

# TDN86/88 实验指导书

阎有运 郭顺京 编

焦作工学院电气工程系

2001 年 3 月

# 目录

|      |                        |      |
|------|------------------------|------|
| 实验一  | 系统认识实验·····            | (1)  |
| 实验二  | 数码转换编程及程序调试·····       | (5)  |
| 1.   | ASCII 码→二进制码           |      |
| 2.   | ASCII 码→BCD 码          |      |
| 3.   | 二进制→ASCII 码            |      |
| 4.   | 十六进制数→ASCII 码          |      |
| 5.   | BCD 码→二进制              |      |
| 实验三  | 运算类编程实验·····           | (12) |
| 1.   | 减运算                    |      |
| 2.   | 法运算                    |      |
| 3.   | 开平方运算                  |      |
| 实验四  | 分支程序设计实验·····          | (18) |
| 实验五  | 循环程序设计·····            | (21) |
| 实验六  | 排序程序设计·····            | (23) |
| 1.   | 气泡法排序程序设计              |      |
| 2.   | 成绩名次表                  |      |
| 实验七  | 子程序设计实验·····           | (27) |
| 1.   | 求最大值和最小值               |      |
| 2.   | 求 N!                   |      |
| 实验八  | 显示程序实验·····            | (31) |
| 实验九  | 中断特性及 8259 应用编程实验····· | (33) |
| 1.   | 一中断源实验                 |      |
| 2.   | 优先级中断实验                |      |
| 3.   | 级连方式实验                 |      |
| 实验十  | 8255 并行接口应用实验·····     | (45) |
| 1.   | 双口方式 0 实验              |      |
| 2.   | 方式 1 实验                |      |
| 实验十一 | 8253 定时/计数器应用实验·····   | (50) |
| 1.   | 单通道方式 0 实验             |      |
| 2.   | 单通道定时中断实验              |      |
| 3.   | 电子发声实验                 |      |
| 实验十二 | DMA 特性及 8237 应用实验····· | (58) |

|      |                     |       |
|------|---------------------|-------|
| 实验十三 | 8251 串行接口应用实验·····  | (65)  |
| 实验十四 | 串行通讯应用实验·····       | (71)  |
| 1.   | 双机通讯实验              |       |
| 2.   | 串口自发自收实验            |       |
| 实验十五 | A/D 转换实验·····       | (78)  |
| 实验十六 | D/A 转换实验·····       | (82)  |
| 实验十七 | 存储器扩展实验·····        | (86)  |
| 1.   | 存储器扩展实验 (1)         |       |
| 2.   | 存储器扩展实验 (2)         |       |
| 实验十八 | 8255 键盘及显示接口实验····· | (91)  |
| 实验十九 | 步进电机实验·····         | (95)  |
| 实验二十 | 计算机控制综和应用实验·····    | (98)  |
| (一)  | 直流电机闭环调速实验·····     | (98)  |
| (二)  | 电热箱闭环控制实验·····      | (110) |
| 附录一  | TDN86/88 系统概述       |       |
| 附录二  | TDN86/88 系统的配制与安装   |       |
| 附录三  | TDDN86/88 系统操作硬件环境  |       |
| 附录四  | TDN86/88 实验程序清单     |       |
| 附录五  | 系统编程信息              |       |

## 实验一 系统认识实验

### 一、实验目的

掌握 TND86/88 教学系统的基本操作。

### 二、实验设备

TND86/88 教学实验系统。

### 三、实验内容及步骤

从 3500H 内存单元开始建立 0—15 共 16 个数据。

实验程序及流程如下：

```
STACK SEGMENT STACK           ; 8088 宏汇编程序
    DW 64 DUP(?)               ; 定义堆栈段
STACK ENDS
CODE SEGMENT                   ; 定义代码段
    ASSUME CS: CODE
START: MOV DI, 3500H           ; 程序开始设数据首址
      MOV CX, 0010H           ; 字节数送入 CX 中
      MOV AX, 0000H
      SAHF
A1:    MOV [DI], AL             ; 写入一字节
      INC DI                   ; 修改地址指针
      INC AX                   ; 修改数据
      DAA                      ; 十进制调整
      LOOP A1                  ; 未填完转移
A2:    JMP A2
CODE ENDS                      ; 代码段结束
      END START                ; 程序结束
```

