

961

TP303-33

J32

华东高校计算机基础教育研究会推荐教材

计算机硬件技术教程实验指导

贾智平 沈 磊 主编



A0937095

中国水利水电出版社
上海交通大学出版社
东南大学出版社

内 容 简 介

本书是中国水利水电出版社已出版的《计算机硬件技术教程——微机原理与接口技术》的配套教材,并增加了计算机原理实验部分。全书共分三章,第一章介绍了与硬件技术相辅相成的汇编语言实验过程,给出了字符及字符串操作、功能调用、数据排序、文件读/写操作等实验内容;第二章介绍了微机硬件接口技术实验,针对常用的微机接口芯片给出了实用性实验内容;第三章编排了多个计算机部件实验,由简单到复杂地引入了计算机整机实验。

本书内容简洁,每一实验除给出了实验目的和内容外,还提供了实验参考提示,以充分发挥实验者的想象力和创造力,提高其自己动手的实践能力。因此本书既适于作为计算机专业本科生的硬件实验指导书,又可供从事计算机硬件开发的科研及工程技术人员参考。

图书在版编目(CIP)数据

计算机硬件技术教程实验指导 / 贾智平, 沈磊主编. —北京: 中国水利水电出版社, 2000.8

华东高校计算机基础教育研究会推荐教材

ISBN 7-5084-0421-1

I. 计… II. ①贾… ②沈… III. 电子计算机-硬件-实验-高等学校-教学参考资料 IV. TP303

中国版本图书馆 CIP 数据核字 (2000) 第 65675 号

书 名	计算机硬件技术教程实验指导
主 编	贾智平 沈 磊
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www. waterpub. com. cn E-mail: sale@waterpub. com. cn 电话: (010) 63202266 (总机)、68331835 (发行部)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787×1092 毫米 16 开本 6.5 印张 147 千字
版 次	2000 年 8 月第一版 2000 年10月北京第二次印刷
印 数	5001—8000 册
定 价	10.00 元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换
版权所有·侵权必究

前 言

学习掌握计算机硬件技术知识一定要突出硬件“有形”这个特点，以实践加深印象，提高计算机设计能力和创新能力。由于计算机学科及硬件技术的飞速发展，导致与计算机相关的新材料、新器件及新应用层出不穷，特别是大规模集成电路的产生使计算机得到了日益广泛的应用，掌握计算机硬件的工作原理和应用技术，已成为当代科研和工程技术人员必不可少的基本技能。

本书是中国水利水电出版社已出版的《计算机硬件技术教程——微机原理与接口技术》的配套教材，并增加了计算机原理实验部分。因此本书既可作为计算机专业本科生的硬件实验指导书，又可供从事计算机硬件开发的科研及工程技术人员参考。

全书共分三章，第1章介绍了与硬件技术相辅相成的汇编语言实验过程，给出了七个实验内容：熟悉DEBUG有关命令、字符串传送、字符匹配程序、字符和数据显示、将键入的字符进行转换、排序实验、文件读/写操作实验；第2章首先介绍了TPC-1型微机实验台基本结构及组成，针对常用的微机接口芯片给出了实用性实验内容，其中包括：计数器/定时器8253A、可编程并行接口8255A、可编程串行接口8251A、中断控制器8259A、DMA控制器8237A、D/A转换器DAC0832、A/D转换器ADC0809。然后，为了使读者全面掌握微机接口技术，提供了一个综合实验内容。第3章介绍了FD-CES计算机实验仪的性能特点及它提供的功能模块使用方法，编排了多个计算机部件实验，包括：时序系统实验、总线传送控制实验、存储器实验、补码加法器实验、原码一位乘法实验、原码一位除法实验、组合逻辑控制器实验，并由简单到复杂地引入了计算机整机实验。

本书内容简洁，每一实验除了给出实验目的和内容外，还提供了实验参考提示，以充分发挥实验者的想象力和创造力，提高其自己动手的实践能力。

本教材由崔巍同志编写第1章，贾智平、张爱武同志编写第2章，沈磊、陈志勇同志编写第3章。

在该书的编写过程中，山东工业大学计算机科学技术系的石冰同志提出了许多宝贵意见，并对全书进行了审校，在此表示衷心感谢。本书同时参考了清华大学1993年版的资料——TPC-1型十六位微机实验培训系统，在此也表示由衷的感谢。

