

前言

微型计算机技术是一门最新科学和实际应用密切结合的科学技术。《微型计算机技术及应用》一书自 91 年 12 月出版并印刷九次以来,作者常常收到读者来信,这些来信除了探讨问题以外,还常常希望能对书中讲述的一些关键性技术提供更多的应用示范。

这类要求和愿望的确是一种推动性力量。来自美国的报道说,根据统计,在计算机软件开发过程中,从流程设计、程序编制到调试运行直至整个程序正确通过,一个有经验的程序员平均每天的程序完成量为 20—30 行,而遇到与关键性技术及硬件有关的编程,则效率将更低。由此想到,如果对微型计算机技术中一些最重要的部分给出行之有效的实例,对许多读者来讲,将可以大大缩短这段摸索和熟悉的时间。

具体说,比如要从磁盘上读取一个文件,或者写入一个文件,怎样用汇编语言程序来实现?对初学者来说,在没有指导的情况下,要编制并调通这两个程序,决非一朝一夕能完成。另外,将一个班级或一个年级的学生按成绩进行排队,这在每个学校的工作中都要用到。为解决这个问题,传统的方法是引入庞大的数据库管理系统。实际上,用汇编语言程序就可以很巧妙很简洁地解决。可是,如何编制这个程序呢?还有,编制计算机钢琴程序时,因为涉及乐理知识而常常被认为是一件颇为麻烦的事,初学者甚至可能不知所措。

像上面这些问题都可以在这本《微型机软件硬件开发指南》中找到满意的答案。读者阅后可能会恍然大悟:原来并不很复杂呀!是的,只要手边有一台 PC 机,不论是何种档次的,你就可以自己马上动手试一试。不过,更重要的是能够由此举一反三,触类旁通。我愿每个读者都能成为程序开发和学以致用的高手和专家。

1991 年至 1992 年,笔者和清华大学计算机工厂的冯一兵等老师一起开发了“TPC-1 型十六位微机实验培训系统”,这样,涉及硬件的部分也可通过实验得以牢固掌握了。比如,怎样利用 8255A 并行端口来传输数据?又怎样利用 8251A 串行端口来传输数据?还有,在设计控制系统时,总要用到 A/D、D/A 转换,那么,怎样来组成这样一个系统并编制程序达到控制的目的呢?在我遇到的读者中,提出关于中断控制器和 DMA 控制器方面问题的使用者较多,这些读者在微型机技术开发中曾做过许多工作。他们想设计一块接口板或者一个子系统,以使用中断控制器或者 DMA 控制器来实现整个系统的关键功能。现在这些读者可以从这本开发指南中找到重要的解决思路,并受到不同程度的启迪。

在开发“TPC-1 型十六位微机实验培训系统”的过程中,除了和我一起负责研制系统的冯一兵老师外,还很感谢清华大学计算机系朱家维老师给予的关心以及清华大学计算机工厂的杨春武老师、汤弘寿老师在各方面给予的支持和帮助;在软件编制和调试过程中,我系计 8 张红斌和计 7 柳瑞恒同学协助做了不少具体工作。笔者在此表示最深切的谢意。

本书提供的程序都是用汇编语言编写的。和各种高级语言相比,汇编语言由于其程序精悍、高速、易于和硬件相接、并可嵌入大部分高级语言中而具有长久、独特的生命力。

戴梅萼

1995. 12

目 录

一、软件部分

实验 1	两个多位十进制数相加的实验	3
实验 2	两个数相乘的实验	6
实验 3	BCD 码相乘的程序	8
实验 4	字符匹配程序	12
实验 5	字符串匹配程序	16
实验 6	从键盘输入数据并显示的程序	19
实验 7	字符和数据的显示程序	22
实验 8	响铃程序	26
实验 9	接收年、月、日信息并显示的程序	28
实验 10	将键盘输入的小写字母转换成大写字母的程序	32
实验 11	保留最长行输入字符的程序	35
实验 12	计算机钢琴的程序	38
实验 13	排序实验	41
实验 14	学生成绩名次表实验	46
实验 15	设置光标的实验	51
实验 16	建立窗口的实验	53
实验 17	计算 $N!$ 的实验	55
实验 18	写文件的实验	60
实验 19	读文件的实验	66
实验 20	显示目录的实验	70

