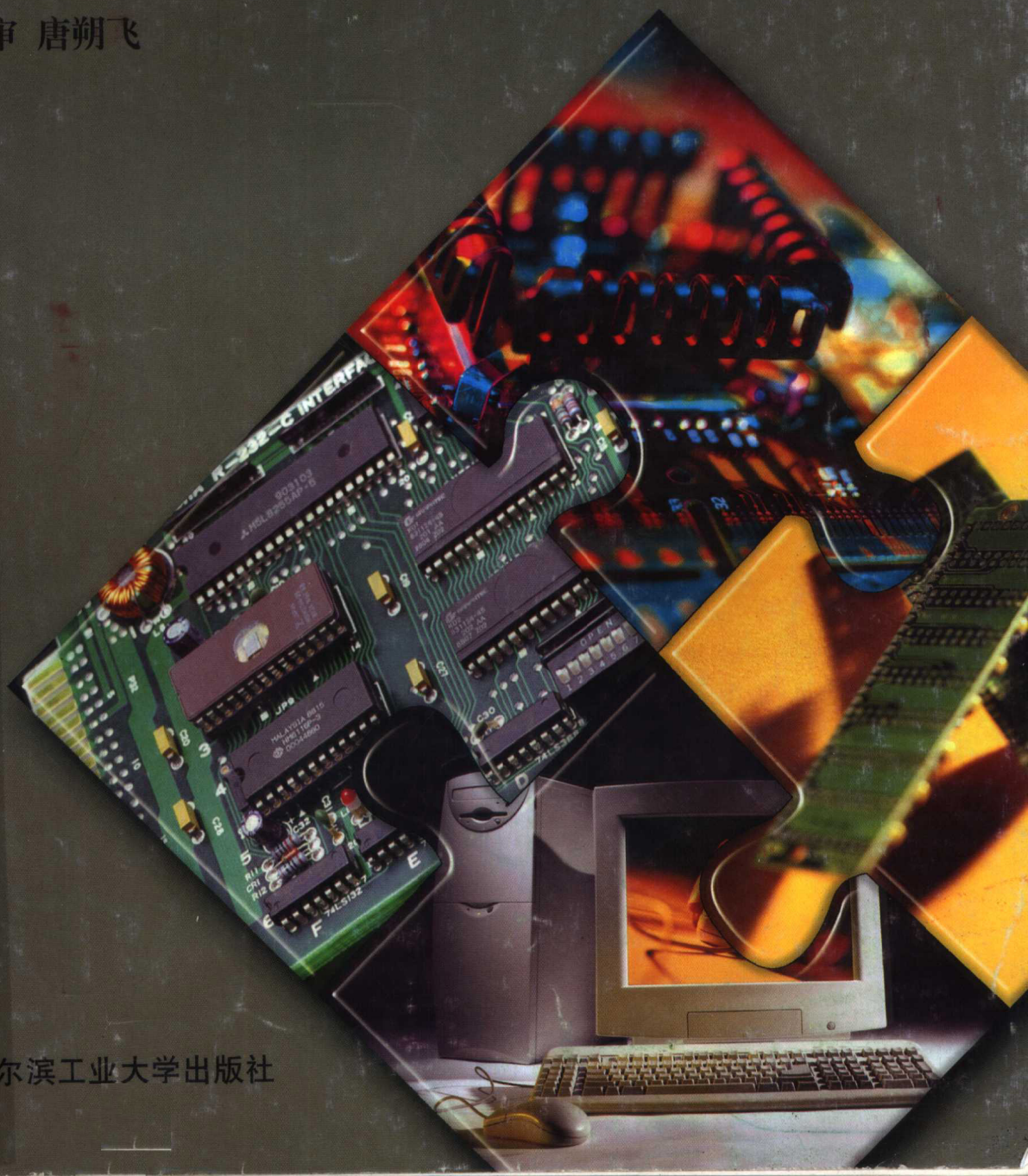


# 计算机组成技术实验指导书

主编 杜惠平 姜新 徐晓华

主审 唐朔飞



哈尔滨工业大学出版社

# 计算机组成技术 实验指导书

主编 杜惠平 姜 新 徐晓华  
主审 唐朔飞

哈尔滨工业大学出版社  
哈尔滨

**图书在版编目(CIP)数据**

计算机组成技术实验指导书/杜惠平主编. —哈尔滨:  
哈尔滨工业大学出版社, 2002. 10

ISBN 7-5603-1794-4

I. 计… II. 杜… III. 计算机体系结构-高等学  
校-教材 IV. TP303

中国版本图书馆 CIP 数据核字(2002)第 081287 号

**出版发行** 哈尔滨工业大学出版社

**社 址** 哈尔滨市南岗区教化街 21 号 邮编 150006

**传 真** 0451 - 6414749

**印 刷** 哈尔滨工业大学印刷厂

**开 本** 787 × 1092 1/16 印张 12.25 插页 1 字数 246 千字

**版 次** 2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

**书 号** ISBN 7-5603-1794-4/TP·180

**印 数** 1 ~ 3 000

**定 价** 15.00 元(附实验报告)

# 目 录

|                               |    |
|-------------------------------|----|
| TDS - MD 实验系统使用介绍 .....       | 1  |
| 实验一 运算类编程实验.....              | 6  |
| 实验二 数码转换编程及程序调试 .....         | 12 |
| 实验三 存储器扩展实验 .....             | 18 |
| 实验四 8255 并行接口应用实验 .....       | 22 |
| 实验五 中断特性及 8259 应用编程实验 .....   | 27 |
| 实验六 8253 定时/计时器应用实验 .....     | 34 |
| 实验七 DMA 特性及 8237 应用实验.....    | 39 |
| 实验八 8251 串行接口应用实验 .....       | 45 |
| 实验九 双机通讯 .....                | 50 |
| 实验十 A/D 转换实验 .....            | 53 |
| 实验十一 D/A 转换实验 .....           | 57 |
| 实验十二 运用 DEBUG 调试工具编程.....     | 60 |
| 实验十三 顺序结构程序设计 .....           | 62 |
| 实验十四 分支与循环结构程序设计 .....        | 64 |
| 实验十五 子程序结构程序设计及系统功能调用 .....   | 66 |
| 附录 1 与 PC 微机联机使用说明 .....      | 67 |
| 附录 2 系统编程信息 .....             | 70 |
| 附录 3 汇编语言程序上机过程 .....         | 74 |
| 附录 4 8086/8088 指令表 .....      | 83 |
| 附录 5 汇编程序出错信息 .....           | 91 |
| 附录 6 液晶显示器使用说明 .....          | 96 |
| 附录 7 实验芯片管脚图 .....            | 97 |
| 附录 8 IWATSU 数字存储示波器使用说明 ..... | 99 |

# TDS—MD 实验系统使用介绍

## 一、系统概述

TDS - MD 微机教学实验及开发系统是为了适应十六位微机原理及接口、十六位微机技术、单片机原理及应用课程的教学实验而设计的微机系统。其系统构成见下表。

