



自动化类专业系列实验教材  
AUTOMATION

EXPERIMENT OF MICROCOMPUTER PRINCIPLE  
AND INTERFACE

# 微型计算机原理与接口 实验技术

曾薄文◎编著



清华大学出版社



自动化类专业系列实验教材  
AUTOMATION

EXPERIMENT OF MICROCOMPUTER PRINCIPLE  
AND INTERFACE

# 微型计算机原理与接口 实验技术

曾薄文◎编著

清华大学出版社  
北京

## 内 容 简 介

本书是为配合“微型计算机原理与接口技术”课程而编写的实验教材,介绍了实验教学中所涉及的实验内容。书中编写了基础性、综合性和创新性三种类型实验,以适应不同层次读者需求。

全书共分6章,第1~2章介绍汇编语言程序设计实验环境和步骤以及几种典型的汇编语言程序设计;第3章介绍微机接口实验平台PITA硬件资源及其使用方法;第4章为微机接口技术基础性实验,介绍几种典型接口芯片使用以及存储器扩展等方面内容;第5章为微机接口技术综合性实验,介绍微型计算机在计算机控制系统中的应用,具有较强的综合性,涵盖了前面大部分知识;第6章为研究创新性实验,以51单片机在工程实际中的应用为背景,包括简单的采样系统设计、典型的控制系统设计以及集散控制系统设计。

本书结合了作者的实验教学和实际工程经验,实例丰富,实用性强,可供大专院校师生使用,也可为相关工程技术人员提供参考。

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实验技术/曾薄文编著. —北京:清华大学出版社,2016

自动化类专业系列实验教材

ISBN 978-7-302-41470-4

I. ①微… II. ①曾… III. ①微型计算机—理论—高等学校—教材 ②微型计算机—接口技术—高等学校—教材 IV. ①TP36-33

中国版本图书馆CIP数据核字(2015)第212282号

责任编辑:文怡

封面设计:李召霞

责任校对:梁毅

责任印制:李红英

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](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

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

印 装 者:三河市中晟雅豪印务有限公司

经 销:全国新华书店

开 本:185mm×230mm

印 张:14.75

字 数:338千字

版 次:2016年9月第1版

印 次:2016年9月第1次印刷

印 数:1~2000

定 价:29.00元

产品编号:059004-01

“微型计算机原理与接口技术”是一门实践性很强的专业技术基础课程,学习过程中应充分重视实验环节,只有通过实验才能加深对理论的学习和理解,提高分析问题和解决问题的能力,为此我们编写了这本实验教材。

全书共分6章,第1~2章介绍汇编语言程序设计实验环境和步骤以及几种典型的汇编语言程序设计;第3章介绍微机接口实验平台 PITA 硬件资源及其使用方法;第4章为微机接口技术基础性实验,包括 8253、8259、8255、8237、8251、ADC0809 等接口芯片以及存储器拓展等方面的内容;第5章为微机接口技术综合性实验,其中温度闭环控制、直流电机速度控制以及定时采样监控系统综合性较强,几乎应用了前面大部分知识;第6章为研究创新性实验,以 51 单片机在工程实际中的应用为背景,根据实际提出系统设计要求,分析系统设计的方法,给出参考方案及其主要硬件设计原理图和部分软件设计源代码。设计内容既有简单的采样系统设计,又有典型的控制系统设计和集散控制系统设计。本书对微机原理学习和计算机控制系统实际工程设计有一定的参考价值。

在本书的编写过程中,吕淑平教授提出了许多宝贵意见,唐都科教仪器公司提供了大力支持和帮助,在此一并表示衷心感谢。由于编者水平有限,书中难免有错误和不妥之处,恳请读者指正。

编者

2016年7月

<b>第 1 章 汇编程序基础实验</b> .....	1
1.1 熟悉 DEBUG 调试程序 .....	1
1.2 算术运算程序设计 .....	7
1.3 数码转换 .....	9
1.4 显示程序.....	11
1.5 串操作程序设计.....	13
<b>第 2 章 汇编程序综合设计实验</b> .....	17
2.1 运算类应用程序设计实验.....	17
2.1.1 数字滤波程序设计 .....	17
2.1.2 PID 控制算法程序设计 .....	19
2.2 分支与循环程序设计实验.....	23
2.2.1 字符串大小写字母转换 .....	23
2.2.2 成绩统计 .....	26
2.3 数码转换实验.....	28
2.3.1 十六进制 ASCII 码转换为 BCD 码 .....	28
2.3.2 二进制数转换为 ASCII 码表示的十进制数 .....	31
2.4 子程序设计实验.....	34
2.4.1 找极值 .....	34
2.4.2 求平方根 .....	38
<b>第 3 章 微机接口实验系统介绍</b> .....	42
3.1 实验系统基本介绍.....	42
3.2 实验系统硬件资源.....	42

