



单片机原理与应用

北京航空航天大学
电子信息工程学院

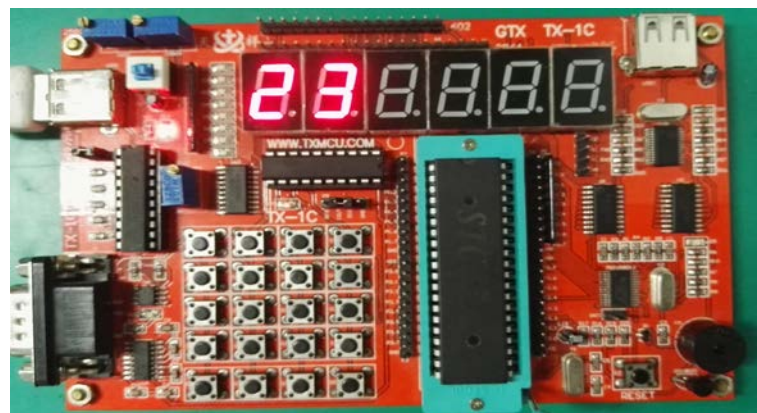
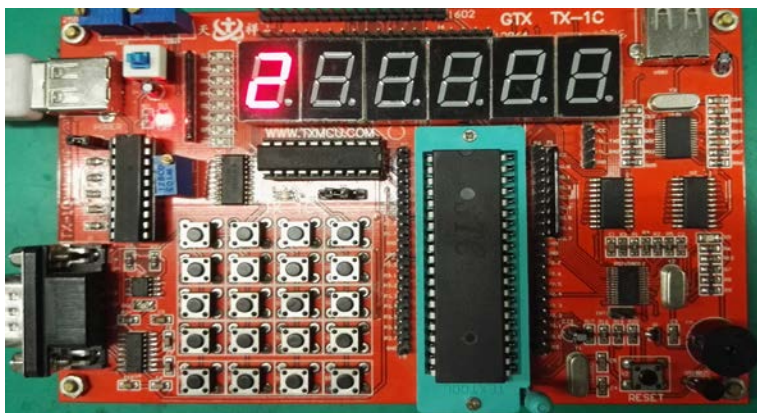
张玉玺

zhangyuxi@buaa.edu.cn

第四章 数码管实验

■ 本章实现功能

- 静态显示：让第一位数码管显示“2”
- 动态显示：让前两位数码管分别显示“2” “3”。



第四章 数码管实验

■ 如何实现功能

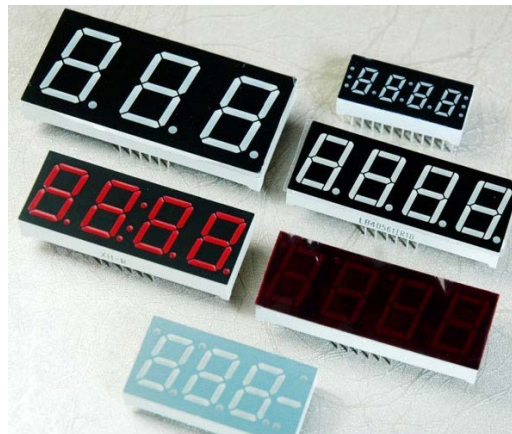
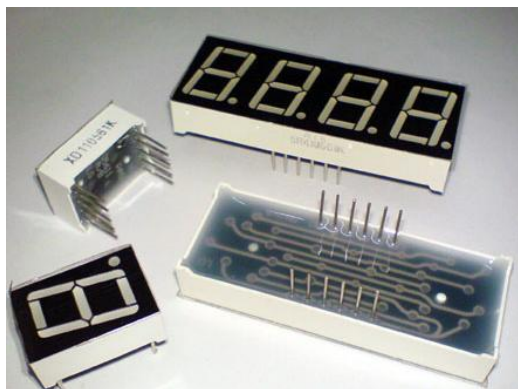
- 数码管工作原理
- I/O寄存器控制
- 静态显示
 - 流程图
 - 程序代码、烧写
- 动态显示
 - 流程图
 - 程序代码、烧写



第四章 数码管实验

■ 数码管工作原理

- 数码管由LED组成
- 内部LED两种电路接法——**共阴**与**共阳**
- 数码管显示由**段选**及**位选**确定
- 两种显示方式——**静态**显示与**动态**显示
- 消去残影

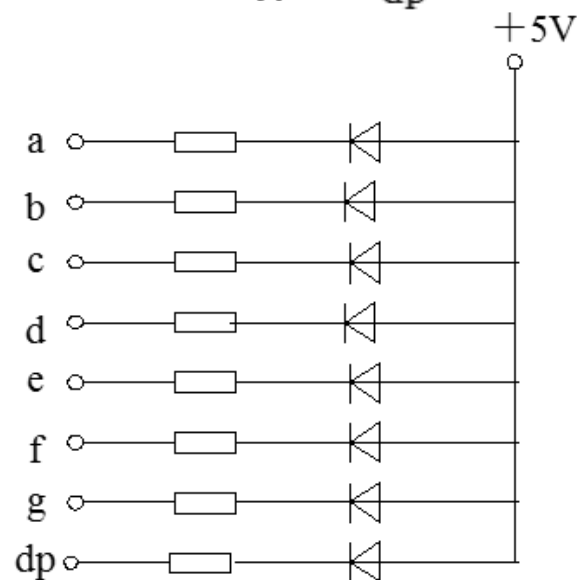
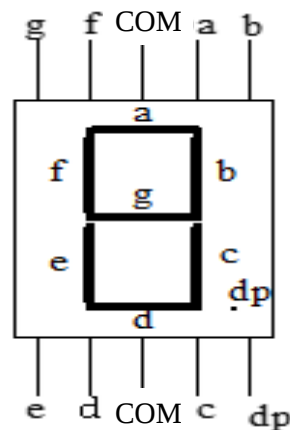


第四章 数码管实验

■ LED两种接法

➤ 共阳极接法

- 所有LED**阳极相连**形成共阳COM端
- 共阳COM端连接到 **+5V**
- LED的阴极接**低电平**时，相应字段亮起



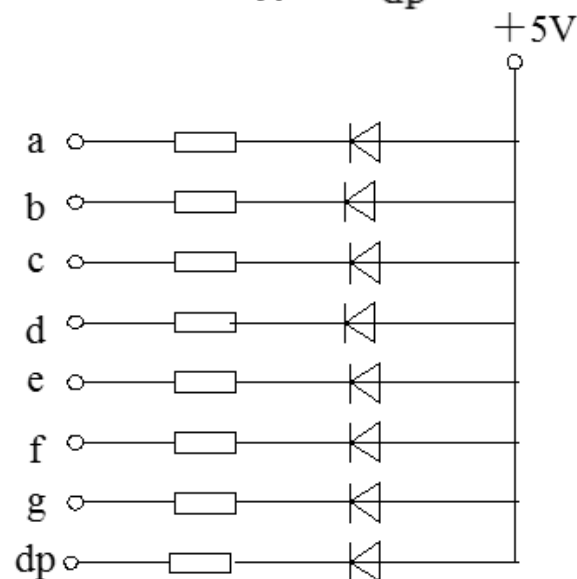
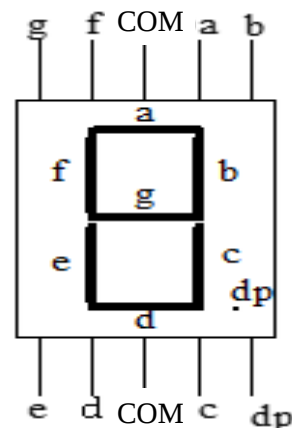
第四章 数码管实验

■ LED两种接法

➤ 共阳极接法

• 共阳极编码

- 0xc0 , 0xf9 , 0xa4 , 0xb0 , 0x99 , 0x92 ,
- 0 1 2 3 4 5
- 0x82 , 0xf8 , 0x80 , 0x90 , 0x88 , 0x83 ,
- 6 7 8 9 A B
- 0xc6 , 0xa1 , 0x86 , 0x8e , 0xff
- C D E F 无显示