| 系统构成方式    | 最小工作方式                                                            |                              |
|-----------|-------------------------------------------------------------------|------------------------------|
| CPU       | 8088/4.7 MHz 8031/6 MHz                                           |                              |
| 存储器       | 系统程序                                                              | EPROM 27512(64K)             |
|           | 数 据                                                               | SRAM 62256(32K)              |
|           | 待扩展                                                               | EPROM 27256, SRAM 62256(32K) |
| 接口芯片及试验单元 | 8251、8253、8255、8259、8259 级连、8237、6264 存储器扩展、A/D、D/A 转换、DIP 开关及发光管 |                              |
| EPROM 编程器 | 可对 EPROM2716 - 27512 快速编程                                         |                              |
| 显示器       | 液晶显示器 2 行 40 列                                                    |                              |
| 键 盘       | 标准 101 键盘                                                         |                              |
| 外设接口      | RS - 232C 串行通讯接口、打印机接口、PC 总线接口、34&40 线外接实验扩展器接口                   |                              |

此系统具有与 PC 机一致的 DEBUG 操作界面以及与 PC 机兼容的 BIOS 功能调用界面,可直接使用汇编语言输入和调试程序,并具有反汇编功能。

## 二、系统操作

### 1. 启动

将系统电源接至 220V 市电,然后打开位于系统左上角的电源开关,则电源指示灯亮,同时液晶显示器上显示:“—WELCOME TO YOU!”。

稍待片刻,显示器上则显示提示符“>”或“—”及光标也在闪烁。“>”为 8088 系统提示符,“—”为 8051 系统提示符。

如果开启电源后,显示“—”提示符,则此时系统工作为 8051 系统状态。而我们要进入 8088 系统状态,则要键入“8”字符,再按“ENTER”键,则显示器上显“>”提示符,此时为 8088 系统工作状态。

如果开启电源后,显示“>”提示符,则此时系统工作为 8088 系统状态。而我们要进入 8051 系统状态,则要键入“5”字符,再按“ENTER”键,则显示器上显示“—”提示符,此时为 8051 系统工作状态。

## 2. 复位

TDS-MD 机右下角有一复位开关,每按动一下,则对系统产生一次复位操作,每次液晶显示器上都显示“—WELCOME TO YOU!”字符,稍待片刻,则显示提示符及光标。此时只对 CPU、接口复位和清断点。