由于编者水平所限，加之时间紧迫，书中难免存在错误和不足之处，敬请读者在使用本书过程中批评指正，以便改进。

编者

2000年5月

1

汇编语言实验

1.1 汇编语言上机过程

用汇编语言编写的源程序，使之运行必须经过以下几个步骤：

- ◆用编辑程序建立一个扩展名为.ASM 的汇编语言源程序文件。
- ◆调用 MASM（或 ASM）汇编程序，对源程序汇编，生成目标文件，扩展名为.OBJ。
- ◆连接（LINK）目标文件，生成可执行文件，扩展名为.EXE。
- ◆运行可执行文件。
- ◆如程序运行有错，可调用 DEBUG 程序对.EXE 文件进行调试。

因此，在系统盘上应具有下列文件：

- ★通用编辑程序 EDIT.COM EDIT.HLP
 QBASIC.EXE QBASIC.HLP 或其他编辑文件。
- ★宏汇编程序 MASM.EXE （或 ASM.EXE）
- ★连接程序 LINK.EXE
- ★调试程序 DEBUG.EXE

1.1.1 编辑汇编语言源程序

可用各种编辑软件来建立汇编语言源程序，例如：DOS 提供的 EDIT 命令、Windows 的记事本和 PE3 等。源程序文件名由用户指定，扩展名为.ASM。关于编辑命令的使用，请参考相应的使用说明。如建立 EXAM.ASM 文件，使用命令 C>EDIT EXAM.ASM<CR>（<CR>表示回车键）。

1.1.2 编译汇编语言

用 MASM（或 ASM）程序汇编源程序，生成浮动目标程序（即扩展名为.OBJ 的文件），这里产生的是浮动的目标模块，是不可执行程序。

在 DOS 状态下，键入 MASM（或 ASM）屏幕显示与操作如下：

A>MASM<CR>

屏幕显示以下各行（一次一行），MASM 提示用户所应键入的信息：

SOURCE FILENAME [.ASM]: EXAM<CR>

OBJECT FILENAME [EXAM.OBJ]<CR>

SOURCE LISTING [NUL.LST]: <CR>

CROSS REFERENCE [NUL.CRF]: <CR>

如果用户在进入 MASM 时就给出源文件名, 则 MASM 直接显示提示询问目标文件名。产生目标文件.OBJ 是汇编的主要目的, 通常用户可以直接回车, 表示采用默认文件名。如果用户想换文件名, 可以在此时键入用户确认的目标文件名。

.LST 文件为列表文件, 若要建立列表文件, 则键入文件名。MASM 将为用户建立扩展名为.LST 列表文件, 它提供全部汇编后的信息。若用户不需要这些信息, 可直接回车。

.CRF 文件为交叉引用文件, 若要建立交叉引用文件, 则键入文件名。MASM 将为用户建立扩展名为.CRF 的交叉引用文件, 这里所得到的交叉引用文件, 必须调用 CREF 程序处理, 以建立一个按字母顺序排列的能被显示或打印的源程序中的符号(包括变量)表。利用它用户可以很快查到符号的相对地址及类型, 这对调试较大的程序是很有用的, 所以用户的盘中还应拷贝 CREF.EXE 文件。若用户不需要这些信息, 可直接回车。

到此为止, 汇编过程已完成。当 MASM 对源程序进行汇编时, 若发现源程序中有语法错误, 则列出有错误的语句, 错误的代码和错误的类型, 最后列出错误的总数, 此时就可分析错误, 再调用编辑程序修改错误, 重新汇编直到汇编无错误。

1.1.3 连接汇编语言

LINK.EXE 文件用来将.OBJ 文件连接成可执行文件, 也可以将若干个独立汇编的模块连接成一个可执行的文件(扩展名为.EXE)。连接过程如下: 在 DOS 下, 键入 C>LINK<CR> 后先显示版本号, 然后依次提问以下四个问题:

OBJECT MODULES [.OBJ]: EXAM<CR>

RUN FILE [EXAM.EXE]: <CR>

LIST FILE [NUL.MAP]: <CR>

LIBRARIES[.LIB]: <CR>

首先询问要连接的目标文件名, 如果有多个要连接的目标文件, 则应一次输入, 各文件名间用“+”号相间隔。目标文件.OBJ 文件名也可在键入 LINK 命令时一起输入。第二个提示询问要产生的可执行文件(.EXE)的文件名, 一般直接回车以默认括号内规定的文件名。第三个提示询问是否建立内存分配图文件(.MAP), 直接回车表示不要建立。最后询问是否用库文件(.LIB), 若汇编文件不用库文件, 直接回车。然后连接程序开始进行连接, 若连接过程有错, 则显示错误信息, 如:

