

《计算机硬件技术》实验作业

第三次仿真实验

学院：汽车学院

姓名：贾林轩

学号：1853688

软件：proteus8

仿真实验三 I/O 接口电路

一、实验目的

熟悉硬件仿真软件 proteus 界面及相关操作，理解电路设计原理及程序设计思路，重点理解 IO 端口的输入输出功能及地址译码功能。

二、实验内容

软件打开所给文件夹 计算机硬件仿真/sim (1) /查询开关.pdspri 示例，理解该硬件电路实现原理。本次实验任务需要在此基础上对原理图或程序做出修改，并成功执行，报告中需要演示并说明该硬件电路实现了什么功能以及创新点。

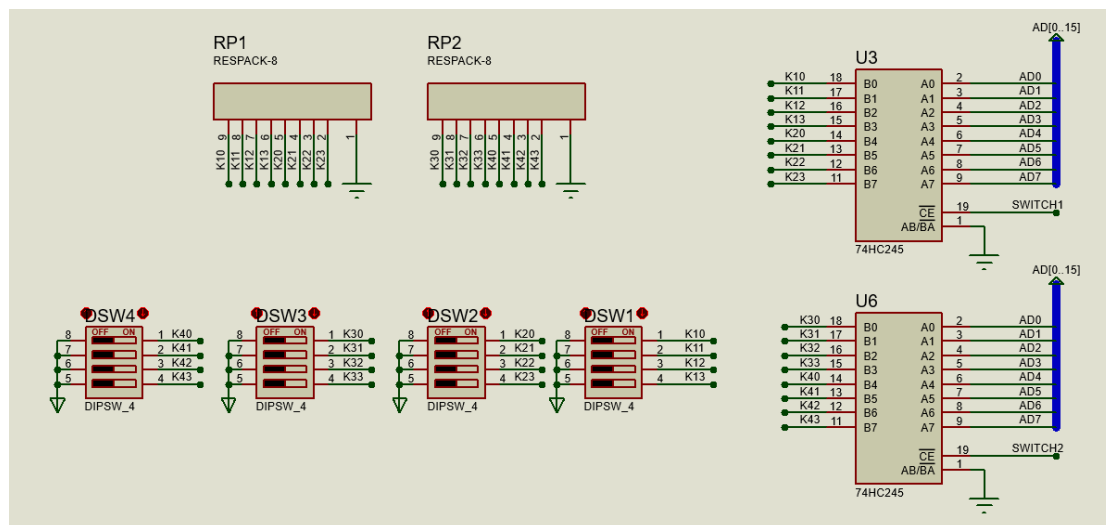
三、实验过程

1. 目标实现功能

- ① 通过四个四通道开关，通过**动态显示**，共同控制一个数码管进行四位十六进制数的显示:页码: P2
- ② 用开关和**矩阵键盘**分别实现数码管的段选和位选，从而实现:“选定显示的位——显示特定的数”的功能:页码: P10

