

计算机组成与接口实验

（计算机原理与接口技术课程使用）

2011 年 5 月. 广州

第一部分 指令系统实验

实验一

求从 0 开始连续 100 个偶数之和，并将结果存入名字为 SUM 的字存储单元中。试用完整的段定义语句编写出实现这一功能的汇编源程序。

答：程序如下：

```
DATA    SEGMENT
SUM     DW    0
DATA    ENDS
STACK   SEGMENT STACK
        DB    200 DUP(0)
STACK   ENDS
CODE    SEGMENT
        ASSUME DS:DATA, SS:STACK, CS:CODE
START:  MOV    AX, DATA
        MOV    DS, AX
        MOV    CX, 100
        MOV    AX, 0
        MOV    BX, 0
NEXT:   ADD    AX, BX
        INC    BX
        INC    BX
        DEC    CX
        JNE    NEXT
        MOV    SUM, AX
        MOV    AH, 4CH
        INT    21H
CODE    ENDS
        END    START
```

实验二

设一存储区中存放有 10 个带符号的单字节数，现要求分别求出其绝对值后存放到原单元中，试编写出汇编程序。

答：程序如下：

```
DATA    SEGMENT
BLOCK   DB    -10, 15H, 20H, -1, -23, 46H, 16H, -33, 65H, 88H
DATA    ENDS
CODE    SEGMENT
        ASSUME DS:DATA, CS:CODE
START:  MOV    AX, DATA
        MOV    DS, AX
        MOV    SI, OFFSET BLOCK
        MOV    CX, 10
```

```

AGAIN:  MOV    AL, [SI]
        TEST   AL, 80H    ;测试是否为正数
        JZ     NEXT
        NEG     AL        ;求负数的绝对值
        MOV     [SI], AL
NEXT:    INC     SI
        LOOP   AGAIN
        MOV     AH, 4CH    ;返回 DOS
        INT     21H
CODE    ENDS
        END     START

```

实验三

假定 DX 和 BX 中各包含有 4 位压缩 BCD 数（1234 和 5678），将 DX 和 BX 中的 BCD 数相加且将和数存入 CX，试编写出实现这一功能的汇编源程序。

答：程序如下：

```

CODE    SEGMENT
        ASSUME CS:CODE
START:  MOV DX, 1234H    ;装入 1234
        MOV BX, 5678H    ;装入 5678
        MOV AL, BL        ;低字节之和
        ADD AL, DL
        DAA                ;低字节调整
        MOV CL, AL        ;和数低字节送 CL
        MOV AL, BH        ;高字节及进位之和
        ADC AL, DH
        DAA                ;高字节调整
        MOV CH, AL        ;和数高字节送 CH
        MOV AH, 4CH
        INT 21H
CODE    ENDS
        END     START

```

实验四

在当前数据段、偏移地址为 DATA 的内存区中，顺序地存放着以 BLOCK 为首址的 8 个字节的无符号数，若要求其算术平均值，并将结果接着这 8 个数存放，试编写此汇编程序。

答：程序如下：

```

DATA    SEGMENT
BLOCK   DB  1, 2, 3, 4, 5, 6, 7, 12
P_SUM   DB  ?
DATA    ENDS
CODE    SEGMENT
        ASSUME CS:CODE, DS:DATA
START:  MOV AX, DATA

```

```

        MOV DS, AX
        MOV CX, 8           ;置循环计数器
        LEA DI, BLOCK       ;取数据块首址
        MOV AX, 0           ;AX 用于数组求和
        MOV AL, [DI]        ;AX 中的初值取第一个数组元素
        INC DI              ;DI 指向下一个数组元素
        DEC CX
P_ADD:  ADD AL, [DI]         ;对数组求和
        ADC AH, 0           ;有进位时，要向高位进位
        INC DI              ;指向下一个数组元素
        LOOP P_ADD
P_DIV:  MOV CL, 3           ;用右移 3 次实现和数除以 8
        SHR AX, CL
        MOV P_SUM, AL       ;1 个字节的平均数存入 P_SUM 单元
        MOV AH, 4CH
        INT 21H
CODE    ENDS
        END START

```

实验五

试编写实现 $-4003H \div 4$ 有符号除法运算的汇编程序。

答：程序如下：

```

CODE    SEGMENT
        ASSUME CS:CODE
START:  MOV AX, -4003H
        CWD           ;DX 被扩展延伸
        MOV CX, 4
        IDIV CX
CODE    ENDS
        END START

```

实验六

设 a、b、c、d 均为 16 位数据常数，依次用 DW 伪指令定义为 A、B、C、D 变量，将常数 35 赋值给变量 F，要求完成 $[(a+b-c)/8]*35+d$ 算式的运算，将结果按低 16 位、高 16 位顺序送 E、E+2 字存储单元中，试编写完成上述算式运算、存储结果及返回 DOS 状态的程序。

（可假设 a=10, b=5, c=7, d=31）

答：程序如下：

```

DATA SEGMENT
A    DW  a
B    DW  b
C    DW  c
D    DW  d
E    DW  2 DUP(?)
F    EQU 35

```

```

DATA    ENDS
STACK  SEGMENT STACK
        DB 200 DUP(?)
STACK  ENDS
CODE    SEGMENT
        ASSUME DS:DATA, SS:STACK, CS:CODE
START:  MOV AX, DATA
        MOV DS, AX
        MOV AX, A
        ADD AX, B
        SUB AX, C
        MOV CL, 3
        SAR AX, CL
        MOV BX, F
        IMUL BX
        ADD AX, D
        ADC AX, 0
        MOV E, AX
        MOV E+2, DX
        MOV AH, 4CH
        INT 21H
CODE    ENDS
        END START

```

实验七

试编写一短程序，将 AX(1356H)中的高 4 位移至 DX 的低 4 位。

答：程序如下：

```

CODE    SEGMENT
        ASSUME CS:CODE
NEXT:    MOV AX, 1356H
        XOR DX, DX
        MOV CX, 4    ;置循环次数 4
        SAL AX, 1
        RCL DX, 1    ;将 AX 中的最高位移入 DX 的最低位
        DEC CX
        JNE NEXT    ;若 CX≠0，则循环
        MOV AH, 4CH
        INT 21H
CODE    ENDS
        END NEXT

