

中国矿业大学计算机学院

2017 级本科生实验报告

课程名称 微机原理与接口实验

实验题目 8255 并行接口实验

学生姓名 袁孝健

学 号 06172151

专业班级 信息安全 2017-01 班

任课教师 王莉

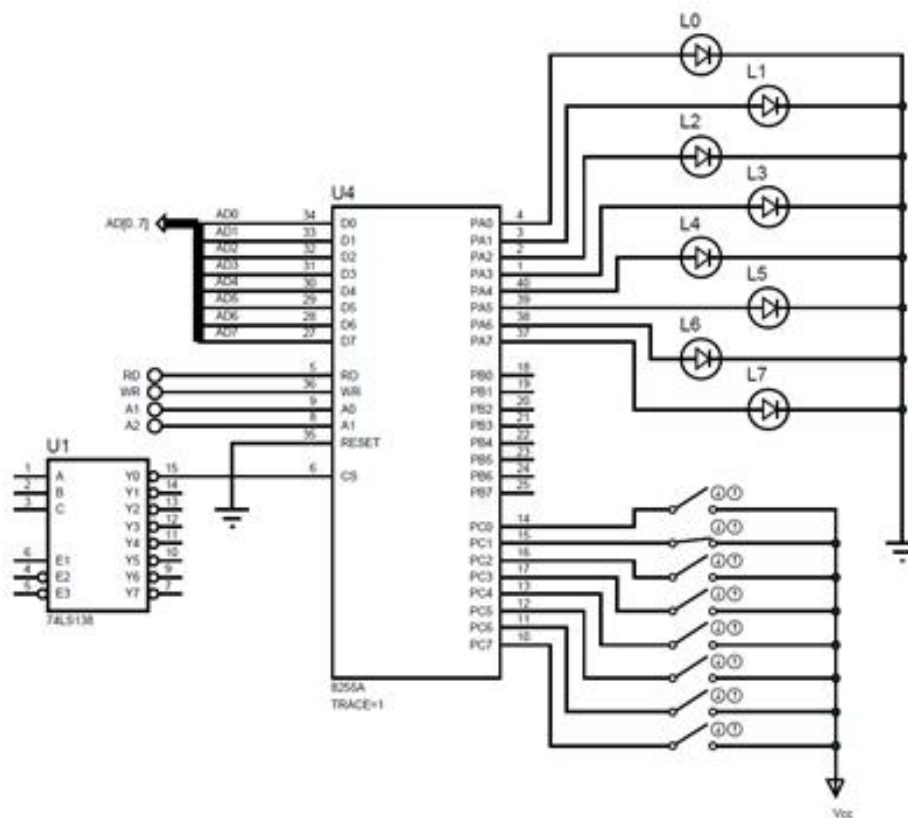
实验一

一、实验任务：

1. 8255 输入、输出实验
2. 利用 8255 设计跑马灯（软延时）

二、实验目的：熟悉汇编语言编程环境，掌握 8255 芯片工作原理、电路设计及利用汇编语言编程。

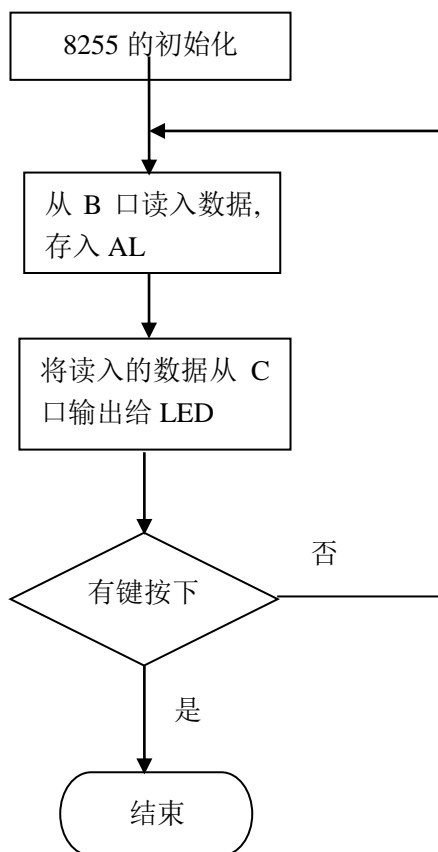
三、实验电路图



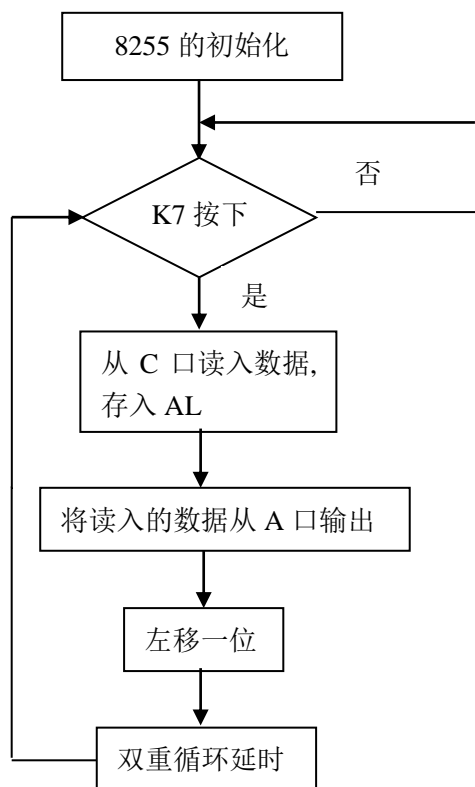
PC7~PC0 连 K7~K0，PA7~PA0 连 L7~L0，CS 连 Y0

四、软件设计程序流程图

1. 8255 输入、输出实验



2. 利用 8255 设计跑马灯（软延时）



五、实验软件代码（加注释）

1. 8255 输入、输出实验

```
1. CODE SEGMENT
2. ASSUME CS:CODE
3. START: MOV AL,10000010B    ; 写控制字
4.      MOV DX,2ABH          ; 控制字地址
5.      OUT DX,AL
6. TEST: MOV DX,2A9H
7.      IN AL,DX              ; 读开关状态
8.      MOV DX,2AAH
9.      OUT DX,AL             ; 将开关状态输出到 LED
10.     JMP TEST              ; 返回开始循环
11. CODE ENDS
12. END START
```

