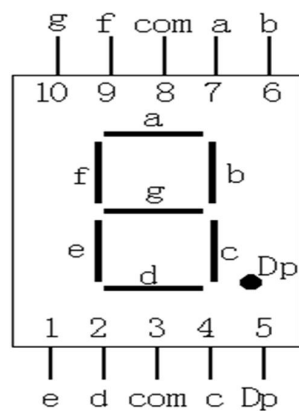


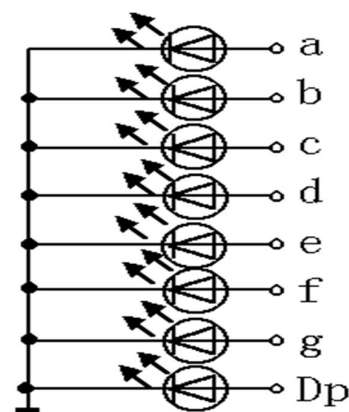
5.4 LED数码显示器

5.4.1 LED的机构与显示编码

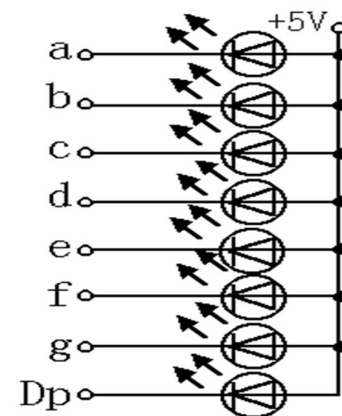
1.LED数码显示器的结构



(a) 外形和引脚



(b) 共阴极结构



(c) 共阳极结构

2. LED显示字形的编码方式

采用七段LED显示器可以显示数字0~9，字母A~F，其显示字形与对应段的点亮是有规律组合的。七段LED显示器包含七段发光二极管和小数位发光二极管，共8位，正好用一个字节来表示，通常将控制二极管点亮的八位二进制数称为段选码。

表5-6 7段LED数码管字型显示代码表

示 字	共阴型	
	Dp g f e d c b a	16进制 显示代码
0	0 0 1 1 1 1 1 1	3FH
1	0 0 0 0 0 1 1 0	06H
2	0 1 0 1 1 0 1 1	5BH
3	0 1 0 0 1 1 1 1	4FH
4	0 1 1 0 0 1 1 0	66H
5	0 1 1 0 1 1 0 1	6DH
6	0 1 1 1 1 1 0 1	7DH
7	0 0 0 0 0 1 1 1	07H
8	0 1 1 1 1 1 1 1	7FH
9	0 1 1 0 1 1 1 1	6FH