```

实验八

试编写一短程序，将 AX 中的低 4 位移至 DX 的高 4 位。

答：程序如下：

```

CODE    SEGMENT
        ASSUME CS:CODE
        MOV CX, 4    ; 置循环次数 4
NEXT:   SHR AX, 1
        RCR DX, 1    ; 将 AX 中的最低位移入 DX 的最高位
        DEC CX
        JNE NEXT    ; 若 CX≠0, 则循环
        MOV AH, 4CH
        INT 21H
CODE    ENDS
        END NEXT

```

实验九

比较两个字符串 MESS1 和 MESS2 所含字符是否完全相同，若相同则显示 ‘MATCH’，若不同则显示 ‘NO MATCH’，试编写此汇编程序。

答：程序如下：

```

DATA    SEGMENT
MESS1   DB   ' I AM A TEACHER.'
MESS2   DB   ' I AM A TEACHAR.'
YES      DB   ' MATCH', 0DH, 0AH, '$'
NO       DB   ' NO MATCH', 0DH, 0AH, '$'
DATA    ENDS
CODE    SEGMENT
        ASSUME CS:CODE, DS:DATA
START:  MOV     AX, DATA
        MOV     DS, AX
BEGIN:  LEA     SI, MESS1
        LEA     DI, MESS2
        MOV     CX, MESS2- MESS1
        REPE    CMPSB
        JNE     DISPNO
        LEA     DX, YES
        MOV     AH, 9
        INT     21H
        MOV     AH, 4CH
        INT     21H
DISPNO: LEA     DX, NO
        MOV     AH, 9
        INT     21H
        MOV     AH, 4CH
        INT     21H
CODE    ENDS
        END     START

```

实验十

已知某数组 ARRAY 中有 20 个有符号字节数，试用汇编语言一完整程序，统计该数据中相邻两数间符号变化（+变- 或 -变+）的次数，并将次数保存在存储单元 NUM 中。

答：程序如下：

```
DATA    SEGMENT
ARRAY   DB  -12, 24, -15, 6, 9, -16, 45, 89, 33, -22, -26, 18, 19, 10, 0, 56, 38, 37, -55, -68
NUM      DB  0
DATA    ENDS
CODE    SEGMENT
        ASSUME CS:CODE , DS:DATA
START:  MOV  AX, DATA
        MOV  DS, AX
        LEA  SI, ARRAY
        MOV  AL, [SI]
        MOV  BL, 0
        MOV  CX, 19
AGAIN:  INC  SI
        XOR  AL, [SI]
        JNS  NEXT
        INC  BL
NEXT:   MOV  AL, [SI]
        LOOP AGAIN
        MOV  NUM, BL
        MOV  AH, 4CH
        INT  21H
CODE    ENDS
        END  START
```

第二部分 汇编语言编程实验

实验一

对两个 8 字节无符号数求和，这两个数分别用变量 DATA1 及 DATA2 表示。将两数之和的最高进位放在 AL 中，两数之和的其他位按从高到低顺序依次放在 SI, BX, CX, DX 中。

解：程序如下：

```
DATA    SEGMENT
DATA1   DB  0CDH, 0BCH, 0ABH, 9AH, 78H, 56H, 34H, 12H
DATA2   DB  12H, 34H, 56H, 78H, 9AH, 0ABH, 0BCH, 0CDH
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
BG:     MOV AX, DATA
        MOV DS, AX
        LEA DI, DATA1
MOV DX, [DI]
        MOV CX, [DI+2]
MOV BX, [DI+4]
MOV SI, [DI+6]
        LEA DI, DATA2
        ADD DX, [DI]
        ADC CX, [DI+2]
ADC BX, [DI+4]
ADC SI, [DI+6]
MOV AL, 0
        ADC AL, 0
        MOV AH, 4CH
        INT 21H
CODE    ENDS
        END BG
```

实验二

比较以存储变量 DATA1 和 DATA2 表示的两个有符号数据的大小，将其中较大数据放在 BX 寄存器中。

解：程序如下：

```
DATA    SEGMENT
DATA1   DW  -113
DATA2   DW  -110
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS:CODE, DS:DATA
BEGIN:  MOV AX, DATA
        MOV DS, AX
        MOV BX, DATA1
```



```

        CMP BX, DATA2
        JGE NEXT
        MOV BX, DATA2
NEXT:   MOV AH, 4CH
        INT 21H
CODE    ENDS
        END BEGIN

```

实验三

从 AA 单元开始的 20 个连续单元中存放有 20 个无符号数，从中找出最大数送入 BB 单元中，试编程。

答：先把第一个数送入 AL 寄存器，将 AL 中的数与后面的 19 个数逐个比较，若 AL 中的数小于相比较的数，将两两交换位置，否则两数不交换位置。在比较过程中，AL 中始终保持较大的数，比较 19 次，则 AL 中必为最大数，然后将其送入 BB 单元中。

```

DATA    SEGMENT
AA      DB  36, 58, 16, 46, 71, 118, 73, 52, 41, 62
        DB  2, 18, 8, 24, 56, 79, 113, 87, 104, 75
BB      DB  ?
DATA    ENDS
CODE    SEGMENT
        ASSUME DS:DATA, CS:CODE
MAIN    PROC FAR
START:  PUSH    DS
        MOV AX, 0
        PUSH    AX
        MOV AX, DATA
        MOV DS, AX
        MOV AL, AA
        MOV BX, OFFSET AA
        MOV CX, 29
LOOP1:  INC BX
        CMP AL, [BX]
        JAE LOOP2
        XCHG    AL, [BX]
LOOP2:  DEC CX
        JNZ LOOP1
        MOV BB, AL
        RET
MAIN    ENDP
CODE    ENDS
        END START

