微机原理与汇编语言课程

设计指导书

微机原理与汇编语言课程组 编

系	：	计算机科学与技术	计算机科学与技术
班级	：	计 203	计 203
学号	：		
姓名	：	Banban	
贡献度	：	60%	40%

xx 大学计算机学院

二〇二二 年 十二 月 三十一 日

目录

1 课设基本要求

1. 学生可以完成以下题目之一，经指导教师检查、验收、提交设计报告、评定成绩。

2. 学生也可以自拟题目进行设计，但需经指导教师审核同意。鼓励结合硬件资源思考选题，并上实验板调试。

3. 此次课程设计的 cpu 平台不作限定，8086、8051、stm32 均可接受，只要符合 cpu+接口电路+配套程序模式即可。

4. 可分组进行，每组 1~3 人。

5. 提交的课程设计报告应标注小组成员贡献度。设计报告应包括：封面（课题名称、专业班级、学号、姓名、成员贡献度）、目录、正文、参考文献等。其中正文部分包括：设计任务要求、系统原理框图、各部分电路原理分析及实现方法（或软件流程）、程序源代码、仿真结果（仿真平台可以选择 proteus）、设计讨论或心得体会。

6. XX 日前提交报告。

   报告电子版邮件发任课教师邮箱：XX @qq.com

   电子版标题和附件格式为：学号+姓名+微机课设报告.doc

2 交通信号灯实时控制系统设计

2.1 设计要求（技术要求）

设有一个十字路口 1、3 为南北方向，2、4 为东西方向，初始状态为四个路口的红灯全亮，之后，1、3 路口的绿灯亮，2、4 路口的红灯亮，1、3 路口方向通车。延时一段时间（如 30 秒）后，1、3 路口的绿灯熄灭，而 1、3 路口的黄灯开始闪烁，闪烁若干次（如 3 秒）以后，1、3 路口红灯亮，而同时 2、4 路口的绿灯亮，2、4 路口方向通车，延时一段时间后，2、4 路口的绿灯熄灭，而黄灯开始闪烁，闪烁若干次以后，再切换到 1、3 路口方向，之后，重复上述过程。示意图如图 2-1。

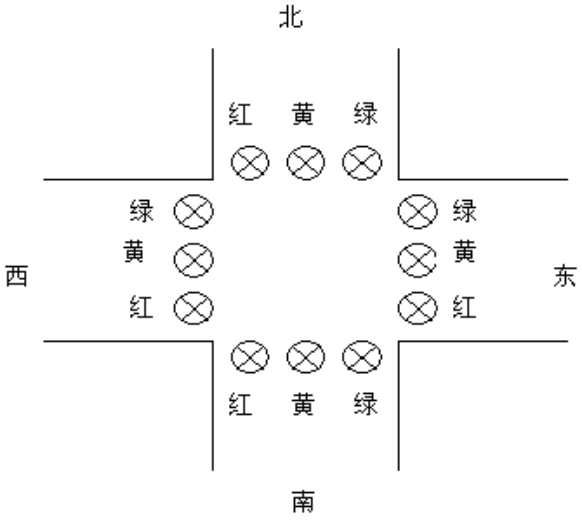


图 2-1　十字路口交通灯

2.2 设计提示

1. 可用实验仪上的 LED 发光二极管模拟主干道和支干道上的红、黄、绿三种信号灯；用 8255 驱动发光二极管。
2. 3 秒或 30 秒延时由定时器/计数器 8253 硬件定时结合软件计数的方法实现，一秒中断后再加软件计数实现。

2.3 需求分析

随着电子技术的发展，计算机在现代科学技术的发展中起着越来越重要的作用。多媒体技术、网络技术、智能信息处理技术、自适用控制技术、数据挖掘与处理技术等都离不开计算机。本课程设计是基于微机原理与接口技

术的简单应用。运用所学的微机原理和接口技术知识完成交通灯系统。通过硬件与软件的结合，用我们刚刚学过的汇编语言编写程序模拟分析了现代城市交通控制与管理问题的现状，结合交通的实际情况阐述了交通灯控制系统的工作原理，给出了一种简单实用的交通灯控制系统的硬件、软件电路设计方案。该系统适用于单主干道的十字路口。现假定其主干道为东西方向，次干道为南北方向。