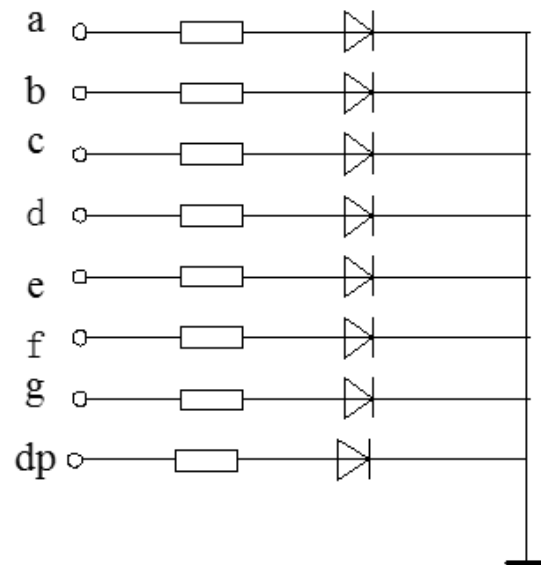
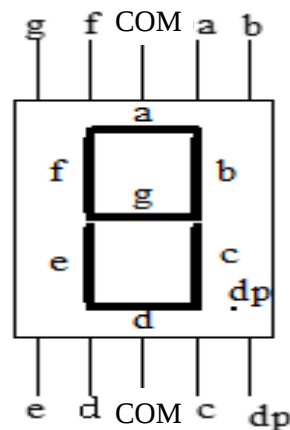


第四章 数码管实验

■ LED两种接法

➤ 共阴极接法

- 所有LED**阴极相连**形成共阴COM端
- 共阴COM端连接到**GND**
- LED的阳极接**高电平**时，相应字段亮起

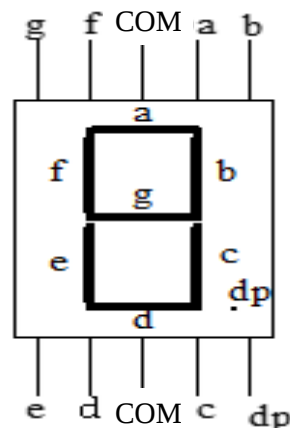


第四章 数码管实验

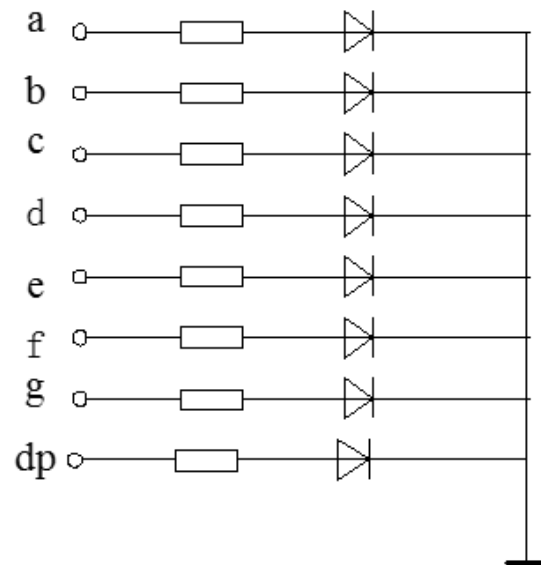
■ LED两种接法

➤ 共阴极接法

• 共阴极编码



- 0x3f , 0x06 , 0x5b , 0x4f , 0x66 , 0x6d ,
- 0 1 2 3 4 5
- 0x7d , 0x07 , 0x7f , 0x6f , 0x77 , 0x7c ,
- 6 7 8 9 A B
- 0x39 , 0x5e , 0x79 , 0x71 , 0x00
- C D E F 无显示



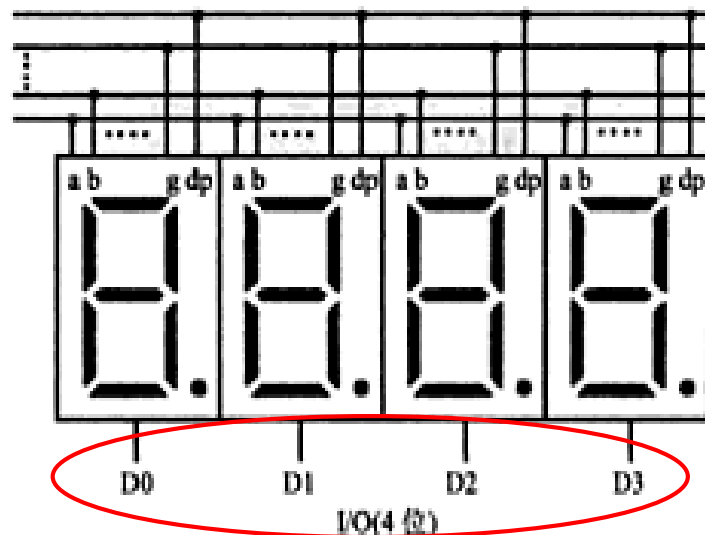


第四章 数码管实验

■ 位选与段选

➤ 位选

- 位选线数目由数码管数目决定
- 控制位选输出表示“**选**择哪个数码管”