```

实验四

试编写一汇编源程序：找出从无符号字节数据存储变量 VAR 开始存放的 N 个数中的最大数放在 BH 中。

解：程序如下：

```
DSEG      SEGMENT
VAR        DB 3,56H, 6, 19H, 23H, 0A5H
N          EQU $-VAR
DSEG      ENDS
CSEG      SEGMENT
          ASSUME CS: CSEG, DS: DSEG
BG:        MOV AX, DSEG
          MOV DS, AX
          MOV CX, N-1
          MOV SI, 0
          MOV BH, VAR[SI]
          JCXZ LAST
AGIN:      INC SI
          CMP BH, VAR[SI]
          JAE NEXT
          MOV BH, VAR[SI]
NEXT:      LOOP AGIN
LAST:      MOV AH, 4CH
          INT 21H
CSEG      ENDS
          END BG
```

实验五

试编写一汇编源程序：将一组有符号存储字节数据按从小到大的顺序排列。设数组变量为 VAR，数组元素个数为 N。

```
D          SEGMENT
VAR        DB -1, -10, -100, 27H, 0AH, 47H
N          EQU $-VAR
D          ENDS
C          SEGMENT
          ASSUME CS: C, DS: D
B:         MOV AX, D
          MOV DS, AX
          MOV CX, N-1
          MOV DX, 1
AG:        CALL SUBP
          INC DX
          LOOP AG
          MOV AH, 4CH
          INT 21H
          SUBP PROC
```

```

        PUSH CX
        MOV CX, N
        SUB CX, DX
        MOV SI, 0
RECMPT: MOV AL, VAR[SI]
        CMP AL, VAR[SI+1]
        JLE NOCH
        XCHG AL, VAR[SI+1]
        XCHG AL, VAR[SI]
NOCH:   INC SI
        LOOP RECMPT
        POP CX
        RET
SUBP    ENDP
C       ENDS
        END B

```

实验六

试编写一源程序：在显示器上显示“HOW ARE YOU?”，然后读一个字符。但不显示此字符，若读入字符是‘Y’，则显示“OK”。

解：程序如下：

```

D       SEGMENT
D1      DB 'HOW ARE YOU?', 0DH, 0AH, '$'
D2      DB 'OK', 0DH, 0AH, '$'
D       ENDS
C       SEGMENT
        ASSUME CS:C, DS:D
BG:     MOV AX, D
        MOV DS, AX
        MOV DX, OFFSET D1
        MOV AH, 9
        INT 21H
        MOV AH, 8
        INT 21H
        CMP AL, 'Y'
        JNE NEXT
        LEA DX, D2
        MOV AH, 9
        INT 21H
NEXT:   MOV AH, 4CH
        INT 21H
C       ENDS
        END BG

```

实验七

试编写一源程序：屏幕显示“PASSWORD?”，随后从键盘读入字符串，并比较这个字符串与程序内部设定的字符串。若二者相同则显示“OK”，否则不作任何显示（0DH 是回车的 ASCII 码）。

解： 程序如下：

```
D      SEGMENT
PASS1  DB '12AB'
N      EQU $-PASS1
D1     DB 'PASSWORD?', 0DH, 0AH, '$'
PASS2  DB 20
        DB ?
        DB 20 DUP(?)
D2     DB 0DH, 0AH, 'OK$'
D      ENDS
C      SEGMENT
        ASSUME CS:C, DS:D, ES:D
BG:    MOV AX, D
        MOV DS, AX
        MOV ES, AX
        LEA DX, D1
        MOV AH, 9
        INT 21H
        LEA DX, PASS2
        MOV AH, 0AH
        INT 21H
        LEA SI, PASS1
        LEA DI, PASS2
        CMP BYTE PTR[DI+1], N
        JNE LAST
        MOV CX, N
        LEA DI, PASS2+2
        CLD
        REPZ CMPSB
        JZ DISOK
LAST:  MOV AH, 4CH
        INT 21H
DISOK: LEA DX, D2
        MOV AH, 9
        INT 21H
        JMP LAST
C      ENDS
END BG
```

第三部分 接口技术实验

实验一

【实验内容】利用 DOS 系统功能调用，从键盘输入一串字符，分别统计字母、数字和其他字符的个数，并输出显示统计结果，编写出实现这一功能的汇编源程序。

解：程序如下：

```
DATA            SEGMENT
MAXSTRING      DB    100
INACT          DB    ?
STRING         DB    100
DISMESS        DB    'PLEASE ENTER A STRING:', 0AH, 0DH, '$'
DIGITAL        DB    'DIGITAL IS:', '$'
LETTER         DB    'LETTER IS:', '$'
OTHERCHAR      DB    'OTHER IS:', '$'
CRLF           DB    0AH, 0DH, '$'
DATA            ENDS
CODE            SEGMENT
                ASSUME CS: CODE, DS: DATA
MAIN            PROC    FAR
                MOV     AX, DATA
                MOV     DS, AX
                MOV     BL, 0
                MOV     BH, 0
                MOV     CH, 0
DISPSTRING      MACRO
                MOV     AH, 9
                INT     21H
                ENDM
DISPCHAR        MACRO
                MOV     AH, 2
                INT     21H
                ENDM
                LEA     DX, DISMESS
                DISPSTRING
BEGIN:          LEA     DX, MAXSTRING
                MOV     AH, 0AH
                INT     21H
                MOV     DL, INACT
                MOV     DH, 0
                INC     DX
                LEA     SI, STRING
REPEAT:         DEC     DX
                JZ      ENDCHE
                MOV     AL, [SI]
```

```

        INC     SI
        CMP     AL, '0'
        JB      OTHER
        CMP     AL, '9'
        JA      NEXT1
        INC     BL
        JMP     REPEAT
NEXT1:  OR      AL, 20H
        CMP     AL, 'a'
        JB      OTHER
        CMP     AL, 'z'
        JA      OTHER
        INC     BH
        JMP     REPEAT
OTHER:  INC     CH
        JMP     REPEAT
ENDCHE: LEA     DX, DIGITAL
        DISPSTRING
        MOV     CL, BL
        CALL    DISP
        LEA     DX, LETTER
        DISPSTRING
        MOV     CL, BH
        CALL    DISP
        LEA     DX, OTHERCHAR
        DISPSTRING
        MOV     CL, CH
        CALL    DISP
        MOV     AH, 4CH
        INT     21H
MAIN    ENDP
DISP    PROC    NEAR
        MOV     AL, CL
        MOV     AH, 0
        MOV     CL, 100
        DIV     CL
        MOV     CL, AH
        MOV     DL, AL
        ADD     DL, 30H
        DISPCHAR
        MOV     AL, CL
        MOV     AH, 0
        MOV     CL, 10
        DIV     CL

