

**基于汇编语言的篮球计分器**

**所属课程：** 8086 微机原理及接口技术

**班级：** 2017 级通信工程6A班

**姓名：** 何尔恒

**学号：** 20172333090

**日期：** 2019/6/14

**指导老师：** 李萍

目录

[一、实验目的 3](#_Toc11435925)

[二、设计要求 3](#_Toc11435926)

[三、系统总览 3](#_Toc11435927)

[四、实验所用模块及连线 4](#_Toc11435928)

[1. 8254 定时/计数器 4](#_Toc11435929)

[a. 8254 的基本特点 4](#_Toc11435930)

[b 8254的内部结构及芯片的引脚定义 4](#_Toc11435931)

[c 芯片的连线 4](#_Toc11435932)

[2. 8259 中断控制器 5](#_Toc11435934)

[a. 8259 的基本特点 5](#_Toc11435935)

[3. 8255 并行输入/输出接口芯片 5](#_Toc11435936)

[a. 8255 的基本特点 5](#_Toc11435937)

[b．8255 内部结构及芯片的引脚定义 5](#_Toc11435938)

[c. 芯片实现功能 5](#_Toc11435939)

[4. 矩阵键盘 6](#_Toc11435940)

[5. 数码管及LED 灯显示 8](#_Toc11435941)

[6.8254发声单元 9](#_Toc11435942)

[7.IR7 中断服务程序 10](#_Toc11435945)

[五、汇编源程序 11](#_Toc11435947)

[六、调试中出现的问题 17](#_Toc11435948)

[1. 按键跳转只能进入一次 17](#_Toc11435949)

[2. 数码管显示不正常 17](#_Toc11435950)

[3. 调试时想知道子程序中的某些程序是否被执行 18](#_Toc11435951)

[七、分析与总结 18](#_Toc11435952)

# 一、实验目的

综合所学微机原理与接口技术相关软件、硬件知识，并应用基础实验所获得的实验设计技能，独立设计解决实际应用问题的系统。

# 二、设计要求

设计一个篮球竞赛用的电子显示屏，要求如下功能：

１、每个队的分数至少用两位数码管来显示，得分有 1 分、2 分和 3 分三种情况，用按键输入；

２、换场功能：当比赛队伍交换场地时，显示的得分也要交换。

３、设计一个 24 秒倒计时电路，并具有时间显示功能，时间间隔为 1 秒；

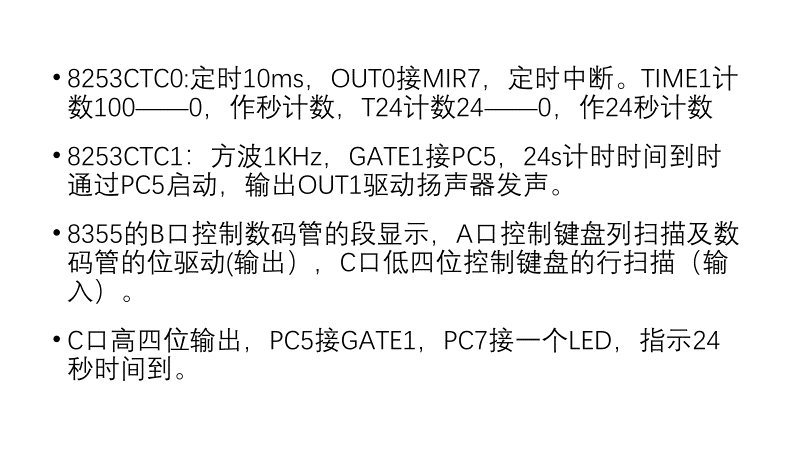
４、设置启动键和暂停/继续键，控制计时器直接启动计时，暂停/继续计时功能；

５、设置复位键，按复位键可随时返回到初始状态，即计时器返回到 24；

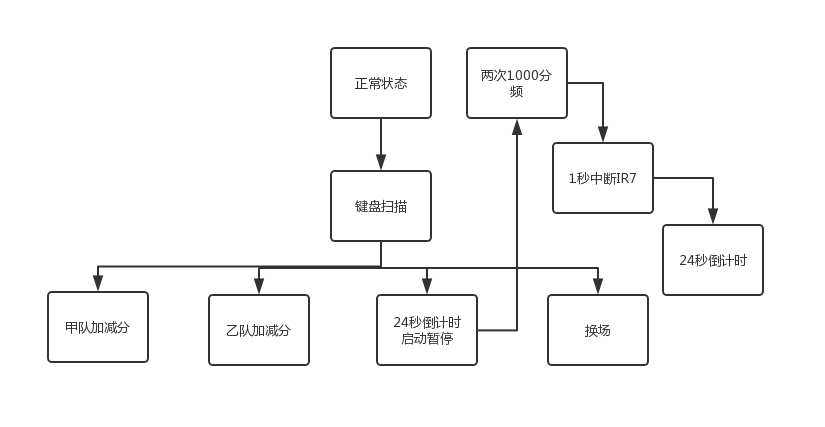
６、计时器递减计时到“00”时，计时器跳回到“24”停止工作，并给出声音和发光提示；

