

華東理工大學  
EAST CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY

# 《 微机原理及接口技术 》 实验报告本

系 别： 计算机科学与工程  
班 级：                                   
学 号： 2101\*\*\*\*  
姓 名： private  
指导教师： 冷春霞

信息科学与工程学院  
2023-2024 学年 第 1 学期

# 实验 1 汇编程序结构和基本调试方法

实验日期： 2023-10-11      实验成绩： \_\_\_\_\_ 教师签名： \_\_\_\_\_

## 一、实验名称

- 实验 1 汇编程序结构和基本调试方法

## 二、实验目的

1. 熟练掌握汇编源程序的编辑、汇编、连接、运行和调试的步骤和方法；熟悉汇编程序 TASM、连接程序 TLINK 和调试工具 TD；并能利用调试工具察看存储器和 CPU 寄存器内容，深刻理解存储单元内容和地址、存储器分段结构、各寄存器作用等基本概念。
2. 掌握数据传送指令、算术运算指令、LOOP 指令、逻辑运算指令和移位指令；编写数据累加等简单的汇编程序。
3. 掌握 DOS 系统功能调用进行输入/输出的方法；编写简单的输入/输出程序。

## 三、实验装置

- 实验地点：信息楼 318 机房
- 所使用的工具软件及环境：TASM、TLINK、TD

## 四、实验内容

1. 掌握汇编源程序的编辑、汇编、连接、运行及调试的步骤和方法，并回答下列问题：
  - (1) 汇编语言源程序文件的扩展名为 .asm。
  - (2) 汇编阶段的主要任务是将汇编语言源程序“翻译”成机器代码（目标代码），产生扩展名为 .obj 的目标文件，也可以产生列表文件和交叉引用文件。常用的汇编工具有 Microsoft 公司的 MASM 和 Borland 公司的 TASM。
  - (3) 如何获得 TASM 命令格式？ TASM /?。
  - (4) 连接阶段的主要任务是把几个模块产生的目标文件连接成一个可执行程序，即扩展名为 .exe 或 .com 的文件。
  - (5) 将目标文件 P1.OBJ、P2.OBJ 和子程序库文件 MYLIB.LIB 连接成一个可执行程序 PROG.EXE 的连接命令是 TLink P1.OBJ P2.OBJ MYLIB.LIB PROG.EXE。
  - (6) 在 TD 调试窗口的 CPU 子窗口中，可以查看到 代码 段的地址、内容和对应的符号指令；在堆栈子窗口中可以查看到堆栈的栈顶位置用 ss 标记。

2. 下列程序实现利用公式  $N^2=1+3+5+\dots+(2N-1)$  来计算  $N^2$  的值。修改错误并回答问题。

```

第 1 行      DATA SEGMENT
第 2 行          N DW 23
第 3 行          SQUARE DW ?
第 4 行      DATA ENDS
第 5 行      CODE SEGMENT
第 6 行          ASSUME CS: CODE, DS: DATA
第 7 行      START: MOV DS, DATA
第 8 行          MOV CX, 2*N-1
第 9 行          MOV AX, 0
第 10 行         MOV BX, 1
第 11 行      ONE:  ADD AX, BX
第 12 行          ADD BX, 1
第 13 行          LOOP ONE
第 14 行          MOV SQUARE, AX
第 15 行          MOV AX, 4C00H
第 16 行          INT 21H
第 17 行      CODE ENDS
第 18 行          END START

```

- (1) 错误 1: 第 7 行, 修改为: MOV AX, DATA    MOV DS, AX  
 错误 2: 第 8 行, 修改为: MOV CX, N  
 错误 3: 第 12 行, 修改为: ADD BX, 2

- (2) 将上述程序修改正确, 汇编、连接通过后, 用 TD 进行调试, 并回答下列问题:

(A) 程序载入后, 系统为该程序分配的代码段地址是 0AC8H、数据段地址是 0AB7H。

(B) 循环体执行前, 寄存器 CX、AX、BX 的内容分别为多少? 循环体执行结束呢?

	CX	AX	BX
循环体执行前	0017	0000	0001
循环体执行结束	0000	0211	002F

(C) 指令 MOV SQUARE, AX 执行后, 各变量的存储单元地址及内容分别是多少?

变量名	段地址	偏移地址	内容
N	0AC7H	0000H	0017H
SQUARE	0AC7H	0002H	0211H

3. 下列程序实现：从键盘上输入一行数字字符（明文），然后按照密码加密规则：'0'→'A'，'1'→'B'，'2'→'C'...将其翻译成密文，并把密文显示输出。请在横线处填写正确指令，补充程序完整。调试运行程序，**给出程序运行结果截图**。

```
DATA SEGMENT
    BUFFER DB 81,?,81 DUP(?)
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START: MOV AX,DATA
        MOV DS,AX
        LEA DX,BUFFER      ;输入一行数字字符到 BUFFER 中
        MOV AH,0AH
        INT 21H
        MOV CL,BUFFER+1
        MOV CH,0
        LEA BX,BUFFER+2    ;首字符地址送 BX
NEXT:  MOV AL,[BX]
        ADD AL,'A'-'0'
        MOV [BX],AL
        INC BX
        LOOP NEXT
        MOV BYTE PTR[BX],'$' ;送字符串结束标记
        MOV AH,9           ;显示 BUFFER 中的密文
        MOV DX,OFFSET BUFFER+2
        INT 21H
        MOV AX,4C00H
        INT 21H
CODE ENDS
        END START
```

程序运行结果截图如下：

DOSBOX 0.74-2, Cpu speed: 3000 cycles, F... — X

```

MYPROG      <DIR>          13-10-2023 19:49
POWER       <DIR>          13-10-2023 19:49
TEST        <DIR>          15-10-2023 17:46
DPMI16BI OVL          60,672 20-10-2016 7:50
MYPROG      TR           68 13-10-2023 20:24
RTM         EXE        120,853 21-02-1996 5:00
TASM        EXE        136,018 19-12-2005 16:10
TASM        HLP        274,644 21-02-1996 5:00
TD          EXE        773,468 21-02-1996 5:00
TD          TR           343 15-10-2023 17:36
TDCONFIG TD           691 15-10-2023 17:36
TLIB        EXE         31,104 29-08-1988 2:00
TLINK        EXE        120,426 21-02-1996 5:00
YLIB        EXE        55,279 08-04-2008 8:08
YLIB        H           1,541 16-01-2005 21:43
YLIB16      ASM         6,152 08-04-2008 8:08
YLIB16      LIB         1,536 17-11-2004 9:44
YLIB32      ASM         6,166 08-04-2008 8:08
YLIB32      LIB         1,536 17-11-2004 9:53
YLIB~1      TXT          121 01-01-2005 21:57
  17 File(s)          1,590,618 Bytes.
   5 Dir(s)          262,111,744 Bytes free.

C:\>TEST\TEST
20030725s_

```

DOSBOX 0.74-2, Cpu speed: 3000 cycles, F... — X

```

POWER       <DIR>          13-10-2023 19:49
TEST        <DIR>          15-10-2023 17:46
DPMI16BI OVL          60,672 20-10-2016 7:50
MYPROG      TR           68 13-10-2023 20:24
RTM         EXE        120,853 21-02-1996 5:00
TASM        EXE        136,018 19-12-2005 16:10
TASM        HLP        274,644 21-02-1996 5:00
TD          EXE        773,468 21-02-1996 5:00
TD          TR           343 15-10-2023 17:36
TDCONFIG TD           691 15-10-2023 17:36
TLIB        EXE         31,104 29-08-1988 2:00
TLINK        EXE        120,426 21-02-1996 5:00
YLIB        EXE        55,279 08-04-2008 8:08
YLIB        H           1,541 16-01-2005 21:43
YLIB16      ASM         6,152 08-04-2008 8:08
YLIB16      LIB         1,536 17-11-2004 9:44
YLIB32      ASM         6,166 08-04-2008 8:08
YLIB32      LIB         1,536 17-11-2004 9:53
YLIB~1      TXT          121 01-01-2005 21:57
  17 File(s)          1,590,618 Bytes.
   5 Dir(s)          262,111,744 Bytes free.

C:\>TEST\TEST
CAADAHCFa
C:\>s_

```

4. 编写完整程序实现：以二进制格式输出无符号变量 X 的值。

DATA SEGMENT

    X DW 20

DATA ENDS

STACK SEGMENT PARA STACK 'STACK'

    DW 512 DUP(?)

STACK ENDS

CODE SEGMENT

    ASSUME CS:CODE, DS:DATA, SS:STACK

START:

    MOV AX, DATA

    MOV DS, AX

    MOV BX, X

    MOV CX, 16

AGAIN:

    MOV DL, 0

    SHL BX, 1

    JNC DIS

    MOV DL, 1

DIS:

    ADD DL, 30H

    DEC CX

    MOV AH, 2

    INT 21H

    JNZ AGAIN

    MOV AH, 2

    MOV DL, 'B'

    INT 21H

MOV AX, 4C00H

INT 21H

CODE ENDS

END START

## 实验 2 选择结构程序、循环结构程序及子程序设计

实验日期： 2023-10-25      实验成绩： \_\_\_\_\_ 教师签名： \_\_\_\_\_

### 一、实验名称

- 实验 2 选择结构程序、循环结构程序及子程序设计

### 二、实验目的

1. 掌握利用转移指令、比较指令和测试指令设计单分支、双分支和多分支选择汇编程序的方法。
2. 掌握循环指令 LOOP、LOOPZ、LOOPNZ 的功能与用途。
3. 掌握循环程序的设计方法，了解多重循环程序的设计要点。
4. 掌握带重复前缀的串处理指令的应用。
5. 掌握子程序的定义、调用及返回方法。
6. 熟悉主程序与子程序间几种参数传递方法。
7. 深刻理解子程序的调用与返回对堆栈的影响。

### 三、实验装置

- 实验地点：信息楼 318 机房
- 所使用的工具软件及环境：TASM、TLINK、TD

### 四、实验内容

1. 下列程序完成对分段函数  $Y = \begin{cases} X - 3 & X < -2 \\ 2X + 6 & -2 \leq X \leq 3 \\ 2 & X > 3 \end{cases}$  进行计算的功能。

要求：修改程序中存在的错误，并运行程序。

```
DATA SEGMENT
    X    DW    ?
    Y    DW    ?
DATA ENDS
CODE SEGMENT
    ASSUME CS: CODE, DS: DATA
START:  MOV    AX, DATA
        MOV    DS, AX
```



```

                MOV  AX, X
                CMP  AX, -2
                JB   L2
L1:             SUB  AX, 3
L2:             CMP  AX, 3
                JBE  L3
                ADD  AX, AX
                ADD  AX, 6
L3:             MOV  AX, 2
DONE:          MOV  Y, AX
                MOV  AX, 4C00H
                INT  21H
CODE           ENDS
                END  START