### 实验步骤

- 1、使用串行通信电缆将实验系统与 PC 微机相连。
- 2、打开 WMD86 进入系统环境如图 1-1 所示。

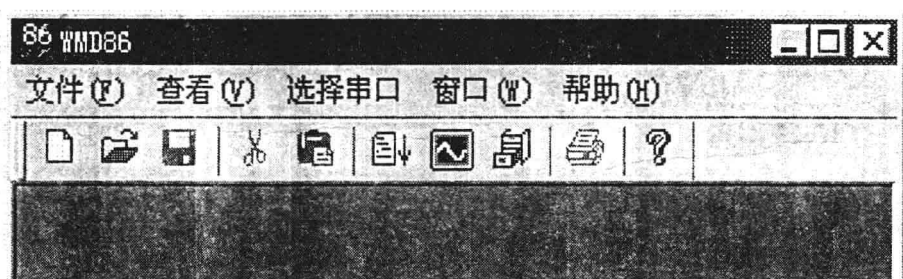


图 1-1

- 3、点击‘选择串口’如图 1-2 所示。可选串行口 1 或串行口 2（计算机后面与 UBS 接口相邻的为串行口 1）。

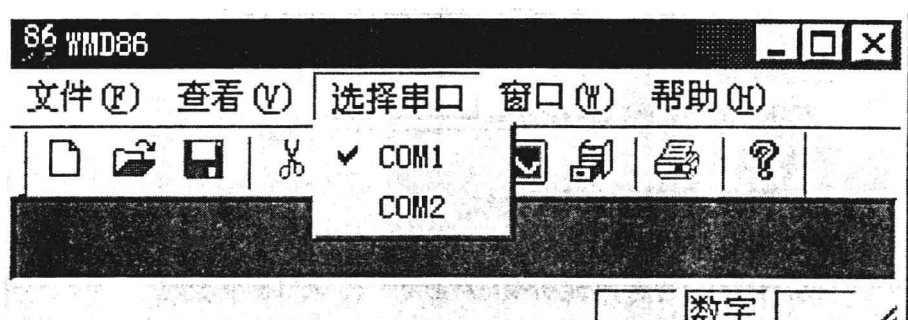


图 1-2

4、点击文件，打开新建源程序，如图 1-3 所示。

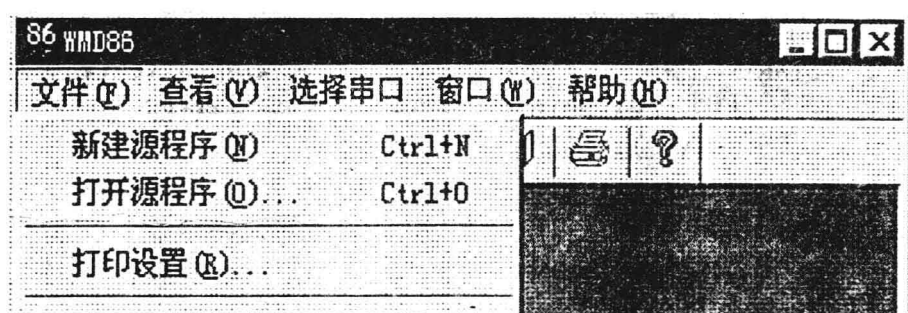


图 1-3

5、输入源程序并保存为 DQ.asm. 如图 1-4 所示。

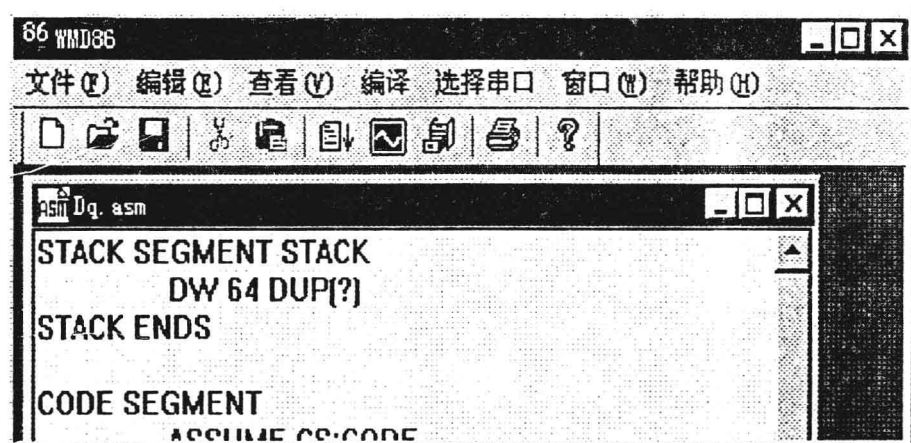


图 1-4

6、点击‘编译’，对源程序进行汇编、连接如图 1-5 所示

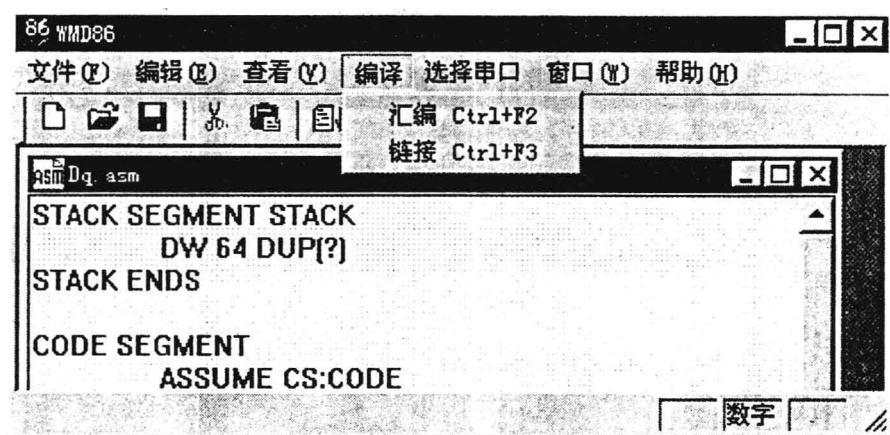


图 1-5

7、汇编、连接无误后点击‘窗口’如图 1-6 所示。

