



普通高等教育“十一五”国家级规划教材配套教材
PUTONG GAODENG JIAOYU SHIYIWU GUOJIAJI GUIHUA JIAOCAI PEITAO JIAOCAI

16/32 WEI WEIJI YUANLI
JI JIEKOU JISHU SHIYAN ZHIDAOSHU

16/32位微机原理 及接口技术实验指导书

刘淑平 朱有产 编著



中国电力出版社
<http://jc.cepp.com.cn>



普通高等教育“十一五”国家级规划教材配套教材
PUTONG GAODENG JIAOYU SHIYIWU GUOJIAJI GUIHUA JIAOCAI PEITAO JIAOCAI

16/32 WEI WEIJI YUANLI
JI JIEKOU JISHU SHIYAN ZHIDAOSHU

16/32位微机原理 及接口技术实验指导书

编 著 刘淑平 朱有产
主 审 许 会



中国电力出版社
<http://jc.cepp.com.cn>

Electrical Engineering

内 容 提 要

本书为与《普通高等教育“十一五”国家级规划教材 16/32 位微机原理及接口技术》相配套的实验指导书。

全书共分为 5 章,系统地介绍了汇编语言上机的 DOS 和 Windows 环境、常用 Debug 调试命令、汇编语言基本结构程序的设计、字符串处理、数制转换、DOS 系统功能调用、软件综合性实验。在硬件方面介绍了微机接口教学实验系统构成及主要操作、存储器扩充实验、中断控制器 8259A 实验、可编程计数器/定时器 8254 实验、并行 I/O 接口 8255A 应用实验、DMA 控制器 8237 应用实验、串行接口 8251A 应用实验、D/A 及 A/D 转换实验,最后介绍了 8255A、8259A、8254 应用综合实验。本书内容简明扼要、由浅入深、循序渐进,结构层次清晰,并且配有图示、提示、部分实验程序及程序注释,便于教学与自学。

本书可作为普通高等院校电气信息类相关专业的本科教材,也可作为成人高等教育、高职高专和各类培训班的教材,还可以作为研究生和从事微机测控及接口技术应用的工程技术人员的参考书。

图书在版编目 (CIP) 数据

16/32 位微机原理及接口技术实验指导书 / 刘淑平, 朱有产
编著. —北京: 中国电力出版社, 2010.8

普通高等教育“十一五”国家级规划教材配套教材
ISBN 978-7-5123-0759-9

I. ①1… II. ①刘… ②朱… III. ①微型计算机—实验—高等学校—教学参考资料 ②微型计算机—接口设备—实验—高等学校—教学参考资料 IV. ①TP36-33 ②TP364.7-33

中国版本图书馆 CIP 数据核字 (2010) 第 158377 号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://jc.cepp.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

*

2010 年 8 月第一版 2010 年 8 月北京第一次印刷
787 毫米×1092 毫米 16 开本 5.5 印张 127 千字
定价 10.00 元

敬告读者

本书封面贴有防伪标签,加热后中心图案消失
本书如有印装质量问题,我社发行部负责退换
版权专有 翻印必究

前 言

本书为与《普通高等教育“十一五”国家级规划教材 16/32 位微机原理及接口技术》相配套的实验指导书。本书以 16 位微机为主、32 位微机为辅组织内容，兼顾基础性、先进性和前沿性。以 CPU 为核心建立微型计算机系统的整体概念，理解硬、软件间的辩证关系，对内容进行了重构。对工科非计算机专业教学软件、硬件“两手”都要抓，以提高理论联系实际能力、自主动手能力和创新能力。本书主要有下列特点：

(1) 在掌握微型计算机基本原理的基础上，侧重介绍如何自己构建微计算机系统。

(2) 内容组织由浅入深、由理论到应用，逐步介绍构建计算机系统组成部分。

(3) 通过基本结构程序的设计与调试，进一步理解指令系统和汇编语言程序设计过程和应用、程序调试基本方法和手段。

(4) 借助 TD-PITE 实验箱，通过常用外围接口芯片应用实验的编程和操作过程，使学生进一步弄清外围接口芯片的工作原理和实际应用，提高了学生的动手能力。

本书共分为 5 章。第 1 章对微型计算机系统的汇编语言上机的 DOS 和 Windows 环境进行了介绍，特别是对常用 Debug 调试命令进行了详细介绍。第 2 章主要介绍了汇编语言基本结构程序的设计和编程及字符串处理、数制转换、DOS 系统功能调用等专题实验，最后要求设计一个综合性实验。第 3 章主要是微机接口教学实验系统的介绍，简要介绍了 TD-PITE 实验装置的构成和主要操作。第 4 章介绍了存储器扩充实验、中断控制器 8259A 实验、可编程计数器/定时器 8254 实验、并行 I/O 接口 8255A 应用实验、DMA 控制器 8237 应用实验、串行接口 8251A 应用实验、D/A 与 A/D 转换实验。第 5 章介绍了 8255A、8259A、8254 应用综合实验。