# 三、系统总览

该计分系统采用不间断的键盘扫面显示例程，同时配合两次1000分频得到的1秒中断IR7从而实现响应功能。系统不断扫描键盘按键的键位，当有相应键盘键位按下之后与相应位的缓存值进行逻辑处理，再通过显示程序显示出来。



以下是整个系统流程的总览图：



# 四、实验所用模块及连线

## 8254 定时/计数器

### 8254 的基本特点

1. 有 3 个独立的 16 位计数器。
2. 每个计数器可编程工作于 6 种不同工作方式。

（3）8254 每个计数器允许的最高计数频率为 10MHz。

（4）计数脉冲可以是有规律的时钟信号，也可以是随机信号。

### b 8254的内部结构及芯片的引脚定义

### c 芯片的连线

### 

## 8259 中断控制器

### 8259 的基本特点

一片 8259A 有 8 条外界中断请求线 IR0～IR7，每一条请求线有一个相应的触发器来保存请求信号，从而形成了中断请求寄存器 IRR（Interrupt Request Register）。正在服务的中断，由中断服务寄存器 ISR（IN Service Register）保存。

b．8259 实现的功能

在主程序中，只有一个死循环，同时不停等待 1s 一次的IR7中断。

## 8255 并行输入/输出接口芯片

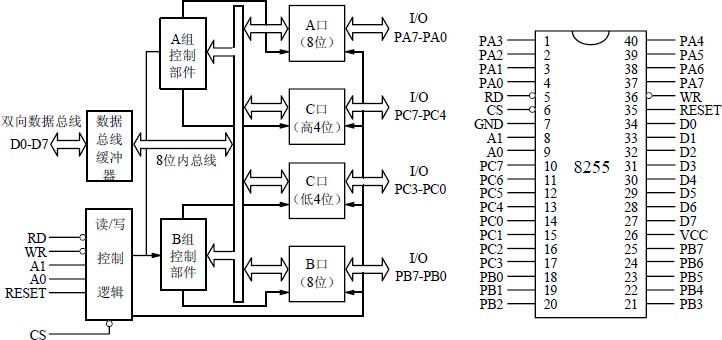
### 8255 的基本特点

8255 可编程外围接口芯片是 Intel 公司生产的通用并行 I/O 接口芯片，它具有 A、

B、C 三个并行接口，用+5V 单电源供电，能在以下三种方式下工作：方式 0--基本输入/

输出方式、方式 1--选通输入/输出方式、方式 2--双向选通工作方式。

### b．8255 内部结构及芯片的引脚定义



### c. 芯片实现功能

本次综合设计题目中需要用到 8255 的并行输入、输出功能实现两队分数显示、24秒倒计时的显示，以及扫描按键实现两队加减分、24秒暂停重置开始、两队分数交换控制。