3.3	硬件实验操作方法	46
<b>第4章</b>	<b>微机接口基础实验</b>	<b>49</b>
4.1	8259 中断控制器实验	49
4.1.1	单一中断源	53
4.1.2	8259 优先级嵌套中断	55
4.1.3	8259 级连	58
4.2	8255 并行口实验	60
4.2.1	8255 方式0 应用	63
4.2.2	8255 方式1 应用	64
4.3	8253 定时/计数器实验	66
4.3.1	8253 计数器应用	69
4.3.2	8253 定时器应用	71
4.3.3	电子发声实验	75
4.4	存储器扩展实验	77
4.4.1	I/O 拓展存储器	79
4.4.2	总线扩展存储器	81
4.5	A/D 转换	83
4.6	8251 串行接口应用	87
4.7	DMA 应用	93
<b>第5章</b>	<b>接口技术综合实验</b>	<b>99</b>
5.1	人机交互接口	99
5.2	波形发生器	108
5.3	串行双机通信	112
5.4	步进电机控制	116
5.5	温度闭环控制	119
5.6	直流电机速度控制	130
5.7	定时采样监控系统	142
<b>第6章</b>	<b>研究创新性实验</b>	<b>149</b>
6.1	数字温度计设计	149
6.2	可燃气体检测仪设计	155
6.3	加油控制系统设计	164

---

6.4	直流电机闭环调速系统设计 .....	172
6.5	直流电机位置闭环控制系统设计 .....	180
6.6	温度集散控制系统设计 .....	186
6.7	基于 CAN 总线的集散监控系统设计 .....	200
<b>参考文献</b> .....		214
实验报告——汇编语言程序设计 .....		215
实验报告——8259A 中断控制器实验 .....		217
实验报告——8255A 并行接口实验 .....		219
实验报告——8253 定时/计数器实验 .....		221
实验报告——人机交互接口实验 .....		223
实验报告——接口综合设计实验 .....		225

# 汇编程序基础实验

## 1.1 熟悉 DEBUG 调试程序

### 1. 实验目的

- (1) 熟悉汇编语言编写的过程。
- (2) 熟悉 DEBUG 调试软件的命令及其格式。
- (3) 熟悉变量定义及其在内存中的分配情况。
- (4) 熟悉标志位的意义。

### 2. 实验设备

PC 一台。

### 3. 编写和调试汇编语言程序的过程

#### 1) 编写简单程序

##### (1) 进入编程环境。

在 Windows 环境,由程序→附件→命令提示符切换到 DOS。由于编程软件一般在 D 盘 TPC 目录下,故需在 DOS 提示符下输入命令切换到 D:\TPC\。具体操作如图 1.1 所示。

##### (2) 编辑源程序(EDIT)。

```
D:\TPC\ EDIT FIRST.ASM ↵
```

建立了源文件并进入编辑环境。注意:汇编源程序必须带 ASM 后缀。编辑程序完成后,按 Alt 键激活菜单,保存文件并退出编辑环境。编辑环境界面如图 1.2 所示。



