第六章 单片机输入输出接口及系统扩展设计

预备知识

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

单片机原理及系统设计

MCS-51的I/O接口形式主要分两类：

- 通过并行端口（P0~P3）直接完成输入输出
  - 单片机并行接口内部结构可参阅第二章；
  - 单片机通过执行指令MOV A,Px完成端口输入；
  - 单片机通过执行指令MOV Px,A完成端口输出；
  - 单片机还可以通过位操作指令完成Px端口中针对某位的输入输出；
- 通过系统总线扩展完成输入输出
  - 单片机总线扩展操作时序可参阅第二章；
  - 单片机通过执行指令MOVX A,@DPTR完成总线输入；
  - 单片机通过执行指令MOVX @DPTR,A完成总线输出。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

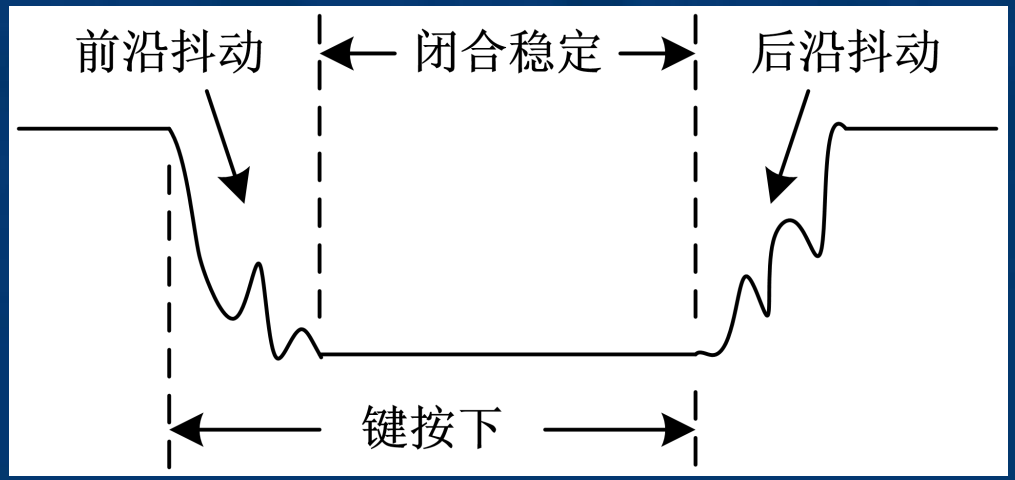
第六章 单片机输入输出接口及系统扩展设计

6.1 键盘及其接口设计

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.1 键盘的基本工作原理

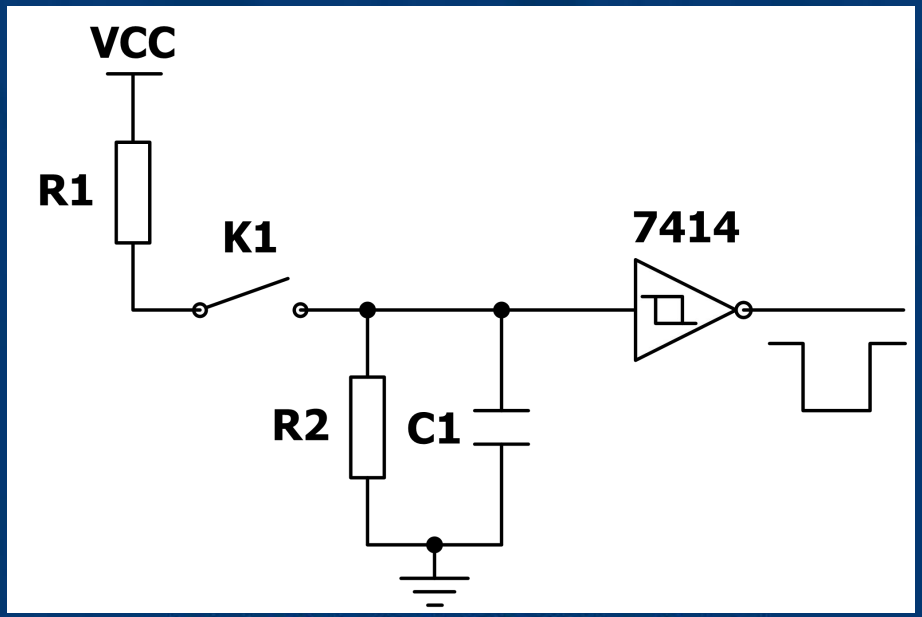
- 按键一般通过机械触点实现通断
- 单片机通过I/O端口输入触点状态判断按键的状态
- 按键按下及弹起时会有抖动现象