```

改错如下：

```

DATA SEGMENT
    X  DW  1
    Y  DW  ?
DATA ENDS
CODE SEGMENT
                ASSUME CS: CODE, DS: DATA
START:         MOV  AX, DATA
                MOV  DS, AX
                MOV  AX, X
                CMP  AX, -2
                JGE  L2
L1:            SUB  AX, 3
                JMP  DONE
L2:            CMP  AX, 3
                JG   L3
                ADD  AX, AX
                ADD  AX, 6
                JMP  DONE
L3:            MOV  AX, 2
DONE:          MOV  BX, AX

```

```

                MOV    AX, 4C00H
                INT     21H
CODE           ENDS
                END     START

```

2. 编写程序，从键盘输入一个字符串，统计其中数字字符、小写字母、大写字母和空格的个数。

DATA SEGMENT

```

X DW 1          ; 存储数字字符个数
Y DW ?          ; 存储小写字母个数

```

DATA ENDS

CODE SEGMENT

```

    ASSUME CS: CODE, DS: DATA

```

START:

```

    MOV AX, DATA
    MOV DS, AX

```

; 从键盘输入字符串

```

    MOV AH, 0Ah
    MOV DX, OFFSET STRBUF
    INT 21h

```

; 统计字符个数

```

    MOV CX, 0          ; 计数器清零
    MOV SI, OFFSET STRBUF+2 ; 跳过输入缓冲区的长度信息
    MOV BX, 0          ; 字符类型计数器清零

```

CountLoop:

```

    LODSB              ; 读取一个字符到 AL
    CMP AL, '$'        ; 判断是否到达字符串结尾
    JE Done            ; 如果到达结尾，跳转到结束
    CMP AL, '0'
    JB NotDigit        ; 如果字符小于'0'，不是数字字符
    CMP AL, '9'

```

JA NotDigit ; 如果字符大于'9', 不是数字字符  
INC WORD PTR [X] ; 增加数字字符个数  
JMP Continue

NotDigit:

CMP AL, 'a'  
JB NotLetter ; 如果字符小于'a', 不是字母  
CMP AL, 'z'  
JA NotLetter ; 如果字符大于'z', 不是字母  
INC WORD PTR [Y] ; 增加小写字母个数  
JMP Continue

NotLetter:

CMP AL, 'A'  
JB NotUpper ; 如果字符小于'A', 不是字母  
CMP AL, 'Z'  
JA NotUpper ; 如果字符大于'Z', 不是字母  
INC WORD PTR [Y] ; 增加小写字母个数  
JMP Continue

NotUpper:

CMP AL, ' ' ; 判断是否为空格  
JNE Continue ; 如果不是空格, 跳转到继续下一个字符  
INC WORD PTR [Y] ; 增加空格个数

Continue:

INC CX ; 计数器加 1  
JMP CountLoop ; 继续下一个字符

Done:

; 统计结果已存储在 X 和 Y 寄存器中, 可以根据需要进行进一步处理

MOV AX, 4C00h ; 结束程序  
INT 21h

CODE ENDS  
END START

3. 下面程序求某数据区中无符号数据最大值，上机运行该程序，观察程序执行时堆栈的变化，并画出以下 6 个堆栈状态图。(1) CALL MAX 指令执行之前；(2) CALL MAX 指令执行之后；(3) 保护现场之后；(4) 恢复现场之后；(5) RET 2 指令执行之后；(6) 执行指令 POP SMAX 之后。

```
DATA    SEGMENT
        BUF      DW    98, 34, 897, 345, 678, 21345, 67, 2
        COUNT EQU  ($-BUF) / 2
        SMAX DW    ?
DATA    ENDS
CODE    SEGMENT
        ASSUME CS: CODE, DS: DATA
START:  MOV  AX, DATA
        MOV  DS, AX
        LEA  AX, BUF
        PUSH AX
        MOV  AX, COUNT
        PUSH AX
        CALL MAX
        POP  SMAX
        MOV  AH, 4CH
        INT  21H
MAX     PROC
        PUSH BP
        MOV  BP, SP
        MOV  SI, [BP+6]
        MOV  CX, [BP+4]
        MOV  BX, [SI]
        DEC  CX
MAX1:   ADD  SI, 2
        CMP  BX, [SI]
```

```

        JAE    NEXT
        MOV    BX, [SI]
NEXT:   LOOP   MAX1
        MOV    [BP+6], BX
        POP     BP
        RET     2

MAX      ENDP
CODE    ENDS
END START

```

FFF0H	
FFF2H	
FFF4H	
FFF6H	
FFF8H	
FFFAH	
FFFCH	0008
FFFEH	0000
0000H	

(1)CALL MAX 执行之前

FFF0H	
FFF2H	
FFF4H	
FFF6H	
FFF8H	
FFFAH	0010
FFFCH	0008
FFFEH	0000
0000H	

(2)CALL MAX 执行之后

FFF0H	
FFF2H	
FFF4H	
FFF6H	
FFF8H	0000
FFFAH	0010
FFFCH	0008
FFFEH	0000
0000H	

(3)保护现场之后

FFF0H	
FFF2H	
FFF4H	
FFF6H	
FFF8H	
FFFAH	0010
FFFCH	0008
FFFEH	5631
0000H	

(4) 恢复现场之后

FFF0H	
FFF2H	
FFF4H	
FFF6H	
FFF8H	
FFFAH	
FFFCH	
FFFEH	5361
0000H	

(5)RET 2 执行之后

FFF0H	
FFF2H	
FFF4H	
FFF6H	
FFF8H	
FFFAH	
FFFCH	
FFFEH	
0000H	

(6)POP SMAX 执行后

4. 修改 4 中程序，要求子程序与主程序间通过寄存器传递参数，求 BUF 中无符号数据的最大值并保存在变量 SMAX 中。其中：

(1) 主程序实现：设置入口参数（BUF 首地址→SI、数组长度→CX）；调用子程序 MAX；将出口参数保存在变量 SMAX 中。

(2) 子程序实现：求数组最大值，并保存在 BX 中作为出口参数返回。

DATA	SEGMENT
BUF	DW    98, 34, 897, 345, 678, 21345, 67, 2
COUNT	EQU (\$-BUF) / 2
	SMAX    DW    ?
DATA	ENDS
CODE	SEGMENT
ASSUME	CS: CODE, DS: DATA
START:	
	MOV AX, DATA
	MOV DS, AX
	LEA SI, BUF
	MOV CX, COUNT
	PUSHSI
	PUSHCX
	CALLMAX
	POP    SMAX
	MOV AH, 4CH
	INT    21H
MAX	PROC
	PUSHBX
	PUSHSI
	PUSHCX
	MOV SI, [BP+6]
	MOV CX, [BP+4]
	MOV BX, [SI]
	DEC CX
MAX1:	
	ADD SI, 2
	CMP BX, [SI]
	JAE NEXT
	MOV BX, [SI]
NEXT:	

```

                                LOOPMAX1
                                MOV [BP+6], BX
                                POP      CX
                                POP      SI
                                POP      BX
                                POP      BP
                                RET      2

MAX                                ENDP

CODE                                ENDS
END                                START

```

## 实验 3 使用 8259 的单级中断控制器

实验日期： 2023-11-24      实验成绩： \_\_\_\_\_ 教师签名： \_\_\_\_\_

### 一、实验名称

- 实验 3 使用 8259 的单级中断控制器

### 二、实验目的

1. 掌握 8259 中断控制器与微机接口的硬件设计原理和要求。
2. 掌握 8259 初始化编程格式和方法。
3. 掌握 8259 中断控制器的应用编程。

### 三、实验装置

- 实验地点：信息楼 418 机房
- 实验设备及器材配置：DJ86PCI 实验箱、配套微机、DJ-8086 系列实验指导书

### 四、实验内容

利用 8259 实现对外部中断的响应和处理，要求程序对每次中断进行计数，并将计数结果送数码显示。

### 五、实验要求

1. 学习并掌握硬件连接步骤与方法后，连接好硬件线路。
2. 阅读并理解实验程序 H8259.ASM 后，运行实验程序。
3. 观察实验结果，理解并分析实验原理和方法。
4. 撰写实验报告，内容包括：实验原理与方法、实验硬件连接电路图、程序流程图、实验代码分析、学生自我小结。

### 六、教师评语

1. 设计思路：[清晰、较清晰、一般、较差]
2. 电路选择：[合理、较合理、一般、较差]
3. 代码：[很好、较好、一般、较差] 实现了设计要求，功能完整
4. 实验报告：格式规范、内容完整、表达清晰准确 [很好、较好、一般、较差]
5. 总评价：



## 七、实验报告

### 1. 实验原理与方法（分析综述相关主要技术的原理与方法）

#### 1、8259 芯片

中断控制器 8259A 是专为控制优先级中断而设计的芯片。

它将中断源优先级排队、辨别中断源以及提供中断矢量的电路集于一片中。因此无需附加任何电路，只需对 8259A 进行编程，就可以管理 8 级中断，并选择优先模式和中断请求方式。即中断结构可以由用户编程来设定。同时，在不需要增加其它电路的情况下，通过多片 8259A 的级联，能构成多达 64 级的矢量中断系统。

#### 2、8259A 的工作方式

（1）设置优先级方式，有全嵌套方式、特殊嵌套方式、优先级自动循环方式、优先级特殊循环方式；

（2）中断屏蔽方式，有普通屏蔽方式和特殊屏蔽方式两种；

（3）结束中断方式，有自动结束中断方式、普通结束中断方式、特殊结束中断方式三种；

（4）中断查询方式，通过读取中断查询字就可查到是否有中断请求和哪一级的优先级最高。

#### 3、8259A 的命令字及编程

##### （1）初始化命令字 ICW

初始化命令字有四个，ICW1-4，按照一定的顺序写入控制口，用以设置 8259A 的初始状态。在 8086

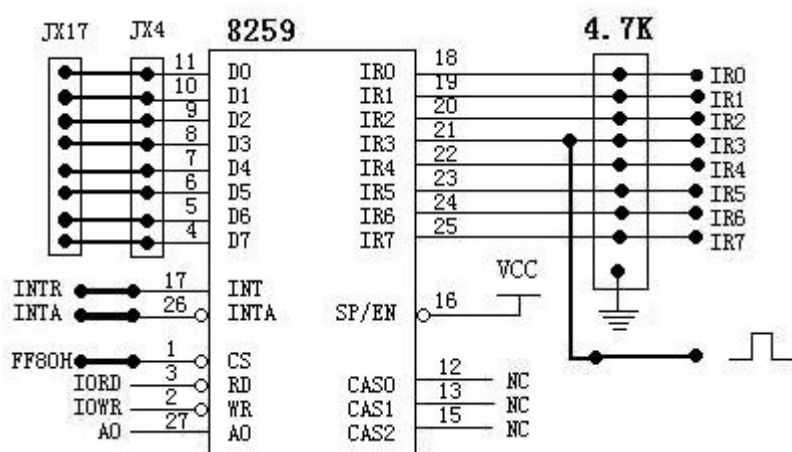
系统中，ICW1、ICW2、ICW4 是必须要有的，而 ICW3 只有在级联时需要使用。