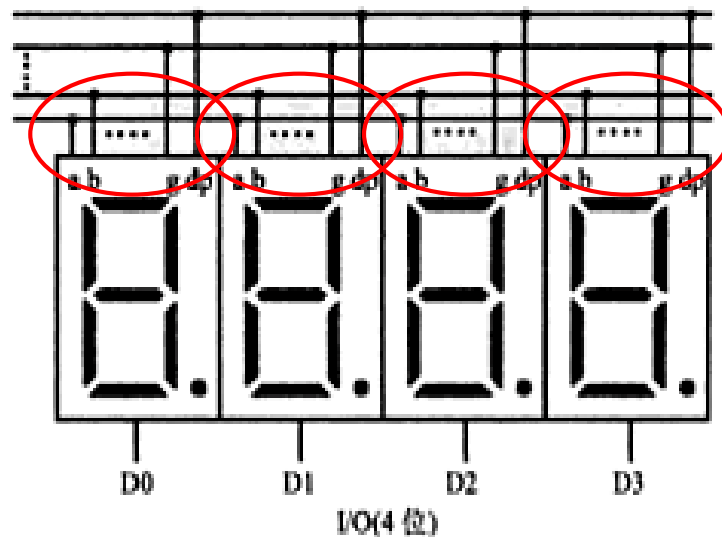


第四章 数码管实验

■ 位选与段选

➤ 段选

- 段选线数目为8，代表数码管8段LED
- 控制段选输出表示“**显示什么字型**”
- 由位选和段选可确定若千个数码管显示各种字型



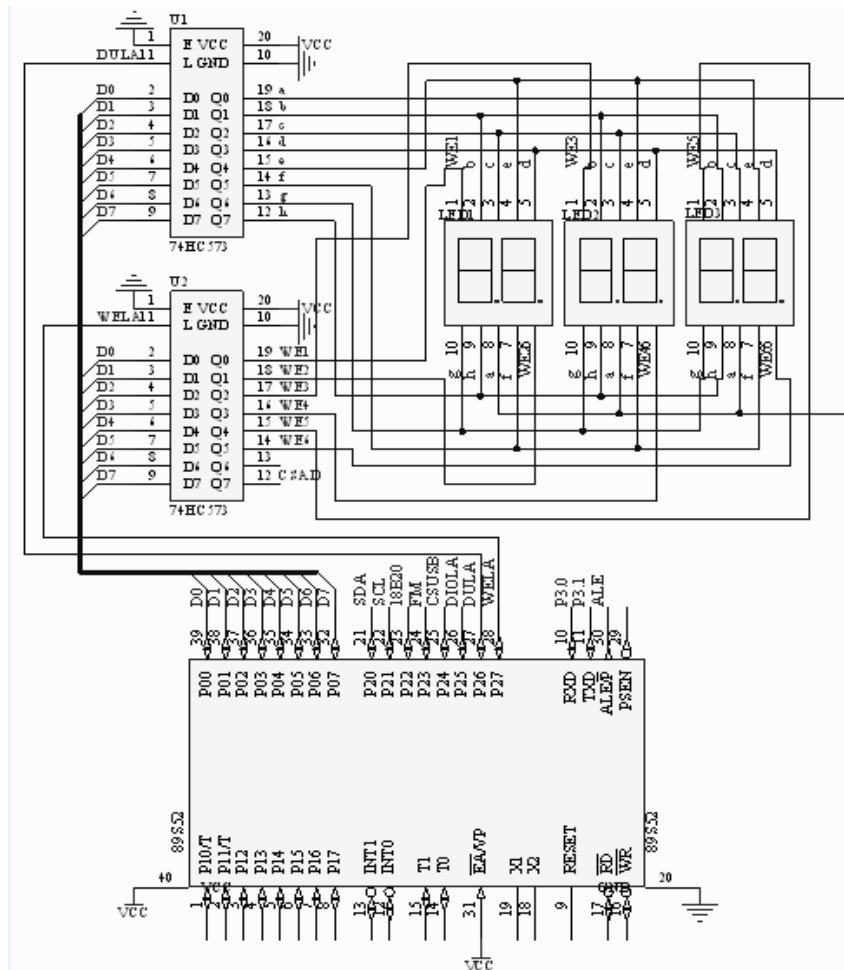


第四章 数码管实验

■ 两种显示方式

➤ 静态显示

- 通过位选线设定要点亮的数码管
- 通过段选线设定要数码管显示的字型
- 由于段选线并联，每个数码管**显示字型一致**！



第四章 数码管实验

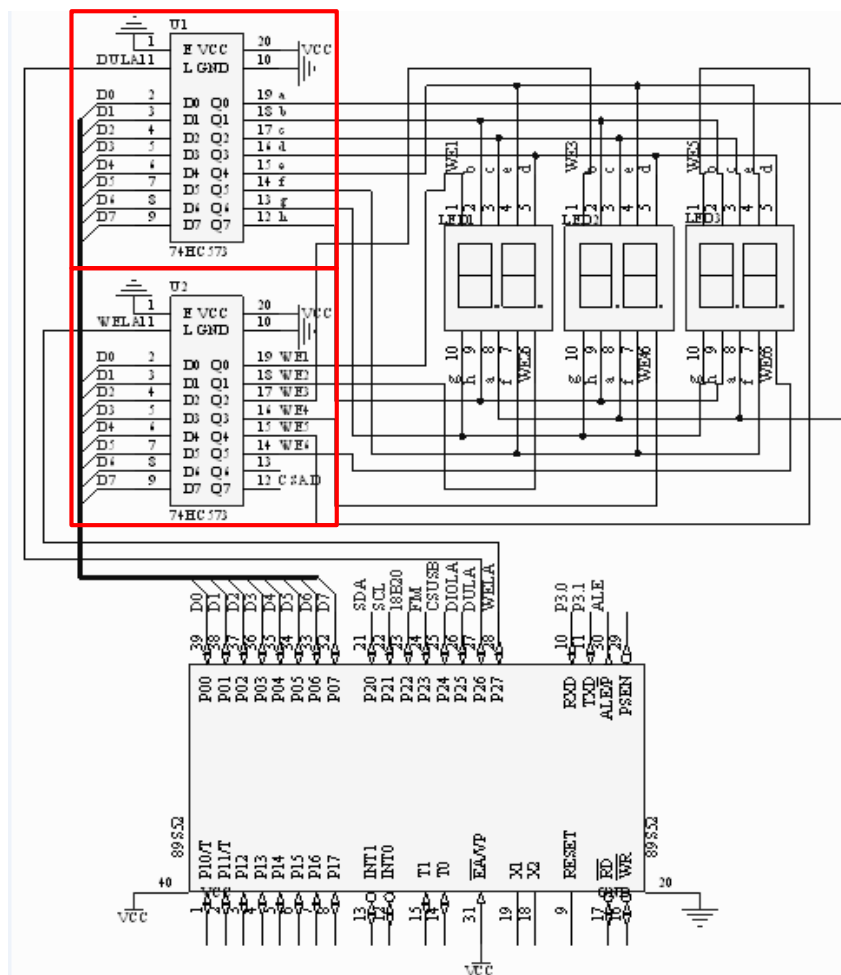
■ 两种显示方式

➤ 静态显示

- 锁存器：“保持/隔离”作用，在I/O有限情况下实现复用。

Inputs			Outputs
$\overline{OE}$	LE	D	O
L	H	H	H
L	H	L	L
L	L	X	O_0
H	X	X	Z

- 数码管用到位选锁存器和段选锁存器



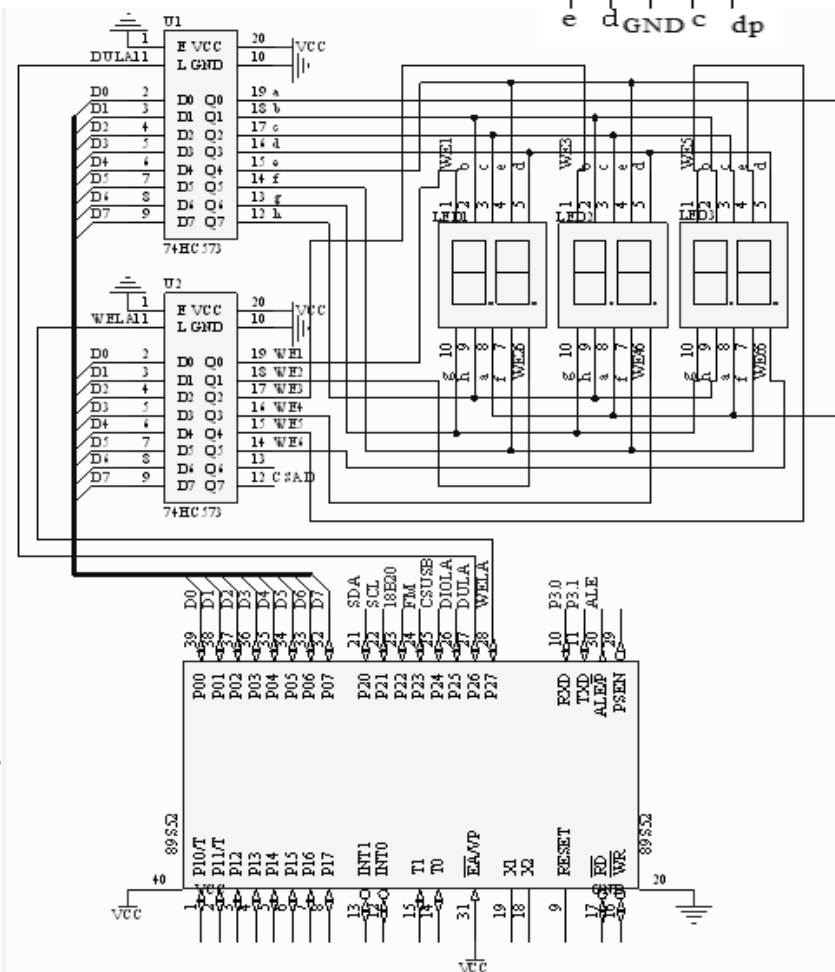
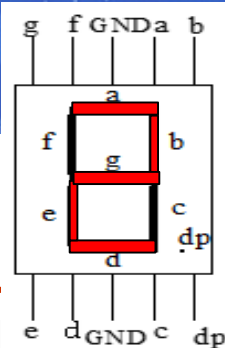


