

华中科技大学
人工智能与自动化学院

**微机原理实验五：
8255 控制交通灯实验**

彭杨哲

U201914634

2021 年 11 月 25 日

1 实验目的

- 了解 8255 芯片的工作原理，熟悉其初始化编程方法以及输入、输出程序设计技巧。学会使用 8255 并行接口芯片实现各种控制功能，如本实验（控制交通灯）等。
- 熟悉 8255 内部结构和与 8086 的接口逻辑，熟悉 8255 芯片的 3 种工作方式以及控制字格式。
- 认真预习本节实验内容，尝试自行编写程序，填写实验报告。

2 实验内容

- 编写程序：使用 8255 的 PA0..2、PA4..6 控制 LED 指示灯，实现交通灯功能。
- 连接线路验证 8255 的功能，熟悉它的使用方法。

3 实验原理图

如图1所示

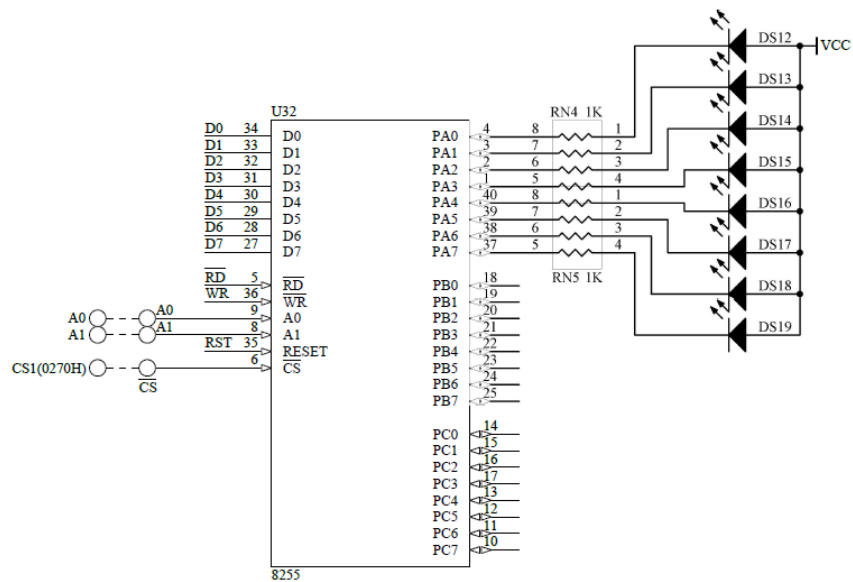


Figure 1: 实验原理图

4 实验步骤

1. 连线说明，如表1所示

D3 区：CS、A0、A1	A3 区：CS1、A0、A1
D3 区：JP23 (PA 口)	F4 区：JP18

Table 1: 连线表

2. 观察实验结果，是否能看到模拟的交通灯控制过程。

5 实验结果

5.1 实验代码

```
1 ;8255交通灯
2 CTRL_COM EQU 0273H;控制端口地址
3 PA_COM EQU 0270H;A口地址
4 PB_COM EQU 0271H;B口地址
5 PC_COM EQU 0272H;C口地址
6
7 _STACK SEGMENT STACK ;堆栈段
8 DW 100 DUP(?) ;定义100个字空间
9 _STACK ENDS
10
11 DATA SEGMENT
12 LED_DATA DB 10111110B;左边绿,右边红
13 DB 10111111B;左边绿灯闪,右边红
14 DB 10111101B;左边黄,右边红
15 DB 11101011B;左边红,右边绿
16 DB 11111011B;左边红,右边绿灯闪
17 DB 11011011B;左边红,右边黄
18 BUFFER DB 100 DUP(0) ;定义数据缓冲区
19 DATA ENDS
20
21 CODE SEGMENT
22 MAIN PROC NEAR
23
24 ASSUME CS:CODE, DS:DATA, SS:_STACK;定义堆栈段地址
25 MOV AX,DATA;加载数据段地址
26 MOV DS,AX;将数据段地址加载到DS
27 MOV ES,AX;设置ES为DATA的地址
28 MOV AX,_STACK;加载堆栈段地址
29 MOV SS,AX;设置SS为STACK的地址
30
31 CALL Init_8255;初始化8255
32
```

```

33     MOV DX,PA_COM;PA口地址
34     LEA BX,LED_DATA;初始状态地址
35
36 LOOP_1:
37     ;左边
38     MOV AL,[BX];初始状态
39     OUT DX,AL;写入初始状态
40     CALL Delay_6s;延时六秒
41
42     MOV AL,[BX];送入初始状态
43     MOV AH,1[BX];送入闪烁状态
44     CALL Blink;闪1.5s
45
46     MOV AL,2[BX];黄灯
47     OUT DX,AL;亮黄灯
48     CALL Delay_3s;黄灯3s
49
50
51     ;右边
52     MOV AL,3[BX];初始状态
53     OUT DX,AL;写入初始状态
54     CALL Delay_6s;延时六秒
55
56     MOV AL,3[BX];送入原始状态
57     MOV AH,4[BX];送入闪烁状态
58     CALL Blink;闪1.5s
59
60     MOV AL,5[BX];黄灯
61     OUT DX,AL;亮黄灯
62     CALL Delay_3s;黄灯3s
63
64     JMP LOOP_1;循环
65
66
67 MAIN ENDP
68
69 Init_8255 PROC NEAR
70     MOV DX,CTRL_COM;控制端口
71     MOV AL,80H;控制字
72     OUT DX,AL;写入控制字
73 ENDP
74
75
76 ;按照输入进行闪烁
77 ;@IN:AL:原始状态,AH:闪烁状态,DX:输出端口地址
78 ;@OUT:无
79 Blink PROC NEAR
80     OUT DX,AL;输出原始状态
81
82     CALL Delay_0s5;稍微延时一下
83     XCHG AH,AL;把高位换到低位来

