

实验一 环境熟悉与I/O地址译码

一、实验目的

掌握I/O地址译码电路的工作原理。

二、实验内容

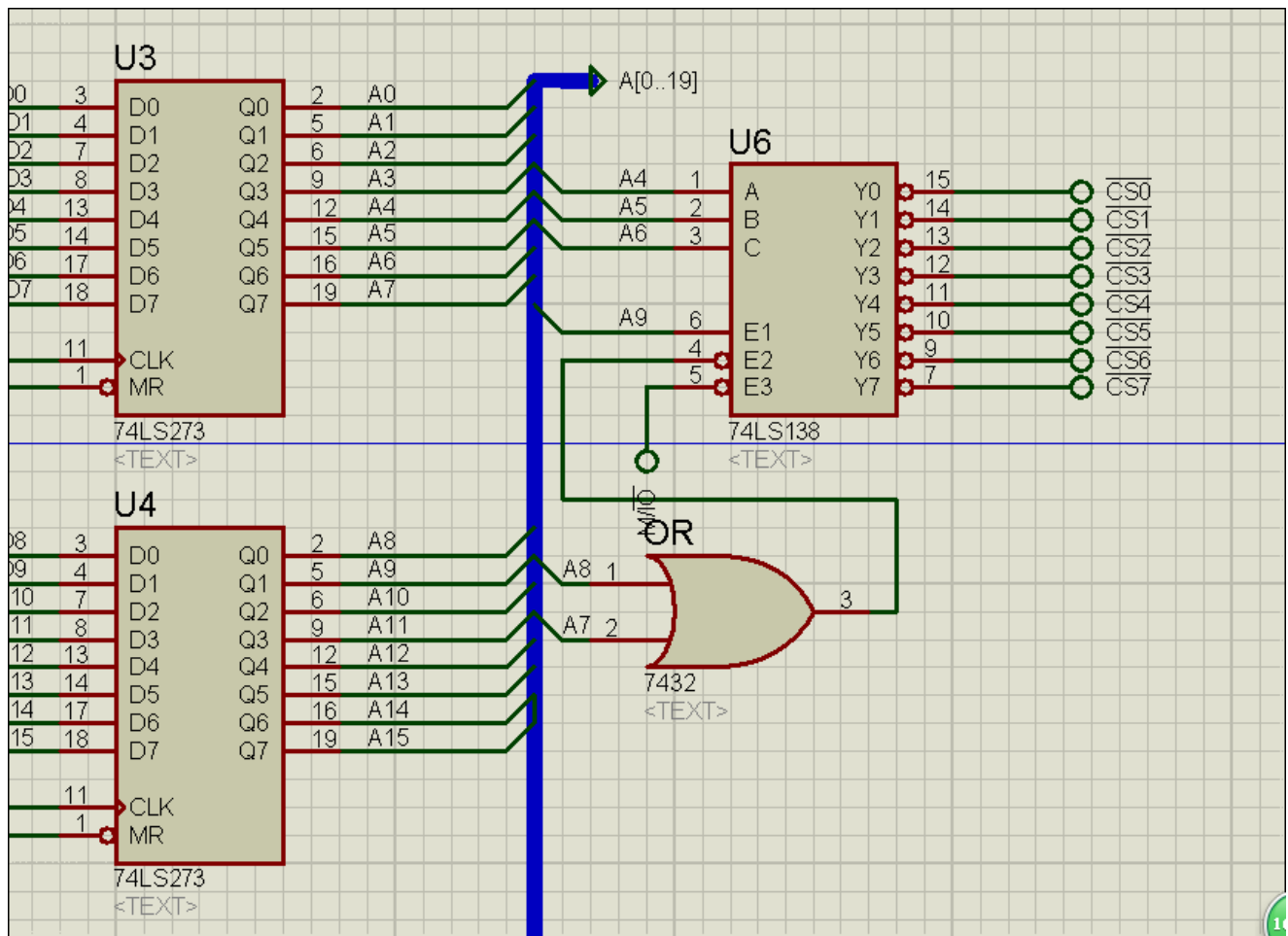
1. 连接AD4~AD9等几根线，使得138译码器Y0#~Y7#译码出的地址依次分别是Y0#：200H~20FH; Y1#：210H~21FH；.....；Y7#：270h~27FH
2. 利用Source->Add/Remove Source files中添加源程序EX-138.ASM（在实验包中已提供），该程序向240H端口写任意数据。如果是第一次运行，Source Code Filename这一项选New来调入源程序。
3. 在Source菜单下点击“Build All”，汇编源代码（全部编译）；执行程序，查看Y0~Y7中是否Y4#脚输出为有效（管脚出的有颜色的小方块为红色是高电平，蓝色为低电平）。
4. 改变程序中的输出地址，使得Y0#, Y1#,Y6#,Y7#分别有效。
5. 改变A9~A3的接线方法，从而得到Y0；388H~38FH；Y1：398H~39FH;；Y7：3F8H~3FFH。并修改上一问的程序，以同样使得Y4#有效。

1) 源程序

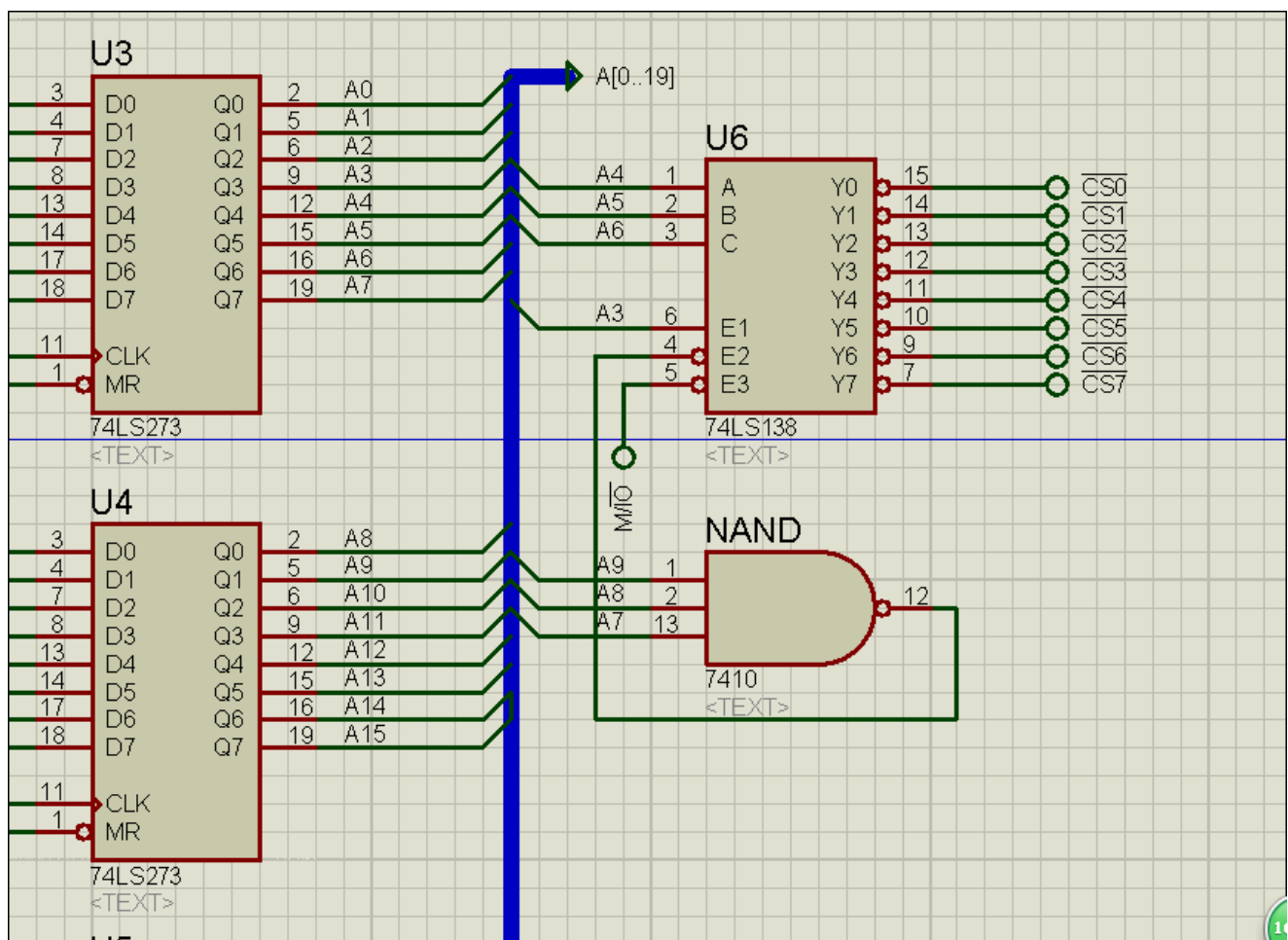
```
1  .8086
2  .MODEL SMALL
3  .STACK
4  .DATA
5      ADDRESS  WORD  3C8H; 此值会变
6  .CODE
7  START:
8      MOV AX,@DATA
9      MOV DS,AX
10     MOV DX,ADDRESS
11     MOV AL,0
12     OUT DX,AL
13     JMP $
14 END START
```