2. 4 设计过程

2.4.1 设计过程简单分析

红，黄，绿灯可分别接在 8255 的 A 口上，灯的亮灭可直接由 8255 输出 0，1 控制。延时及闪烁由软件编程实现。

2.4.2 硬件原理

设计电路如图 2-2 所示：

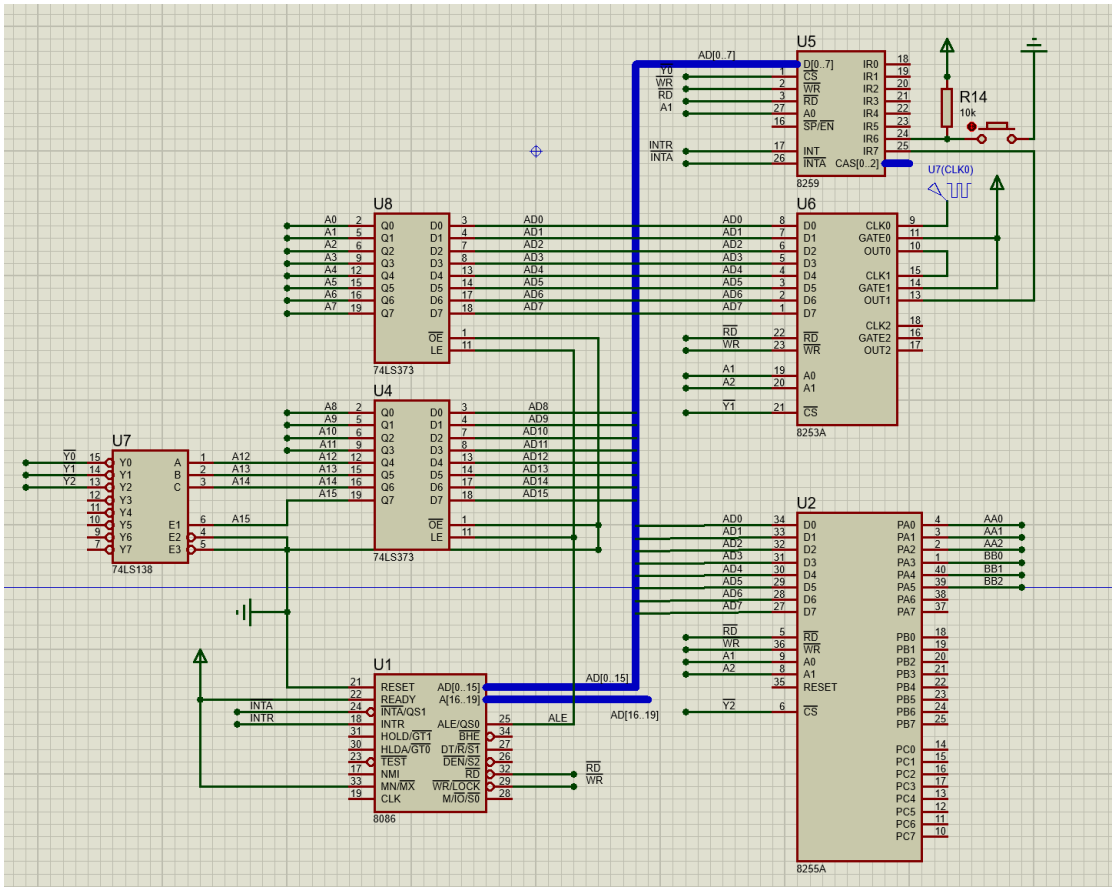


图 2-2 部分实验电路图

二极管设计电路如图 2-3 所示：

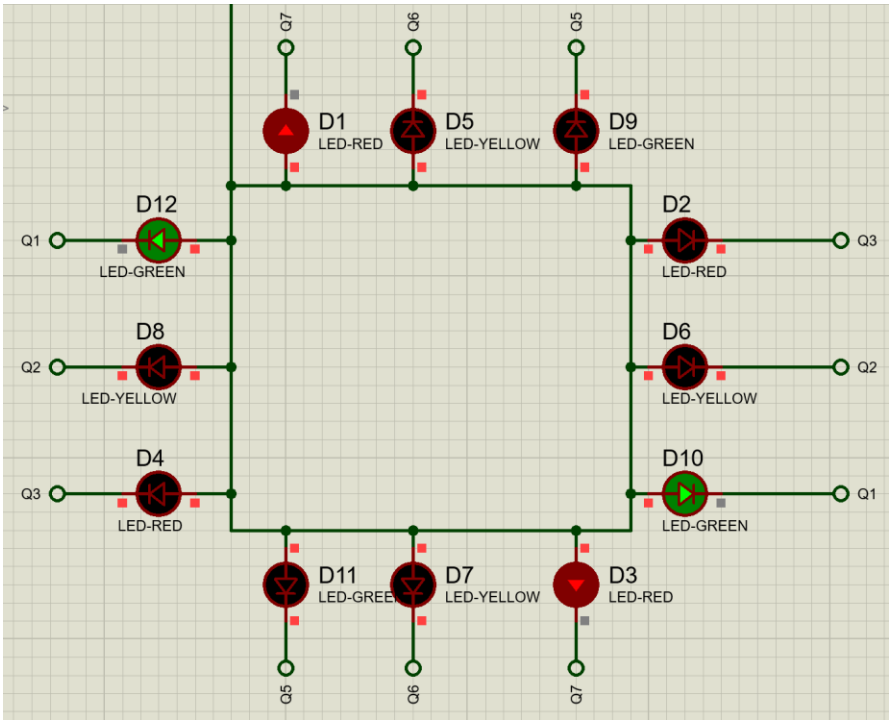


图 2-3 二极管设计电路图

2.4.3 主要芯片资料

8255A 芯片

1. 8255A 的内部结构图见图 2-4：

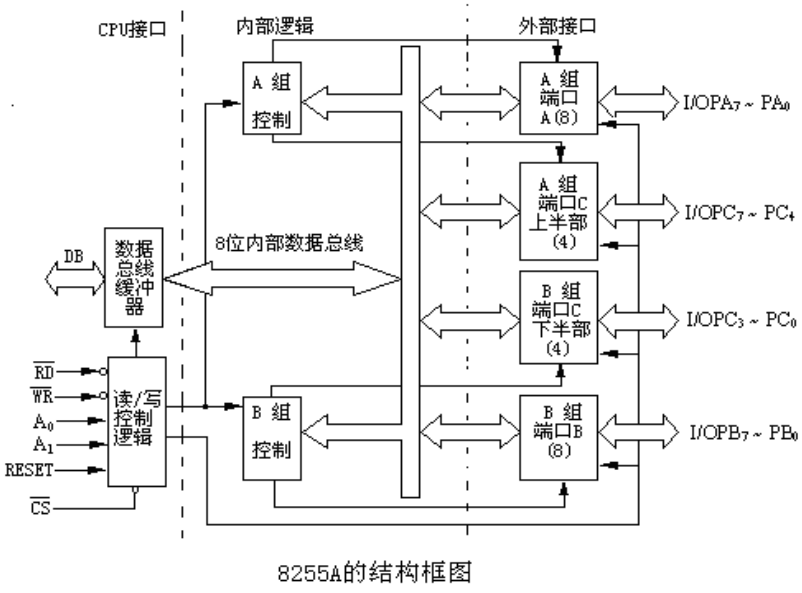


图 2-4 8255A 的内部结构图

2. 8255A 芯片介绍

8255A 是一个 40 引脚的双列直插式集成电路芯片，按功能可把 8255A 分为三个逻辑电路部分，即：口电路、总线接口电路和控制逻辑电路。

1） 数据口电路

8255A 共有三个 8 位口，其中 A 口和 B 口是单纯的数据口，供数据 I/O 使用。而 C 口则既可以作数据口，又可以作控制口使用，用于实现 A 口和 B 口的控制功能。

数据传送中 A 口所需的控制信号由 C 口高位部分（PC7～PC4）提供，因此把 A 口和 C 口高位部分合在一起称之为 A 组；同样理由把 B 口和 C 口低位部分（PC3～PC0）合在一起称之为 B 组。

2） 总线接口电路

总线接口电路用于实现 8255A 和单片微机的信号连接。其中包括：

- 数据总线缓冲器：数据总线缓冲器为 8 位双向三态缓冲器，可直接和 80C51 的数据线相连，与 I/O 操作有关的数据、控制字和状态信息都是通过该缓冲器进行传送。
- 读/写控制逻辑：与读写有关的控制信号有 CS—片选信号（低电平有效）、RD—读信号（低电平有效）、WR—写信号（低电平有效）A0、A1—端口选择信号。8255A 共有四个可寻址的端口（即 A 口、B 口、C 口和控制寄存器），用二位地址编码即可实现选择。

参见下表。

$\overline{CS}$	A1	A0	$\overline{RD}$	$\overline{WR}$	所选端口	操作
0	0	0	0	1	A 口	读端口 A
0	0	1	0	1	B 口	读端口 B
0	1	0	0	1	C 口	读端口 C
0	0	0	1	0	A 口	写端口 A
0	0	1	1	0	B 口	写端口 B
0	1	0	1	0	C 口	写端口 C
0	1	1	1	0	控制寄存器	写控制字
1	×	×	×	×	/	数据总线缓冲器输出高阻抗

RESET—复位信号（高电平有效）。复位之后，控制寄存器清除，各端口被置为输入方式。

读写控制逻辑用于实现 8255A 的硬件管理：芯片的选择，口的寻址以及规定各端口和单片微机之间的数据传送方向。

控制逻辑电路：控制逻辑电路包括 A 组控制和 B 组控制，合在一起构成 8 位控制寄存器。用于存放各口的工作方式控制字

3. 8255A 工作方式及数据 I/O 操作

1） 8255A 的工作方式

8255A 共有三种工作方式，即方式 0、方式 1、方式 2.