#### 功能规划：

A B C 口 方式 0

A 口：低 6 位输出 LED 数码管位选（6 个数码管哪个显示）

低 4 位输出键盘列扫描信号

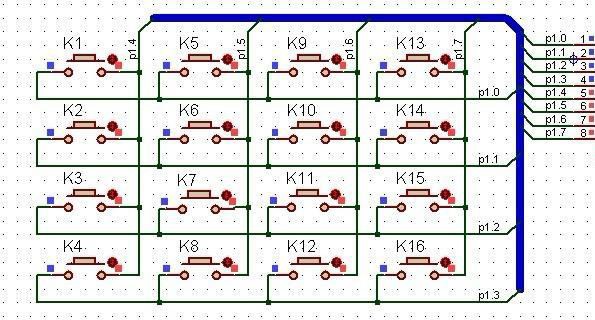
B 口：输出 LED 数码管段选（数码管显示什么数字）

C 口：低 4 位输入键盘行信号

## 矩阵键盘

* 1. 键盘内部构造

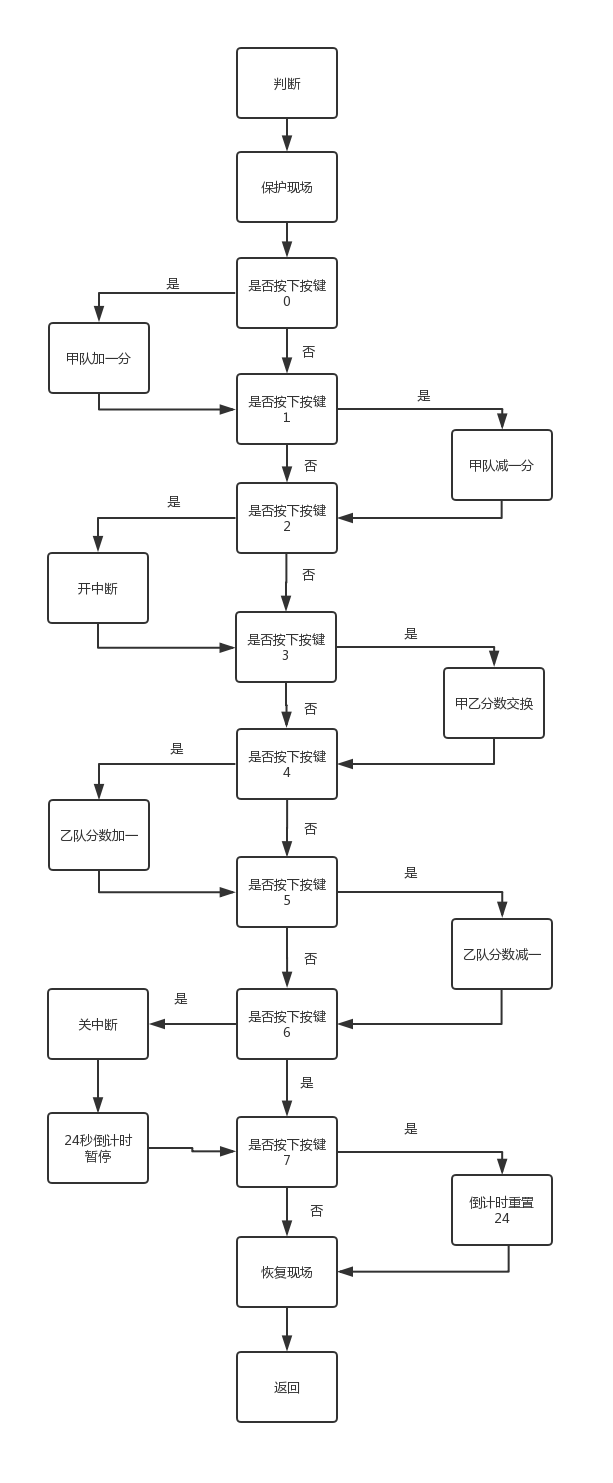
**列扫描信号（输出）**



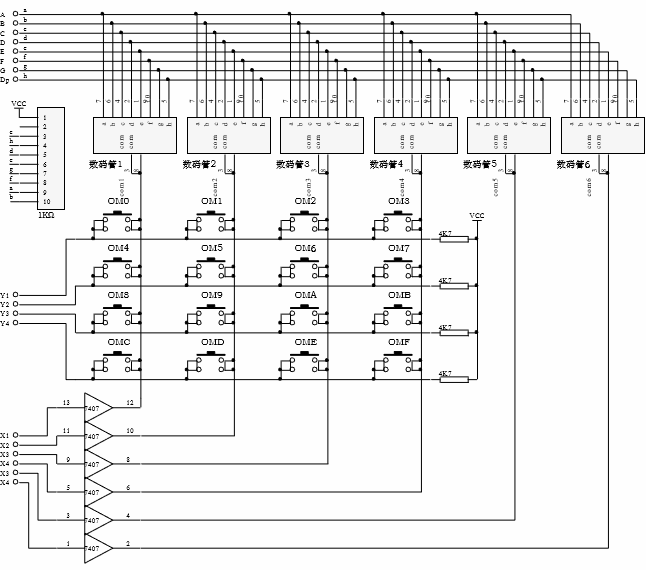
**行扫描信号（输入）**

* 1. 键盘扫描的原理

通过输出列扫描信号，然后读取行信号的信号作为输入，如果判断有按键按下，那么就逐行输出列扫描信号，判断是具体哪一个按键被按下了。