2) 电路原理图（138译码部分）

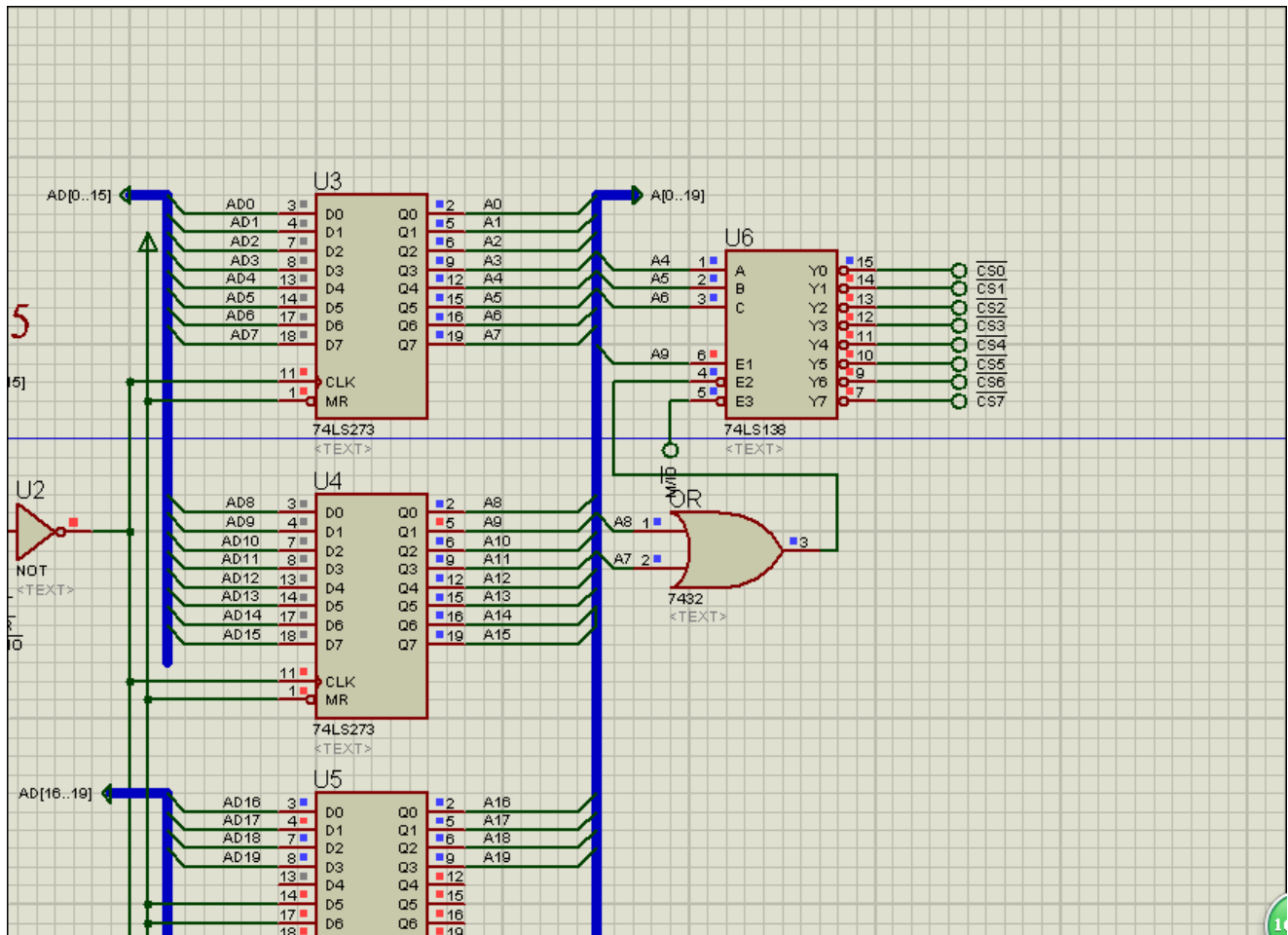
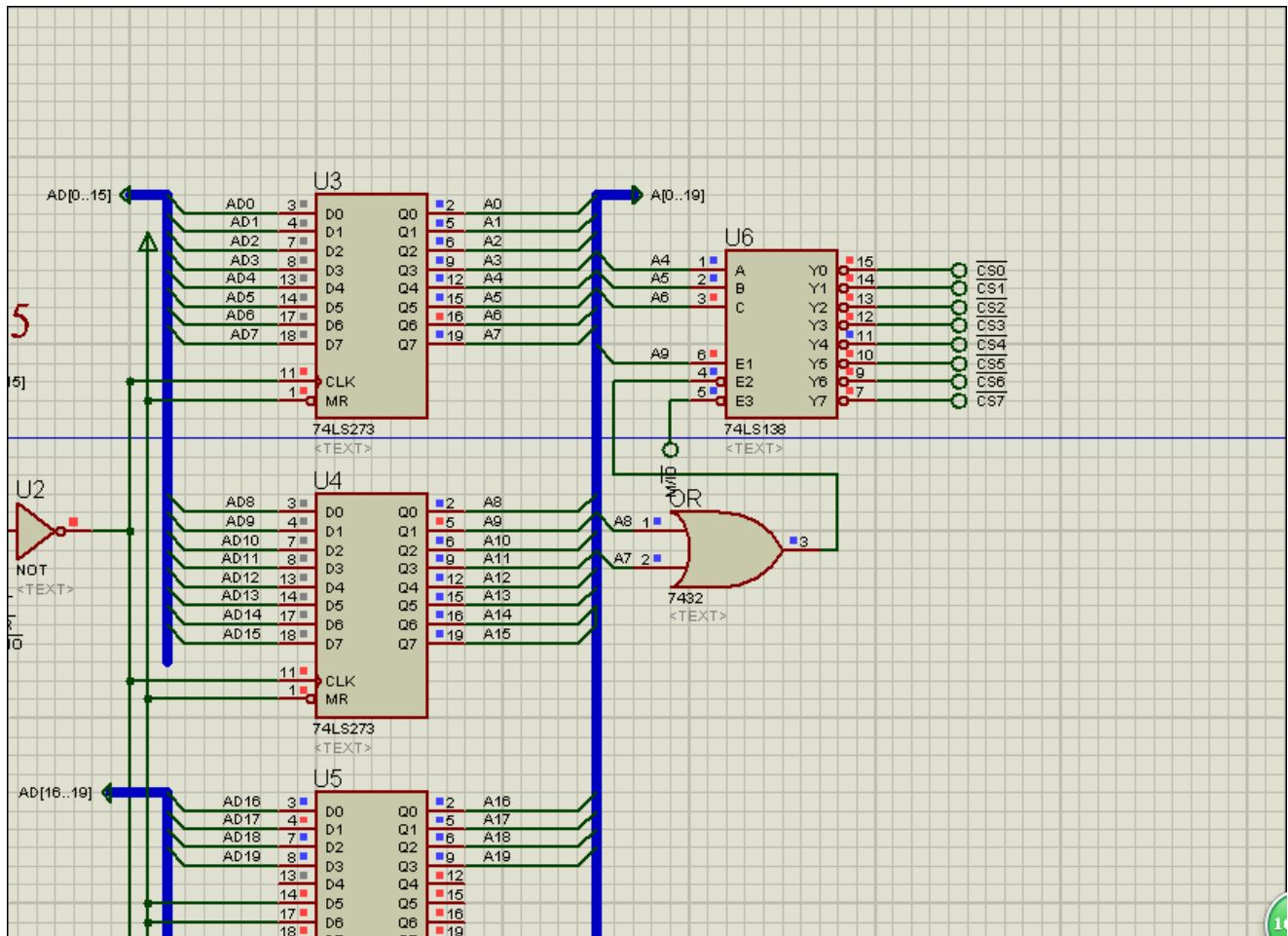
1.

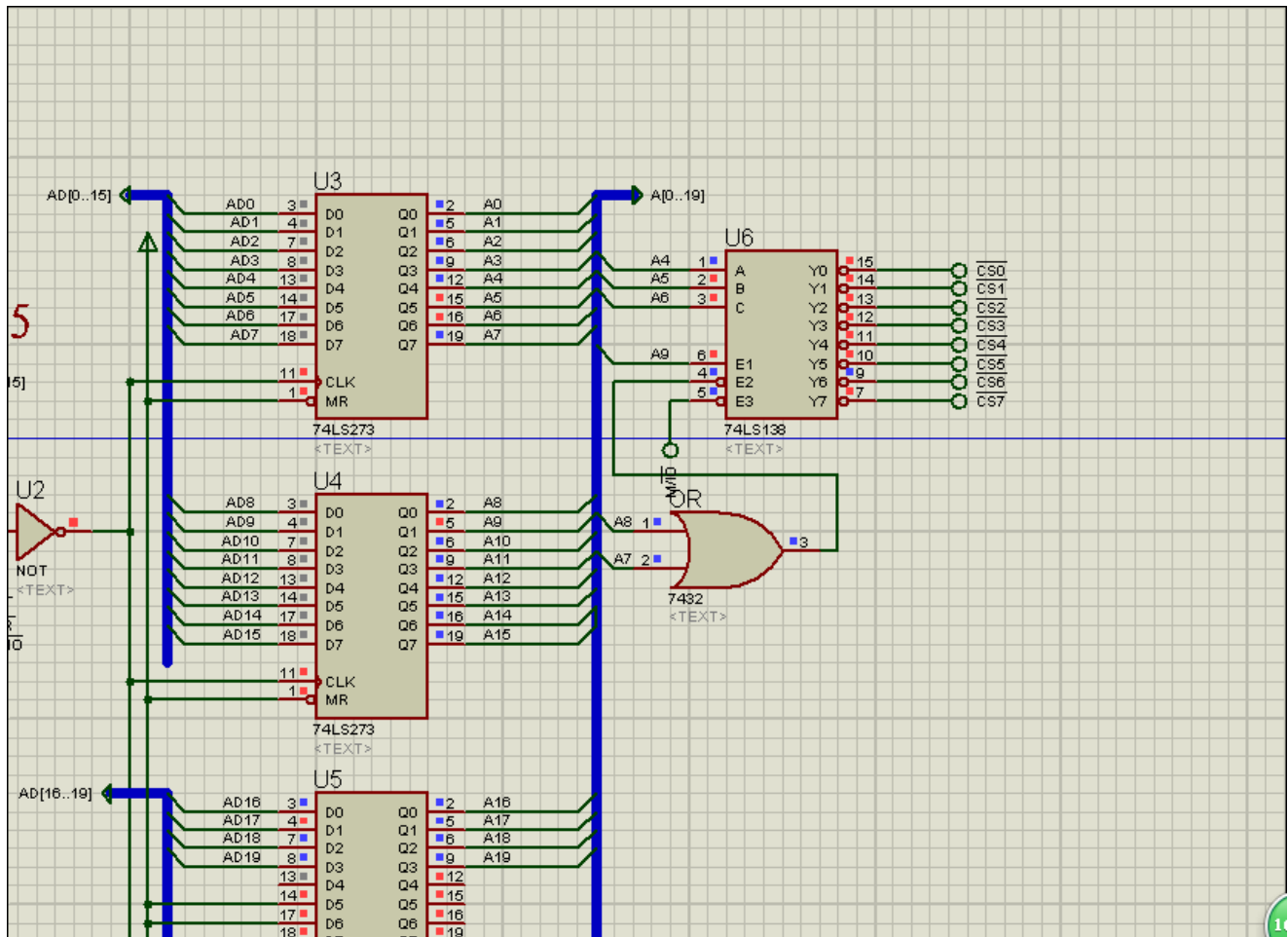
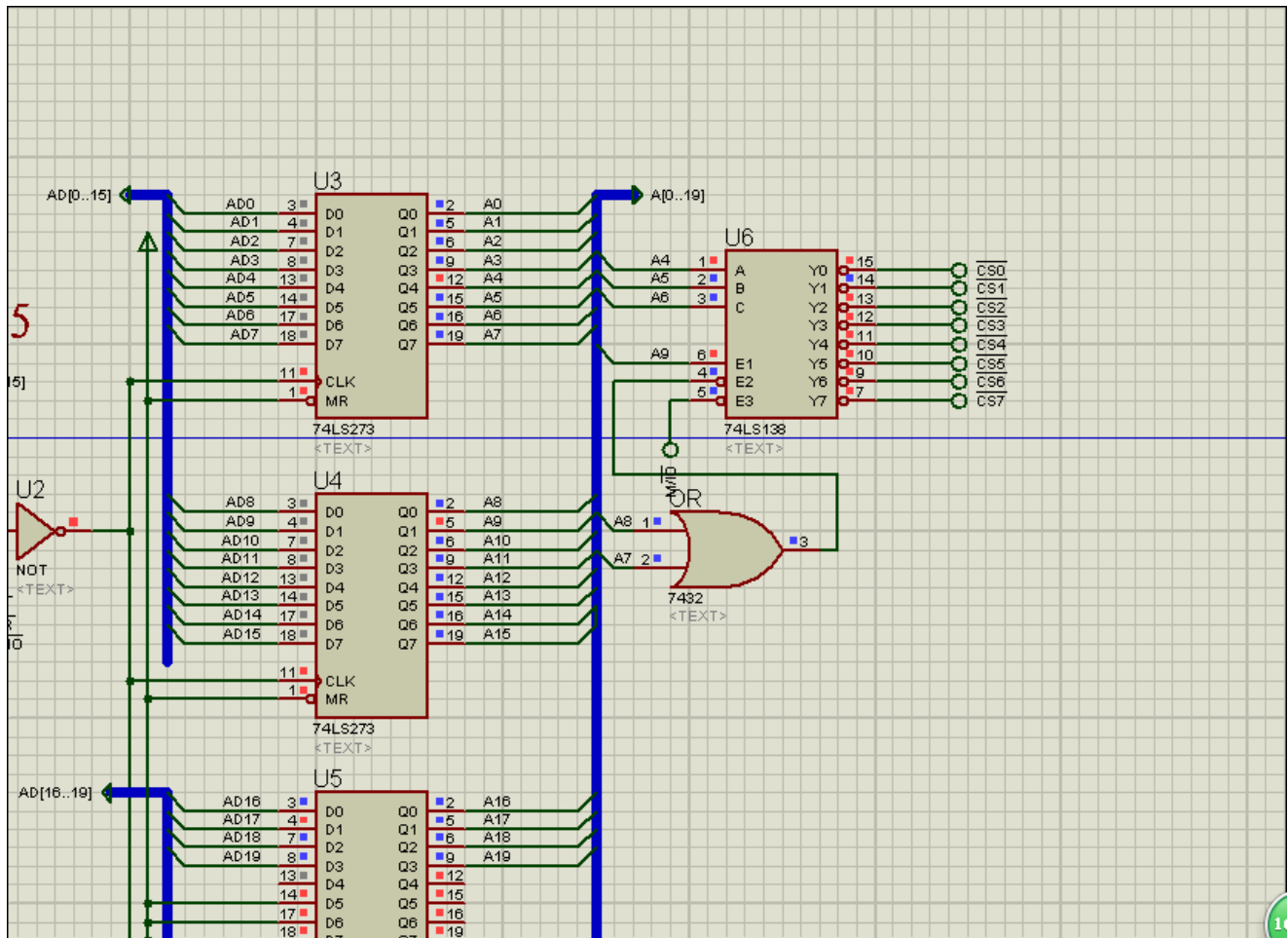