2） 方式 0 基本输入/输出方式

- 方式 0 下，可供使用的是两个 8 位口（A 口和 B 口）及两个 4 位口（C 口高 4 位部分和低 4 位部分）。四个口可以是输入和输出的任何组合。
- 方式 0 适用于无条件数据传送，也可以把 C 口的某一位作为状态位，实现查询方式的数据传送。

3） 方式 1 选通输入/输出方式

- A 口和 B 口分别用于数据的输入/输出。而 C 口则作为数据传送的联络信号。具体定义见表 7－2。可见 A 口和 B 口的联络信号都是三个，如果 A 或 B 只有一个口按方式 1 使用，则剩下的另外 13 位口线仍然可按方式 0 使用。如果两个口都按方式 1 使用，则还剩下 2 位口线，这两位口线仍然可以进行位状态的输入输出。
- 方式 1 适用于查询或中断方式的数据输入/输出。

4） 方式 2 双向数据传送方式

只有 A 口才能选择这种工作方式，这时 A 口既能输入数据又能输出数据。在这种方式下需使用 C 口的五位线作控制线，信号定义如表 7－2 所示。方式 2 适用于查询或中断方式的双向数据传送。如果把 A 口置于方式 2 下，则 B 口只能工作于方式 0。

C 口位线	方式 1		方式 2	
	输入	输出	输入	输出
PC_7		$\overline{OBFA}$		$\overline{OBFA}$
PC_6		$\overline{ACKA}$	$\overline{ACKA}$	
PC_5	IBFA		IBFA	
PC_4	$\overline{STBA}$		$\overline{STBA}$	
PC_3	INTRA	INTRA	INTRA	INTRA
PC_2	$\overline{STBB}$	$\overline{ACKB}$		
PC_1	IBFB	$\overline{OBFB}$		
PC_0	INTRB	INTRB		

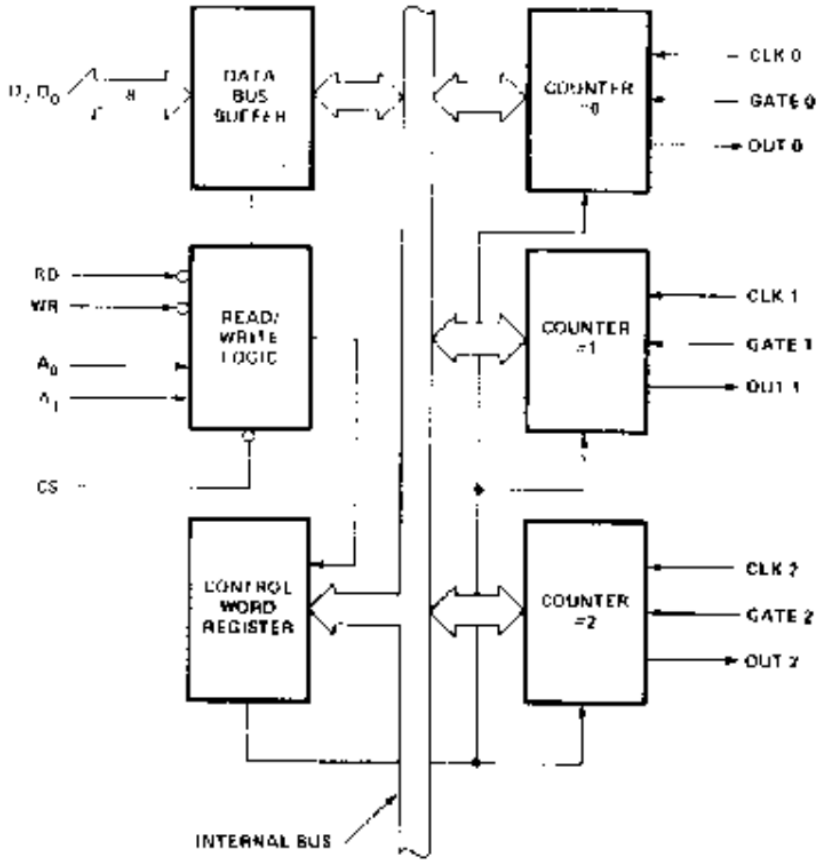
表 7-5 8255A C 口联络信号定义

8253A 芯片

1. 8253 介绍

计数器 0、计数器 1、计数器 2 是三个完全独立、结构相同的计数器，每一个都由一个 16 位的可预置的减法计数器构成。[1]8253 的每个计数器都有 6 种工作方式，这 6 种工作方式的主要区别在于输出的波形不同，计数过程中 GATE 信号对计数操作的影响不同，启动计数器的触发方式不同等。