```

```

        MOV     CL, AH
        MOV     DL, AL
        ADD     DL, 30H
        DISPCHAR
        MOV     DL, CL
        ADD     DL, 30H
        DISPCHAR
        LEA     DX, CRLF
        DISPSTRING
        RET
DISP    ENDP
CODE    ENDS
        END     MAIN

```

实验二

【实验内容】把地址偏移量为 100H 单元开始的 256 个单元分别写入数据 00H, 01H, 02H, 03H, ..., FFH, 并用 DOS 功能调用显示各单元的内容。为了显示清晰明了, 要求每行显示 16 个数据, 两个数据之间用空格分开, 编写出实现这一功能的汇编源程序。

解: 程序如下:

```

        DATA    SEGMENT
                ORG    100H
        BUF1     DB    256 DUP(?)
        COUNT    EQU    $-BUF1
        DATA    ENDS
        CODE     SEGMENT
                ASSUME CS:CODE, DS:DATA
        START:   MOV     AX, DATA
                MOV     DS, AX
                MOV     SI, OFFSET BUF1
                MOV     CX, COUNT
                PUSH    SI
                PUSH    CX
                XOR     AL, AL
        NEXT:    MOV     [SI], AL
                INC     AL
                INC     SI
                LOOP    NEXT
                POP     CX
                POP     SI
        ZRBH:    MOV     BH, 16
        NEXT1:   MOV     AL, [SI]
                MOV     BL, AL
                SHR     AL, 1
                SHR     AL, 1

```

```

        SHR     AL, 1
        SHR     AL, 1
        CMP     AL, 0AH
        JC      JIA30
        ADD     AL, 7
JIA30:   ADD     AL, 30H
        CALL    DISP
        MOV     AL, BL
        AND     AL, 0FH
        CMP     AL, 0AH
        JC      JIA30A
        ADD     AL, 7
JIA30A:  ADD     AL, 30H
        CALL    DISP
        MOV     AL, 20H
        CALL    DISP
        INC     SI
        LOOP    NEXT2
        JMP     STOP
NEXT2:   DEC     BH
        JNZ     NEXT1
        MOV     AL, 0AH
        CALL    DISP
        MOV     AL, 0DH
        CALL    DISP
        JMP     ZRBH
DISP     PROC    NEAR
        MOV     DL, AL
        MOV     AH, 2
        INT     21H
        RET
DISP     ENDP
STOP:    MOV     AH, 4CH
        INT     21H
CODE     ENDS
        END     START

```

实验三 (8253)

【实验内容】以 1MHz 的信号为时钟信号，利用 8253 产生一个周期为 1ms 的方波。

解：程序如下：

```

        CODE     SEGMENT
        MAIN     PROC FAR
                ASSUME CS: CODE
START:    CLI

```



```

MOV DX, 303H
MOV AL, 36H
OUT DX, AL
MOV DX, 300H
MOV AL, 0E8H
OUT DX, AL
MOV AL, 03H
OUT DX, AL
MOV DX, 303H
MOV AL, 74H
OUT DX, AL
MOV DX, 301H
MOV AL, 0AH
OUT DX, AL
MOV AL, 00H
OUT DX, AL
STI
MOV AH, 4CH
INT 21H
MAIN ENDP
CODE ENDS
END START

```

实验四(8255)

【实验内容】读入开关量到 8255A，再将其通过指示灯显示。以 8255A 的 A 口作输入，B 口作输出，输入用开关，输出用发光二极管，要求当输入不全为 0 时，输入与输出保持一致。当输入为全 0 时，发光二极管闪烁。

解：程序如下：

```

PORTA=308H
PORTB=309H
CONTROL=30BH
DCONS=10H
DATA SEGMENT
MESS DB 0DH, 0AH, 'PRESS ANYKEY TO EXIT...', 0AH, 0DH, '$'
DATA ENDS
CG SEGMENT 'CODE'
ASSUME CS: CG, DS: DATA
BEGIN: PUSH DS
        XOR AX, AX
        PUSH AX
        MOV AX, DATA
        MOV DS, AX
        MOV DX, OFFSET MESS

```

```

MOV AH, 9
INT 21H
MOV DX, CONTROL
MOV AL, 90H
OUT DX, AL
MOV AH, 0FFH
MOV BL, 0
LP:    PUSH AX
        MOV AH, 0BH
        INT 21H
        CMP AL, 0
        JNE BACK
        POP AX
        MOV DX, PORTA
        IN AL, DX
        NOP
        NOP
        TEST AL, AH
        JZ SHIFT
        MOV DX, PORTB
        OUT DX, AL
        JMP LP
SHIFT:  MOV AL, BL
        MOV DX, PORTB
        OUT DX, AL
        CALL DELAY
        SHL BL, 1
        TEST BL, AH
        JNZ LP
        MOV BL, 1
        JMP LP
BACK:   POP AX
        RETF
DELAY   PROC NEAR
        PUSH AX
        MOV BH, DCONS
DDLY:   MOV CX, 0
DELAY1: DEC CX
        LOOP DELAY1
        DEC BH
        TEST BH, AH
        JNZ DDLY
        POP AX
DELAY   ENDP

