微机原理和接口技术

第十九讲 人机接口技术1

提 纲

1. 键盘基础知识

2. 独立式键盘接口技术

3. 矩阵式键盘接口技术

4. 段码式LED接口技术

5. 点阵式LED接口技术

提 纲

1．键盘基础知识

键盘基础知识

键盘：微机系统中最常用的输入设备，用户通过键盘输入命令、数据，实现人机交互。

键盘与微控制器的接口包括硬件与软件两部分。

- 硬件是指键盘的组织，即键盘结构及其与MCU的连接方式。
- 软件是指对按键操作的识别与分析，称为键盘管理程序。

键盘管理程序：

- **识键：**判断是否有键按下。若有，则进行译码；若无，则等待或转做别的工作。
- **译键：**识别出哪一个键被按下，并产生相应的键值。
- **去抖动：**消除按键按下或释放时产生的抖动。
- **键值分析：**根据键值，执行对应按键的处理程序。

键盘基础知识

1．键盘的组织

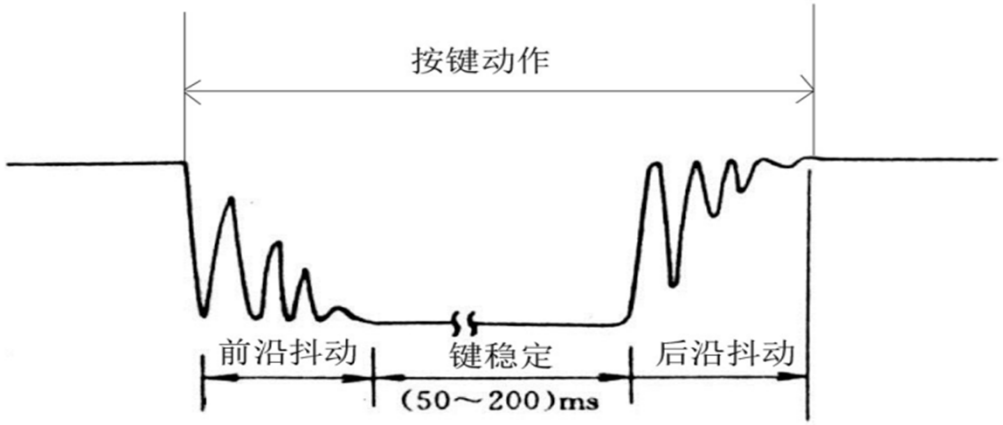
键盘可分为编码式键盘或非编码式键盘。

- **编码式键盘：**由键盘和专用键盘编码器（键盘管理芯片）两部分构成。键盘管理芯片自动完成键盘的扫描和译码。编码式键盘使用很方便，成本相对较高。常用的大规模集成电路键盘管理芯片如HD7279等。
- **非编码式键盘：**只简单地提供按键的通断信号，但某键按下时，键盘送出一个闭合（低电平）信号。该按键键值的确定必须借助于软件来实现。所以 非编码式键盘的软件比较复杂，占用CPU时间多。但成本低、使用灵活，在微机系统中，得到广泛应用。