2. 利用 8255 设计跑马灯（软延时）

```
1. ;端口地址(起始地址 2A0)
2. PORTA EQU 2A0H
3. PORTB EQU 2A1H
4. PORTC EQU 2A2H
5. PORT_CRO EQU 2A3H
6.
7. CODE SEGMENT
8. ASSUME CS:CODE
9. START:MOV DX,PORT_CRO
10.     MOV AL,10001001B
11.     OUT DX,AL           ;AL 输出到 PORT_CR0
12.     MOV AL,01H          ;AL 赋初值（第一个灯亮）
13.     MOV AH,AL           ;将 AL 的值先暂存到 AH
14. TEST1:MOV DX,PORTC      ;读入 C 端口状态
15.     IN AL,DX            ;将 C 端口状态输入到 DX
16.     CMP AL,80H          ;用最高位开关控制跑马灯的开关
17.     JNZ TEST1           ;若相减不等于 0，即开关为低位，则跳转回 TEST 循环检测
18.     MOV AL,AH           ;将暂存在 AH 的初值赋给 AL
19.     MOV DX,PORTA        ;对 A 口输出状态
20.     OUT DX,AL           ;将 AL 的值输出到端口 A，第一次即第一个灯亮
21.     ROL AL,1            ;AL 中的值循环左移（1 位）
22.
23.     MOV BX,300H         ;外层循环，BX 赋值为 300H(此处可根据情况调整)
```

```

24. ANGIN1:
25.      MOV CX,65535      ;内存循环, CX 赋值为 65535
26. ANGIN2:
27.      LOOP ANGIN2      ;LOOP 指令 CX 自减循环
28.      DEC BX            ;内存循环执行完毕, BX 自减
29.      JNZ ANGIN2        ;外层循环若未执行完毕, 则继续循环
30.
31.      MOV AH,AL          ;将移位后的 AL,再暂存到 AH
32.      JMP TEST1          ;继续循环(若开关保持高位, 循环到 MOV AL,AH 语句)
33. CODE ENDS
34. END START

```

六、实验结果及体会

1. 实验结果

(1) 8255 输入、输出实验

成功实现了用 8 个开关分别控制 8 个 LED 灯的开关以验证了 8255 输入输出接口的情况。

(2) 利用 8255 设计跑马灯（软延时）

用 K7 来进行开关控制，实现了跑马灯功能，当 K7 按下后，8 个二极管由最右端 L0 以间隔 1 秒逐个向左发光一直到 L7，L7 发光结束后返回 L0 继续循环发光；关闭 K7，跑马灯停止循环。

2. 收获体会

本次实验是第一次对芯片进行编程，通过对 8255 的编程，初步掌握了 8255 编程方法，对此芯片有了更加深入的了解；同时也加强了对汇编语言的使用，认识到了汇编语言在底层硬件的重要作用。在实验的过程中边调试边纠错，也锻炼了我的实验动手能力，收获很多。