```

```
CG          ENDS
            END BEGIN
```

实验五（8259A）

【实验内容】要求使用 8259A的IRQ₀，用单脉冲信号模拟外部中断信号，使中断程序在屏幕上显示信息，在中断程序中，要求依次显示a~z这 26 个字母。

```
STACK      SEGMENT STACK
            DB 200 DUP(0)

STACK      ENDS
C8259      EQU 20H
CODE       SEGMENT 'CODE'
MAIN       PROC FAR
            ASSUME CS:CODE, SS:STACK

START:     XOR AX, AX
            MOV DS, AX
            LEA AX, IRQ0
            MOV DS: 20H, AX
            MOV DX, C8259
            MOV AL, 00010011B
            OUT DX, AL
            INC DX
            MOV AL, 00001000B
            OUT DX, AL
            MOV AL, 00001101B
            OUT DX, AL
            MOV AL, 11111110B
            OUT DX, AL
            MOV AL, 20H
            OUT 20H, AL
            MOV BL, 'a'
            MOV AL, BL
            MOV AH, 1
            INT 14H
            STI
            JMP $

IRQ0:      CLI
            INC BL
            CMP BL, '{'
            JNZ NEXT
            MOV AH, 4CH
            INT 21H

NEXT:      MOV AL, BL
            MOV AH, 1
            INT 14H
```

```

MOV AL, 20H
OUT 20H, AL
STI
IRET
MAIN    ENDP
CODE    ENDS
END START

```

实验六 (8250)

【实验内容】两台微机之间按 **RS232C** 标准协议使用 **COM1** 进行串行通信，构成一个字符或数据的各位按时间先后，从低位到高位一位一位地传送。

解：程序如下：

```

STACK   SEGMENT PARA STACK 'STACK'
        DB 256 DUP(0)
STACK   ENDS
CODE    SEGMENT PARA PUBLIC 'CODE'
START   PROC FAR
        ASSUME CS: CODE
        PUSH DS
        MOV AX, 0
        PUSH AX
        MOV DX, 3FBH
        MOV AL, 80H
        OUT DX, AL
        MOV DX, 3F8H
        MOV AL, 12
        OUT DX, AL
        INC DX
        MOV AL, 0
        OUT DX, AL
        MOV DX, 3FBH
        MOV AL, 0AH
        OUT DX, AL
        MOV DX, 3FCH
        MOV AL, 03H
        OUT DX, AL
        MOV DX, 3F9H
        MOV AL, 0
        OUT DX, AL
FOREVER: MOV DX, 3FDH
        IN AL, DX
        TEST AL, 1EH
        JNZ ERROR

```

```

TEST AL, 01H
JNZ RECEIVE
TEST AL, 20H
JZ FOREVER
MOV AH, 1
INT 16H
JZ FOREVER
MOV AH, 0
INT 16H
MOV DX, 3F8H
OUT DX, AL
MOV AH, 2
MOV DL, AL
INT 21H
CMP AL, 27
JNZ NEXT
MOV AH, 4CH
INT 21H
NEXT: JMP FOREVER
RECEIVE: MOV DX, 3F8H
IN AL, DX
AND AL, 7FH
CMP AL, 27
JNZ DISP
MOV AH, 4CH
INT 21H
DISP: PUSH AX
MOV BX, 0
MOV AH, 14
INT 10H
POP AX
CMP AL, 0DH
JNZ FOREVER
MOV AL, 0AH
MOV BX, 0
MOV AH, 14
INT 10H
JMP FOREVER
ERROR: MOV DX, 3F8H
IN AL, DX
MOV AL, '?'
MOV BX, 0
MOV AH, 14
INT 10H

```

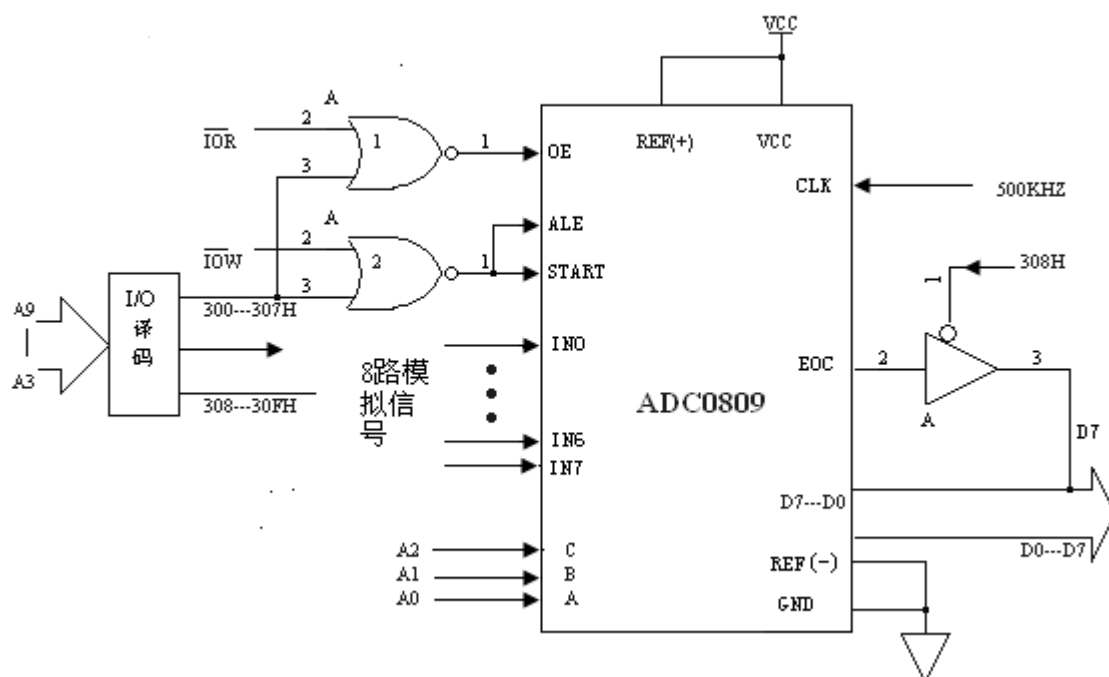
```

                                JMP FOREVER
START      ENDP
CODE       ENDS
END START

```

实验七(A/D 及 D/A)

【实验内容】如果要在 PC/XT 机上采用 ADC0809 设计一块 8 通道的数据采集卡,要求以 200Hz 的速率对每个通道均采集 1024 个数据,也就是每隔 5ms 对各通道轮流采集一个数据,然后将它们存到数据段中以 DBUF 为始址的数据缓冲区中。数据存放的次序须与通道号一致,即从通道 0 开始,先依次存入每个通道的第一个数据,再存入各通道的第二个数据,直到各通道都存满 1024 个数据为止。



