



21世纪普通高等院校规划教材 • 信息技术类

WEIXING JISUANJI YUANLI JI YINGYONG
SHIYAN ZHIDAO

微型计算机 原理及应用实验指导

主编 岐世峰



西南交通大学出版社
[Http://press.swjtu.edu.cn](http://press.swjtu.edu.cn)

微型计算机原理及应用实验指导

主 编 岐世峰

副主编 秦振涛

西南交通大学出版社

· 成 都 ·

前 言

本书是西南交通大学出版社出版的《微型计算机原理及应用》的配套教学辅助教材，适用于高等院校工科电气类、机电类和应用型计算机本科各专业的教学。本书是一门专业基础课教材，目的在于让学生理论与实际结合，理解与掌握微型计算机的基本组成、工作原理、各类接口部件的功能，以及如何与系统连接构建微机系统等方面的知识，使学生具有微机应用系统软硬件开发的初步能力，全书共分 2 部分。

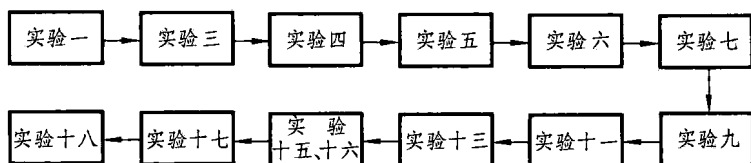
本书第一部分为实验指导书，其中包含 20 个实验，基本涵盖了《微型计算机原理及应用》一书的主要内容。结合每章教学内容，安排规划上机实践任务；对程序的开发、调试方法进行详细的介绍，使读者完全可以根据书中所述独立完成各个程序。在授课过程中，教师可以根据实际情况，指导学生完成部分验证性实验，然后留出一部分时间，让学生参照实验教材，完成教材中列出的其他实验项目，这部分实验的目的是给学生一定的空间，培养学生自主学习、独立开发的能力。

本书的第二部分为课程设计指导书，其中包含 10 个项目，主要目的是了解并掌握微型计算机的工作原理，具备初步的独立设计能力，掌握微型计算机的基本组成和结构，运用已学过的知识来理解和把握新型计算机的工作原理。

作为教材，使用者可以根据自己的需求，选取相应的内容进行教学。有 3 种建议方案：

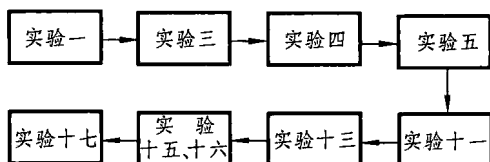
方案一：

选取 12 个实验项目，根据具体要求，可以安排 20~24 个学时。



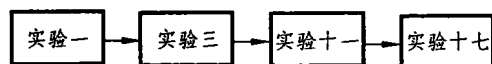
方案二：

选取 8 个实验项目，根据具体要求，可以安排 12~16 个学时。



方案三：

如果要学会汇编语言的调试、微型计算机的基本工作原理、各类接口部件的功能，可选用此方案，安排 8 个学时左右。



本书由岐世峰担任主编，秦振涛任副主编，实验 1～实验 19 由秦振涛编写，实验 20、课程设计和附录由岐世峰编写。全书由岐世峰负责审核、统稿。

由于编者水平有限，加之时间仓促，书中不足之处难免，恳请广大读者批评指正。

编 者

2009 年 7 月

目 录

第一部分 实验指导	1
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
实验三 运算类编程实验	12
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
实验七 子程序设计实验	27
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
实验十 8255 并行接口应用实验	43
实验十一 8253 定时/计数器应用实验	48
实验十二 DMA 特性及 8237 应用实验	56
实验十三 8251 串行接口应用实验	63
实验十四 串行通讯应用实验	69
(一) 双机通讯实验	69
(二) 串口自发自收实验	73
实验十五 A/D 转换实验	76
实验十六 D/A 转换实验	80
实验十七 存储器扩展实验	84
实验十八 8255 键盘及显示接口实验	89
实验十九 步进电机实验	93
实验二十 计算机控制综合应用实验	96
(一) 直流电机闭环调速实验	96
(二) 电热箱闭环控制实验	108
第二部分 课程设计指导	121
附录一 系统集成软件使用说明	124
附录二 系统编程信息	134
附录三 课程设计参考程序	137
参考文献	150