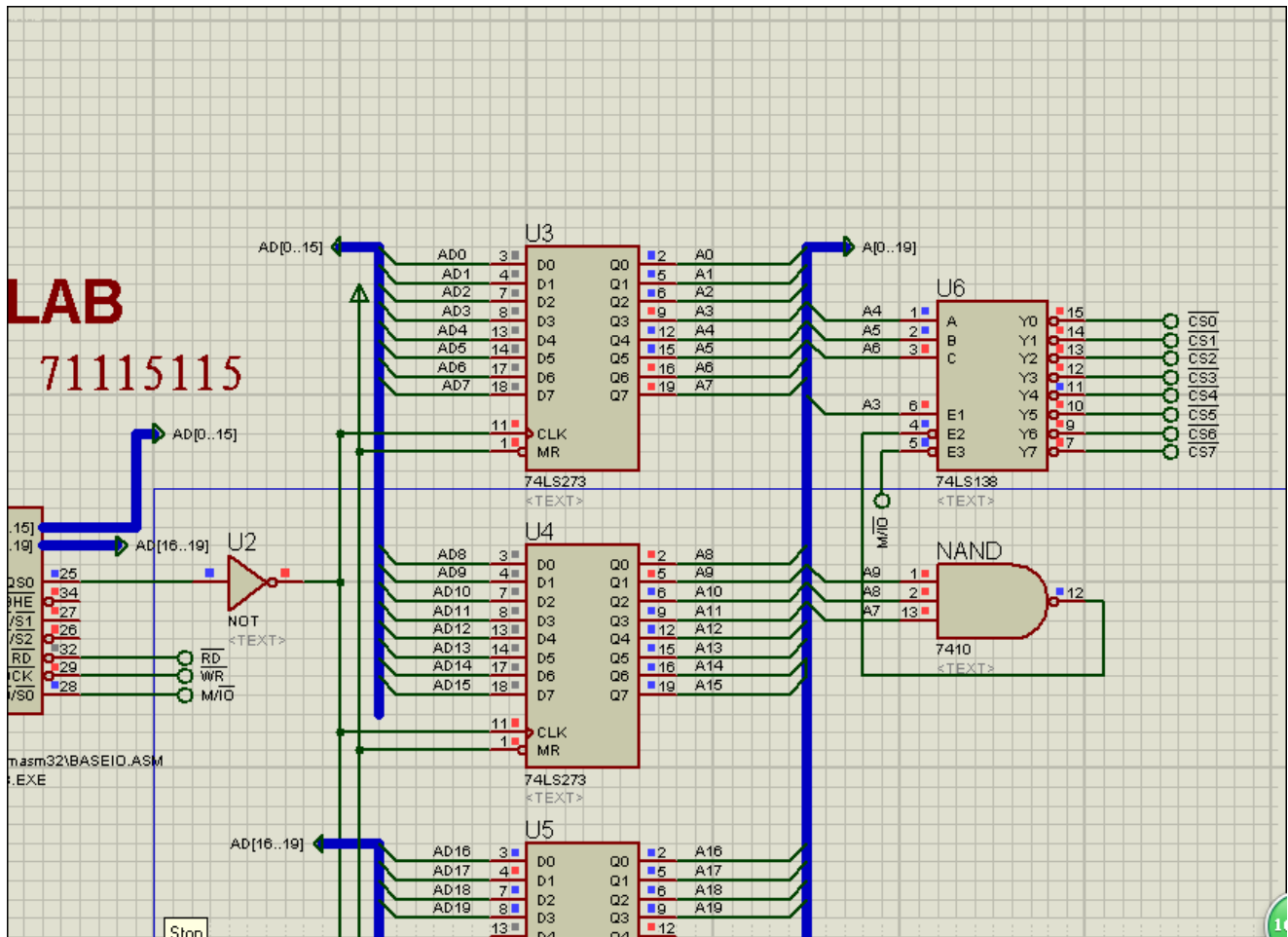
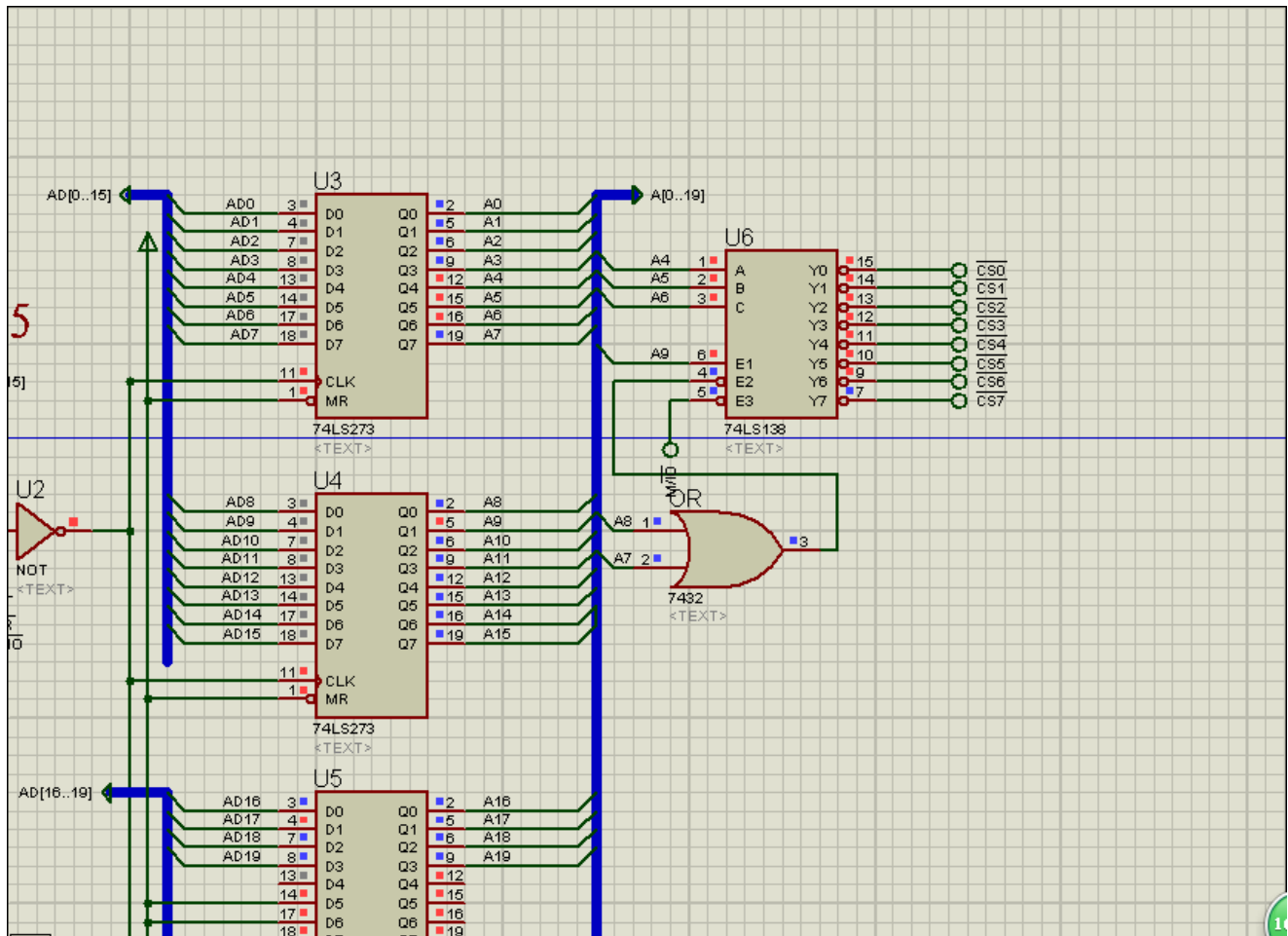
2.