第四章 数码管实验

■ 两种显示方式

➤ 静态显示

- 举例：让第一个数码管显示数字“2”（数码管为**共阴**）
- 数字“2”的**共阴极编码**为 01011011 (0x5b)
- 位选为高时，P0输出位选信号11111110 (0xfe)。此时拉低位选，位选信号被**锁存**
- 段选为高时，P0输出段选信号01011011 (0x5b)。此时拉低段选，段选信号被**锁存**
- 完成显示



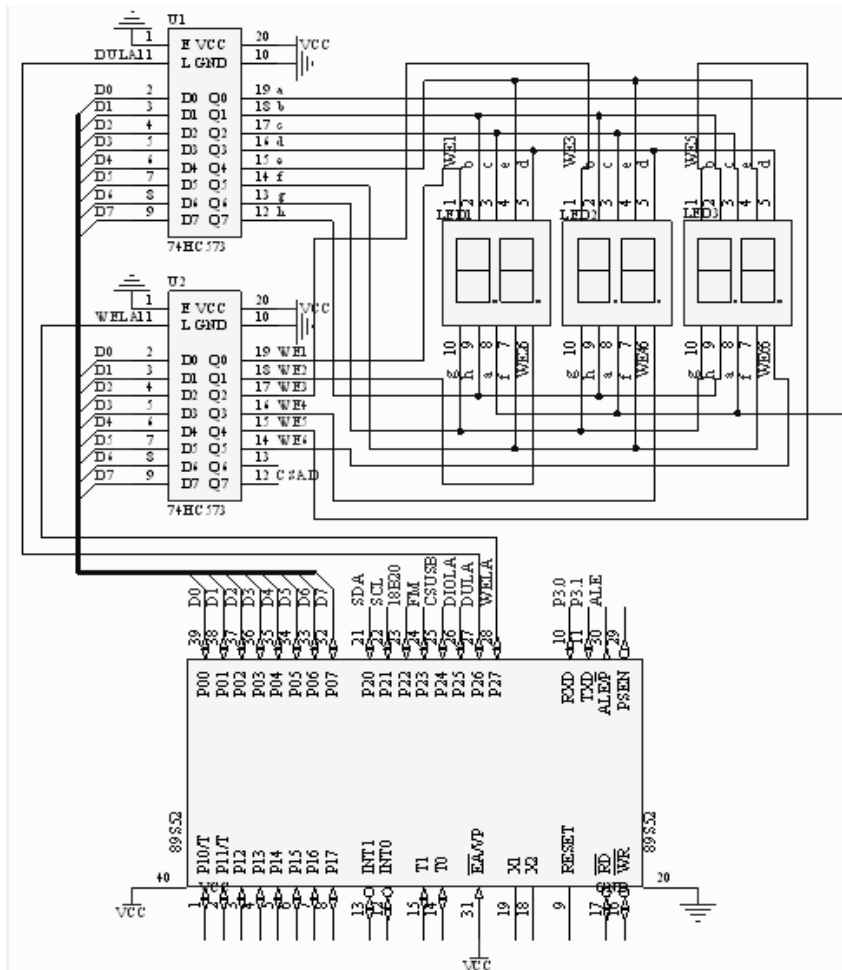


第四章 数码管实验

■ 两种显示方式

➤ 动态显示

- 静态显示缺点：所有数码管显示**相同数字**。
- 动态：每次只选通**一个**数码管，通过锁存器锁存字型
- **停留短暂时间**后选通另一个数码管，显示其它字型
- 循环以上过程，由于视觉暂留，可以看到不同数码管显示不同字型



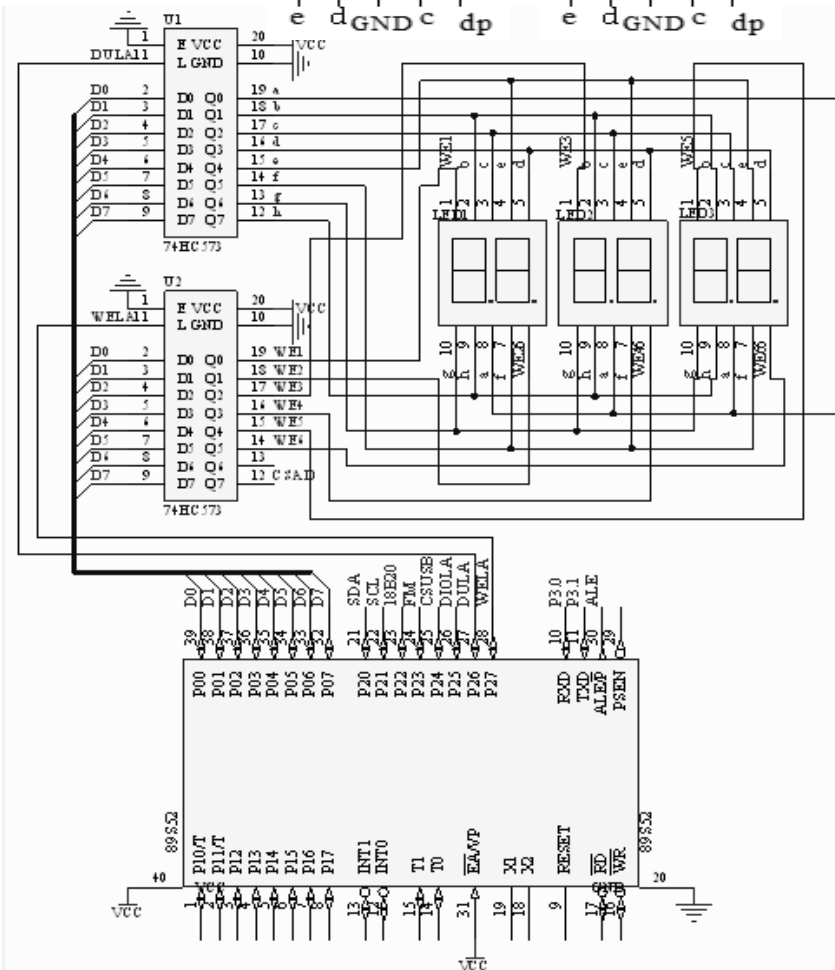
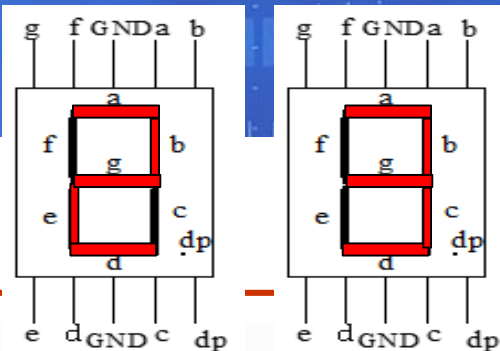


第四章 数码管实验

■ 两种显示方式

➤ 动态显示

- 举例：让前两位数码管分别显示“2”、“3”
- 只选通第一个数码管让其显示“2”，其他数码管**关闭**，**停留1ms**
- 只选通第二个数码管让其显示“3”，其他数码管**关闭**，**停留1ms**
- 循环以上过程，完成显示





