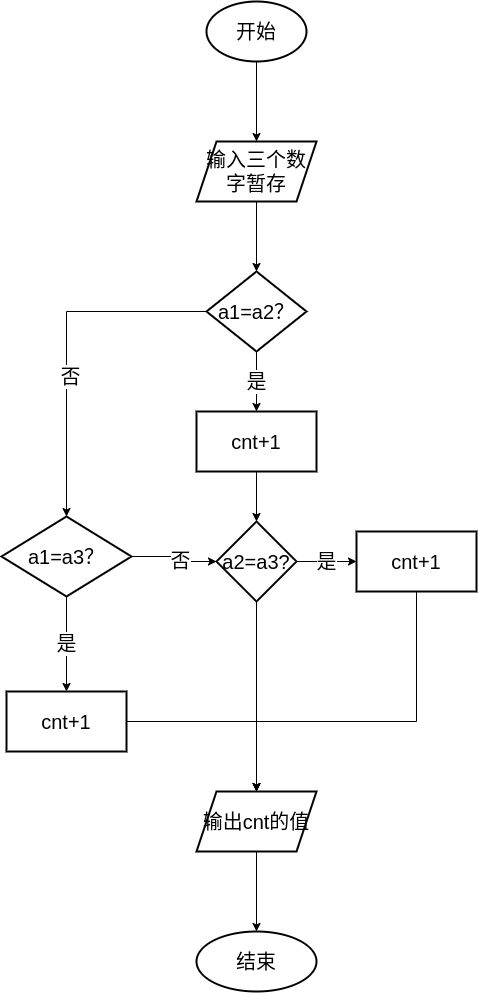
实验一

* + 1. 题目描述：

输入三个数字，判断三个数字中相同的数字个数。

* + 1. 程序流程图：



* + 1. 源代码：

DATAS SEGMENT

D DB 3 DUP(?); 将数据首先保存到D数组

D1 DW ?

D2 DW ?

D3 DW ?

D4 DB 30h

MESSAGE DB 'please input three number(hex，1Byte):',13,10,'$'

;此处输入数据段代码

DATAS ENDS

STACKS SEGMENT

;此处输入堆栈段代码

STACKS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS,SS:STACKS

START:

MOV AX,DATAS

MOV DS,AX

LEA DX,MESSAGE

MOV AH,09H ;dos 9号功能调用输出字符串，以$结尾

INT 21H ; 输出提示

MOV CX, 3H ;控制输入3个数

MOV SI,OFFSET D;利用寄存器间接寻址

;输入一个数据要求为2位的16进制的数，表示一个字节

INPUT: MOV BX ,0; 将输入的数据暂时保留BX,初始化位0

MOV AH,01H ; dos 1号功能调用输入1个字符到AL

INT 21H ; 读入字节高位

CMP Al, 'a' ; 输入的是a..f

JB C1

SUB AL,20H; 变成小写字母

C1: CMP AL,'A' ;

JB C2

SUB AL,7H; 与数字‘9’隔7

C2: SUB AL,30H

MOV BL,AL

SHL BL,1

SHL BL,1

SHL BL,1

SHL BL,1; 循环左移4位，移到高位

INT 21H

CMP Al, 'a' ; 输入的是a..f

JB C3

SUB AL,20H; 变成小写字母

C3: CMP AL, 'A' ;

JB C4

SUB AL,7H; 与数字‘9’隔7

C4: SUB AL,30H

ADD BL,AL

MOV [SI], BL;将数据送往内存D的位置

INC SI ; si+1

MOV AH,02H ; dos 2号功能输出1个字符

MOV DL,13 ;输出字符

INT 21H ; 输出\r使得表示将光标移至行首,dos系统下需要这样做

MOV DL,10

INT 21H ; 输入\n 换行

LOOP INPUT ;循环

MOV SI,OFFSET D

MOV AX,0H; 因为AX16,D8位，用Al,所以ax初始化为0

MOV AL,BYTE PTR [SI]

MOV D1,AX

MOV AL,BYTE PTR [SI+1]

MOV D2,AX

MOV AL,BYTE PTR [SI+2]