中国矿业大学计算机学院

2017 级本科生实验报告

课程名称 微机原理与接口实验

实验题目 8253+8255 实验

学生姓名 袁孝健

学 号 06172151

专业班级 信息安全 2017-01 班

任课教师 王莉

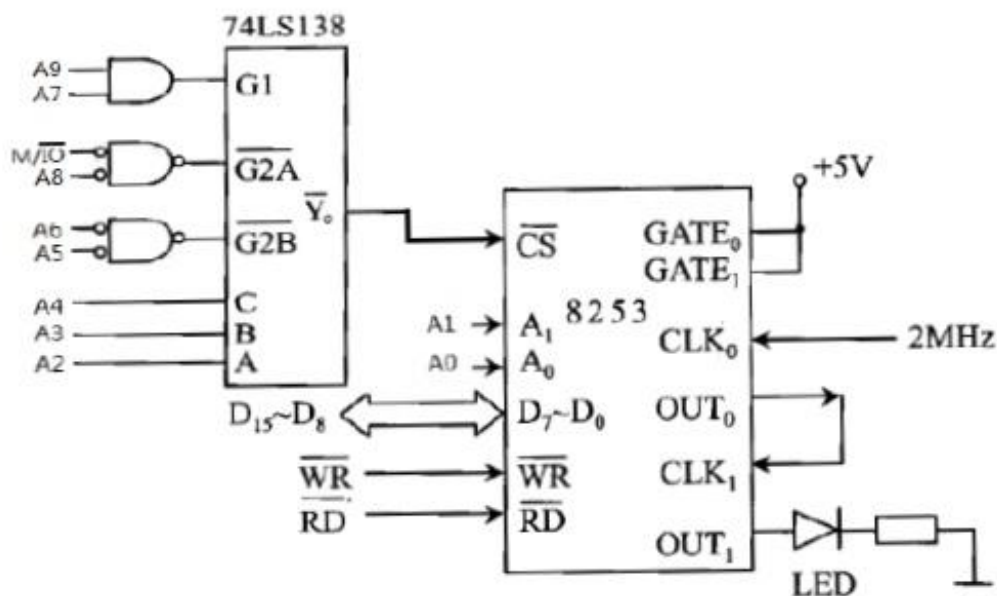
实验二

一、实验题目：

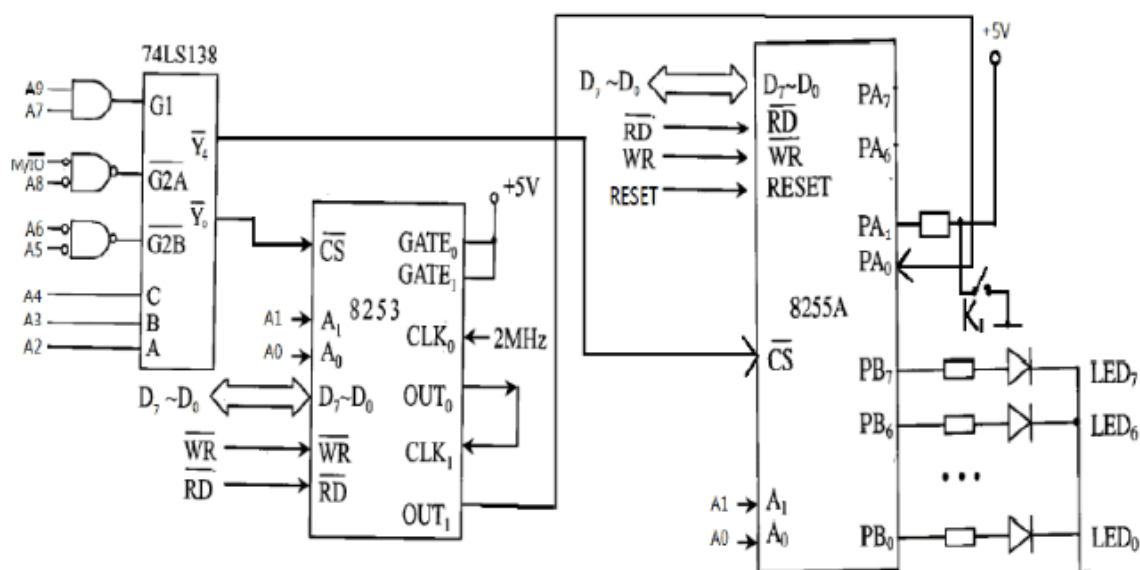
1. 8253 定时实验，利用 8253 完成 1 秒的延时。
2. 利用 8253 硬件延时控制跑马灯运行。
3. 选做 8255 键盘显示实验