2. 8253 结构图



3. 8253 端口地址

信号线	寄存器	编址
IOY2	定时器 0	40H
	定时器 1	41H
	定时器 2	42H
	控制寄存器	43H

4. 8253 控制字如下

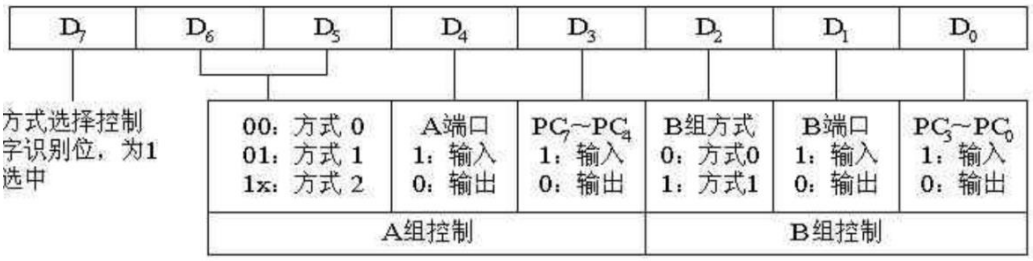
SC1	SC0	RW1	RW2	M2	M1	M0	BCD

选择计数器
00：计数器 0
01：计数器 1
10：计数器 2
11：非法

00：锁存
01：只读/写低 8 位
10：只读/写高 8 位
11：先读/写低 8 位
再读/写高 8 位

模式选择
000：模式 0
001：模式 1
X10 模式 2
X11 模式 3
100：模式 4
101：模式 5

1：BCD 码
0：用二进制



8259 芯片

1. 8259A 功能介绍

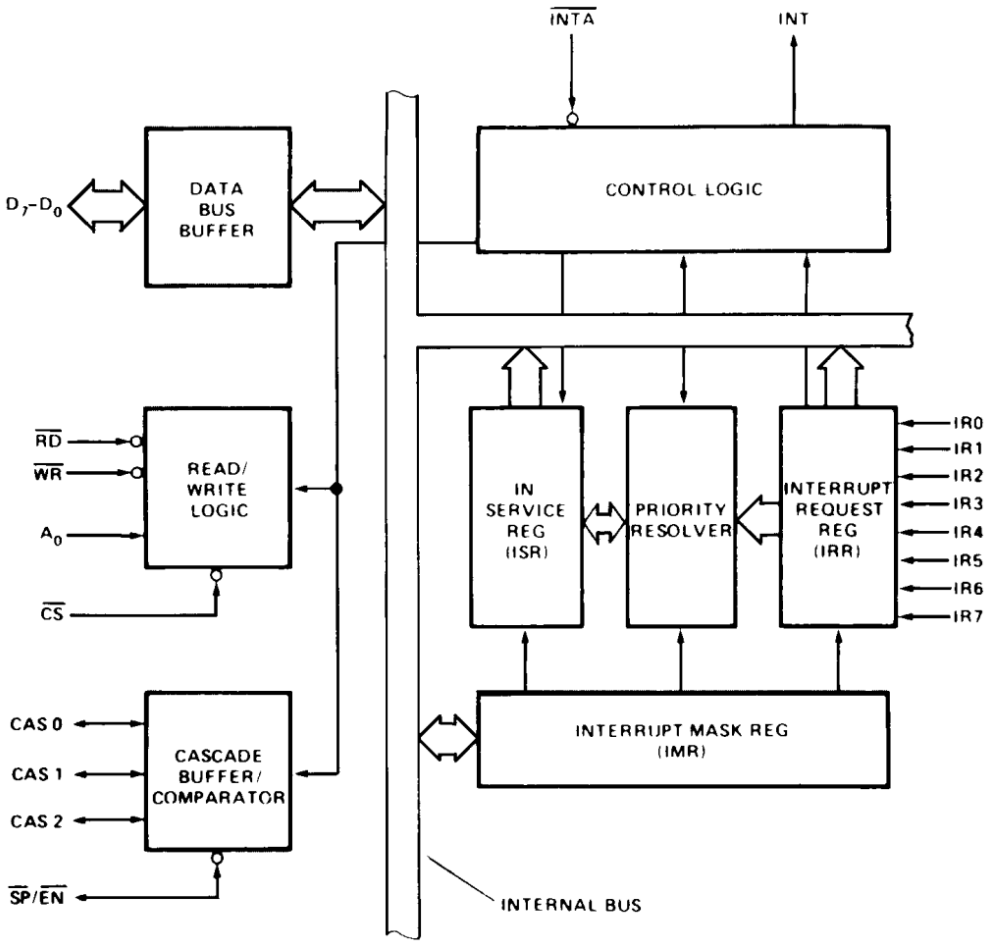
(1)可以直接管理 8 个中断原，及联方式下不用附加电路就可以管理 64 个可屏蔽中断原，并具有优先权判决功能。