##### 操作命令字 OCW

操作命令字有三个，OCW1-3，在 8259A 工作期间，可以通过向它写入操作命令，让它按照需要的方式工作。OCW 是在应用程序内部设置的，没有规定写入的先后顺序，但写入的端口地址有明确的规定，OCW1 必须写入奇地址口，OCW2、OCW3 需要写入偶地址口。

### 2. 实验硬件连接（给出实验硬件线路图及连接说明）


#### 1、硬件线路图：





## 2、实验硬件连接

1) 实验中 8259A 中断序号及对应变量地址如下：

中断序号	0	1	2	3	4	5	6	7
变量地址	20H 23H	24H 27H	28H 2BH	2CH 2FH	30H 33H	34H 37H	38H 3BH	3CH 3FH

2) 本实验中使用 3 号中断源 IR3，“”插孔和 IR3 相连，中断方式为边沿触发方式，每拨二次 AN 开关产生一次中断，满 5 次中断，显示“8259—good”。如果中断源电平信号不符合规定要求，则自动转到 7 号中断，显示“Err”。

3) 连接及步骤说明

- 将 8259 的 INT 连 8088 的 INTR；
- 将 8259 的 INTA 连 8088 的 INTA；
- “”插孔和 8259 的 3 号中断 IR3 插孔相连，“”端初始为低电平；
- 将 8259 的 CS 端接 FF80H 孔；
- 运行实验程序，在系统处于命令提示符“P.”状态下，按 SCAL 键，输入 12D0，按 EXEC 键，系统显示 8259—1；
- 拨动 AN 开关按钮，按满 10 次显示 good。

## 3. 实验程序流程图

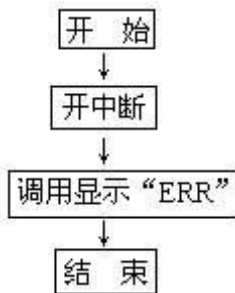
1. 主程序流程图



2. IR3 中断服务程序流程图



3. IR7 中断服务程序流程图



#### 4. 实验代码分析（给出汇编程序代码，并进行必要注释）

(1) 8259 初始化的程序代码

```

MOV AL,13H                ;写 ICW1
MOV DX,INTPORT1 OUT DX,AL
MOV AL,08H                ;写 ICW2
MOV DX,INTPORT2 OUT DX,AL
MOV AL,09H                ;写 ICW4
OUT DX,AL
MOV AL,0F7H               ;写 OCW1 OUT DX,AL
  
```

(2) IR3 中断服务程序的代码

INTREEUP3:

```
CLI ;IR3 中断服务程序
PUSH AX ;压栈保护现场
PUSH BX
PUSH CX
PUSH DX
MOV AL,INTCNT
CALL CONVERS ;拆送缓冲区
MOV BX,OFFSET BUF ;077BH
MOV AL,10H
MOV CX,05H
```

INTRE0:

```
MOV [BX],AL ;高五位缓冲区送熄灭码
INC BX
LOOP INTRE0
MOV AL,20H ;中断返回码 MOV DX,INTPORT1
OUT DX,AL
ADD INCNT,01H
CMP INCNT,0AH ;按满 10 次显示 good
JNA INTRE2 ;中断次数小于 5 次继续
CALL BUF2 ;中断次数结束 DISP:good
```

INTRE1:

```
CALL DISP
JMP INTRE1
```

CONVERS:

```
AND AL,0FH ;拆送缓冲区
MOV BX,offset buf ;077AH
MOV [BX+5],AL
RET
```

INTRE2:

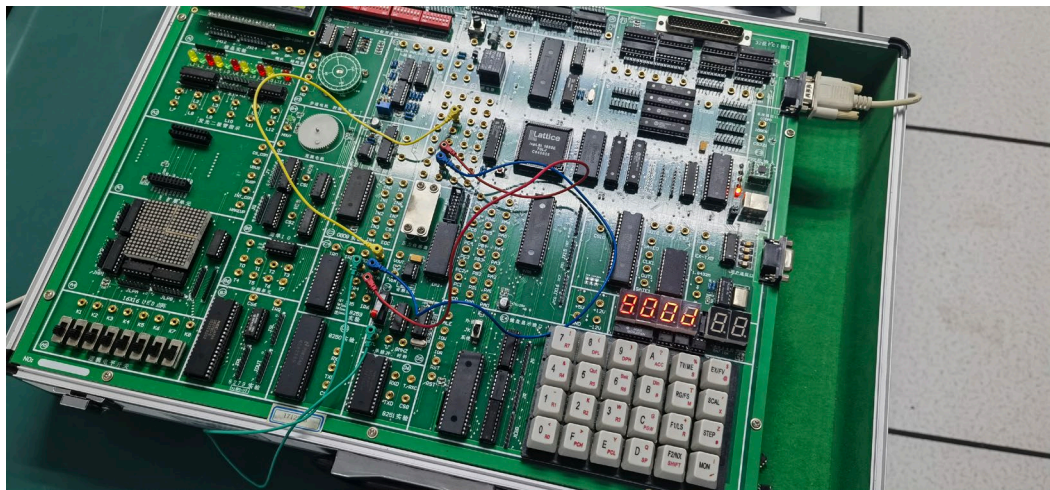
```
MOV AL,20H
MOV DX,INTPORT1
OUT DX,AL
POP DX
```

;IR3 中断返回

### 5. 学生自我小结（所遇问题和个人体会等方面总结）

在实验过程中，由于机器与电脑的连线比较松，所以需要注意检查软件左下方是否显示已连接。并且需要键编译运载才能进行拨动机器。

在多次检查连接成功后,拨动开关 AN 按钮,按满 10 次后显示屏显示了‘good’.



通过本次实验，我了解了 8259 中断控制器与微机接口的硬件设计原理和要求，掌握了 8259 初始化编程格式和方法，对 8259 芯片有了更加深入的认识。同时也体会到了硬件编程的乐趣。

## 实验 4 8255A 可编程并行接口

实验日期： 2023-12-06      实验成绩：                           教师签名：                     

### 一、实验名称

- 实验 4 8255A 可编程并行接口

### 二、实验目的

1. 掌握 8255A 接口的硬件设计原理和要求。
2. 掌握 8255A 的工作方式和编程原理。
3. 掌握 8255A 的初始化方法、并行接口传输数据的方法。

### 三、实验装置

- 实验地点：信息楼 418 机房
- 实验设备及器材配置：DJ86PCI 实验箱、配套微机、DJ-8086 系列实验指导书

### 四、实验内容及要求

1. 参考“DJ-8086 系列实验指导书”第 54 页“实验四 8255A 并行口实验（一）”，学习并掌握硬件连接步骤与方法后，连接好硬件线路，并运行 H8255-1.ASM，查看实验结果。
2. 设计如下功能电路，并编制相应的程序，调试运行最终达到设计要求（选做一）。
  - 1) 用 8255A 实现 12 个开关控制 12 个发光二极管的亮灭。
  - 2) 用 8255A 控制 12 个（或 8 个）发光二极管轮流亮灭。
  - 3) 用 8255A 控制 12 个（或 8 个）发光二极管间隔发亮。
3. 撰写实验报告，要求包括以下内容：实验原理与方法、实验硬件连接电路图、程序流程图、实验代码分析、学生自我小结。

### 五、教师评语

1. 设计思路：[清晰、较清晰、一般、较差]
2. 电路选择：[合理、较合理、一般、较差]
3. 代码：[很好、较好、一般、较差] 实现了设计要求，功能完整
4. 实验报告：格式规范、内容完整、表达清晰准确 [很好、较好、一般、较差]
5. 总评价：

## 六、实验报告

### 1. 实验原理与方法（分析综述相关主要技术的原理与方法）

#### 1、8255A 芯片简介

8255A 是一种通用的可编程并行 I/O 接口芯片，它是为 Intel 系列微处理器设计的配套电路，也可用在其他的微处理器系统中。

通过对 8255A 进行编程，芯片可工作与不同的工作方式。在微型计算机系统中，用 8255A 做接口时，通常不需要附加外部逻辑电路，就可直接在 CPU 于外设之间提供数据通道。

8255A 具有 A、B、C 三个并行接口，均为 8 位的输入输出端口，通过外部的 24 根输入输出线与外设交换数据或进行通信联络。其中 A、B 口都可作为 8 位的输入或输出端口，C 口既可作为 8 位的输入或输出用，又可作为两个 4 位的输入输出用，常用来配合 A、B 口的工作，分别用来产生 A、B 口的输出控制信号以及输入 A 口和 B 口的端口状态信号。

#### 2、8255A 的控制字

8255A 有两类控制字：一类用于定义各端口的工作方式，称为方式选择控制字；另一类用于对 C 端口的任一位进行置位或复位操作，称为置位/复位控制字。D7 位为这两个控制字的标志位，为 1 时是前者，为 0 时是后者。8255A 的工作方式

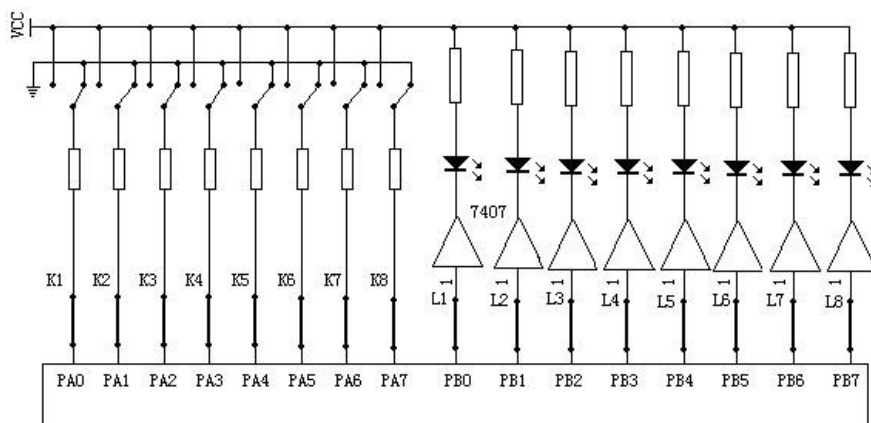
#### 3、8255A 能在以下三种方式下工作：

- 方式 0：基本输入/输出方式，适用于不需要用应答信号的简单输入输出场合；
- 方式 1：选通输入/输出方式，适用于查询和中断传送（通常为输出）；
- 方式 2：双向选通工作方式，只有 A 口可以工作于这种方式。