二、实验目的：熟悉汇编语言编程环境，掌握 8253 芯片工作原理、电路设计及利用汇编语言编写接口软件。

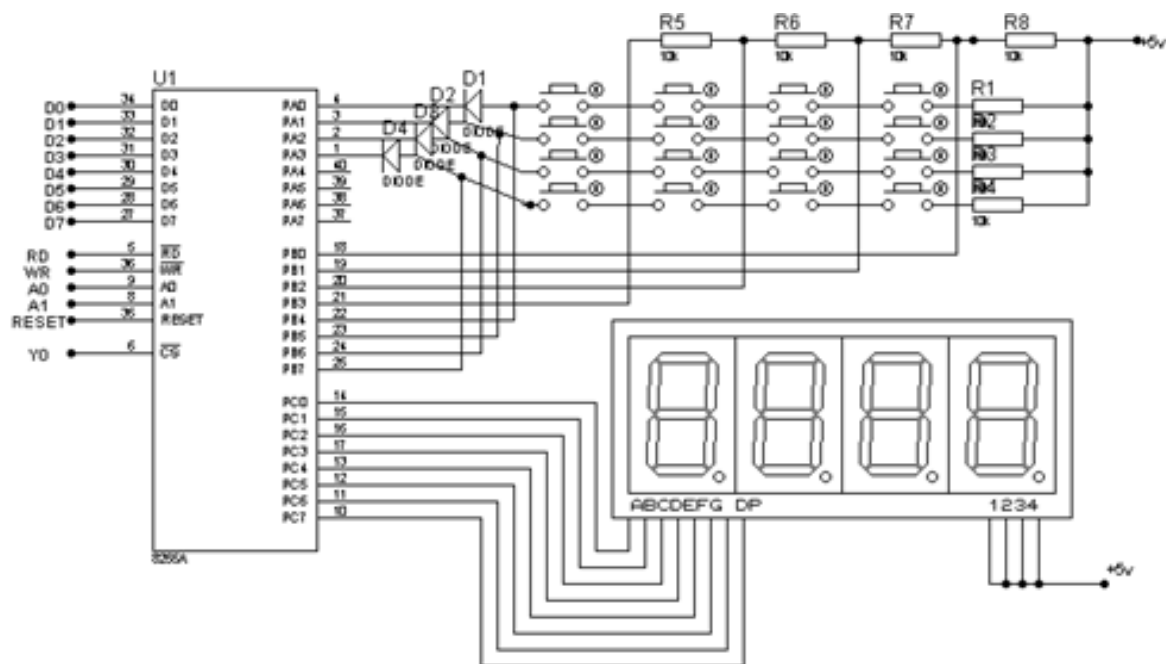
三、实验电路图



8253 定时实验，利用 8253 完成 1 秒的延时



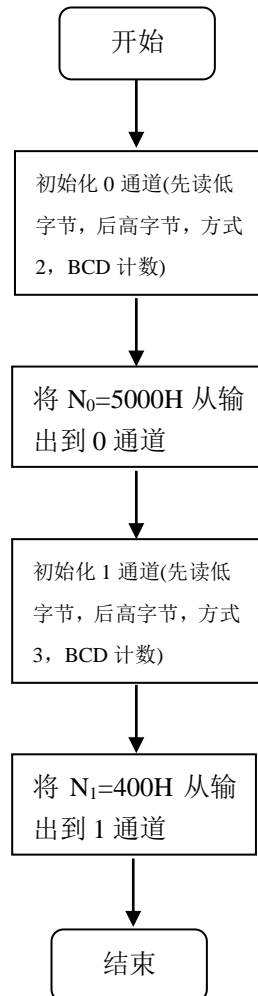
利用 8253 硬件延时控制跑马灯运行



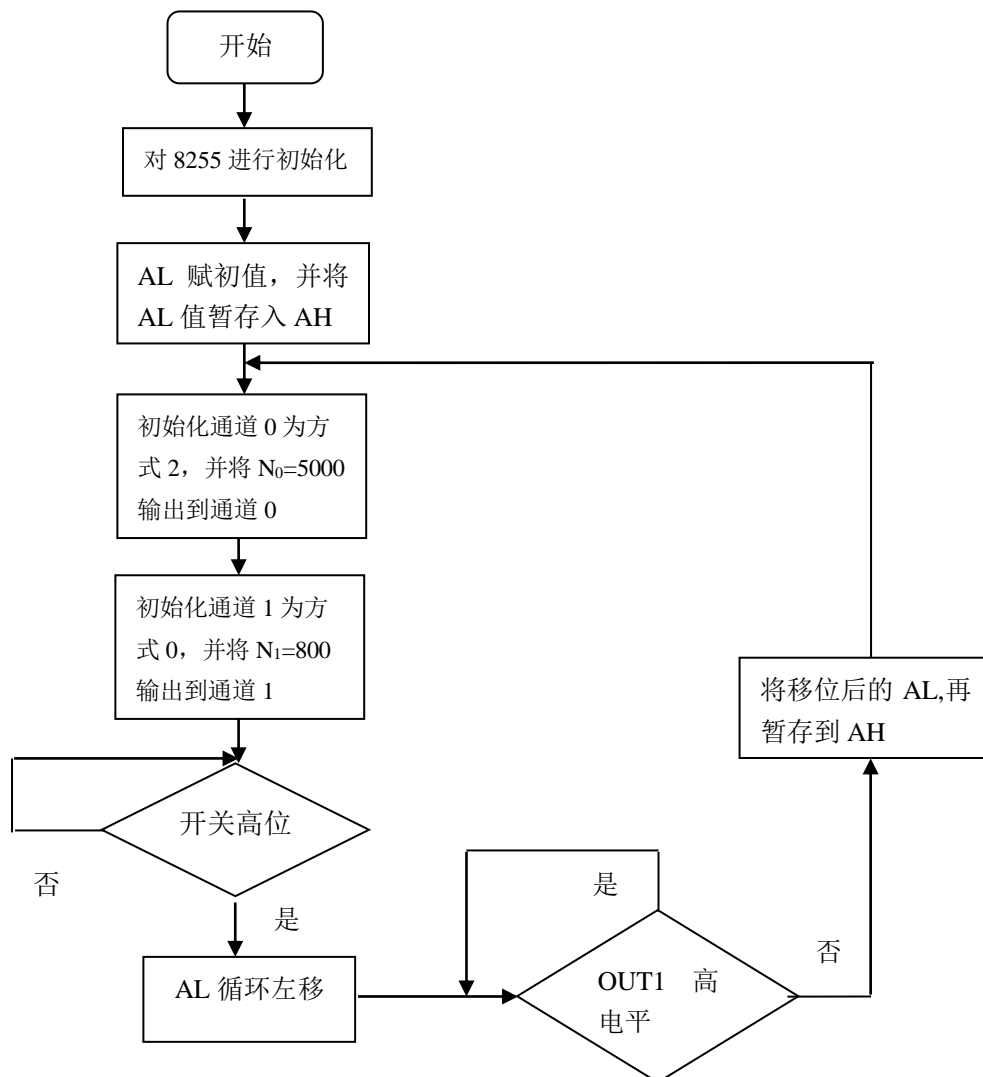
8255 键盘显示实验

四、软件设计程序流程图

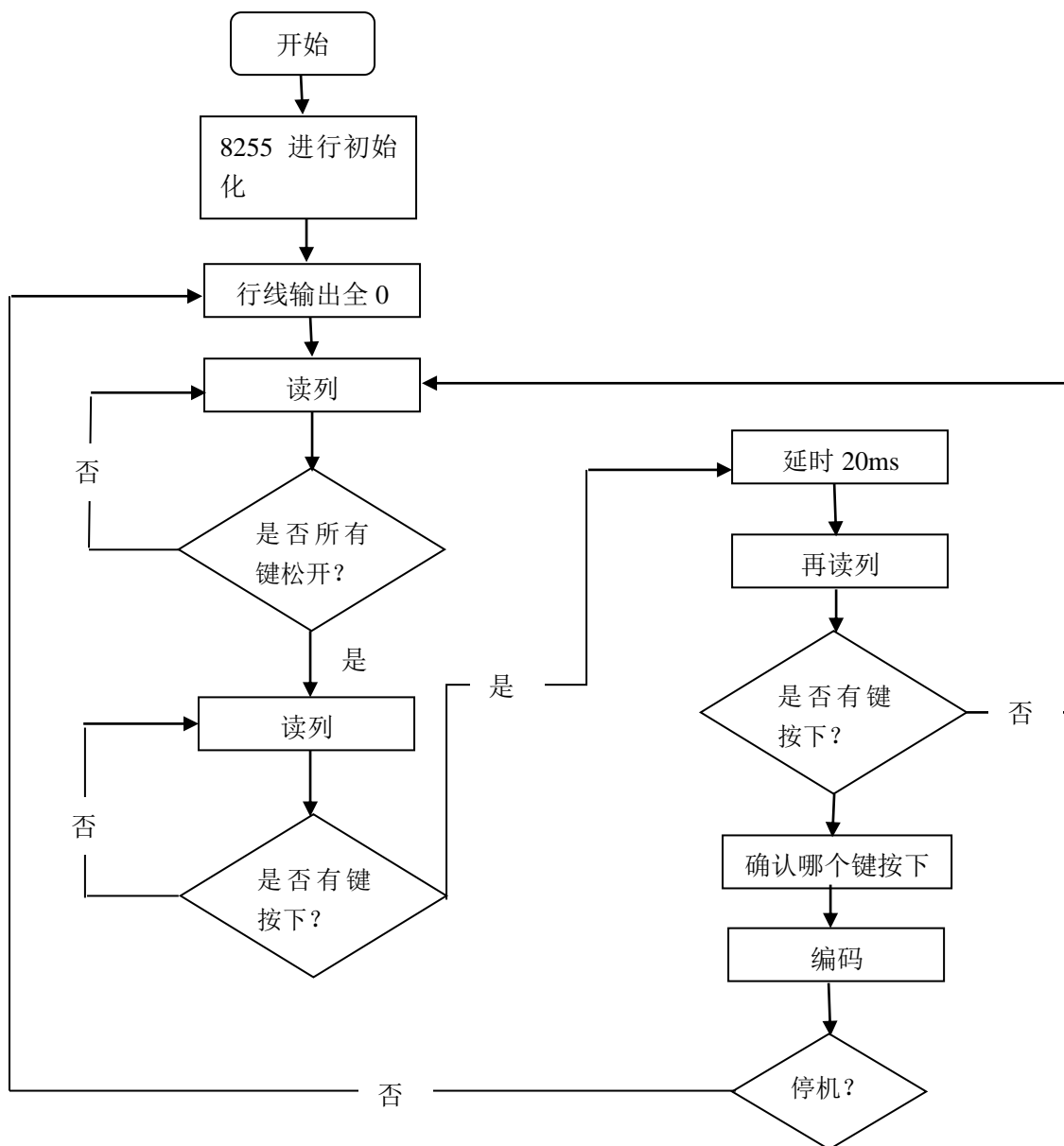
1. 8253 定时实验，利用 8253 完成 1 秒的延时。



2. 利用 8253 硬件延时控制跑马灯运行。



3. 选做 8255 键盘显示实验



五、实验软件代码（加注释）

1. 8253 定时实验，利用 8253 完成 1 秒的延时。

```
1. CLK_0 EQU 298H
2. CLK_1 EQU 299H
3. CLK_2 EQU 29AH
4. CLK_CTL EQU 28BH
5.
6. CODE SEGMENT
7. ASSUME CS:CODE
```

```

8.
9. START:
10. ;0 通道初始化
11. MOV DX,CLK_CTL
12. MOV AL,00110101B ;通道 0 控制字, 先读低字节, 后高字节, 方式 2, BCD 计数
13. OUT DX,AL
14. MOV DX,CLK_0
15. MOV AL,00H
16. OUT DX,AL
17. MOV AL,50H ;取 N0=5000, 则 N1=2MHz/5000=400Hz
18. OUT DX,AL
19.
20. ;1 通道初始化
21. MOV DX,CLK_CTL
22. MOV AL,01110111B ;通道 1 控制字, 先读低字节, 后高字节, 方式 3, BCD 计数
23. OUT DX,AL
24. MOV DX,CLK_1
25. MOV AL,00H
26. OUT DX,AL
27. MOV AL,08H ;周期为 2 秒, 所以 N1=400/(1/2)=800
28. OUT DX,AL
29.
30. CODE ENDS
31. END START

```

2. 利用 8253 硬件延时控制跑马灯运行。

```

1. ;端口地址
2. PORT_A EQU 280H
3. PORT_B EQU 281H
4. PORT_C EQU 282H
5. PORT_CTL EQU 283H
6.
7. CLK_0 EQU 298H
8. CLK_1 EQU 299H