第四章 数码管实验

■ 延时程序

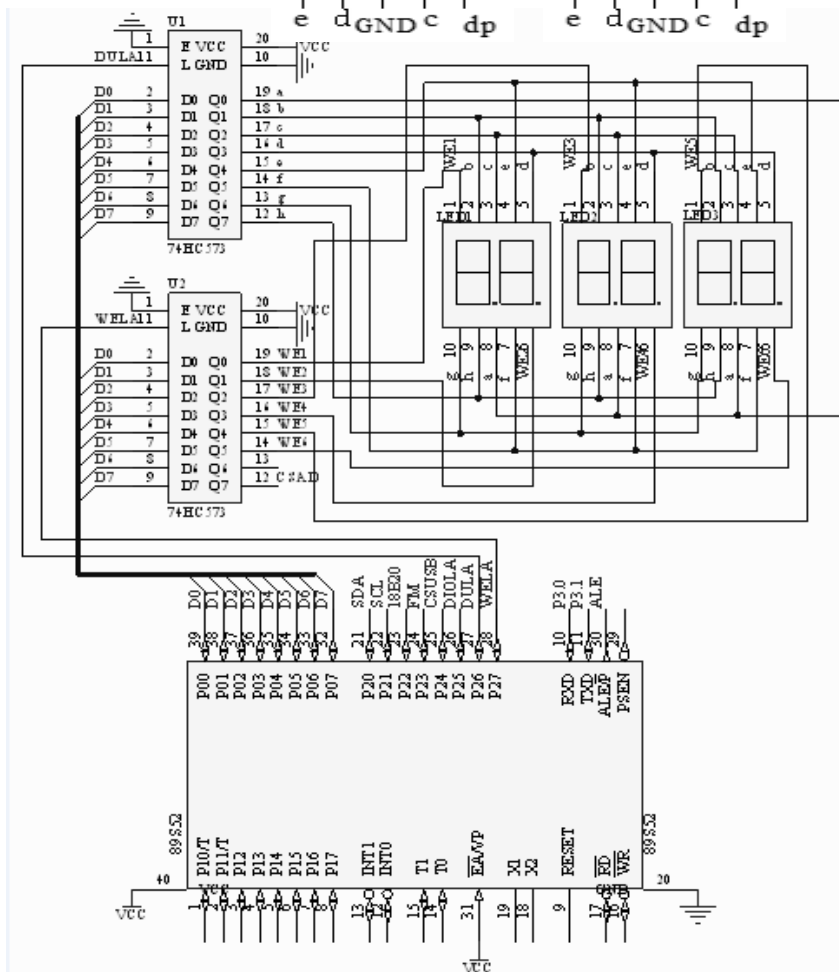
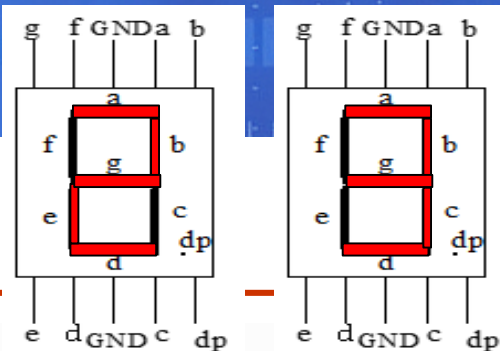
➤ 通过for循环

- 精度要求不高的场合

➤ 通过定时器计数

- 精度要求高的场合

```
void delays(unsigned int z)
{
    unsigned int i, j;
    for(i=z; i>0; i--)
        for(j=110; j>0; j--);
}
```



第四章 数码管实验

■ 消去残影

➤ 为什么有残影？

- 代码执行需要时间

```
dula=1;  
P0=0x5b;  
dula=0;  
P0=0xff;  
wela=1;  
P0=0xfe;  
wela=0;  
delayms(1);  
...
```

在打开位选前
清空P0数据

残影



残影

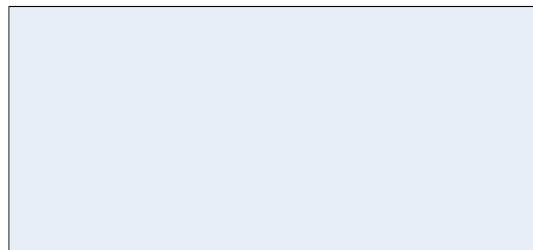
➤ 怎么消去残影？

- 在对端口赋新值时，先清空上一次的数据

第四章 数码管实验

■ I/O寄存器控制

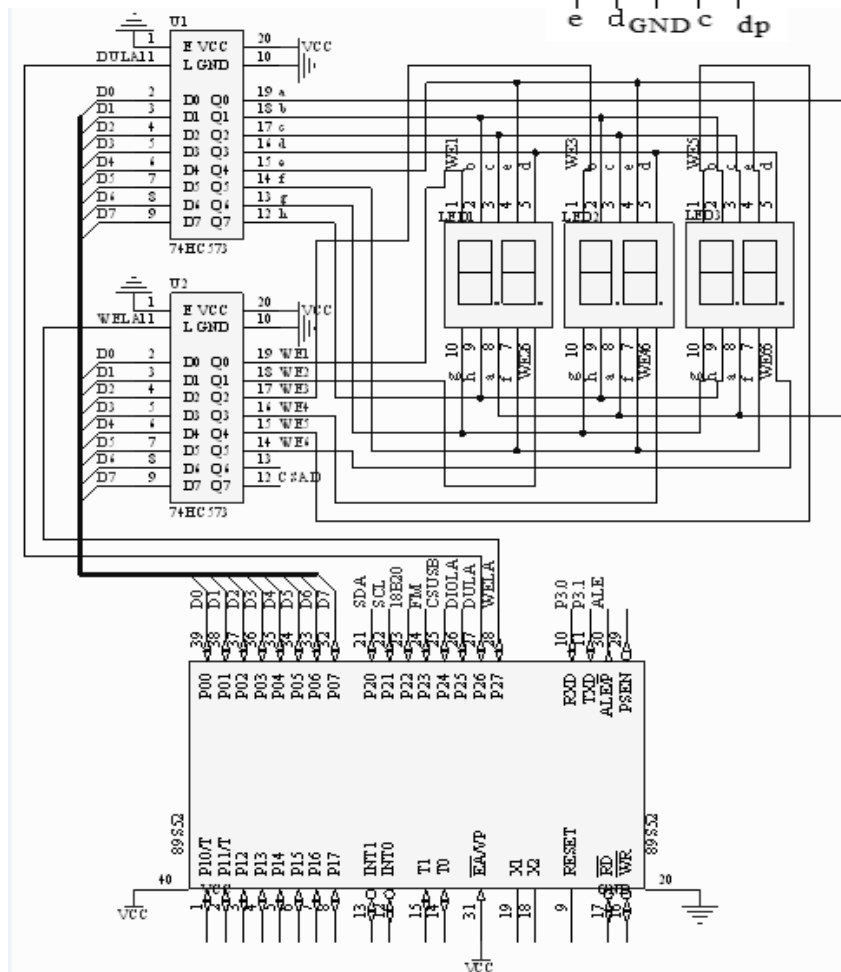
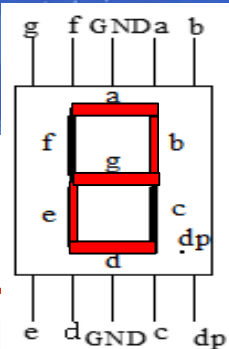
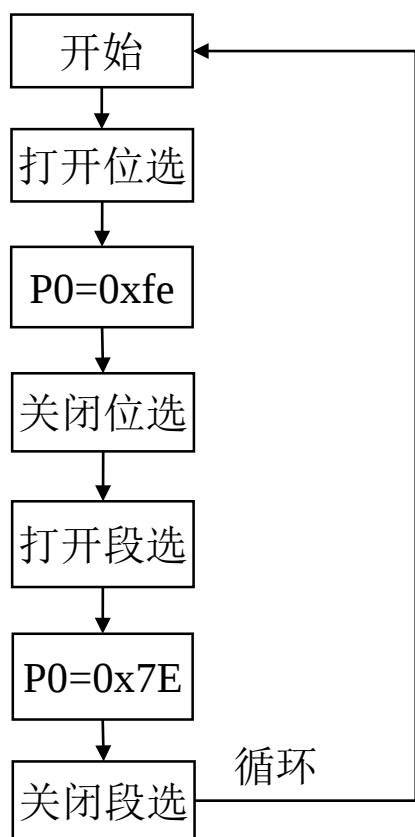
- 控制单片机I/O可**输出**高低电平，完成对数码管位选和段选的控制
- 通过**I/O寄存器**可完成对I/O口控制
- **单片机4个I/O寄存器地址**
 - P0:80H
 - P1:90H
 - P2:A0H
 - P3:B0H
- 支持**段寻址**和**位寻址**