二、硬件部分

实验 21	8253 计数器/定时器的实验	76
实验 22	8255A 并行口的实验(一)	79
实验 23	8255A 并行口的实验(二)	84
实验 24	8255A 并行口的实验(三)	89
实验 25	8251A 串行口的实验	92
实验 26	8259A 中断控制器实验	101
实验 27	D/A 实验	105
实验 28	A/D 实验	110
实验 29	RAM 实验	118
实验 30	DMA 实验	121
实验 31	LED 显示实验	125
实验 32	微机接口综合实验	130

一、软件部分

实验 1 两个多位十进制数相加的实验

一、实验目的

1. 学习数据传送和算术运算指令的用法。
2. 熟悉在 PC 机上建立、汇编、链接、调试和运行汇编语言程序的过程。

二、实验内容

将两个多位十进制数相加。要求加数和被加数均以 ASCII 码形式各自顺序存放在以 DATA1 和 DATA2 为首的 5 个内存单元中(低位在前),结果送回 DATA1 处。

三、程序框图

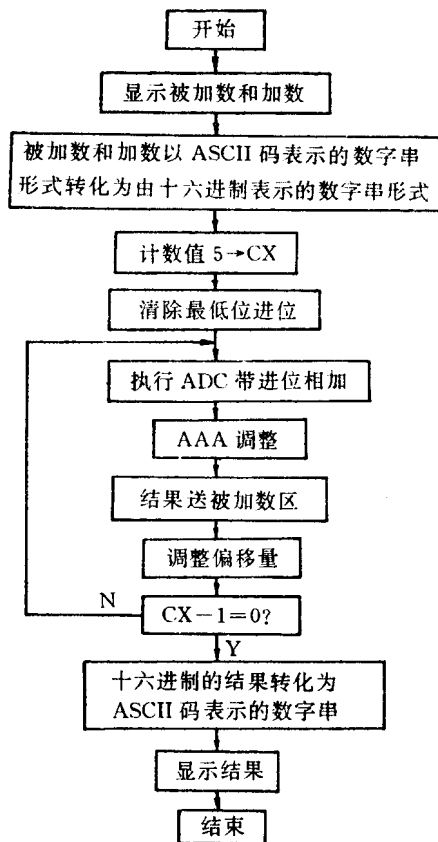


图 1 两个多位十进制数相加的流程图

四、程序清单

```
CRLF      MACRO                                ;建立宏指令 CRLF
          MOV      DL,0DH
          MOV      AH,02H
          INT      21H
          MOV      DL,0AH
          MOV      AH,02H
          INT      21H
ENDM

DATA      SEGMENT                                数据段
DATA1     DB   33H,39H,31H,37H,34H                ;第一个数据(作为加数)
DATA2     DB   36H,35H,30H,38H,32H                ;第二个数据(作为被加数)
DATA      ENDS
STACK     SEGMENT                                ;堆栈段
STA       DB   20 DUP (?)
TOP       EQU   LENGTH STA
STACK     ENDS
CODE      SEGMENT                                ;程序段
ASSUME    CS:CODE,DS:DATA,SS:STACK,ES:DATA
START:    MOV     AX,DATA
          MOV     DS,AX
          MOV     AX,STACK
          MOV     SS,AX
          MOV     AX,TOP
          MOV     SP,AX
          MOV     SI,OFFSET DATA2
          MOV     BX,05
          CALL    DISPL                            ;显示被加数
          CRLF
          MOV     SI,OFFSET DATA1
          MOV     BX,05
          CALL    DISPL                            ;显示加数
          CRLF
          MOV     DI,OFFSET DATA2
          CALL    ADDA                            ;加法运算
          MOV     SI,OFFSET DATA1
          MOV     BX,05                            ;显示结果
          CALL    DISPL
          CRLF
          MOV     AX,4C00H
```