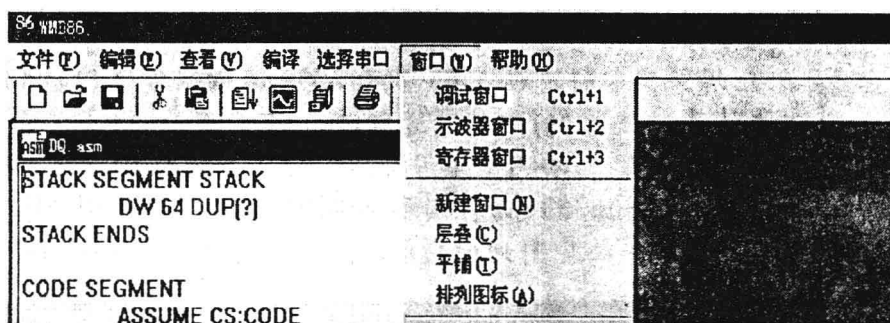


图 1-6

8、点击‘调试窗口’如图 1-7 所示

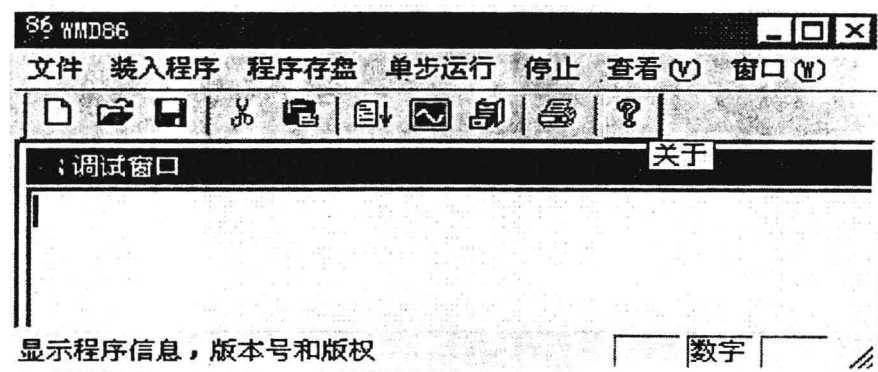


图 1-7

9 按系统中的 RESET 键出现 ‘welcome to you’界面如图 1-8 所示。

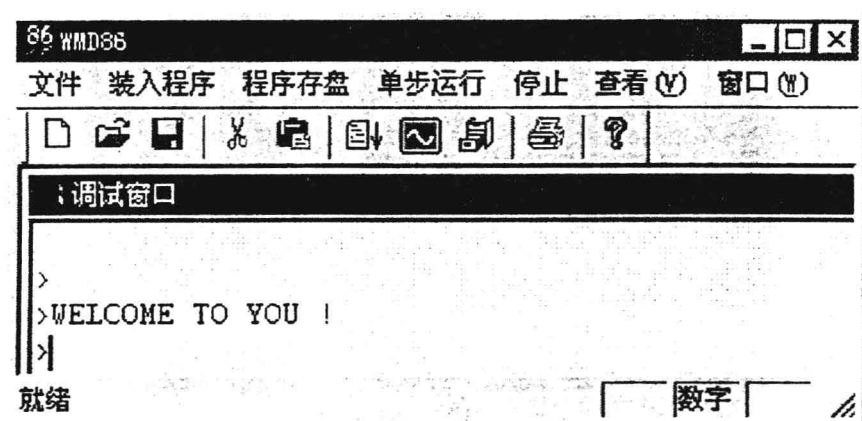
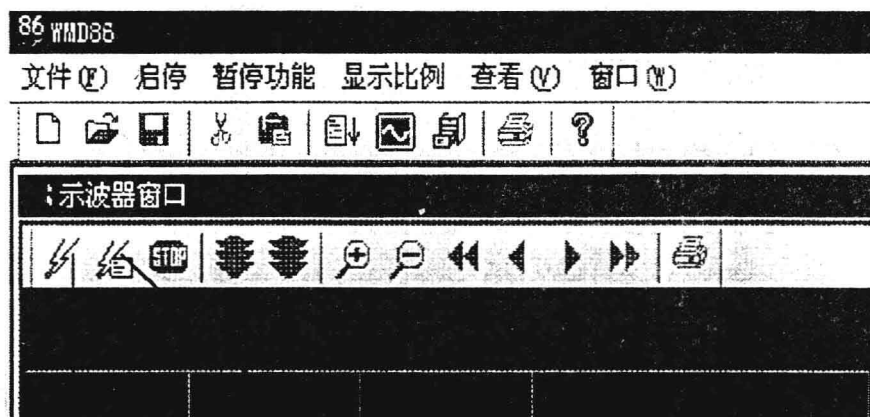


图 1-8

- 10、装入 DQ.EXE 程序，确认程序段地址 CS: 0000 及偏移量地址 IP: 2000.
- 11、提示装载完毕后使用 U0000: 2000 命令进行反汇编，检查程序是否正确装入。
- 12、在 ">" 提示符下键入 G=0000: 2000 (在系统默认段值 CS=0000 情况下也可直接键入 G=2000) 可连续运行程序。
- 13、使用 D0000: 3500 可查看 3500H-350FH 单元中的内容是否为 0-15 共 16 个数。可用 E3500 修改内存单元的内容。