本书由刘淑平、朱有产编著，其中，第 1、2 章由朱有产编写，第 3、4、5 章由刘淑平编写。本书的编写得到华北电力大学微机原理教研室、微机原理实验室和西安唐都科教仪器公司有关老师的支持，在此表示诚挚的谢意。全书由沈阳工业大学许会教授审阅，并提出了大量宝贵意见，在此表示最诚挚的感谢。

限于笔者水平，书中难免有错误和不妥之处，敬请广大同仁和读者提出宝贵意见。

编 者
2010 年 6 月

目 录

前言	
第 1 章 汇编语言程序设计概述	1
1.1 汇编语言开发环境.....	1
1.2 汇编语言实验过程详细步骤.....	2
第 2 章 80X86 汇编语言实验内容	8
2.1 系统认识实验.....	8
2.2 顺序程序设计实验.....	15
2.3 分支程序设计实验.....	15
2.4 循环程序设计实验.....	16
2.5 子程序设计实验.....	17
2.6 字符串处理程序实验.....	17
2.7 数制转换实验.....	18
2.8 DOS 功能调用实验.....	18
2.9 汇编程序设计综合实验.....	19
第 3 章 微机接口教学实验系统	20
3.1 微机接口实验系统简介.....	20
3.2 微机接口实验系统操作.....	24
第 4 章 微机接口应用实验	31
4.1 系统认识实验.....	31
4.2 存储器扩展实验.....	33
4.3 8259A 中断控制实验.....	34
4.4 8254 定时/计数器应用实验.....	43
4.5 8255 并行接口实验.....	48
4.6 8237 DMA 应用实验.....	57
4.7 8251 串行接口应用实验.....	61
4.8 A/D 及 D/A 转换实验.....	68
第 5 章 微机接口应用综合实验	73
5.1 8255 键盘及显示接口实验.....	73
5.2 步进电机实验.....	77
5.3 课程设计综合实验.....	79
参考文献.....	81

第1章 汇编语言程序设计概述

1.1 汇编语言开发环境

1.1.1 实验设备

硬件设备：PC 机一台。

1.1.2 汇编语言开发环境

1. DOS 环境

在计算机的任一逻辑盘上建一目录，把下面四个文件拷贝到该文件夹下：

- (1) 编辑软件 EDIT.exe;
- (2) 汇编软件 MASM.exe;
- (3) 连接程序 LINK.exe;
- (4) 调试程序 DEBUG.exe.

例如，在逻辑盘 D 下建一子目录 MASM，把上述文件拷贝到该目录下。该目录在后面的汇编语言上机过程中还会用到。

2. Windows 集成开发环境

具体操作见第 2 章的 2.1。

1.1.3 汇编语言程序上机操作一般过程

首先要用编辑软件建立汇编语言的源程序文件。用汇编语言编写的源程序不能由计算机直接执行。必须通过具有“翻译”功能的系统程序——汇编程序（Assembler）汇编产生相应的目标代码文件，再通过连接程序生成可执行文件，最后可以用 DEBUG 上机调试或运行程序。汇编语言程序上机过程，即编辑、汇编、连接、调试过程，如图 1-1 所示。用汇编语言编写的程序叫源程序，源程序经汇编程序翻译后所得的机器指令代码称为机器语言目标程序，简称目标程序。



图 1-1 汇编语言程序上机过程

1. 编写汇编语言源程序

将要解决的应用问题用汇编语言语句编写成程序，并建立扩展名为.ASM 的源程序文件。可任选一种编辑软件，来输入、建立或修改源程序。不要用格式控制符，要求编辑完成的文件扩展名必须是.ASM。

编辑源程序文件可任选：WORD、写字板、书写器、EDIT.COM、行编辑软件 EDLIN.COM 等软件中的一款。

2. 汇编

汇编是将汇编语言源程序翻译成机器语言程序的过程，即将*.ASM 汇编语言源文件转换为*.OBJ 文件。由系统软件汇编程序完成此翻译过程。

汇编程序有 ASM.exe、宏汇编 MASM.exe 及 TASM 等，一般使用宏汇编 MASM.exe，因为它比 ASM.exe 功能强。TASM 适用于 8086/8088 至 Pentium 系列 CPU 指令系统所编写的汇编语言程序，是比较先进的汇编工具。