	INT	21H	
DISPL	PROC	NEAR	;显示子程序
DS1:	MOV	AH,02	
	MOV	DL,[SI+BX-1]	;显示字符串中一字符
	INT	21H	
	DEC	BX	;修改偏移量
	JNZ	DS1	
	RET		
DISPL	ENDP		
ADDA	PROC	NEAR	
	MOV	DX,SI	
	MOV	BP,DI	
	MOV	BX,05	
AD1:	SUB	BYTE PTR [SI+BX-1],30H	
	SUB	BYTE PTR [DI+BX-1],30H	
	DEC	BX	;将 ASCII 码表示的数字串转化为十六进制的数字串
	JNZ	AD1	
	MOV	SI,DX	
	MOV	DI,BP	
	MOV	CX,05	;包括进位,共 5 位
	CLC		;清进位位
AD2:	MOV	AL,[SI]	
	MOV	BL,[DI]	
	ADC	AL,BL	;带进位相加
	AAA		;非组合 BCD 码的加法调整
	MOV	[SI],AL	;结果送加数区
	INC	SI	
	INC	DI	;指向下一位
	LOOP	AD2	;循环
	MOV	SI,DX	
	MOV	DI,BP	
	MOV	BX,05	
AD3:	ADD	BYTE PTR [SI+BX-1],30H	
	ADD	BYTE PTR [DI+BX-1],30H	
	DEC	BX	;十六进制的数字串转化为 ASCII 码表示的数字串
	JNZ	AD3	
	RET		
ADDA	ENDP		
CODE	ENDS		
END	START		

实验 2 两个数相乘的实验

一、实验目的

掌握乘法指令和循环指令的用法。

二、实验内容

实现十进制数的乘法。被乘数和乘数均以 ASCII 码形式存放在内存中,乘积在屏幕上显示出来。

三、程序框图

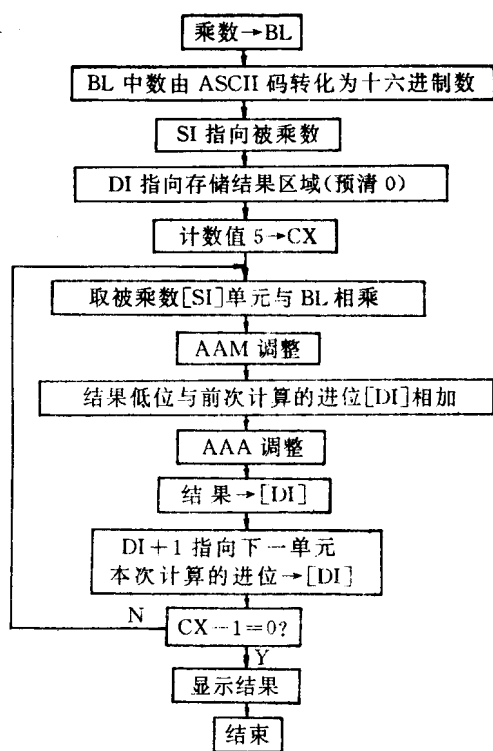


图 2 两个数相乘的流程图

四、程序清单

```
DATA    SEGMENT
DATA1    DB      32H,39H,30H,35H,34H
DATA2    DB      33H
RESULT   DB      6 DUP (00H)
```



```

DATA      ENDS
STACK     SEGMENT
STA       DB          20 DUP (?)
TOP       EQU          LENGTH STA
STACK     ENDS
CODE      SEGMENT
ASSUME CS: CODE, DS: DATA, SS: STACK, ES: DATA
START:    MOV          AX, DATA
          MOV          DS, AX
          MOV          AX, STACK
          MOV          SS, AX
          MOV          AX, TOP
          MOV          SP, AX
          MOV          SI, OFFSET DATA2
          MOV          BL, [SI]          ;乘数 2→BL
          AND          BL, 00001111B    ;屏蔽高 4 位, ASCII 码转化为十六进制数
          MOV          SI, OFFSET DATA1
          MOV          DI, OFFSET RESULT
          MOV          CX, 05
LOOP1:    MOV          AL, [SI]
          AND          AL, 00001111B    ;取被乘数, 并将 ASCII 码变成十六进制数
          INC          SI              ;指向被乘数的下一字符单元
          MUL          BL              ;相乘
          AAM              ;AAM 调整
          ADD          AL, [DI]        ;结果低位与前次计算的进位相加
          AAA              ;AAA 调整
          MOV          [DI], AL
          INC          DI              ;结果送下一单元
          MOV          [DI], AH
          LOOP         LOOP1          ;计算结果进位送存储单元
          MOV          CX, 06
          MOV          SI, OFFSET RESULT
DISPL:    MOV          AH, 02
          MOV          DL, [SI]
          ADD          DL, 30H
          INT          21H
          DEC          SI
          LOOP         DISPL          ;显示结果
          MOV          AX, 4C00H
          INT          21H          ;结束
CODE      ENDS
END       START