```

```

84      OUT DX,AL;输出闪烁状态
85      CALL Delay_0s5;
86      CALL Delay_0s5
87      XCHG AH,AL;把低位换回到低位来
88      OUT DX,AL;输出原始状态
89
90      CALL Delay_0s5;延时相应的时间
91      CALL Delay_0s5
92      XCHG AH,AL;把高位换到低位来
93      OUT DX,AL;输出闪烁状态
94      CALL Delay_0s5;
95      CALL Delay_0s5
96      XCHG AH,AL;把低位换回到低位来
97      OUT DX,AL;输出原始状态
98  ENDP
99
100 ;延时 6s
101 ;@IN: 无
102 ;@OUT: 无
103 Delay_6s PROC NEAR
104     CALL Delay_3s;延时到相应的时间
105     CALL Delay_3s
106 ENDP
107
108 ;延时 3s
109 ;@IN: 无
110 ;@OUT: 无
111 Delay_3s PROC NEAR
112     CALL Delay_0s5;延时到相应的时间
113     CALL Delay_0s5
114     CALL Delay_0s5
115     CALL Delay_0s5
116     CALL Delay_0s5
117     CALL Delay_0s5
118 ENDP
119
120 ;延时 0.5s
121 ;@IN: 无
122 ;@OUT: 无
123 Delay_0s5 PROC NEAR
124     PUSH CX;保存CX值
125     MOV CX,3000H;500ms对应的周期数
126     LOOP $ ;循环
127     POP CX;恢复CX值
128     RET
129 Delay_0s5 ENDP
130
131 CODE ENDS
132 END MAIN

```

5.2 观察实验结果

烧写入代码后，可以看出最左侧的绿色 LED 灯和最右侧的红色 LED 灯亮起。然后经过几秒后绿色 LED 灯开始闪烁，几秒后左侧黄灯亮起，几秒后左侧红灯亮起右侧绿灯亮起，然后几秒后右侧绿灯开始闪烁，几秒后右侧亮起黄灯，再几秒后右侧红灯亮起左侧绿灯亮起，开始一个新的循环。

6 思考题

1. 如何对 8255 的 PC 口进行位操作？

答：先对控制端口写入控制字，然后每次 IN，OUT 时端口地址使用 0272H 作为输入和输出端口地址通过对 8255A 的控制口写入置位/复位控制字，可使端口 C 的任意一个引脚的输出单独置 1 或置 0。例如，写入控制字 01H，即可将 PC0 端口置为 1，写入控制字 03H，即可将 PC1 端口置为 1，即 D3 到 D1 为位选择，D0 为置位值。

2. 绘制本实验的详细实验电路图。

答：如图2所示

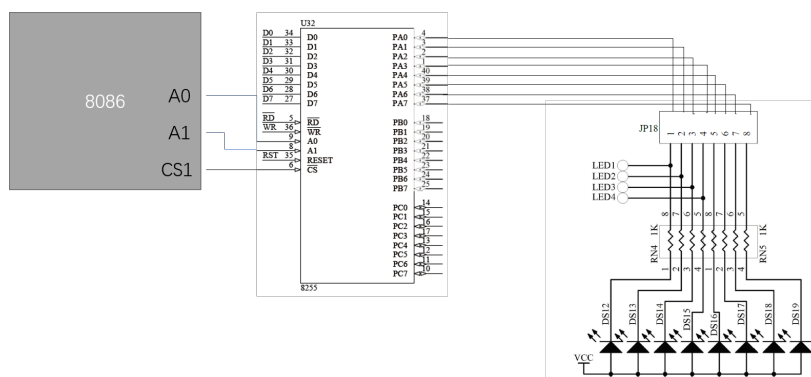


Figure 2: 详细实验电路图

3. 注释每各条指令的功能

答：详细注释见代码部分