- 14、在以后的实验中将会用到示波器，示波器窗口如图 1-9 所示。



启动不运行程序

启动并运行程序

图 1-9

## 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. 熟悉键盘使用方法。
3. 熟悉调试程序的方法。

### 二. 实验设备

TDN86/51 或 TDN86/88 教学实验系统一台

### 三. 实验内容及步骤

计算机输入设备输入的信息一般是由 ASCII 码或 BCD 码表示的数据或字符, CPU 一般均用二进制数进行计算或其他信息处理, 处理结果的输出又必须依照外设的要求变为 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 2-1 所示, 数码对应关系如表 2-1 所示。

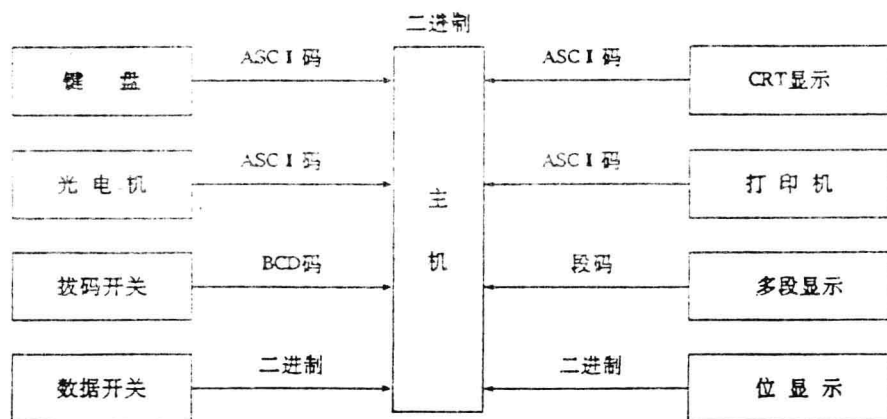


图 2-1 数码转换关系

表 2-1 数码对应关系

| 十六进制数 | BCD 码 | 二进制机器码 | ASC II 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 |