南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

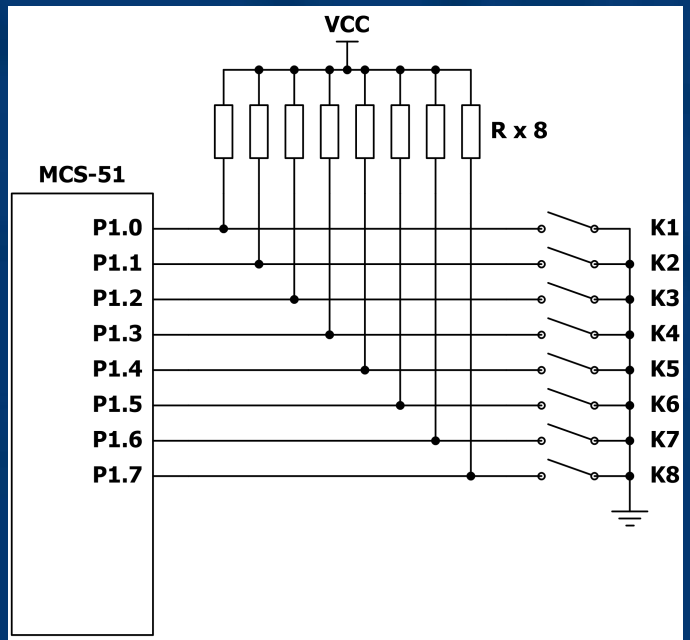
6.1.1 键盘的基本工作原理

- 对抖动现象的处理方法：
  - 使用如图所示的消抖电路（成本高，体积大）；
  - 使用软件消抖（电路简化，软件复杂度提高）；
  - 使用专用集成电路。



6.1.2 独立式键盘接口设计

- 独立式键盘中每个按键都单独连接到单片机的一个端口引脚上，由程序分别处理；
- 多个独立式按键组合在一起就构成了独立式键盘：

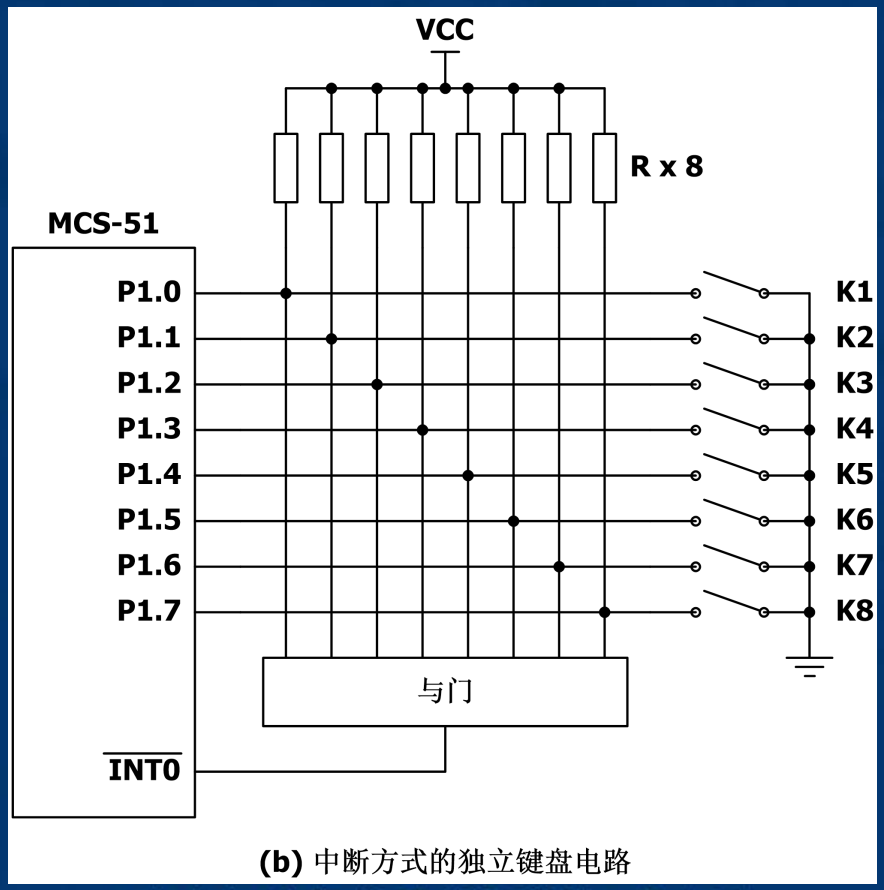


(a) 扫描方式的独立键盘电路

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.2 独立式键盘接口设计

- 也可如下图所示，采用中断的方式响应按键事件



(b) 中断方式的独立键盘电路

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.2 独立式键盘接口设计

- 独立式键盘程序设计

```
#include <reg51.h>
/* 定义延时10ms函数 */
void Delay10ms(void)
{......}

/* 定义各按键的处理函数 */
void ProcessK1(void)
{......}
void ProcessK2(void)
{......}
......

/* 主程序 */
main()
{
    unsigned char Key;

    P1 = 0xff;                          /* 置P1为输入方式 */
    while(1)
    {
        Key = P1;                       /* 读入按键状态 */
        if(P1 != 0xff)                  /* 如果有键按下 */
        {
            Delay10ms();                /* 延时10ms消抖动 */
            Key = P1;                   /* 再次读入按键状态 */
        }
        else continue;                  /* 无键按下，返回继续查询 */
```


南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.2 独立式键盘接口设计

- 独立式键盘程序设计

```
        while(P1 != 0xff);                  /* 等待按键释放后再进行处理 */
        switch(Key)                         /* 根据按键状态进行分支处理 */
        {
            case 0xfe:                      /* P1.0为低，K1按下 */
                ProcessK1();
                break;
            case 0xfd:                      /* P1.1为低，K2按下 */
                ProcessK2();
                break;
            ......