3) 运行结果贴图（138译码及上面两个273的输出）







实验二 可编程中断控制器8259

一、实验目的

1. 掌握8259的基本工作原理和编程方法。
2. 深入了解中断的概念和实地址模式下中断处理程序的编写方法。

二、实验内容

将接口实验包中所带的EX-8259.DSN文件用ISIS 7打开。按手册接线并执行。

```

1  .MODEL SMALL
2  .8086
3  .stack
4  .data
5  come byte 00000000B ; 计算中断到来次数的变量
6  .code
7  start:
8      mov ax,@data
9      mov ds,ax
10     cli
11     mov ax,0 ; init interrupt vector TABLE
12     mov es,ax
13     mov si,0 ; 由于模拟器的错误,所有中断向量需要指向同一个地址
14     mov cx,255
15 l:   mov ax,offset int0
16     mov es:[si],ax
17     mov ax,seg int0
18     mov es:[si+2],ax
19     add si,4
20     loop l
21     mov ax,offset int0
22     mov es:[si],ax
23     mov ax,seg int0
24     mov es:[si+2],ax
25     mov ax,@data
26     mov ds,ax
27     mov al,00010011b ; init 8259
28     mov dx,210h
29     out dx,al ; ICW1
30     mov al,60h
31     mov dx,212h
32     out dx,al ; ICW2
33     mov ax,00000001h
34     out dx,al ; ICW4
35     mov ax,0
36     out dx,al ; OCW1
37     sti
38     jmp $
39 int0 proc
40     push ax
41     push dx
42     push ds
43     mov ax,@data
44     mov ds,ax
45     mov ax,0
46     mov al,come ; 中断到来,计数器+1
47     inc al
48     mov come,al
49     mov al,20h ; 发EOI命令
50     mov dx,210h
51     out dx,al
52     pop ds
53     pop dx

```



```

54     pop ax
55     iret
56 int0 endp
57 END start

```

三、运行结果贴图（执行三次中断，每次中断后的8086寄存器的截图）

The screenshot displays the ISIS Professional (Animating) software interface. The main window shows the 8086 Source Code for a program named 'EX-8259.ASM'. The code defines an interrupt procedure 'into' that pushes the AX, DX, and DS registers, increments a counter 'come', and sends an EOI command. The registers window shows the state of the 8086 registers after the first interrupt.

8086 Source Code - U1

```

8037 mov dx, 210h
803A out dx, al          ; ICW1
803B mov al, 60h
803D out dx, al          ; ICW2
8040 out dx, al          ; ICW4
8041 mov ax, 00000001h
8044 out dx, al          ; OCW1
8045 mov ax, 0
8048 out dx, al          ; OCW1
8049 sti
804A jmp $
804L ---- into proc
804C push ax
804D push dx
804E push ds
804F mov ax, @data
8052 mov ds, ax
8054 mov ax, 0
8057 mov al, come        ; 中断到来, 计数器+1
805A inc al
805C mov come, al
805F mov al, 20h         ; 发EOI命令
8061 mov dx, 210h
8064 out dx, al
8065 pop ds
8066 pop dx
8067 pop ax
8068 iret
806L ---- into endp

```

8086 Registers - U1

PC: 0000	AX: 0001	SI: 03FC	LA: 0845C
OP: A2 0A 00	DX: 0212	DI: 0000	LA: 00000
PR: B0 20 BA	SP: 03F4	BP: 0000	LA: 08464
			LA: 08070

The circuit diagram shows the 8259A interrupt controller (U7) connected to the 8086. The 8259A is configured with CS1, WR, RD, A1, INTR, and INTA. The 8086 is connected to the 8259A via AD[0..7], D[0..7], and CAS[0..2]. A 10k resistor (R1) is connected to the 8259A's output.

8086 Source Code - U1

```

EX-8259.ASM
8037 mov dx,210h
8038 out dx,al ; ICW1
8039 mov al,60h
8040 out dx,al ; ICW2
8041 mov ax,00000001h
8044 out dx,al ; ICW4
8045 mov ax,0
8048 out dx,al ; OCW1
8049 sti
804A jmp $
804C ---- into proc
804C push ax
804D push dx
804E push ds
804F mov ax,@data
8052 mov ds,ax
8054 mov ax,0
8057 mov al,come ; 中断到来, 计数器+1
805A inc al
805C mov come, al
805F mov al,20h ; 发EOI命令
8061 mov dx,210h
8064 out dx,al
8065 pop ds
8066 pop dx
8067 pop ax
8068 iret
806A ---- into endp

```

8086 Registers - U1

PC: mov [+000A],al	Op: A2 0A 00	Pr: B0 20 BA
CS: 0800	IP: 005C	LA: 0805C
AX: 0002	BX: 0000	
CX: 0000	DX: 0212	
DS: 0806	SI: 03FC	LA: 0845C
ES: 0000	DI: 0000	LA: 00000
SS: 0807	SP: 03F4	LA: 08464
BP: 0000	LA: 08070	
FL:		

U7 8259A

AD[0..7] ← D[0..7]

CS1 CS 18

WR WR 19

RD RD 20

A1 A0 21

16 SP/EN 22

INTR INT 23

INTA INTA 24

CAS[0..2] 25

R1 10k <TEXT>

8259A <TEXT>

23 Message(s) [U1] Digital breakpoint at time 2.6023s (3.0000us elapsed) - breakpoint is reached +3700.0 +3000.0 th

8086 Source Code - U1

```

EX-8259.ASM
8029 mov ax,seg into
802C mov es:[si+2],ax
8030 mov ax,@data
8033 mov ds,ax
8035 mov al,00010011b ; init 8259
8037 mov dx,210h
8038 out dx,al ; ICW1
8039 mov al,60h
8040 out dx,al ; ICW2
8041 mov ax,00000001h
8044 out dx,al ; ICW4
8045 mov ax,0
8048 out dx,al ; OCW1
8049 sti
804A jmp $
804C ---- into proc
804C push ax
804D push dx
804E push ds
804F mov ax,@data
8052 mov ds,ax
8054 mov ax,0
8057 mov al,come ; 中断到来, 计数器+1
805A inc al
805C mov come, al
805F mov al,20h ; 发EOI命令
8061 mov dx,210h
8064 out dx,al

```

8086 Registers - U1

PC: mov [+000A],al	Op: A2 0A 00	Pr: B0 20 BA
CS: 0800	IP: 005C	LA: 0805C
AX: 0003	BX: 0000	
CX: 0000	DX: 0212	
DS: 0806	SI: 03FC	LA: 0845C
ES: 0000	DI: 0000	LA: 00000
SS: 0807	SP: 03F4	LA: 08464
BP: 0000	LA: 08070	
FL: PF		

U7 8259A

AD[0..7] ← D[0..7]

CS1 CS 18

WR WR 19

RD RD 20

A1 A0 21

16 SP/EN 22

INTR INT 23

INTA INTA 24

CAS[0..2] 25

R1 10k <TEXT>

8259A <TEXT>

30 Message(s) [U1] Digital breakpoint at time 3.6533s (3.0000us elapsed) - breakpoint is reached +3700.0 +3000.0 th

实验三 可编程定时器计数器8253

一、实验目的

掌握8253的基本工作原理、编程方法及其应用。

二、实验内容

一) 研究定时计数器 (选)

1) 源程序

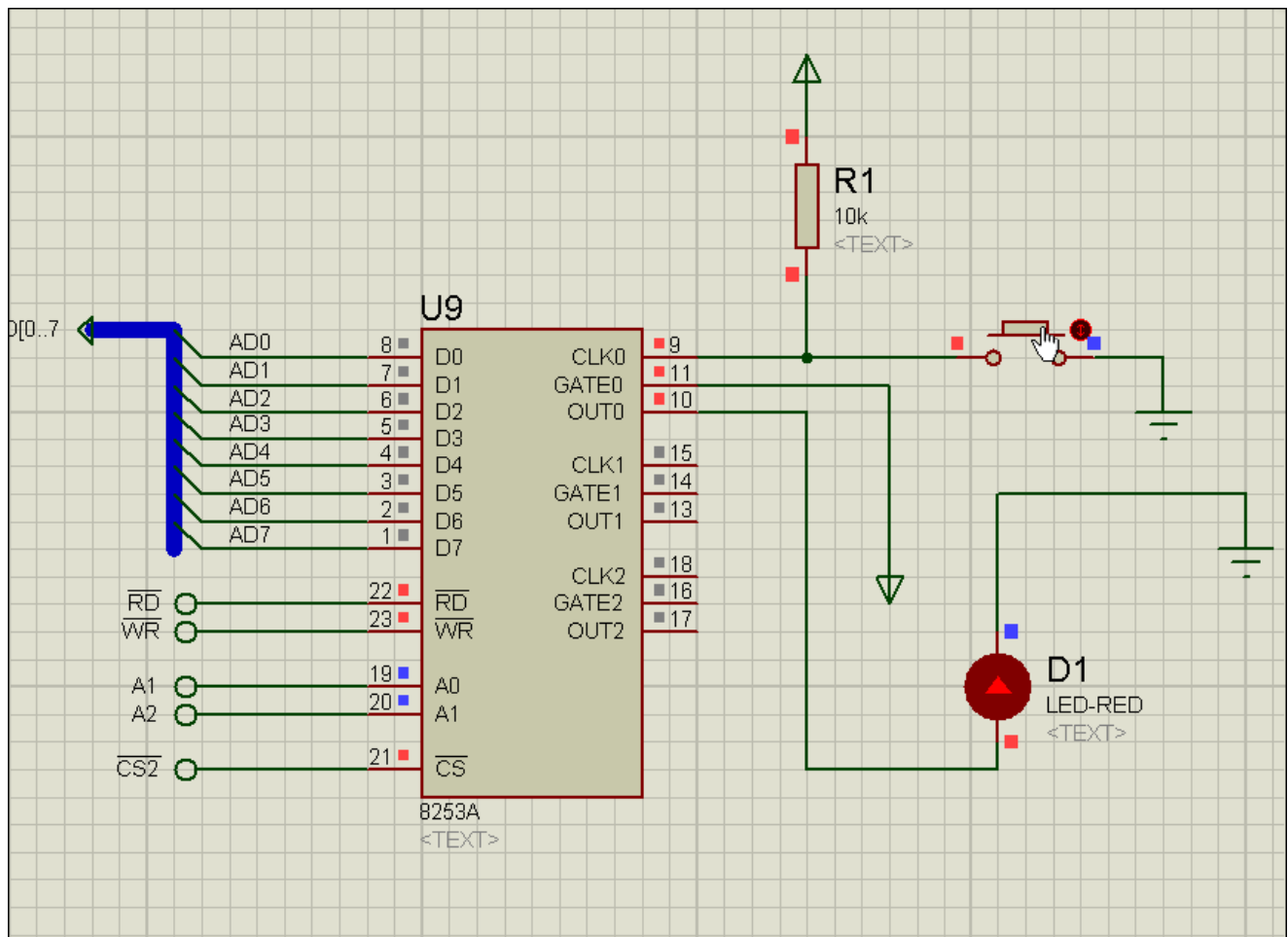
```
1  .8086
2  .MODEL SMALL
3  .STACK
4  .DATA
5  .CODE
6  START:
7      MOV AL,00010001B ; 使用计数器0, 只写低八位, 模式0, BCD码
8      MOV DX,226H
9      OUT DX,AL
10     MOV AL,10H
11     MOV DX,220H
12     OUT DX,AL
13     END START
```

