

# 微机原理实验报告

## 扩展实验四 键盘扫描

[REDACTED]

[REDACTED]

[REDACTED]

2021/12/28

## 一、实验目的

实验仪器上包含一组 16 位的键盘，编写代码，实现对 16 位键盘的扫描。

## 二、实验设备

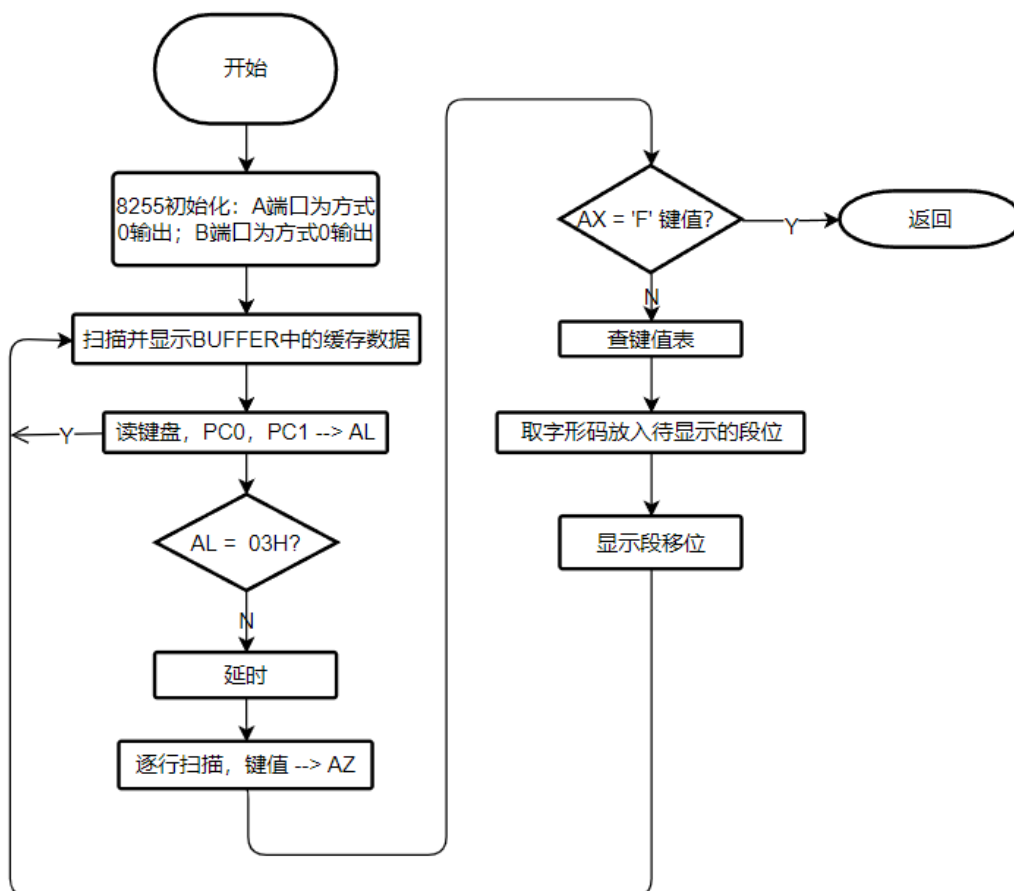
STAR 系列实验仪一套、PC 机一台。

## 三、实验内容

结合前面的几次实验， 实现对实验仪器上 16 位键盘的扫描；

分别按下 F1~F16 按键，数码管上显示不同的数值，例如按下 F1，数码管显示 1，按下 F2，数码管上显示 2，按下 F16，数码管上显示 16。

## 三、程序框图



## 四、实验步骤

### 1、连线说明：

D3 区：CS、A0、A1      —— A3 区：CS1、A0、A1

D3 区：PC0、PC1      —— F5 区：KL1、KL2

D3 区：JP20、B、C      —— F5 区：A、B、C

### 2、在F5 区的键盘上输入数字

### 3、结果显示在F5区的数码管上

## 五、画出程序流程图编写并调试程序

## 六、实验扩展及思考

1. 绘制本实验的详细软件流程图；
2. 结合前面几次的实验内容，给出本次实验的连线表；
3. 扼要注释每指令的功能 。

## 程序

---

```
;8255 行扫描法获取键值
;8255 PA, PB, PC, 控制口地址
PORTA EQU 0270H
PORTB EQU 0271H
PORTC EQU 0272H
CONTR EQU 0273H

_STACK      SEGMENT      STACK
            DW           100      DUP(?)
_STACK      ENDS

DATA        SEGMENT

;数码管字符显示编码
Table DB 0C0H,0F9H,0A4H,0B0H ;[0-3]
        DB 099H,92H, 82H, 0F8H ;[4-7]
        DB 80H, 90H, 88H, 83H  ;[8-11]
        DB 0C6H,0A1H,86H, 8EH  ;[12-15]
        DB 0FFH                ;BLANK

;数码管待显示的数值, 0FH 不显示
BUFFER      DB 0FH, 0FH, 0FH, 0FH
             DB 0FH, 0FH, 0FH, 0FH

;要写入的教码管序号 0-7
Seg2WR      DB 0
DATA        ENDS

CODE        SEGMENT

MAIN        PROC        NEAR
ASSUME      CS:CODE, DS:DATA, SS:_STACK, ES: DATA

            MOV AX, DATA
            MOV DS, AX
            MOV ES, AX

            MOV AX, _STACK
            MOV SS, AX

            MOV AL, 89H
```

```

MOV DX, CONTR
OUT DX, AL ;写入 8255 控制字
CALL InitKeyDisplay
WAIT:
    MOV DX, PORTC ;DX→C 端口
    IN AL, DX
    CMP AL, 03H ;PC0, PC1
    JNE WAIT
LOOP_Init:
    CALL Loop_WR ;扫描 AB 端口
    JMP LOOP_Init ;循环