## 1. 将 ASC II 码表示的十进制数转换为二进制数

十进制表示为:

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

 $D_i$  代表十进制数 1, 2, 3, ..., 9, 0;

上式转换为:

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

由式(2)可归纳十进制数转换为二进制的方法：从十进制数的最高位  $D_n$  开始作乘 10 加次位的操作，依次类推，则可求出二进制数结果。程序流程及参考程序如下：(规定：被转换的 ASCII 码十进制数存放在 3500H-3504H 单元中，转换结果存在于 3510-3511H 单元中。)

```
STACK SEGMENT STACK
```

```
    DW 64 DUP(?)
```

```
STACK ENDS
```

```
CODE SEGMENT
```

```
    ASSUME CS:CODE
```

```
START:  MOV SI, 3500H
```

```
        MOV DI, 3510H
```

```
        MOV BX, 000AH
```

```
        MOV CX, 0004H
```

```
        MOV AH, 00H
```

```
        MOV AL, [SI]
```

```
        SUB AL, 30H
```

```
A1:     IMUL BX
```

```
        ADD AL, [SI-01]
```

```
        SUB AL, 30H
```

```
        INC SI
```

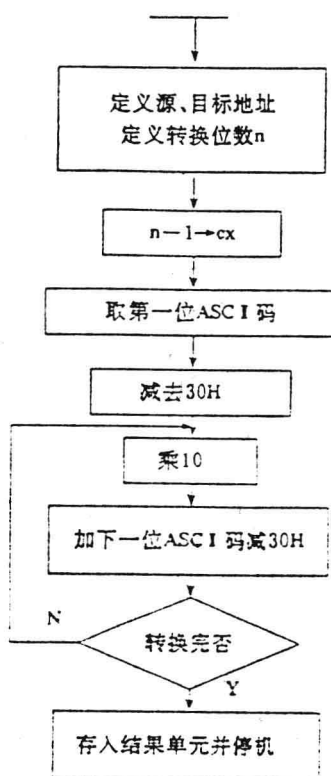
```
        LOOP A1
```

```
        MOV [DI], AX
```

```
A2:     JMP A2
```

```
CODE ENDS
```

```
END START
```



## 实验步骤

- (1) 输入程序并检查无误，经汇编、连接后装入系统。
- (2) 在 3500-3504H 单元存入十进制数 12 的 ASCII 码，即 E3500↙，并输入 30 30 30 31 32。
- (3) G=0000: 2000↙，运行程序，并用 CTRL+C 来中断程序，返回监控状态。
- (4) 用 D3510↙ 查看结果，应为：3510 0C 00。
- (5) 反复试几组数，考察程序的正确性。

## 2. 将十进制数的 ASCII 码转换为 BCD 码

设从键盘输入的五位十进制数的 ASCII 码已存放在 3500H 起始的内存单元内，把它转换为 BCD 码后，再按位分别存入 350AH 起始的内存单元内。若输入的不是十进制数的 ASCII 码，则对应存放结果的单元内容为“FF”。由表 2 可知，一字节 ASCII 码取其低四位即变为 BCD 码。实验程序及流程如下：

```

STACK SEGMENT STACK
    DW 64 DUP(?)
STACK ENDS
CODE SEGMENT
    ASSUME CS:CODE
START:  MOV CX, 0005H    ; 循环计数器赋初值
        MOV DI, 3500H   ; ASC II 码首址
A1:     MOV BL, 0FFH     ; 将错误标志送入 BL
        MOV AL, [DI]    ; 送 ASC II 码至 AL
        CMP AL, 3AH     ; 比较 AL 与 3A
        JNB A2          ; 不低于 3A 则转 A2
        SUB AL, 30H     ; 低于 3A 则取 ASC II 码低 4 位
        JB A2           ; 低于 30 则转 A2
        MOV BL, AL      ; 否则 AL 内容送入 BL, 取代 FF
A2:     MOV AL, BL       ; 结果或错误标志送入 AL
        MOV [DI-0AH], AL
        INC DI
        LOOP A1
A3:     JMP A3

CODE ENDS
END START