2. 硬件电路设计思路

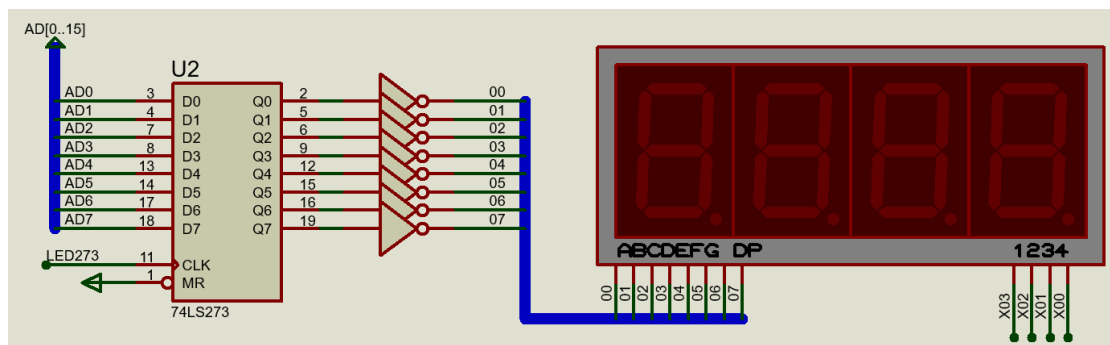
① 输入接口



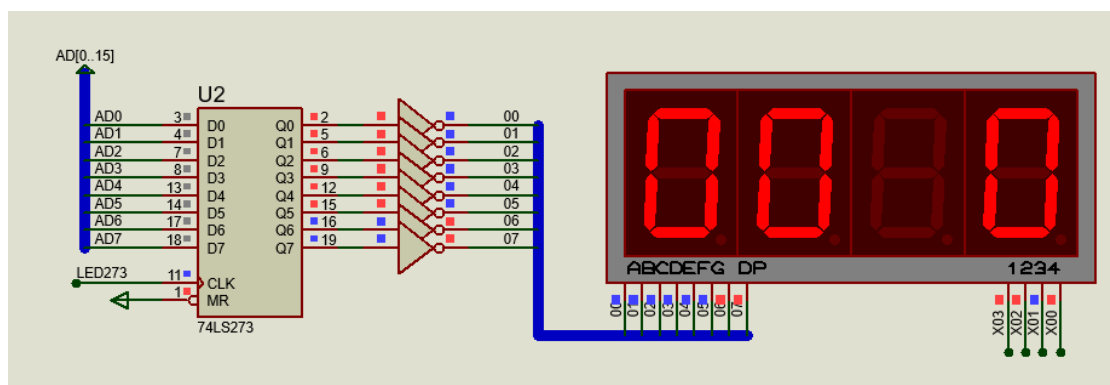
通过下拉电阻控制开关闭合无输入时为低电阻，四个开关分别对应数码管的四个位，第一位对应开关连接到第一片 74HC245 的低四位（K10-K13），第二位对应开关连接到第一片芯片的高四位（K20-K23），第三位对应开关连接到第二片芯片的低四位（K30-K33），第四位对应开关连接到第二片芯片的高四位（K40-K43）；

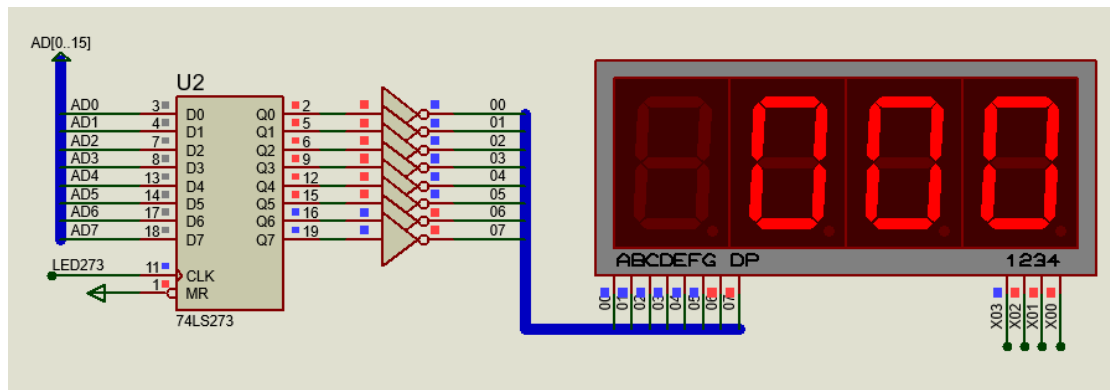
通过地址译码选定输入接口（SWITCH1 和 SWITCH2）后即可从接口读到开关状态信息（下文软件程序部分详述）；

② 输出接口

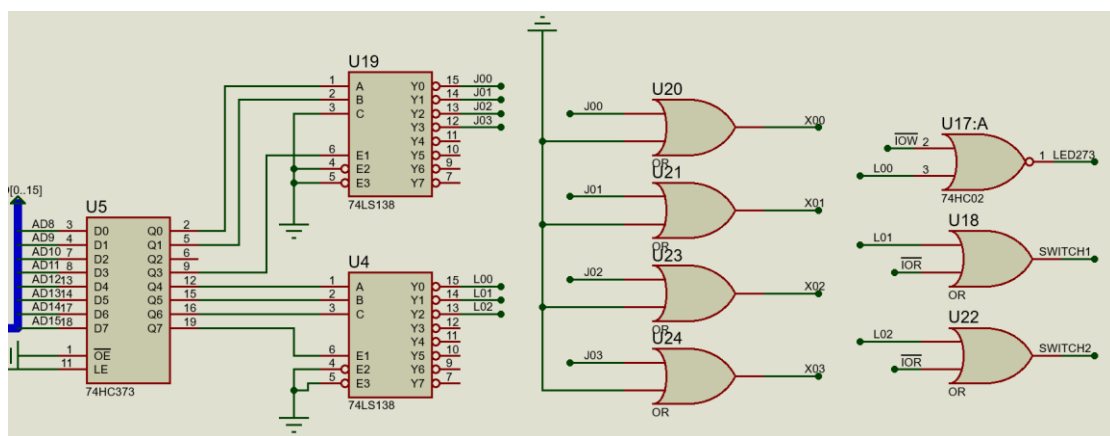


通过地址译码进行位选（X00, X01, X02, X03），选定不同的位后，将数据信号输入到输出接口（LED273），重复上述过程，当显示频率较低时（通过延时子程序实现），可以看到输出有明显的闪烁（如下图所示），当显示频率超过人眼的识别频率时，即看不见闪烁，达到动态显示效果；