非编码式键盘可分为独立式键盘和矩阵式键盘。

2．按键抖动与消除

触点式按键在闭合和断开瞬间存在抖动过程，即存在抖动现象，前后沿抖动时间一般在5ms～10ms。按键的稳定时间与按键动作有关，通常大于50ms。



按键抖动可能导致微机对一次按键操作作出多次响应，所以要去抖动。

（1）硬件电路去抖动：需要利用RS触发器等构成去抖动电路（很少使用）。

（2）软件延时法：当检测到有键按下时，用软件延时10ms～20ms，等待键稳定后重新再判一次，以躲过触点的抖动期。

键盘基础知识

3. 键盘的工作方式

微机系统中CPU对键盘进行扫描时，要兼顾两方面的问题：

- **要及时响应，保证系统对按键的每一次操作都能作出响应；**
- **不能占用CPU过多的时间。**

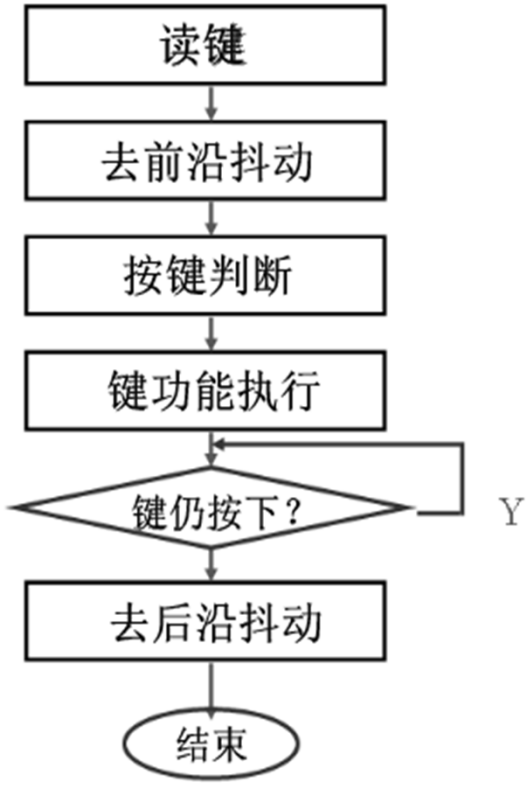
键盘的三种工作方式：

- **编程扫描方式（查询方式）：**是利用CPU在完成其他工作的空闲，调用键盘扫描程序。当CPU在运行其他程序时，不会响应按键操作。
- **定时扫描方式：**用定时器产生定时中断，在定时中断中对键盘进行扫描。定时中断周期一般应≤50ms。该方式常会出现CPU常空扫描状态。
- **中断工作方式：**当有键按下时，利用硬件产生外部中断请求，CPU响应中断后对键盘进行扫描。**该方式优于定时扫描方式，既能及时响应按键操作，又节省CPU时间。**

4. 按键连击的消除与利用

连击：一次按键操作被多次执行的现象称为连击，有利有弊。

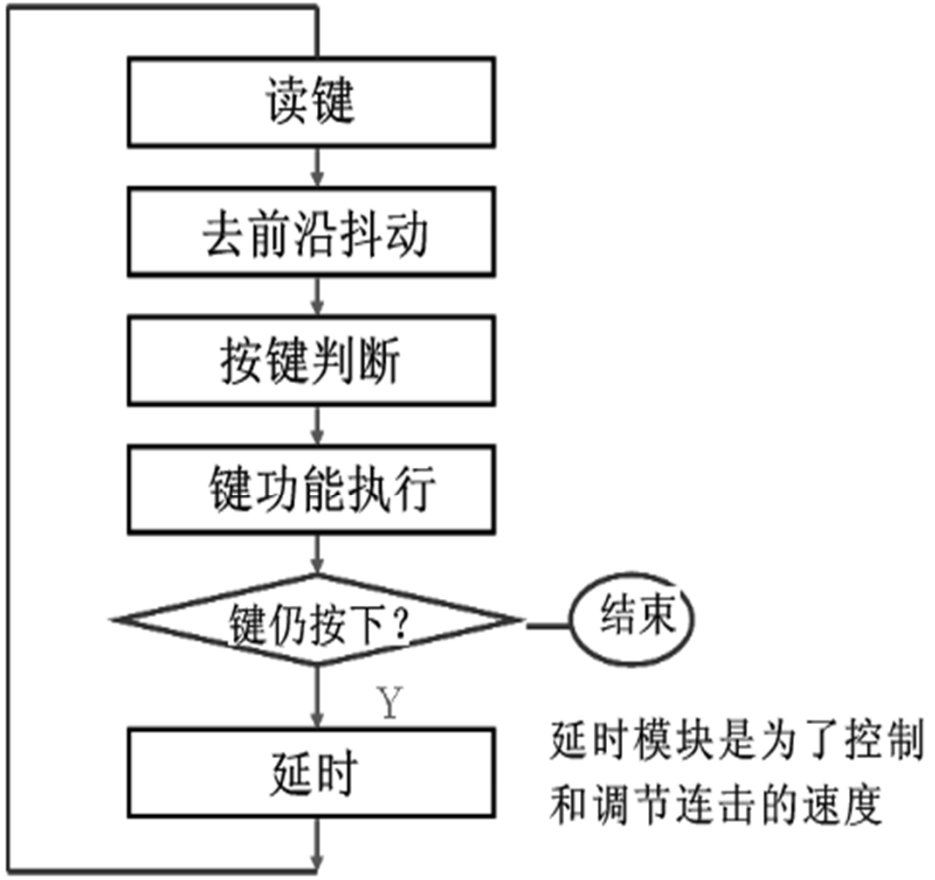
- **按键连击的消除：在程序中加入等待按键释放的处理，保证一次操作只被响应一次。**