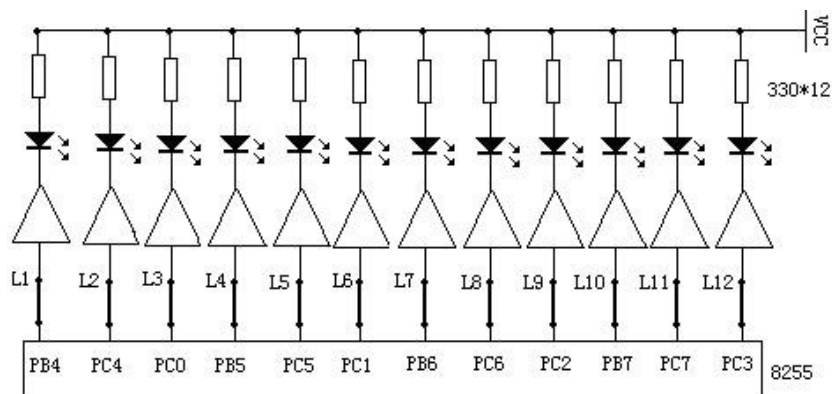
### 2. 实验硬件连接（给出实验硬件线路图及连接说明）

#### 1、硬件线路图：

##### （1）8255A 并行接口实验（一）



(2) 用 8255A 实现 12 个开关控制 12 个发光二极管的亮灭



## 2、实现硬件连接

### (1) 8255A 并行接口实验（一）

在本实验中 8255A 端口 A 工作在方式 0 并作为输入口，读取 K1-K8 个开关量，PB 口工作在方式 0

并作为输出口。硬件连接及步骤说明如下：

- 在系统显示监控提示符“P.”时，按 SCAL 键，传送 EPROM 中的实验程序到内存中；注：必须先传送 EPROM 后，再往下操作；

- 将 8255A 芯片 A 口的 PA0-PA7 依次和开关量输入 K1-K8 相连；

- 将 8255A 芯片 B 口的 PB0-PB7 依次接 L1-L8；

- 运行实验程序，在系统显示监控提示符“P.”时，输入 11B0，按 EXEC 键，系统显示执行提示符“┐”拨动 K1-K8，L1-L8 会跟着亮灭。

### (2) 用 8255A 实现 12 个开关控制 12 个发光二极管的亮灭

本实验中 8255A 三个端口均工作在方式 0，并处于输出状态。各发光二极管共阳极，使其点亮应使 8255A 相应端口的位清 0。硬件连接及步骤说明如下：

- 在系统显示监控提示符“P.”时，按 SCAL 键；

- 按硬件线路图连好实验线路；

- 8255A: PC0—>L3, PC1—>L6, PC2—>L9, PC3—>L12, PC4—>L2, PC5—>L5, PC6—>L8,

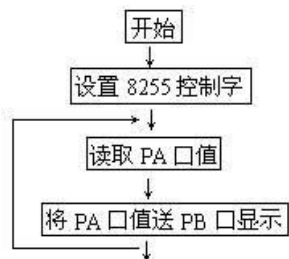
PC7—>L11; PB4—>L1, PB5—>L4, PB6—>L7, PB7—>L10;

- 在“P.”状态时，输入 11E0 后，按 EXEC 键，系统显示执行符“┐”，同时 L1-L12 发光二极管模拟交通灯显示。

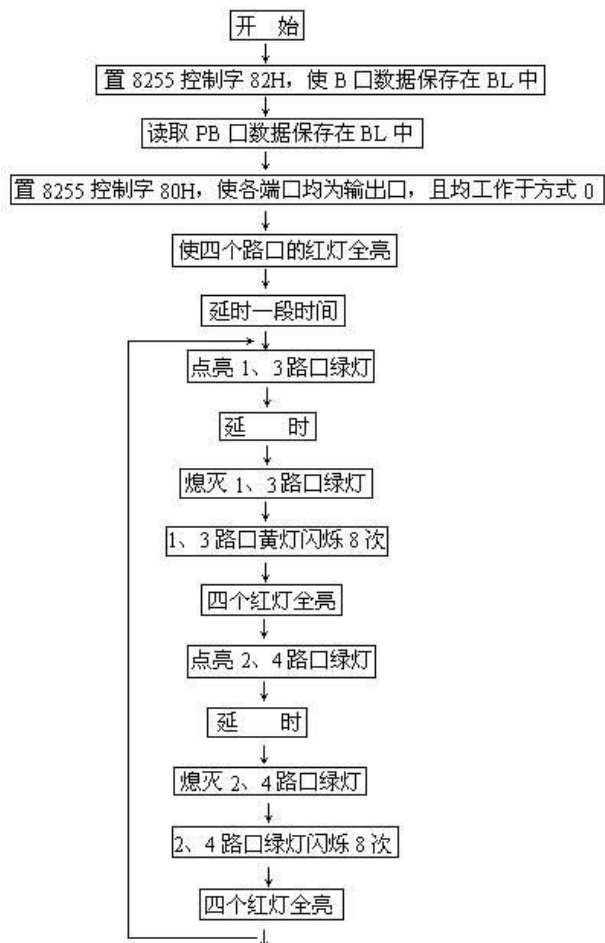
## 3. 实验程序流程图

### (1) 8255A 并行接口实验（一）





(2) 用 8255A 实现 12 个开关控制 12 个发光二极管的亮灭



#### 4. 实验代码分析 (给出汇编程序代码, 并进行必要注释)

(1) 8255A 并行接口实验 (一)

CODE

SEGMENT

ASSUME CS:CODE

IOCONPT EQU 0FF2BH ;定义 8255 控制口

IOBPT EQU 0FF29H ;定义 8255 PB 口

IOAPT EQU 0FF28H ;定义 8255 PA 口

ORG 11B0H

START:

```

MOV AL,90H                ;定义 PA 输入,PB 输出
MOV DX,IOCONPT            ;控制口
OUT DX,AL                 ;写命令字
NOP                       ;延时
NOP
NOP
IOLED1:
MOV DX,IOAPT
IN AL,DX                  ;读 PA 口
MOV DX,IOBPT              ;PB 口
OUT DX,AL                 ;写 PB 口
MOV CX,0FFFFH             ;延时

DELAY:
LOOP DELAY
JMP IOLED1                ;循环

CODE ENDS
END START

```

(2) 用 8255A 实现 12 个开关控制 12 个发光二极管的亮灭

```

CODE
    SEGMENT
    ASSUME      CS:CODE
    IOCONPT     EQU 0FF2BH    ;8255 控制口
    IOAPT       EQU 0FF28H    ;PA 口
    IOBPT       EQU 0FF29H    ;PB 口
    IOCPT       EQU 0FF2AH    ;PC 口
    ORG         11e0H

START:
MOV AL,82H        ;PB 输入,PA,PC 输出
MOV DX,IOCONPT
OUT DX,AL         ;写命令字
MOV DX,IOBPT      ;读 PB 口
IN AL,DX
MOV BYTE PTR DS:[0601H],AL ;保存 PB 口数据
MOV DX,IOCONPT    ;8255 控制口
MOV AL,80H

```

	OUT DX,AL	;写命令字, PA,PB,PC 输出
	MOV DX,IOBPT ;	
	MOV AL,DS:[0601H]	
	OR AL,0F0H	
	OUT DX,AL	
	MOV DX,IOCPT	
	MOV AL,0F0H	
	OUT DX,AL	;四路口红灯全亮
	CALL DELAY1	;延时
IOLED0:		
	MOV AL,10100101B	
	MOV DX,IOCPT	
	OUT DX,AL	;南北绿灯亮,东西红灯亮
	CALL DELAY1	;延时
	CALL DELAY1	;延时
	OR AL,0F0H	
	OUT DX,AL	;南北绿灯灭
	MOV CX,8H	
IOLED1:		
	MOV DX,IOBPT	
	MOV AL,DS:[0601H]	
	AND AL,10101111B	
	OUT DX,AL	;南北黄灯亮
	CALL DELAY2	;延时
	OR AL,01010000B	
	OUT DX,AL	;南北黄灯灭
	CALL DELAY2	;延时
	LOOP IOLED1	;南北黄灯闪烁 8 次
	MOV DX,IOCPT	
	MOV AL,0F0H	
	OUT DX,AL	;四路口红灯全亮
	CALL DELAY2	;延时
	MOV AL,01011010B	
	OUT DX,AL	;东西绿灯亮
	CALL DELAY1	;延时

```

        CALL DELAY1
        OR AL,0F0H
        OUT DX,AL                ;东西绿灯灭
        MOV CX,8H

IOLED2:
        MOV DX,IOBPT
        MOV AL,DS:[0601H]
        AND AL,01011111B
        OUT DX,AL                ;东西黄灯亮
        CALL DELAY2              ;延时
        OR AL,10100000B
        OUT DX,AL                ;东西黄灯灭
        CALL DELAY2              ;延时
        LOOP IOLED2              ;东西黄灯闪烁 8 次
        MOV DX,IOCPT
        MOV AL,0F0H
        OUT DX,AL                ;四路口红灯全亮
        CALL DELAY2
        JMP IOLED0              ;循环继续

DELAY1:
        PUSH AX
        PUSH CX
        MOV CX,0030H

DELY2:
        CALL DELAY2
        LOOP DELY2
        POP CX
        POP AX
        RET

DELAY2:
        PUSH CX
        MOV CX,8000H

DELA1:
        LOOP DELA1
        POP CX

```

RET

CODE ENDS

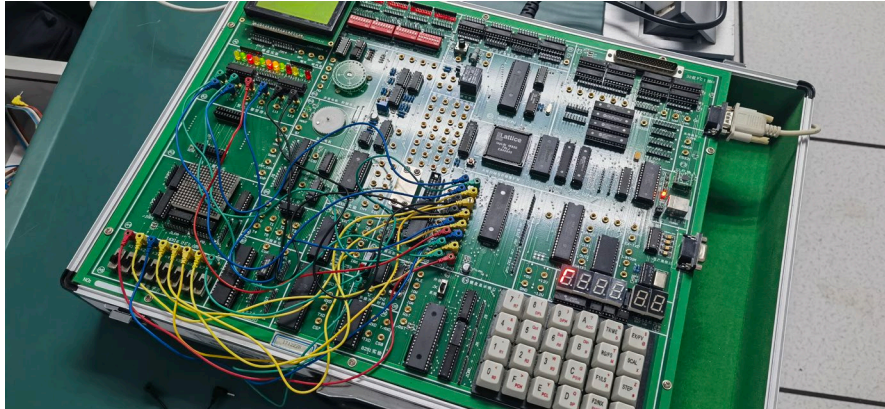
END START

5. 学生自我小结（所遇问题和个人体会等方面总结）

本次实验(一)，拨动开关 K1~K8，灯 L1~L8 会跟着亮灭

本次实验(二)，L1~L12 发光二极管模拟交通灯显示。

实验(二)中，把 OUT0 连接多个二极管



通过本次实验，我了解了 8255A 可编程并行 I/O 接口芯片的硬件设计原理，掌握了 8255A 的初始化方法、并行接口传输数据的方法。学了了利用这些芯片进行交通信号灯的模拟编程。

## 实验 5 8253 可编程定时器/计数器

实验日期： 2023-12-15      实验成绩： \_\_\_\_\_ 教师签名： \_\_\_\_\_

### 一、实验名称

- 实验 5 8253 可编程定时器/计数器

### 二、实验目的

1. 掌握 8253 与微机接口的原理和方法。
2. 掌握 8253 定时器/计数器的工作方式和编程原理。

### 三、实验装置

- 实验地点：信息楼 418 机房
- 实验设备及器材配置：DJ86PCI 实验箱、配套微机、DJ-8086 系列实验指导书

### 四、实验内容

1. 阅读“DJ-8086 系列实验指导书”第 61 页“实验六 定时器 / 计数器”，掌握硬件连接步骤与方法后，连接好硬件线路，并运行 H8253.ASM，查看实验结果。
2. 设计实现如下功能电路，并编制相应的程序：用 8253 来控制一个发光二极管的点亮与熄灭，要求点亮 10 秒钟后再让它熄灭 10 秒，并重复上述过程。