5.4.2 LED数码显示器的接口方法与显示电路

显示电路主要有静态显示和动态显示二种。

1.静态显示：当显示某一字符时，相应段的发光二极管恒定的导通或截止。

静态显示时，其公共阳极（或阴极）接VCC（或GND），一直处于显示有效状态，每一位的段选线与一个8位并行I/O口相连送出显示字符。

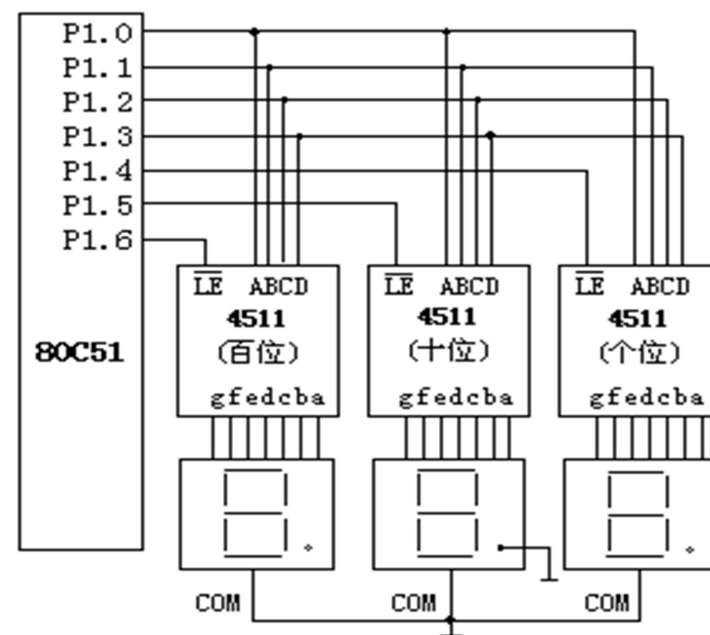


图5.17 BCD码静态显示电路

例4 按图所示的静态显示电路原理图，试编制一段显示子程序（小数点固定在第二位），已知显示数据的个位、十位和百位分别存放在内存30H～32H中。

```
SEND: MOV P1, #11100000B; 令P1.4=0, P1.5=P1.6=1, 选通个位
      ORL P1, 30H          ; 输出个位显示数
      MOV P1, #11010000B ; 令P1.5=0, P1.4=P1.6=1选通十位
      ORL P1, 31H          ; 输出十位显示数
      MOV P1, #10110000B ; 令P1.6=0, P1.5=P1.4=1选通百位
      ORL P1, 32H          ; 输出百位显示数
      RET
```

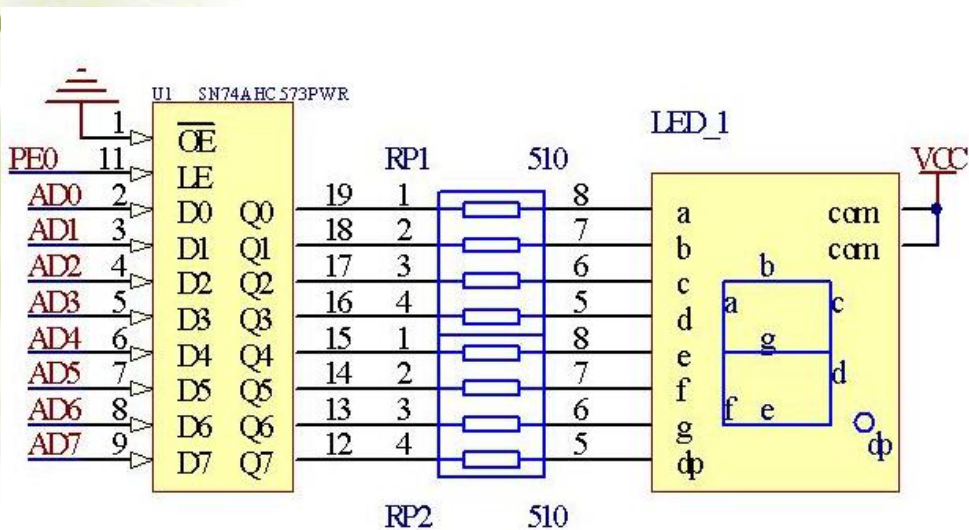
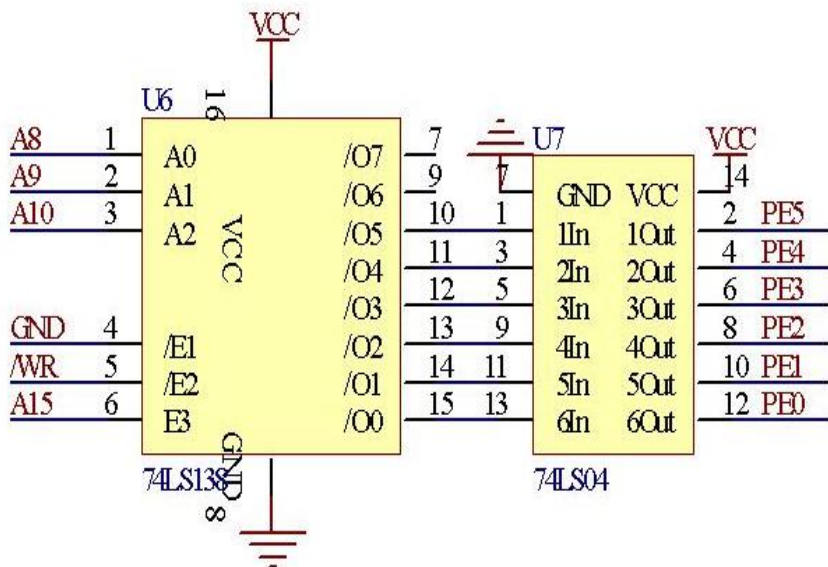
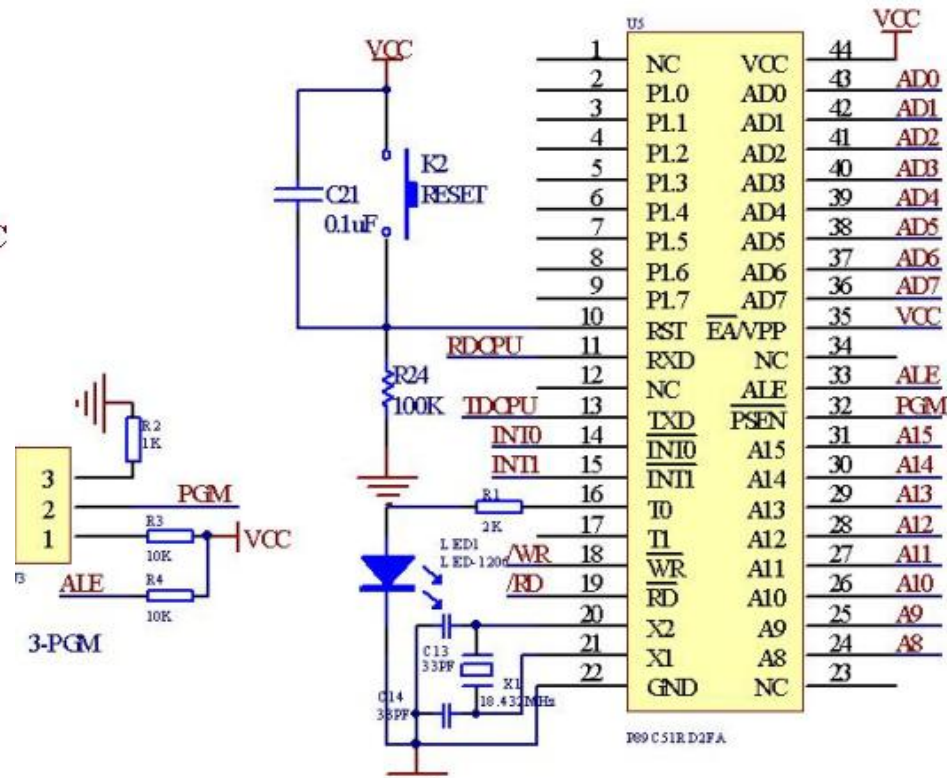



