CX=0153 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
```

```
DS=2000 ES=13F2 SS=13F2 CS=13F2 IP=0105 NV UP DI PL NZ NA PO NC
```

```
13F2:0105 8ED0 MOV SS,AX
```

-T4(回车) ;用 Tn 命令来跟踪几条指令,n 为需跟踪的指令条数

```
AX=2000 BX=2050 CX=0153 DX=0000 SP=FFFE BP=0000 SI=3050 DI=3050
```

```
DS=2000 ES=13F2 SS=2000 CS=13F2 IP=0111 NV UP DI PL NZ NA PO NC
```

```
13F2:0111 BCFF5F MOV SP,5FFF
```

```
AX=2000 BX=2050 CX=0153 DX=0000 SP=5FFE BP=0000 SI=3050 DI=3050
```

```
DS=2000 ES=13F2 SS=2000 CS=13F2 IP=0114 NV UP DI PL NZ NA PO NC
```

```
13F2:0114 B119 MOV CL,19
```

```
AX=2000 BX=2050 CX=0119 DX=0000 SP=5FFE BP=0000 SI=3050 DI=3050
```

```
DS=2000 ES=13F2 SS=2000 CS=13F2 IP=0116 NV UP DI PL NZ NA PO NC
```

```
13F2:0116 88CB MOV BL,CL
```

```
AX=2000 BX=2019 CX=0119 DX=0000 SP=5FFE BP=0000 SI=3050 DI=3050
```

```
DS=2000 ES=13F2 SS=2000 CS=13F2 IP=0118 NV UP DI PL NZ NA PO NC
```

```
13F2:0118 B4F0 MOV AH,F0
```

用 G 命令执行指定区域中的指令:

-G=0100 013F(回车) ;从 IP=0100H 开始执行到 IP=013FH,并显示寄存器的内容

```
AX=2019 BX=2019 CX=0119 DX=0101 SP=SFFF BP=2019 SI=3050 DI=3050
```

```
DS=2000 ES=13F2 SS=2000 CS=13F2 IP=013F NV UP DI PL NZ NA PO NC
```

```
13F2:013F 8EC3 MOV ES,BX
```

用 D 命令可查看指定存储单元后的 128 个存储单元的内容。如若需查看数据段偏移地址为 3050H 的存储单元的内容,可用下面的命令。从中可看出 3050H 单元的内容为 00H。

-D DS : 3050(回车)

```
2000:3050 00 01 00 00 96 00 00 00-B4 00 00 00 D3 00 00 00 .....4...S...
```

```
2000:3060 F2 00 00 00 10 01 00 00-2F 01 00 00 4D 01 00 00 r...../..M...
```

```
2000:3070 6C 01 00 00 20 09 ZD 0D-SD 00 00 00 SD 00 00 00 1...-.]...].-.
```

```
2000:3080 20 00 09 00 ZD 00 0D 00-5D 00 00 00 SD 00 00 00 ...-...]....]...
```

```
2000:3090 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .'...-.-.-.---..
```

```
2000:30A0 ZE 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ...-.....
```

```
2000:30B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....-.-.-...
```

```
2000:30C0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....-.-.-....
```

5. 退出

从 DEBUG 退出时,使用如下命令:

-Q(回车)

1.2.3 DEBUG 调试练习

修改并调试以下程序,使之完成向以 3000H 开始的内存单元中存入 31 个先自然递增然后又自然递减的数据(00H~0FH~00H)的功能。程序从 CS: 0100H 开始存放。调试完成后程序命名为 LXB.COM 并存盘。

步骤如下:

(1) 用汇编 A 命令输入程序(L1、L2 是标号,不输入。指令 JNZ L1 和 LOOP L2 中的 L1 和 L2 用标号所指指令的偏移地址代替)。

```
MOV AX, 3000
MOV DS, AX
MOV SI, 0
MOV CL, 0FH
XOR AX, AX
L1: MOV [SI], AL
    INC AL
    DEC CL
    JNZ L1
    MOV CX, 10H
L2: MOV [SI], AL
    INC SI
    DEC AL
    LOOP L2
    INT 20
```

(2) 用反汇编 U 命令显示程序及目标码。

-U 0100(回车)

13F2 : 0100 B80030	MOV AX, 3000
13F2 : 0103 SED8	MOV DS, AX
13F2 : 0105 BE0000	MOV SI, 0000
13F2 : 0108 B10F	MOV CL, 0F
13F2 : 010A 31C0	XOR AX, AX
13F2 : 010C 8804	MOV [SI], AL
13F2 : 010E FEC0	INC AL
13F2 : 0110 FEC9	DEC CL
13F2 : 0112 75F8	JNZ 010C
13F2 : 0114 B91000	MOV CX, 0010

```

13F2 : 0117 8804      MOV [SI], AL
13F2 : 0119 46        INC SI
13F2 : 011A FEC8      DEC AL
13F2 : 011C E2F9      LOOP 0117
13F2 : 011E CD20      INT 20

```

(3) 用 W 命令将文件存入磁盘,文件名定为 LXB.COM。

(4) 用 G 命令运行程序到 INT 20H(注意:不执行 INT 20H 命令)。

-G=0100 011E (回车)

```

AX=00FF BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0010 DI=0000
DS=3000 ES=13F2 SS=13F2 CS=13F2 IP=011E NV UP DI NG NZ AC PE NC
13F2 : 011E CD20      INT 20

```

(5) 用 D 命令查看数据段中 0000H 处数据,检查是否将数据按要求存入该区域。

-D DS : 0000 (回车)

```

3000:0000  0F 0E 0D 0C 0B 0A 09 08-07 06 05 04 03 02 01 00 - .....- .
3000:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....- -
3000:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....- -
3000:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ....- -
3000:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ...'- -
3000:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ..- -
3000:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....-
3000:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 - ....- -

```

检查内存可发现数据只有从 0FH 到 00H,说明程序存在一定的问题。

(6) 用 T 命令单步执行程序(执行到 INT 20H)并检查问题所在。

可看出在程序中少了一条“INC SI”指令,应在适当的位置加入该指令。

(7) 修改程序。

用块移动指令 M 将 010EH 以后的程序下移两个字节后插入 INC SI 指令。

-M CS:010E 011F CS:0110 (回车)

-A 010E (回车)

```

13F2:010E  INC SI
13F2:010F  NOP
13F2:0110  ^C

```

用反汇编命令检查程序,特别注意转移指令中偏移地址是否有变化,若有可用反汇编 A 命令进行修改后再反汇编。反汇编后程序如下:

```

13F2 : 0100 B80030      MOV AX, 3000
13F2 : 0103 8ED8        MOV DS, AX
13F2 : 0105 BE0000      MOV SI, 0000
13F2 : 0108 B10F        MOV CL, 0F

```