MOV D3,AX ; 将数据分别送往D1,D2,D3

MOV AX,D1

CMP AX,D2

JE L1 ; D1，D2相等跳转L1位置

CMP AX,D3

JE L2

MOV AX,D2

CMP AX,D3

JE L2

JMP L

L1: MOV D4,31H

CMP AX,D3

JNE L

MOV D4,32H

JMP L

L2: MOV D4,31H

JMP L

L: MOV AH,02H ; dos 2号功能输出1个字符

MOV DL,D4

INT 21H

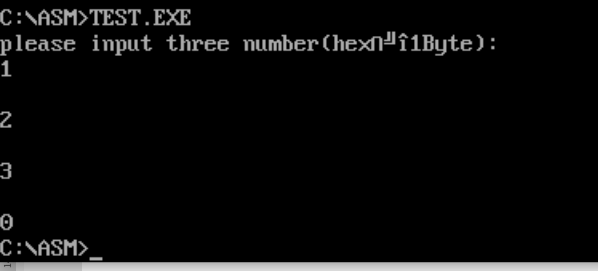
MOV AH,4CH ; 随机输入一个字符，用来暂停程序，观看结果

INT 21H

CODES ENDS

END START

* + 1. 实验结果：



* + 1. 实验中遇到的问题：

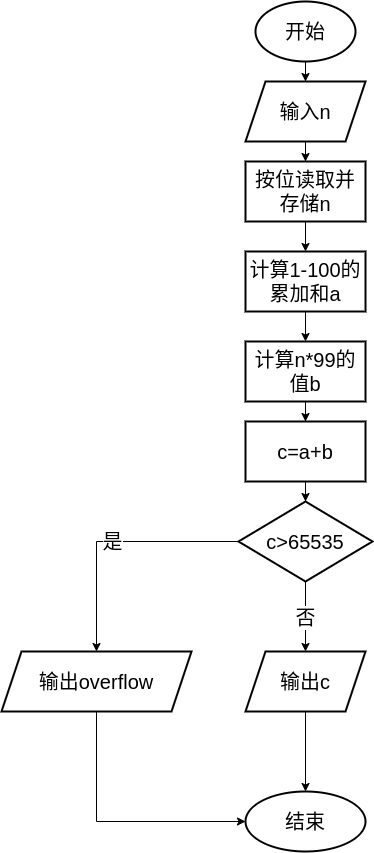
无

实验二

* + 1. 题目描述：

用户给出某个自然数N，计算N+(N+1)+(N+2)+...+(N+99)的累加和。

* + 1. 程序流程图：



* + 1. 源代码：

DATAS SEGMENT

;此处输入数据段代码

;D1 DB 100 DUP(?)

I DB ?

MES DB 'input number n (0<=n<=65535)',13,10,'$'

OV DB 'OVERFLOW',13,10,'$'

DATAS ENDS

STACKS SEGMENT

;此处输入堆栈段代码

STACKS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS,SS:STACKS

START:

MOV AX,DATAS

MOV DS,AX

LEA DX,MES

MOV AH,09H

INT 21H

MOV CX,0

MOV DL,10

FUN1: MOV AH,1

INT 21H

CMP AL,0DH

JE FUN2

SUB AL,30H

AND AX,00FFH ;把高位清空

MOV BX,AX

MOV AX,CX ;CX是之前已输入的高位数字

MUL DL ;AX和DL相乘

ADD AX,BX

CMP AX,605

JNB OVF

MOV CX,AX

CMP BL,0DH

JNZ FUN1

FUN2: MOV BX,CX

MOV CX,100

MOV AX,0

FUN3: ADD AX,CX

LOOP FUN3

PUSH AX

MOV AX,BX

SUB AX,1

MOV DX,0064H

MUL DX ;结果存AX

POP BX

ADD AX,BX

PUSH AX

MOV CX,10

MOV BX,10000

JMP BEG

FUN4: XOR DX,DX

MOV AX,BX

DIV CX

CMP AX,0

JL S

JE S

MOV BX,AX

BEG: XOR DX,DX

POP AX

DIV BX

PUSH DX

ADD AL,30H

MOV DL,AL

MOV AH,2

INT 21H

JMP FUN4

OVF: LEA DX,OV

MOV AH,09H

INT 21H

S:

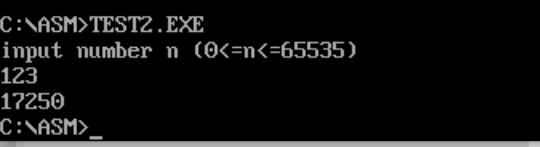
MOV AH,4CH

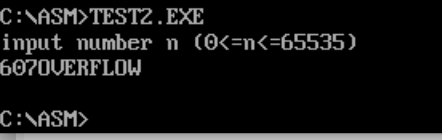
INT 21H

CODES ENDS

END START

* + 1. 实验结果：



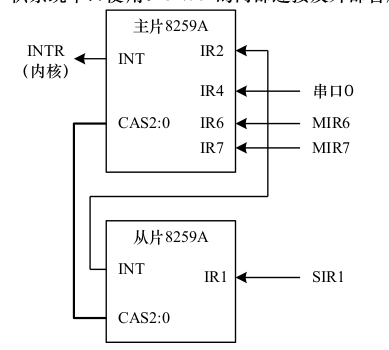


* + 1. 遇到的问题：

无

实验三

1. 题目描述：
   1. 每按动一次单次脉冲，产生一次外部中断，在显示屏上输出一个字符‚7。
   2. KK1＋连接到主片8259 的 IR7 上，KK2＋连接到从片 8259 的 IR1 上，当按一次 KK1＋时，显示屏上显示字符M7，按一次KK2＋时，显示字符S1。
2. 硬件原理图：