### 五、实验要求

1. 按“实验内容”中的要求，连接好硬件线路，编写相应的汇编语言源程序，运行程序。观察实验结果，理解并分析实验原理和方法。
2. 撰写实验报告，要求包括以下内容：实验原理与方法、实验硬件连接电路图、程序流程图、实验代码分析、学生自我小结。

### 六、教师评语

1. 设计思路：[清晰、较清晰、一般、较差]
2. 电路选择：[合理、较合理、一般、较差]
3. 代码：[很好、较好、一般、较差] 实现了设计要求，功能完整
4. 实验报告：格式规范、内容完整、表达清晰准确 [很好、较好、一般、较差]
5. 总评价：

## 七、实验报告

### 1. 实验原理与方法（分析综述相关主要技术的原理与方法）

#### 1、8253 芯片简介

8253 是一种常用的计数器/定时器芯片，被称为可编程间隔定时器。

8253 内部具有 3 个独立的 16 位计数器通道，通过对它进行编程，每个计数器通道均可按 6 种不同的方式工作，并且都可以按二进制或十进制格式进行计数，其计数频率范围为 0-2MHz，用+5V 单电源供电。

在微型计算机系统中，常需要用到 8253 实现定时功能。例如，在 IBM PC 机中，需要有这样一个实时时钟以实现计时功能，还要求按一定的时间间隔对动态 RAM 进行刷新。此外扬声器的发声也是由定时信号来驱动的。在计算机实时控制和处理系统中，则要按一定的采样周期对处理对象进行采样，或定时检测某些参数等等，都需要定时信号。再者，在许多微机应用系统中，还会用到计时功能，需对外部事件进行计数。此外，8253 还适用于许多其它的情况，例如用作可编程方波频率产生器、分频器、程控单脉冲发生器等等。

#### 2、8253 的 6 种工作方式

方式 0：计数结束中断；方式 1：可编程频率发生器；方式 2：频率发生器；方式 3：方波频率发生器；方式 4：软件触发的选通信号；方式 5：硬件触发的选通信号。

对于方式 0，在写入控制字后，输出端即变低，计数结束后，输出端由低变高，常用该输出信号作为中断源。其余 5 种方式写入控制字后，输出均变高。方式 0 可用来实现定时或对外部事件进行计数。

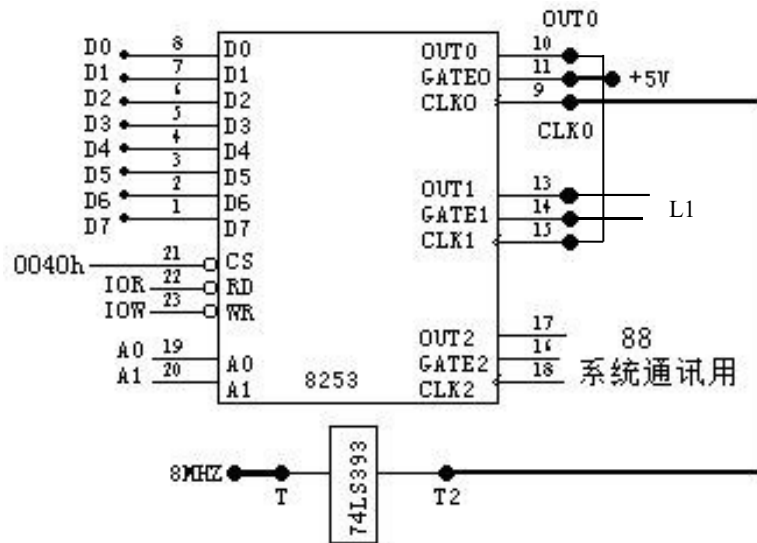
方式 1 用来产生单脉冲。

方式 2 用来产生序列负脉冲，每个负脉冲的宽度与 CLK 脉冲的周期相同。方式 3 用于产生连续的方波。方式 2 和 3 都实现对时钟脉冲进行 n 分频。

方式 4 和方式 5 的波形相同，都在计数器回 0 后，从 OUT 端输出一个负脉冲，其宽度等于一个时钟周期。但方式 4 由软件（设置计数值）触发计数，而方式 5 由硬件（门控信号 GATE）触发计数。这 6 种工作方式中，方式 0、1、4，计数初值装进计数器后，仅一次有效。如果要使通道再次按此方式工作，必须重新装入计数值。对于方式 2、3、5，在减 1 计数到 0 值后，8253 会自动将计数值重装进计数器。

### 2. 实验硬件连接（给出实验硬件线路图及连接说明）

#### 1、硬件线路图：



## 2、实验硬件连接

按照上图连接好实验线路；

将 8253 的 GATE0、GATE1 接+5V；

将 8253 的 CLK0 插孔接分频器 74LS393（左下方）的 T2 插孔，分频器的频率源为 8.0MHZ，T→8.0MHZ 由于实验室暂时没有示波器，我们采用通道级联的方式，然后观察一个发光二极管的点亮与熄灭来

验证实验结果。将通道 0 与通道 1 级联：OUT0 接到 CLK1，OUT1 接到发光二极管 L1；

运行实验程序，在系统提示符“P.”状态下，按 SCAL 键，输入 1290，按 EXEC 键，要求点亮 10 秒钟后再熄灭 10 秒，重复上述亮灭过程。

## 3. 实验程序流程图



## 4. 实验代码分析（给出汇编程序代码，并进行必要注释）

CODE

SEGMENT

ASSUME CS: CODE

ORG 1290H



```

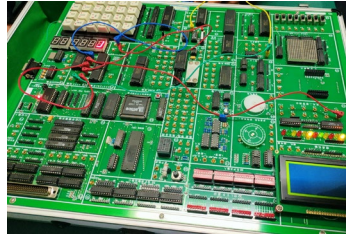
START:
    JMP TCONT
    TCONTRO EQU 0043H
    TCON0 EQU 0040H
    TCON1 EQU 0041H
TCONT:
    MOV DX,TCONTRO                ;8253 控制口
    MOV AL,00110101B              ;命令字,通道 0
                                    ;先低后高,方式 2,BCD
    OUT DX,AL                     ;写命令字
    MOV DX,TCON0                  ;计数器 0 端口
    MOV AL,00H                   ;计数初值低位
    OUT DX,AL
    MOV AL,80H                   ;计数初值高位
    OUT DX,AL
    MOV DX,TCONTRO                ;8253 控制口
    MOV AL,01110111B              ;命令字,通道 1,
                                    ;先低后高,方式 3,BCD
    OUT DX,AL                     ;写命令字
    MOV DX,TCON1                  ;计数器 1 端口
    MOV AL,00H                   ;计数初值低位
    OUT DX,AL
    MOV AX,200H                   ;计数初值高位
    OUT DX,AX
    JMP $
CODE ENDS
END START

```

## 5. 学生自我小结（所遇问题和个人体会等方面总结）

因为 OUT1 输出的脉冲周期最大只有  $0.5 \mu s * 65536 = 32768 \mu s = 32.768ms$ , 达不到 20 秒的要求, 所以本实验中需要将 8253 的几个通道级联来解决这个问题

本次实验, 我们小组实现了让灯重复交替亮和灭 10 秒钟。



通过本次实验，更加深入地了解 8253 可编程计数器/定时器芯片的硬件设计原理和要求，掌握了 8253 的工作方式、计数初值设置以及初始化方法，并能够对其进行初始化编程。

## 实验 6 ADC0809 模/数转换接口实验

实验日期： 2023-12-22      实验成绩： \_\_\_\_\_ 教师签名： \_\_\_\_\_

### 一、实验名称

- 实验 6 ADC0809 模/数转换接口实验

### 二、实验目的

1. 了解模/数转换的基本原理。
2. 掌握 ADC0809 与微机接口的原理和方法。
3. 熟悉 ADC0809 的使用方法。

### 三、实验装置

- 实验地点：信息楼 418 机房
- 实验设备及器材配置：DJ86PCI 实验箱、配套微机、DJ-8086 系列实验指导书

### 四、实验内容

利用实验系统上电位器提供的可调电压作为 0809 模拟信号的输入，编制程序，将模拟量转换为数字量，通过数码管显示出来。

### 五、实验要求

1. 学习并掌握硬件连接步骤与方法后，连接好硬件线路。
2. 阅读并理解实验程序 H0809.ASM 后，运行实验程序。
3. 实验结果，理解并分析实验原理和方法。
4. 撰写实验报告，要求包括以下内容：实验原理与方法、实验硬件连接电路图、程序流程图、实验代码分析、学生自我小结。

### 六、教师评语

1. 设计思路：[清晰、较清晰、一般、较差]
2. 电路选择：[合理、较合理、一般、较差]
3. 代码：[很好、较好、一般、较差] 实现了设计要求，功能完整
4. 实验报告：格式规范、内容完整、表达清晰准确 [很好、较好、一般、较差]
5. 总评价：

## 七、实验报告

### 1. 实验原理与方法（分析综述相关主要技术的原理与方法）

#### 1、ADC0809 简述

ADC0809 是 NSC公司的8通道8位逐次逼近式A/D转换器。速度快、分辨率高，芯片成本较低，在数据采集系统中广泛应用。

转换原理以逐次逼近原理为基础，即把输入电压 $V_i$ 与一组从参考电压分层得到的量化电压比较，从最大量化电压开始，由粗到细逐次进行，根据每次比较的结果，确定相应的位是 1 还是 0。不断比较，不断逼近，直到两者的差别小于某一误差范围时，即完成了一次转换。

#### 2、ADC0809 的工作过程

ADC0809 将多路模拟开关和 8 位 ADC 集成在一个芯片内，便于构成多通道数据采集系统。对指定的通道采集一个数据的过程：

- （1）选择当前转换的通道，即将通道号编码送到 C、B 和 A 引脚上。
- （2）在 START 和 ALE 脚上加一个正脉冲，将通道选择码锁存并启动 A/D 转换。可以通过执行 OUT 指令产生负脉冲，经反相后形成正脉冲，也可由定时电路或可编程定时器提供启动脉冲。
- （3）转换开始后，EOC 变低，经过64个时钟周期后，转换结束，EOC 变高。
- （4）转换结束后，可通过执行 IN 指令，设法在 OE 脚上形成一个高电平脉冲，打开输出缓冲器的三态门，让转换后的数字量出现在数据总线上，并被读入累加器中。