```

实验3 BCD 码相乘的程序

一、实验目的

掌握用组合的 BCD 码表示数据,并熟悉怎样实现组合 BCD 码乘法运算。

二、实验内容

实现 BCD 码的乘法。要求乘数 1 和乘数 2 以组合的 BCD 码形式存放,各占一个内存单元,乘积存放在另外两个内存单元中。

由于没有组合的 BCD 码乘法指令,程序中采用将乘数 1 作为计数器,累加另一个乘数的方法得到计算结果。

三、程序框图

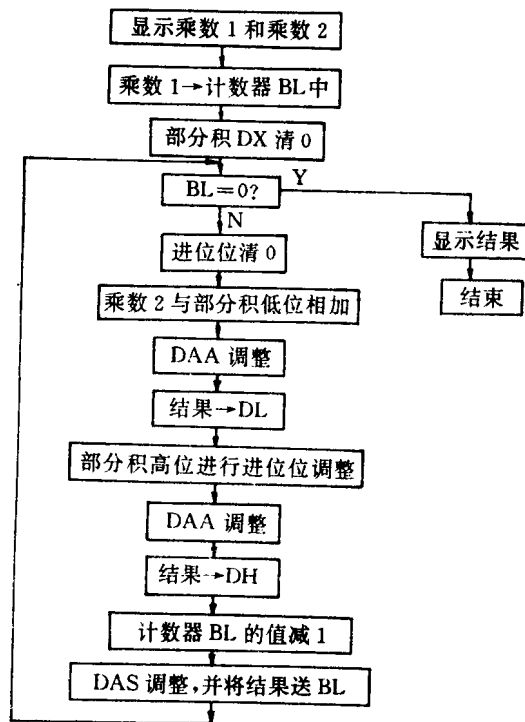


图 3 BCD 码相乘的流程图

四、程序清单

DISPL	MACRO		; 显示功能宏指令
	ADD	DL, 30H	
	MOV	AH, 02	

```

        INT            21H
        ENDM

CRLF    MACRO                                ;回车换行宏指令
        MOV            DL,0DH
        MOV            AH,02
        INT            21H
        MOV            DL,0AH
        MOV            AH,02
        INT            21H
        ENDM

DATA    SEGMENT
DATA1    DB            01001000B                ;48H
DATA2    DB            01110101B                ;75H
RESULT    DW            ?
DATA    ENDS
STACK    SEGMENT
STA        DB            20 DUP (?)
TOP        EQU            LENGTH STA
STACK    ENDS
CODE    SEGMENT
ASSUME    CS:CODE,DS:DATA,SS:STACK,ES:DATA
START:    MOV            AX,DATA
        MOV            DS,AX
        MOV            AX,STACK
        MOV            SS,AX
        MOV            AX,TOP
        MOV            SP,AX
        MOV            DI,OFFSET RESULT        ;指向结果存储区
        MOV            SI,OFFSET DATA1        ;指向乘数 1
        MOV            AH,[SI]
        MOV            CL,04
        SHR            AH,CL                    ;显示乘数 1 高位字节
        MOV            DL,AH
        DISPL
        MOV            AL,[SI]
        MOV            BL,AL                    ;显示乘数 1 低位字节
        AND            AL,00001111B
        MOV            DL,AL
        DISPL
        CRLF                                ;回车换行
        MOV            SI,OFFSET DATA2        ;指向乘数 2