1. 源代码：
   1. 单次脉冲实验：

SSTACK SEGMENT STACK

DW 32 DUP(?)

SSTACK ENDS

CODE SEGMENT

ASSUME CS:CODE

START: PUSH DS

MOV AX, 0000H

MOV DS, AX

MOV AX, OFFSET MIR7

MOV SI, 003CH

MOV [SI], AX

MOV AX, CS

MOV SI, 003EH

MOV [SI], AX

CLI

POP DS

MOV AL, 11H

OUT 20H, AL

MOV AL, 08H

OUT 21H, AL

MOV AL, 04H

OUT 21H, AL

MOV AL, 01H

OUT 21H, AL

MOV AL, 6FH

OUT 21H, AL

STI

AA1: MOV AX, 0120H

INT 10H

JMP AA1

MIR7: STI

CALL DELAY

MOV AX, 0138H

INT 10H

MOV AX, 0120H

INT 10H

MOV AL, 20H

OUT 20H, AL

IRET

DELAY: PUSH CX

MOV CX, 0F00H

AA0: PUSH AX

POP AX

LOOP AA0

POP CX

RET

CODE ENDS

END START

* 1. 级联实验：

SSTACK SEGMENT STACK

DW 32 DUP(?)

SSTACK ENDS

DATA SEGMENT

SIGN DB 00H

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START: PUSH DS

MOV AX, 0000H

MOV DS, AX

MOV AX, OFFSET MIR7

MOV SI, 003CH

MOV [SI], AX

MOV AX, CS

MOV SI, 003EH

MOV [SI], AX

MOV AX, OFFSET SIR1

MOV SI, 00C4H

MOV [SI], AX

MOV AX, CS

MOV SI, 00C6H

MOV [SI],AX

CLI

POP DS

MOV AL, 11H

OUT 20H,AL

MOV AL,08H

OUT 21H, AL

MOV AL, 04H

OUT 21H,AL

MOV AL, 11H

OUT 21H, AL

MOV AL, 11H

OUT 0A0H, AL

MOV AL, 30H

OUT 0A1H, AL

MOV AL, 02H

OUT 0A1H, AL

MOV AL, 01H

OUT 0A1H, AL

MOV AL, 0FDH

OUT 0A1H, AL

MOV AL, 6BH

OUT 21H, AL

STI

AA1: MOV AX, 0120H

INT 10H

JMP AA1

MIR7: STI

MOV SIGN, 01H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

MOV AX, 014DH

INT 10H

CALL DELAY

CALL DELAY

CALL DELAY

CALL DELAY

CALL DELAY

CALL DELAY

CALL DELAY

MOV AX, 014DH

MOV SIGN, 00H

INT 10H

MOV AL, 20H

OUT 20H, AL

IRET

SIR1:

MOV AX, 0153H

INT 10H

CALL DELAY

CALL DELAY

CALL DELAY

CALL DELAY

CMP SIGN, 01H

JNZ M

JMP BACK

M: MOV AX, 0153H

INT 10H

MOV AL, 20H

OUT 0A0H, AL

BACK: OUT 20H, AL

IRET

DELAY: PUSH CX

MOV CX, 0F00H

AA0: PUSH AX

POP AX

LOOP AA0

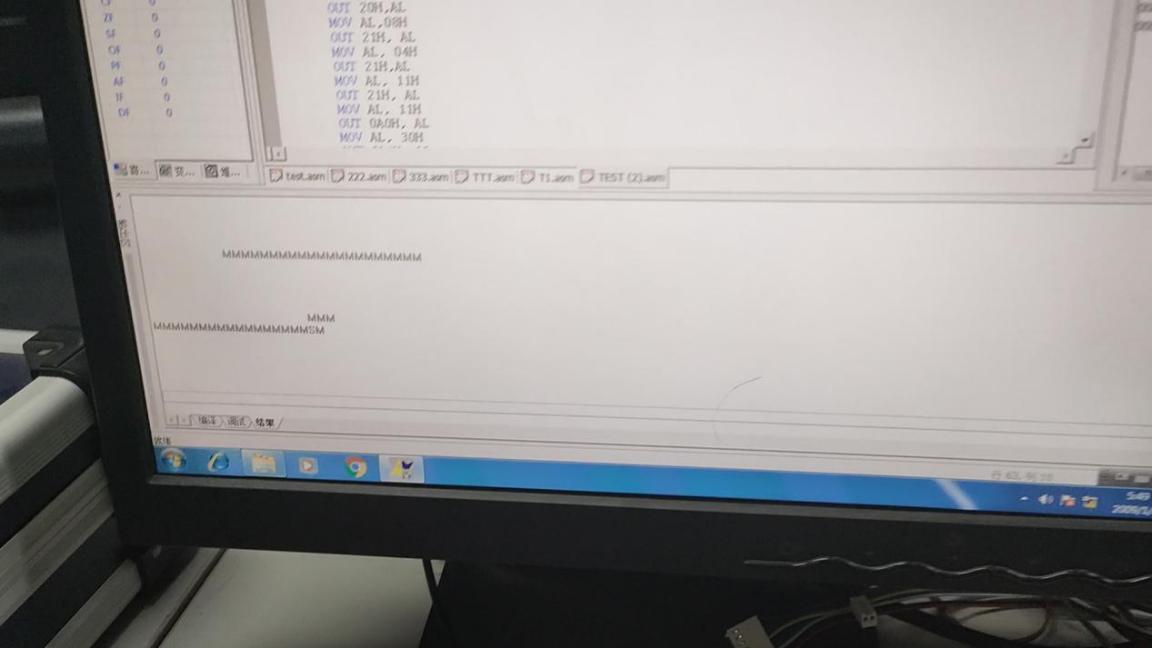
POP CX

RET

CODE ENDS

END START

1. 运行结果：

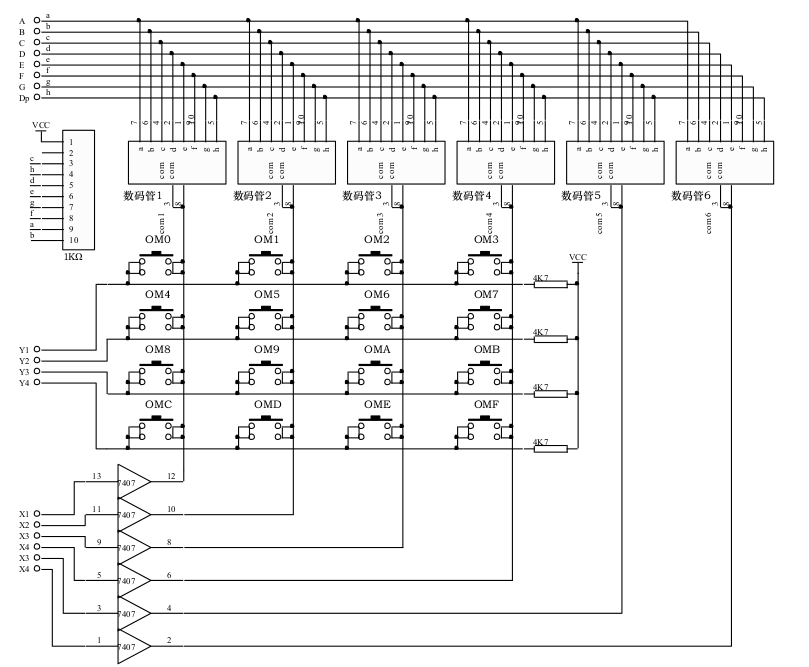


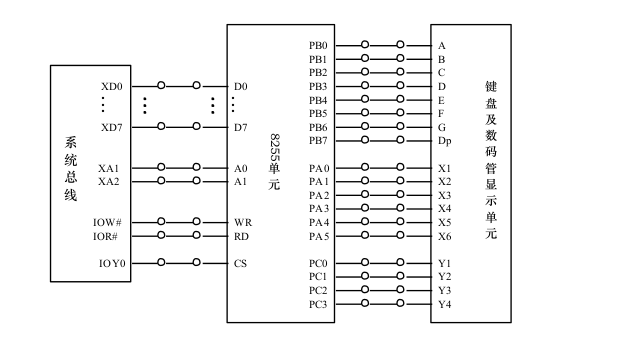
1. 出现的问题：

无。

实验四

1. 题目描述：
   1. 将键盘进行编号，记作 0～F，当按下其中一个按键时，将该按键对应的编号在一个数码管上显示出来，当再按下一个按键时，便将这个按键的编号在下一个数码管上显示出来，数码管上可以显示最近 4 次按下的按键编号。
   2. 输出同组两个人的姓名首字母+序号
2. 硬件原理图：





1. 源代码：
   1. 显示按键编号：

MY8255\_A EQU 0600H

MY8255\_B EQU 0602H

MY8255\_C EQU 0604H

MY8255\_CON EQU 0606H

SSTACK SEGMENT STACK

DW 16 DUP(?)