图 附1-2 锁存电路



图附1-3.锁存译码电路



图附1-1.7段数码管静态显示电路原理图之CPU电路

上图中U5为P89C51单片机，它具有64KB的FLASH和1KB的RAM,支持ISP（在线系统下编程），用于控制数码管数据显示。电路中共有4位7段共阳极数码管，用于显示数据；74HC573是8位锁存器，用于锁存各个数码管的显示数据。74HC138是3-8译码器，它和6反相器74HC04一起实现对各个数码管的锁存器控制地址的译码。

```

/*****
***
7段数码管静态显示程序
*****/
#include <reg52.h>
#include <stdio.h>
#include <stdlib.h>
#include <absacc.h>
#include <ctype.h>
#include <string.h>
#include <stddef.h>
#include <intrins.h>
#include <absacc.h>
#include <stdarg.h>

/* 4个7段数码管锁存器地址 */
#define LED1ADDR 0x8000
#define LED2ADDR 0x8100
#define LED3ADDR 0x8200
#define LED4ADDR 0x8300
// 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f
/* 0-9 a-f 的7段数码管显示编码 */
unsigned char
    ledcode[16]={0xC0,0xF3,0x89,0xA1,
0xB2,0xA4,0x84,0xF1,0x80,0xA0,0x90,0x
86,0xCC,0x83,0x8C,0x9C};