WARNING: NO STACK SEGMENT

THERE WAS 1 ERROR DETECTED

有错误就要修改源文件, 重新汇编、连接直至无错。若用户程序直接使用系统堆栈, 则可不理会“NO STACK SEGMENT”的警告提示。

1.1.4 汇编语言运行

汇编语言源程序经过汇编、连接以后生成的.EXE 文件, 可在 DOS 下直接键入文件名运行。例如: C>EXAM<CR> 则 EXAM.EXE 文件会装入内存并从程序中起始的地址运行。程序如正确无误, 执行完后控制将能正常返回 DOS 操作系统。若运行结果在存储单元里, 或发现程序运行错误, 或想跟踪程序的执行, 那么就需要用 DEBUG 程序。

1.1.5 汇编语言调试

DEBUG 是专为汇编语言设计的一个调试工具，它通过单步、设置断点等方式为汇编语言程序的调试提供了非常有效的调试手段。

在 DOS 的提示下，可键入命令：

C>DEBUG<文件名.扩展名> 或 C>DEBUG

第一种方式是 DOS 把 DEBUG.COM 调入内存后把被调试的程序也调入内存。对于扩展名为.EXE 的文件程序装入内存的地址从偏移 0 开始，装入后显示 DEBUG 提示符“-”。第二种方式是，调用 DEBUG 后，立即显示提示符“-”，在提示符“-”下可执行 DEBUG 的各种命令。

1.2 常用的 DEBUG 命令介绍

(1) 显示内存单元内容的命令 D

格式 1: D[地址]

显示从指定地址开始的 80H 个单元的内容。若命令中没有指定起始地址，则从上一个 D 命令所显示的最后一个单元开始，若以前没有使用过 D 命令，则从 DEBUG 初始化的 DS 段，加上偏移量 0100H 作为起始地址。

格式 2: D[始地址][末地址]

此命令显示指定范围的内容。如：-D DS: 500 530<CR>显示 500H--530H 单元的内容。

(2) 修改存储单元内容的命令 E

格式 1: E[地址][单元内容表]

用命令给定的内容表去代替指定范围的内存单元内容。如：-E DS: 100 AA BB CC DD <CR>则将 0AAH、0BBH、0CCH 及 0DDH 依次送入 DS: 100H 开始的 4 个单元。

格式 2: E[地址]

一个单元一个单元的修改内存的内容。如：-E CS: 108<CR> 则只显示出 CS 段内 108H 单元的内容，等待修改。修改后可按空格键显示下一个单元的内容，利用空格键可修改连续的内存单元，用回车键结束该命令。

(3) 填充存储单元内容的命令 F

格式: F[地址][单元内容表]

用命令给定的内容表去填充指定范围的内存单元内容。如：-F 1200: 100 4 11223344 <CR>则将 11H、22H、33H 及 44H 依次送入 1200: 100H 开始的 4 个单元。

(4) 检查和修改寄存器内容的命令 R

格式 1: R

显示 CPU 内部所有寄存器的内容和全部标志位状态。

格式 2: R[寄存器名]

检查和修改指定寄存器的内容。如键入 -R IP<CR>响应如下：

IP 0356: 显示 IP 原来的内容

0100: 则将 0100H 送 IP。如不修改可直接按回车键。

格式 3: RF

可显示和修改标志位状态。

(5) 运行命令 G

格式 1: G[=始地址][断点] [断点]

其中断点最多允许设置 10 个。功能为: 从指定的地址开始执行程序, 当到达指定断点时, 停止执行, 并显示寄存器、标志和下一条要执行的指令。如: -G=0100 010B 则从 0100H 开始执行到 010BH 停止。

格式 2: G

从当前 CS: IP 开始一直执行下去, 直到 INT 3 指令为止。

格式 3: G[=地址]

从指定地址开始执行下去。

(6) 跟踪命令 T

格式 1: T[=地址]

单条跟踪命令, 即执行时逐条跟踪, 从指定地址起执行一条指令后停下来, 显示所有寄存器的内容及标志位的值。如未指定地址则从当前的 CS: IP 开始执行。

格式 2: T[=地址] [N]

多条跟踪, 从指定地址开始执行 N 条指令后停下来。

另外, 与 T 命令类似的还有一条 P 命令, 该命令可执行一个循环、一个重复的字符串、一个软件中断或一个程序调用, 即遇到 CALL 或 INT 等指令时, 不会进入相应的子程序内逐条执行。