```

### 实验步骤

- (1) 输入程序并检查无误，经汇编、连接后装入系统。
- (2) 在 3500-3504H 单元中存放五位十进制数的 ASC II 码。  
即：E3500↙，并输入 31、32、33、34、35。
- (3) G=0000: 2000↙，运行程序。
- (4) D350A↙，显示结果为：  
0000: 350A 01 02 03 04 05 CC ...
- (5) 反复试几组数，考察程序的正确性。

### 3. 将十六位二进制数转换为 ASC II 码表示的十进制数

十六位二进制数的值域为 0-65535，最大可转换为五位十进制数。

算法：五位十进制数可表示为：

$$N_D = D_4 \times 10^4 + D_3 \times 10^3 + D_2 \times 10^2 + D_1 \times 10 + D_0$$

$D_i$ ：表示十进制数 0~9

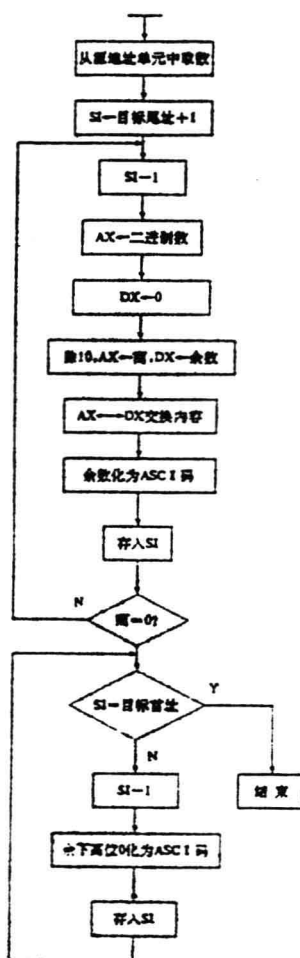
因此，将十六位二进制数转换为五位 ASC II 码表示的十进制数，就是求  $D_1$ - $D_4$ ，并将它们转化为 ASC II 码。实验程序及流程如下：（设源数存于 3500-3504H 单元中，结果数存于 3510-3514H 单元中）

```

STACK SEGMENT STACK
    DW 64 DUP(?)
STACK ENDS

CODE SEGMENT
    ASSUME CS:CODE
START:  MOV SI, 3500H
        MOV DX, [SI]
        MOV SI, 3515H
A1:     DEC SI
        MOV AX, DX
        MOV DX, 0000H
        MOV CX, 000AH
        DIV CX
        XCHG AX, DX
        ADD AL, 30H
        MOV [SI], AL
        CMP DX, 00C0H
        JNE A1
A2:     CMP SI, 3510H
        JZ A3
        DEC SI
        MOV AL, 30H
        MOV [SI], AL
        JMP A2
A3:     JMP A3
CODE ENDS
END START

```



## 实验步骤

- (1) 输入程序并检查无误，经汇编、连接后装入系统。
- (2) 在 3500-3501H 单元中存放 0C00，运行程序并检查结果，应看到 3510H-3514H 单元中的数依次为 30 30 30 31 32。
- (3) 反复试几组数，并运行程序，观察结果。

## 4. 十六进制数转换为 ASCII 码

设经过 CPU 处理后的十六进制数存放于起始地址为 3500H 的内存单元中，把它们转换成 ASCII 码后，再分别存入起始地址为 350AH 的内存单元中。从表 2-1 中可知十六进制数加 30H 即可得到 0H-9H 的 ASCII 码，而要得到 AH-FH 的 ASCII 码，则需再加 7H。实验程序及流程如下：

STACK SEGMENT STACK

DW 64 DUP(?)

STACK ENDS

CODE SEGMENT

ASSUME CS:CODE

START: MOV CX, 0004H

MOV DI, 3500H

MOV DX, [DI]