            case 0x7f                       /* P1.7为低，K8按下 */
                ProcessK8();
                break;
            default:
                continue;                   /* 继续扫描 */
        }
    }
}
```


南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.2 独立式键盘接口设计

- 中断方式的按键程序设计

```
#include <reg51.h>
/* 定义存储按键状态的全局变量 */
unsigned char Key;

/* 延时10ms函数及各按键处理函数同上例，略 */
/* 定义中断服务程序 */
void Int0ISR(void) interrupt 0
{
    Delay10ms();                        /* 延时消抖动 */
    Key = P1;                           /* 读取按键状态 */
}

/* 主程序 */
main()
{
    P1 = 0xff;                          /* 置P1为输入方式 */
    Key = 0;                            /* 初始化Key */

    IT0 = 1;                            /* 设置Int0为下降沿触发 */
    EX0 = 1;                            /* 允许Int0中断 */
    EA = 1;                             /* 打开总中断使能 */
```


南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.2 独立式键盘接口设计

- 中断方式的按键程序设计

```
    while(1)
    {
        if(Key == 0) continue;          /* 如果无键按下则继续等待 */
        switch(Key)                     /* 根据按键状态进行分支处理 */
        {
            case 0xfe:                  /* P1.0为低，K1按下 */
                ProcessK1();
                break;
            case 0xfd:                  /* P1.1为低，K2按下 */
                ProcessK2();
                break;
            ......
            case 0x7f                   /* P1.7为低，K8按下 */
                ProcessK8();
                break;
            default:
                Key = 0;
                continue;               /* 继续扫描 */
        }
        Key = 0;
    }
}
```