键连击现象的消除

4. 按键连击的消除与利用

- **按键连击的利用：如设置“增1”、“减1”两个按键，利用按键的连击，长按住“增1”、“减1”键，则参数会不断增加或减少。可以替代0~9的数字键，有效减少按键数量。**



键连击现象的利用

5. 重键保护与实现

重键：由于操作不慎，可能会造成同时有几个键被按下，这种现象称为重键。出现重键时，就产生了如何识别和作出响应的问题。

处理重键的技术：

（1）“N键锁定”：

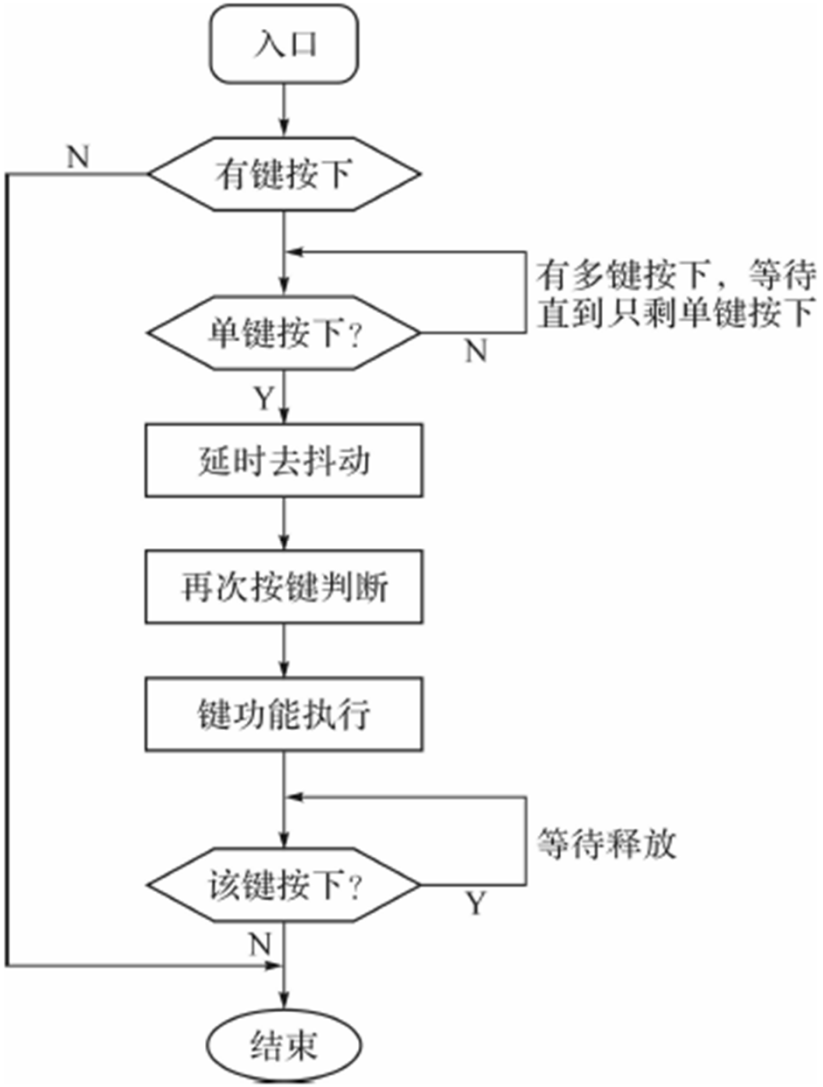
当扫描到有多个键被按下时，只把最后一个释放的按键作为有效键进行响应。

（2）“N键轮回”：

当扫描到有多个键被按下时，对所有按下的按键依次产生键值并作出响应。

5. 重键保护与实现

在微机系统中，通常采取单键按下有效、多键按下无效的策略，即采用N键锁定方法。



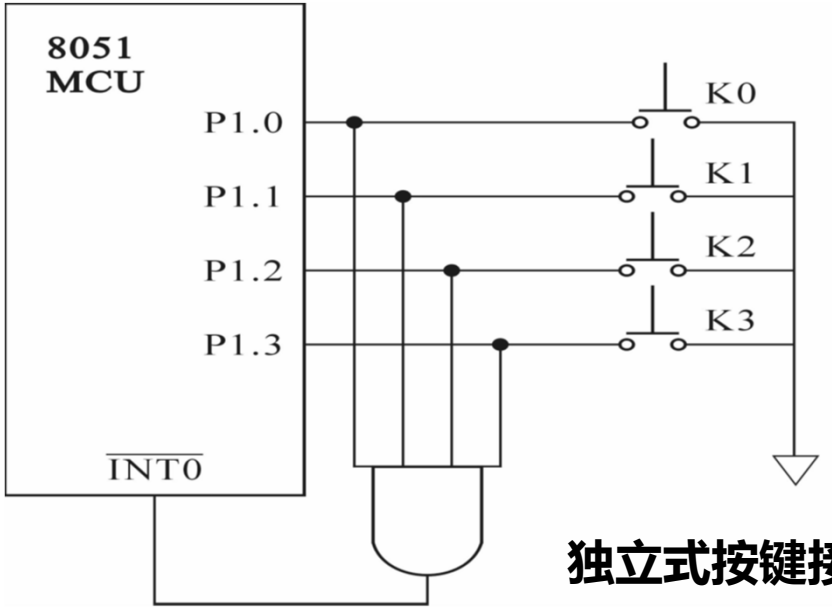
提 纲

2. 独立式键盘接口技术

独立式键盘接口技术

独立式键盘：每个按键占用一根I/O口线。无按键按下时，各I/O口线输入状态为高电平；当有按键按下时，I/O口线变为低电平。只要CPU检测到某一I/O口线为“0”，便可判别对应按键被按下。