3. 连接

将不可执行的二进制文件转换为可执行文件，即将*.OBJ 文件转换为*.EXE 文件。

完成连接的程序有 LINK.exe 或 TLINK.exe，它们将汇编产生的机器代码程序（即.OBJ 文件）连接成可执行程序文件（即.EXE 文件）。

4. 程序调试 (DEBUG)

进行汇编语言程序调试和文件格式转换，常用的软件调试工具有：动态调试程序 DEBUG.com、文件格式转换程序 EXE2BIN.exe、交叉引用程序 CREF.exe、动态调试程序 TD (Turbo Debugger)，可在 DOS 和 Windows 环境下运行，用户界面友好。

1.2 汇编语言实验过程详细步骤

1.2.1 DOS 环境实验过程详细步骤

1. 建立源程序

运行任一款编辑软件，输入源程序。保存源程序文件，且扩展名为.ASM。

2. 汇编

用编辑软件建立的源程序.ASM 文件，必须经过汇编才能产生.OBJ 文件。在 Windows 环境下，单击屏幕左下角的“开始”——“所有程序”——“附件”——“命令提示符”，键入“D:”并回车，再键入“cd masm”并回车，则命令提示符窗口如图 1-2 所示。

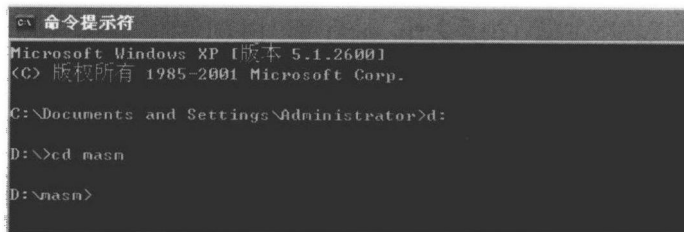


图 1-2 命令提示符窗口

在 D:\masm>提示符下键入“MASM”并回车，则在屏幕上显示：

```

Microsoft(R) Macro assemble Version 5.00
Copyright(C) Microsoft Corp 1981-1985,1987 ALL rights reserved
Source filename [.ASM]:
  
```

输入要汇编的汇编语言源程序文件名，并回车，则屏幕显示：

```

OBJ filename[<file>.obj]:
  
```

直接键入回车，则生成中括号中指定文件名的.obj 文件；也可以在上述输入自己的文件名后再键入回车，则生成自己指定文件名的.obj 文件。然后屏幕显示：

```

Source listing[nul.lst]:
  
```

直接键入回车, 则不生成.lst 文件; 若要生成.lst 文件, 则键入一个文件名后键入回车, 屏幕显示:

```
Cross-reference[nul.crf]:
```

直接键入回车, 则不生成.crf 文件; 若要生成.crf 文件, 则键入一个文件名后键入回车, 如源文件没有语法错误时屏幕显示:

```
49748 + 451356 Bytes symbol space free
0 Warning Errors
0 Severe Errors
D:\masm>
```

说明: 通过上述汇编后生成以下几个文件。

(1) .OBJ 文件。 .OBJ 文件是必须生成的一个目标代码文件, 当源程序中无语法错误时, 则在当前工作盘上自动存入一个.OBJ 文件, 供下一步连接用。

当源程序有语法错误时, 会出现类似如下形式的错误信息提示:

```
文件名.ASM(12): error A2009: Symbol not defined: AX1
文件名.ASM(15): error A2009: Symbol not defined: BRANCH1
50824 + 450280 Bytes symbol space free
0 Warning Errors (警告错误)
2 Severe Errors (严重错误)
D:\masm>
```

上述有两行错误信息, 错误信息格式是:

源程序文件(行号): 错误信息代码: 错误信息说明

若严重错误总数不为 0, 则.OBJ 文件没有生成, 需回到编辑状态下修改汇编源程序, 并重新汇编, 如此往复, 直到无错误为止。

(2) .LST 文件(列表文件)。 .LST 文件对调试程序有帮助, 它是将源程序中各语句及其对应的目标代码和符号表以清单方式列出。如果需要, 则在屏幕显示的第二个提问的冒号后输入文件名即可; 如果不需要, 则直接按回车键。

(3) .CRF 文件(交叉引用文件)。交叉引用.CRF 文件给出了源程序中定义的符号引用情况, 按字母顺序排列。 .CRF 文件不可显示, 须用 CREF.EXE 系统程序将.CRF 文件转换为.REF 文件后方可显示输出。