2) 讨论题

如果把方式0改成方式1，电路不动，则按下BUTTON后，计数器值会否减1？为什么？

- 不会，因为模式一计数是从gate处于上升沿时开始，而这里gate连接电源，始终保持高电平，不会出现上升沿。

3) 接线原理图和运行结果图



二) 信号发生器

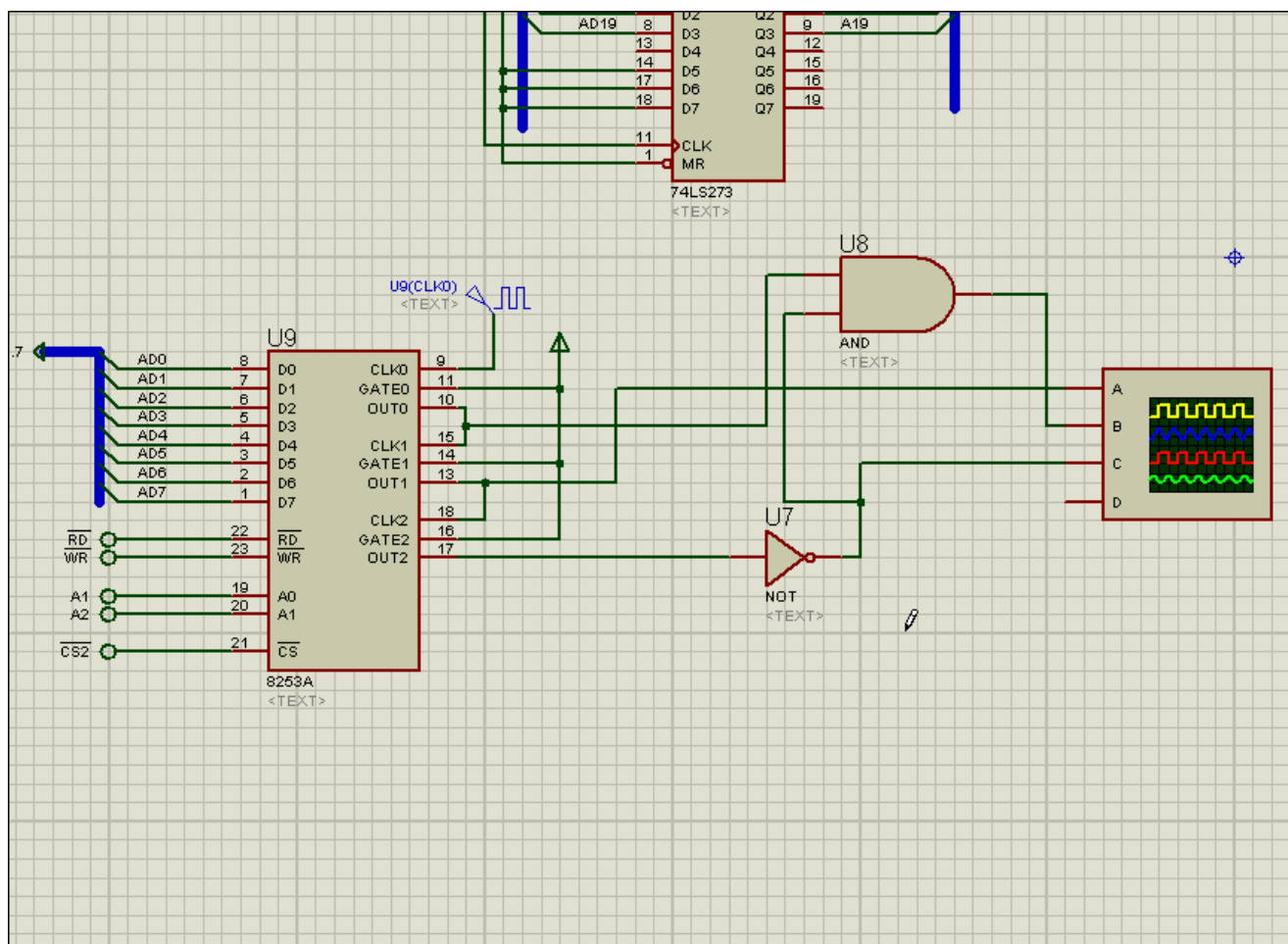
1) 源程序

```

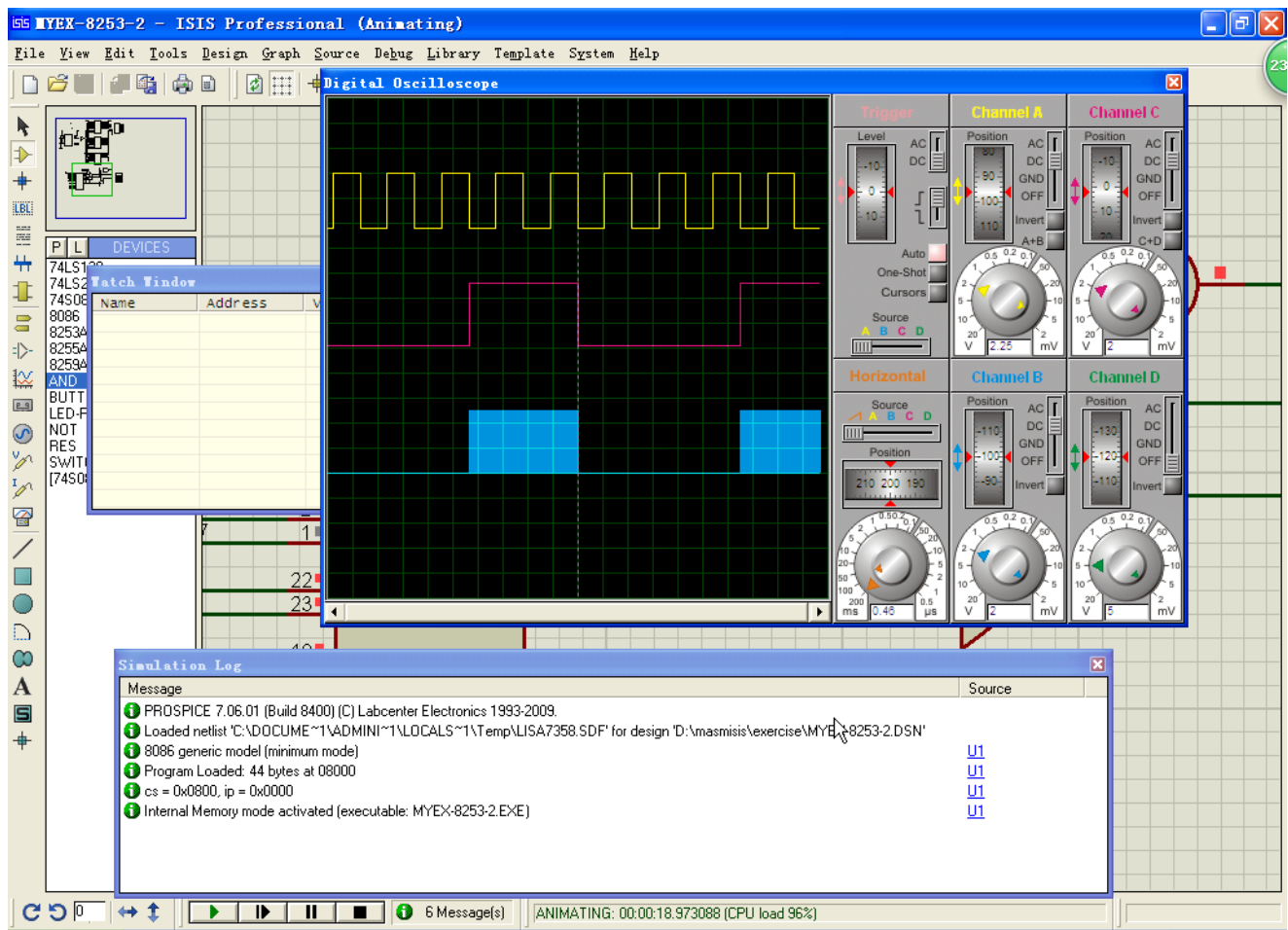
1  .8086
2  .MODEL SMALL
3  .STACK
4  .DATA
5  .CODE
6  START:
7      MOV AL,00110111B ; 使用计数器0，先写低八位，后写高八位，使用模式三，BCD码
8      MOV DX,226H
9      OUT DX,AL
10     MOV AX,1000H
11     MOV DX,220H
12     OUT DX,AL
13     MOV AL,AH
14     OUT DX,AL ; 计数器初始值为1000
15
16     MOV AL,01110111B ; 使用计数器1，先写低八位，后写高八位，使用模式三，BCD码
17     MOV DX,226H
18     OUT DX,AL
19     MOV AX,1000H
20     MOV DX,222H
21     OUT DX,AL
22     MOV AL,AH
23     OUT DX,AL ; 计数器初始值为1000
24
25     MOV AL,10010111B ; 使用计数器2，只使用低八位，使用模式三，BCD码
26     MOV DX,226H
27     OUT DX,AL
28     MOV AL,5H
29     MOV DX,224H
30     OUT DX,AL ; 计数器初始值为5
31 END START