(2)能为中断原提供中断向量码。

(3)可以对每一及中断进行屏蔽控制。

(4)可提供多种可供选择的工作方式，并能通过编程进行控制。

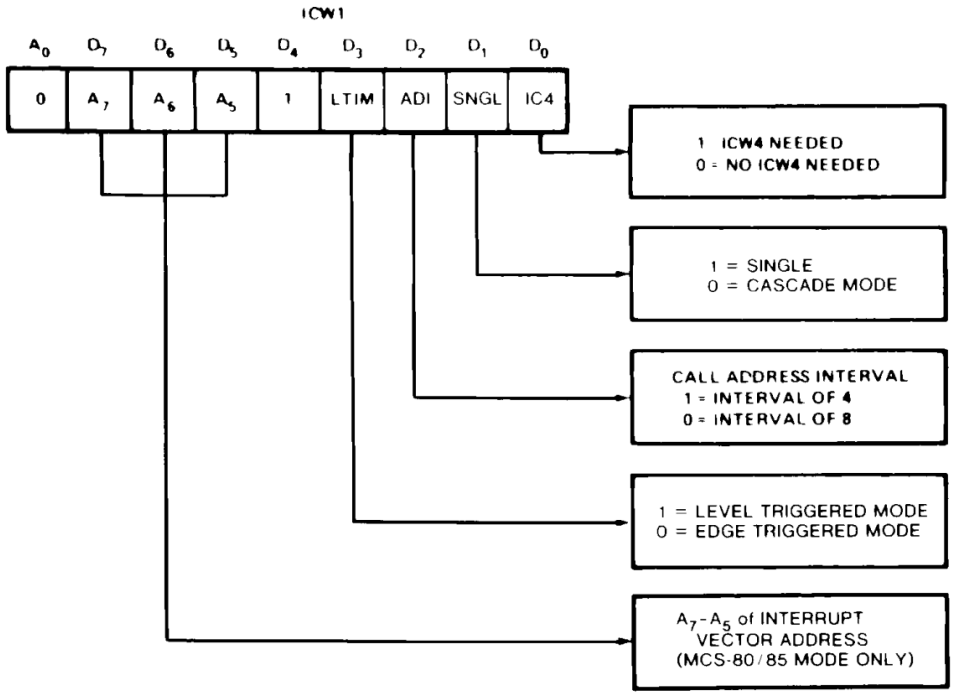
2. 8259A 内部结构图

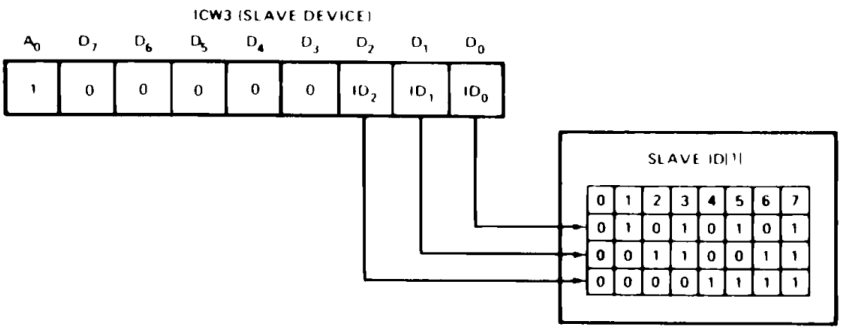


3. 8259 的工作方式

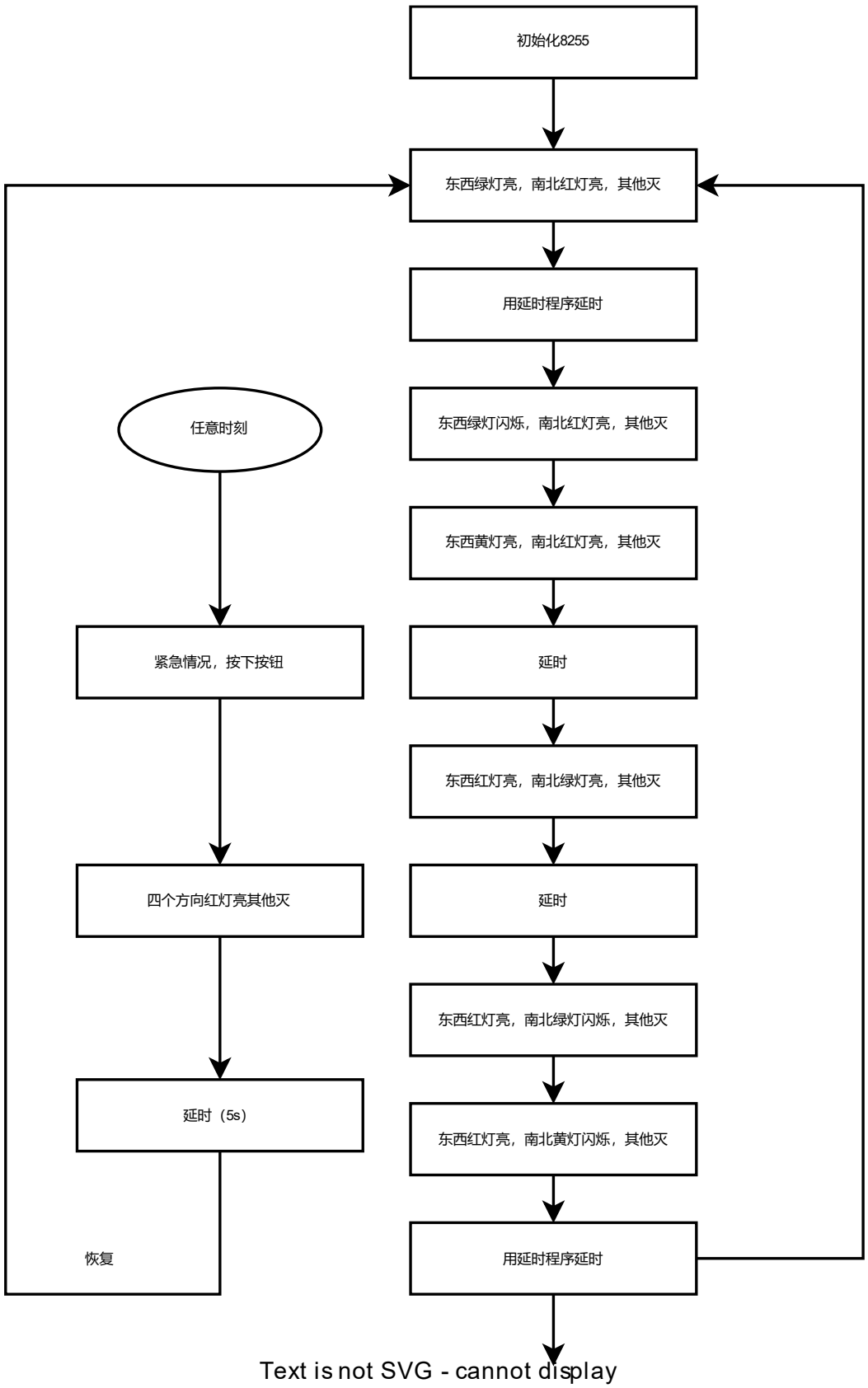
- 中断优先权方式，8259A 中断优先权的管理方式有固定优先权方式和自动循环优先权方式两种。
- 中断嵌套方式，8259A 的中断嵌套方式分为完全嵌套和特殊完全嵌套两种。
- 中断屏蔽方式，中断屏蔽方式是对 8259A 的外部中断源 IR7～IR0 实现屏蔽的一种中断管理方式，有普通屏蔽方式和特殊屏蔽方式两种。
- 中断结束方式，中断结束方式是指 CPU 为某个中断请求服务结束后，应及时清除中断服务标志位，否则就意味着中断服务还在继续，致使比它优先级低的中断请求无法响应。
- 中断触发方式，8259A 中断请求输入端 IR7～IR0 的触发方式有电平触发和边沿触发两种，由初始化命令字 ICW1 中的 LTIM 位来设定。
- 总线连接方式，8259A 数据线与系统数据总线的连接有缓冲和非缓冲两种方式。如果 8259A 数据线与系统数据总线直接相连，那么 8259A 工作在非缓冲方式控制字。