若按后一种格式输入, 系统会依次出现四个提示信息, 比第一种格式多了一项源程序文件名的输入。应答方式与第一种格式相同。

3. 连接目标程序

汇编后生成的.OBJ 文件, 其目标代码中的所有地址都是浮动的偏移地址, 机器不能直接运行。必须用连接程序(LINK.exe)对其进行连接装配定位, 产生.EXE 可执行文件, 方可运行。在系统提示符(D:\masm>)下键入“LINK<.obj 的文件名>”并回车, 屏幕上出现以下提示信息:

```
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.
Run File [<file>.exe]:
```

缺省情况下, 生成源程序文件名(即方括号中的默认值)的可执行文件。如果不修改可

执行文件的文件名，则直接键入回车；否则，键入可执行文件的文件名，并回车，则生成的可执行文件以该文件名为名字。屏幕显示：

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

连续键入回车后，返回系统提示符。连接后，可生成.EXE 和.MAP 两种文件。

(1) .EXE 文件（可执行文件）。若连接过程中无错误，则自动在当前工作盘生成.EXE 文件。若有错误，则会给出错误信息提示，这时必须重新修改源程序，再汇编、连接。

(2) .MAP 文件（连接映像文件）。MAP 是一个列表清单文件，一般不用生成。

(3) .LIB 库文件。连接程序询问是否使用某种库文件，不需使用库文件时直接按回车键。

(4) 若要连接多个模块目标程序，可用“+”把它们连接起来。在系统提示符下键入下列 LINK 命令并回车：

```
D:\masm>LINK files1+files2+files3
```

屏幕显示同 LINK 一个模块目标程序一样。

4. 调试和运行程序

经过汇编、连接后生成的.EXE 文件，可在 DOS 系统直接运行，只要键入可执行文件的文件名，并回车即可。

例如：D:\masm><文件名>

DOS 的 COMMAND.com 模块将该程序装配到内存，并设置和分配启动地址。也可在 DEBUG 调试程序下运行。如何用 DEBUG 程序来运行和调试程序将在 DEBUG 调试程序中详细讨论。

5. DEBUG 调试程序

DEBUG.exe 程序是专门为开发汇编语言程序而设计的一种调试工具，具有跟踪程序执行、观察中间运行结果、显示和修改寄存器或存储单元内容等多种功能。它能使程序设计人员或用户触及到机器内部，因此可以说它是 80X86CPU 的心灵窗口，也是我们学习汇编语言必须掌握的调试工具。

6. DEBUG 程序使用

在 DOS 系统提示符下键入命令并回车：

```
D:\masm>DEBUG [盘符:][路径]文件名.EXE
```

这时屏幕上出现 DEBUG 的提示符“-”，表示系统在 DEBUG 管理之下，此时可以用 DEBUG 进行程序调试。若所有选项省略，仅把 DEBUG 装入内存，可对当前内存中的内容进行调试，或者再用 N 和 L 命令，从指定盘上装入要调试的程序；若命令行中有文件名，则 DOS 系统把 DEBUG 程序调入内存后，再由 DEBUG 将指定可执行文件装入内存。

7. DEBUG 常用命令简介

(1) 汇编命令 A。

格式：A[起始地址]

功能：将输入源程序的指令汇编成目标代码并从指定地址单元开始存放。若缺省起始地址，则从当前 CS: 100 地址开始存放。A 命令按行汇编，主要是用于小段程序的汇编或对目标程序的修改。

(2) 反汇编命令 U。

格式 1: U[起始地址]

格式 2: U[起始地址][结束地址][字节数]

功能: 格式 1 从指定起始地址处开始将 32 个字节的目標代码转换成汇编指令形式, 若缺省起始地址, 则从当前地址 CS:IP 开始。格式 2 将指定范围的内存单元中的目标代码转换成汇编指令。

(3) 显示、修改寄存器命令 R。

格式: R[寄存器名]

功能: 若给出寄存器名, 则显示该寄存器的内容并可进行修改。若缺省寄存器名, 则按以下格式显示所有寄存器的内容及当前值(不能修改)。例如:

```
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0B03 ES=0B03 SS=0B03 CS=0B03 IP=0100 NV UP EI PL NZ NA PO NC
-R AX          ;输入命令
AX 0000       ;显示 AX 的内容
:            ;供修改,直接键入 AX 新值并回车;若不修改,则直接键入回车。
```

若对标志寄存器进行修改, 在 DEBUG 提示符下键入 RF, 并回车。例如:

```
-R F
NV UP DI PL NZ NA PO NC -
```

屏幕显示如上信息, 分别表示 OF、DF、IF、SF、ZF、AF、PF、CF 的状态。若不修改则按回车键; 要修改, 则个别输入一个或多个此标志的相反值, 再按回车键。R 命令只能显示、修改 16 位寄存器。