(7) 汇编命令 A

该命令提供了在 DEBUG 方式下输入和汇编源程序的手段。

格式: A[地址]

从指定地址开始, 可以输入汇编语言的语句, 并被汇编成机器码从指定的地址单元开始存放。若命令中没有指定地址, 则从 CS: 0100H 单元开始存放。必须注意 DEBUG 把键入的数字均看成十六进制数。

(8) 反汇编命令 U

若在内存某区域中, 已经有了某个程序的目标代码, 为了了解此程序的内容, 可以用 U 命令进行反汇编。

格式 1: U[起始地址]

从指定地址开始, 反汇编 32 个字节。若没有指定地址, 则以上一个 U 命令的最后一条指令的下一个单元为起始地址。若未使用过 U 命令, 则从 CS: 100H 开始。

格式 2: U[起始地址] [结束地址]

对指定范围内的存储单元进行反汇编。

(9) 输入命令 I

此命令从指定的端口输入一个字节并显示出来。

格式: I[口地址]

(10) 输出命令 O

此命令能向指定端口输出一个字节。

格式: O [口地址][字节]

例如: -O 2F0 20<CR> : 把 20H 输出到 2F0H 端口。

(11) 文件命名命令 N

把给出的文件名(可包含路径名)格式化在文件控制块中,以便在其后用 L 或 W 命令将文件装入和存盘。

格式: N[文件名]

(12) 文件装入命令 L

此命令把由 N 命令定义的文件的内容或指定的扇区范围的内容装入到指定地址开始的存储区中。

格式 1: L[地址]

装入格式化了的文件控制块所指定的文件,如未指定地址则装入从 CS: 100H 开始的存储区中。

格式 2: L[地址][驱动器号][起始逻辑扇区][装入扇区个数]

如: -L 200 0 0 B<CR> 则将 A 盘上 0 扇区开始的 11 个扇区的内容装入 DS: 0200H 开始的存储区中。

(13) 文件写命令 W

此命令把数据写入指定的文件中或写入磁盘的指定扇区。

格式 1: W[地址]

把指定的存储区中的数据写入由 N 命令所指定的文件中,如未指定地址则数据从 CS: 100H 开始。要写入文件的字节数应先放入 BX 和 CX 中。

格式 2: W[地址][驱动器号][起始逻辑扇区][写扇区个数]

(14) 退出 DEBUG 命令 Q

格式: Q

退出 DEBUG, 返回 DOS。

DEBUG 命令中, 还有其他的命令, 读者可参阅有关资料。

1.3 汇编程序设计实验

1.3.1 实验一 熟悉 DEBUG 有关命令

一、实验目的

DEBUG 有关命令的使用方法。

二、实验内容

利用 DEBUG 的汇编命令 A 输入以下程序段:

C>DEBUG

-A

MOV SI, 1200

MOV DI, 1400

MOV AX, [SI]

```

        ADD    AX, [DI]
        MOV    [SI], AX
        MOV    AX, [SI+2]
        ADC    AX, [DI+2]
        MOV    [SI], AX
        INT    3
-E      DS: 1200                      ; 修改内存单元内容
        1200 11 22 33 44
-E      DS: 1400
        1200 11 11 77 11
-G                               ; 运行程序
-D      DS: 1200                      ; 检查结果是否正确

```

1.3.2 实验二 字符串传送实验

一、实验目的

1. 学习 REP MOVSB 指令的使用方法。
2. 熟悉在 PC 机上建立、汇编、链接、调试和运行 8086 汇编语言程序的过程。

二、实验内容

1. 把 40 个字母的字符串从 SBUFFER 开始的存储区传送到 DBUFFER 开始的存储区。

源程序如下：

```

DATA SEGMENT                      ; 定义数据段
SBUFFER DB 40 DUP ('A')
DATA ENDS
EXTRA SEGMENT                     ; 定义附加段
DBUFFER DB 40 DUP (?)
EXTRA ENDS
CODE SEGMENT                      ; 定义代码段
    ASSUME CS: CODE, DS: DATA, ES: EXTRA
START:  MOV AX, DATA              ; 给段寄存器赋值
        MOV DS, AX
        MOV AX, EXTRA
        MOV ES, AX
        LEA SI, SBUFFER
        LEA DI, DBUFFER
        CLD                       ; 设置方向标志
        MOV CX, 40
        REP MOVSB
        MOV AX, 4C00H             ; 返回 DOS