第四章 数码管实验

流程图

静态显示





第四章 数码管实验

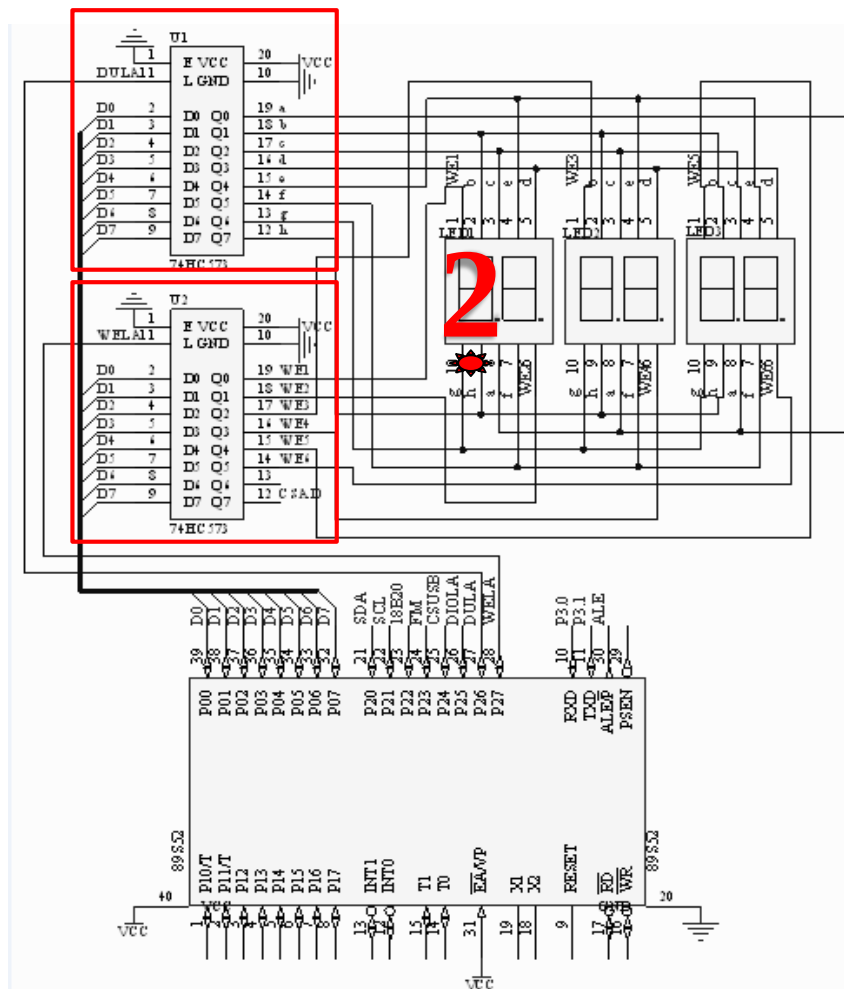
■ 程序代码

➤ 静态显示

```
#include <reg51.h>
sbit dula=P2^6; //段选锁存器地址
sbit wela=P2^7; //位选锁存器地址

void main()
{
    wela=1; //打开位选
    P0 = 0xFE; //选通第一个数码管
    wela=0; //关闭位选

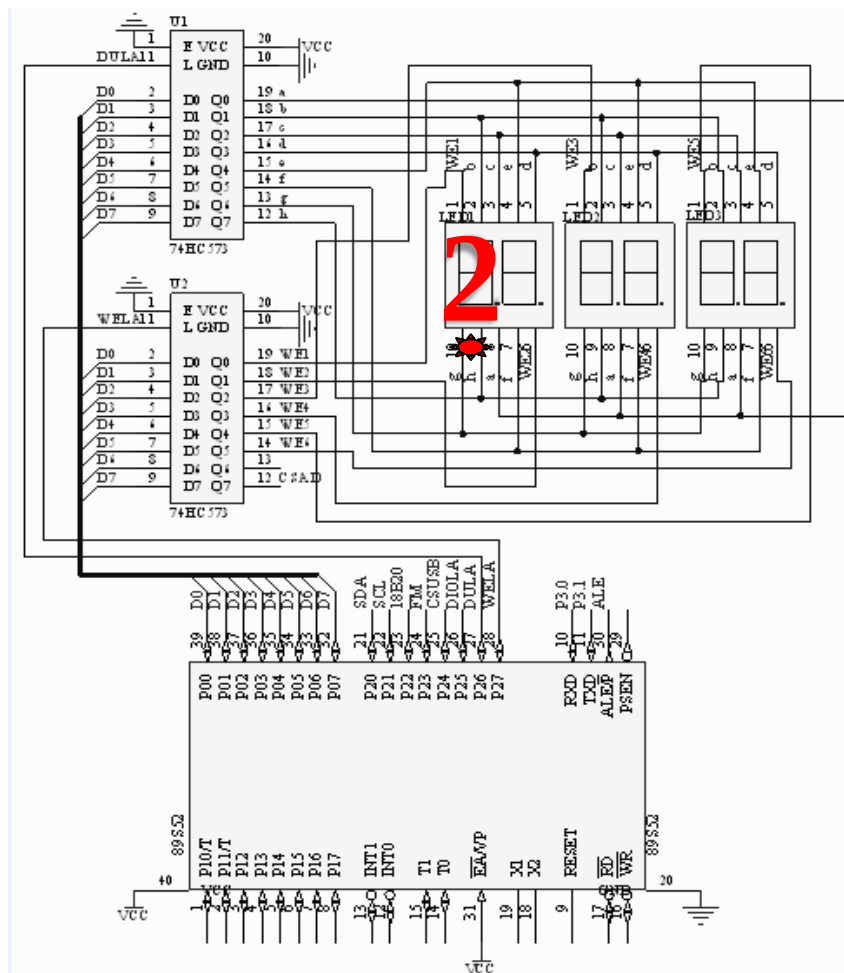
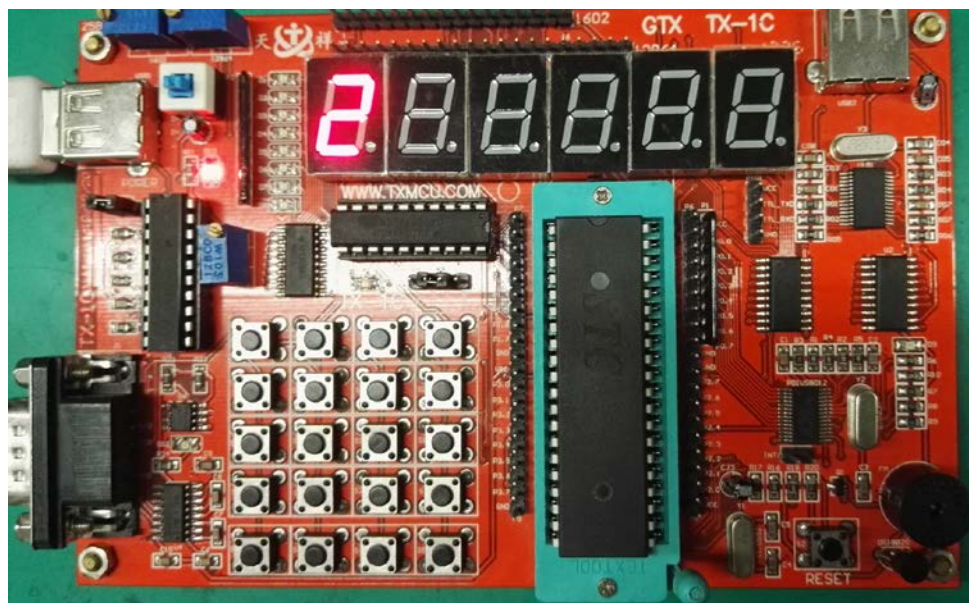
    dula=1; //打开段选
    P0=0x5b; //输出字符“2”
    dula=0; //关闭位选
    while(1); //停留
}
```