```

2) 接线原理图



3) 运行结果图



4) 编程与调试心得（遇到的问题 and 解决的办法，以及获得的收获）

对于OUT2口的波形如果不取反的话就会得到占空比为3:5的方波。

实验五 七段数码管实验

一、实验目的

掌握数码管显示数字的原理，进一步复习8255的应用。

二、实验内容

1. 静态显示（选）

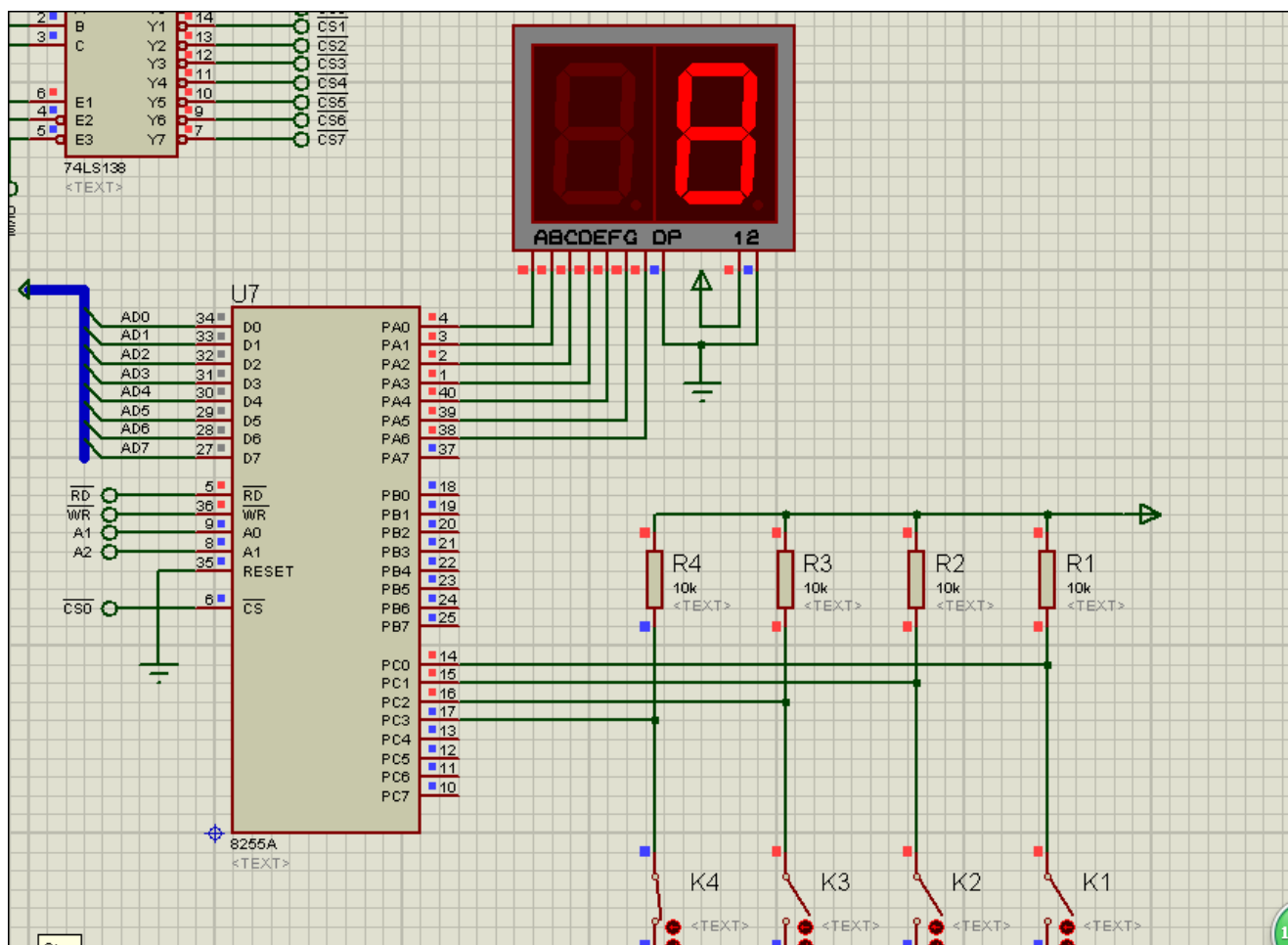
1) 源程序

```

1  .8086
2  .MODEL SMALL
3  .STACK
4  .DATA
5      ARR BYTE 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H
6  .CODE
7  START:
8      MOV AX, @DATA
9      MOV DS, AX
10     MOV AL, 100000001B ; A组模式0, A口和C口高四位为输出。B组模式0, B口为输出, C口低四位为输入。
11     MOV DX, 206H
12     OUT DX, AL
13     MOV DX, 204H
14     IN AL, DX
15     NOT AL
16     MOV AH, 0H
17     MOV DL, 16
18     DIV DL
19     MOV AL, AH
20     MOV AH, 0H
21     MOV BX, AX
22     LEA DI, ARR
23     MOV AL, [DI+BX]
24     MOV DX, 200H
25     OUT DX, AL
26     JMP $
27 END START

```

2) 接线原理图和运行结果图



2.动态展示

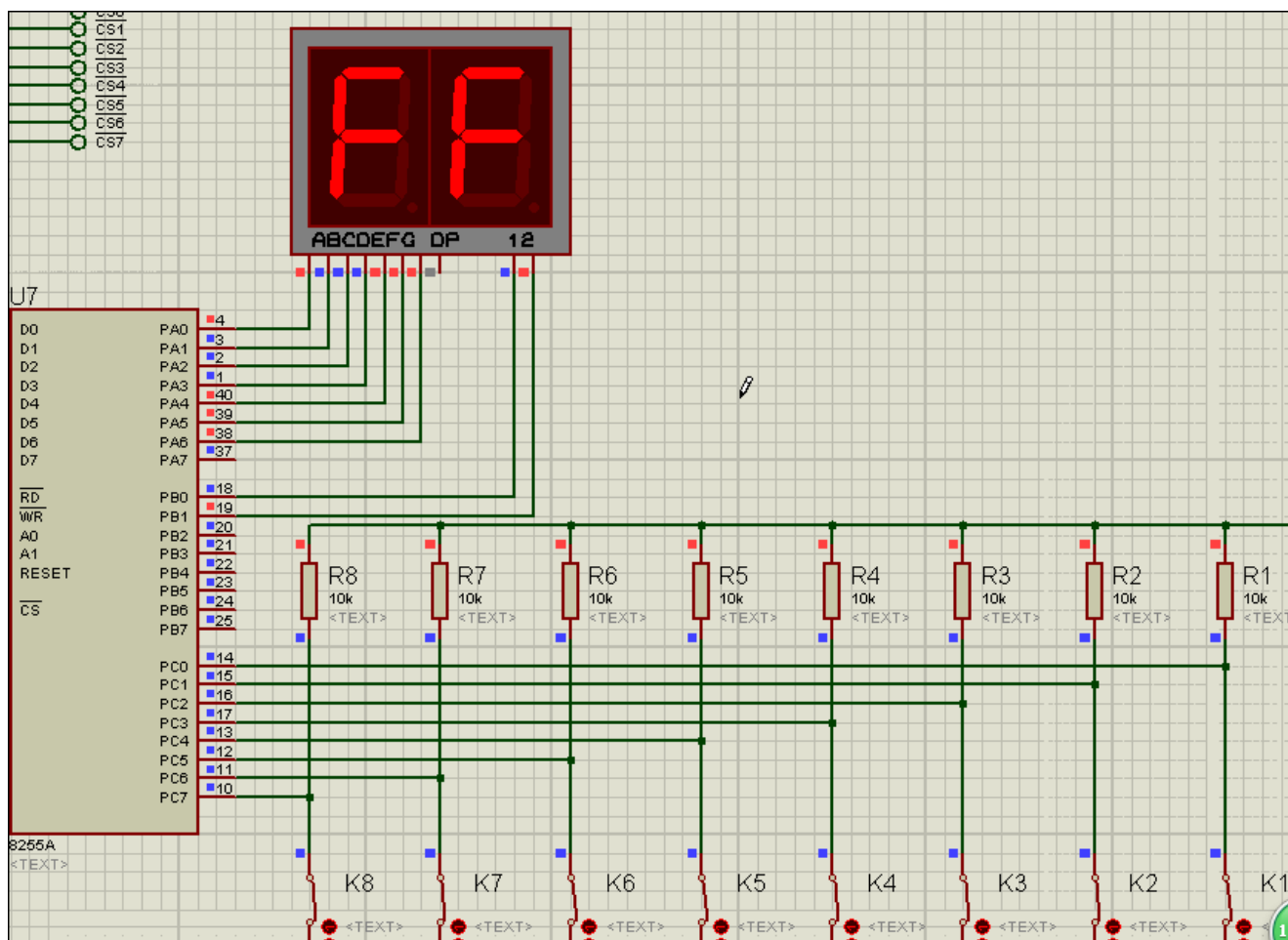
1) 源程序

```

1  .8086
2  .MODEL SMALL
3  .STACK
4  .DATA
5      ARR BYTE 3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH,77H,7CH,39H,5EH,79H,71H
6  .CODE
7  START:
8      MOV AX,@DATA
9      MOV DS,AX
10     MOV AL,10001001B
11     MOV DX,206H
12     OUT DX,AL
13     LEA DI,ARR
14 LP:
15     MOV DX,204H
16     IN AL,DX
17     NOT AL
18     MOV AH,0
19     MOV DL,16
20     DIV DL
21     MOV SI,AX
22     MOV AL,01H
23     MOV DX,202H
24     OUT DX,AL
25     MOV BX,SI
26     MOV BH,0
27     MOV DX,200H
28     MOV AL,[DI+BX]
29     OUT DX,AL
30     MOV CX,64H
31     LOOP $
32     MOV AL,02H
33     MOV DX,202H
34     OUT DX,AL
35     MOV BX,SI
36     MOV BL,BH
37     MOV BH,0
38     MOV DX,200H
39     MOV AL,[DI+BX]
40     OUT DX,AL
41     MOV CX,64H
42     LOOP $
43     JMP LP
44 END START