(4) 显示存储单元命令 D。

格式 1: D [起始地址]

格式 2: D [起始地址][结束地址][字节数]

功能: 格式 1 从起始地址开始按十六进制显示 80H 个单元的内容, 每行 16 个单元, 共 8 行, 每行右边显示 16 个单元的 ASCII 码, 不可显示的 ASCII 码则显示“·”。格式 2 显示指定范围内存储单元的内容, 显示方式与格式 1 一样。如果不写起始地址或地址范围, 则从当前地址开始按格式 1 显示。例如, 键入 D200 并回车, 屏幕显示如下:

```
-D 200
0B03:0200 43 46 49 FE C4 8A 07 3C-20 74 01 AA 43 E2 BC C3 CFI....< t..C...
0B03:0210 F6 46 04 02 75 43 8B D5-83 C2 05 57 B8 00 6C BB .F..uC.....W..l.
0B03:0220 40 00 33 C9 8B F2 BA 01-01 CD 21 5F 73 15 E8 8C @.3.....!_s...
0B03:0230 DB 3D 02 00 74 23 3D 03-00 74 1E 3D 05 00 74 19 .=.t#=.t.=.t.
0B03:0240 E9 AB D8 8B D8 B8 00 44-CD 21 B4 3E CD 21 F6 C2 .....D.!>.!..
0B03:0250 80 75 53 F6 46 04 04 74-4D 8B 56 05 80 FA 00 74 .uS.F..tM.V....t
0B03:0260 05 80 FE 3A 74 02 B2 40-80 CA 20 80 EA 60 E8 3C ....t..@.. ..`.<
0B03:0270 E4 73 06 E8 47 DB E9 75-D8 8B D5 83 C2 05 8A 7E .s..G..u.....~
```

表示从 DS:0200H 开始显示 128 个单元内容, 其中 DS 值不同的计算机环境其值不同。

-D 100 120; 表示显示 DS:0100~DS:0120 单元的内容

说明: 在 DEBUG 中, 地址表示方式有如下形式:

段寄存器名：相对地址，如：DS:100

段基值：偏移地址（相对地址），如：23A0:1500

(5) 修改存储单元命令 E。

格式 1：E[起始地址][内容表]

格式 2：E[地址]

功能：格式 1 按内容表的内容修改从起始地址开始的多个存储单元内容，即用内容表指定的内容来代替存储单元当前内容。例如：

```
-E DS: 0100 'VAR' 12 34
```

表示从 DS:0100 为起始单元的连续五个字节单元内容依次被修改为'V'、'A'、'R'、12H、34H。

格式 2 是逐个修改指定地址单元的当前内容。例如：

```
-E DS: 0010
156F: 0010 41.5F
```

其中，156F:0010 单元（注意：不同计算机环境 DS 值不一定是 156F）原来的值是 41H，5FH 为输入的修改值。若只修改一个单元内容，这时按回车键即可；若还想继续修改下一个单元内容，则应按空格键，就显示下一个单元内容，需修改就键入新内容，不修改再按空格跳过，如此重复直到修改完毕，按回车键返回 DEBUG “-” 提示符。如果在修改过程中，将空格键换成按“-”键，则表示可以修改前一个单元内容。

(6) 连续运行命令 G。

格式：G [=起始地址] [第一断点地址 [第二断点地址……]]

功能：CPU 从指定起始地址开始执行，依次在第一、第二等断点处中断。若缺省起始地址，则从当前 CS:IP 指示地址开始执行一条指令。最多可设置 10 个断点。

(7) 跟踪命令 T。

格式：T [起始地址] [正整数]

功能：从指定地址开始执行“正整数”条指令，若缺省“正整数”，表示执行一条指令，若两项都缺省，表示从当前 CS:IP 指示地址开始执行一条指令。

(8) 指定文件命令 N。

格式：N <文件名或扩展名>

功能：指定即将调入内存或从内存写入磁盘的文件名。该命令应该用在 L 命令和 W 命令之前。

(9) 装入命令 L。

格式 1：L [起始地址] [盘符号] [扇区号] [扇区数]

格式 2：L [起始地址]

功能：格式 1 根据盘符号，将指定扇区的内容装入到指定起始地址的存储区中。

格式 2 将 N 命令指定的文件装入到指定起始地址的存储区中，若省略起始地址，则装入到 CS:100 处或按原来文件定位约定装入到相应位置。

(10) 写磁盘命令 W。

格式 1：W <起始地址> [驱动器号] <起始扇区><扇区数>

格式 2: W [起始地址]