③ 译码电路



上文分析到，对四位数码管进行位选只需两位地址码即可，因此将 Q0, Q1

（AD8, AD9）接到 74LS138 的 A, B 端，将 C 端接地；

根据此译码电路设计，可分析出各输入输出接口地址：

9000H: 输入接口 SWITCH1

0A00H: 输入接口 SWITCH2

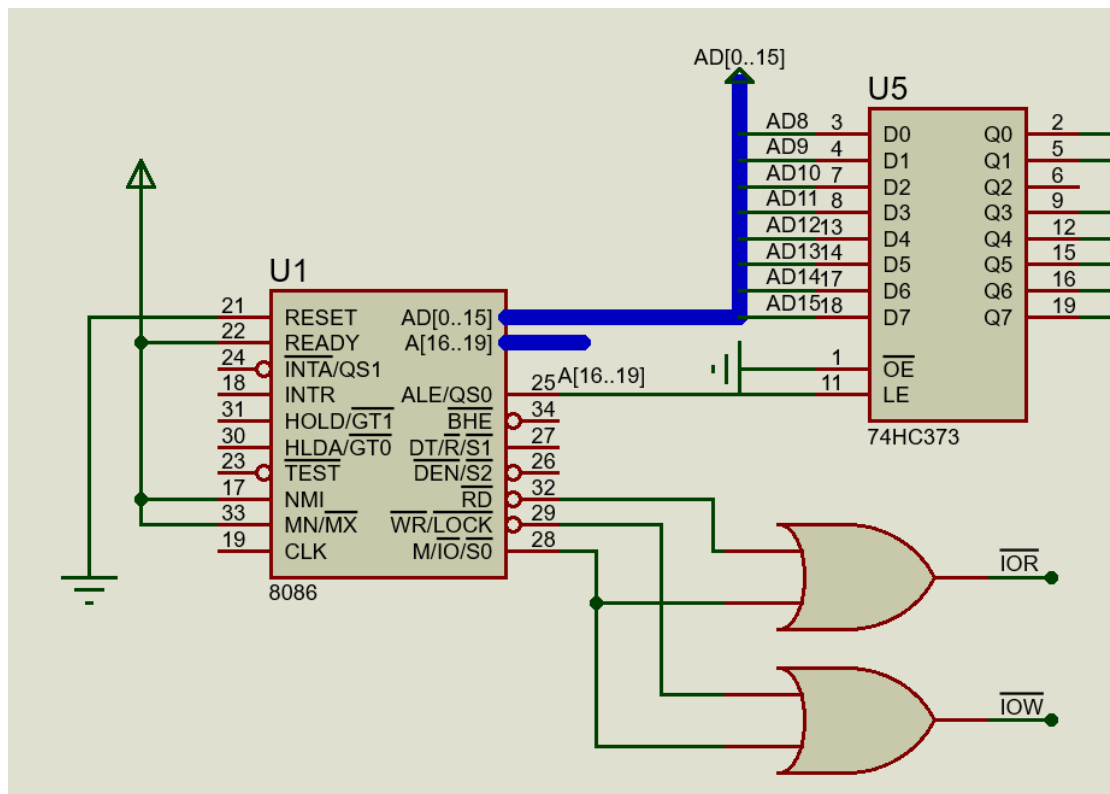
8800H: 输出接口 1（位选数码管第 1 位，数据输出到 LED273）

8900H: 输出接口 2（位选数码管第 2 位，数据输出到 LED273）

8A00H: 输出接口 3（位选数码管第 3 位，数据输出到 LED273）

8B00H: 输出接口 4（位选数码管第 4 位，数据输出到 LED273）

④ CPU 及外围电路（无变化）



3. 软件程序设计思路（不同点已加粗表示）

```
STACK SEGMENT 'STACK'           ;定义堆栈段
    STA DB 100 DUP(?)           ;定义堆栈大小
    TOP EQU LENGTH STA          ;栈顶偏移地址
STACK ENDS
```

```
DATA SEGMENT 'DATA'
    ORG 2H                       ;数据段起始地址
    LED1 EQU 8800H               ;输出端口 1 地址
    LED2 EQU 8900H               ;输出端口 2 地址
    LED3 EQU 8A00H               ;输出端口 3 地址
    LED4 EQU 8B00H               ;输出端口 4 地址
    SWITCH1 EQU 9000H            ;输入端口 SWITCH1 地址
    SWITCH2 EQU 0A000H           ;输入端口 SWITCH2 地址
```

SEG7 DB 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h ;SEG7 储存七段管
码表 (0-F 共十六个数)

DB 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h

DATA ENDS

CODE SEGMENT 'CODE'

ASSUME CS:CODE, SS:STACK, DS:DATA

START: MOV AX, DATA

MOV DS, AX

MOV AX, STACK

MOV SS, AX

MOV AX, TOP

MOV SP, AX ;初始化段寄存器

LEA BX, SEG7 ;取 7 段码表基地址

MOV AX, 0H ;AX 清零

GO:

;低两位

;第一位

MOV DX, SWITCH1 ;状态接口的地址

IN AL, DX ;读入开关状态

AND AL, 0FH ;保留低 4 位

MOV SI, AX ;作为 7 段码表的表内位移量

MOV AL, [BX+SI] ;取 7 段码

MOV DX, LED4 ;输出接口地址

OUT DX, AL ;输出到对应接口

;CALL DELAY

;第二位

MOV DX, SWITCH1 ;状态接口的地址

IN AL, DX	;读入开关状态
AND AL, 0F0H	;保留高 4 位
MOV CL, 4	
SHR AL, CL	;AL 右移四位
MOV SI, AX	;作为 7 段码表的表内位移量
MOV AL, [BX+SI]	;取 7 段码
MOV DX, LED1	;输出接口地址
OUT DX, AL	;输出到对应接口
;CALL DELAY	
;高两位	
;第三位	
MOV DX, SWITCH2	;状态接口的地址
IN AL, DX	;读入开关状态
AND AL, 0FH	;保留低 4 位
MOV SI, AX	;作为 7 段码表的表内位移量
MOV AL, [BX+SI]	;取 7 段码
MOV DX, LED2	;输出接口地址
OUT DX, AL	;输出到对应接口
;CALL DELAY	
;第四位	
MOV DX, SWITCH2	;状态接口的地址
IN AL, DX	;读入开关状态
AND AL, 0F0H	;保留高 4 位
MOV CL, 4	
SHR AL, CL	;AL 右移四位
MOV SI, AX	;作为 7 段码表的表内位移量
MOV AL, [BX+SI]	;取 7 段码
MOV DX, LED3	;输出接口地址