#### 3、ADC0809 的工作过程

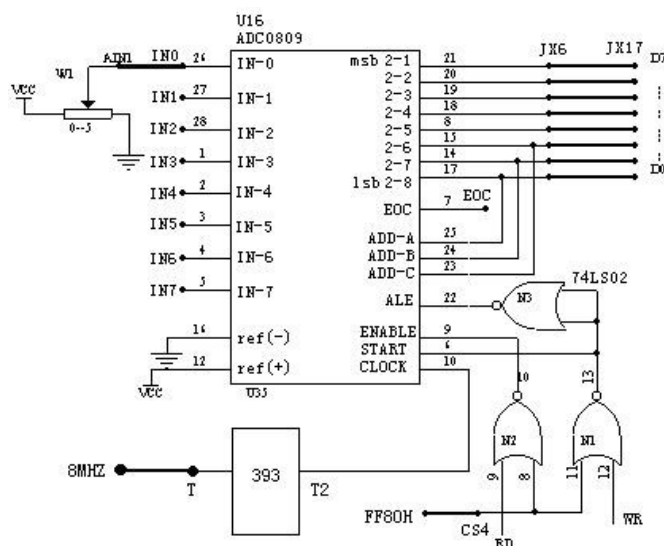
- （1）用定时中断控制采样率，用地址信号选择通道；
- （2）用8255A控制ADC0809；

### 2. 实验硬件连接（给出实验硬件线路图及连接说明）

#### 1、硬件线路图：

#### 2、实验硬件连接及步骤

- （1）将 0809CS4插孔连到译码输出FF80H插孔。
- （2）将通道0模拟量输入端IN0连电位器W1的中心插头 AOUT1(0—5V)插孔,8MHZ→T。



(3) 运行实验程序。

在 PC机和实验系统联机状态下,运行该实验程序,可用鼠标左键单击菜单栏“文件”或工具栏“打开图标”,弹出“打开文件”的对话框,然后打开598K8ASM文件夹,点击H0809.ASM文件,单击“确定”即可装入A/D转换实验源文件,再单击工具栏中编译装载,即可完成源文件自动编译、装载目标代码功能,再单击“调试”中“R 运行”或工具图标运行,即开始运行程序。

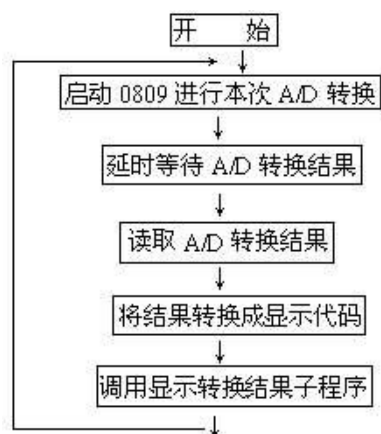
如果在没有 PC 机的情况下,则在 8086K 系统显示监控提示符“P.”时,用户段地址固定为 0000 (省略输入),按 SCAL键,输入起始偏移地址1000,按EXEC键,在系统上显示“0809 XX”。“XX”表示输入模拟量转换后的数字量。

(4) 调节电位器W1,显示器上会不断显示新的转换结果。模拟量和数字量对应关系的典型值为:

0V→00H+2.5V→80H +5V→FFH

(5) 按RST 键退出。

### 3. 实验程序流程图



### 4. 实验代码分析 (对从标号 ADCON 开始至指令 JMP ADCON 的代码部分中每一条指令进行注释)

```
ADCON:      MOV AX,00          ;选择通道 0
            MOV DX,ADPORT      ;0809 端口地址
            OUT DX,AL          ;启动 AD 转换
            MOV CX,0500H       ;延时,等待 AD 转换结束
DELAY:      LOOP DELAY
            MOV DX,ADPORT      ;0809 端口地址
            IN AL,DX           ;读 AD 转换值
            CALL CONVERS        ;AD 转换值拆送显示缓冲区
            CALL DISP           ;显示子程序
            JMP ADCON          ;循环继续
```

#### **5. 学生自我小结（所遇问题和个人体会等方面总结）**

通过本次实验，初步掌握 ADC0809 与微机接口的原理和方法，并且从实验中更加深入了解教材中所学的模/数转化原理。

# 实验 7 DAC0832 数/模转换接口实验

实验日期： 2023-12-22      实验成绩： \_\_\_\_\_ 教师签名： \_\_\_\_\_

## 一、实验名称

- 实验 7 DAC0832 数/模转换接口实验

## 二、实验目的

1. 了解数/模转换的基本原理。
2. 掌握 DAC0832 与微机接口的硬件设计原理和要求。
3. 熟悉 DAC0832 的使用方法；

## 三、实验装置

- 实验地点：信息楼 418 机房
- 实验设备及器材配置：DJ86PCI 实验箱、配套微机、DJ-8086 系列实验指导书

## 四、实验内容

利用 DAC0832 通过 D/A 转换交替产生方波和锯齿波。

## 五、实验要求

1. 学习并掌握硬件连接步骤与方法后，连接好硬件线路。
2. 阅读并理解实验程序 H0832-1.ASM 后，运行实验程序。
3. 实验结果，理解并分析实验原理和方法。
4. 撰写实验报告，要求包括以下内容：实验原理与方法、实验硬件连接电路图、程序流程图、实验代码分析、学生自我小结。

## 六、教师评语

1. 设计思路：[清晰、较清晰、一般、较差]
2. 电路选择：[合理、较合理、一般、较差]
3. 代码：[很好、较好、一般、较差] 实现了设计要求，功能完整
4. 实验报告：格式规范、内容完整、表达清晰准确 [很好、较好、一般、较差]
5. 总评价：

## 七、实验报告

### 1. 实验原理与方法（分析综述相关主要技术的原理与方法）

#### 1、DAC0832 简述

D/A 转换器(DAC)是把输入数字量转换为与输入量成比例的模拟信号的器件。多数 D/A 转换器把数字量(如二进制编码)变成模拟电流, 如要转换成模拟电压, 还要使用电流/电压转换器(I/V)。

NSC公司生产DAC0832,是一种内部带有数据输入寄存器的8 位 D/A 转换器,采用低功耗 CMOS 工艺制成,芯片内有 R-2R 梯形电阻网络,用于对参考电压产生的电流进行分流,完成模数转换,转换结果以一组差动电流 Iout1 和 Iout2 输出。它可直接与 8088、8086 等微处理器的总线相连。

主要参数为:

- 分辨率            8 位
- 转换时间        1  $\mu$  s
- 满量程误差     $\pm 1$ LSB
- 参考电压         $\pm 10$ V
- 单电源            +5V~+15V

#### 2、DAC0832 的工作方式

改变控制信号的时序和电平,可使 DAC0832 处于不同的工作方式:

##### (1) 直通方式

当 ILE 接高电平,CS、WR1、WR2 和 XFER 都接数字地时,使得输入寄存器和 DAC 寄存器均处于直通方式,8 位数字量一旦到达 DI7~DI0 输入端,就立即加到 8 位 D/A 转换器,被转换成模拟量。

##### (2) 单缓冲方式

只要把两个寄存器中的任何一个接成直通方式,而用另一个锁存数据,DAC 就可处于单缓冲工作方式。一般的做法是将 WR2 和 XFER 都接地,使 DAC 寄存器处于直通方式。另外,把 ILE 接高电平,CS 接端口地址译码信号,WR1 接 CPU 系统总线的 IOW 信号,这样便可通过执行一条 OUT 指令,选中该端口,使 CS 和 WR1 有效,将数据锁存到输入寄存器中,实现 D/A 转换。

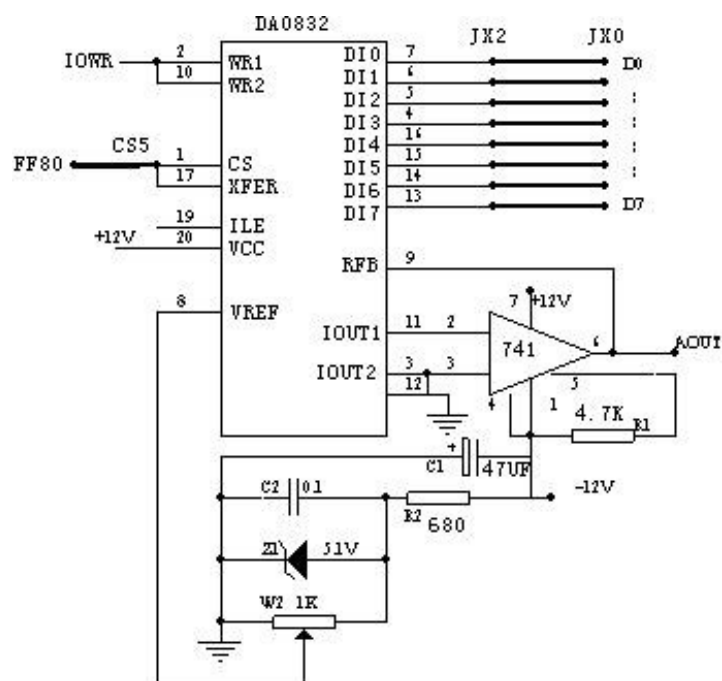
##### (3) 双缓冲方式

在这种情况下,需要在程序的控制下,把要转换的数据先打入输入寄存器,然后再在某个时刻启动 D/A 转换。这样可以做到对某数据转换的同时,能进行下一个数据的输入,因此转换速度较高。这时,可将 ILE 接高电平,WR1 和 WR2 接 CPU 的 IOW,CS 和 XFER 分别接两个不同的 I/O 地址译码信号。执行 OUT 指令时,WR1 和 WR2 均变低电平。这样,可先执行一条 OUT 指令,选中 CS 端口,把数据写入输入寄存器;再执行第二条 OUT 指令,选中 XFER 端口,把输入寄存器内容写入 DAC 寄存器,实现D/A 转换。



## 2. 实验硬件连接（给出实验硬件线路图及连接说明）

### 1、硬件线路图：



### 2、实验硬件连接及步骤

#### (1) 端口

• 首先须由 CS 片选信号确定量 DAC 寄存器的端口地址，然后锁存一个数据通过 0832 输出，典型程序如下：

```
MOV DX, DAPORT ; 0832 口地址
```

```
MOV AL, DATA ; 输出数据到 0832
```

```
OUT DX, A
```

• 产生波形信号的周期由延时常数确定。

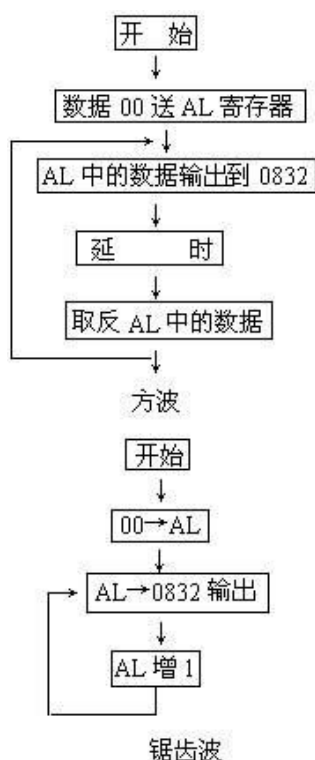
#### (2) 实验步骤

• 0832 片选信号 CS5 插孔和译码输出 FF80H 插孔相连。

• 运行实验程序，在系统显示监控提示符“P.”时，按 SCAL 键，待系统返 P 后，输入 10D0，再按 EXEC 键，在系统上显示执行符“┐”。

• 用示波器测量 0832 左侧 AOUT 插孔，应有方波和锯齿波交替输出。

### 3. 实验程序流程图



### 4. 实验代码分析（对从标号 ADCON 开始至指令 JMP ADCON 的代码部分中每一条指令进行注释）

```

START:  MOV AL,0FFH    ;设置 DA 转换数据
        MOV CX,0800H   ;波形个数

DACON1: MOV DX,DAPORT  ;写 0832 端口地址
        OUT DX,AL      ;启动 DA 转换
        PUSH CX        ;压栈
        MOV CX,0400H   ;延时
        LOOP $
        POP CX         ;出栈
        NOT AL         ;设置 DA 转换数据反码
        LOOP DACON1    ;未完继续
        MOV DX,DAPORT  ;写 0832 端口地址
        MOV AL,00H     ;设置 DA 转换数据
        MOV CX,0008H   ;波形个数
        MOV BX,0FFFFH  ;波形个数

DACON2: OUT DX,AL      ;启动 DA 转换
        INC AL         ;DA 转换数据加 1
  