- **优点：**结构简单，按键识别容易。
- **缺点：**当按键较多时，占用I/O口线多，只适用于按键较少的系统。

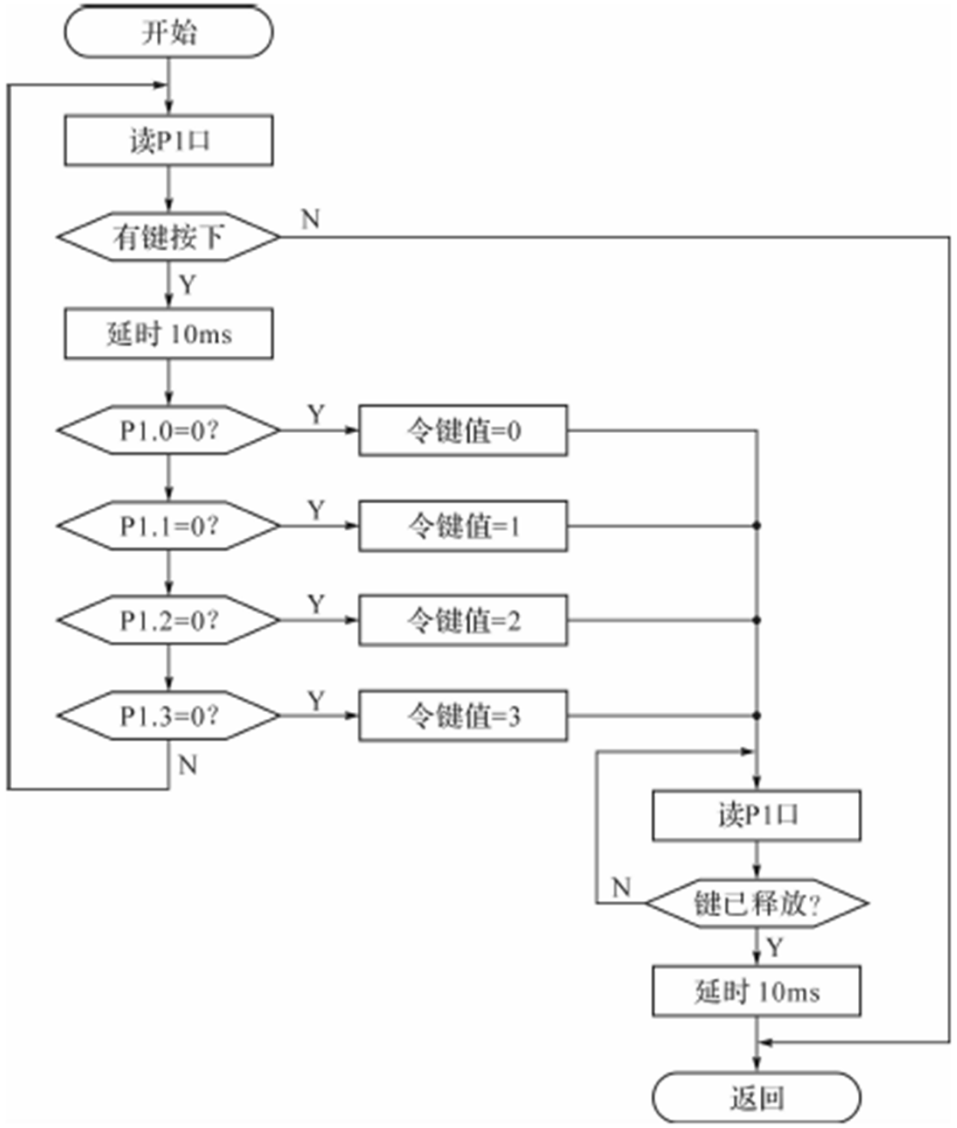


独立式按键接口电路

独立式键盘接口技术

程序流程（查询式）：

- 首先判断有无键按下，若检测到有键按下，延时10ms去抖动；
- 再逐位查询是哪个按键按下并执行相应按键的处理程序；
- 最后等待按键释放并延时10ms去除后沿抖动。



独立式键盘接口技术

中断扫描思路：

无键按下时，4与门输入全为高电平，不会产生中断。当任一键按下时，INT0变为低电平，向MCU请求中断。MCU响应中断，扫描按键，得到键值。

汇编主程序：

```
        ORG     0000H
        SJMP    MAIN
        ORG     0003H
        LJMP    INT0SUB                 ；外部中断0中断程序
        ORG     0100H
        MAIN:   SETB    IT0                     ；设置INT0为下降沿触发
方式
        SETB    EX0                     ；允许INT0中断
        SETB    EA                      ；允许CPU中断
        CLR     KEYFLAG                 ；清“有键按下”标志
        LOOP:   JNB     KEYFLAG, LOOP           ；等待中断
        CLR     KEYFLAG
        LCALL   KEYPROCESS              ；执行按键处理程序
        SJMP    LOOP
```

独立式键盘接口技术

汇编中断程序：

扫描按键，键值存放在R3中。

- K0的键值=0；
- K1的键值=1；
- K2的键值=2；
- K3的键值=3；

				K3	K2	K1	K0
0	0	0	0	P1.3	P1.2	P1.1	P1.0

A

C

```
            ORG     0200H
INT0SUB:    LCALL   DELAY10ms       ;去前沿抖动
            MOV     R3,#00H         ；设置键值初值
            MOV     A,P1
            ANL     A,#0FH
            CJNE    A,#0FH,SCAN     ；判断是否有按键按下
            MOV     A,#0FFH
            SJMP    NOKEY           ；不是正常的按键操作
SCAN:       MOV     R2,#4           ；设置查询按键数
SCAN1:      RRC     A
            JNC     FINDKEY         ；找到闭合的键
            INC     R3
            DJNZ    R2,SCAN1
            MOV     A,#0FFH         ；没有扫描到有效按键
            SJMP    NOKEY

FINDKEY:    MOV     A,R3
            SETB    KEYFLAG         ;建立“有键按下”标志
WAIT:       MOV     A,P1
            ANL     A,#0FH
            CJNE    A,#0FH,WAIT     ；等待按键释放
            LCALL   DELAY10ms       ;去后沿抖动
NOKEY:      RETI
```