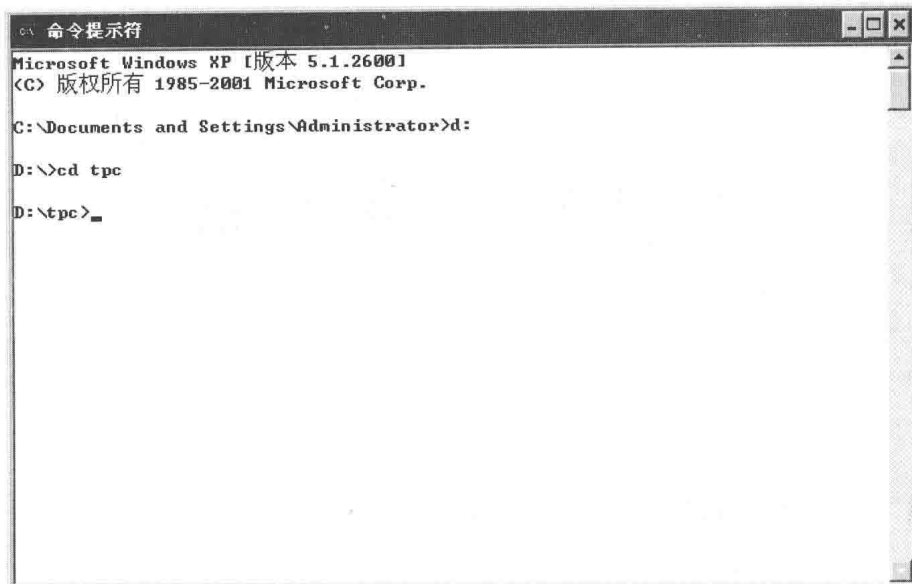


图 1.1 进入 TPC 目录过程

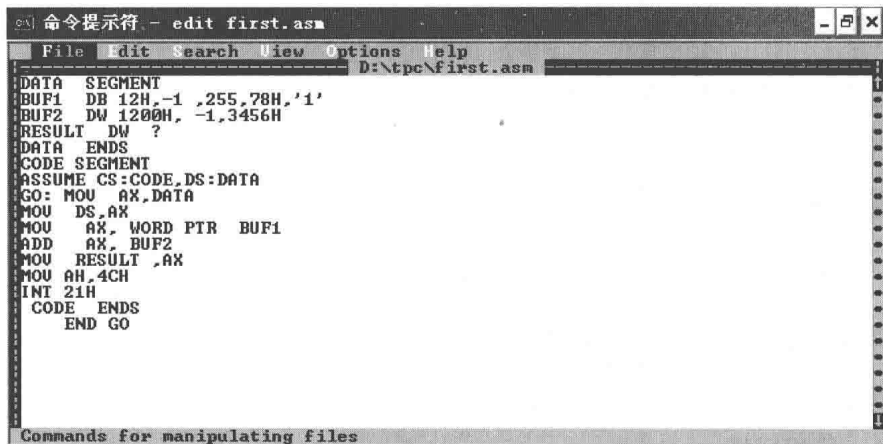
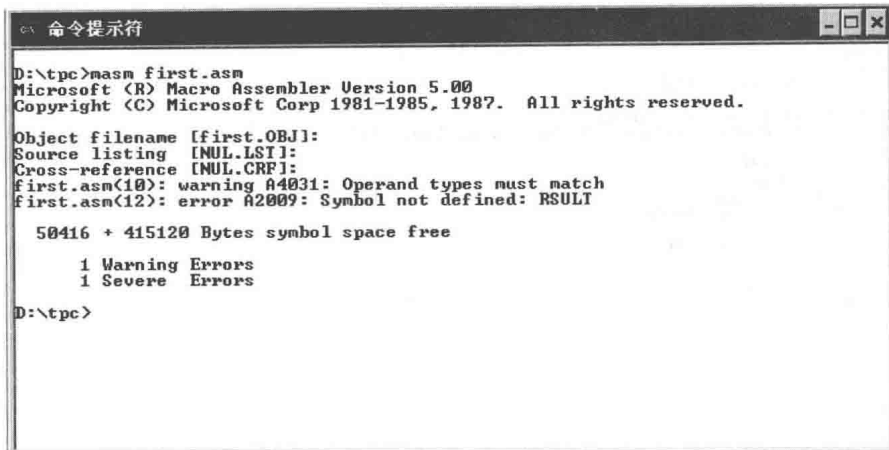


图 1.2 编辑源程序界面

### (3) 编译源程序(MASM)。

```
D:\TPC\MASM FIRST.ASM ↵
```

接下来的提示中不用输入任何命令,直接回车。生成的目标文件默认 FIRST.OBJ。如果编译器检查出语法错误,编译器会分别给出错误和警告所在的行数和原因,如图 1.3 所示。



```
命令提示符
D:\tpc>masm first.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [first.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:
first.asm(10): warning A4031: Operand types must match
first.asm(12): error A2009: Symbol not defined: RESULT

    50416 + 415120 Bytes symbol space free

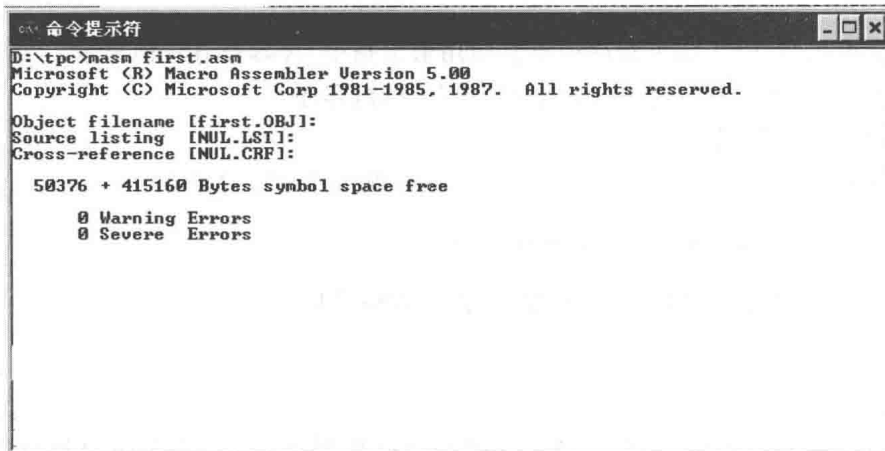
        1 Warning Errors
        1 Severe Errors

D:\tpc>
```