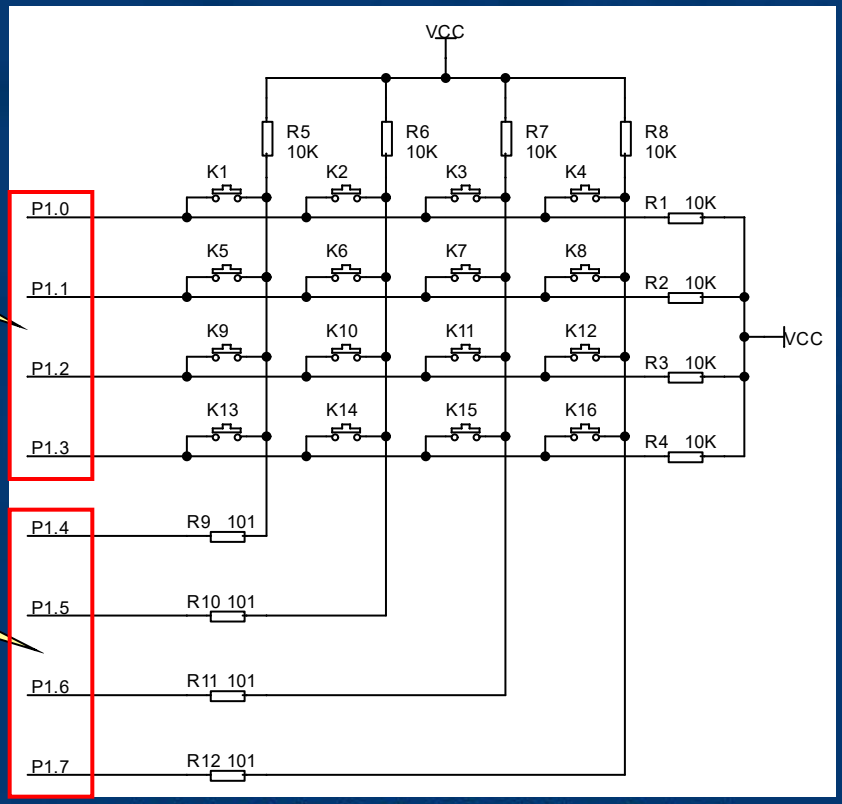

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.3 行列式键盘接口设计

- 由横跨行线和列线的开关矩阵组成的键盘
- 行列式键盘可有效节约端口引脚的使用

和行线连接的单片机端口，作为输出

和列线连接的单片机端口，作为输入



南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

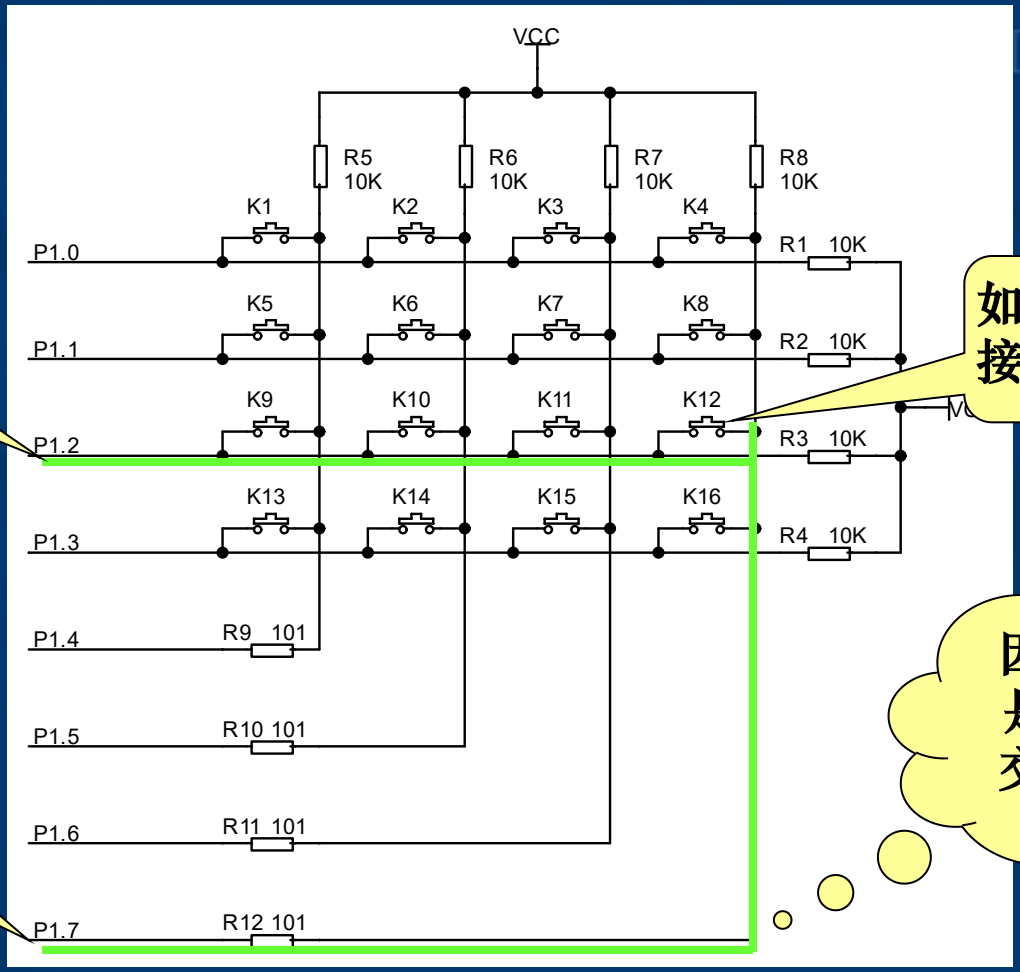
6.1.3 行列式键盘接口设计

行列式键盘按键识别原理

- 当行线作为输出，列线作为输入时：
  - 定时控制某根行线输出低电平，且不断循环；
  - 读入所有列线，如果结果不为全1，则有按键按下；
  - 找到读入为0的列，结合目前输出低电平的行，即可判断出具体的按键位置。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.3 行列式键盘接口设计



P1.2输出低电平

P1.7读到低电平

如果此时K12按下，则接通了P1.2和P1.7

因此，判断出是P1.2和P1.7交叉点的按键按下

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.3 行列式键盘接口设计

按键抖动的问题

- 可采用如下方法解决按键抖动问题：

(1) 方法1

- 扫描到按键按下后，等待20ms，再次读入按键状态进行确认。这种方法很方便，但是等待过程CPU不能处理别的事情，浪费了CPU的处理能力。

(2) 方法2

- 采用定时间隔扫描和异或法判断按键，将间隔值设置成大于抖动时间的值，例如20ms甚至更大，这样即使某次扫描正好处于抖动阶段，也不会对结果产生影响（但会使判断结果延时一次扫描间隔输出）。

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.3 行列式键盘接口设计

行列式键盘程序设计方法

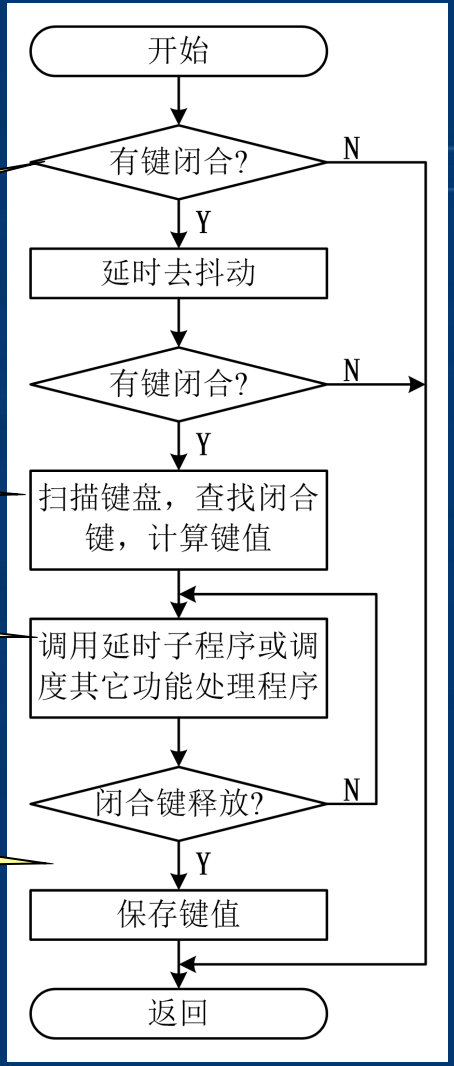
- 方法1：循环扫描

通过行输出0，列读入进行按键判断

具体判断按键的行/列

处理其他事务

等待按键释放后再保存键值，可避免重复处理



南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.3 行列式键盘接口设计

- 循环扫描方式的行列式按键程序设计

```
#include <reg51.h>
void Delay10ms(void)                    // 延时函数
{
    int i=0;
    for(i=0;i<2000;i++);                // 根据晶振频率等实际情况决定循环次数
}
void main(void)
{
    unsigned char KeyCode,Key,Line,i;
    while(1)
    {
        P1 = 0xf0;                      // 列线设置为输入，行线全部输出0
        Key = P1 & 0xf0;                // 读取列线值
        if(P1 != 0xf0)                  // 有键按下
        {
            Delay10ms();                // 延时去抖动
            Key = P1 & 0xf0;            // 再次读取按键状态
            if(Key == 0xf0)
                continue;               // 按键抖动，重新扫描
        } else continue;                // 无键按下，重新扫描
        for(i=0;i<4;i++)                // 开始循环扫描，确认具体按键
        {
            P1 = 0xff;
            Line = 0x01;
            Line <<= i;                 // 从第1行开始依次输出0
            P1 &= (~Line);
            Key = P1 & 0xf0;            // 读取行列式键盘列线状态
```