A1: MOV AX, DX

AND AX, 000FH ; 取低四位值

CMP AL, 0AH ; 判断是否<9

JB A2 ; 若在 0-9 范围内, 则转 A2

ADD AL, 07H ; 若为 "A-F", 加 7

A2: ADD AL, 30H ; 转换为 ASCII 码

MOV [DI-0AH], AL

INC DI

PUSH CX ; 保护循环计数器内容

MOV CL, 04H ; 移位次数送至 CL

SHR DX, CL

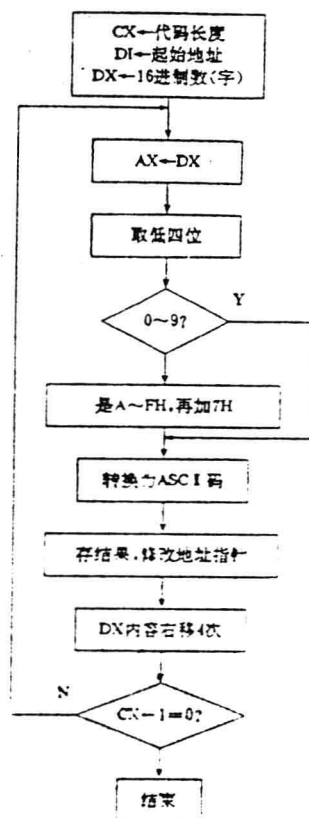
POP CX

LOOP A1

A3: JMP A3

CODE ENDS

END START



## 实验步骤

- (1) 输入程序并检查无误, 经汇编、连接后装入系统。
- (2) 在 3500-3501H 单元中存入四位十六进制数 203B, 即: E3500✓, 并输入 3B、20。
- (3) G=0000: 2000✓, 运行以上程序。
- (4) D350A✓, 显示结果为: 0000: 350A 42 33 30 32 CC ...  
输入数与结果 ASCII 码对应顺序相反。
- (5) 反复试几组数, 考察程序的正确性。

## 5. BCD 码转换为二进制码

设四个二位十进制数的 BCD 码存放在起址为 3500H 的单元中, 转换出的二进制数码存入起址为 3510H 的内存单元中, 实验程序及流程图如下:

```
STACK SEGMENT,STACK
```

```
DW 64 DUP(?)
```

```
STACK ENDS
```

```
CODE SEGMENT
```

```
ASSUME CS:CODE
```

```
START: MOV CX,0004H
```

```
MOV DI,3500H
```

```
A1: MOV AL,[DI]
```

```
ADD AL,AL ; 乘 2
```

```
MOV BL,AL
```

```
ADD AL,AL ; 乘 2
```

```
ADD AL,AL ; 乘 2
```

```
ADD AL,BL ; 乘 10
```

```
INC DI
```

```
MOV AH,00H
```

```
ADD AL,[DI] ; ECD 码十位与个位加
```

```
MOV [DI-0FH],AX ; 存结果
```

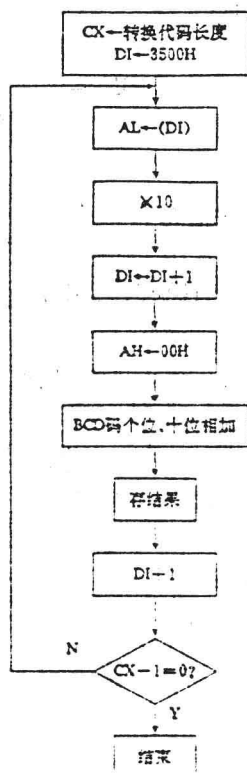
```
INC DI
```

```
LOOP A1
```

```
A2: JMP A2
```

```
CODE ENDS
```

```
END START
```



## 实验步骤

(1) 输入程序并检查无误，经汇编、连接后装入系统。

(1) 在 3500-3507 单元中存入四个十进制数的 ECD 码，即：

E3500 ✓，输入 01、02、03、04、05、06、07、08。

(2) G=0000:2000 ✓，运行以上程序。

(3) D3510 ✓，显示结果为 3510 0C 00 22 00 38 00 4E 00。

(4) 反复试几组数，考察程序的正确性。

## 四. 思考题

1. 程序 2 将一个五位十进制数转换为二进制数（十六位）时，这个十进制数最小可为多少，最大又可为多少？为什么？

2. 将一个十六位二进制数转换为 ASCII 码十进制数时，如何确定 DI 的值？

3. 在十六进制数转换为 ASCII 码时，存转换结果后，为什么要把 DX 向右移四次？

4. 自编 ASCII 码转换十六进制、十六进制小数转换二进制、二进制转换 BCD 码的程序，并调试运行。

## 实验三 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

TDN86/51 或 TDN86/88 教学实验系统一台

### 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，可对表 3-1 所示的数据类型进行算术运算。

| 数制  | 二进制      |     | BCD 码    |            |
|-----|----------|-----|----------|------------|
|     | 带符号      | 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)   |

表 3-1 数据类型算术运算表

#### 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。实验程序如下：

```
STACK SEGMENT STACK
```

```
    DW 64 DUP(?)
```

```
STACK ENDS
```

```
DATA SEGMENT
```

```
XL  DW ?           ;请在此处给 X 低位赋值
```

```
XH  DW ?           ;请在此处给 X 高位赋值
```

```
YL  DW ?           ;请在此处给 Y 低位赋值
```

```
YH  DW ?           ;请在此处给 Y 高位赋值
```