OUT DX, AL ;输出到对应接口

;CALL DELAY

JMP GO

DELAY PROC ; ----延时子程序----

PUSH CX

MOV CX, 3FFH

DELAY1: NOP

LOOP DELAY1

POP CX

RET

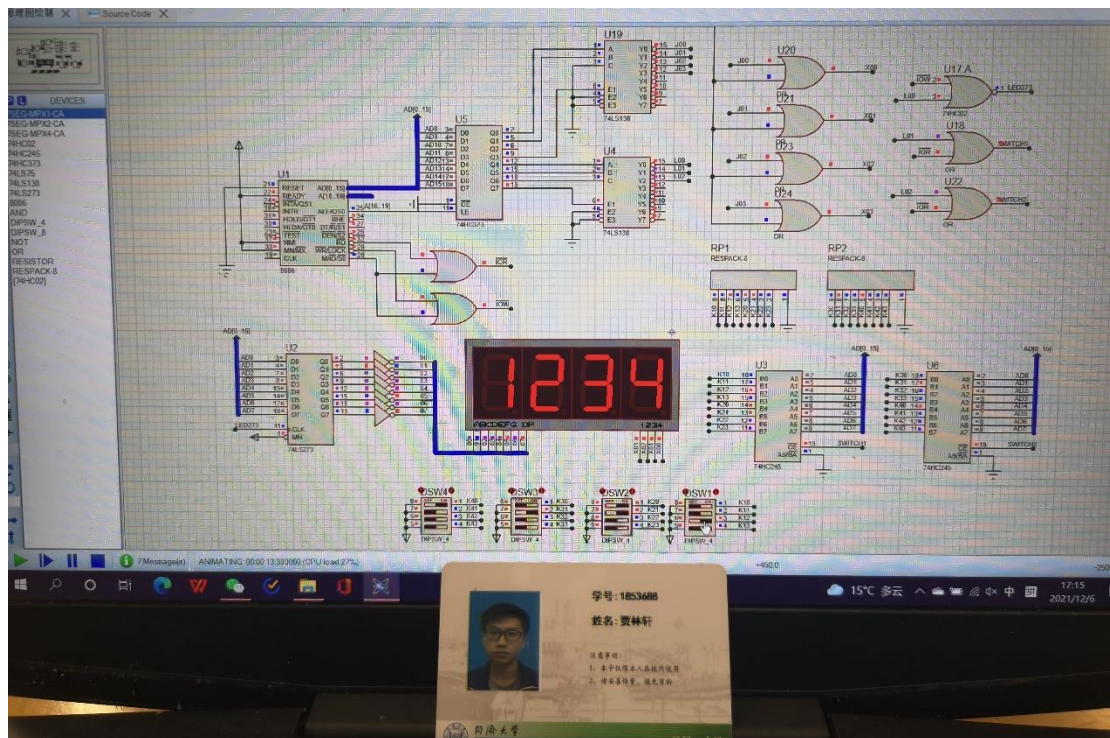
DELAY ENDP

CODE ENDS

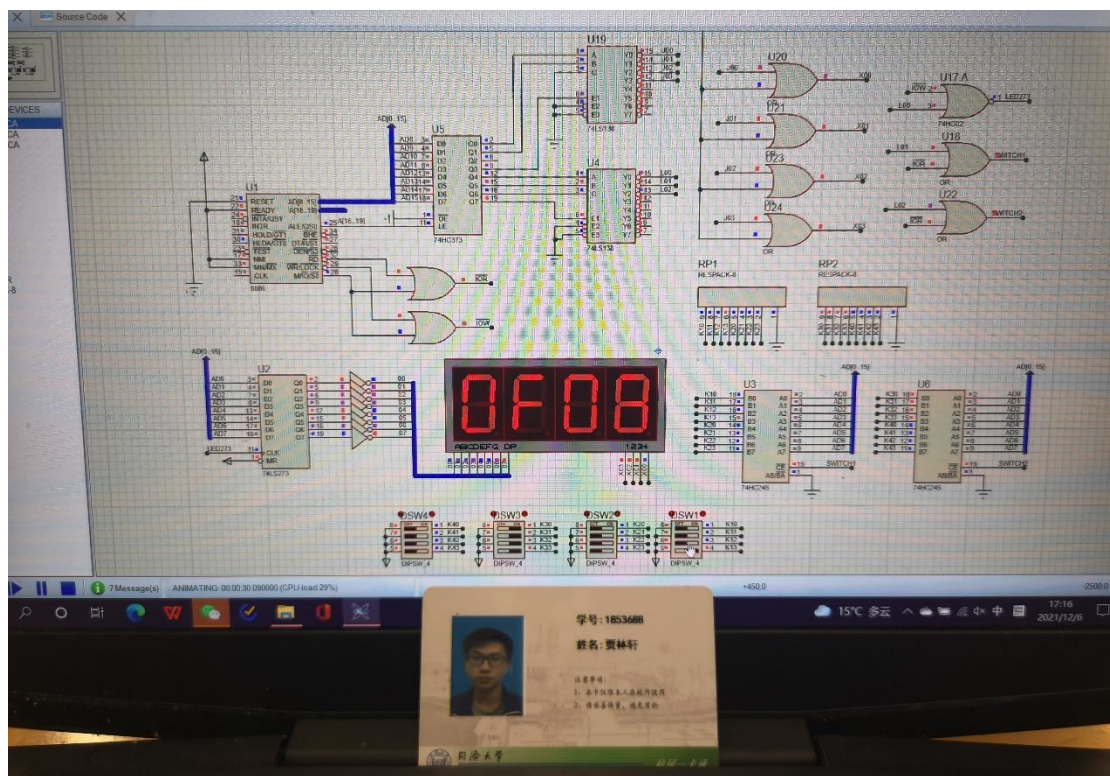
END START

四、仿真结果

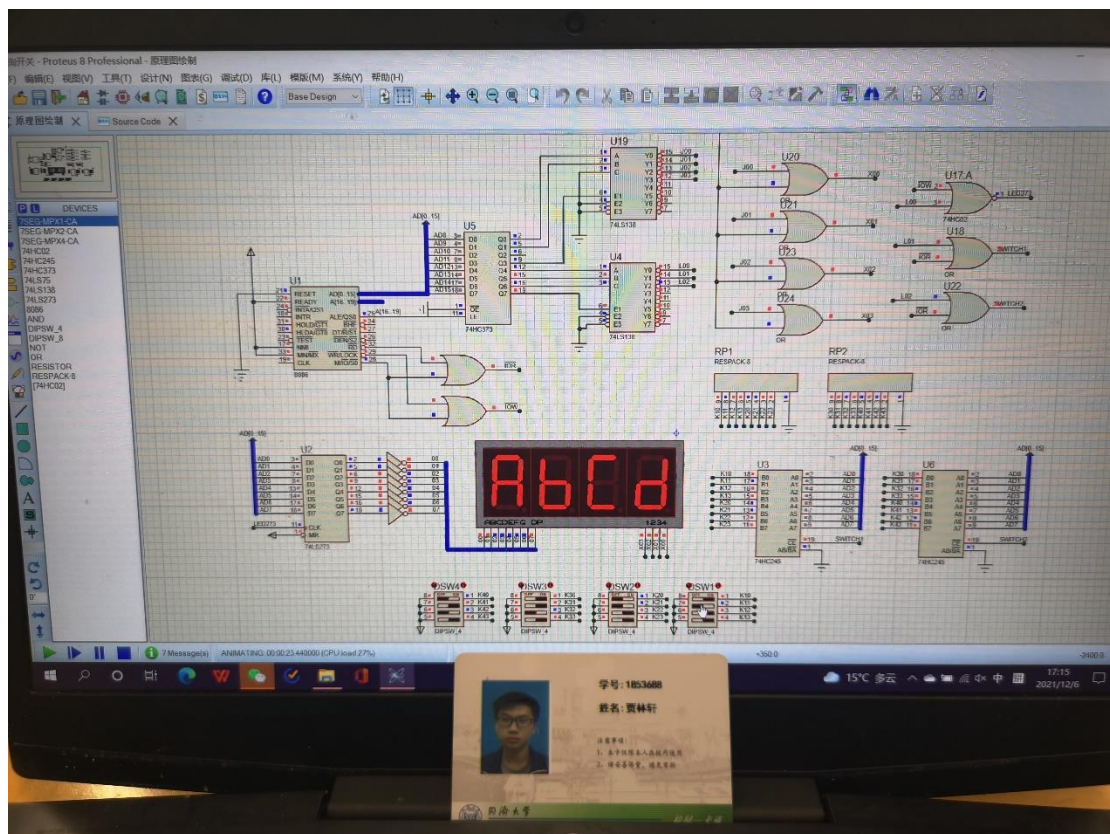
显示数字 1234



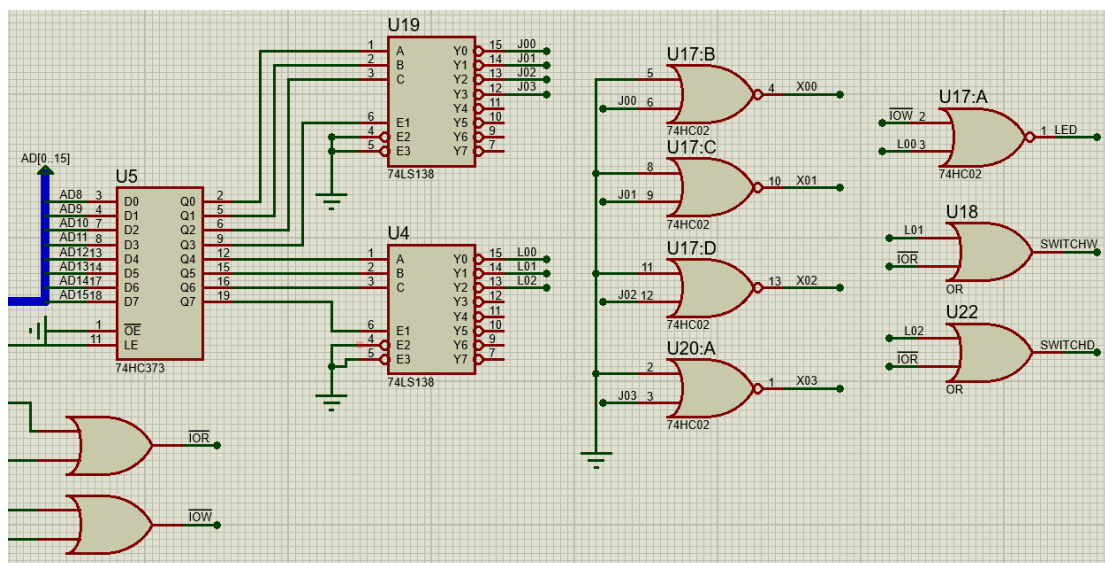
显示数字 0F08



显示数字 ABCD



3. 输入输出端口地址译码电路



根据此译码电路设计，可分析出各输入输出接口地址：

9000H: 位选输入接口 SWITCHW

0A000H: 段选输入接口 SWITCHD

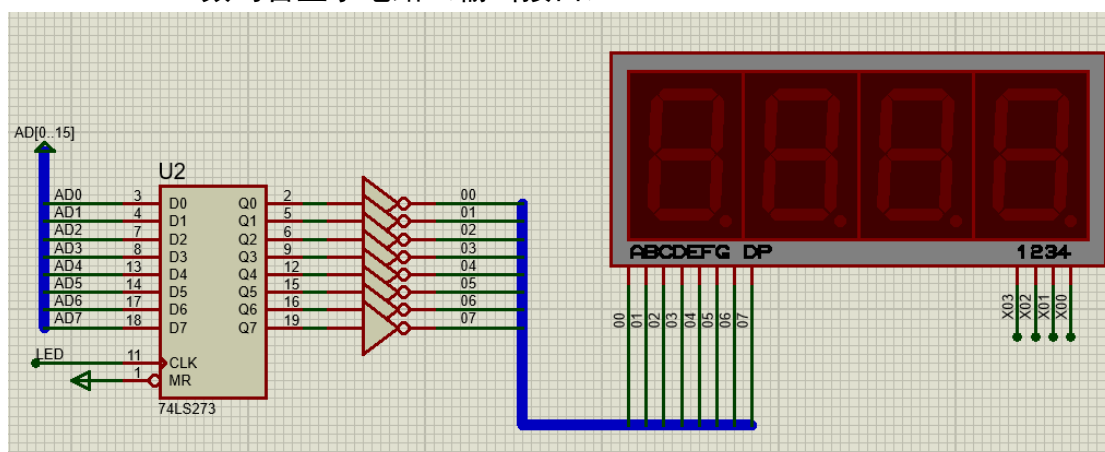
8800H: 输出接口 1（位选数码管第 1 位，数据输出到 LED）

8900H: 输出接口 2（位选数码管第 2 位，数据输出到 LED）

8A00H: 输出接口 3（位选数码管第 3 位，数据输出到 LED）

8B00H: 输出接口 4（位选数码管第 4 位，数据输出到 LED）

4. 数码管显示电路（输出接口）



二. 软件程序设计

```
STACK SEGMENT 'STACK'           ;定义堆栈段