4. 8259 的部分控制字





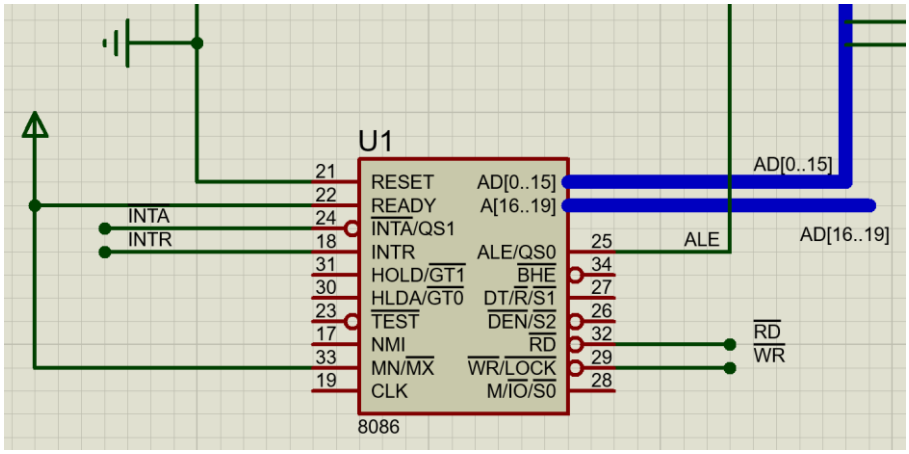
2.4.4 程序流程图



Text is not SVG - cannot display

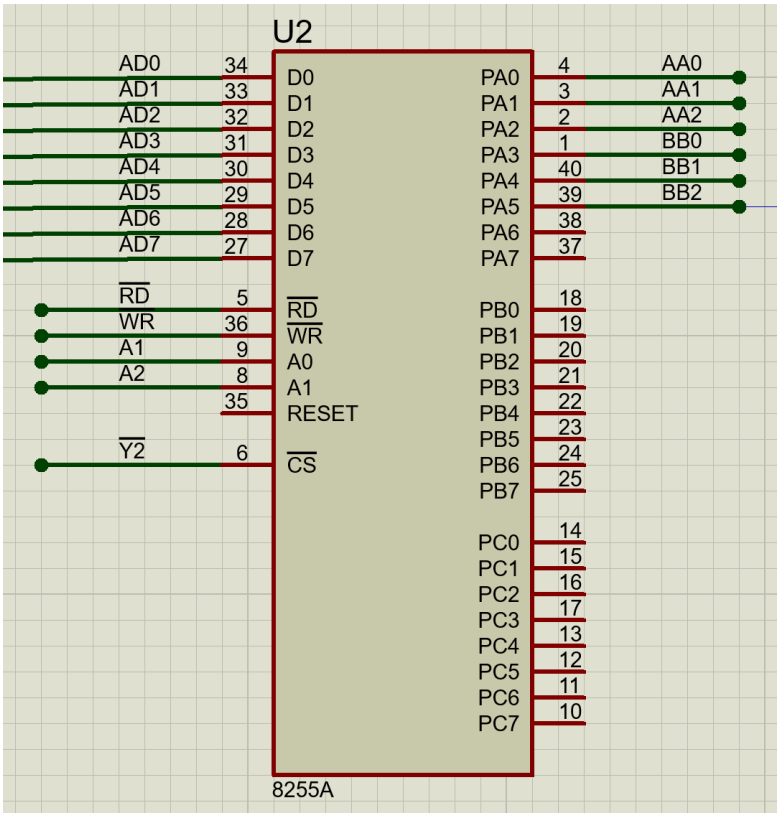
2.4.5 部分电路图

8086 电路



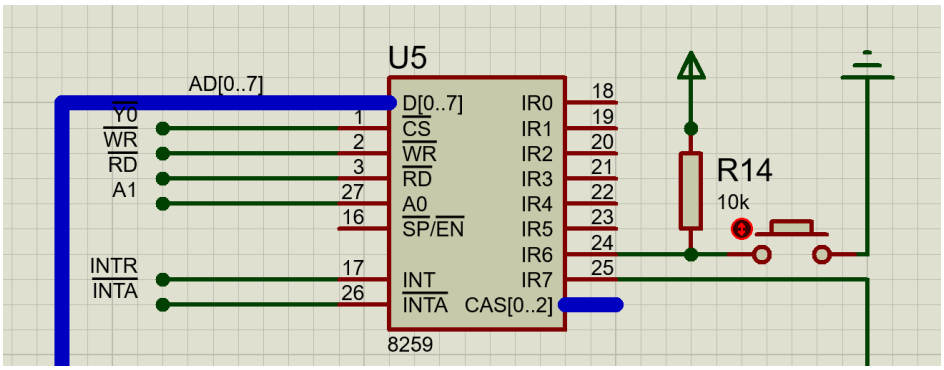
8086[2]最小系统由 Intel 8086 微处理器、74273 TTL 带公共时钟复位八 D 触发器、以及 74154 TTL4 线—16 线译码器等组成。8086 有 20 位地址线，其中高 4 位 A19-A16 与状态线 S6-S3 分时复用，低 16 位 AD15-AD0 与数据线分时复用。

8255A 电路



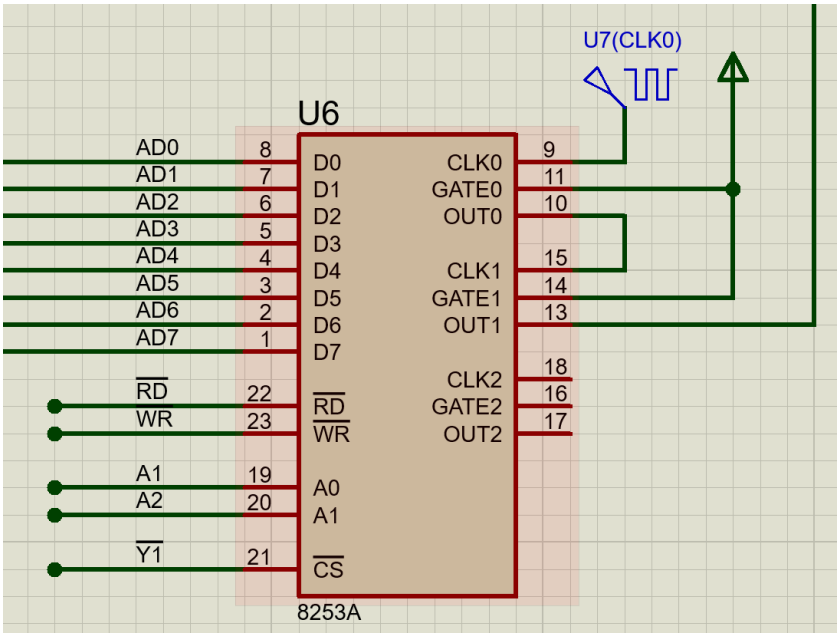
对于外围电路，采用一片 8255A 来驱动发光二极管。8255A 是专为 INTEL 公司的微处理器配套的接口芯片，8255A 为可编程芯片，可用程序设定改变其工作状态，CPU 通过它直接与外设相连接。8255A 各端口的正常工作需要事先写入控制寄存器的方式控制字即 8255A 的初始化编程。

8059 电路



外部开关接 8259 的 IR0 端，它的中断优先级别高，设置成边缘触发，当按下开关，产生一个高电平的脉冲，就可以停止原来的状态来执行更高级别的中断子程序，这就可以处理紧急情况，让主干道和支干道都变成红灯。

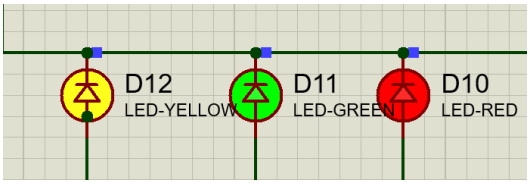
8083A 电路



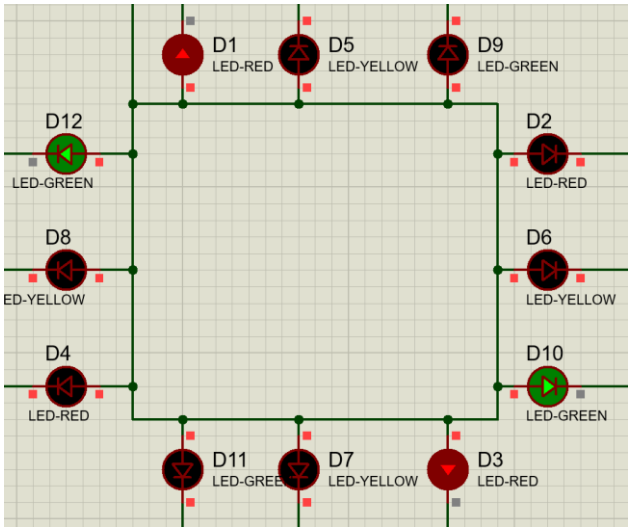
在 8253A 的使用中，其输入信号频率不能超过 2MHz，否则长时间使用，芯片过热，容易烧毁。而一个计数通道单独使用，其可写入的最大计数初值为 65536，则起最大计时长为 32768us，范围过小，两个计数通道串联使用，其最大计时长可满足需要。