第一部分

实验指导

实验一 系统认识实验

一、实验目的

掌握 TDN86/51 或 TDN86/88 教学系统的基本操作。

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

1. 系统操作练习 (1)

1) 程序的输入与修改

从 3500H 内存单元开始建立 0~15 共 16 个数据。

(1) 使用串行通讯电缆将实验系统与 PC 机相连。

(2) 将位于线路板右下角的系统状态选择开关拨至 86 挡 (此操作仅针对 86/51 系统), 开启实验系统。

(3) 在系统软件所在目录下键入 MD86 $\swarrow$, 选择对应串口号, 进入集成操作软件环境, 出现系统提示符 ">". 按下 F1 功能键, 进入全屏幕编辑界面, 按 ALT+F 建立新文件 (NEW), 即可开始输入源程序。实验程序及流程为

```
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; 程序开始设数据区首址
        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

```

(4) 输入程序后, 按 F2 键保存程序 (规定扩展名为 .ASM), 使用 ALT+X 返回集成软件环境。

(5) 按 F2 键, 输入源程序名 (省略扩展名, 系统默认为 .ASM), 对源程序进行汇编, 生成目标文件 (*.obj) 及错误信息文件 (*.M)。若给出 LST 文件名则生成相应 *.LST 文件。

(6) 汇编无误后, 按 F3 键输入文件名 (对汇编生成的 *.obj 文件进行连接, 连接信息显示于屏幕上, 生成相应 *.EXE 可执行文件。

(7) 按 F5 键, 输入 *.EXE 可执行文件名, 填入程序段地址 CS: 0000 及偏移量地址 IP: 2000, PC 开始将程序从磁盘装入到教学系统内存, 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, 见表 1.1。

表 1.1

显示信息	键入信息
>	U2000
2000 BF0035 MOV DI, 3500	U
2003 B91000 MOV CX, 0010	U
...	...

(8) 当发现源程序输入错误或需调整时, 在调试界面下可用 A 命令来修改, 如修改 2000 句为 MOV DI, 3500 的操作见表 1.2。

表 1.2

显示信息	键入信息
>	A2000
0000: 2000	MOV DI, 3500
0000: 2003	
...	...

2) 运行程序

TDN86/51 系统提供了单步运行、设断点运行、连续运行等方式，具体操作如下：

(1) 单步运行。其操作见表 1.3，每运行一条指令后会显示下一条待执行指令并以蓝底白字显示变化寄存器的内容，重复 T↵（或按 F4 键）就可一步一步地运行，直至程序结束。

表 1.3

显示信息	键入信息
>	B↵
[0:]	2009
[1:]	↵
>	

(2) 连续运行。在“>”提示符下键入 G=0000：2000↵（在系统默认段址 CS=0000 情况下也可直接键入 G=2000↵）可连续运行程序，在运行过程中，可通过 Ctrl+C 终止程序运行，返回监控状态；也可通过 Ctrl+S 暂停程序运行，按任意键继续。

(3) 断点运行。在程序中可用 B 命令定义断点，系统规定至多定义 10 个断点，如上例中定义了 2009 地址为断点，每次键入 GB=0000：2000↵程序连续运行至断点时，程序中断并显示当前各寄存器内容。GB 是 G 命令的扩充，表示含断点连续运行程序，断点仅当系统复位时清除。

(3) 内存单元内容的显示。使用 D=0000：3500↵可查看 3500H~350FH 单元中的内容是否为 0~15 共 16 个数。

(4) 内存单元内容的修改。若要修改某一单元内容，可进行如下操作，其中，“空格”键用于向待编辑单元的高地址方向移动地址，而“-”键则向反方向移动地址；用↵来响应，就退出 E 命令，见表 1.4。

表 1.4

显示信息	键入信息
>	E3500↵
0000：3500 00_	01 space
0000：3501 01_	Space
0000：3502 02_	-
0000：3503 03_	↵
>	