SSTACK ENDS

DATA SEGMENT

DTABLE DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H

DB 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START: MOV AX, DATA

MOV DS, AX

MOV SI, 3000H

MOV AL, 00H

MOV [SI],AL

MOV [SI+1],AL

MOV [SI+2],AL

MOV [SI+3],AL

MOV [SI+4],AL

MOV [SI+5],AL

MOV DI, 3005H

MOV DX, MY8255\_CON

MOV AL,81H

OUT DX,AL

BEGIN: CALL DIS

CALL CLEAR

CALL CCSCAN

JNZ INK1

JMP BEGIN

INK1: CALL DIS

CALL DALLY

CALL CLEAR

CALL CCSCAN

JNZ INK2

JMP BEGIN

INK2: MOV CH,0FEH

MOV CL,00H

COLUM: MOV AL, CH

MOV DX, MY8255\_A

OUT DX, AL

MOV DX, MY8255\_C

IN AL,DX

L1: TEST AL, 01H

JNZ L2

MOV AL, 00H

JMP KCODE

L2: TEST AL,02H

JNZ L3

MOV AL, 04H

JMP KCODE

L3: TEST AL, 04H

JNZ L4

MOV AL, 08H

JMP KCODE

L4: TEST AL, 08H

JNZ NEXT

MOV AL, 0CH

KCODE: ADD AL,CL

CALL PUTBUF

PUSH AX

KON: CALL DIS

CALL CLEAR

CALL CCSCAN

JNZ KON

POP AX

NEXT: INC CL

MOV AL, CH

TEST AL, 08H

JZ KERR

ROL AL, 1

MOV CH, AL

JMP COLUM

KERR: JMP BEGIN

CCSCAN: MOV AL, 00H

MOV DX, MY8255\_A

OUT DX, AL

MOV DX, MY8255\_C

IN AL, DX

NOT AL

AND AL, 0FH

RET

CLEAR: MOV DX, MY8255\_B

MOV AL, 00H

OUT DX, AL

RET

DIS: PUSH AX

MOV SI, 3000H

MOV DL, 0DFH

MOV AL, DL

AGAIN: PUSH DX

MOV DX, MY8255\_A

OUT DX, AL

MOV AL, [SI]

MOV BX, OFFSET DTABLE

AND AX, 00FFH

ADD BX, AX

MOV AL, [BX]

MOV DX, MY8255\_B

OUT DX, AL

CALL DALLY

INC SI

POP DX

MOV AL, DL

TEST AL, 01H

JZ OUT1

ROR AL, 1

MOV DL, AL

JMP AGAIN

OUT1: POP AX

RET

DALLY: PUSH CX

MOV CX, 0006H

T1: MOV AX, 009FH

T2: DEC AX

JNZ T2

LOOP T1

POP CX

RET

PUTBUF: MOV SI, DI

MOV [SI], AL

DEC DI

CMP DI, 2FFFH

JNZ GOBACK

MOV DI, 3005H

GOBACK: RET

CODE ENDS

END START

* 1. 显示同组人首字母+序号（不清空缓存）

MY8255\_A EQU 0600H

MY8255\_B EQU 0602H

MY8255\_C EQU 0604H

MY8255\_CON EQU 0606H

SSTACK SEGMENT STACK

DW 16 DUP(?)