此系统机内存储单元有掉电保护功能,所以按动复位开关及关闭系统电源,系统存储单元中的内容仍不会改变。

## 三、DEBUG 命令及操作

TDS-MD 机具有同 PC 机一致的 DEBUG 操作界面,以及与 PC 机兼容的基本输入、输出系统(BIOS)。

以下所用的“↵”字符为“ENTER”键,“x”为所要键入的字符标志。

### 1. A 汇编程序命令

> A x x x x ↵ ; x x x x 为所要输入程序的首地址。

> A ↵ ;此时默认的地址为 2000H。

### 2. B 断点设置命令

> B ↵ ;为所要设置断点的命令。

> 0: x x x x ↵ ; x x x x 为所要设置的第一个断点地址。

> 1: ↵ ;此时如果继续输入 4 位地址,则是第二个断点地址。  
系统允许最多设置 10 个断点。如果此时键入“ENTER”键,则表明断点就设到这里为止。清除断点的方法有两种,按复位键或重新上电均可。

### 3. D 显示地址单元中的数据命令

> D x x x x , x x x x ↵ ;第一个 x x x x 地址为要显示单元首地址。  
第二个 x x x x 地址为要显示单元末地址。

> D x x x x ↵ ;如果此时不键入第二个末地址,那么只显示这个单元首地址等 8 个字节单元的内容。

### 4. E 修改地址单元中的数据命令

> E x x x x ↵ ; x x x x 为要修改的地址单元。

> 0000: x x x x = x x ; x x 为要修改的地址单元中的数据,如果此时要修改它,只需键入要修改的数据,再键入“↵”键则可。

一旦进入 E 命令状态,就可通过“SPACE”键使存储单元地址 x x x x 向高地址移动,通过“—”键使存储单元地址向低地址移动,可方便地查看存储单元中的数据或直接填入新数据,来修改某一地址存储单元中的数据。若键入“↵”,就退出 E 命令。

## 5. G 连续运行程序命令

> G = x x x x ↵

; x x x x 为连续运行程序的首地址, G 格式表示无断点连续运行命令。

> G B = x x x x ↵

; x x x x 为要运行程序的首地址, GB 格式表示带断点连续运行命令。但执行到断点地址时, 运行程序就自动停止。

## 6. M 数据块搬家命令

> M x x x x , x x x x x x x x ↵; 第一个 x x x x 是要搬家数据块的首地址, 在其后要键入一个空格键, 第二个 x x x x 为要搬家数据块的末地址, 在其后要键入一个空格键, 最后一个 x x x x 为要搬家的数据块所要移到的地址单元的首地址。

## 7. R 寄存器显示与修改命令

> R ↵

显示 CS = x x x x , IP = x x x x , DS = x x x x , AX = x x x x , FX = x x x x

> R 寄存器名 ↵

; 显示寄存器内容。如键入“RCS”则显示 CS = x x x x \_\_, 如果要修改 CS 内容, 则在“\_\_”后键 4 位数据, 再键入“↵”则可。

## 8. T 单步运行程序命令

> T = x x x x ↵

; 表示从 x x x x 地址起单步执行一条命令, 每执行完毕一次, 系统就会显示 CS, DS, IP, AX 的内容。

## 9. U 反汇编程序命令

> U x x x x ↵

; x x x x 为反汇编程序的首地址, 只显示首地址的当前程序。

> U x x x x x x x x ↵

; 第一个 x x x x 为反汇编程序的首地址, 在其后要键入一个空格键, 第二个为反汇编程序的末地址。显示从首地址到末地址间的程序代码及反汇编程序。

# 四、DEBUG 命令练习

## 1. 练习程序

从 3500H 内存单元开始建立 0~15 共 16 条数据。程序如下:



| 地址(H) | 助 记 符        | 注 释     |
|-------|--------------|---------|
| 2000  | MOV DI,3500  | ;设数据区首址 |
| 2003  | MOV CX,0010  | ;字节数→CX |
| 2006  | MOV AX,0000  |         |
| 2009  | SAHF         |         |
| 200A  | MOV B[DI],AL | ;写入一字节  |
| 200C  | INC DI       | ;修改地址指针 |
| 200D  | INC AX       | ;修改数据   |
| 200E  | DAA          | ;十进制调整  |
| 200F  | LOOP 200A    | ;未填完转移  |
| 2011  | JMP 2011     |         |

## 2. 源程序录入

开机后出现“>”提示符后即可输入源程序,过程如下:

```
> A 2 0 0 0↵
> 0000:2000      MOV DI,3500↵
> 0000:2003      MOV CX,0010↵
> 0000:2006
:
> 0000:2011      JMP 2011↵
> 0000:2013↵      ;源程序录入完则可按“↵”键,退出编辑状态,返回系统。
>
```