2. 系统操作练习（2）

将内存 3500H 单元开始的 0~15 共 16 个数传递到 3600H 单元开始的数据区中。实验流

程如图 1.1 所示，程序为

```

STACK SEGMENT STACK
    DW 64 DUP (?)
STACK ENDS
CODE SEGMENT
    ASSUME CS: CODE
START: MOV     SI, 3500H; 建立源地址指针
        MOV     DI, 3600H; 建立目标地址指针
        MOV     CX, 0008H; 字数送入 CX 寄存器中
A1:     MOV     AX, [SI]; 取源数据一字送入 AX 中
        MOV     [DI], AX; 将源数据送入目标
                        地址单元中
        INC     SI; 修正指针
        INC     SI
        INC     DI
        INC     DI
        DEC     CX; 判断是否传送完毕
        JNZ     A1
A2:     JMP     A2
CODE ENDS
END START

```

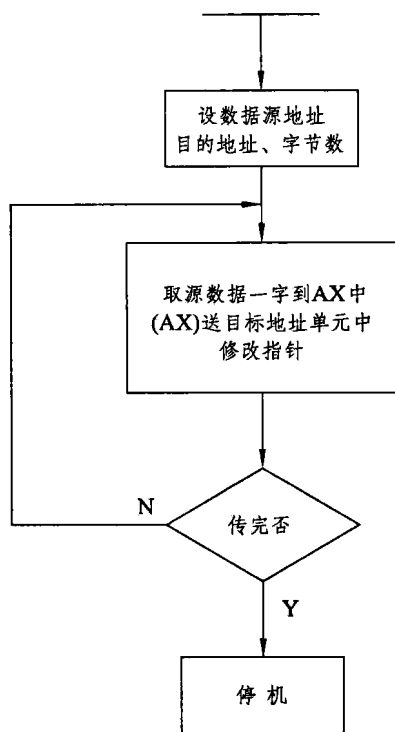


图 1.1 实验流程

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) E3500✓，在 3500H~3510H 单元中分别送入 00、01、02、…。
- (3) G=0000：2000✓，运行以上程序，按 Ctrl+C 中断，返回监控。
- (4) D3600✓，显示结果：3600 00 01 02 …。

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，数码对应关系见表 2.1。

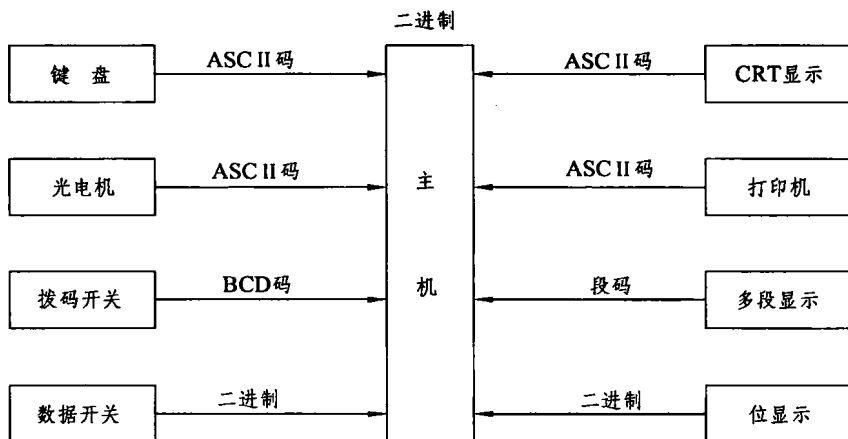


图 2.1 数码转换关系

表 2.1 数码对应关系

十六进制数	BCD 码	二进制机器码	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

1. 将 ASCII 码表示的十进制数转换为二进制数

十进制表示为

$$D_n \times 10^n + D_{n-1} \times 10^{n-1} + \cdots + D_0 \times 10^0 = D_i \times 10^i D_i \quad (2.1)$$

上式转换为：

$$\sum_{i=0}^n D_i \times 10^i = ((\cdots (D_n \times 10 + D_{n-1}) + D_{n-2}) \times 10 + \cdots + D_1) \times 10 + D_0 \quad (2.2)$$