南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.3 行列式键盘接口设计

- 循环扫描方式的行列式按键程序设计

```
            switch(Key)
            {
                case(0xe0):                 // 第一列有键按下
                    KeyCode = 0x10 | (i+1);
                    break;
                case(0xd0):                 // 第二列有键按下
                    KeyCode = 0x20 | (i+1);
                    break;
                case(0xb0):                 // 第三列有键按下
                    KeyCode = 0x30 | (i+1);
                    break;
                case(0x70):                 // 第四列有键按下
                    KeyCode = 0x40 | (i+1);
                    break;
                default:                    // 本行无键按下或有重键
                    KeyCode = 0;
                    break;
            }
            if(KeyCode != 0) break;         // 确认一个按键后即退出
        }
        // 等待按键释放再执行按键处理程序
        while( (P1 & 0xf0) != 0xf0);
        switch(KeyCode)
        {
            //......                        // 根据键码调用相应的按键处理程序（略）
        }
    }
}
```


南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.1.3 行列式键盘接口设计

行列式键盘程序设计方法

- 方法2：定时中断扫描

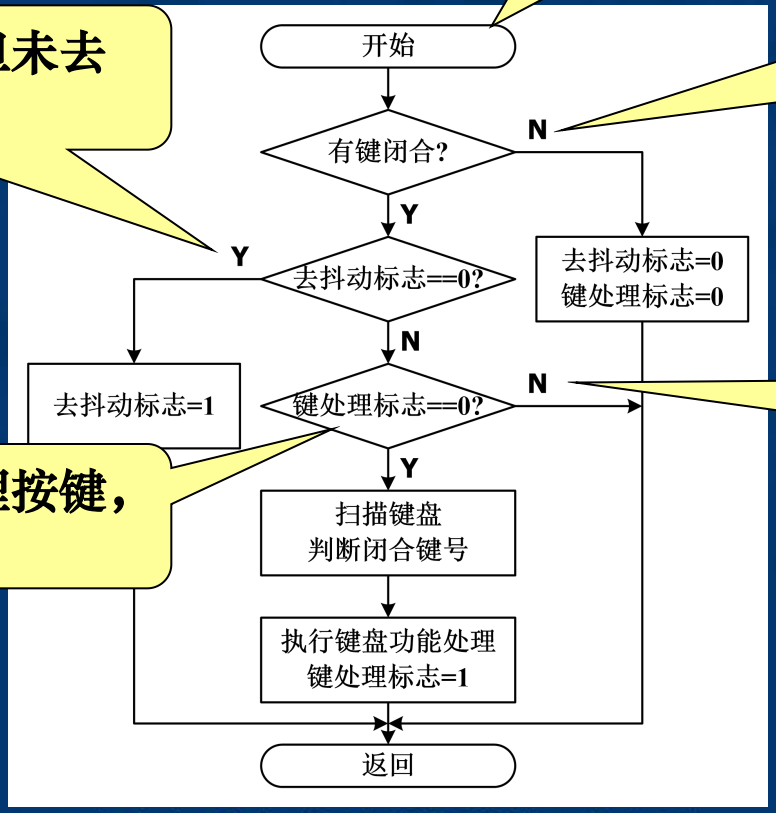
定时中断服务程序，每隔固定时间间隔执行一次

按键一旦释放，则清所有标志，开始下一个判断过程

扫描到有键按下，但未去抖动，则先去抖动

按键已处理，但未释放，则不再处理

已去抖动，但未处理按键，则处理按键



南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

第六章 单片机输入输出接口及系统扩展设计

6.2 LED显示器及接口设计

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.2.1 数码管显示器的结构和原理

1、单个LED的驱动

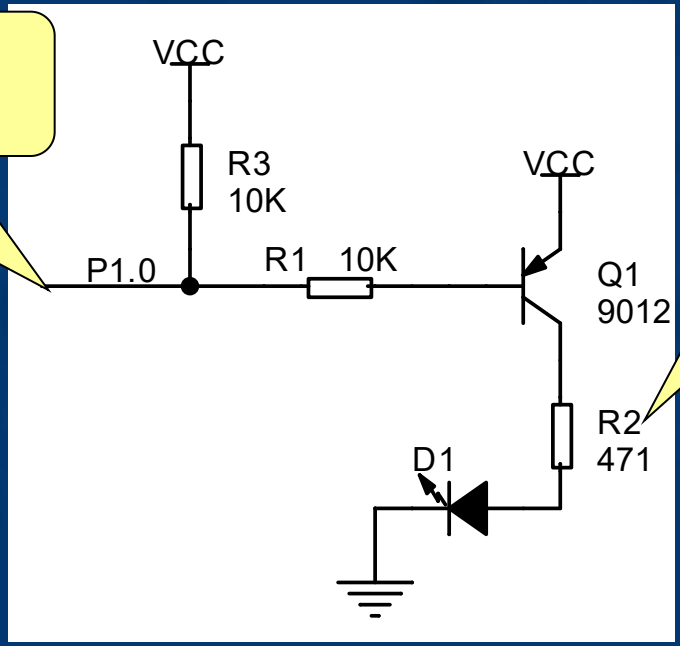
- 通过单片机端口引脚灌电流方式驱动：

6.2.1 数码管显示器的结构和原理

1、单个LED的驱动

- 通过单片机端口驱动三极管的方式驱动：

端口引脚输出低电平，则Q1导通，LED点亮



限流电阻，避免流过LED电流过大

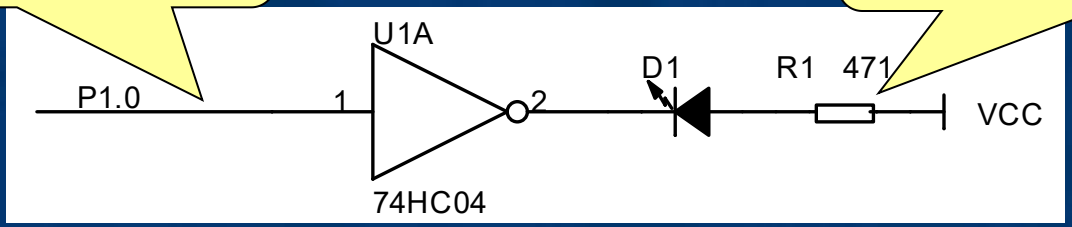
6.2.1 数码管显示器的结构和原理

1、单个LED的驱动

- 通过单片机端口驱动专用集成电路的方式驱动：

端口引脚输出高电平，则LED点亮

限流电阻，避免流过LED电流过大

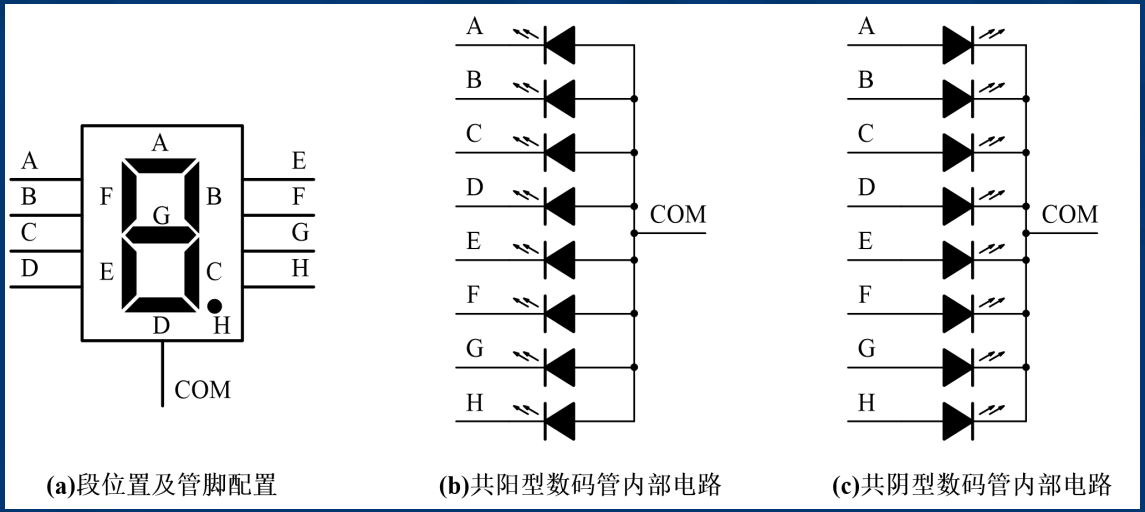


南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.2.1 数码管显示器的结构和原理

2、单个数码管的驱动

- 将多个LED封装在一起，即可构成笔划式显示器件，因常用于显示数字，故称“数码管”；
- 根据LED电路连接方式，数码管可分为共阳/共阴两种型号；
- 下图为最常见的8段数码管的外形及结构示意图：