输入程序过程中,若在回车前发现本行有错误字符,则可按“↵”键回抹并重新输入。当按“↵”键后,系统提示“E”时,地址指针仍会停留在当前行,待你重新输入正确程序。

## 3. 用 U 命令进行反汇编

源程序输入后,必须进行反汇编,查看程序是否正确。

```
> U x x x x↵
> 2000      BF0035      MOV      DI,3500
> U↵
> 2003      B91000      MOV      CX,0010
> U↵
:
```

每键入一次“U↵”则显示一条程序和机器码,一直把所有录入程序检查完毕即可。



#### 4. 修改源程序

当发现源程序有错误时,可做相应的修改。例如修改 2000 句 MOV DI,3500 为 MOV DI,2500。

```
> A 2 0 0 0↵
```

```
> 0000:2000      MOV DI,2500↵
```

```
> 0000:2003↵
```

```
>
```

#### 5. 存储单元的显示与修改

例:查看 3500H~3507H 单元的内容,0~7 共 8 个字节数据。

```
> D 3 5 0 0↵
```

```
> 0000:3500 x x x x x x x x x x x x x x      ; x x 为 3500H~3507H  
                                         单元中的数据。
```

例:如果要修改 3500H~3507H 单元中的内容。

```
> E 3 5 0 0↵
```

```
> 0000:3500 = AA _ CC↵      ; 现在 3500 中的内容为 AA,如果要修改为 CC,只要键入  
                             “CC”则可。如果此时还要改 3501 等单元中的内容,只需  
                             按一下“空格键”,重复上述步骤即可。否则,按“ENTER”  
                             键,返回到“>”状态。
```

### 五、特别注意事项

1. 系统电源关闭后,不能立即开启。关闭到重新开启至少需要间隔 30 秒钟。
2. 系统使用完毕后,必须先关闭机内电源,然后拔掉电源插头,将电源线放入机箱右侧线盒内。
3. 实验使用排线连线时,请将排线稍作倾斜,晃动插接到排针上,注意手感,不要强力插接,以免排线损坏。
4. 开机后观察提示符,确定所进入的系统是那个 CPU 系统。若不是所需系统,请按启动一节的方法切换。
5. 汇编程序的数字一律是不带 H 后缀的 16 进制数。
6. MOV 类操作一定要在[ ]之前标注 W(字)或 B(字节),如 MOV B[2010],AL;  
MOV W[2010],AX。

# 实验一 运算类编程实验

## 一、实验目的

1. 掌握使用运算类指令编程及调试的方法。
2. 掌握运算类指令对各状态标志位的影响及其测试方法。

## 二、实验设备

TDS - MD 微机一台。

## 三、实验内容及步骤

8086/8088 指令系统提供了实现加、减、乘、除运算的基本指令,可对数据类型算术运算表所示的数据类型进行算术运算。

数据类型算术运算表:

| 数制  | 二进制      |     | BCD 码    |          |
|-----|----------|-----|----------|----------|
|     | 带符合      | 无符号 | 组合       | 非组合      |
| 运算符 | +、-、×、÷  |     | +、-      | +、-、×、÷  |
| 操作数 | 字节、字、多精度 |     | 字节(二位数字) | 字节(一位数字) |

### (一) 加减运算

#### 1. 二进制双精度加法运算

计算  $x + y = z$ , 将结果  $z$  存入某存储单元。本实验参考程序及存储单元分配如图 1 所示。

| 地址(H) | 助记符             | 注释          |
|-------|-----------------|-------------|
| 2000  | MOV AX, W[3500] | ;X 低位送 AX   |
| 2003  | ADD AX, W[3504] | ;X 低位加 y 低位 |
| 2007  | MOV W[3508], AX | ;存低位和       |
| 200A  | MOV AX, W[3502] | ;X 高位送 AX   |
| 200D  | ADC AX, W[3506] | ;X 高位加 y 高位 |
| 2011  | MOV W[350A], AX | ;存高位和       |
| 2014  | INT             |             |

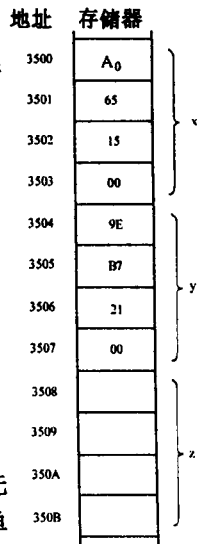


图 1

本实验程序是双精度(2 个 16 位,即 32 位)运算,利用累加器 AX,先求低 16 位和,并存入低址存储单元,后求高 16 位和,再存入高址存储单元。由于低位和可能向高位有进位,因而高位字相加语句须用 ADC 指令,则低位相加有进位时,CF = 1,高位字相加时,同时加上 CF 中的 1。

## 实验步骤

- (1) 输入实验程序并检查无误。
- (2) 按存储单元分配图(图 1), 用  $> E3500 \leftarrow$  命令, 在 3500H ~ 3507H 单元中存入二进制数 A0651500 和 9EB72100。
- (3)  $G = 2000 \leftarrow$  运行程序。
- (4)  $D3508 \leftarrow$  显示计算结果: 3508 3E 1D 37 00 ...。
- (5) 反复试几组数, 考查程序的正确性。