SSTACK ENDS

DATA SEGMENT

DTABLE DB 6DH, 3FH, 00H, 00H, 39H, 06H, 00H, 00H

DB 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START: MOV AX, DATA

MOV DS, AX

MOV SI, 3000H

MOV AL, 00H

MOV [SI],AL

MOV [SI+1],AL

MOV [SI+2],AL

MOV [SI+3],AL

MOV [SI+4],AL

MOV [SI+5],AL

MOV DI, 3005H

MOV DX, MY8255\_CON

MOV AL,81H

OUT DX,AL

BEGIN: CALL DIS

CALL CLEAR

CALL CCSCAN

JNZ INK1

JMP BEGIN

INK1: CALL DIS

CALL DALLY

CALL CLEAR

CALL CCSCAN

JNZ INK2

JMP BEGIN

INK2: MOV CH,0FEH

MOV CL,00H

COLUM: MOV AL, CH

MOV DX, MY8255\_A

OUT DX, AL

MOV DX, MY8255\_C

IN AL,DX

L1: TEST AL, 01H

JNZ L2

MOV AL, 00H

JMP KCODE

L2: TEST AL,02H

; JNZ L3

JNZ NEXT

MOV AL, 04H

; JMP KCODE

;L3: TEST AL, 04H

; JNZ L4

; MOV AL, 08H

; JMP KCODE

;L4: TEST AL, 08H

;MOV AL, 0CH

KCODE: ADD AL,CL

CALL PUTBUF

MOV CL, 01H

ADD AL,CL

CALL PUTBUF

MOV CL, 01H

ADD AL, CL

CALL PUTBUF

MOV CL, 00H

ADD AL, CL

CALL PUTBUF

MOV CL, 00H

ADD AL, CL

CALL PUTBUF

MOV CL, 00H

ADD AL, CL

CALL PUTBUF

PUSH AX

KON: CALL DIS

CALL CLEAR

CALL CCSCAN

JNZ KON

POP AX

NEXT: INC CL

MOV AL, CH

TEST AL, 08H

JZ KERR

ROL AL, 1

MOV CH, AL

JMP COLUM

KERR: JMP BEGIN

CCSCAN: MOV AL, 00H

MOV DX, MY8255\_A

OUT DX, AL

MOV DX, MY8255\_C

IN AL, DX

NOT AL

AND AL, 0FH

RET

CLEAR: MOV DX, MY8255\_B

MOV AL, 00H

OUT DX, AL

RET

DIS: PUSH AX

MOV SI, 3000H

MOV DL, 0DFH

MOV AL, DL

AGAIN: PUSH DX

MOV DX, MY8255\_A

OUT DX, AL

MOV AL, [SI]

MOV BX, OFFSET DTABLE

AND AX, 00FFH

ADD BX, AX

MOV AL, [BX]

MOV DX, MY8255\_B

OUT DX, AL

CALL DALLY

INC SI

POP DX

MOV AL, DL

TEST AL, 01H

JZ OUT1

ROR AL, 1

MOV DL, AL

JMP AGAIN

OUT1: POP AX

RET

DALLY: PUSH CX

MOV CX, 0006H

T1: MOV AX, 009FH

T2: DEC AX

JNZ T2

LOOP T1

POP CX

RET

PUTBUF: MOV SI, DI

MOV [SI], AL

DEC DI

CMP DI, 2FFFH

JNZ GOBACK

MOV DI, 3005H

GOBACK: RET

CODE ENDS

END START

* 1. 显示同组人首字母+序号（清空缓存）

MY8255\_A EQU 0600H

MY8255\_B EQU 0602H

MY8255\_C EQU 0604H

MY8255\_CON EQU 0606H

SSTACK SEGMENT STACK

DW 16 DUP(?)

SSTACK ENDS

DATA SEGMENT

DTABLE DB 6DH, 3FH,39H, 06H, 00H, 00H , 00H, 00H

DB 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START: MOV AX, DATA

MOV DS, AX

MOV SI, 3000H

MOV AL, 00H

MOV [SI],AL

MOV [SI+1],AL

MOV [SI+2],AL

MOV [SI+3],AL

MOV [SI+4],AL

MOV [SI+5],AL

MOV DI, 3005H

MOV DX, MY8255\_CON

MOV AL,81H

OUT DX,AL

BEGIN: CALL DIS

CALL CLEAR

CALL CCSCAN

JNZ INK1

JMP BEGIN

INK1: CALL DIS

CALL DALLY

CALL CLEAR

CALL CCSCAN

JNZ INK2

JMP BEGIN

INK2: MOV CH,0FEH

MOV CL,00H

COLUM: MOV AL, CH

MOV DX, MY8255\_A

OUT DX, AL

MOV DX, MY8255\_C

IN AL,DX

L2: TEST AL,01H

; JNZ L3

JNZ NEXT

MOV AL, 00H

; JMP KCODE

;L3: TEST AL, 04H

; JNZ L4

; MOV AL, 08H

; JMP KCODE

;L4: TEST AL, 08H

;MOV AL, 0CH

KCODE: ADD AL,CL

CALL PUTBUF

MOV CL, 01H

ADD AL,CL

CALL PUTBUF

PUSH AX

KON: CALL DIS

CALL CLEAR

CALL CCSCAN

JNZ KON

POP AX

NEXT: INC CL

MOV AL, CH

TEST AL, 08H

JZ KERR

ROL AL, 1

MOV CH, AL

JMP COLUM

KERR: JMP BEGIN

CCSCAN: MOV AL, 00H

MOV DX, MY8255\_A

OUT DX, AL

MOV DX, MY8255\_C

IN AL, DX

NOT AL

AND AL, 0FH

RET

CLEAR: MOV DX, MY8255\_B

MOV AL, 00H

OUT DX, AL

RET

DIS: PUSH AX

MOV SI, 3000H

MOV DL, 0DFH

MOV AL, DL

AGAIN: PUSH DX

MOV DX, MY8255\_A

OUT DX, AL

MOV AL, [SI]