地址      存储器

|      |    |   |   |
|------|----|---|---|
| 3500 | AD | } | X |
| 3501 | 65 |   |   |
| 3502 | 15 |   |   |
| 3503 | 00 |   |   |
| 3504 | 9E | } | Y |
| 3505 | 87 |   |   |
| 3506 | 21 |   |   |
| 3507 | 00 |   |   |
| 3508 |    | } | Z |
| 3509 |    |   |   |
| 350A |    |   |   |
| 350B |    |   |   |

```

ZL DW ?
ZH DW ?
DATAENDS
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA
START: MOV AX, DATA
        MOV DS, AX
        MOV AX, XL      : X 低位送 AX
        ADD AX, YL      : X 低位加 Y 低位
        MOV ZL, AX      : 存低位和
        MOV AX, XH      : X 高位送 AX
        ADC AX, YH      : X 高位加 Y 高位
        MOV ZH, AX
A1:     JMP A1
CODE ENDS
END START

```

本实验程序是双精度（2 个 16 位，既 32 位）运算，利用累加器 AX，先求低十六位和，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需用 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) 输入程序并检查无误，经汇编、连接后装入系统。（设:CS:0000H, IP:2000H）
- (2) 用 U0000:2000 查看 MOV AX,XXXX(DATA)语句，即得到数据段地址 CS:XXXX 用 E 命令 EXXXX:0000 给 XL, XH, YL, YH 赋值存入二进制数 A0 65 15 00 和 9E B7 21 00。
- (3) G=0000: 2000，运行以上程序。
- (4) DXXXX:0008，显示计算结果： 3E 1D 37 00 CC …。
- (5) 反复试几组数，考察程序的正确性。

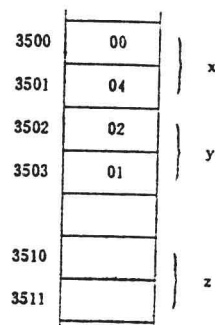
## 2. 十进制数的 BCD 码减法运算

计算  $X - Y = Z$ ，其中，X、Y、Z 为 BCD 码。实验程序及流程如下： 地址 存贮器

```

STACK SEGMENT STACK
    DW 64 DUP(?)
STACK ENDS
DATA SEGMENT
X   DW ?      :请在此处给 X 赋值
Y   DW ?      :请在此处给 Y 赋值
Z   DW ?

```

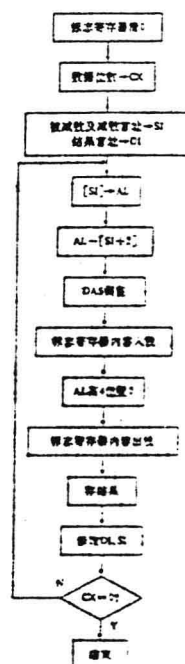


```

DATAENDS
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA
START:  MOV AX, DATA
        MOV DS, AX
        MOV AH, 00H
        SAHF
        MOV CX, 0002H
        MOV SI, OFFSET X
        MOV DI, OFFSET Z
A1:     MOV AL, [SI]
        SBB AL, [SI+02H]
        DAS
        PUSHF
        AND AL, 0FH
        POPF
        MOV [DI], AL
        INC DI
        INC SI
        LOOP A1
A2:     JMP A2

CODE ENDS
END START

```



### 实验步骤

- (1) 输入实验程序并检查无误，经汇编、连接后装入系统。(设:CS:0000H, IP:2000H)
- (2) 用 U0000:2000 查看 MOV AX, XXXX(DATA) 语句，即得到数据段段地址 CS:XXXX。用 E 命令 EXXXX:0000/ 给 X,Y 赋值存入 40 和 12 的 BCD 码: 00 04 02 01。
- (3) G=0000: 2000/，运行以上程序。
- (4) DXXXX:0004/，显示计算结果: 08 02 CC ...。
- (5) 反复试几组数，考察程序的正确性。

### 3. 乘法运算

本实验实现十进制数的乘法，被乘数和乘数均以 BCD 码形式存放于内存中，乘积在屏幕上显示，实验程序及流程如下：

```

STACK SEGMENT STACK
        DW 64 DUP(?)
STACK ENDS

```