2.4.6 仿真实现结果

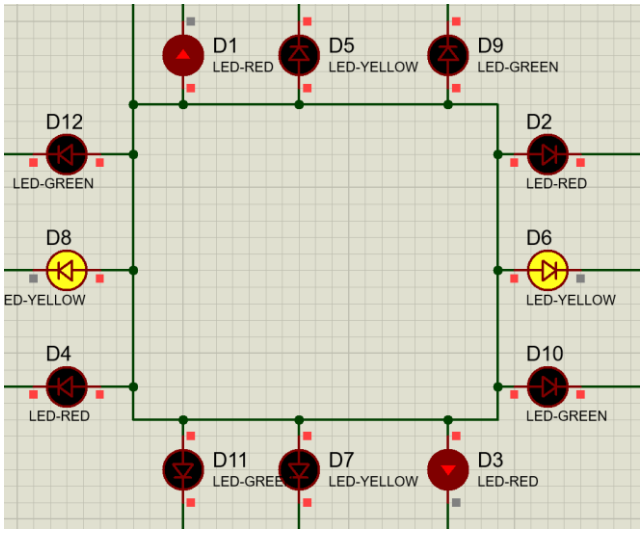
1. 用发光二极管 LED 灯模拟交通信号，用 LED 红灯模拟红灯，用 LED 绿灯模拟绿灯，用 LED 黄灯模拟黄灯。



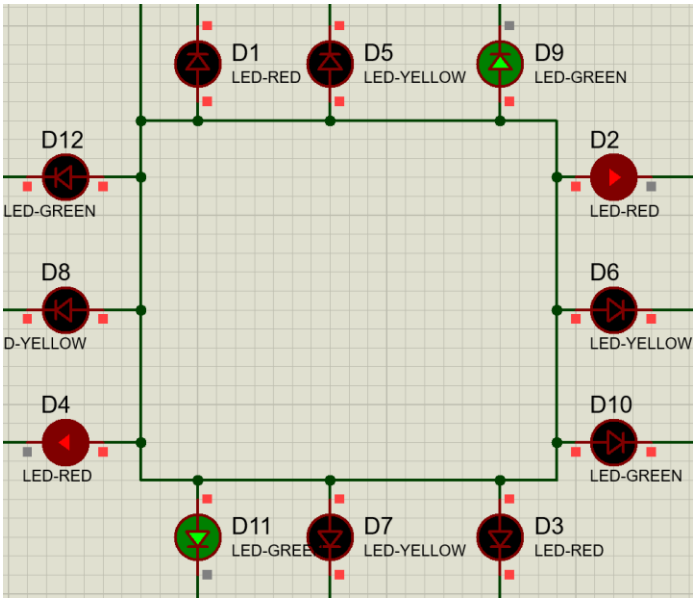
2. 正常情况下，A、B 两车道轮流放行，A 车道绿灯放行，B 车道红灯亮禁止通行，A 允许通行时间为 10s，其中 7s 为绿灯。



当秒数变为 3s 时，绿灯熄灭，黄灯亮起，用于警告，B 道为 10s 的红灯禁止通行，计时结束，A 道的黄灯熄灭，红灯亮起禁止通行，B 道的红灯熄灭，绿灯亮起允许通行。[3]

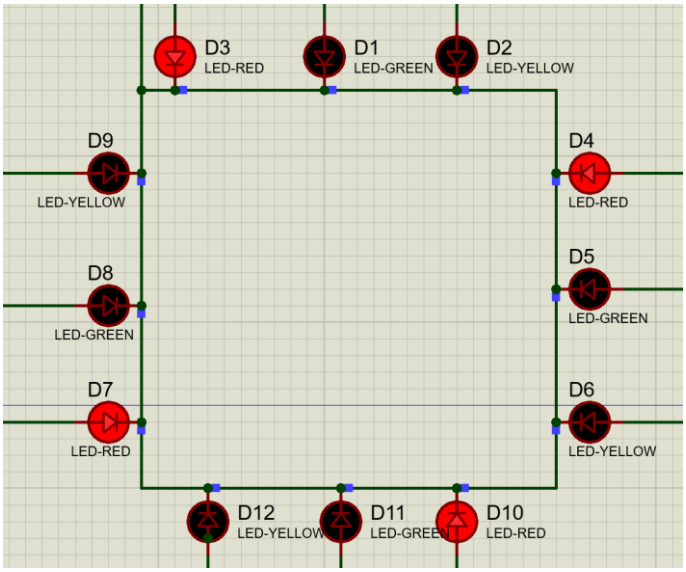


A 道为 10s 红灯禁止通行，B 道 7s 绿灯，当还剩 3s 之后，B 道绿灯熄灭，黄灯亮起，用于警告，计时结束，B 道黄灯熄灭，红灯亮起，A 道红灯熄灭，绿灯亮起。



如此循环。

3. 扩展功能：在紧急情况下，按下 Botton，实现紧急情况，从 5s 开始进行计数，回到正常时候刚跳变时刻重新开始正常情况下的计数。



3 设计中的问题

1. 问题：如何在一个中断的基础上嵌入另一个中断。
   解决方法：通过改变 8259 的控制字 OCW1 进行开中断 00111111，因为 IMRi=1 时，该位就被屏蔽，IMRi=0 时，且该位优先级高的画，则允许中断。
2. 问题：如何在紧急情况下让二极管跳变为紧急情况变成只有红灯亮起。
   解决方法：在 IR6 中断服务子程序中加入此段查表代码用 8255 的 B 口和 C 口进行输出。

4 总结

本次课程设计是要设计一个交通灯系统，主要功能如上已有细述。在本次对交通灯的设计过程中以此来加深对微机接口技术的理解，提高了自己的动手能力。首先着手对硬件电路的设计，本次课程设计主要采用了 8255A 接口电路。由于对各个芯片不熟悉，通过课本了解到了它们的引脚及功能、工作方式、内部结构和控制字。

然后就是对程序的设计,想要设计出一个实用的控制系统需要了解程序流程，先画出了流程图，然后对代码进行编写，在编写过程中遇到了很多问题。对芯片的不了解也导致编程的很多的问题，要么灯都不显示，要么灯显示不全，再要么红灯绿灯时间分配不合理。最后经过一段时间的研究，查阅了很多资料并和老师同学讨论后终于一一解决。

最后，由于自己太晚才开始做这个课程设计，导致很多功能没有实现好，例如想通过 8253 来对交通灯进行定时计数，但是最后都由于时间紧而选择软件实现延时闪烁。不管怎样，经过这次课程设计，我获益颇多。将微机原理这门课程中的理论与实践相结合起来，对芯片的功能也有了进一步认识理解。

感谢老师和另一位同学的帮助，才有今天这份微机课设报告，正值新年将至，也祝愿各位指导老师在新的一年里身体健康！兔年吉祥！

5 参考文献

[1] 李现国,张艳.Proteus 仿真在微机原理及接口技术教学中的应用[J].实验技术与管理,2010,27(12):125-127.DOI:10.16791/j.cnki.sjg.2010.12.039.