功能: 格式 1 把指定地址开始的内容数据写到磁盘上指定扇区中。

格式 2 将起始地址的 $BX \times 1000H + CX$ 个字节内容存放到由 N 命令指定的文件中。在格式 2 的 W 命令之前, 除用 N 命令指定存盘文件名外, 还必须将要写的字节数用 R 命令送入 BX 和 CX 中。

(11) 退出命令 Q。

格式: Q

功能: 退出 DEBUG, 返回到操作系统。

1.2.2 其他 DEBUG 调试命令

DEBUG 调试常用命令见表 1-1, 请参考有关书籍。

表 1-1 DEBUG 调试常用命令

名 称	含 义	命 令 格 式
A(Assemble)	逐行汇编	A [address]
C(Compare)	比较两内存块	C range address
D(Dump)	显示内存单元 (区域) 内容	D [address] D [range]
E(Enter)	修改内存单元 (区域) 内容	E address list
F(Fill)	填充内存单元 (区域)	F range list
G(GO)	连续执行程序	G [=address] [address]
H (Hexarithmic)	两参数进行十六进制运算	H value value
I(Input)	从指定端口地址读取并显示一个字节值	I port address
L(Load)	装入某个文件或特定磁盘扇区的内容到内存	L [address]
M(Move)	将内存块内容复制到另一个内存块	M range range
N(Name)	指定要调试的可执行文件参数	N [d:[path] filename [.exe]]
O(Output)	将字节值发送到输出端口	O port address byte
Q(Quit)	退出 DEBUG	Q
R(Register)	显示或修改一个或多个寄存器内容	R [register name]
S(Search)	在某个地址范围搜索一个或多个字节值	S range list
T(Trace)	单步 / 多步跟踪	T or T [address] [Value]
U(Unassemble)	反汇编并显示相应原语句	U [address] or U [range]
W(Write)	文件或数据写入特定扇区	W [address [drive sector sector]]
?	显示帮助信息	?

第2章 80X86 汇编语言实验内容

1. 实验设备

硬件：PC 计算机一台；软件：汇编软件。

2. 实验目的

- (1) 熟悉汇编语言程序设计方法。
- (2) 熟悉汇编程序调试过程。
- (3) 掌握使用数据传送、运算类、转移等指令编程及调试方法。
- (4) 掌握顺序结构、分支结构、循环结构和子程序调用的编程方法。

3. 实验及实验报告要求

- (1) 编程、调试、运行、检查结果，并记录。

要求：对程序反复输入几组数据，并运行程序，记录结果。

- (2) 实验报告要求。实验报告中应包含实验内容、源程序、实验步骤、运行结果、结果分析是否正确及存在的问题。

2.1 系统认识实验

2.1.1 实验目的

掌握微型计算机汇编软件调试汇编源程序的基本操作。熟悉传送类指令、简单运算类指令来编写汇编语言源程序，掌握调试过程及实验结果分析。

2.1.2 实验内容

(1) 从 3500H 内存单元开始建立 0~15 共 16 个数据（建立的数据为 00H, 01H~09H, 0AH~0FH），并运行之，检查记录结果。

1) 分析：①3500H 为偏移地址，对应的段基址采用默认的 DS 当前值；②程序中，如果不涉及堆栈，则堆栈段可省略。

2) 实验程序：

```
DATA SEGMENT
ORG 3500H                ; 设定起始偏移地址, 如果没有该语句, 默认的偏移地址为 0000H
MEM1 DB 16 DUP(?)
DATA ENDS
CODE SEGMENT            ; 定义代码段
    ASSUME CS:CODE, DS:DATA
START: MOV AX, DATA
      MOV DS, AX
      MOV DI, 3500H      ; 程序开始设数据区首址
      MOV CX, 0010H     ; 字节数送入 CX 中
      MOV AH, 00H
      SAHF              ; 清标志寄存器指令
```

```

MOV AL, 00H
A1:  MOV [DI],AL      ;写入一字节
      INC DI         ;修改地址指针
      INC AL         ;修改数据
      LOOP A1        ;未填满转移
CODE ENDS           ;代码段结束
END START          ;程序结束

```

(2) 如果(1)建立的数据为BCD码表示的十进制数据(即建立的数据为00H, 01H~09H, 10H~15H), 修改上述程序并调试运行, 检查记录结果。

提示: BCD运算, 是二进制编码的十进制运算, 使用指令ADD和DAA或INC和DAA共同完成, 仅能使用AL进行运算。例如, 15+28进行运算, 编程如下:

```

MOV AL, 15H      ;15H是15的BCD码表示
ADD AL, 28H     ;28H是28的BCD码表示, 这里完成15H+28H=3DH运算
DAA             ;将AL中的结果3DH调整为BCD码, 结果为43H