```

```

/* 存放要显示的数字值 */
unsigned char val[4];//
/* 写锁存器的子程序 */
void wraddr(unsigned int addr,unsigned char d)
{unsigned char xdata *pa;

pa=(unsigned char xdata *)addr;
*pa=d;
}
/* 显示子程序 */
void display(void)
{unsigned int addr[4]={LED1ADDR,LED2ADDR,
LED3ADDR,LED4ADDR};

int i;
for(i=0;i<4;i++)
{
    if((val[i]>=0)&&(val[i]<=15))
    {
        wraddr(addr[i],ledcode[val[i]]);
    }
    else
    {
        wraddr(addr[i],0xff);
    }
}
}

```

```
/* 初始化子程序 */
void init(void)
{
    wraddr(LED1ADDR,0XFF);
    wraddr(LED2ADDR,0XFF);
    wraddr(LED3ADDR,0XFF);
    wraddr(LED4ADDR,0XFF);
    //输入要显示的数值
    val[0]=1;
    val[1]=2;
    val[2]=3;
    val[3]=4;
}
/* 主程序入口 */
int main(void)
{
    init();

    while(1)
    {
        display();
    }
}
```

2. 动态显示：一位一位地轮流点亮各个显示器。

LED显示器的动态显示是将所有显示位的段选线的同名端并连在一起，并由一个**8位I/O口**控制，形成段选线的多位复用。而各显示位的公共阳极端或公共阴端极则分别由相应的**I/O口线**控制，实现各位的分时选通，由于人眼有视觉暂留现象，只要每位显示间隔足够短（**1~2ms**），则会造成多位同时点亮的假象。

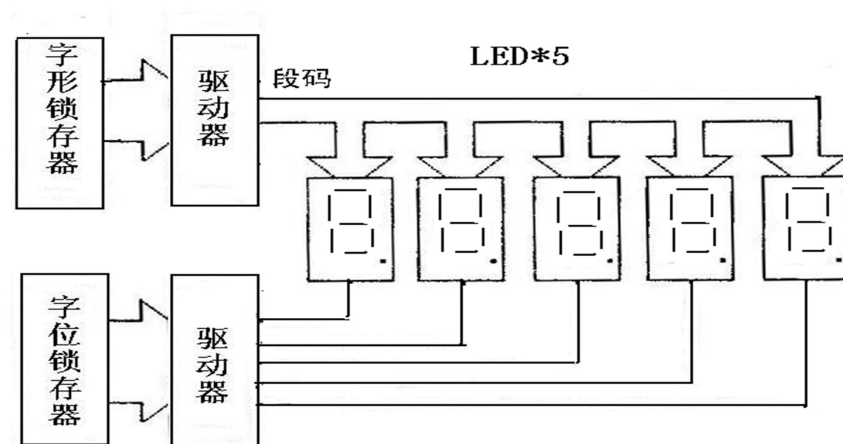


图 动态显示电路结构图

3.动态显示电路举例应用

例5 按图5.19，试编制循环扫描（10次）显示子程序，已知被显示的字段码已经存放在以40H（低位）为首址的连续8字节内部RAM中。

解题思路：先确定74LS377的端口地址，由于是P2.7接74LS377的片选端，所以只需P2.7=0即可使得74LS377芯片有效；从电路图中可知74LS138的使能端已经有效，所以不必在代码中设置其端口地址了。然后从低位开始依次发送显示码显示，每发送一个显示码就必须调用依次延时2ms的子程序，执行动态刷新后，再发下一位的显示代码，如此重复直至发完所有数据。