    STA DB 100 DUP(?)           ;定义堆栈大小
    TOP EQU LENGTH STA          ;栈顶偏移地址
STACK ENDS

DATA SEGMENT 'DATA'
    ORG 2H                      ;数据段起始地址
    LED1 EQU 8800H              ;输出端口 1 地址
    LED2 EQU 8900H              ;输出端口 2 地址
    LED3 EQU 8A00H              ;输出端口 3 地址
    LED4 EQU 8B00H              ;输出端口 4 地址
    LED DW LED1, LED2, LED3, LED4 ;将输出端口制表方便取用
    SWITCHW EQU 9000H           ;位选输入端口 SWITCHW 地址
    SWITCHD EQU 0A000H          ;段选输入端口 SWITCHD 地址
    SEG7 DB 3fh, 06h, 5bh, 4fh, 66h, 6dh, 7dh, 07h
    DB 7fh, 6fh, 77h, 7ch, 39h, 5eh, 79h, 71h
DATA ENDS

CODE SEGMENT 'CODE'
    ASSUME CS:CODE, SS:STACK, DS:DATA

START: MOV AX, DATA
        MOV DS, AX
        MOV AX, STACK
        MOV SS, AX
        MOV AX, TOP
        MOV SP, AX              ;初始化段寄存器
        MOV AX, 0H              ;AX 清零

GO:
    MOV DX, SWITCHW             ;取位选接口地址
    IN AX, DX                   ;从位选接口读位选信息
    MOV SI, AX                  ;位选信息作为偏移地址
    LEA BX, LED                 ;取输出接口表
    MOV CX, [BX+SI]             ;暂存输出接口地址

    MOV DX, SWITCHD             ;取段选接口地址
    IN AX, DX                   ;从端选接口读段选信息
    MOV SI, AX                  ;段选信息作为偏移地址
    LEA BX, SEG7                ;取七段管码表
    MOV AL, [BX+SI]             ;取管码
    MOV DX, CX                  ;取输出接口地址
```



```
OUT DX, AL          ;输出
CALL DELAY          ;延时
```

```
JMP GO
```

```
DELAY PROC    ; ----延时子程序----
```

```
PUSH CX
```

```
MOV CX, 03FFH
```

```
DELAY1: NOP
```

```
LOOP DELAY1
```

```
POP CX
```

```
RET
```