```

```
        INT 21H
CODE    ENDS
        ENDS START
```

2. 实验步骤。

(1) 调用编辑程序 EDIT 建立以 EXMOVS.ASM 为文件名的源文件。

```
C>EDIT EXMOVS.ASM
```

(2) 调用 MASM 程序产生目标程序。

```
C>MASM EXMOVS
```

若被汇编的程序没有语法错误，则显示如下信息：

```
0 Warning Errors
```

```
0 Servers Errors
```

若出错，从第一步开始修改程序。

(3) 调用 LINK 程序产生可执行文件。

```
C>LINK EXMOVS;
```

若显示 Warning: NO STACK segment

```
There was 1 error detected
```

则可以继续做下一步

(4) 调用 DEBUG 运行程序。

```
C>DEBUG EXMOVS.EXE
```

```
-U
```

```
18F4: 0000 B8F618 MOV AX, 18F6
```

```
18F4: 0003 8ED8MOV DS, AX
```

```
18F4: 0005 B8F918 MOV AX, 18F9
```

```
18F4: 0008 8EC0MOV ES, AX
```

```
18F4: 000A 8D360000LEA SI, [0000]
```

```
18F4: 000E 26 ES:
```

```
18F4: 000F 8D3E0000LEA DI, [0000]
```

```
18F4: 0012 FC CLD
```

```
18F4: 0013 B92800 MOV CX, 0028
```

```
18F4: 0016 F3 REPZ
```

```
18F4: 0017 A4 MOVSB
```

```
18F4: 0018 MOV AX, 4C00
```

```
18F4: INT 21
```

G17 ; 运行程序

```
AX=18F9 BX=0000 CX=0000 DX=0000 SP=0000
```

```
SI=0028 DI=0028 DS=18F6 ES=18F9 SS=18F4
```

```
CS=18F4 IP=0017 NV UP DI PL ZR NA PE NC
```

```
18F4: 0017 MOV AX, 4C00
```

```
-D18F6: 0
```

```

18F6: 0000 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAA
18F6: 0010 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAA
18F6: 0020 41 41 41 41 41 41 41 41-00 00 00 00 00 00 00 00 AAAAAAAAAAAAAAAAAA
18F6: 0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 AAAAAAAAA.....
.
.
.
-18F9: 0
18F9: 0000 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAA
18F9: 0010 41 41 41 41 41 41 41 41-41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAAAA
18F9: 0020 41 41 41 41 41 41 41 41-00 00 00 00 00 00 00 00 AAAAAAAAA.....
.
.
.
-Q
C>

```

1.3.3 实验三 字符匹配程序实验

一、实验目的

掌握提示信息的使用方法及键盘输入信息的用法。

二、实验内容

编写程序，实现两个字符串的比较。如相同，则显示“MATCH”，否则，显示“NO MATCH”。

三、参考流程

字符匹配程序实验流程如图 1-1 所示。

1.3.4 实验四 字符和数据显示实验

一、实验目的

掌握字符和数据的显示方法。

二、实验内容

先显示信息“INPUT CHARS, THE END FLAG IS \$”，再接收字符。如为 0~9 中任意数字，则计数器加 1，并显示数据；如为非数字，则直接显示，但不计数。程序以“\$”字符结束，并显示所按数字键计数结果。