```

2) 接线原理图和运行结果图



3) 编程与调试心得（遇到的问题 and 解决的办法，以及获得的收获）

刚开始只有一个数字有显示，后来才想到通过PB0/1片选高位还是低位亮。其实是高位和低位交替亮。这个实验让我对8255的使用有了更深的体会。

实验六 交通灯控制实验

一、实验目的

通过并行接口8255实现十字路口交通灯的模拟控制,进一步掌握对并行口的使用和中断的使用。

二、交通灯变化规律

十字路口交通灯的变化规律要求

- 1) 南北路口的绿灯、东西路口的红灯同时亮10秒，然后
- 2) 南北路口的黄灯闪烁2次（闪烁周期2秒），同时东西路口的红灯继续亮，然后
- 3) 南北路口的红灯、东西路口的绿灯同时亮10秒，然后
- 4) 南北路口的红灯继续亮，同时东西路口的黄灯2次（闪烁周期2秒），然后
- 5) 转1重复。

三、实验内容

红黄绿三色交通灯控制。将接口实验包中所带的EX-825X-1.DSN文件用ISIS 7打开，将D6、D5、D4作为南北路口的交通灯与PA7、PA6、PA5相连；D3、D2、D1作为东西路口的交通灯与PA2、PA1、PA0相连(方式0)。利用8253产生1秒的中断信号(系统已经为CLK0提供了一个1MHz的时钟输入)，在中断处理程序中用程序处理10秒延迟和两次黄灯闪烁的问题。编程使六个灯按交通灯变化规律燃灭。

1) 源程序

```

1  .8086
2  .MODEL SMALL
3  .STACK
4  .DATA
5      COUNT BYTE 0H
6  .CODE
7  START:
8      MOV AX, @DATA
9      MOV DS, AX
10     MOV DX, 226H
11     MOV AL, 00110111B
12     OUT DX, AL
13     MOV AX, 1000H
14     MOV DX, 220H
15     OUT DX, AL
16     MOV AL, AH
17     OUT DX, AL ;C0
18     MOV DX, 226H
19     MOV AL, 01110111B
20     OUT DX, AL
21     MOV AX, 1000H
22     MOV DX, 222H
23     OUT DX, AL
24     MOV AL, AH
25     OUT DX, AL ;C1
26     MOV AL, 10000000B
27     MOV DX, 206H
28     OUT DX, AL
29     MOV AL, 10000001B
30     MOV DX, 200H
31     OUT DX, AL
32     CLI
33     MOV AX, 0
34     MOV ES, AX
35     MOV SI, 0
36     MOV CX, 255
37 L:   MOV AX, OFFSET INTO
38     MOV ES: [SI], AX
39     MOV AX, SEG INTO
40     MOV ES: [SI+2], AX
41     ADD SI, 4
42     LOOP L
43     MOV AX, OFFSET INTO
44     MOV ES: [SI], AX
45     MOV AX, SEG INTO
46     MOV ES: [SI+2], AX
47     MOV AX, @DATA
48     MOV DS, AX
49     STI
50     MOV AL, 00010011B ; INIT 8259
51     MOV DX, 210H
52     OUT DX, AL ; ICW1
53     MOV AL, 00001000B

```

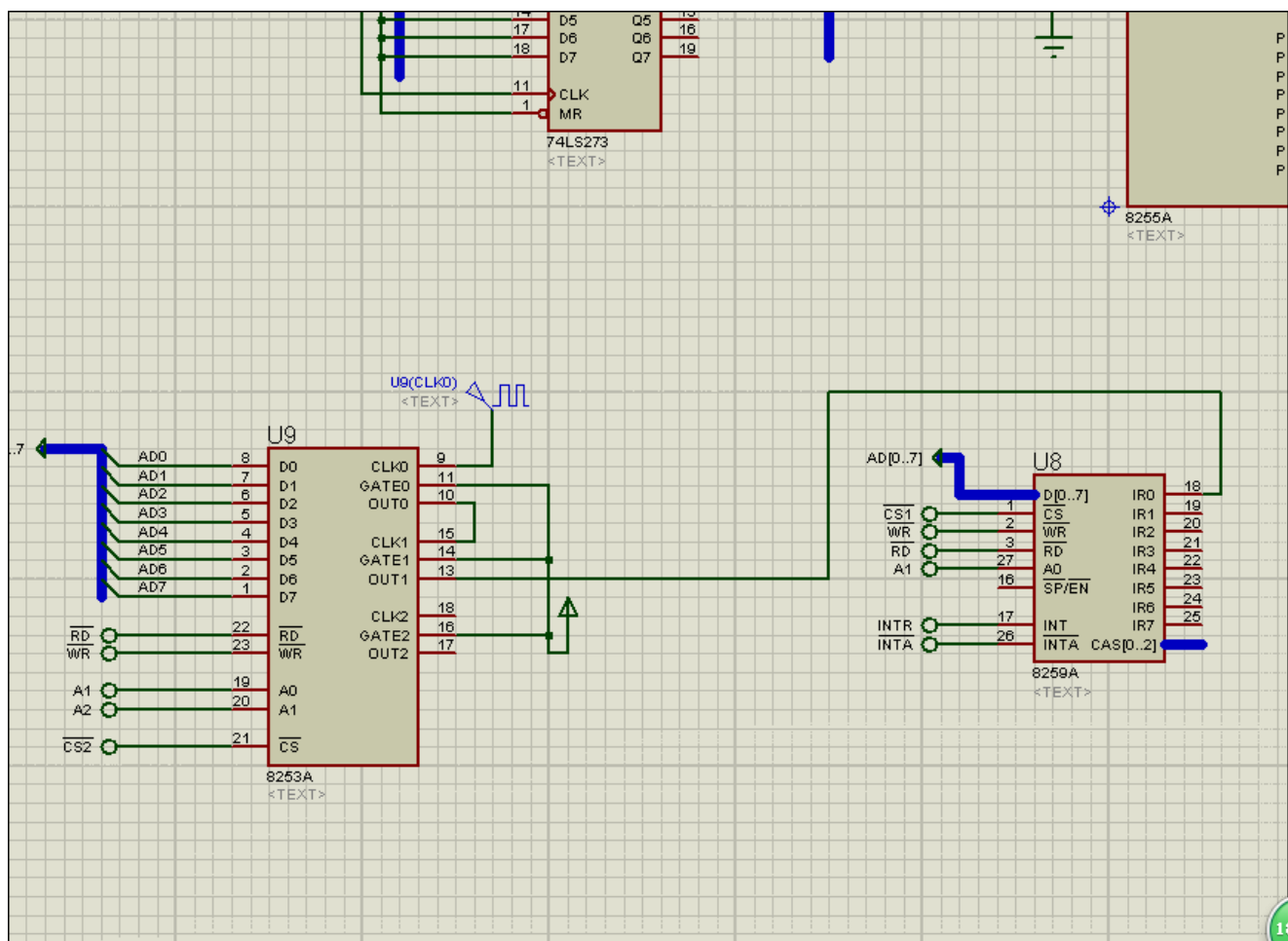
```

54      MOV DX, 212H
55      OUT DX, AL
56      MOV AL, 01H
57      OUT DX, AL
58      JMP $
59      INT0 PROC
60      PUSH SI
61      PUSH AX
62      PUSH DX
63      PUSH DS
64      MOV AX, @DATA
65      MOV DS, AX
66      MOV AL, COUNT
67      INC AL
68      MOV COUNT, AL
69      CMP AL, 10
70      JZ D01
71      CMP AL, 11
72      JZ D02
73      CMP AL, 12
74      JZ D01
75      CMP AL, 13
76      JZ D02
77      CMP AL, 14
78      JZ D03
79      CMP AL, 24
80      JZ D04
81      CMP AL, 25
82      JZ D05
83      CMP AL, 26
84      JZ D04
85      CMP AL, 27
86      JZ D05
87      CMP AL, 28
88      JZ D06
89      JMP EXIT
90  D06:
91      MOV AL, 10000001B ;左绿右红
92      MOV DX, 200H
93      OUT DX, AL
94      MOV AL, 0
95      MOV COUNT, AL
96      JMP EXIT
97  D01:
98      MOV AL, 01000001B ;左黄灯亮
99      MOV DX, 200H
100     OUT DX, AL
101     JMP EXIT
102  D02:
103     MOV AL, 00000001B ;左黄灯暗
104     MOV DX, 200H
105     OUT DX, AL
106     JMP EXIT

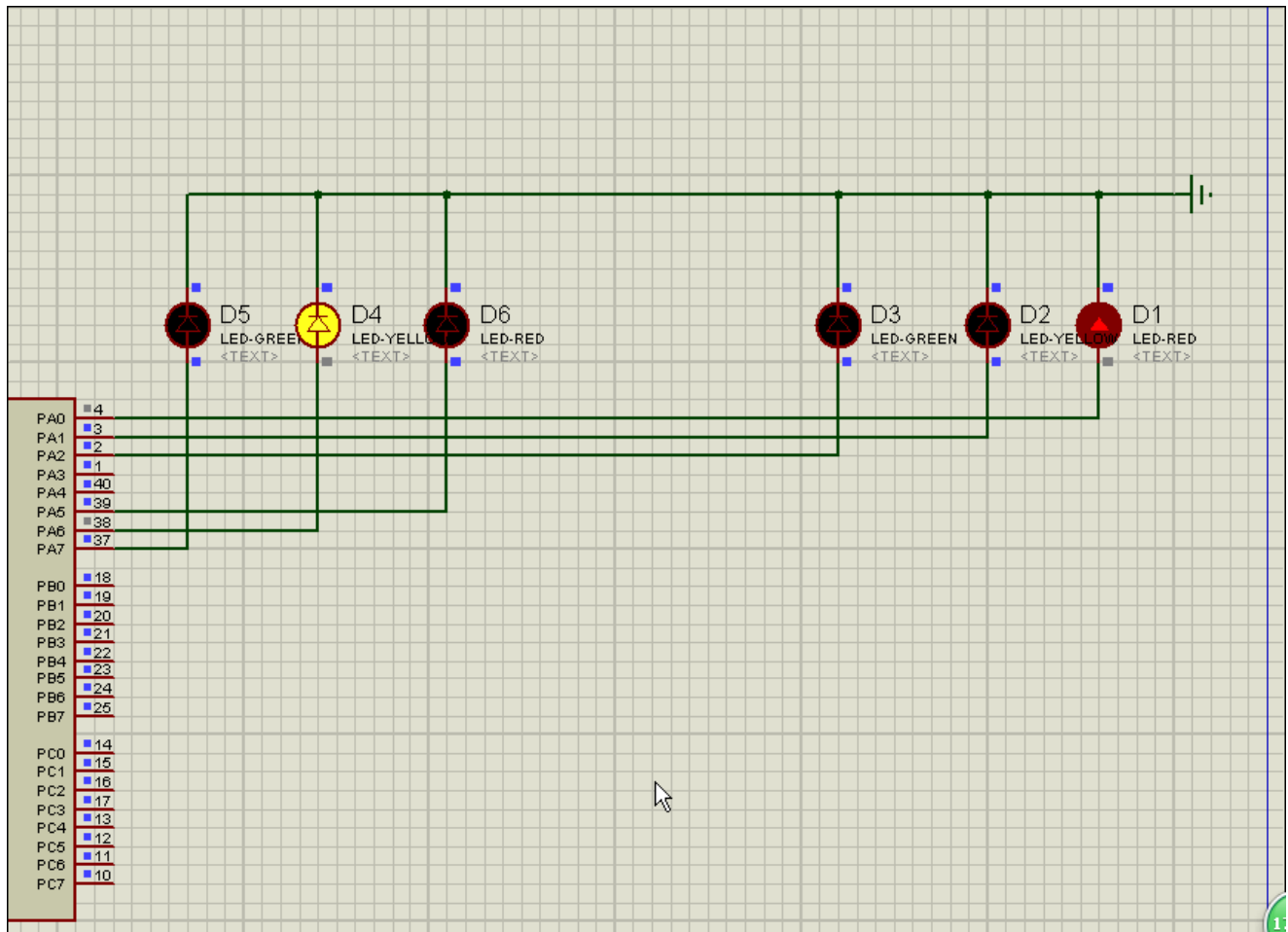
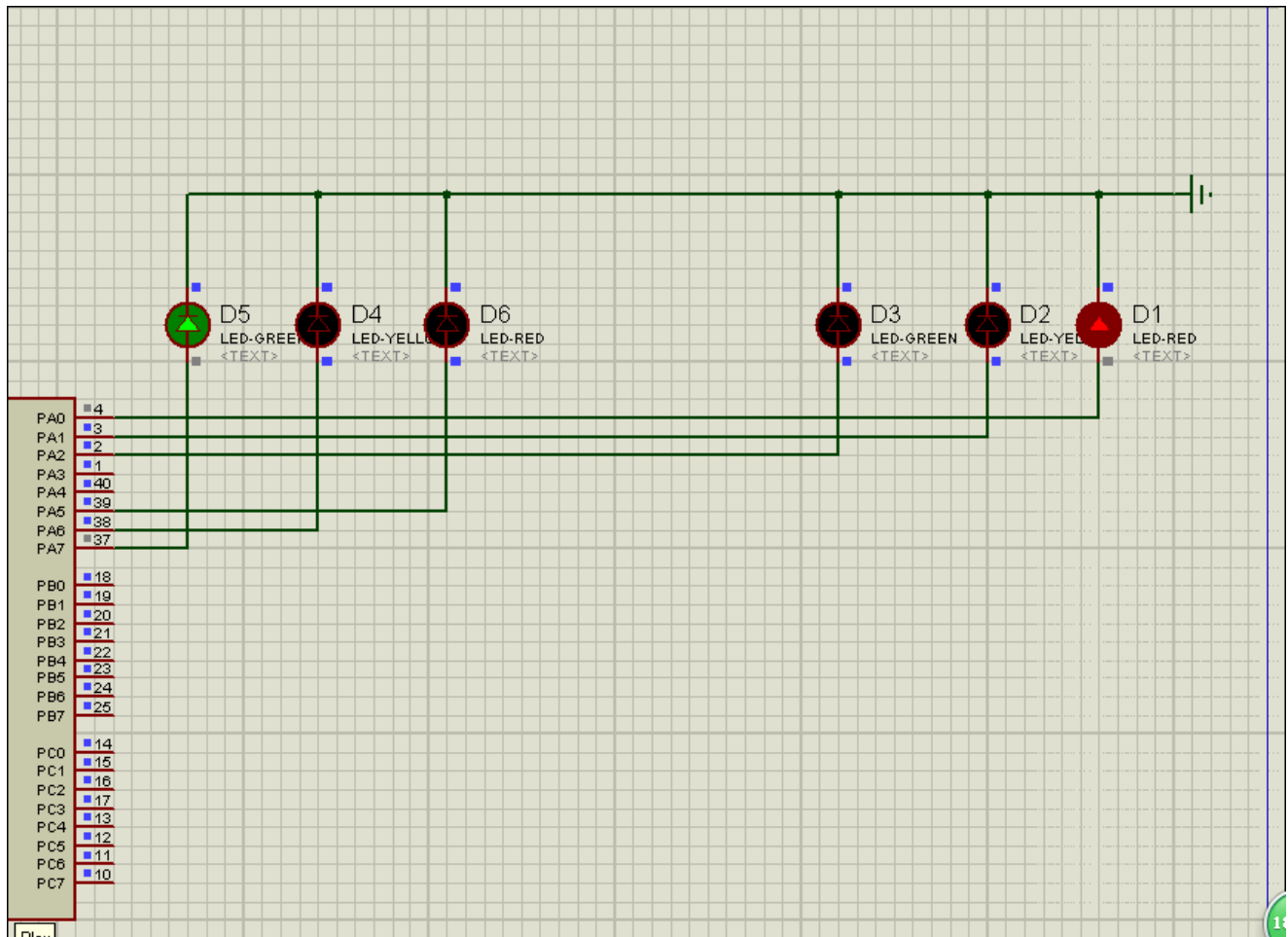
```