```

	MOV	AH,[SI]	
	MOV	CL,04	
	SHR	AH,CL	;显示乘数 2 高位字节
	MOV	DL,AH	
	DISPL		
	MOV	AL,[SI]	
	AND	AL,00001111B	
	MOV	DL,AL	;显示乘数 2 低位字节
	DISPL		
	CRLF		;回车换行
	XOR	DX,DX	;DX(部分积)清 0
S1:	CMP	BL,0	
	JZ	S2	;乘数 1 的值已减为 0 否?
	CLC		
	MOV	AL,[SI]	
	ADC	AL,DL	;乘数 2 加部分积低位
	DAA		;组合的 BCD 码调整
	MOV	DL,AL	;结果送部分积高位
	MOV	AL,00H	
	ADC	AL,DH	;往部分积高位送进位
	DAA		;组合 BCD 码调整
	MOV	DH,AL	;结果送部分积高位
	MOV	AL,BL	
	DEC	AL	;乘数 1 的值减 1
	DAS		;组合 BCD 码减法调整
	MOV	BL,AL	;结果送 BL
	JMP	S1	
S2:	MOV	[RESULT],DX	;结果送存储单元
	MOV	SI,DX	
	MOV	CL,4	
	SHR	DH,CL	
	MOV	DL,DH	
	DISPL		;显示结果高 2 位
	MOV	DX,SI	
	AND	DH,00001111B	
	MOV	DL,DH	
	DISPL		
	MOV	DX,SI	
	MOV	CL,4	
	SHR	DL,CL	
	DISPL		;显示结果低 2 位
	MOV	DX,SI	

```

        AND          DL,00001111B
        DISPL
        CRLF          ;回车换行
S3:      MOV          AX,4C00H
        INT           21H          ;返回 DOS
CODE
END      START
```

实验 4 字符匹配程序

一、实验目的

掌握汇编语言中串操作指令的使用方法。

二、实验内容

用串操作指令设计程序,实现在存储区(长度为 100H)中寻找空格字符(20H)。退出时给出是否找到的信息。

三、编程提示

1. 8086 中用于字符串检索的指令为 SCASB/SCASW,用 AL 中的字节或者 AX 中的字与位于 ES 段由 DI 寄存器所指的内存单元的字节或字相比较。通过前缀的控制,可以实现在 DI 所指的字符串中,寻找第一个与 AL(或 AX)的内容相同的字节(或字)。

2. 对于所有的串操作指令,都需要注意方向标志的设置。指令 CLD 使方向标志 DF 清 0,从而使 SI 和 DI 作自动增量修改。指令 STD 使 DF 置 1,从而使 SI 和 DI 作自动减量修改。

四、程序框图

见图 4 和图 5。

五、程序清单

```
CRLF      MACRO                                ,回车换行宏指令
            MOV     DL,0DH
            MOV     AH,02H
            INT     21H
            MOV     AH,02H
            MOV     DL,0AH
            INT     21H
ENDM
DATA      SEGMENT
MESS1     DB        'INPUT THE SEGMENT ADDR : $'
MESS2     DB        'INPUT THE OFFSET ADDR : $'
MESS3     DB        'OK! HAS FOUND!',0DH,0AH,' $'
MESS4     DB        'NO SPACE CHARACTER!',0DH,0AH,' $'
ERROR     DB        'INPUT ERROR!',0DH,0AH,' $'
DATA      ENDS
STACK     SEGMENT
STA       DW        32 DUP (?)
```