分析: 可以选用 8253 芯片来产生定时脉冲, 控制采样率。假设加到 8253 的 CLK₀ 的时钟脉冲的频率为 1MHz, 编程使通道 0 工作于方式 2, 由于采样率 $f_s = 200\text{Hz}$, 当选用时间常数为 $1\text{MHz}/200\text{Hz} = 5000$ 使 8253 工作时, 则可从 OUT₀ 端输出 200 Hz 的负脉冲序列, 即每隔 5ms 会从 8253 的 OUT₀ 引脚输出一个正跳变脉冲, 该脉冲加到 PC 机上为用户保留的 IRQ₂ 中断请求输入端, 即加到系统板上 8259A 的 IR₂ 引脚上, 在 8259A 的控制下定时向 CPU 发中断请求, 在每次中断时进行采样。

解: 程序如下:

```

DATA      SEGMENT
DBUF      DB      8*1024 DUP(?)
DATA      ENDS
CODE      SEGMENT
ASSUME CS: CODE, DS: DATA

```

```

AD_8      PROC    FAR
          MOV     AX, DATA
          MOV     DS, AX
          CLI
          CLD
          MOV     AX, 0
          MOV     ES, AX
          MOV     DI, 4*0AH
          MOV     AX, OFFSET ADINT
          STOSW
          MOV     AX, SEG ADINT
          STOSW
          MOV     DX, 31BH
          MOV     AL, 00110101B
          OUT     DX, AL
          MOV     DX, 318H
          MOV     AX, 5000H
          OUT     DX, AL
          MOV     AL, AH
          OUT     DX, AL
          MOV     AL, 11111001B
          OUT     21H, AL
          MOV     SI, OFFSET DBUF
          MOV     BX, 1024
          STI
AGAIN:     CMP     BX, 0
          JNZ     AGAIN
          MOV     AL, 11111101B
          OUT     21H, AL
          MOV     AH, 4CH
          INT     21H
          RET
AD_8      ENDP
ADINT     PROC    NEAR
          MOV     CX, 0008H
          MOV     DX, 300H
NEXT:     OUT     DX, AL
          PUSH    DX
          MOV     DX, 308H
          JNZ     POLL
POLL:     IN      AL, DX
          TEST    AL, 80H
          JNZ     POLL
NO_END:   IN      AL, DX

```

```

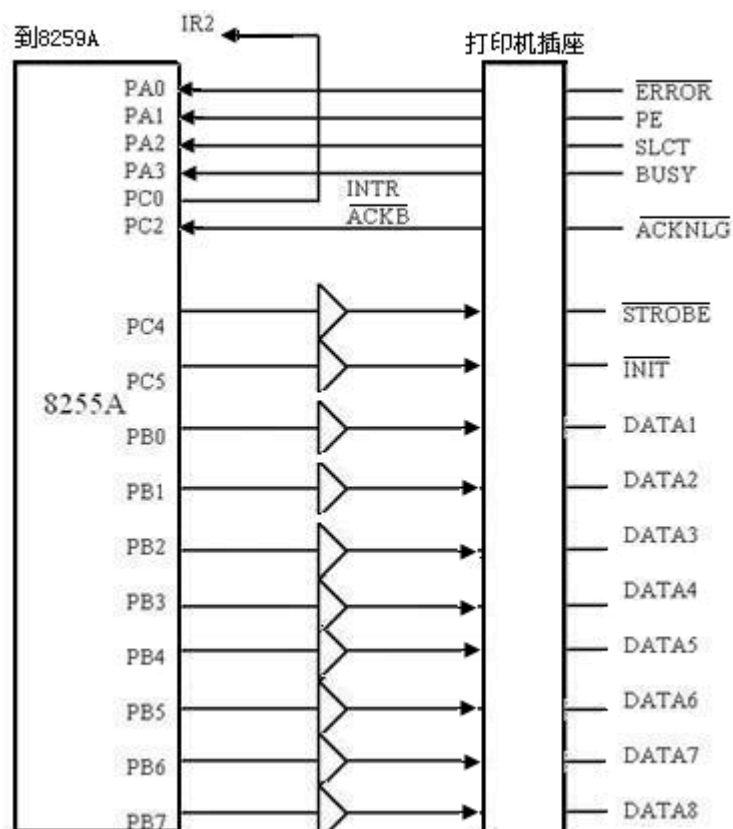
TEST    AL, 80H
JZ      NO_END
POP     DX
IN      AL, DX
MOV     [SI], AL
INC     DX
INC     SI
LOOP    NEXT
DEC     BX
MOV     AL, 20H
OUT     20H, AL
IRET
ADINT   ENDP
CODE    ENDS
END

```

实验八

【实验内容】如果要在打印机上打印字符串“*How do you do!*”，试编程。

【实验要求】设该系统中，8255A 的 A 口、B 口、C 口和控制字寄存器的端口地址分别为 2F8H、2FAH、2FCH 和 2FEH。中断控制器 8259A 的偶地址端口地址为 2F0H，奇地址端口地址为 2F2H。在计算机中，用来管理打印机的程序称为打印机驱动程序，现要求采用中断方式管理打印机的驱动程序的方法。



PRINTER TEST

=====

； 断口地址分配表

PORT_A	EQU	2F8H
PORT_B	EQU	2FAH
PORT_C	EQU	2FCH
PORT_CTL	EQU	2FEH
PORT_0	EQU	2F0H
PORT_1	EQU	2F2H

； 数据段

DATA	SEGMENT	
MESS_1	DB	'How do you do!'
	DB	0DH, 0AH
MESS_LEN	EQU	\$-MESS_1
PRINT_DONE	DB	0
POINTER	DW	0
COUNT	DB	0
PRNT_ERR	DB	0
DATA	ENDS	

； 堆栈段

STACK	SEGMENT	STACK
	DW	50 DUP(0)
TOP	LABEL	WORD
STACK	ENDS	

打印主程序

=====

； 打印主程序

CODE	SEGMENT	
	ASSUME	CS:CODE, DS:DATA, SS:STACK
MAIN	PROC	FAR
	MOV	AX, STACK
	MOV	SS, AX
	LEA	SP, TOP
	MOV	AX, CS
	MOV	DS, AX
	MOV	DX, OFFSET PRNT_INT
	MOV	AH, 25H
	MOV	AL, 0AH
	INT	21H