图 1.3 编译 FIRST.ASM 出错界面

重新进入源文件的编辑状态,根据编译器的提示对其进行修改。修改保存后,再重新对源文件编译。

编译结束没有错误和警告才可以进行链接操作。编译无错误界面如图 1.4 所示。



```
命令提示符
D:\tpc>masm first.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [first.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

    50376 + 415160 Bytes symbol space free

        0 Warning Errors
        0 Severe Errors
```

图 1.4 编译 FIRST.ASM 无错误界面

#### (4) 链接目标文件(LINK)。

```
D:\TPC\LINK FIRST.OBJ ↵
```

接下来的提示中不用输入任何命令,直接回车。由于连接器的原因,链接时会给出如图 1.5 所示的错误和警告,可以不用理睬。如果链接时有其他错误,则要检查错误并修改源程序,并重新编译、链接。

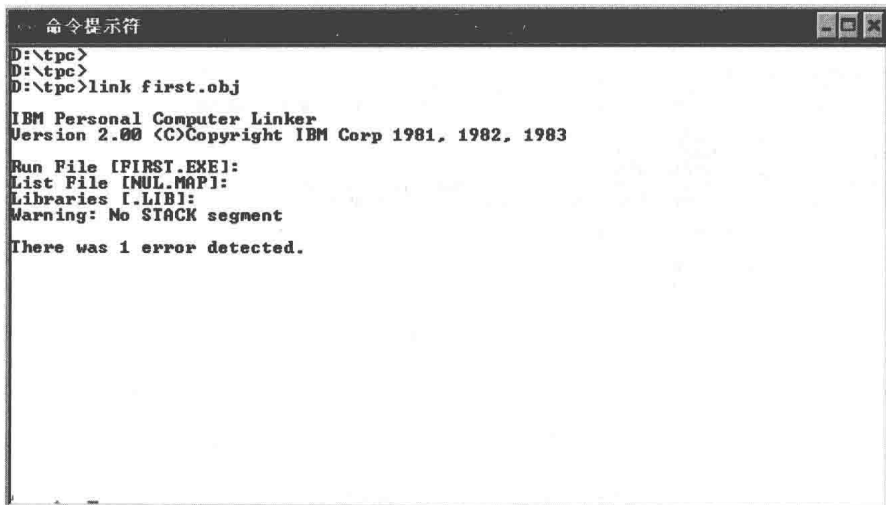


图 1.5 链接 FIRST.OBJ 界面

## (5) 运行程序。

```
D:\TPC\ FIRST.EXE ✓
```

运行生成的可执行文件,如果程序中没有使用输出指令,屏幕没有任何显示。为了观察程序运行结果和调试程序的需要,可使用 DEBUG 调试程序。

## 2) DEBUG 的使用

假设前面生成的可执行文件为 FIRST.EXE,则 DEBUG 命令如下:

```
D:\TPC\DEBUG FIRST.EXE ✓ ; 注意 EXE 后缀
```

DEBUG 中出现的数都是十六进制数,但都省略了后缀 H。

## (1) 反汇编。

格式: -U

执行一次命令只能反汇编 32 个字节。如果程序代码较短,则会把内存中其他代码反汇编,如图 1.6 所示,在 INT 21H 后都是随机代码。如果程序较长,则可多次执行反汇编命令。