第四章 数码管实验

■ 烧写

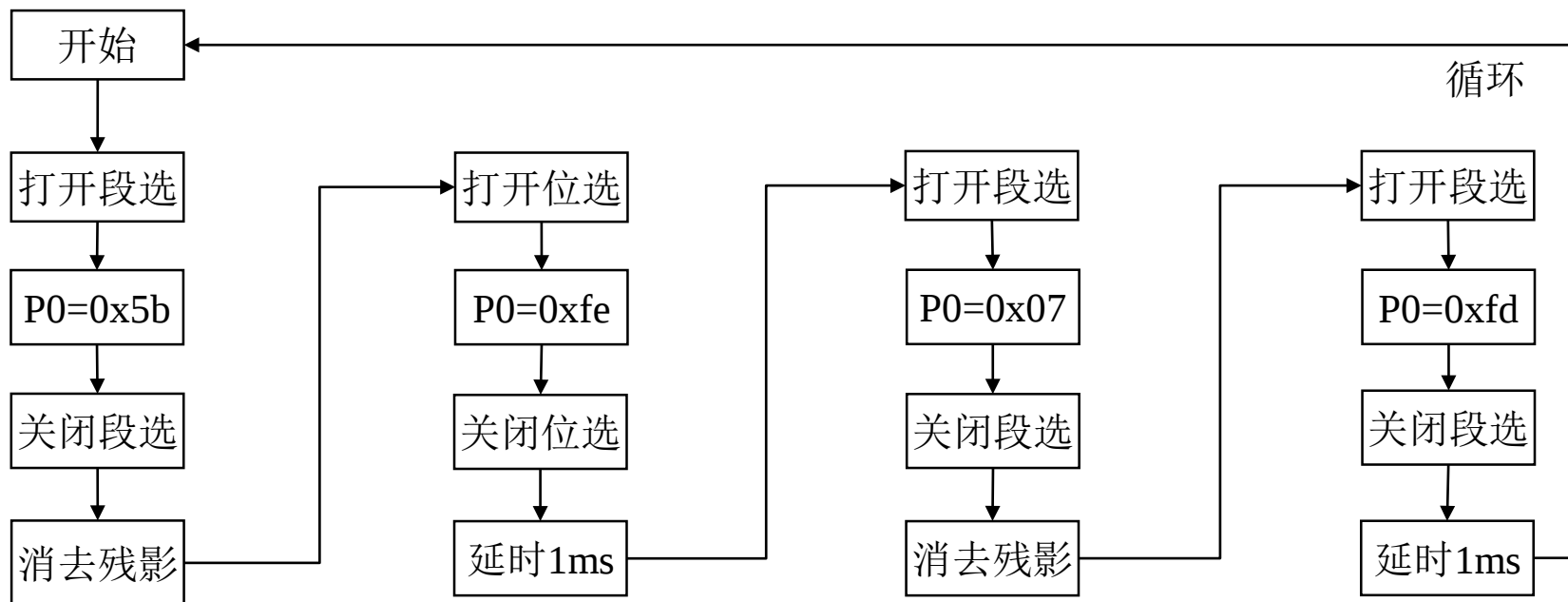
➤ 观察现象：



第四章 数码管实验

■ 流程图

➤ 动态显示

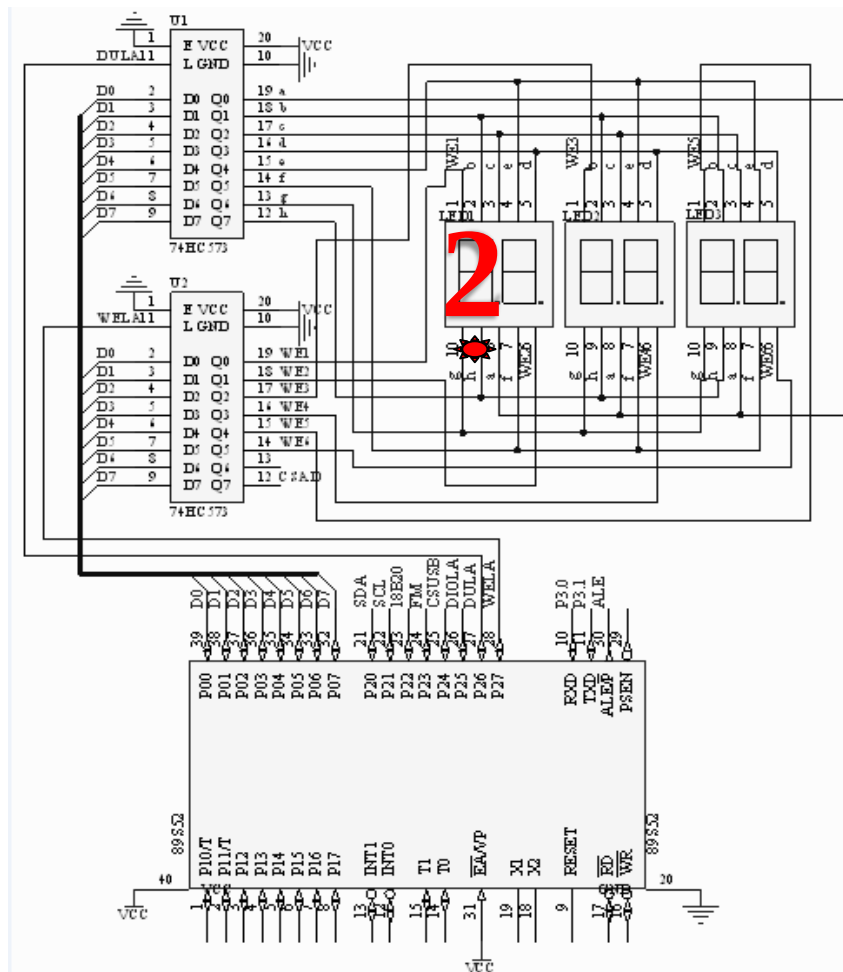




■ 程序代码

► 动态显示

```
void main()
{
    while(1) {
        dula=1;
        P0=0x5b; //显示数字 “2”
        dula=0;
        P0=0xff; //消去残影
        wela=1;
        P0=0xfe; //选通第一个数码管
        wela=0;
        delayms(1); //延时1ms
    }
}
```