9. CLK_2 EQU 29AH
10. CLK_CTL EQU 29BH
11.
12. CODE SEGMENT
13. ASSUME CS:CODE
14.
15. START:
16. MOV DX,PORT_CTL
17. MOV AL,10001001B

```

```

18. OUT DX,AL      ;AL 输出到 PORT_CR0
19. MOV AL,01H     ;AL 赋初值（第一个灯亮）
20. MOV AH,AL      ;将 AL 的值先暂存到 AH
21.
22. RUN:
23. ;0 通道初始化
24. MOV DX,CLK_CTL
25. MOV AL,00110101B ;通道 0 控制字，先读低字节，后高字节，方式 2，BCD 计数
26. OUT DX,AL
27. MOV DX,CLK_0
28. MOV AL,00H
29. OUT DX,AL
30. MOV AL,50H      ;取 N0=5000,则 N1=2MHz/5000=400Hz
31. OUT DX,AL
32. ;1 通道初始化
33. MOV DX,CLK_CTL
34. MOV AL,01110001B ;通道 1 控制字，先读低字节，后高字节，方式 0，BCD 计数
35. OUT DX,AL
36. MOV DX,CLK_1
37. MOV AL,00H
38. OUT DX,AL
39. MOV AL,08H      ;周期为 2 秒，所以 N1=400/(1/2)=800
40. OUT DX,AL
41.
42. TEST1:
43. ;开关控制部分
44. MOV DX,PORT_C   ;读入 C 端口状态
45. IN AL,DX        ;将 C 端口状态输入到 DX
46. CMP AL,00000010B ;用最高位开关控制跑马灯的开关
47. JZ TEST1        ;若相减不等于 0，即开关为低位，则跳转回 TEST 循环检测
48.
49. ;开关为高位，执行下列代码
50. MOV AL,AH       ;将暂存在 AH 的初值赋给 AL
51. MOV DX,PORT_A   ;对 A 口输出状态
52. OUT DX,AL       ;将 AL 的值输出到端口 A，第一次即第一个灯亮
53. ROL AL,1        ;AL 中的值循环左移（1 位）
54. MOV BL,AL
55.
56. TEST2:
57. MOV DX,PORT_C   ;读入 8254 的 OUT1 状态（与 PC0 相连）
58. IN AL,DX
59. CMP AL,00000001B ;判断 OUT1 是否为高电平，是则继续循环，不是继续运行
60. JNZ TEST2
61.