反汇编后屏幕从左至右依次是:代码段地址(CS)、指令偏移地址(IP)、指令机器码、指令源代码。

## (2) 单步执行。

```
格式: -T      ; 执行 1 条指令
      -T 4    ; 执行 4 条指令
```

```

命令提示符 - DEBUG FIRST.EXE
D:\t\pc>
D:\t\pc>DEBUG FIRST.EXE
-U
116D:0000 B86C11      MOU    AX,116C
116D:0003 8ED8          MOU    DS,AX
116D:0005 A10000        MOU    AX,[0000]
116D:0008 03060500     ADD    AX,[0005]
116D:000C A30B00        MOU    [000B],AX
116D:000F B44C          MOU    AH,4C
116D:0011 CD21          INT    21
116D:0013 0000          ADD    [BX+SI],AL
116D:0015 0000          ADD    [BX+SI],AL
116D:0017 0000          ADD    [BX+SI],AL
116D:0019 0000          ADD    [BX+SI],AL
116D:001B 0000          ADD    [BX+SI],AL
116D:001D 0000          ADD    [BX+SI],AL
116D:001F 0000          ADD    [BX+SI],AL

```

图 1.6 使用 DEBUG 的反汇编命令界面

### (3) 断点运行。

格式：-G IP ; IP 为要执行到的断点的指令偏移地址

如图 1.6 所示,如果需要程序在指令 MOV AH,4CH 停下来,应先找到其对应 IP 为 000FH,然后输入 G 000F ↵,程序就会运行到指定指令处停下来。

### (4) 查看内存。

格式：-D 段寄存器名:0000 [n]

如察看数据段前 20 个单元的内容：

```
-D DS:0000 0013
```

也可以省掉察看的内存单元个数 n,显示系统默认的单元个数。如：

```
-D DS:0000
```

### (5) 修改内存。用给定的内容代替指定范围单元。

格式：-E 地址 内容表

如：-E DS:0000 01 F3 '1',将用 01H、0F3H、31H 代替数据段中 0000~0002 原有的数。

### (6) 察看寄存器。

格式：-R

### (7) 退出 DEBUG 调试。

格式：-Q

## (8) 标志位显示和状态对照表。

	1	0
CF	CY	NC
ZF	ZR	NZ
SF	NG	PL
AF	AC	NA
OF	OV	NV
PF	PE	PO
DF	DN	UP
IF	EI	DI

**4. 实验内容**

编写能涉及变量操作和标志位变化的汇编程序。

**5. 实验步骤**

- (1) 编辑、编译、链接程序。
- (2) 反汇编后对照观察程序的机器码和源代码。
- (3) 观察各逻辑段的分配情况。
- (4) 观察变量在程序执行前后在内存的变化情况。注意：查看变量初值时，要给对应段寄存器赋值后，变量才真正建立。
- (5) 执行程序观察标志位的变化。

**6. 实验报告要求**

- (1) 整理程序清单，并加以适当注释。
- (2) 写出程序运行结果，给出内存前后的变化。
- (3) 写出用 DEBUG 查看内存的方法。
- (4) 写出用 DEBUG 单步和断点调试程序的命令。

**7. 参考程序**

参考程序如下：

```

DATA    SEGMENT
BUF1    DB  12H, -1, 255, 78H, '1'
CNT     EQU  $ - BUF1           ; EQU 对应的常量不占用内存
BUF2    DW  1200H, -1, 3456H
RESULT  DW  ?
DATA    ENDS

```

```
CODE    SEGMENT
        ASSUME  CS: CODE, DS: DATA
GO:     MOV  AX, DATA
        MOV  DS, AX
        MOV  AX, WORD PTR  BUF1
        ADD  AX, BUF2
        MOV  RESULT, AX
        MOV  AH, 4CH
        INT  21H
CODE    ENDS
        END  GO
```

## 1.2 算术运算程序设计

### 1. 实验目的

- 1) 掌握算术运算程序的设计方法。
- 2) 掌握符号数和无符号数算术运算的区别。

### 2. 实验设备

PC 一台。

### 3. 实验内容

#### 1) 知识提要

(1) 加、减法指令：符号和无符号数加减法使用相同指令，加法指令为 ADD 和 ADC；减法指令为 SUB 和 SBB。

乘、除法指令：符号数为 IMUL, IDIV；无符号数为 MUL, DIV。

(2) 混合运算时注意数据位长的扩展，符号数扩展可采用指令 CBW 或 CWD，拓展字节数据必须存放在 AL 中，拓展字数据必须存放在 AX 中。

**注：**无符号数扩展时在高字节或高字直接补零，不能采用指令 CBW 或 CWD。

#### 2) 实验要求

编程完成单字节符号数混合运算式  $(X + A * B - Y) / C$ ，其中 X、A、B、Y、C 为字节变量。

### 4. 实验步骤

- (1) 画出程序流程图。

- (2) 编写源程序、编译和链接。
- (3) 用 DEBUG 调试程序,观察结果。
- (4) 修改几组数据反复验证。

## 5. 实验报告要求

- (1) 整理程序清单,并加以适当注释。
- (2) 写出程序运算数据和结果。
- (3) 给出无符号字节数据的混合运算式 $(X+A * B-Y)/C$ 的程序。