提 纲

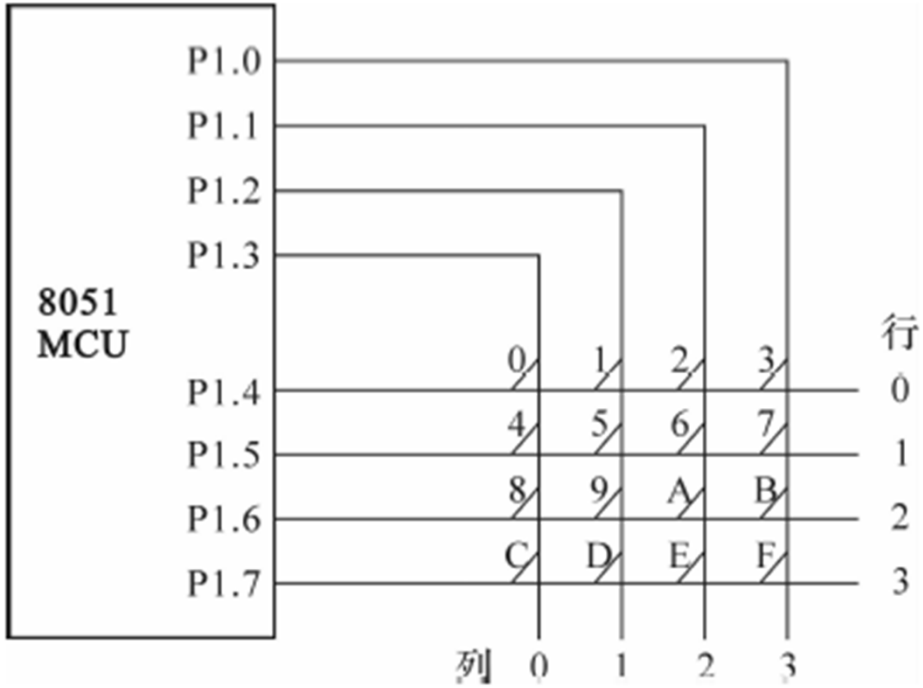
3. 矩阵式键盘接口技术

矩阵式键盘接口技术

矩阵式键盘：需要行线和列线，按键位于行线和列线的交叉点上；m×n矩阵键盘只需要m + n条口线。

按键数目较多的系统中，矩阵式键盘比独立式按键要节省很多I/O口线。

矩阵式键盘判别按键的方法有行扫描法和线反转法。



P1.4-P1.7为行扫描输出线；

P1.0-P1.3是列输入线。

若将4个列信号连接到一个4输入的与门，与门输出连接到外部中断引脚，则有键按下时，就会向CPU请求中断。

矩阵式键盘接口技术

1.行扫描法

1）粗扫描：判别是否有键按下。

- 所有行线输出均为0（相当于各行接地）,然后读入列值P1.3-P1.0。
- 如果**读入的P1.3-P1.0的值均为1**，说明没有键按下；
- 如果**读入的P1.3-P1.0的值不全为1**，说明有键按下。

若有键按下，延时10ms去抖动后，进行细扫描。

2）细扫描：识别哪个键按下。

➢ **逐行扫描**：依次给各行线输出低电平，然后读列值。

- 先令行P1.4输出0，其余输出1，然后读入列值。
- 若读入的**P1.0-P1.3为全1**，说明该行无键按下；再对下一行进行扫描；
- 若读入的**P1.0-P1.3不为全1**，则说明该行有键按下，要求出其键值。
- 直至全部行扫描完毕。