三、参考流程

字符和数据显示实验流程如图 1-2 所示。

四、注意事项

以上流程图假定计数值 ≤ 9 ，则显示计数结果时，仅将计数寄存器 BL 内容与 30H 相或后送 DL，调用 INT 21H 的 2 号功能即可。

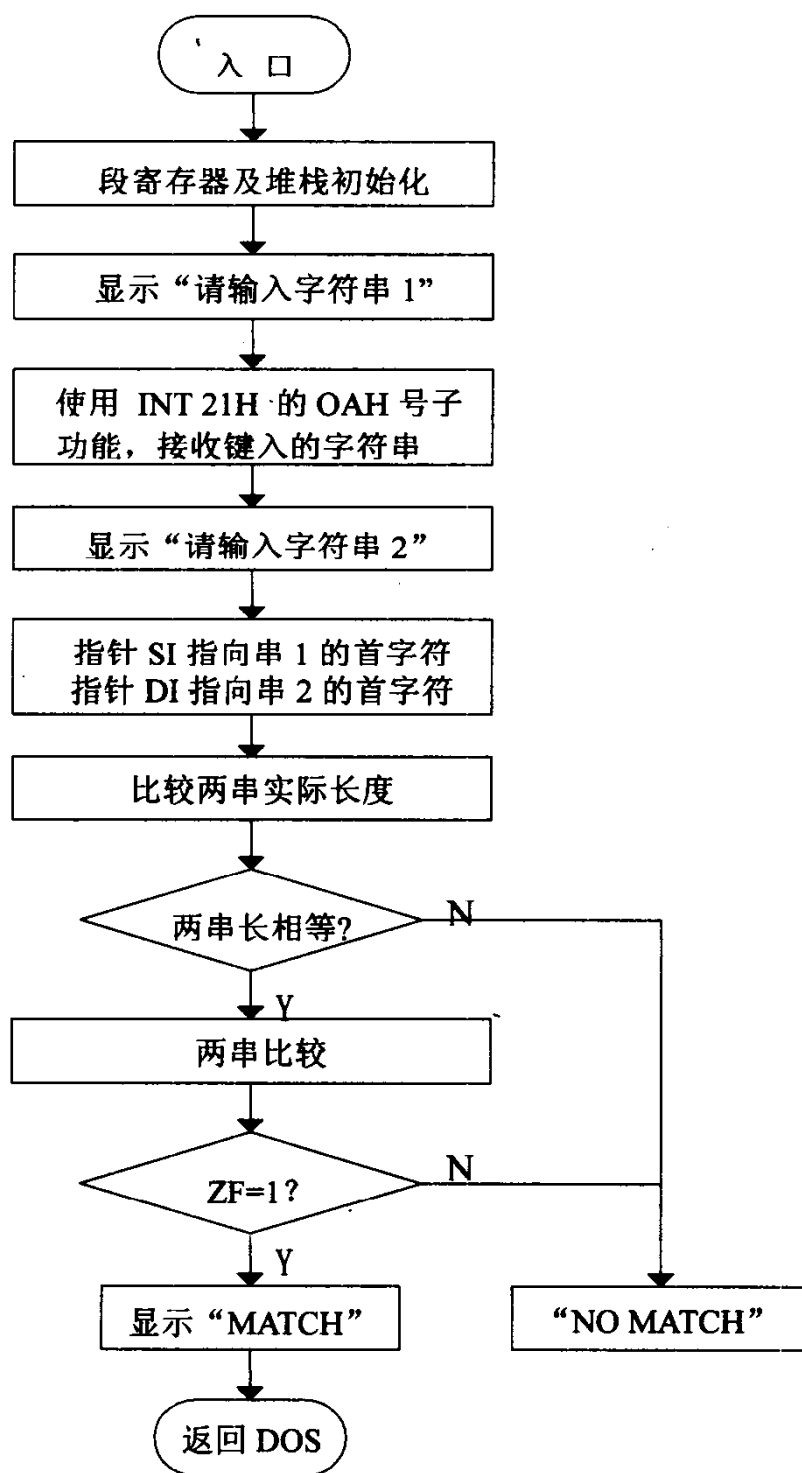


图 1-1 字符匹配程序实验流程图

思考：若计数值 ≥ 10 时，考虑如何将 BX 内容转换成十进制数，并显示结果。

1.3.5 实验五 将键入的字符进行转换实验

一、实验目的

了解小写字母和大写字母在计算机的表示方法，并学习如何进行转换。

二、实验内容

接收键盘字符（以“Q”为结束符），并将其中的小写字母转变为大写字母，然后进行显示。

三、参考流程

字符大小写转换实验流程如图 1-3 所示。