## 2. 十进制数的 BCD 码相减运算

计算  $x - y = z$ , 其中  $x, y, z$  为 BCD 码。本实验程序流程图, 参考程序及存储单元分配如图 2 所示。

| 地址(H) | 助记符             |
|-------|-----------------|
| 2100  | MOV AH,00       |
| 2102  | SAHF            |
| 2103  | MOV CX,0002     |
| 2106  | MOV SI,3500     |
| 2109  | MOV DI,3510     |
| 210C  | MOV AL,B[SI]    |
| 210E  | SBB AL,B[SI+02] |
| 2111  | DAS             |
| 2112  | PUSH F          |
| 2113  | AND AL,0F       |
| 2115  | POP F           |
| 2116  | MOV B[DI],AL    |
| 2118  | INC DI          |
| 2119  | INC SI          |
| 211A  | LOOP 210C       |
| 211C  | INT             |

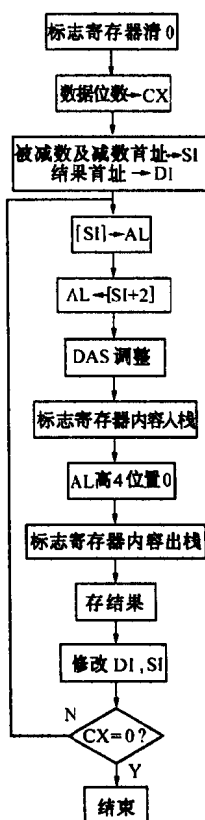
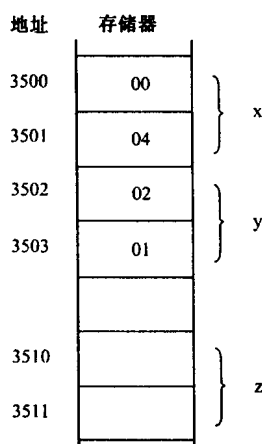


图 2

## 实验步骤

- (1) 输入实验程序并检查无误。
- (2)  $> E3500 \leftarrow$  在 3500H ~ 3503H 单元中存入 40 和 12 的 BCD 码: 00、04、02、01。
- (3)  $> G = 2100 \leftarrow$  运行程序。
- (4)  $> D3510 \leftarrow$ , 显示计算结果: 3501 08 02 ...。

(5)反复试几组数,考查程序的正确性。

## (二)乘除运算

### 1.考察乘法指令 MUL、IMUL 对状态标志位的影响

乘法指令 MUL、IMUL 对状态标志 CF、OF 都是通过指令执行后最高位上产生进(借)位、溢出来影响的,图 3 我们具体给出了一个实验程序的流程图及参考程序,其设计思想是:取 3000H 单元开始的 10 个无符号数,将其各自乘以 2,若有溢出,显示“0”,有进位显示“C”,否则,只显示间隔符“,”,我们规定每一数相应的标志显示之间均用“,”来间隔。于是,我们就可以在数据区放各种试验数来考查乘法对标志的影响。

| 地址(H) | 助记符          | 注释      |
|-------|--------------|---------|
| 2200  | MOV SI,3000  | ;源数据首址  |
| 2203  | MOV CX,000A  | ;数据个数   |
| 2206  | MOV BX,0002  |         |
| 2209  | MOV AX,W[SI] | ;取数     |
| 220B  | CLC          |         |
| 220C  | IMUL BX      | ;乘 2    |
| 220E  | JO 221E      | ;溢出转    |
| 2210  | JB 2222      |         |
| 2212  | MOV AL,2C    | ;显示“,”  |
| 2214  | INC SI       |         |
| 2215  | INC SI       |         |
| 2216  | MOV AH,01    | ;显示功能调用 |
| 2218  | INT 10       |         |
| 221A  | LOOP 2209    | ;完否     |
| 221C  | JMP 221C     | ;停机     |
| 221E  | MOV AL,4F    | ;显示“0”  |
| 2220  | JMP 2214     |         |
| 2222  | MOV AL,43    | ;显示“C”  |
| 2224  | JMP 2214     |         |
| 2226  | INT          |         |