```

(3) 编程完成 $1+2+3+4+5+6+7+8+9$ 的求和运算。调试运行程序, 并记录运算结果。

(4) 将内存3500H单元开始存放的0~9共10个字节数传递到3600H单元开始的数据区中(可分同一段内传送和段间传送)。

提示: 该程序应该编写数据段, 从3500H开始存放0~9共10个数; 从3600H定义10个字节单元, 准备接收数据。数据段定义如下:

```

DATA SEGMENT
      ORG 3500H
MEM1 DB 0,1,2,3,4,5,6,7,8,9
      ORG 3600H
MEM2 DB 10 DUP(?)
DATA ENDS

```

思考题:

- 1) 如果上述数据段定义中没有ORG语句, 则从什么内存地址开始存放数据?
- 2) 如果采用字数据传送, 请修改程序, 并调试运行程序, 记录运行结果。
- 3) 如果采用串传送指令(MOVSB, MOVSW), 请修改程序, 并调试运行程序, 记录运行结果。

提示: 如果使用MOVSB、MOVSW指令, 要将原始数据串的段基地址放入DS寄存器, 接收数据的目的串段基地址放入ES寄存器, 可以使DS与ES设置成同一个段基址(即逻辑段重叠); 也可以设置成不同逻辑段, 即数据段和附加数据段。

2.1.3 实验步骤

以实验内容(1)为例的实验步骤如下。实验内容(2)~(4)请自行依照实验步骤完成调试, 并记录实验结果。

1. 输入源程序

打开WINDOWS环境下的汇编开发环境, 输入源程序。在桌面上双击快捷方式图标“Masm for Windows 集成实验环境”。出现图2-1所示界面, 其中右侧是源程序模板, 按提示输入编写好的源程序。

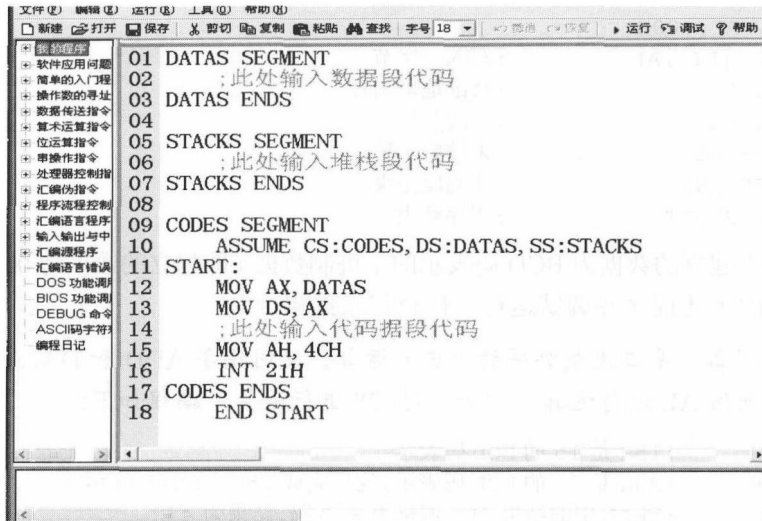


图 2-1 汇编源程序模板

图 2-1 说明:

在右侧窗口给出了段结构模板，根据模板输入源程序。模板中的段名可以自己修改定义，没有涉及的段可省略不写。

输入汇编语言源程序后，保存程序（规定扩展名为.asm），使用 ALT+X 返回集成软件环境。输入源程序后，如图 2-2 所示。



图 2-2 输入源程序后界面

2. 汇编

编译/汇编成目标文件，单击运行菜单下“编译成目标文件（OBJ）”，等待系统汇编，汇

编成功后，汇编信息显示于屏幕下面的窗口中，显示编译成功（见图 2-3），则可进行第 3 步。如果显示错误提示信息，则返回第 1 步修改源程序后再汇编，直到无错误为止。