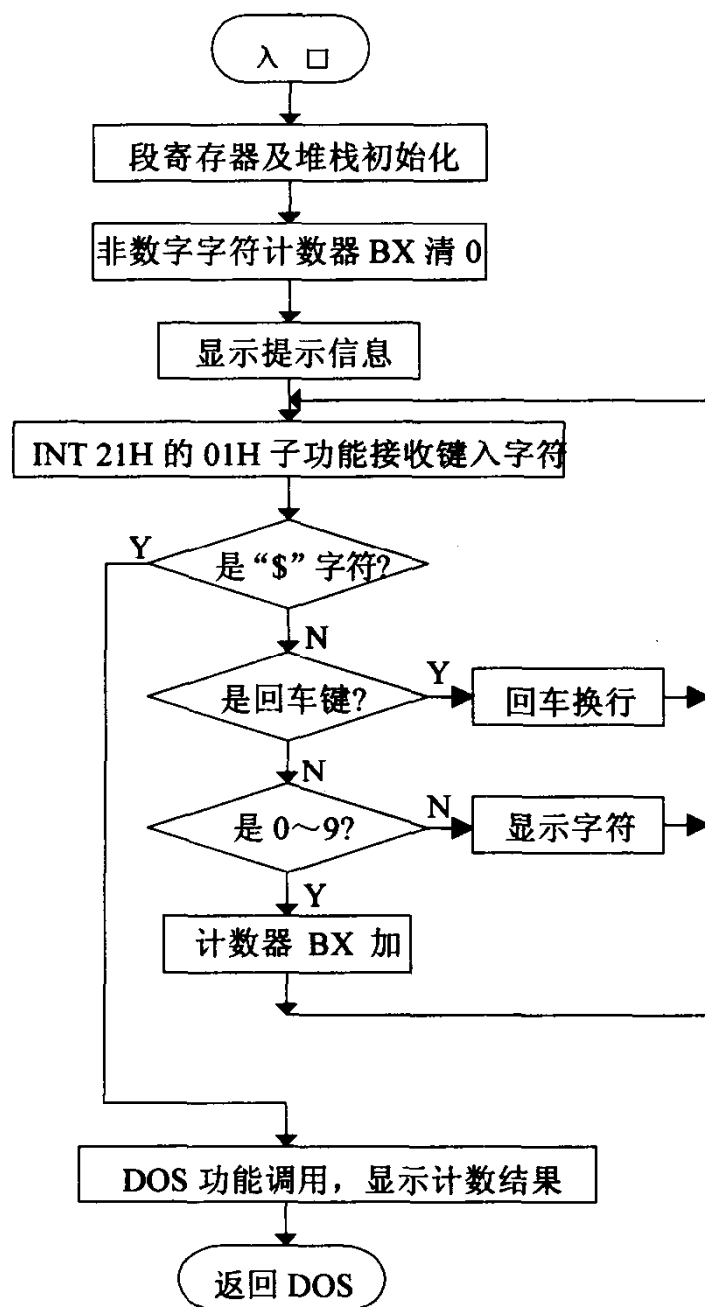


图 1-2 字符和数据显示实验流程图

1.3.6 实验六 排序实验

一、实验目的

掌握用汇编语言编写排序的思路和方法。

二、实验内容

从首地址为 1000H 开始存放 50 个数, 要求设计程序将这些数由小到大排序, 排序后的数仍放在此区域中。

三、参考流程

数据排序实验流程如图 1-4 所示。

1.3.7 实验七 文件读、写操作实验

一、实验目的

掌握写文件的 DOS 功能调用方法。

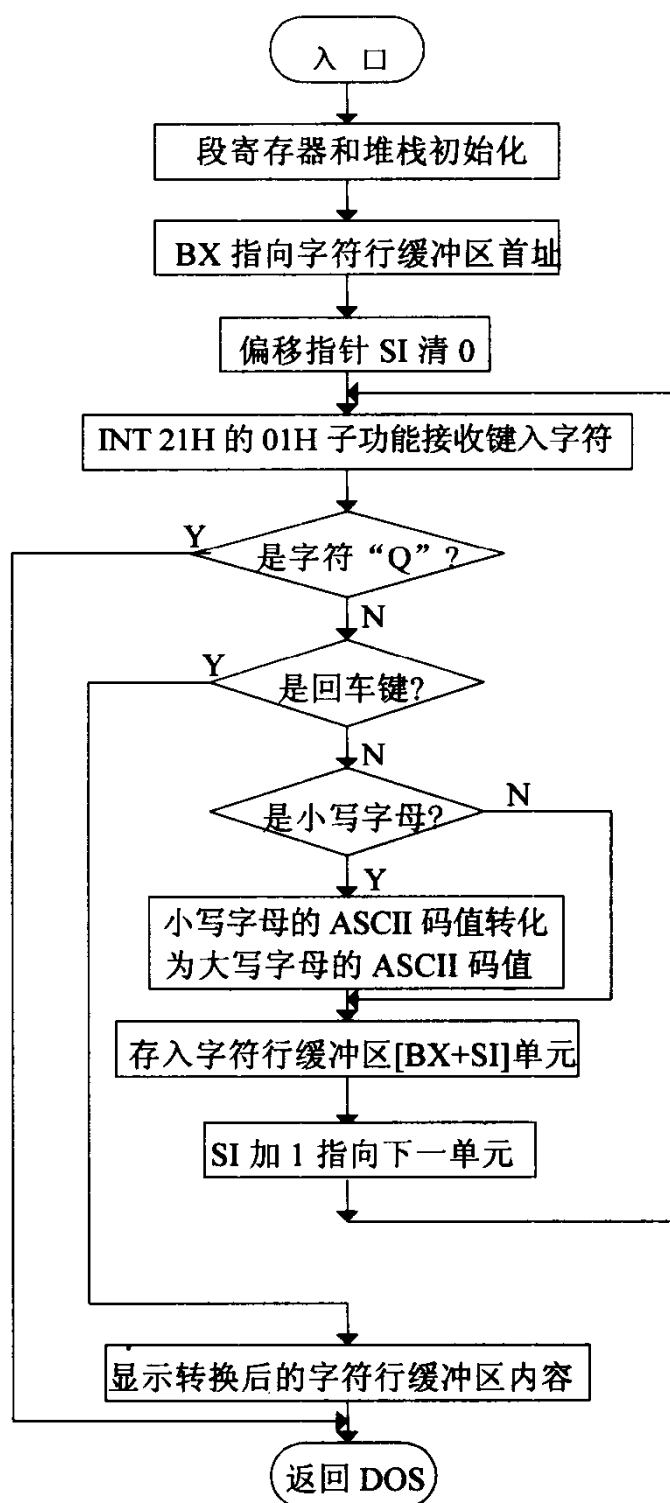


图 1-3 字符大小写转换实验流程图

二、实验内容

编写程序，将内存区域中用调试程序设置好的一连串数据（以 CTRL-Z 为结束符）作为一个文件存入磁盘，文件名为 DATA.AAA。