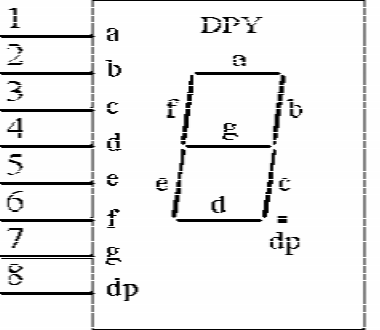


* 1. 实验板键盘及数码管模块原理图

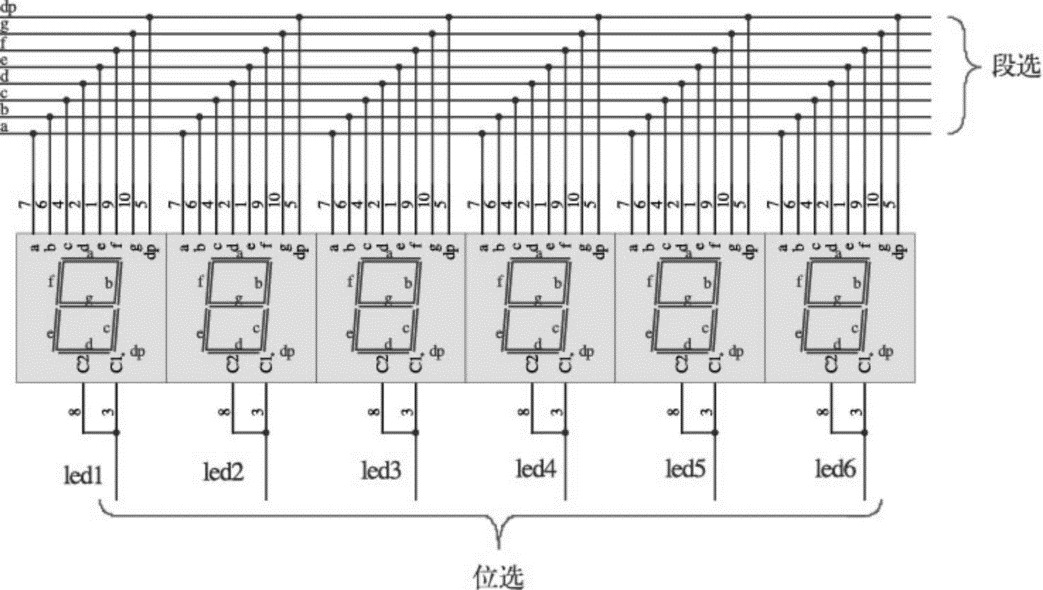


## 数码管及LED 灯显示

* 1. 数码管内部构造

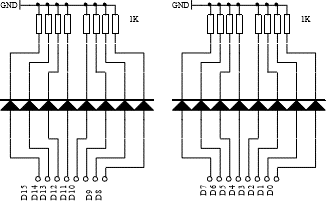


**单个数码管的示意图**



**六个数码管的连接示意图**

* 1. LED 灯原理电路



* 1. 数码管扫描的原理

往段选信号中输入需要显示的数字（通过查表指令），然后通过往位选信号信号中输入需要显示的数码管的位置（逻辑 0 是选中），然后以 20ms 的扫描周期，就能实现数码管的显示。