```

```
DEC BX
CMP BX,0000H    ;未完继续
JNZ DAON2
LOOP DAON2      ;未完继续
JMP START       ;循环继续
```

#### 5. 学生自我小结（所遇问题和个人体会等方面总结）

通过本次实验，我基本了解数/模转换的基本原理。初步掌握了 DAC0832 的使用方法。

## 实验 8 综合设计型实验

实验日期： 2023-12-27      实验成绩： \_\_\_\_\_ 教师签名： \_\_\_\_\_

### 一、实验名称

- 实验 8 综合设计型实验

### 二、实验目的

通过对本设计课题的实施，使学生进一步掌握及运用所学的有关硬件课程知识，学以致用，锻炼和培养学生的动手能力以及开拓设计思路。

### 三、实验装置

- 实验地点：信息楼 418 机房
- 实验设备及器材配置：DJ86PCI 实验箱、配套微机、DJ-8086 系列实验指导书

### 四、实验内容及要求

使用 2 个及 2 个以上的接口芯片或其他元件组成相应的电路，实现某一功能电路，并编制相应的程序，调试运行最终达到设计要求，最后要求书写实验报告，要求内容完整、格式规范、逻辑清晰、设计合理，要有设计思路和个人总结并要达到本课程设计的要求。

如：定时检测开关状态并显示（采用 8255A、8253、8259 芯片实现，8253 周期性地每隔固定时间就产生一次中断，中断服务程序中检测开关状态并在指示灯上显示出来）、交通灯管理（采用 8255A、8253、8259 芯片实现）、定时采样（采用 ADC0809、8253、8259 芯片实现）、电子闹钟（用键盘输入倒计时的时间，用 8253 产生定时，定时时间到产生中断，8255 输出到数码管显示，同时喇叭发出声音）、电梯控制（模拟控制电梯，采用 8255A、8253、8259 芯片实现）。

### 五、教师评语

1. 设计思路：[清晰、较清晰、一般、较差]
2. 电路选择：[合理、较合理、一般、较差]
3. 代码：[很好、较好、一般、较差] 实现了设计要求，功能完整
4. 实验报告：格式规范、内容完整、表达清晰准确 [很好、较好、一般、较差]
5. 总评价：

## 六、实验报告

### 1. 实验原理与方法（分析综述相关主要技术的原理与方法）

交通灯管理（采用 8255A、8253、8259 芯片实现）,在“实验四 8255A 并行口实验（二）”的基础上加入定时功能。

#### 1) 8255A 芯片:

##### 1、8255A 芯片简介

8255A 是一种通用的可编程并行 I/O 接口芯片,它是为 Intel 系列微处理器设计的配套电路,也可用在其他的微处理器系统中。

通过对 8255A 进行编程,芯片可工作与不同的工作方式。在微型计算机系统中,用 8255A 做接口时,通常不需要附加外部逻辑电路,就可直接在 CPU 于外设之间提供数据通道。

8255A 具有 A、B、C 三个并行接口,均为 8 位的输入输出端口,通过外部的 24 根输入输出线与外设交换数据或进行通信联络。其中 A、B 口都可作为 8 位的输入或输出端口,C 口既可作为 8 位的输入或输出用,又可作为两个 4 位的输入输出用,常用来配合 A、B 口的工作,分别用来产生 A、B 口的输出控制信号以及输入 A 口和 B 口的端口状态信号。

##### 2、8255A 的控制字

8255A 有两类控制字:一类用于定义各端口的工作方式,称为方式选择控制字;另一类用于对 C 端口的任一位进行置位或复位操作,称为置位/复位控制字。D7 位为这两个控制字的标志位,为 1 时是前者,为 0 时是后者。

##### 3、8255A 的工作方式

8255A 能在以下三种方式下工作:

- 方式 0: 基本输入/输出方式,适用于不需要用应答信号的简单输入输出场合;
- 方式 1: 选通输入/输出方式,适用于查询和中断传送(通常为输出);
- 方式 2: 双向选通工作方式,只有 A 口可以工作于这种方式。

#### 2) 8253 芯片:

##### 1、8253 芯片简介

8253 是一种常用的计数器/定时器芯片,被称为可编程间隔定时器。

8253 内部具有 3 个独立的 16 位计数器通道,通过对它进行编程,每个计数器通道均可按 6 种不同的方式工作,并且都可以按二进制或十进制格式进行计数,其计数频率范围为 0-2MHz,用 +5V 单电源供电。

在微型计算机系统中,常需要用到 8253 实现定时功能。例如,在 IBM PC 机中,需要有这样的一个实时时钟以实现计时功能,还要求按一定的时间间隔对动态 RAM 进行刷新。此外扬声器的发声也是由定时信号来驱动的。在计算机实时控制和处理系统中,则应按一定的采样周期对处理对象进行采样,或定时检测某些参数等等,都需要定时信号。再者,在许多微机应用系统中,还会用到计时功能,需对外部事件进行计数。此外,8253 还适用于许多其它的情况,例如用作可编程方波频率产生器、分频器、程控单脉冲发生器等。

##### 2、8253 的 6 种工作方式

方式 0: 计数结束中断;方式 1: 可编程频率发生;方式 2: 频率发生器;方式 3: 方波频率发生器;方式 4: 软件触发的选通信号;方式 5: 硬件触发的选通信号。

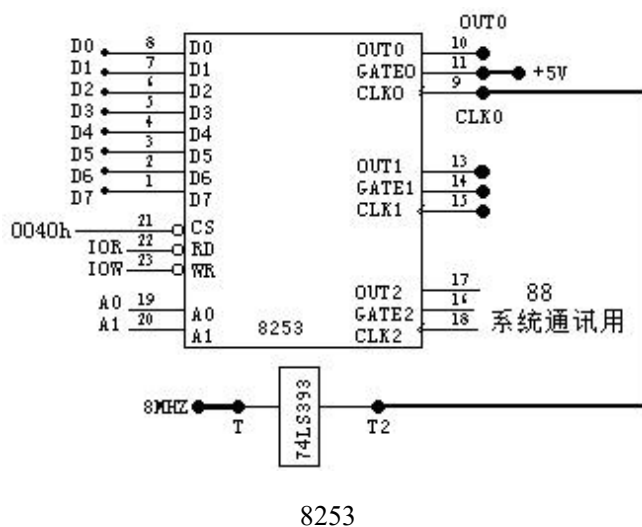
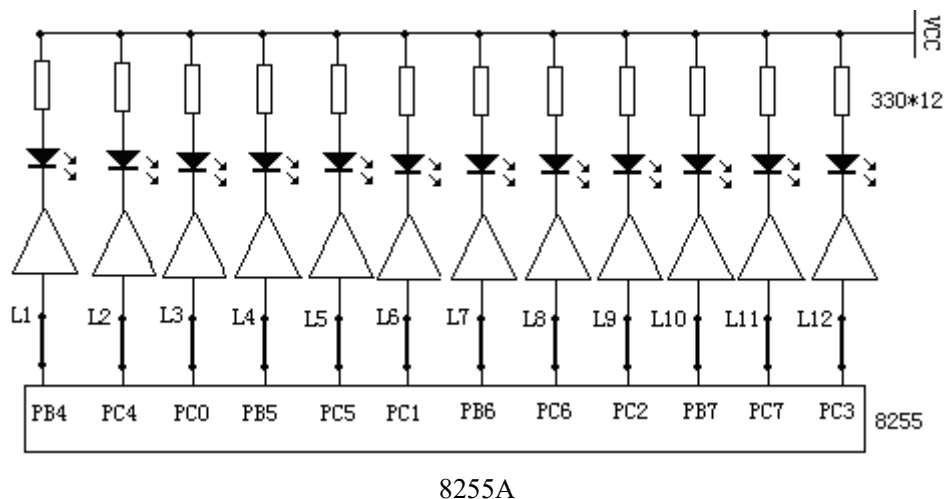
对于方式 0,在写入控制字后,输出端即变低,计数结束后,输出端由低变高,常用该输出信号作为中断源。其余 5 种方式写入控制字后,输出均变高。

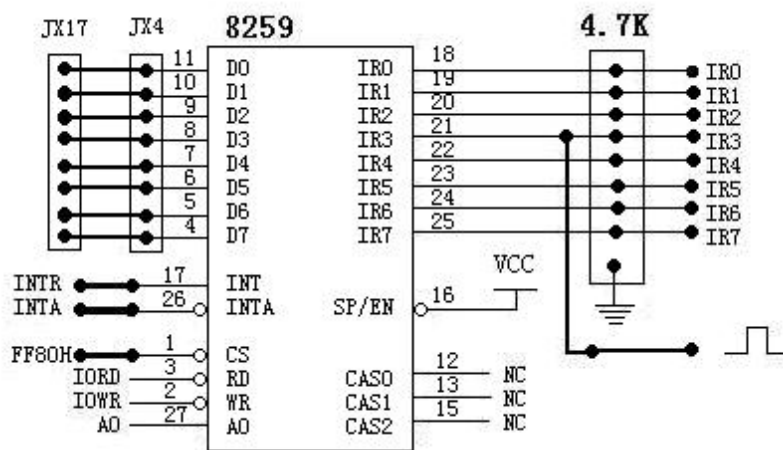
### 3) 8259 芯片:

中断控制器 8259A 是专为控制优先级中断而设计的芯片。它将中断源优先级排队、辨别中断源以及提供中断矢量的电路集于一片中。因此无需附加任何电路，只需对 8259A 进行编程，就可以管理 8 级中断，并选择优先模式和中断请求方式。即中断结构可以由用户编程来设定。同时，在不需要增加其它电路的情况下，通过多片 8259A 的级联，能构成多达 64 级的矢量中断系统。