```

```

62. MOV AL,BL
63. MOV AH,AL          ;将移位后的 AL,再暂存到 AH
64.
65. JMP RUN            ;继续循环(若开关保持高位, 循环到 MOV AL,AH 语句)
66.
67. CODE ENDS
68. END START

```

3. 8255 键盘显示实验

```

1. ;端口地址
2. PORT_A EQU 280H
3. PORT_B EQU 281H
4. PORT_C EQU 282H
5. PORT_CTL EQU 283H
6. ;键盘扫描码表
7. DATA SEGMENT
8. TABLE_A DB 77H,7BH,7DH,7EH,0B7H,0BBH,0BDH,0BEH
9. ;          0   1   2   3   4   5   6   7
10.          DB 0D7H,0DBH,0DDH,0DEH,0E7H,0EBH,0EDH,0EEH
11.;          8   9  10  11  12  13  14  15
12.;七段码表
13. TABLE_C DB 3FH,06H,5BH,4FH,66H,6DH,7DH,0FH
14.          DB 7FH,67H,77H,7CH,39H,5EH,79H,71H
15. DATA ENDS
16.;堆栈段
17. STACK SEGMENT STACK
18. DW 50 DUP(0)
19. TOP_STACK LABEL WORD
20. STACK ENDS
21.;代码段
22. CODE SEGMENT
23. ASSUME CS:CODE,DS:DATA,SS:STACK
24. START:
25. MOV AX,STACK
26. MOV SS,AX          ;给堆栈段赋地址
27. LEA SP,TOP_STACK   ;sp 指向堆栈栈顶
28. MOV AX,DATA
29. MOV DS,AX          ;给数据段赋地址
30.
31.;初始化, 方式 0, A 口输出, B,C 口输入
32. MOV DX,PORT_CTL    ;指向控制口
33. MOV AL,10001011B    ;方式 0, A 口输出, C 口高位输入, B 口方式 0, B 口输入, C 口低
                        四位输入

```

```

34.OUT DX,AL      ;写入控制字
35.
36.;向所有行送 0
37.MOV DX,PORT_A  ;指向 A 口
38.MOV AL,00H
39.OUT DX,AL      ;向 A 口各位输出 0
40.
41.;读列，查看是否所有键均松开
42.MOV DX,PORT_B  ;指向 B 口
43.WAIT_OPEN:
44.IN AL,DX       ;键盘状态读入 B 口
45.AND AL,0FH     ;保留低四位（只查看低 4 位，列值）
46.CMP AL,0FH     ;低四位是否均为 1？（各键均松开）
47.JNE WAIT_OPEN  ;低四位不均为 1，则继续循环这个函数，直到低四位均为 1
48.
49.;各键均已松开，再查看列是否有 0，即是否有键按下
50.WAIT_PRE:
51.IN AL,DX
52.AND AL,0FH
53.CMP AL,0FH
54.JE WAIT_PRE    ;低四位均为 1 时（没有键按下），循环这个函数，直到有键按下
55.
56.;有键按下，延时 20ms，防止抖动
57.MOV CX,16EAH
58.DELAY:
59.LOOP DELAY     ;循环 16EAH 次
60.
61.;再查列，看键是否被压着
62.IN AL,DX
63.AND AL,0FH
64.CMP AL,0FH
65.JE WAIT_PRE    ;如果没有键按下，则说明之前是抖动，就跳回到等待到按键函数
66.
67.;若键依旧被压着，则确定哪一个键被压下
68.MOV AL,0FEH    ;AL=1111 1110 B
69.MOV CL,AL      ;CL=1111 1110 B
70.NEXT_ROW:
71.MOV DX,PORT_A
72.OUT DX,AL      ;先让第一行全部置 0（低电平）
73.MOV DX,PORT_B
74.IN AL,DX       ;读 B 口状态
75.AND AL,0FH     ;取低 4 位（列值）
76.CMP AL,0FH     ;查看低 4 位（列值）看是否有键按下（是否均为 1）
77.JNE ENCODE     ;否，说明有键按下，则转去编码函数

```

```

78. ROL CL,01      ;左移一次，让下一行都变为 0
79. MOV AL,CL
80. JMP NEXT_ROW   ;再次循环函数，查看第二行
81.
82. ;找到一列为低电平，对压键的行列值编码
83. ENCODE:
84. MOV BX,000FH    ;建立地址指针，先指向 F 键（15）对应的地址
85. IN AL,DX        ;读 B 口状态（行列号）
86. NEXT_TRY:
87. CMP AL,TABLE_A[BX] ;读入的行列值与表中查的相等吗？
88. JE DONE         ;相等，转出到完成函数
89. DEC BX          ;不相等，则转到下一个地址（往小的慢慢转）
90. JNS NEXT_TRY    ;BX 还大于 0 时，继续跳回到 NEXT_TRY 函数
91. MOV AH,01       ;一直查到地址为负数都没查到，置出错码 01→AH
92. JMP EXIT        ;直接跳到退出程序
93. DONE:
94. MOV AL,BL        ;BL 中存有键的 16 进制代码（按下的键是几）
95. MOV BX,OFFSET TABLE_C
96. XLAT
97. MOV DX,PORT_C
98. OUT DX,AL
99. MOV CX,0FFH
100. JMP DELAY2
101. DELAY2: LOOP DELAY2
102. JMP START
103. EXIT:
104. HLT
105. CODE ENDS
106. END

```

六、实验结果及体会

通过第二次实验，我了解了 8253 芯片的基本工作原理和工作方式，并通过硬延时跑马灯的编程，熟悉汇编代码的编写，以及如何用汇编语言级 8253 形成硬延时控制。在实验过程中也遇到了一系列的问题，通过与同学的讨论请假以及对理论知识的补充，最后一步步独立完成实验，感觉自己对微机原理这门课的掌握又有了很大的提升。