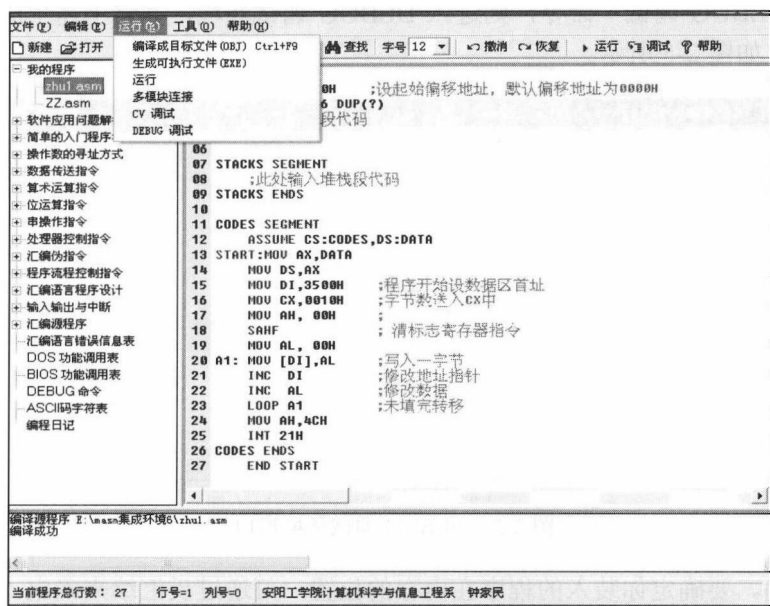


图 2-3 编译/汇编命令

3. 连接

连接生成可执行文件。单击运行菜单下“生成可执行文件 (EXE)”，等待系统对汇编生成的.obj 文件进行连接，连接成功后，生成相应的可执行文件 (.EXE)，信息显示于屏幕下方的窗口上，如图 2-4 所示。如果连接成功，则显示生成*.EXE 文件成功，进行下一步 DEBUG 调试。否则，返回第 1 步，修改源程序，再汇编、连接，直到无错误信息为止。



图 2-4 连接命令及连接成功后的信息显示

4. 在 DEBUG 调试环境下进行程序的调试过程

通过 DEBUG 调试可以检查程序每一条指令及程序执行结果，以便查错和纠错。单击运行菜单下的“DEBUG 调试”命令，则进入 DEBUG 调试环境，并将上述生成的可执行文件自动装入内存，如图 2-5 所示。

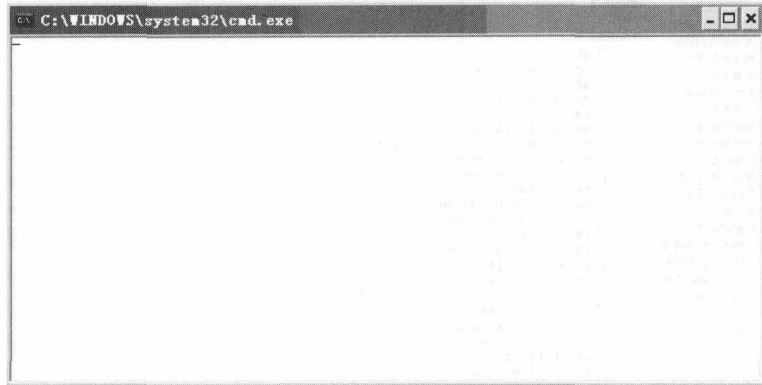


图 2-5 DEBUG 调试环境窗口

(1) 反汇编。要确定你装入的程序在内存的位置，可通过反汇编来确定。反汇编是显示装入内存的汇编语言程序及其机器指令在内存的情况，在 DEBUG 命令提示符“-”下键入 U 并回车，操作过程如下：

-U ✓ 回车；从当前 IP 指令指针开始反汇编。即刚装入内存的指令开始反汇编；或者
-U 0000 ✓ 回车；从起始偏移地址 0000H 开始反汇编，即刚装入内存的指令开始反汇编。
上述两种格式的命令功能一样，反汇编结果如下：

```
-u
0EB2:0000 B8610B      MOV     AX,0B61
0EB2:0003 8ED8          MOV     DS,AX
0EB2:0005 BF0035      MOV     DI,3500
0EB2:0008 B91000      MOV     CX,0010
0EB2:000B B400          MOV     AH,00
0EB2:000D 9E           SAHF
0EB2:000E B000          MOV     AL,00
0EB2:0010 8805          MOV     [DI],AL
0EB2:0012 47           INC     DI
0EB2:0013 FEC0          INC     AL
0EB2:0015 E2F9          LOOP   0010
0EB2:0017 B44C          MOV     AH,4C
0EB2:0019 CD21          INT     21
0EB2:001B 67           DB     67
0EB2:001C 7261          JB     007F
0EB2:001E 6D           DB     6D
0EB2:001F 2E           CS:
0EB2:0020 0D0A0D      OR     AX,0D0A
-
```

1) 键入 U 命令后，屏幕显示刚装入的汇编程序。