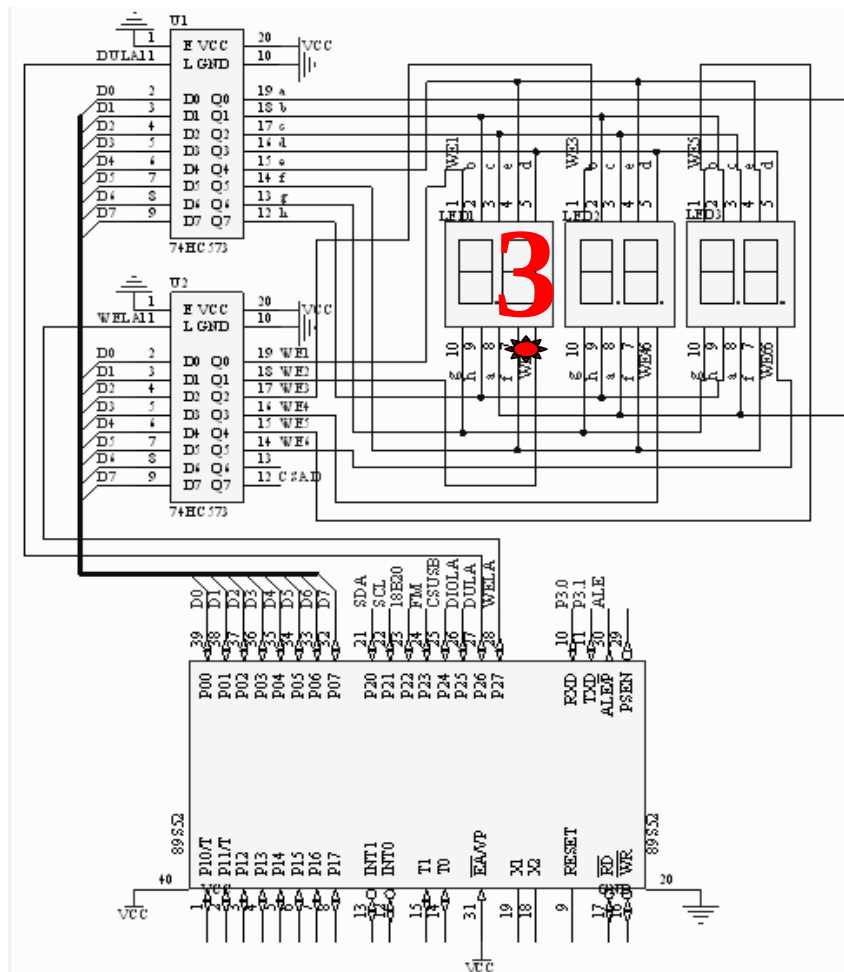
第四章 数码管实验

■ 程序代码

➤ 动态显示

```

dula=1;
P0=0x4f; //输出数字“3”
dula=0;
P0=0xff; //消去残影
wela=1;
P0=0xfd; //选通第二个数码管
wela=0;
delayms(1); //延时1ms
}
}
void delayms(int x); //延时函数
{
    int i, j;
    for(i=x; i>0; i--)
        for(j=110; j>0; j--);
}
    
```

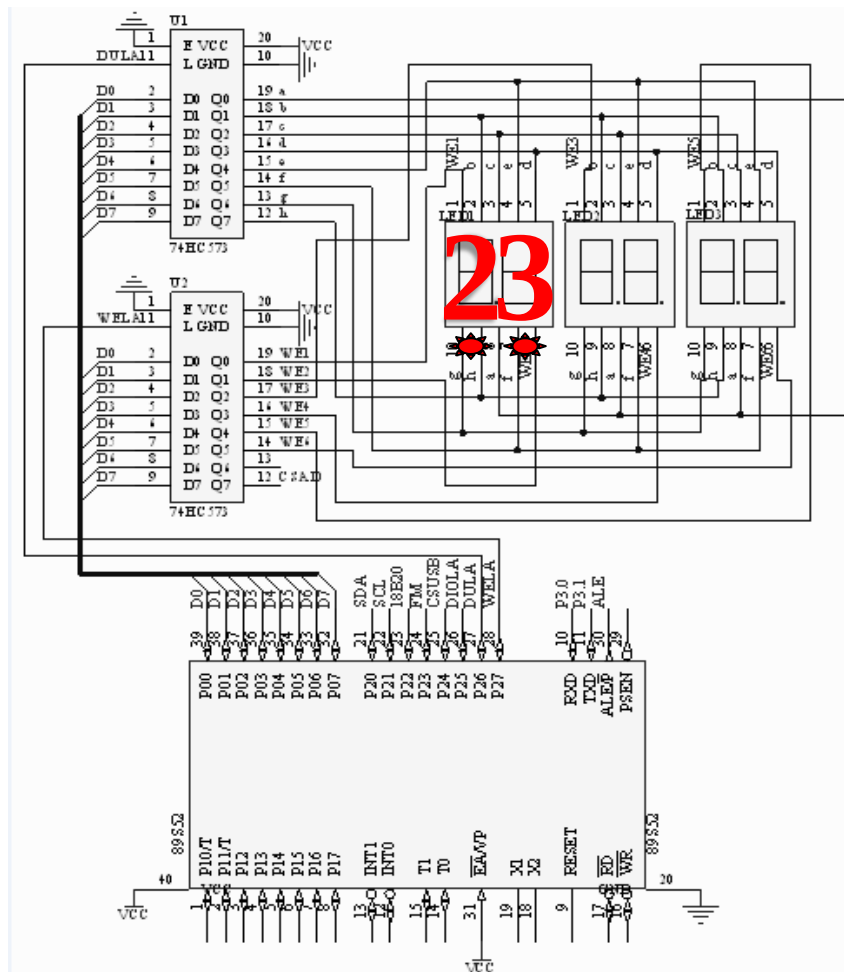
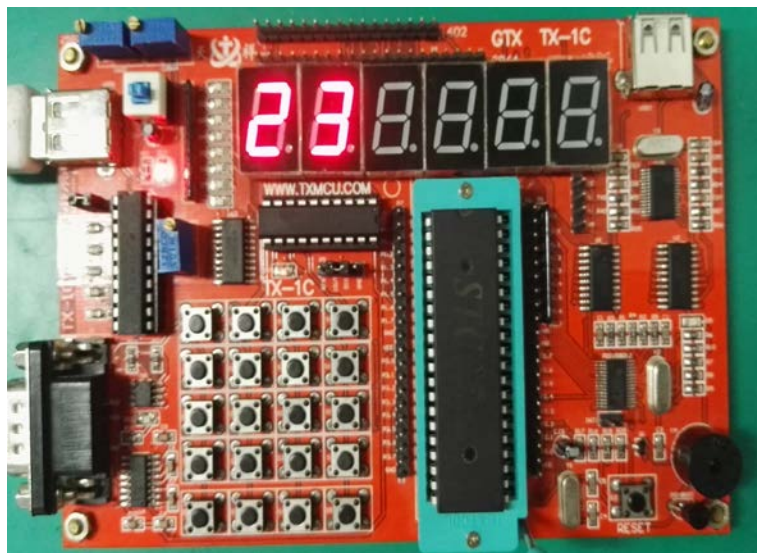




第四章 数码管实验

■ 烧写

➤ 观察现象：



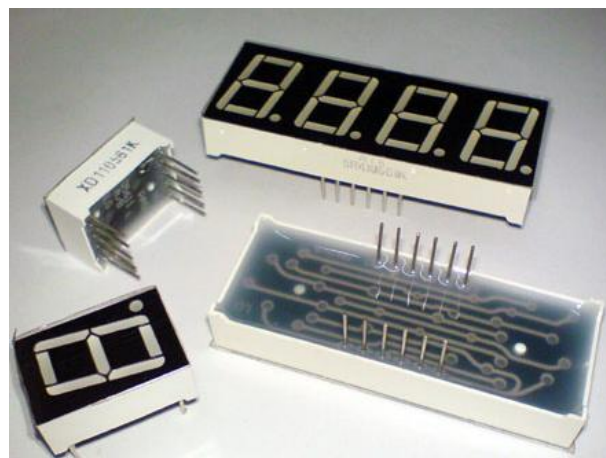
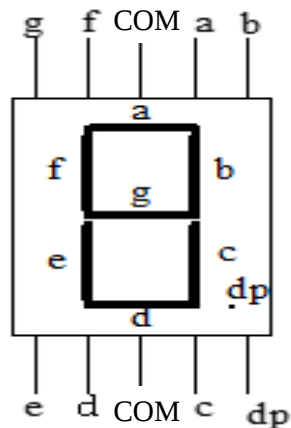
小结

■ 数码管由8个LED构成

- 共阴与共阳
- 位选与段选
- 静态显示与动态显示
- 消去残影

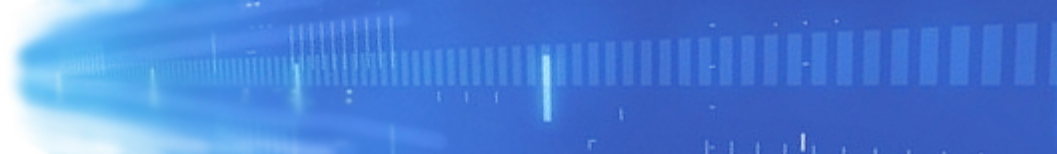
■ 锁存器

- 保持/隔离
- I/O复用





北京航空航天大学
BEIHANG UNIVERSITY



再见