MOV BX, OFFSET DTABLE

AND AX, 00FFH

ADD BX, AX

MOV AL, [BX]

MOV DX, MY8255\_B

OUT DX, AL

CALL DALLY

INC SI

POP DX

MOV AL, DL

TEST AL, 01H

JZ OUT1

ROR AL, 1

MOV DL, AL

JMP AGAIN

OUT1: POP AX

RET

DALLY: PUSH CX

MOV CX, 0006H

T1: MOV AX, 009FH

T2: DEC AX

JNZ T2

LOOP T1

POP CX

RET

PUTBUF: MOV SI, DI

MOV [SI], AL

DEC DI

CMP DI, 2FFFH

JNZ GOBACK

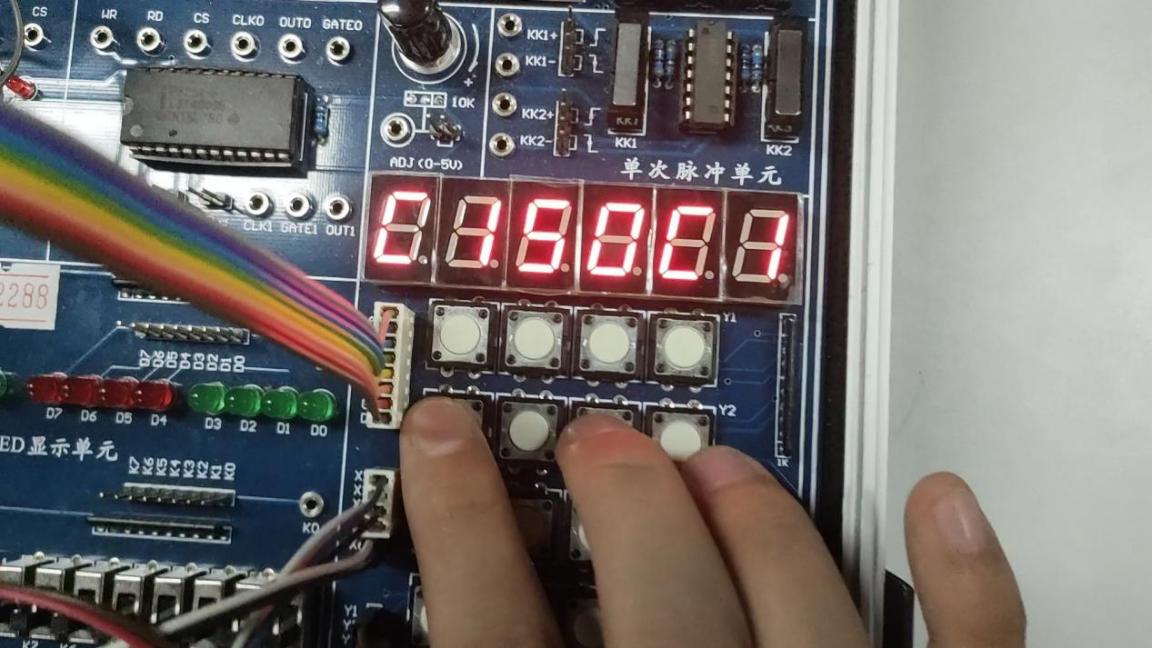
MOV DI, 3005H

GOBACK: RET

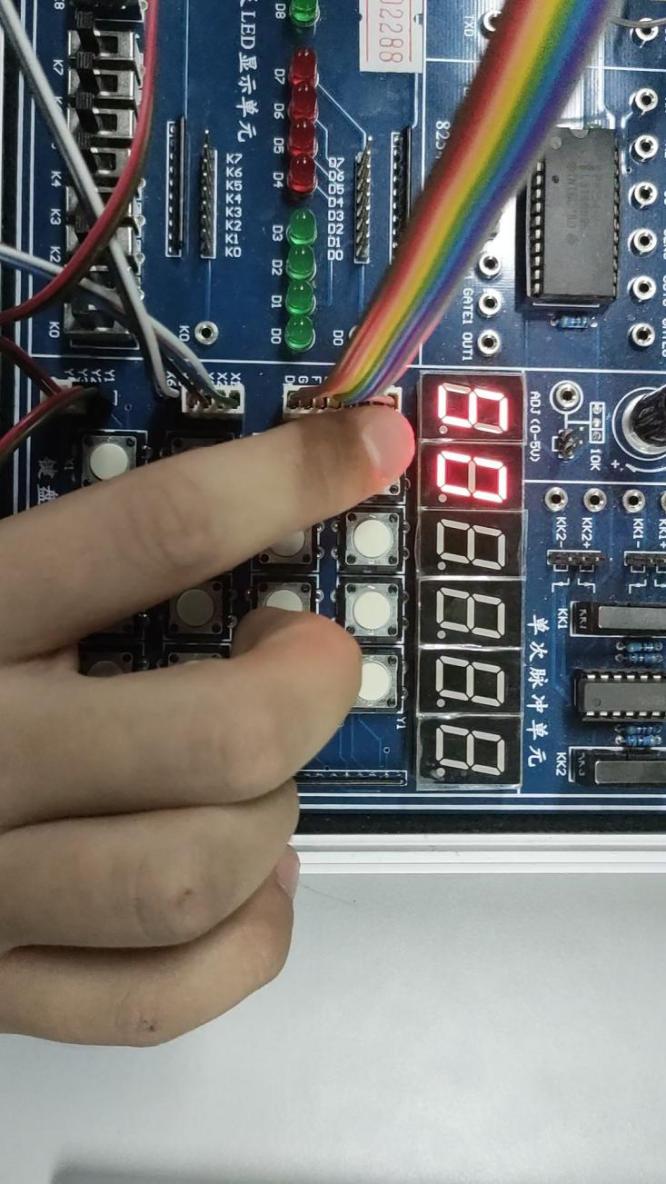
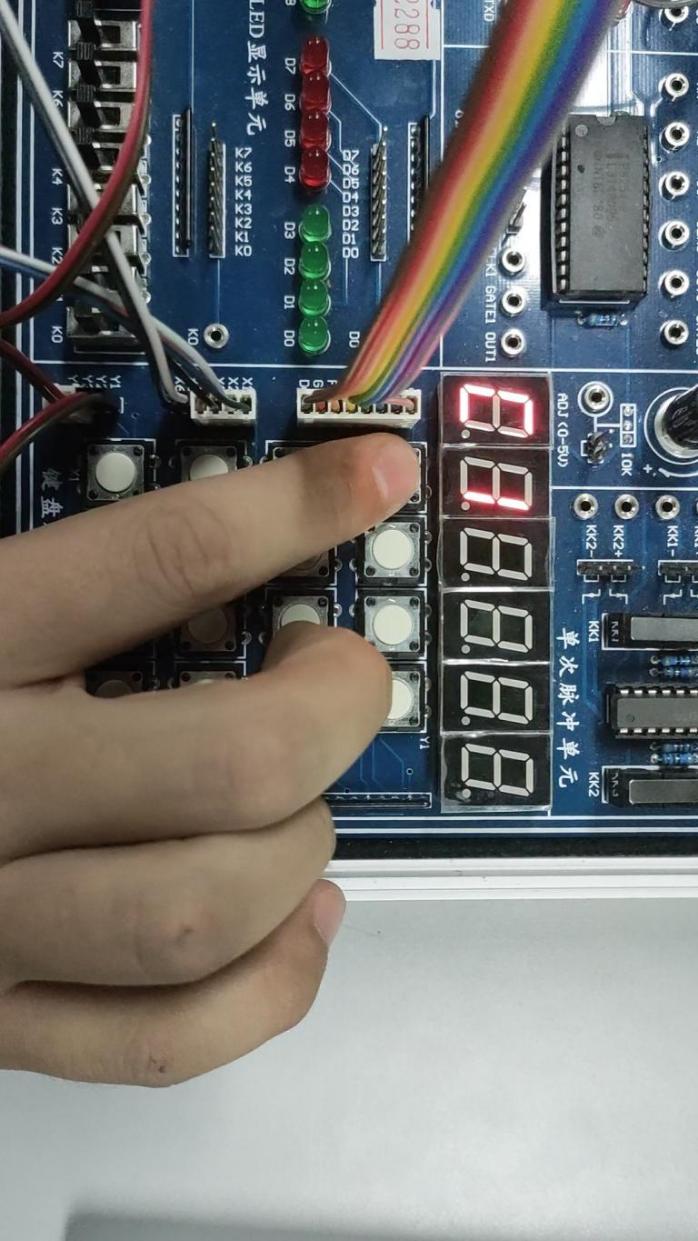
CODE ENDS

END START

1. 实验结果
   1. 显示同组人首字母+序号（不清空）



* 1. 显示同组人首字母+序号（清空）



1. 实验中的问题：

无。