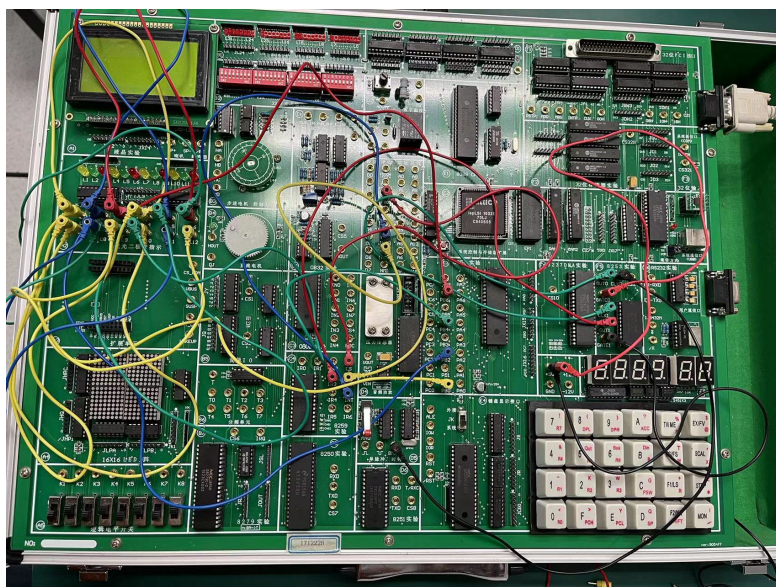
## 2. 实验设计思路（包括硬件、软件的设计图或说明等）

在“实验四 8255A 并行口实验（二）”的基础上加入定时功能。实验硬件连线如下图所示：





8259



实验线路连接图

### 3. 实验设计思路（包括硬件、软件的设计图或说明等）

使用 8259、8255 和 8253 芯片设计红绿灯控制电路的实验原理主要涉及三个部分：8259 芯片用于中断管理，8255 芯片用于并行输入输出控制，8253 芯片用于定时和计数。以下是对这三部分原理的详细解释：

#### 1. 8259 芯片（中断控制器）：

- 8259 芯片是一个可编程中断控制器，它可以管理多达 8 个中断源。在交通信号控制系统中，8259 可以用来处理各种中断请求，例如定时器中断、外部事件中断等。
- 通过设置 8259 的命令和控制寄存器，可以实现对中断的屏蔽、优先级控制等功能。这样，系统可以在适当的时候响应中断，执行相应的中断服务程序，如切换红绿灯、启动定时器等。

#### 2. 8255 芯片（并行输入输出控制器）：

- 8255 芯片提供并行输入输出功能，它可以用来控制交通信号灯的亮灭。在交通信号控制系统中，8255 通常用于输出控制，将 CPU 的命令输出到各个信号灯上。
- 8255 的输出端口可以设置为高电平有效或低电平有效，通过输出数据寄存器来控制各个

灯的状态。例如，可以通过设置 A 口来控制红灯，B 口来控制黄灯，C 口来控制绿灯。

### 3. 8253 芯片（定时器/计数器）：

- 8253 芯片是一个可编程定时器/计数器，它包含三个 16 位的计数器，可以用来实现定时和计数功能。在交通信号控制系统中，8253 可以用来实现精确的时间控制，如红灯持续时间、绿灯间隔时间等。

- 8253 有多种工作方式，包括计数结束中断、可编程频率发生、方波频率发生等。通过设置 8253 的控制字，可以配置它的工作方式和计数初值。例如，可以通过方式 3 来产生一个 1Hz 的方波信号，用于控制黄灯的闪烁。

结合以上三个芯片的功能，可以设计一个红绿灯控制电路。系统首先通过 8259 管理中断，当接收到外部事件中断时（如定时器中断），8255 根据预设的信号灯控制策略输出信号，控制红绿灯的亮灭。同时，8253 用于生成精确的时间控制信号，确保交通信号灯按照规定的时间间隔切换。通过这种方式，可以实现一个自动化、精确控制的交通信号系统。

## 4. 实验的实现（实现方法和代码）

CODE SEGMENT ;交通灯管理.asm

ASSUME CS:CODE

INTPORT1 EQU 0FF80H  
INTPORT2 EQU 0FF81H  
INTQ3 EQU INTREEUP3  
TCONTRO EQU 0043H  
TCON1 EQU 0041H  
TCON0 EQU 0040H  
IOCONPT EQU 0FF2BH  
IOCPT EQU 0FF2AH  
IOBPT EQU 0FF29H

ORG 12D0H

START: JMP s0

FIRST DB 0FEH

s0:

CLD  
MOV AX,0H  
MOV ES,AX  
MOV DI,002CH  
LEA AX,SUB1  
STOSW  
MOV AX,seg sub1  
STOSW



```

MOV AL,13H                                ;8259 初始化
MOV DX,INTPORT1
OUT DX,AL
MOV AL,08H
MOV DX,INTPORT2
OUT DX,AL
MOV AL,09H
OUT DX,AL
MOV AL,0F7H
OUT DX,AL

```

```

MOV AL,80H                                ;8255 初始化
MOV DX,IOCONPT
OUT DX,AL
MOV DX,IOBPT
MOV AL,FIRST
OUT DX,AL

```

```

MOV DX,TCONTRO                            ;8253 初始化
MOV AL,15H
OUT DX,AL
MOV DX,TCON0
MOV AL,00H
OUT DX,AL

```

```

MOV DX,TCONTRO
MOV AL,76H
OUT DX,AL
MOV DX,TCON1
MOV AX,2400                               ; 在此实现定时，代码中设置时间间隔 800 毫秒，可改为其他数值
OUT DX,AL
MOV AL,AH
OUT DX,AL

```

```

        STI
WAITING:
        JMP WAITING
        RET

SUB1:
        CLI

        ROL FIRST,2
        MOV DX,IOPBT
        MOV AL,FIRST
        OUT DX,AL

        MOV AL,20H
        MOV DX,INTPORT1
        OUT DX,AL
        STI
        IRET

CODE ENDS
END    START

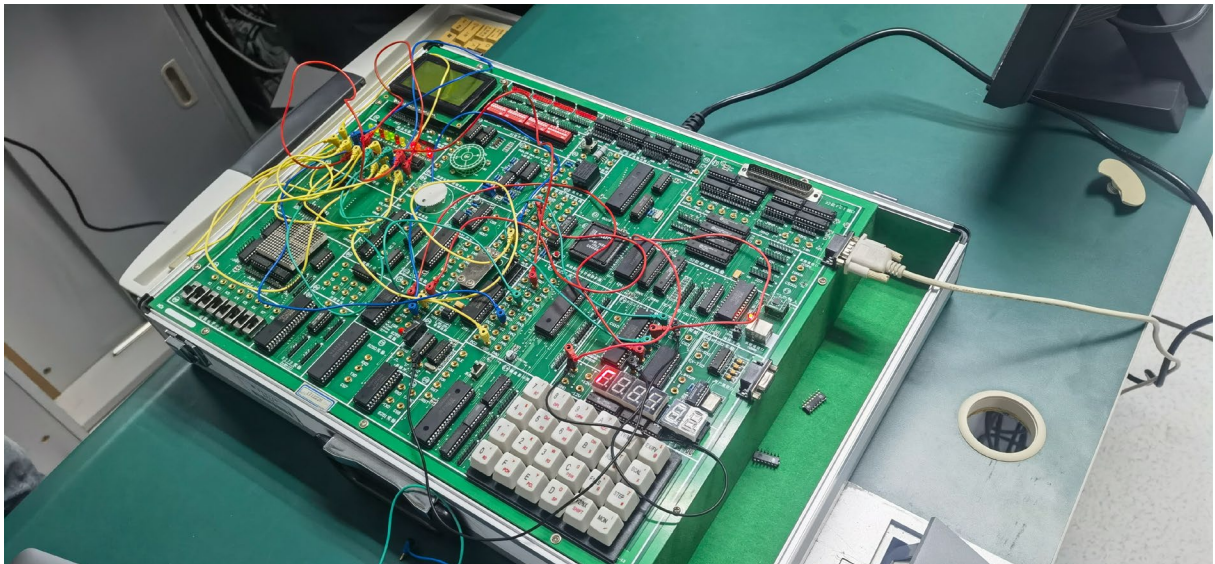
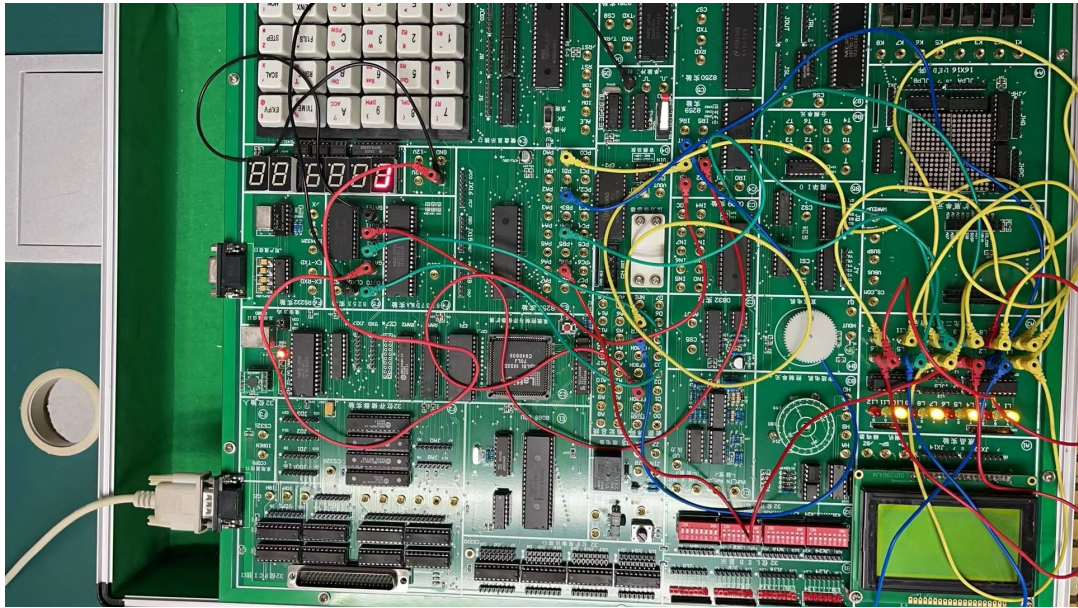
```

## 5. 学生自我小结（从设计、硬件连接和编程实现方面所遇到的问题和亲身体会等方面进行总结）

通过本次实验，我们根据所学的知识进行设计硬件连线。在综合实验中，我们小组使用了 8253，8259，8255 芯片进行设计交通灯管理系统。

通过本学期的实验，提升了我对于汇编语言控制硬件设施从而实现特定功能有了更为深刻的认知。在实验过程中，虽然遇到了较多的困难，但是在小组成员的合作下，成功解决了一系列问题，受益匪浅。最终成功完成实验。

实验结果如下：



##### 五、教师评语：

6. 设计思路：[清晰、较清晰、一般、较差]；
7. 电路选择：[合理、较合理、一般、较差]；
8. 代码：[准确，简洁，功能完整]，[很好、较好、一般、较差地] 实现了设计要求；
9. 实验报告内容：[完整、格式规范、逻辑清晰]，表达[很、较、一般、不]准确；
10. 表述自己所作工作：[清晰表述、较清晰表述、一般、较差]；
11. 回答教师提问：[准确、较准确、一般、不能回答]；

总评价：