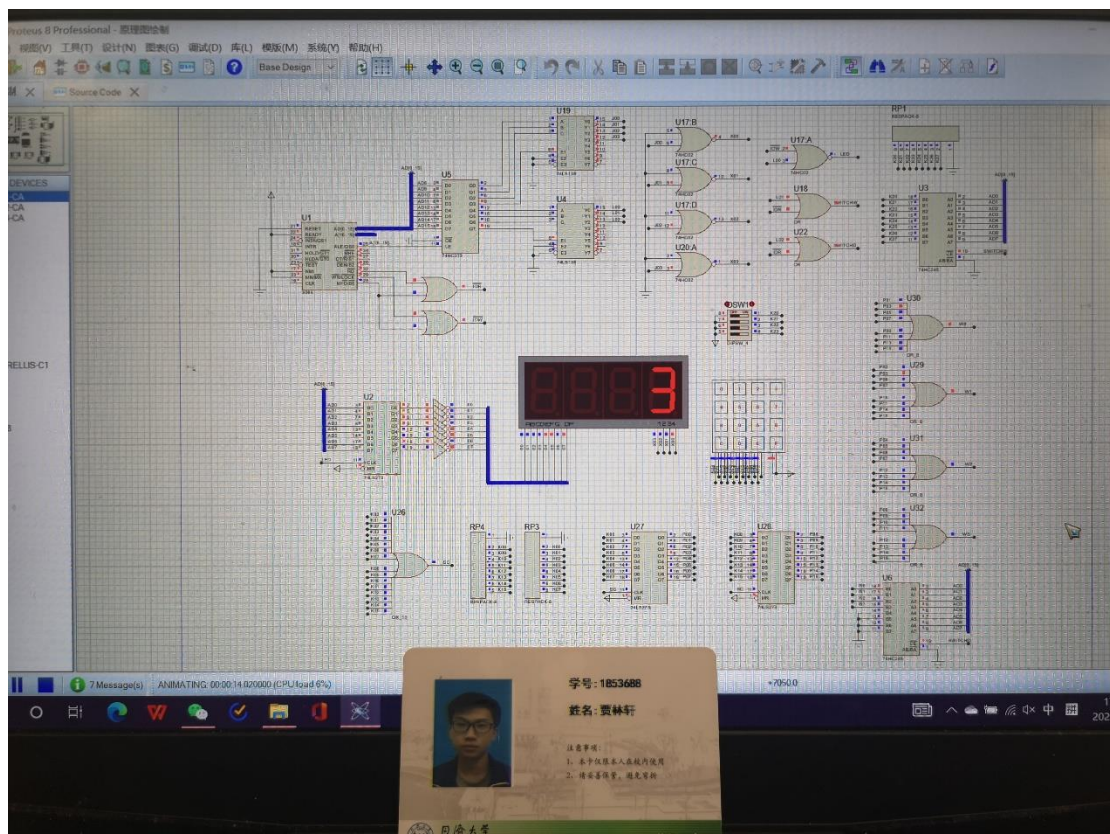
```
DELAY ENDP
```

```
CODE ENDS
```

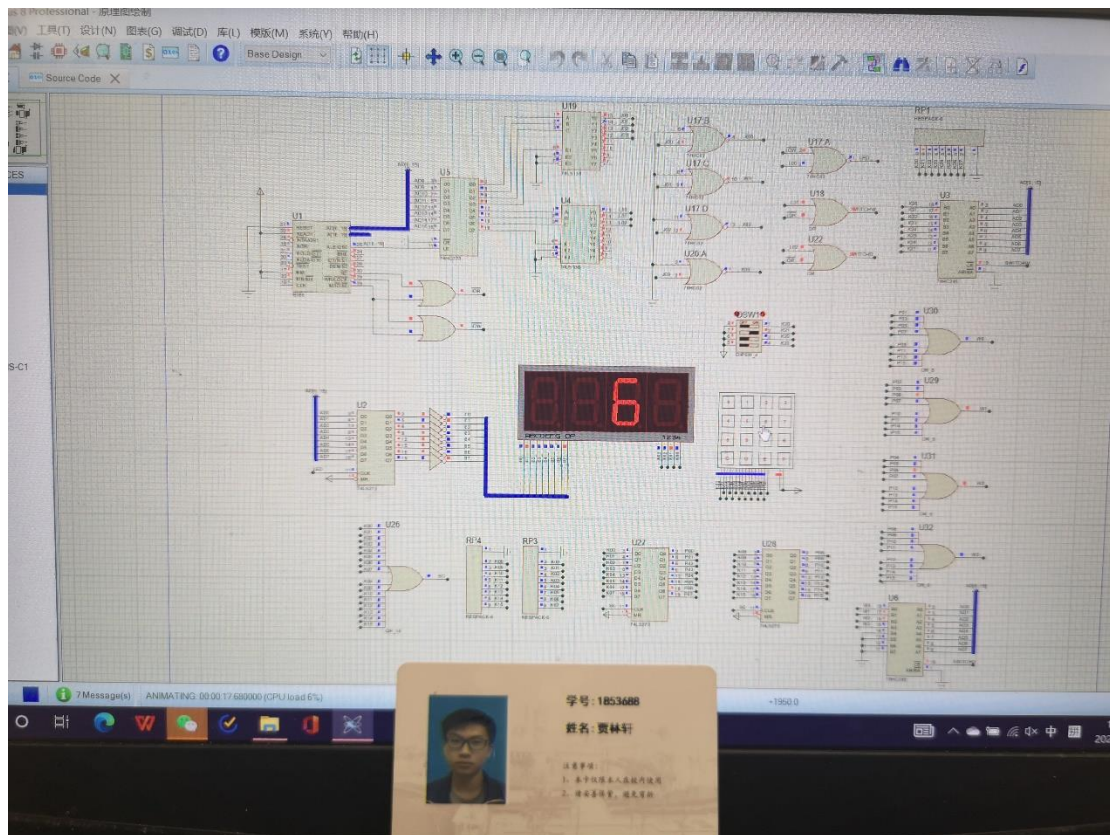
```