中国矿业大学计算机学院

2017 级本科生实验报告

课程名称 微机原理与接口实验

实验题目 DAC0832+ADC0809 实验

学生姓名 袁孝健

学 号 06172151

专业班级 信息安全 2017-01 班

任课教师 王莉

实验三

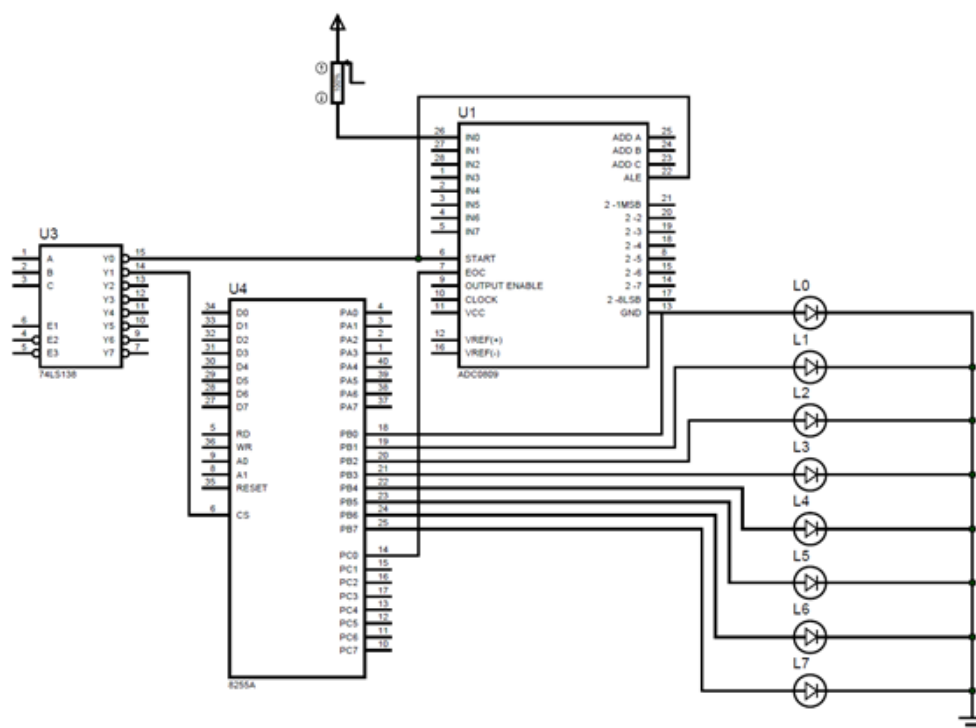
一、实验题目：

1. 利用 0809 完成 1 个通道的模拟量采集，并利用 8255 输出二进制码。
2. 利用 0832 产生锯齿波输出，并在直流电机上观察结果。

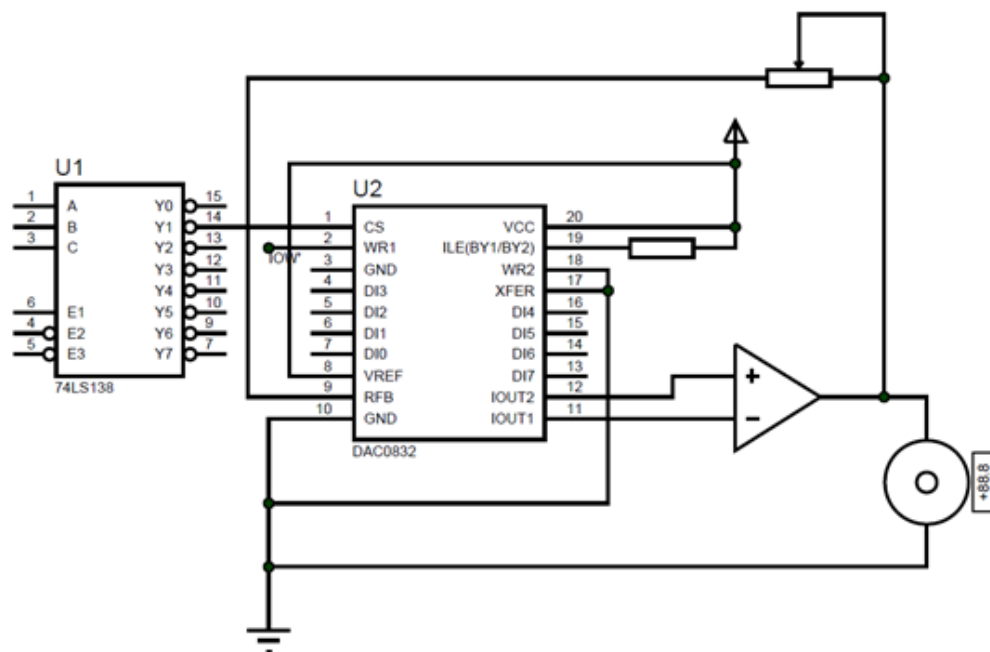
二、实验目的：熟悉汇编语言编程环境，掌握 0809 及 0832 芯片工作原理、电路设计及利用汇编语言编写接口软件。