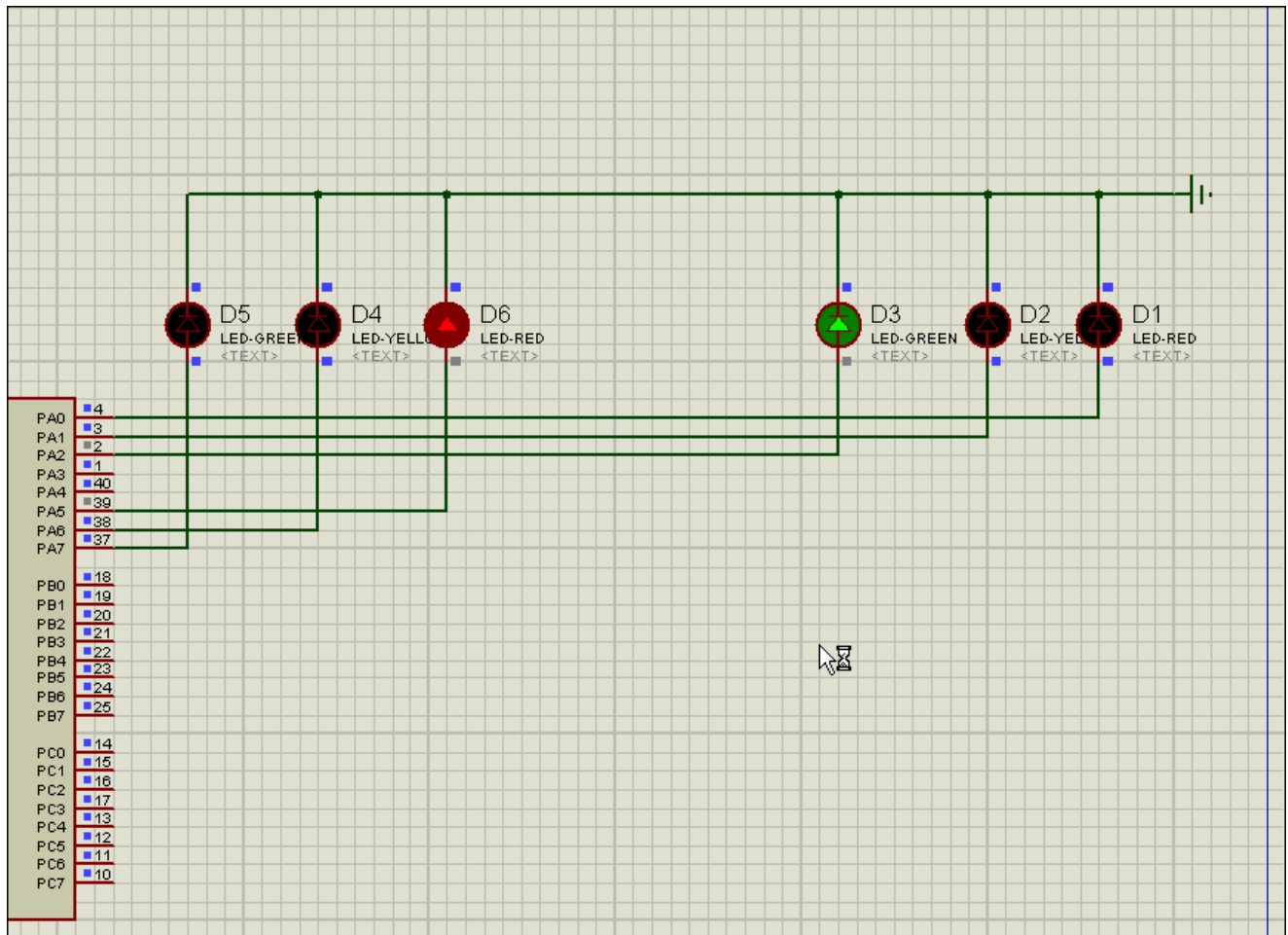
```
107 D03:
108     MOV AL,00100100B ;左红右绿
109     MOV DX,200H
110     OUT DX,AL
111     JMP EXIT
112 D04:
113     MOV AL,00100010B ;右黄灯亮
114     MOV DX,200H
115     OUT DX,AL
116     JMP EXIT
117 D05:
118     MOV AL,00100000B ;右黄灯暗
119     MOV DX,200H
120     OUT DX,AL
121     JMP EXIT
122 EXIT:
123     MOV AL,20H
124     MOV DX,210H
125     OUT DX,AL
126     POP DS
127     POP DX
128     POP AX
129     POP SI
130     IRET
131     INT0 ENDP
132 END START
```

2) 接线原理图



3) 运行结果截图





4) 编程与调试心得（遇到的问题 and 解决的办法，以及获得的收获）

循环很多，一不小心就会写错。如何让黄灯闪烁也想了很久。