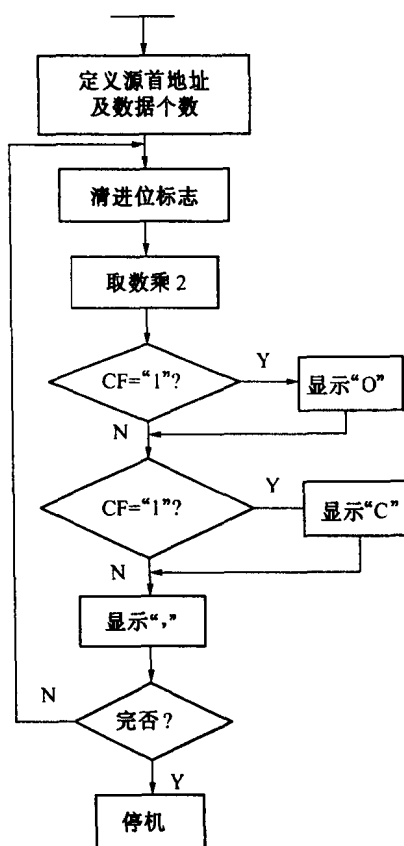


图 3

## 实验步骤

- (1) 输入实验程序并检查无误。
  - (2) 准备 10 个数存放 3000H~3009H 存储单元中,运行程序,观察显示结果并认真分析。
  - (3) 反复试几组数,观察结果。
  - (4) 用无符号 MUL 指令代替带符号的整数乘法 IMUL 指令,重复步骤(2)(3)。
- 2.考察除法指令 DIV、IDIV 以及 ASCII 码调整指令对标志的影响情况

除法指令 DIV、IDIV 执行后,不影响任何状态标志位,而 ASCII 码调整指令会影响 PF、SF、ZF,对 AF、CF、OF 标志位影响不确定。

本实验程序流程图及参考程序如图 4 所示,其设计思想是:取 3000H ~ 3009H 存储单元中的五个无符号数,将其用 101H 除,若有溢出或进位,就分别显示“O”或“C”在进行 ASCII 码转换后,若奇偶位为偶(PF = 1),则显示“P”,为负(SF = 1)显示“S”。为了解每个数运算后对状态标志位的影响,在每个数运算完后均显示“;”。

| 地址(H) | 助记符          | 注释         |
|-------|--------------|------------|
| 2300  | MOV SI,3000  |            |
| 2303  | MOV CX,0005  |            |
| 2306  | MOV DX,0000  |            |
| 2309  | MOV AH,00    |            |
| 230B  | SAHF         | ;消标志位      |
| 230C  | MOV AX,W[SI] | ;取数        |
| 230E  | MOV BX,0101  |            |
| 2311  | DIV BX       | ;除 101     |
| 2313  | AAD          | ;ASCII 码调整 |
| 2315  | JO 2329      |            |
| 2317  | JB 232D      |            |
| 2319  | JP 2331      |            |
| 231B  | JS 2335      |            |
| 231D  | MOV AL,3B    | ;显示“;”     |
| 231F  | INC SI       |            |
| 2320  | INC SI       |            |
| 2321  | MOV AH,01    |            |
| 2323  | INT 10       |            |
| 2325  | LOOP 2306    |            |
| 2327  | JMP 2327     |            |
| 2329  | MOV AL,4F    | ;显示“O”     |
| 232B  | JMP 231F     |            |
| 232D  | MOV AL,43    | ;显示“C”     |
| 232F  | JMP 231F     |            |
| 2331  | MOV AL,50    | ;显示“P”     |
| 2333  | JMP 231F     |            |
| 2335  | MOV AL,53    | ;显示“S”     |
| 2337  | JMP 231F     |            |
| 2339  | INT          |            |

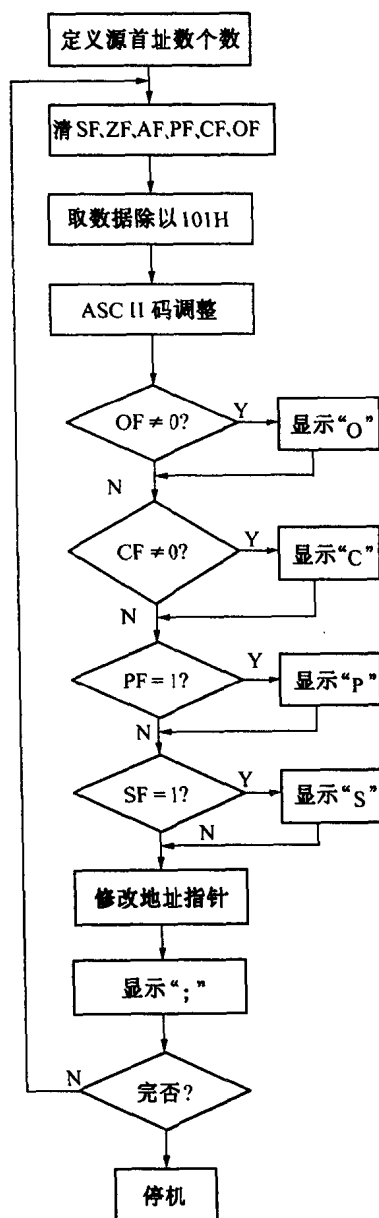


图 4

## 实验步骤

- (1) 输入程序并调通, 在 3000H ~ 3009H 存储单元中填入 5 个 16 位数。
- (2) 用 DIV 除法指令, 判定它对 CF, OF, PF, SF 标志的影响, 记录显示结果。
- (3) 用 IDIV 除法指令代替 DIV 执行程序, 记录结果。