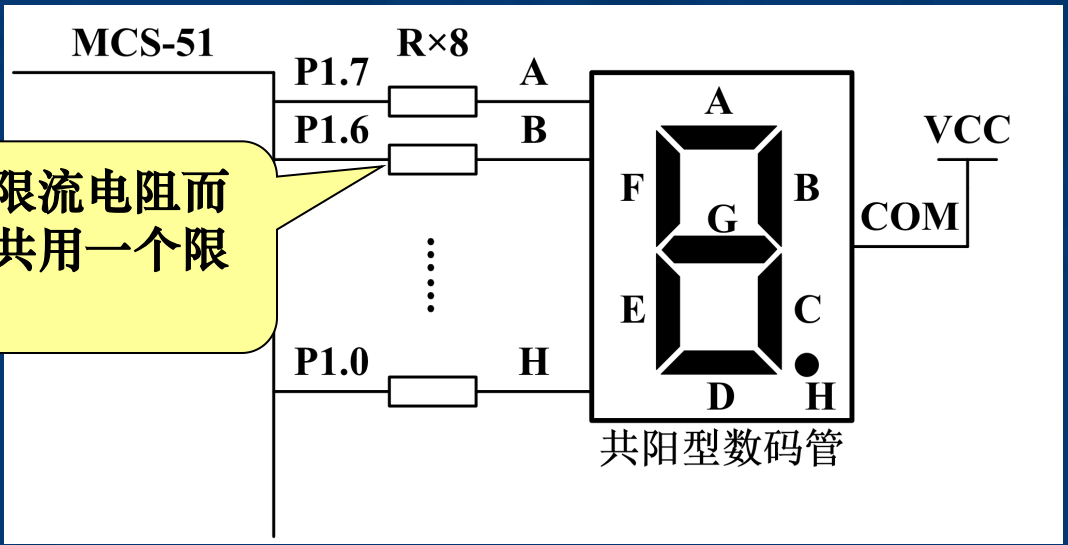
(a)段位置及管脚配置　(b)共阳型数码管内部电路　(c)共阴型数码管内部电路

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.2.1 数码管显示器的结构和原理

2、单个数码管的驱动

- 单片机驱动数码管中不同的笔划点亮，即可构成不同的字型；
- 单片机驱动共阳型数码管的典型电路如下：



为什么每段一个限流电阻而不是整个数码管共用一个限流电阻?

6.2.1 数码管显示器的结构和原理

2、单个数码管的驱动

- 构成各种字型的驱动编码称为数码管的字型码；
- 对于共阳型的数码管，单片机端口输出0相应笔划点亮，根据”0”～”F”不同字型数码管点亮的段，以及单片机端口引脚和数码管各段的连接顺序，相应的字形码如下页表所示：

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269

6.2.1 数码管显示器的结构和原理

字符	P17	P16	P15	P14	P13	P12	P11	P10	编码
	a	b	c	d	e	f	g	dp	
0	0	0	0	0	0	0	1	1	0x03
1	1	0	0	1	1	1	1	1	0x9F
2	0	0	1	0	0	1	0	1	0x25
3	0	0	0	0	1	1	0	1	0x0D
4	1	0	0	1	1	0	0	1	0x99
5	0	1	0	0	1	0	0	1	0x49
6	0	1	0	0	0	0	0	1	0x41
7	0	0	0	1	1	1	1	1	0x1F
8	0	0	0	0	0	0	0	1	0x01
9	0	0	0	0	1	0	0	1	0x09
A	0	0	0	0	0	1	0	1	0x05
b	1	1	0	0	0	0	0	1	0xC1
C	1	1	1	0	0	1	0	1	0xE5
d	1	0	0	0	0	1	0	1	0x85
e	0	1	1	0	0	0	0	1	0x61
F	0	1	1	1	0	0	0	1	0x71

南京邮电大学计算机学院 倪晓军
nixj@njupt.edu.cn 13505151269