## 6.8254发声单元

## 

## 

## 7.IR7 中断服务程序

## 

# 五、汇编源程序

;=======================================================

; 文件名: KeyLed.asm

; 功能描述: 键盘及数码管显示实验，通过8255控制。

; 8255的B口控制数码管的段显示，A口控制键盘列扫描

; 及数码管的位驱动，C口控制键盘的行扫描。

; 按下按键，该按键对应的位置将按顺序显示在数码管上。

;=======================================================

MY8255\_A EQU 0600H

MY8255\_B EQU 0602H

MY8255\_C EQU 0604H

MY8255\_CON        EQU 0606H

A8254 EQU 06C0H

B8254 EQU 06C2H

C8254 EQU 06C4H

CON8254 EQU 06C6H

SSTACK        SEGMENT STACK

DW 32 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, 04H

MOV [SI+2],AL

MOV AL, 02H

MOV [SI+3],AL

MOV AL,00H

MOV [SI],AL

MOV [SI+1],AL

MOV [SI+4],AL

MOV [SI+5],AL

MOV DI,2000H

MOV [DI],AL

MOV DI,3005H

MOV DX,MY8255\_CON                ;写8255控制字

MOV AL,81H

OUT DX,AL

PUSH DS

MOV AX, 0000H

MOV DS, AX

MOV AX, OFFSET IRQ7                ;取中断入口地址

MOV SI, 003CH                        ;中断矢量地址

MOV [SI], AX                        ;填IRQ7的偏移矢量

MOV AX, CS                                ;段地址

MOV SI, 003EH

MOV [SI], AX                        ;填IRQ7的段地址矢量

CLI

POP DS

;初始化主片8259

MOV AL, 11H

OUT 20H, AL                                ;ICW1

MOV AL, 08H

OUT 21H, AL                                ;ICW2

MOV AL, 04H

OUT 21H, AL                                ;ICW3

MOV AL, 01H

OUT 21H, AL                                ;ICW4

MOV AL, 0EFH                                ;OCW1

OUT 21H, AL

;8254

MOV DX, CON8254

MOV AL, 34H                                ;计数器0，方式2

OUT DX, AL

MOV DX, A8254

MOV AL, 0E8H

OUT DX, AL

MOV DX, A8254

MOV AL, 03H

OUT DX, AL

MOV BP,1000

MOV DX, CON8254

MOV AL, 01110111B

OUT DX, AL

MOV DX, B8254

MOV AL,00

OUT DX,AL

MOV AL,10H

OUT DX,AL

STI

BEGIN: CALL DIS                                ;调用显示子程序

CALL CLEAR                                ;清屏

CALL CCSCAN                                ;扫描

JNZ INK1

JMP BEGIN

INK1: CALL DIS

CALL DALLY

CALL DALLY

CALL CLEAR

CALL CCSCAN

JNZ INK2                                ;有键按下，转到INK2

JMP BEGIN

;========================================

;确定按下键的位置

;========================================

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 ;is L1?

JNZ L2

MOV AL,00H ;L1

JMP KCODE

L2: TEST AL,02H ;is L2?

JNZ L3

MOV AL,04H ;L2

JMP KCODE

L3: TEST AL,04H ;is L3?

JNZ L4

MOV AL,08H ;L3

JMP KCODE

L4: TEST AL,08H ;is L4?

JNZ NEXT

MOV AL,0CH ;L4

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

;========================================

;键盘扫描子程序

;========================================

CCSCAN: MOV AL,00H

MOV DX,MY8255\_A

OUT DX,AL

MOV DX,MY8255\_C

IN AL,DX

NOT AL

AND AL,0FH

RET

;========================================

;清屏子程序

;========================================

CLEAR: MOV DX,MY8255\_B

MOV AL,00H

OUT DX,AL

RET

;========================================

;显示子程序

;========================================

DIS: PUSH AX

MOV SI,3000H

MOV DL,0DFH

MOV AL,DL

AGAIN: PUSH DX

PUSH BX

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 BX

POP DX

MOV AL,DL

TEST AL,01H

JZ OUT1

ROR AL,1

MOV DL,AL

JMP AGAIN

OUT1: POP AX

RET

;====== 延时子程序 ======

DALLY: PUSH CX

MOV CX,0006H

T1: MOV AX,009FH

T2: DEC AX

JNZ T2

LOOP T1

POP CX

RET

;========================================

;存键盘值到相应位的缓冲中

;========================================

PUTBUF:

CALL PANDUAN

GOBACK: RET

PANDUAN:PUSH AX

CMP AL, 00H

JZ JIA

CMP AL, 01H

JZ JIAN

CMP AL, 02H

JZ KAIZHONGDUAN

CMP AL, 03H

JZ HUANCHANG

CMP AL, 04H

JZ JIA1

CMP AL, 05H

JZ JIAN1

CMP AL, 06H

JZ ZANTING

CMP AL, 07H

JZ CHONGZHI

RET

JIA:         MOV SI,3000H ；甲队加分

MOV AH,[SI+5]

MOV AL,[SI+4]

ADD AL,01H

AAA

MOV [SI+5],AH

MOV [SI+4],AL

POP AX

RET

JIA1:         MOV SI,3000H ；乙队加分

MOV AH,[SI+1]

MOV AL,[SI]

ADD AL,01H

AAA

MOV [SI+1],AH

MOV [SI],AL

POP AX

RET

JIAN: MOV SI,3000H ；甲队减分

MOV AH,[SI+5]

MOV AL,[SI+4]

SUB AL,01H

AAS

MOV [SI+5],AH

MOV [SI+4],AL

POP AX

RET

JIAN1: MOV SI,3000H ；乙队减分

MOV AH,[SI+1]