； 初始化 8259A, 使 IR2 中断允许

MOV	DX, PORT_0
MOV	AL, 00010011B

```

OUT    DX, AL
MOV    DX, PORT_1
MOV    AL, 00001000B
OUT    DX, AL
MOV    AL, 00000001B
OUT    DX, AL
MOV    AL, 11111001B
OUT    DX, AL

```

； 初始化 8255A， B 口方式 1 输出， A 口方式 0 输入， C 口高 4 位为输出

```

MOV    DX, PORT_CTL
MOV    AL, 10010100B
OUT    DX, AL
STI

```

； 通过 PC4 向打印机送高电平选通信号

```

MOV    AL, 00001001B
OUT    DX, AL

```

； 初始化打印机， 从 PC5 引脚送出 50us 宽的 INIT 负脉冲

```

MOV    AL, 00001011B
OUT    DX, AL
MOV    AL, 00001010B
OUT    DX, AL
MOV    CX, 17H
PAUSE_1: LOOP    PAUSE_1
MOV    AL, 00001011B
OUT    DX, AL

```

； 从端口 A 读取打印机状态， 已准备好状态应为 AL=XXXX0101

```

MOV    PRNT_ERR, 0
MOV    DX, PORT_A
IN     AL, DX
AND    AL, 0FH
CMP    AL, 00000101B
JZ     SEND_IT
MOV    CX, 16EAH
PAUSE_2: LOOP    PAUSE_2
IN     AL, DX
AND    AL, 0FH
CMP    AL, 00000101B
JZ     SEND_IT
MOV    PRNT_ERR, 1
JMP    FIN

```

； 已准备好， 建立指向信息存储区的指针， 已打印完标志清 0， 表示未打印完

```

SEND_IT: MOV    AX, OFFSET MESS_1
MOV    POINTER, AX
MOV    PRNT_DONE, 0

```

```

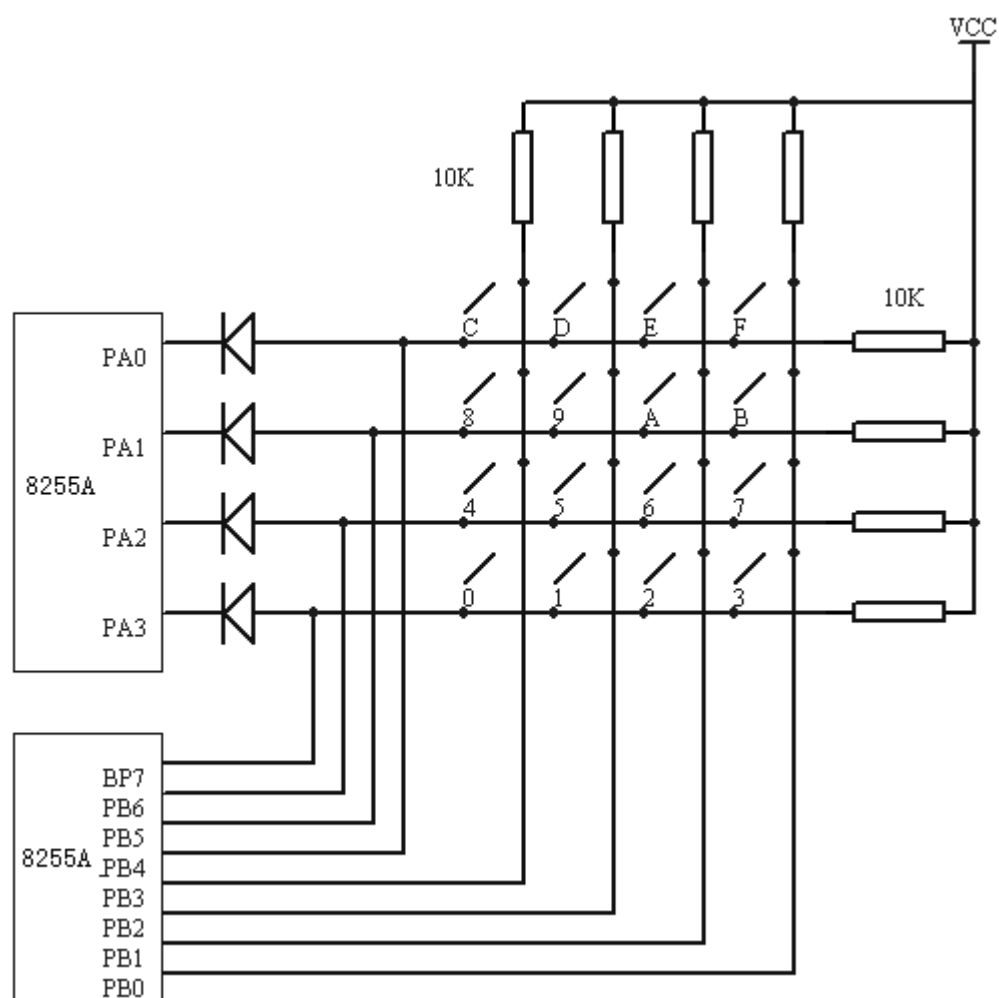
                                MOV     COUNT, MESS_LEN
; 置位 PC2, 使 8255A 的 INTEB 置 1, 允许中断
                                MOV     DX, PORT_CTL
                                MOV     AL, 00000101B
                                OUT     DX, AL
; 等待打印机中断
                                WAIT_INT: JMP     WAIT_INT
                                FIN:     NOP
                                    NOP
                                    MOV     AH, 4CH
                                    INT     21H
                                MAIN     ENDP
; 打印驱动中断服务子程序
                                PRNT_INT PUSH    AX
                                    PUSH    BX
                                    PUSH    DX
                                    STI
                                    MOV DX, PORT_B
                                    MOV BX, POINTER
                                    MOV AL, [BX]
                                    OUT DX, AL
; 通过 PC4 向打印机发选通负脉冲
                                    MOV DX, PORT_CTL
                                    MOV AL, 00001000B
                                    OUT DX, AL
                                    MOV AL, 00001001B
                                    OUT DX, AL
; 增量地址指针, 减量计数器
                                    INC POINTER
                                    DEC COUNT
                                    JNZ NEXT
; 字符已打印完, 复位 PC2, 禁止 8255PC0 上的中断请求
                                    MOV AL, 00000100B
                                    OUT DX, AL
                                    MOV PRNT_DONE, 1
                                NEXT:  MOV AL, 00100000B
                                    MOV DX, PORT_0
                                    OUT DX, AL
                                    POP DX
                                    POP BX
                                    POP AX
                                    IRET
                                CODE    ENDS
                                END