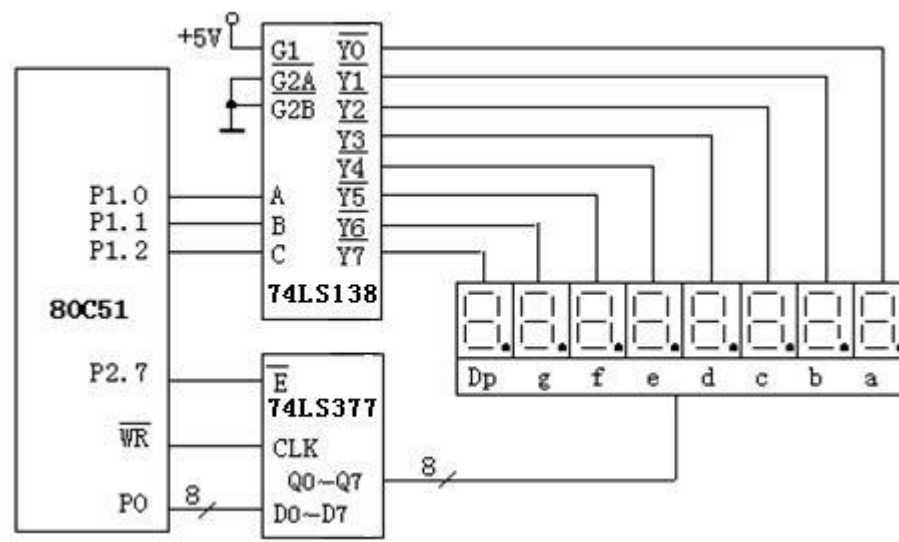
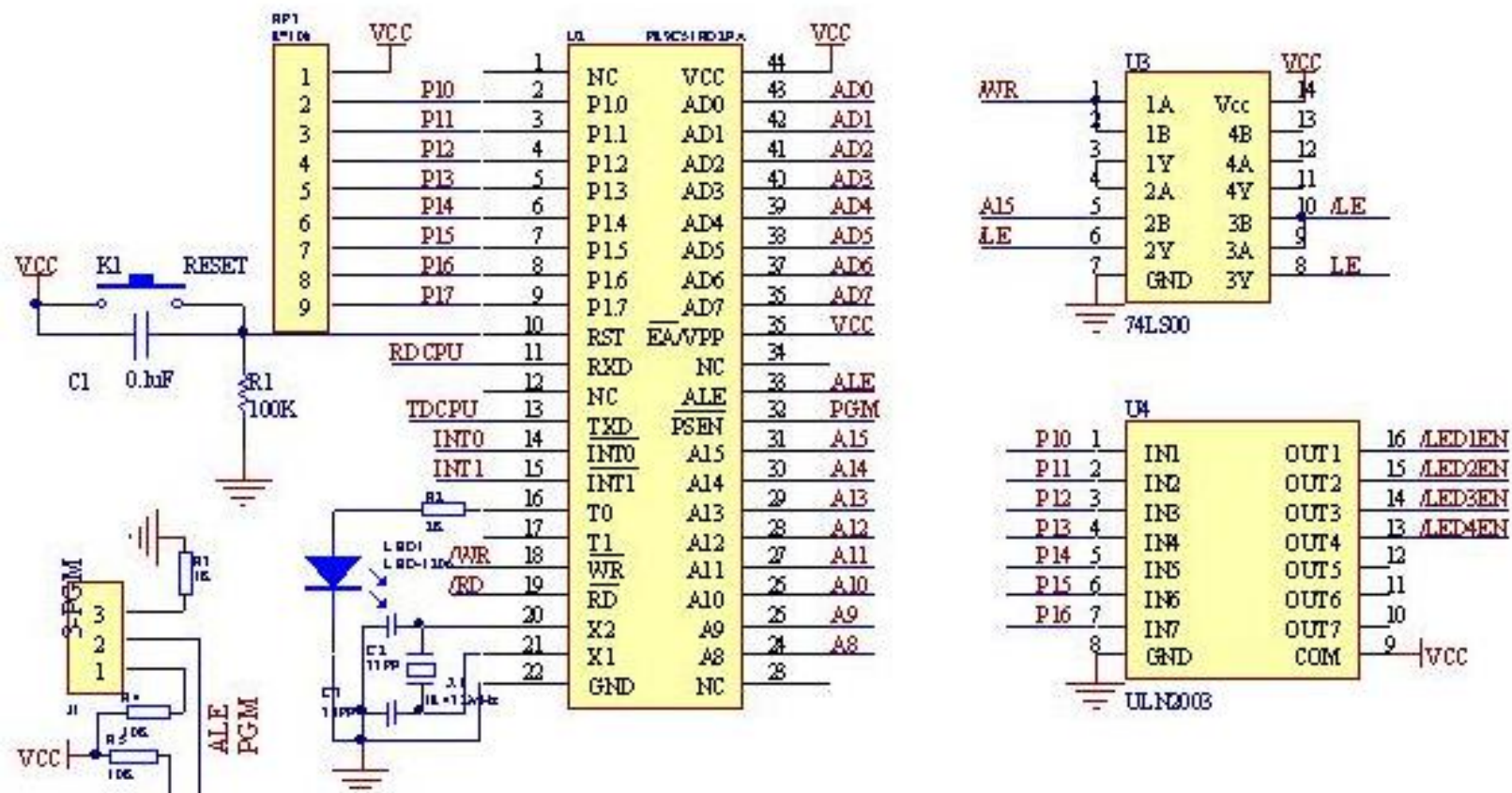