由式 (2.2) 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。程序流程如图 2.2 所示，参考程序为(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 3510H~3511H 单元中)。

```
STACK SEGMENT STACK
```

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

```
STACK ENDS
```

```
CODE SEGMENT
```

```
ASSUME CS: CODE
```

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

```
MOV DI, 3510H
```

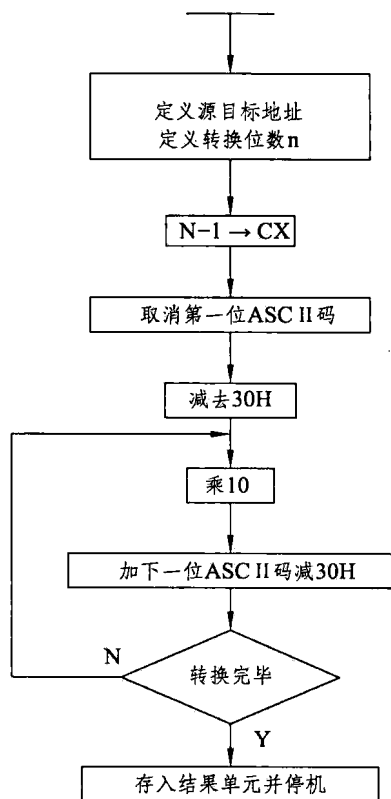


图 2.2 程序流程

```

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 3500H~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 2.1 可知，一字节 ASCII 码取其低四位即变为 BCD 码。实验程序及流程为

```

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; ASCII 码首址
A1:     MOV     BL, 0FFH; 将错误标志送入 BL
        MOV     AL, [DI]; 送 ASCII 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 ASCII 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 3500H~3504H 单元中存放五位十进制数的 ASCII 码, 即 E3500 $\swarrow$, 并输入 31、32、33、34、35。
- (3) G=0000:2000 $\swarrow$, 运行程序。
- (4) D350A $\swarrow$, 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 ASCII 码表示的十进制数

十六位二进制数的值域为 0~65 535, 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 ASCII 码表示的十进制数, 就是求 D_1-D_4 , 并将它们转化为 ASCII 码。实验程序流程如图 2.3 所示, 程序为(设源数存于 3500H~3501H 单元中, 结果数存于 3510H~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

```

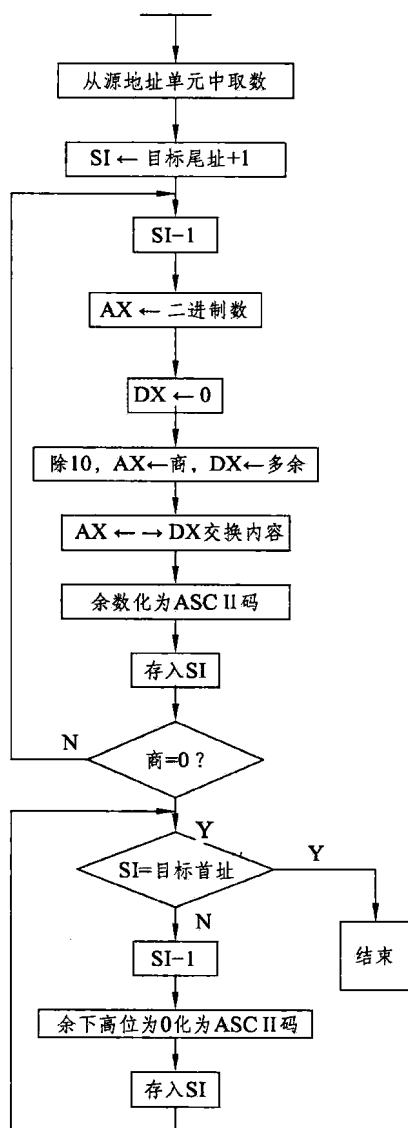


图 2.3 程序流程

```

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, 0000H
      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 3500H~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。实验程序流程如图 2.4 所示，程序为

```

STACK SEGMENT STACK
      DW 64 DUP (?)
STACK ENDS
CODE SEGMENT
      ASSUME CS: CODE