### (三) 开平方运算

8086/8088 指令系统中有乘除法指令但没有开平方指令。因此, 开平方运算是通过程序来实现的。用减奇数法开平方, 用减奇数法可求得近似平方根, 获得平方根的整数部分。

我们知道,  $N$  个自然数中的奇数之和等于  $N^2$ , 即:  $\sum_{k=1}^n (2K - 1) = N^2$

$$1 + 3 + 5 = 9 = 3^2$$

$$1 + 3 + 5 + 7 = 16 = 4^2$$

$$1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 = 64 = 8^2$$

若现要作  $\sqrt{S}$  的运算, 那么就可以从  $S$  中逐次减去自然数中的奇数 1, 3, 5, ..., 一直进行到相减数为零或不够减下一个自然数的奇数为止, 然后统计减去自然数的奇数个数, 它就是  $S$  的近似平方根。设被开平方数为正整数, 存放在 3500H 单元。程序流程图如图 5 所示, 实验参考程序如下;

| 地址(H) | 助记符             | 注释           |
|-------|-----------------|--------------|
| 2400  | MOV AX, W[3500] | ; 取被开方数      |
| 2403  | MOV CL, 00      | ; 清平方根       |
| 2405  | MOV DX, 0001    | ; 奇数初值       |
| 2408  | SUB AX, DX      | ; 减奇数        |
| 240A  | JB 2413         | ; 不够减则结束, 存根 |
| 240C  | INC CL          | ; 够减, 奇数个数计数 |
| 240E  | ADD DX, 02      | ; 形成下一个奇数    |
| 2411  | JMP 2408        |              |
| 2413  | MOV B[350A], CL | ; 存平方根       |
| 2417  | INT             | ; 返回监控       |

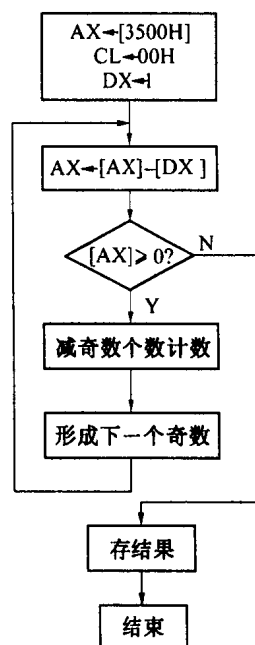


图 5

## 实验步骤

- (1) 输入程序并检查无误。
- (2) 在 3500H ~ 3501H 单元中存入被开方数 0010, 用 > E3500  
↵ 输入 10、00。

- (3) > G = 2400 ↵ 运行程序。
- (4) > D3500 ↵ 显示结果为: 350A 04 00 ...。
- (5) 反复试几组数, 考查程序的正确性。

#### 四、思考题

1. 编写有符号数  $A_1B_1 + A_2B_2$  的程序,  $A_1A_2B_1B_2$  均为符号数。
2. 编写两个数值长度不等的 BCD 码相加程序。
3. 为什么乘法指令执行后, OF、CF 同时为“1”或同时为“0”? 有没有不同时为“0”或不同时为“1”的情况呢?
4. 乘法指令 MUL 和 IMUL, 除法指令 DIV 和 IDIV 对状态标志位的影响一样吗?
5. 试编程, 由键盘输入一个十进制数, 并将其转换为二进制数, 并进行开平方运算。



## 实验二 数码转换编程及程序调试

### 一、实验目的

1. 掌握不同进制数及编码相互转换的程序设计方法,加深对数码转换的理解。
2. 进一步熟悉键盘使用方法。
3. 进一步熟悉调试程序的方法。

### 二、实验设备

TDS—MD 微机一台

### 三、实验内容及步骤

计算机输入设备输入的信息一般是 ASCII 码或 BCD 码表示的数据或字符,CPU 处理信息一般均用二进制数进行计算或其它处理,处理结果输出的外设又必须依外设的要求变为 ASCII 码、BCD 码或七段显示码等。因此,在应用软件中各类数制的转换和代码的转换是必不可少的。