END START
```

三. 仿真结果

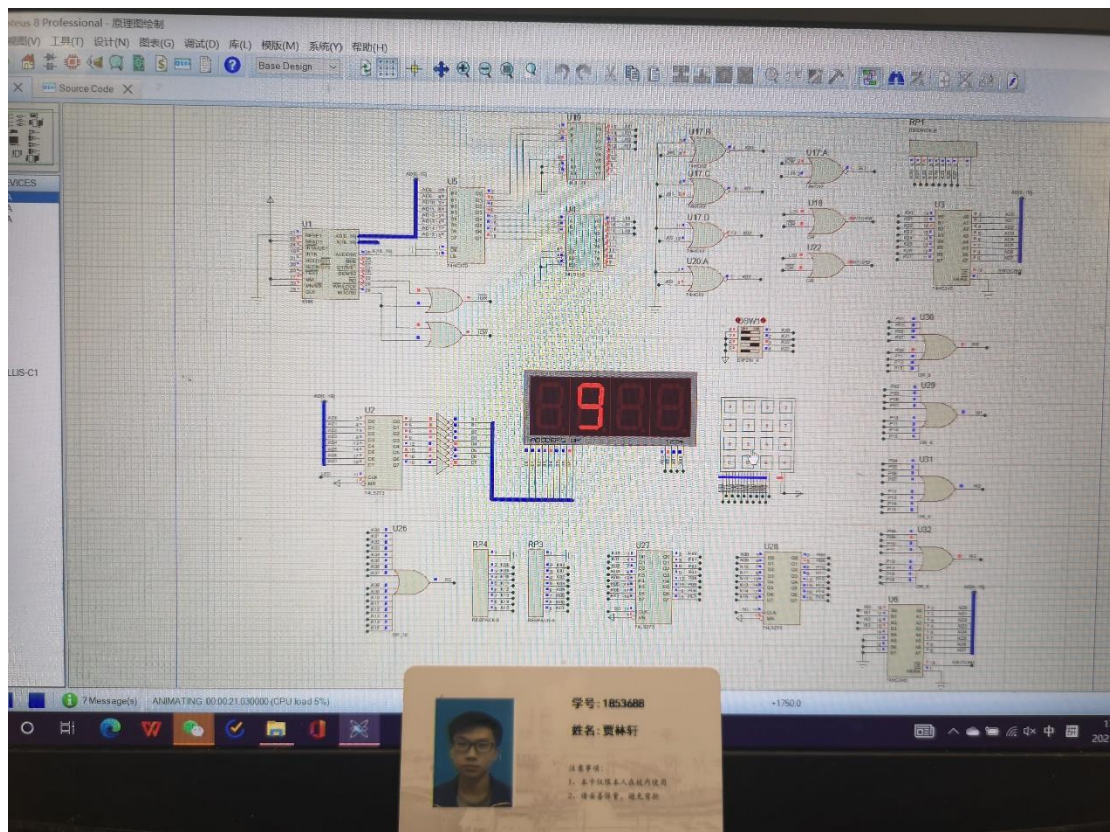
1. 在第 1 位显示 3



2. 在第 2 位显示 6



3. 在第 3 位显示 9



4. 在第 4 位显示 12

