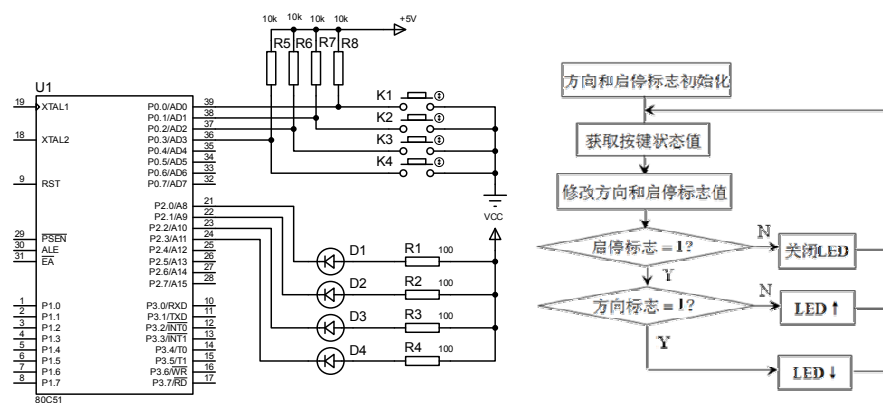
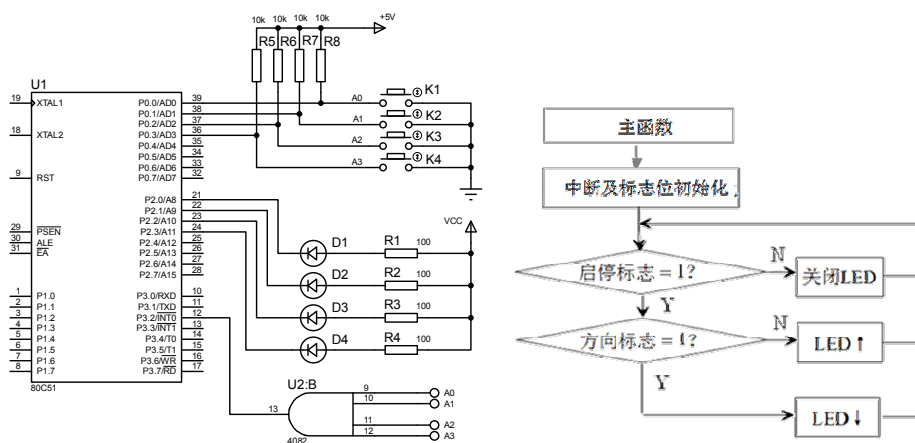


实例 3 将第 4 章改用中断方式实现



问题: 当 CPU 运行于 LED 循环环节时, 将因不能及时检测按键状态, 而使按键操作不灵敏。

硬件改动: 增加一只 4 输入与门电路, 使按键闭合电平作为外部中断信号。



软件改动: 将标志位修改功能放在中断函数中进行。

```

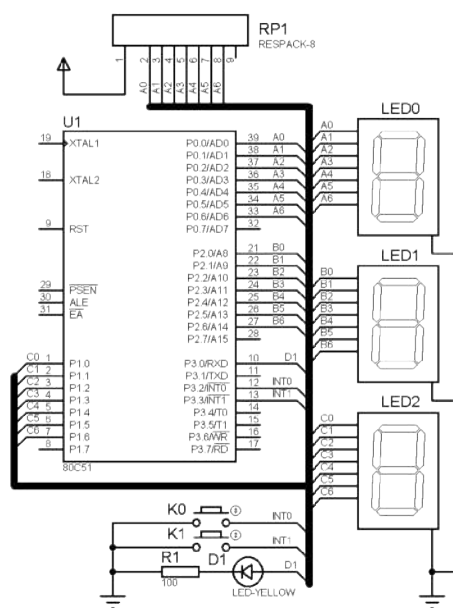
01 #include "reg51.h"
02 char led[]={0xfe,0xfd,0xfb,0xf7}; //LED花样数据
03 bit dir=0,run=0; //全局变量
04 void delay(unsigned int time);
05 key() interrupt 0 { //键控中断函数
06     switch (P0 & 0x0f) { //修改标志位状态
07         case 0x0e:run=1;break;
08         case 0x0d:run=0,dir=0;break;
09         case 0x0b:dir=1;break;
10         case 0x07:dir=0;break;
11     }
12 }
13 void main() {
14     char i;
15     IT0=1;EX0=1;EA=1; //边沿触发、INT0允许、总中断允许
16     while(1) {
17         if (run)
18             if (dir) //若run=dir=1,自上而下流动
19                 for(i=0;i<3;i++){
20                     P2=led[i];
21                     delay(200);
22                 }
23             else //若run=1,dir=0,自下而上流动
24                 for(i=3;i>0;i--){
25                     P2=led[i];
26                     delay(200);
27                 }
28             else P2=0xff; //若run=0,灯全灭
29     }
30 }
31 void delay(unsigned int time){
32     unsigned int j = 0;
33     for(;time>0;time--){
34         for(j=0;j<125;j++);
35     }
36 }

```

实例 4 中断嵌套演示

功能:

- LED0 代表主函数的功能；
- K0 产生 INTO 中断请求，调用 LED1 显示功能；
- K1 产生 INT1 中断请求，调用 LED2 显示功能；
- 设置 INT1 为高级中断，INT0 为低级中断；
- 高级中断运行时，若有低级中断请求，则 D1 点亮；高级中断结束后，低级中断才能运行。



```

01 #include "reg51.h"
02 char led_mod[] = {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07,
03                  0x7f, 0x6f}; //显示字模
04 sbit D1=P3^0;
05 void delay(unsigned int time){//延时
06     unsigned char j;
07     for(j=255;time>0;time--){
08         for(;j>0;j--);
09     }
10 key0() interrupt 0 {                //K0中断函数
11     unsigned char i;
12     D1=IE0;                          //IE0状态输出
13     for(i=0;i<=9;i++){              //字符0-9循环1圈
14         P2=led_mod[i];
15         delay(35000);
16     } P2=0x40;                       //结束符“-”
17 }
18
19 key1() interrupt 2 {                //K1中断函数
20     unsigned char i;
21     for(i=0;i<=9;i++){              //字符0-9循环1圈
22         D1=IE0;                      //IE0状态输出
23         P1=led_mod[i];
24         delay(35000);
25     } P1=0x40;                       //结束符“-”
26 }
27
28 void main(){
29     unsigned char i;
30     TCON = 0x05;                     //脉冲触发方式
31     PX0=0;PX1=1;                     //INT1优先
32     D1=0;P1=P2=0x40;                 //输出初值
33     IE=0x85;                         //开中断
34     while(1){
35         for(i=0;i<=9;i++){           //字符0-9无限循环
36             P0=led_mod[i];
37             delay(35000);
38         }

```