```

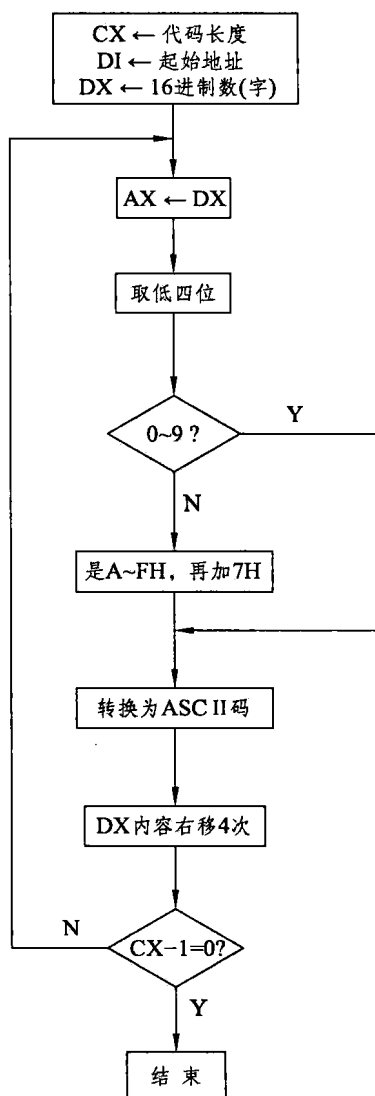


图 2.4 程序流程

```

START:  MOV    CX, 0004H
        MOV    DI, 3500H
        MOV    DX, [DI]
A1:     MOV    AX, DX
        AND    AX, 000FH; 取低4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 $\swarrow$, 并输入 3B、20。
- (3) G=0000:2000 $\swarrow$, 运行以上程序。
- (4) D350A $\swarrow$, 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, 实验程序流程如图 2.5 所示, 程序为

```

STACK SEGMENT STACK
        DW 64 DUP (?)
STACK ENDS
CODE SEGMENT
        ASSUME CS: CODE

```

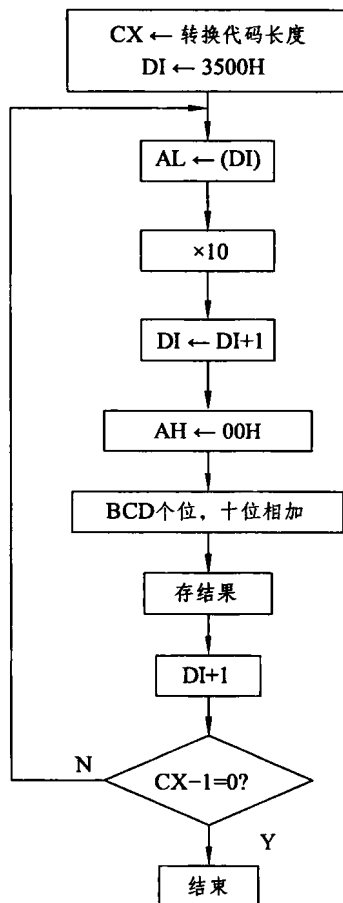


图 2.5 程序流程

```

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]; BCD 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。
- (2) 在 3500H~3507H 单元中存入四个十进制数的 BCD 码，即 E3500↙，输入 01、02、03、04、05、06、07、08。
- (3) G=0000：2000↙，运行以上程序。
- (4) D3510↙，显示结果为：
3510 0C 00 22 00 38 00 4E 00
- (5) 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 5 位十进制数转换为二进制数（16 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 16 位二进制数转换为 ASCII 码十进制数时，如何确定 D_1 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 D_x 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。

表 3.1 数据类型算术运算表

数字	二进制		BDC 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）

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，实验程序为

```

STACK SEGMENT STACK
DW 64 DUP (?)
STACK ENDS
DATA SEGMENT
XL DW ?; 在此处给 X 低位赋值
XH DW ?; 在此处给 X 高位赋值
YL DW ?; 在此处给 Y 低位赋值
YH DW ?; 在此处给 Y 高位赋值

```

地址	存储器
3500H	A0
3501H	65
3502H	15
3503H	00
3504H	9E
3505H	B7
3506H	21
3507H	00
3508H	
3509H	
350AH	
350BH	