MOV AL,[SI]

SUB AL,01H

AAS

MOV [SI+1],AH

MOV [SI],AL

POP AX

RET

KAIZHONGDUAN: MOV AL, 6FH                                ;OCW1

OUT 21H, AL

POP AX

RET

HUANCHANG:MOV SI,3000H

PUSH BX

MOV BL,[SI]

XCHG BL,[SI+4]

MOV [SI],BL

POP BX

MOV AL,[SI+1]

XCHG AL,[SI+5]

MOV [SI+1],AL

POP AX

RET

ZANTING: MOV AL, 0EFH

OUT 21H, AL

POP AX

RET

CHONGZHI:MOV SI,3000H

MOV AL, 04H

MOV [SI+2],AL

MOV AL, 02H

MOV [SI+3],AL

MOV AL,00H ;C口高4位输出0000，蜂鸣器停，灯灭

MOV DX,MY8255\_C

OUT DX,AL

POP AX

RET

JIANBACK: RET

IRQ7:        PUSH AX

DEC BP

JNZ BACKIRQ

MOV BP, 1000

MOV SI,3000H

MOV AL,[SI+2]

MOV AH,[SI+3]

SUB AL,01H

AAS

CMP AX,0000H ；减到0？

JNZ CC

MOV AL, 0EFH

OUT 21H, AL

MOV AL,0F0H ;C口高4位输出1111，蜂鸣器响，灯亮

MOV DX,MY8255\_C

OUT DX,AL

MOV AX,0000H

CC: MOV [SI+2],AL

MOV [SI+3],AH

BACKIRQ:POP AX

MOV AL, 20H

OUT 20H, AL                                ;中断结束命令

IRET

CODE        ENDS

END START

# 

# 六、调试中出现的问题

## 按键跳转只能进入一次

**解决方法：**既然能跳转，就不是跳转判断的问题。在经历了很长的时间后才发现跳转了一次之后进入的子程序没有加 RET 返回，所以只能进入一次。得到一个经验就是，要学会从调试中锁定错误出现的地方，可以通过逐行注释掉的方法去查找错误。

## 数码管显示不正常

实验箱提供的可使用的寄存器较少，在不同的子程序中调用时不做好保护现场很容易导致子程序中对相关寄存器的赋值影响主程序特别是显示子程序的正确性，所以要特别留心。

**解决方法：**观察调试软件左边的寄存器使用情况及右边的相关地址中的数值变换情况，选用空余寄存器和做好保护现场。

## 调试时想知道子程序中的某些程序是否被执行

在一些逻辑判断时，通常会需要知道某一个程序是否被执行，但是在调试的现象当中看不出来。

**解决方法：** 使用断点的方法，在子程序里的某一个语句当中，制造一个断点， 然后执行，如果没有执行到这句话，那么屏幕断点所在的一行就不会高亮。有执行到则相反。或者对相关程序进行暂时屏蔽处理，执行处理过后的程序并观察相关结果。

# 七、分析与总结

持续时间长达数周的微机原理综合设计终于告一段落，从一开始的查找资料， 小组讨论，小组实验报告，编程，调试，到现在完成个人的实验报告。小组讨论实现方法，思维很开放，思路很多样，也让我了解到了许多能实现同一功能的不同方法。在撰写小组实验报告的同时，需要我们对于芯片十分了解，包括控制字和 芯片能实现的功能及连线。

接下来的一周，便是编程和硬件调试及实现。编程的过程需要保持头脑清醒，首先要确保总体逻辑是正确的。但实际编写当中难免会出现一些隐蔽的逻辑问题，这种问题很难避免，但我们可以通过规范的编写习惯，让我们在调试当中更容易发现这些隐蔽的错误。

在调试过程中，我深深地体会到编写代码之后马上上机实验是十分重要的。有某些程序代码上是正确的，但是上机实验之后不能得出应有的结果，让我花了好几个小时的调试时间，十分的不值得。发现原因竟然是一个不起眼的子程序中缺少了返回指令，导致程序一直卡在死循环，实在是很不应该。

总的来说，这次综设还是完成得比较顺利。最重要的是，我学到了很多很实用的东西，无疑推动我在专业中更好地进步，这门课程许多的知识点并不只是对于8086才有用的，现代的许多CPU都是在 8086 的基础上发展起来的，很多方面都是互通的，希望我在未来的学习中能把《微机原理》的相关知识运用上。