[2] Liu, Yu-Cheng, and Glenn A. Gibson. Microcomputer systems: The 8086/8088 family: Architecture, programming, and design. Prentice-Hall, Inc., 2000.

[3] 刘心红,郭福田,孙振兴,曾丽丽.Proteus 仿真技术在单片机教学中的应用[J].实验技术与管理,2007(03):96-98+102.DOI:10.16791/j.cnki.sjg.2007.03.029.

6 附录

源代码

```
DATA SEGMENT
  CS8259A EQU 8000H
  CS8259B EQU 8002H
  ICW1 EQU 00010011B
  ICW2 EQU 00100000B
  ICW4 EQU 00000001B
  OCW1 EQU 00111111B
  ;8253
  CNT0 EQU 9000H
  CNT1 EQU 9002H
  CNT2 EQU 9004H
  CTL  EQU 9006H
  ;8255
  A_PORT EQU 0A000H
  B_PORT EQU 0A002H
  C_PORT EQU 0A004H
  CT_PORT EQU 0A006H
  FLAG DW 20
  TAB DB 3FH,06H,5BH,4FH,66H,6DH,
7DH,07H,7FH,67H,77H,7CH,39H,5EH
,79H,71H
DATA ENDS

STACK SEGMENT STACK
  STA DB 512 DUP(0FFH)
  TOP EQU $-STA
STACK ENDS

CODE SEGMENT PUBLIC 'CODE'
  ASSUME
CS:CODE,DS:DATA,SS:STACK
  ORG 100H
START:
  MOV AX,DATA
  MOV DS,AX
  MOV AX,STACK
  MOV SS,AX
  MOV AX,TOP
  MOV SP,AX
  CLI
  PUSH DS
  MOV AX,0
  MOV DS,AX
  MOV BX,156
  MOV AX,OFFSET INT7
  MOV [BX],AX
  MOV AX,0
  MOV [BX+2],AX
  POP DS
  PUSH DS
```

```
MOV AX,0
MOV DS,AX
MOV BX,26H*4
MOV AX,OFFSET INT6
MOV [BX],AX
MOV AX,0
MOV [BX+2],AX
POP DS

;8259 初始化
MOV DX,CS8259A
MOV AL,ICW1
OUT DX,AL
MOV DX,CS8259B
MOV AL,ICW2
OUT DX,AL
MOV AL,ICW4
OUT DX,AL
MOV AL,OCW1
OUT DX,AL
STI

;8255 初始化
MOV AL,80H
MOV DX,CT_PORT
OUT DX,AL
MOV AL,00
MOV DX,A_PORT
OUT DX,AL

;8253 初始化
MOV DX,CTL
MOV AL,00110110B
OUT DX,AL
MOV DX,CNT0
MOV AX,1000
OUT DX,AL
MOV AL,AH
OUT DX,AL
MOV DX,CTL
MOV AL,01110110B
OUT DX,AL
MOV DX,CNT1
MOV AX,1000
  OUT DX,AL
  MOV AL,AH
  OUT DX,AL

LP:
  NOP
  STI
  JMP LP

INT6:
  CLI
  PUSH AX
  PUSH BX
  PUSH CX
  PUSH DX
  STI
  MOV DX,A_PORT
  MOV AL,00001001B
  OUT DX,AL
  MOV CX,5

NEXT:
  LEA BX,TAB
  ADD BX,CX
  MOV AL,[BX]
  MOV DX,B_PORT
  OUT DX,AL
  LEA BX,TAB
  ADD BX,CX
  MOV AL,[BX]
  MOV DX,C_PORT
  OUT DX,AL
  CALL DELAY100
  LOOP NEXT
  MOV DX,CS8259A
  MOV AL,20H
  OUT DX,AL
  CLI
  POP DX
  POP CX
  POP BX
  POP AX
  OUT DX,AL
  STI
```

```
  IRET

;延时程序
DELAY100 PROC
  PUSH CX
  MOV CX,0
  LOOP $
  LOOP $
  LOOP $
  MOV CX,15000
  LOOP $
  POP CX
  RET
DELAY100 ENDP

INT7:
  CLI
  PUSH AX
  PUSH BX
  PUSH CX
  PUSH DX
  MOV CX,FLAG

 MOV AL,[BX]
  MOV DX,C_PORT
  OUT DX,AL
  MOV DX,B_PORT
  OUT DX,AL
  JMP EXIT

LEDB:
  LEA BX,TAB
  ADD BX,CX
  MOV AL,[BX]
  MOV DX,B_PORT
  OUT DX,AL
  LEA BX,TAB
  ADD BX,CX
  MOV AL,[BX]
  MOV DX,C_PORT
  OUT DX,AL
  JMP EXIT

A0:
  CMP CX,0
  JG B0
  MOV FLAG,20

B0:
  DEC FLAG
  CMP CX,10
  JAE A0
  CMP CX,3
  JBE B1
  MOV AL,00010001B
  MOV DX,A_PORT
  OUT DX,AL
  JMP LEDB

B1:
  MOV AL,00100001B
  MOV DX,A_PORT
  OUT DX,AL
  LEA BX,TAB
  ADD BX,CX


  SUB CX,10
  CMP CX,3
  JBE A1
  MOV AL,00001010B
  MOV DX,A_PORT
  OUT DX,AL
  JMP LEDA

A1:
  MOV AL,00001100B
  MOV DX,A_PORT
  OUT DX,AL
  LEA BX,TAB
  ADD BX,CX
  MOV AL,[BX]
  MOV DX,C_PORT
  OUT DX,AL
  MOV DX,B_PORT
  OUT DX,AL
  JMP EXIT
```

```
LEDA:
  LEA BX,TAB
  ADD BX,CX
  MOV AL,[BX]
  MOV DX,C_PORT
  OUT DX,AL
  LEA BX,TAB
  ADD BX,CX
  MOV AL,[BX]
  MOV DX,B_PORT
  OUT DX,AL

EXIT:
  MOV DX,CS8259A
  MOV AL,20H
  OUT DX,AL
  CLI
  POP DX
  POP CX
  POP BX
  POP AX
  STI
  IRET
CODE ENDS
  END START
```