```

实验九

【实验内容】试编写一个键盘程序。

【实验要求】已知 16 个键分别为数字 0~9 和 A~F，键盘排列的连线及其接口电路如图所示。16 个键排成 4 行×4 列的矩阵，接到微型机系统中由两片 8255A 组成的一对端口上。其中，端口 A 作输出，端口 B 作输入。端口地址为：PORT_A: FF9H; PORT_B : 0FFBH; PORT_CTL: 0FFFH。



参考程序如下：

;端口地址

PORT_A EQU 0FF9H

PORT_B EQU 0FFBH

PORT_CTL EQU 0FFFH

DATA SEGMENT

0 1 2 3 4 5 6

	TABLE	DB	77H, 7BH, 7DH, 7EH, 0B7H , 0BBH, 0BDH,
0BEH			
			8 9 A B C D E
F			
		DB	0D7H, 0DBH, 0DDH, 0DEH, 0E7H, 0EBH, 0EDH, 0EEH
	DATA	ENDS	
	STACK	SEGMENT	STACK
		DW 50	DUP(0)
	TOP_STAC	LABEL	WORD
	STACK	ENDS	
	CODE	SEGMENT	
		ASSUME	CS: CODE, DS: DATA, SS: STACK
	START:	MOV	AX, STACK
		MOV	SS, AX
		LEA	SP, TOP_STACK
		MOV	AX, DATA
		MOV	DS, AX
		MOV	DX, PORT_CTL
		MOV	AL, 10001011B
		OUT	DX, AL
		MOV	DX, PORT_A
		MOV	AL, 00H
		OUT	DX, AL
		MOV	DX, PORT_B
	WAIT_OPEN:	IN	AL, DX
		AND	AL, 0FH
		CMP	AL, 0FH
		JNE	WAIT_OPEN
	WAIT_PRES:	IN	AL, DX
		AND	AL, 0FH
		CMP	AL, 0FH
		JE	WAIT_PRES
		MOV	CX, 16EAH
	DELAY:	LOOP	DELAY
		IN	AL, DX
		AND	AL, 0FH
		CMP	AL, 0FH
		JE	WAIT_PRES
		MOV	AL, 0FEH
		MOV	CL, AL
	NEXT_ROW:	MOV	DX, PORT_A
		OUT	DX, AL
		MOV	DX, PORT_B
		IN	AL, DX

```

                                AND     AL, 0FH
                                CMP     AL, 0FH
                                JNE     ENCODE
                                ROL     CL, 01
                                MOV     AL, CL
                                JMP     NEXT_ROW
ENCODE:                        MOV     BX, 000FH
                                IN      AL, DX
NEXT_TRY:                      CMP     AL, TABLE[BX]
                                JE      DONE
                                DEC     BX
                                JNS     NEXT_TRY
                                MOV     AH, 01
                                JMP     EXIT
DONE:                          MOV     AL, BL
                                MOV     AH, 00
EXIT:                          HLT
CODE                           ENDS
                                END

```

实验十

【实验内容】模拟电子琴程序设计。

【程序设计流程】可以分解成两个部分问题：即如何控制计算机的扬声器发音、如何使不同按键对应约定的声音。

模拟演奏程序算法：

第一步：初始化，建立分频数与发音键表；

第二步：清屏，并显示“PIANO 2004.8.15.\$”；

第三步：等待按键，有键按下则往下执行；

第四步：判断若是退出键，则返回操作系统；

第五步：判断若是停止发音键，则断开送往扬声器的电平信号，并转第三步；

第六步：评断若不是退出、不是停止发音、也不是发音键，转第三步；

第七步：查出与发音键对应的发音分频数，并启动 8253 的通道 2，转第三步。

第三步采用 DOS 的中断调用“键盘无回显”方式输入，即按下键时，不会在屏幕上出现相应显示。

第四步可约定回车键为退出键。

第五步可约定空格键为停止发音键。

第七步为了对 8253 通道 2 进行初始化。可向 43H 端口送控制字 B6H，含意是：选择通道 2，使其按方式 3 工作（送出方波信号），向通道 2 送 16 位分频数，先送低 8 位，后送高 8 位，通道 2 按二进制计数。


```

MOV DH, 14
MOV DL, 60
MOV AH, 6
INT 10H
MOV AH, 2
MOV BH, 0
MOV DH, 13
MOV DL, 30
INT 10H
MOV AH, 9
MOV DX, OFFSET DISP
INT 21H
MOV SI, 0
MOV CX, 42
AG1: MOV DX, 12H
MOV AX, 3280H
DIV D1[SI]
MOV DD1[SI], AX
ADD SI, 2
LOOP AG1
AG2: MOV AH, 8
INT 21H
CMP AL, 0DH
JZ LAST
CMP AL, 20H
JNZ NEXT
IN AL, 61H
AND AL, 0FCH
OUT 61H, AL
JMP AG2
NEXT: MOV CX, 42
MOV BX, OFFSET DDD1
SCAN: CMP AL, [BX]
JZ FIND
INC BX
LOOP SCAN
JMP AG2
FIND: MOV SI, 42
SUB SI, CX
SHL SI, 1
MOV AL, 0B6H
OUT 43H, AL
MOV AX, DD1[SI]
OUT 42H, AL

```



```
MOV AL, AH
OUT 42H, AL
IN AL, 61H
OR AL, 3
OUT 61H, AL
JMP AG2
LAST: IN AL, 61H
AND AL, 0FCH
OUT 61H, AL
MOV AH, 4CH
INT 21H
PROGRAM ENDS
END START
```