## 6. 参考程序及流程

算法运算流程图如图 1.7 所示。编程时扩展指令的使用注意中间结果的暂存。  
参考程序如下:

```

DATA    SEGMENT
X        DB  7FH
Y        DB  7EH
A        DB  80H
B        DB  0FFH
C        DB  08H
RESULT  DB  2  DUP(?)
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE , DS:DATA
START:  MOV  AX , DATA
        MOV  DS, AX
        MOV  AL, A
        IMUL B           ; AX = A * B
        MOV  BX, AX      ; 暂存 A * B
        MOV  AL, X
        CBW              ; 扩展 X
        ADD  BX , AX
        MOV  AL, Y
        CBW
        SUB  BX, AX
        MOV  AX, BX
        DIV  BYTE PTR C  ; 商
        MOV  RESULT+1 , AH ; 余数
        MOV  AH, 4CH      ; 返回 DOS 操作系统
        INT  21H
CODE    ENDS
        END  START

```

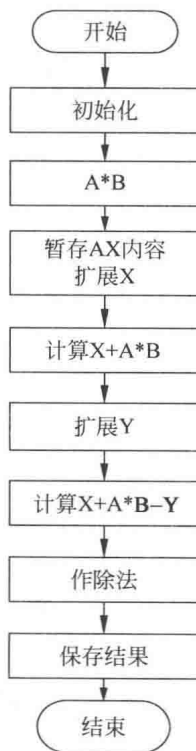


图 1.7 算术运算流程图

## 1.3 数码转换

### 1. 实验目的

- (1) 掌握不同进制及编码相互转换的程序设计方法。
- (2) 进一步掌握调试程序的方法。

### 2. 实验设备

PC 一台。

### 3. 实验内容

#### 1) 知识提要

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 处理信息一般采用二进制数进行计算和处理, 处理后输出给外设的结果又必须按外设的要求变为 ASCII 码、BCD 码或七段显示码等, 因此在应用软件中各类数制和编码的转换是必不可少的。

#### 2) 实验要求

将存放在 BUF 中的用 5 个 ASCII 码表示的十进制数转换成二进制数存放到 RESULT 中。

编程提示: 先将 ASCII 码转换成一位十进制数, 多位十进制数又可表示为

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

式中,  $D_i$  代表转换成的十进制数。式(1-1)可转换为

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

由式(1-2)可归纳出将十进制数转换为二进制数的方法: 从十进制数的最高位  $D_n$  开始作乘 10 加次位数的操作, 将结果再乘 10 再加下一个次位数, 如此重复, 即可求出二进制数的结果。

### 4. 实验步骤

- (1) 画出程序流程图。



- (2) 编写源程序、编译和链接。
- (3) 用 DEBUG 调试程序,观察结果。
- (4) 修改几组数据反复验证。

## 5. 实验报告要求

- (1) 整理程序清单,并加以适当注释。
- (2) 写出程序运算数据和结果。

## 6. 参考程序

参考程序如下:

```

DDATA SEGMENT
BUF    DB 33H, 34H, 32H, 35H, 36H      ; 表示十进制数: 34256
RESULT DW ?
DDATA ENDS

CODE SEGMENT
    ASSUME CS: CODE, DS: DDATA
START: MOV AX, DDATA
        MOV DS, AX
        MOV SI, OFFSET BUF
        MOV BX, 10
        MOV CH, 04H
        MOV AH, 00H
        MOV AL, [SI]
        SUB AL, 30H                      ; 将 ASCII 表示的最高位转换为十进制数
        INC SI                          ; 指向下一个数
LOP:    IMUL BX                          ; 乘 10, 采用字相乘
        ADD AL, [SI]                    ; 加次高位
        ADC AH, 0                       ; 乘积有效结果在 AX 中
        SUB AL, 30H
        SBB AH, 0
        INC SI
        DEC CH
        JNZ LOP
        MOV RESULT, AX
        MOV AH, 4CH
        INT 21H
CODE ENDS
    END START

```