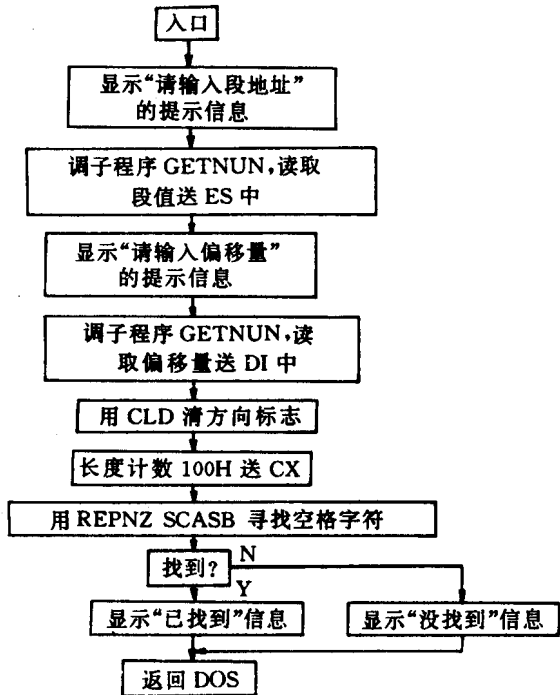


图 4 字符匹配主程序流程图

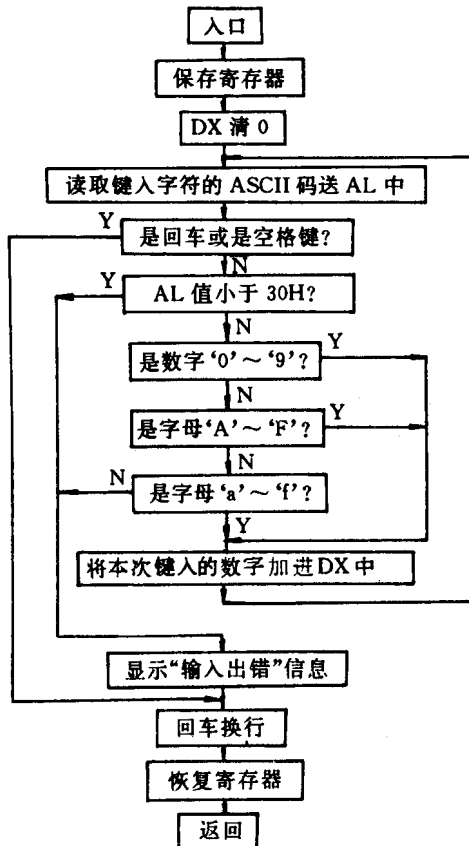


图 5 字符匹配 GETNUN 子程序流程图

```

TOP      DW      ?
STACK    ENDS
CODE     SEGMENT
ASSUME   CS:CODE,DS:DATA,ES:DATA,SS:STACK
START:   MOV     AX,DATA
          MOV     DS,AX           ;初始化
          MOV     ES,AX
          MOV     AH,09H
          MOV     DX,OFFSET MESS1
          INT     21H             ;显示信息 1
          CALL    GETNUM          ;读取键入的十六进制数送 DX
          MOV     ES,DX           ;段址送 CS
          MOV     AH,09H
          MOV     DX,OFFSET MESS2
          INT     21H             ;显示提示信息 2
          CALL    GETNUM          ;读取键入的数
  
```

	MOV	DI,DX	;偏移量送 DI
	CLD		;清方向标志
	MOV	CX,0100H	;长度为 100H 字节
	MOV	AL,20H	;空格符 20H
	REPNZ	SCASB	
	JNZ	AAA	;全都不为 20H 则转 AAA
	MOV	AH,09H	
	MOV	DX,OFFSET MESS3	
	INT	21H	;显示‘找到’信息
	JMP	BBB	
AAA:	MOV	AH,09H	
	MOV	DX,OFFSET MESS4	
	INT	21H	;显示‘没找到’信息
BBB:	MOV	AX,4C00H	
	INT	21H	;结束
GETNUM	PROC	NEAR	;读取键入的十六进制数子程序
	PUSH	CX	;保存
	XOR	DX,DX	;DX 清 0
GGG:	MOV	AH,01H	
	INT	21H	;读取键入字行的 ASCII 码送 AL
	CMP	AL,0DH	
	JZ	PPP	
	CMP	AL,20H	
	JZ	PPP	;是回车键或空格键则转 PPP
	SUB	AL,30H	
	JB	KKK	;非数字转 KKK
	CMP	AL,0AH	
	JB	GETS	;在‘0’~‘9’间转 GETS
	CMP	AL,11H	
	JB	KKK	;非数字转 KKK
	SUB	AL,07H	
	CMP	AL,0FH	
	JBE	GETS	;在‘A’~‘F’间转 GETS
	CMP	AL,2AH	
	JB	KKK	
	CMP	AL,2FH	
	JA	KKK	;不在‘a’~‘f’间转 KKK
	SUB	AL,20H	;‘a’~‘f’的 ASCII 码值转化为 16 进制 数值
GETS:	MOV	CL,04	
	SHL	DX,CL	
	XOR	AH,AH	

	ADD	DX,AX	;将本次键入的数字加进 DX 中
	JMP	GGG	;接收下一个键入的数字
KKK:	MOV	AH,09H	
	MOV	DX,OFFSET ERROR	
	INT	21H	;显示输入出错信息
PPP:	PUSH	DX	
	CRLF		;回车换行
	POP	DX	
	POP	CX	
	RET		
GETNUM	ENDP		
CODE	ENDS		
END	START		