图5.19 共阴型动态显示电路图

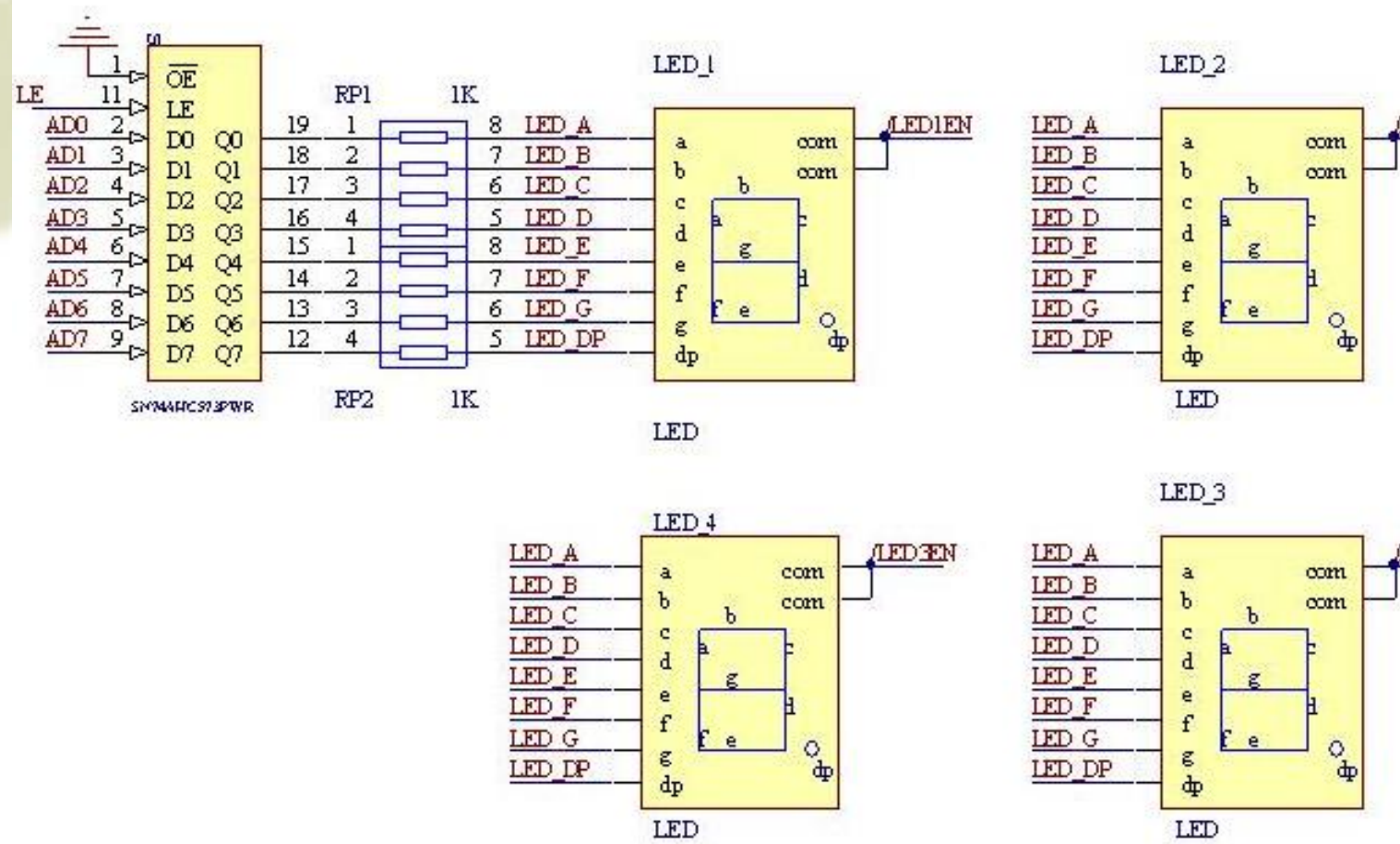
编程如下:

```
DIR4: MOV    R1, #10                ; 置循环扫描次数
      MOV    DPTR, #7FFFH           ; 置74LS377口地址
DLP1: ANL     P1, #11111000B; 令P1.0=P1.1=P1.2=0, 第0位先显示
      MOV    R0, #40H               ; 置R0为显示字段码首址
DLP2: MOV     A, @R0                 ; 读显示字段码
      MOVX   @DPTR, A               ; 输出显示字段码
      LCALL  Delay_2ms              ; 调用延时2ms子程序
      INC    R0                     ; 指向下一位字段码
      INC    P1                     ; 选通下一位显示
      CJNE   R0, #48H, DLP2         ; 判8位扫描显示完否?未完继续
      DJNZ   R1, DLP1               ; 8位扫描显示完毕, 判10次循环完否?
      CLR    A                       ; 10次循环完毕, 显示暗
      MOVX   @DPTR, A
      RET                             ; 子程序返回
```