计算机与外设间的数码转换关系如图 1 所示,数码对应关系如表 1 所示。

表 1 数码对应关系

| 十六进制数 | BCD 码 | 二进制<br>机器码 | ASCII 码 | 七段码 |     |
|-------|-------|------------|---------|-----|-----|
|       |       |            |         | 共阳  | 共阴  |
| 0     | 0000  | 0000       | 30H     | 40H | 3FH |
| 1     | 0001  | 0001       | 31H     | 79H | 06H |
| 2     | 0010  | 0010       | 32H     | 24H | 5BH |
| 3     | 0011  | 0011       | 33H     | 30H | 4FH |
| 4     | 0100  | 0100       | 34H     | 19H | 66H |
| 5     | 0101  | 0101       | 35H     | 12H | 6DH |
| 6     | 0110  | 0110       | 36H     | 02H | 7DH |
| 7     | 0111  | 0111       | 37H     | 78H | 07H |
| 8     | 1000  | 1000       | 38H     | 00H | 7FH |
| 9     | 1001  | 1001       | 39H     | 18H | 67H |
| A     |       | 1010       | 41H     | 08H | 77H |
| B     |       | 1011       | 42H     | 03H | 7CH |
| C     |       | 1100       | 43H     | 46H | 39H |
| D     |       | 1101       | 44H     | 21H | 5EH |
| E     |       | 1110       | 45H     | 06H | 79H |
| F     |       | 1111       | 46H     | 0EH | 71H |

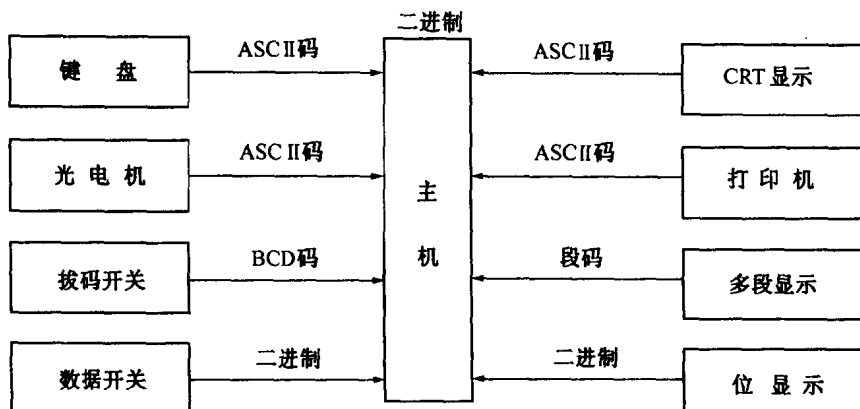


图 1

### 1. 将 ASCII 码表示的十进制数转换为二进制数

十进制数可表示为:

$$D_n \times 10^n + D_{n-1} \times 10^{n-1} + \cdots + D_0 \times 10^0 = \sum_{i=0}^n D_i \times 10^i \quad (1)$$

$D_i$  代表十进制数 1,2,3...9,0; 上式可转换为:

$$\sum_{i=0}^n D_i \times 10^i = ((D_n \times 10 + D_{n-1}) \times 100) + D_{n-2}) \times 10 + \cdots + D_1) \times 10 + D_0 \quad (2)$$

由式(2)可归纳十进制数转换为二进制数的方法:从十进制的最高位  $D_n$  开始作乘 10 加次位的操作,将结果再乘 10 再加下一个次位,如此重复,则可求出二进制数结果来。程序流程图及参考程序如图 2。这里我们规定:被转换 ASCII 码十进制数存放在 3500H ~ 3504H 单元中,而转换结果存在 3510 ~ 3511H 单元中。

| 地址(H) | 助记符             |
|-------|-----------------|
| 2000  | MOV SI,3500     |
| 2003  | MOV DI,3510     |
| 2006  | MOV BX,000A     |
| 2009  | MOV CX,0004     |
| 200C  | MOV AH,00       |
| 200E  | MOV AL,B[SI]    |
| 2010  | SUB AL,30       |
| 2012  | IMUL BX         |
| 2014  | ADD AL,B[SI+01] |
| 2017  | SUB AL,30       |
| 2019  | INC SI          |
| 201A  | LOOP 2012       |
| 201C  | MOV W[DI],AX    |
| 201E  | INT             |

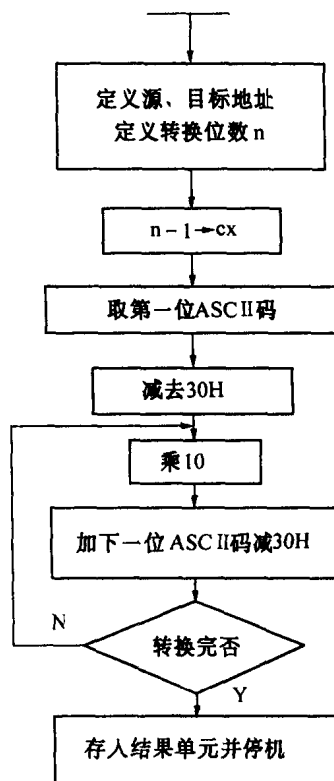


图 2