编写程序，使它相当于 TYPE 命令的功能。先由屏幕显示提示信息，再读取文件名（含路径），然后在屏幕上显示文件内容。

三、编程提示

1. 对于文件的读写操作，DOS 提供了两种手段。第一种手段是使用 FCB 进行存取；第二种手段是用文件代号法进行存取。文件代号法支持目录路径，并且对错误采用了更统一的办法处理，是 DOS 推荐的存取的方法。

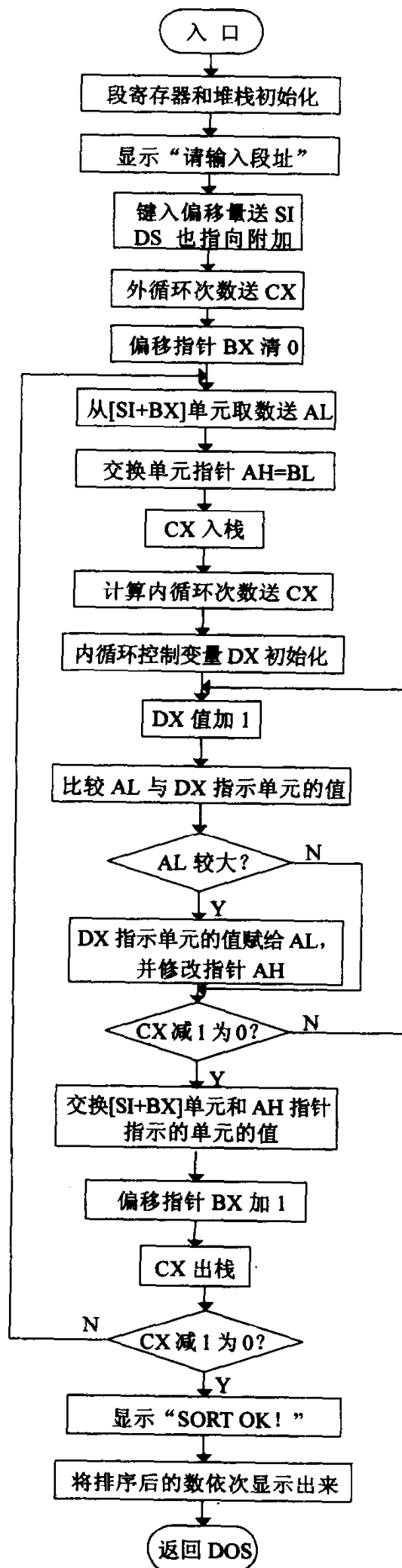


图 1-4 数据排序实验流程图

2. 使用文件代号法, 要求文件名(含路径)用 ASCII 字符串, 且以空字符 00H 结尾。

3. 文件代号法存取出现错误, 则功能调用返回时, 要将 CY 标志置 1, 同时在 AX 寄存器中返回统一的出错代码。程序中要用 JC 指令进行错误判定, 确认出错后再将错误代码显示出来, 以便分析出错原因。