```

;扫描 AB 端口和键盘

```

Loop_WR PROC NEAR
    PUSH SI ;进栈保护
    MOV SI, 00H
LOOP_1:
    CALL WR_AB ;数码管显示, AB 端口置位
    CALL GetKey ;扫描键盘
    INC SI
    CMP SI, 8
    JNE LOOP_1 ;不相等再写入
    POP SI
    RET
Loop_WR ENDP

```

;对键盘, 数码管控制器 8255 初始化

```

InitKeyDisplay PROC NEAR
    PUSH DX
    PUSH AX
    MOV DX, CONTR ;DX 指向控制端口
    MOV AL, 80H ;8255 为方式 0 输出模式
    OUT DX, AL
    MOV AL, 0FH ;所有行线置低电平
    MOV DX, PORTA
    OUT DX, AL
    MOV DX, PORTB ;所有列线置低电平
    OUT DX, AL
    MOV AL, 00H ;PC 输出为高电平
    MOV DX, PORTC
    OUT DX, AL
    POP AX

```

```
    POP DX
    RET
InitKeyDisplay ENDP
```

;获取按键值

```
GetKey PROC    NEAR
```

```
    PUSH BX
    PUSH DX
    PUSH CX
    PUSH SI
    LEA SI,buffer
    CALL Display8
    POP SI
```

```
    MOV DX, CONTR    ;DX 指向控制端口
    MOV AL, 82H       ;8255 为 A 方式 0 输出, B 方式 0 输入模式
    OUT DX, AL
    ;所有行线置低电平
    ;读入列线, 是否全为 1
    MOV DX, PORTA     ;DX 指向 A 口
    MOV AL, 00H       ;选通各行
    OUT DX, AL
    MOV DX, PORTB     ;DX 指向 B 口
```

WAIT:

```
    IN AL, DX         ;检查各行
    CMP AL, 00H       ;是否和 A 口选通一致
    JE  DONE          ;一致即没有按键压下
    ;去除按键抖动
    MOV AH, AL
    MOV CX, 100
```

DELAY:

```
    LOOP DELAY        ;延时
    IN AL, DX         ;再次读入 PB
    CMP AL, AH         ;判断和延时之前是否相同
    JE  STABLE        ;相同则按键稳定
    JMP WAIT          ;有抖动, 重新读取
```

```
    ;行线依次置为低电平, 读入列线
    ;判断是否有键按下
    ;并计算键值
```

STABLE:

```
    MOV BL, 0         ;置按键号初值
    MOV BH, 1         ;逐行扫描
```

```

    MOV CX, 4      ;设置行计数
FINDROW:
    MOV AL, BH
    MOV DX, PORTA
    OUT DX, AL     ;第一行置低电平
    MOV DX, PORTB
    IN AL, DX      ;取列值
    CMP AL, 0FH    ;判断是否有键按下
    JNZ FINDCOL    ;有, 寻找键值
    ROL BH, 1      ;BH 左移一位, 选通下一行
    ADD BL, 4      ;行增加 1, 键值增加 4
    DEC CX         ;计数 CX --> CX-1
    JNZ FINDROW    ;重复执行
    JMP DONE       ;转无按键下压处理
FINDCOL:
    ROR AL, 1      ;依次处理被按键列值
    JNC RIGHT      ;找到该列的列号
    INC BL         ;没找到, 按键号 BL+1
    JMP FINDCOL    ;重复执行
RIGHT:
    MOV AL, BL     ;按键号 --> AL
DONE:
    IN AL, DX
    AND AL, 03H
    CMP AL, 03H
    JNE WAIT_0     ;检测如果未松开, 就一直停在这里
    MOV BX, SI     ;BX, 要显示的数字
    MOV AL, AH     ;写入原来读入的 AL
    ;不相等代表确实有按键按下
    TEST AL, 02H   ;检测是否是 PC0
    JNZ MODIFY_BUFFER ;相等直接修改 BUFFER
    ADD BX, 8      ;不相等就代表是 PC1, 后 8 位

```

;依据得到的数字 BX, 扫描到的数码管 SI, 修改 Buffer

```

MODIFY_BUFFER :
    ;修改当前扫描到的数码管 Buffer
    MOV DL, Seg2WR
    MOV DH, 00H
    MOV DI, DX
    MOV Buffer[DI], BL ;修改成要显示的数字
    MOV AX, DI
    MOV CL, 8
    INC AX           ;指向下一位
    DIV CL

```

```
XCHG AL, AH          ;取余数
MOV Seg2WR, AL
GetKey ENDP
```

```
WR_AB PROC NEAR
    ;进栈保护
    PUSH BX
    PUSH AX
    PUSH DX
    PUSH CX

    ;A → AL
    MOV BX, OFFSET Buffer
    MOV AL, BX[SI]
    MOV BL, AL
    MOV AL, Table[BX]

    ;写入 A 端口
    MOV DX, PORTA
    OUT DX, AL

    ;B → AL
    MOV AL, 0FEH
    MOV CX, SI
    ROL AL, CL

    ;写入 B 端口
    MOV DX, PORTB
    OUT DX, AL

    ;出栈
    POP CX
    POP DX
    POP AX
    POP BX
    RET
WR_AB ENDP
```



## 八、实验电路图