三、实验电路图

1. 利用 0809 完成 1 个通道的模拟量采集，并利用 8255 输出二进制码。

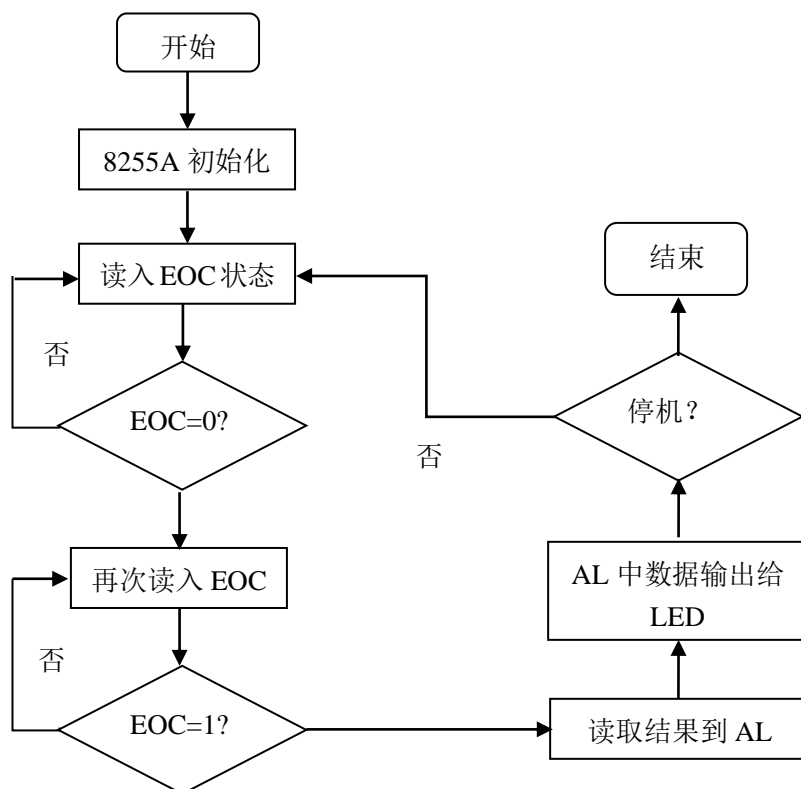


2. 利用 0832 产生锯齿波输出，并在直流电机上观察结果。

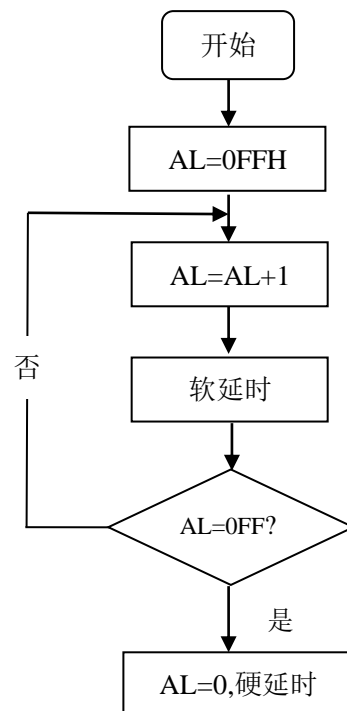


四、软件设计程序流程图

1. 利用 0809 完成 1 个通道的模拟量采集，并利用 8255 输出二进制码。



2. 利用 0832 产生锯齿波输出，并在直流电机上观察结果。



五、实验软件代码（加注释）

1. 利用 0809 完成 1 个通道的模拟量采集，并利用 8255 输出二进制码。

```
1. PORT_A EQU 290H
2. PORT_B EQU 291H
3. PORT_C EQU 292H
4. PORT_CTL EQU 293H
5. ADC0809 EQU 280H
6.
7. CODE SEGMENT
8. ASSUME CS:CODE
9. START:
10. ;初始化 8255
11. MOV DX,PORT_CTL
12. MOV AL,10011001B ;A 口输入, B 口输出
13. OUT DX,AL
14.
15. ;读 EOC 判断是否开始转换
16. AGAIN:
17. ;MOV DX,PORT_A
18. ;IN AL,DX
19. ;TEST AL,10000000B ;判断 EOC 是否为低电平
```

```

20.;JNZ AGAIN    ;若不是低电平，循环等待
21.
22.;输出任意一个 AL,启动转换
23.MOV DX,ADC0809
24.OUT DX,AL
25.
26.;读 EOC 判读转换是否结束
27.IF_END:
28.MOV DX,PORT_A
29.IN AL,DX
30.TEST AL,10000000B ;判断 EOC 是否为高电平
31.JZ IF_END    ;若不是高电平，循环等待
32.;输出数据
33.MOV DX,ADC0809
34.IN AL,DX     ;将转换后数据读入 AL
35.MOV DX,PORT_B
36.OUT DX,AL    ;将数据输出到 8255 的 B 口
37.JMP AGAIN   ;循环采集
38.
39.CODE ENDS
40.END START

```

2. 利用 0832 产生锯齿波输出，并在直流电机上观察结果。

```

1. CODE SEGMENT
2. ASSUME CS:CODE
3. START:
4. ;8255 初始化
5. MOV DX,28BH
6. MOV AL, 10000010B
7. OUT DX,AL
8.
9. ;8253 初始化
10.MOV DX,283H
11.MOV AL,00110101B ;通道 0，方式 2，BCD 码
12.OUT DX,AL
13.MOV DX,280H
14.MOV AL,00H
15.OUT DX,AL
16.MOV AL,50H      ;N0=5000
17.OUT DX,AL
18.
19.MOV DX,283H
20.MOV AL,01110111B ;通道 1，方式 3，BCD 码

```

```

21. OUT DX,AL
22. MOV DX,281H
23. MOV AL,00H
24. OUT DX,AL
25. MOV AL,08H          ;N1=800
26. OUT DX,AL
27. ;8253 硬延时
28. WAIT_H:
29. MOV AL,00H
30. OUT DX,AL
31. MOV DX,289H
32. IN AL,DX
33. TEST AL,1
34. JNZ WAIT_H          ;不等于 0 循环，否则继续执行
35. WAIT_L:
36. MOV DX,289H
37. IN AL,DX
38. TEST AL,1
39. JZ WAIT_L           ;等于 0 循环，否则继续执行
40.
41. UP:
42. MOV DX,290H
43. OUT DX,AL
44. INC AL              ;AL 自增
45. MOV CX,65535
46. DELAY: LOOP DELAY   ;软延时，控制斜率
47. CMP AL,0FFH         ;判断是否到达最大值
48. JNZ UP
49. JMP WAIT_H
50. CODE ENDS
51. END START

```

六、实验结果及体会

通过本次实验，我对 A/D 转换、D/A 转换有了更深入的理解，同时对 0809 芯片以及 0832 芯片有了比较全面的认识，并且通过两个具体的实验掌握了它们的编程方法，学会了将 0809、0832 与前两次实验学习的芯片结合起来使用，对整个课程有了更加连贯的掌握。通过三次实验已初步掌握了微机常见接口以及它们的汇编编程方法。