下图中U3为P89C51单片机，它具有64KB的FLASH和1KB的RAM,支持ISP（在线系统下编程），用于控制数码管数据显示。电路中共有4位7段共阳极数码管，用于显示数据；74HC573是8位锁存器，它和4与非门反相器74HC00共同用于锁存各个数码管的显示数据，ULN2003为大电流驱动芯片，是为了驱动LED的。



例附2 -2 7段数码管动态显示电路原理图之CPU和锁存电路



例 附2 -1 7段数码管动态显示电路原理图之数码管驱动电路


```
/*  
*****
```

7段数码管动态显示程序

```
*****  
*****/
```

```
#include <reg52.h>           // 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f  
#include <stdio.h>          /* 0-9 a-f 的7段数码管显示编码 */  
#include <stdlib.h>          unsigned char ledcode[16]={0x3F,0x0C,0x76,0x5E,  
#include <absacc.h>          0x4D,0x5B,0x7B,0x0E,0x7F,0x5F,0x6F,0x79,0x33,0x7C,  
#include <ctype.h>           0x73,0x63};  
#include <string.h>          /* 存放要显示的数值值 */  
#include <stddef.h>          unsigned char val[4];//  
#include <intrins.h>         unsigned curried;  
#include <absacc.h>            
#include <stdarg.h>          /* 写锁寄存器的子程序 */  
                                void wraddr(unsigned int addr,unsigned char d)  
                                {          unsigned char xdata *pa;  
                                  
                                pa=(unsigned char xdata *)addr;  
                                *pa=d;  
                                }  
  
#define TIMER0H 0xc4;  
#define TIMER0L 0x00;  
/* 7段数码管锁存器地址 */  
#define LEDADDR 0x8000 }
```

```

/* 显示子程序 */
void display(void)
{
    unsigned char en[4]={1,2,4,8};

    if((val[currled]>=0)&&(val[currled]<=15))
    {
        wraddr(LEDADDR,ledcode[val[currled]]);
        P1=en[currled];
    }
    else
    {
        wraddr(LEDADDR,0x00);
        P1=en[currled];
    }
    if((currled>=0)&&(currled<3))
    {
        currled++;
    }
    else
    {
        currled=0;
    }
}

void time0(void) interrupt 1
{
    display();
    TL0 = TIMER0L;
    TH0 = TIMER0H;
}

```

```

/* 初始化子程序 */
void init(void)
{
    //timer0 init
    TL0 = TIMER0L;
    TH0 = TIMER0H;
    TR0=1;
    ET0=1;//test

    // irq init
    EX0=0;
    IT0=0;
    EX1=0;
    IT1=0;
    EA=1;
    //输入要显示的数值
    val[0]=1;
    val[1]=2;
    val[2]=3;
    val[3]=4;
    currled=0;
}

/* 主程序入口 */
int main(void)
{
    init();
    while(1)
    {
    }
}

```