矩阵式键盘接口技术

1.行扫描法

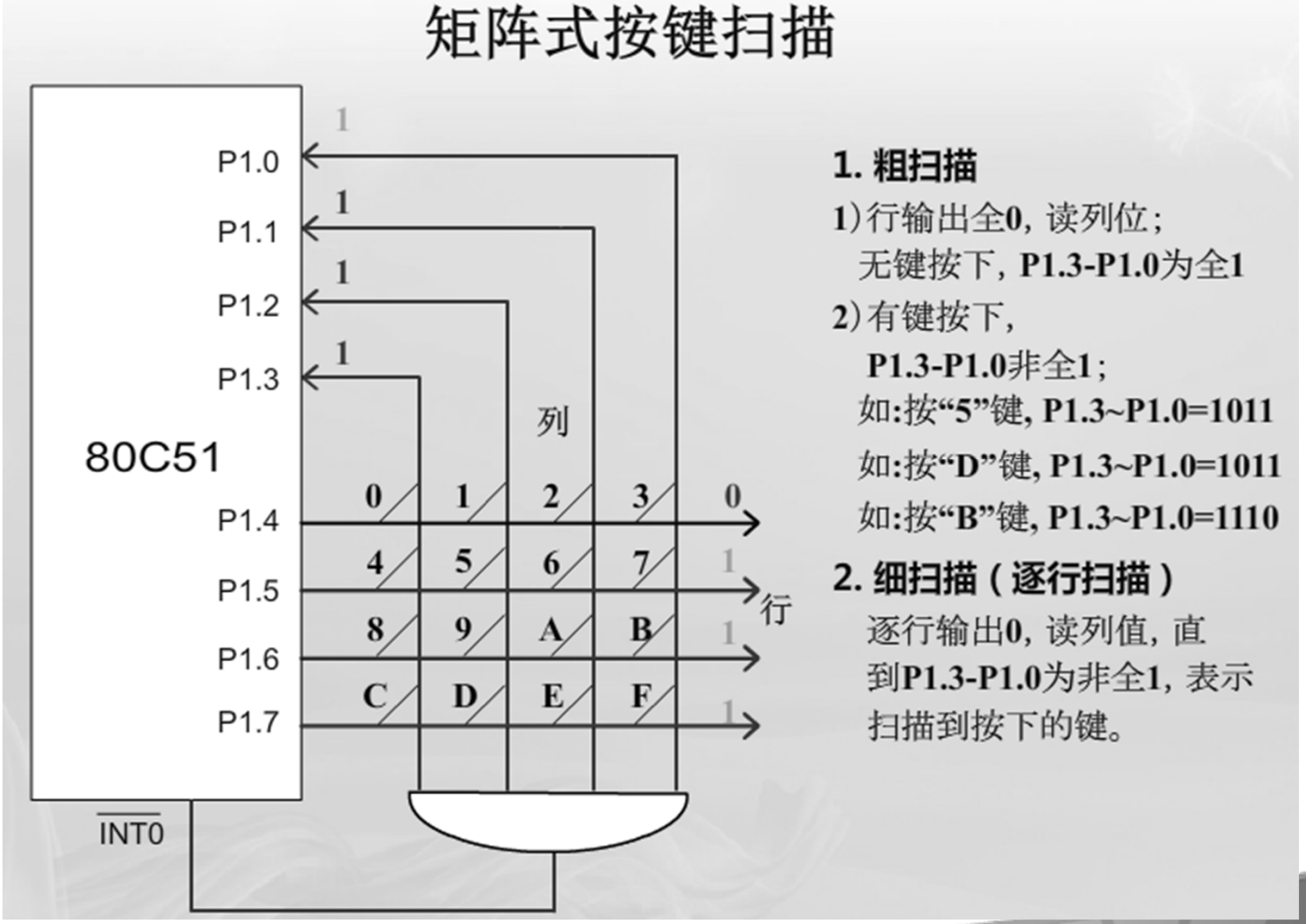
- **键值确定。按键位置与键值定义如下。**

列号 键值 行首键号	0	1	2	3
00	0	1	2	3
04	4	5	6	7
08	8	9	A	B
0C	C	D	E	F

闭合键的键值=行首键号+列号

矩阵式键盘接口技术

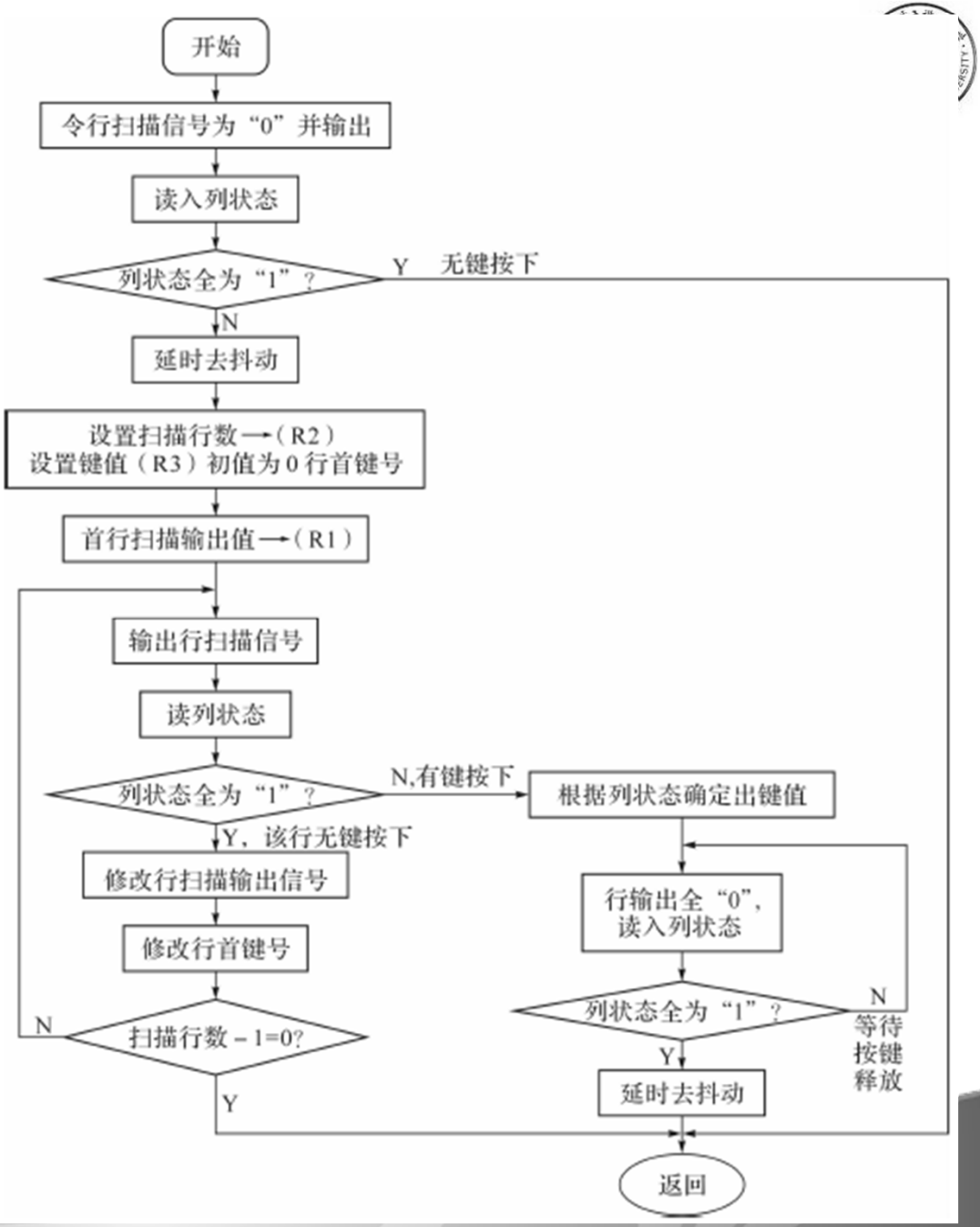
1.行扫描法



矩阵式键盘接口技术

行扫描法程序流程

为保证一次按键操作，CPU只响应一次，程序需等按下的按键释放后再结束。



矩阵式键盘接口技术

行扫描法程序（汇编）：

```
                ORG     0100H
KeySCAN: MOV    P1，#0FH；行输出“0”；P1.3~P1.0设置为输入方式
         MOV    A，P1
         ANL    A，#0FH
         CJNE   A，#0FH，HAVEKEY；判断是否有键按下
         SJMP   Nokey
HAVEKEY:LCALL   delay10ms
         MOV    R3，#0  ；设置键值为0行首键号
         MOV    R2，#4  ；扫描行数
         MOV    R1,11101111B
AGAIN:   MOV    P1，R1          ；首行扫描输出
         MOV    A，P1           ；读取列状态
         ANL    A，#0FH
         CJNE   A，#0FH,FINDKEY；判断该行是否有键按下
         MOV    A，R1   ；若没有键按下，则修改扫描行输出
         RL     A
         MOV    R1，A
         MOV    A，R3   ；修改行首键号
         ADD    A，#4
         MOV    R，A
         DJNZ   R，AGAIN
```

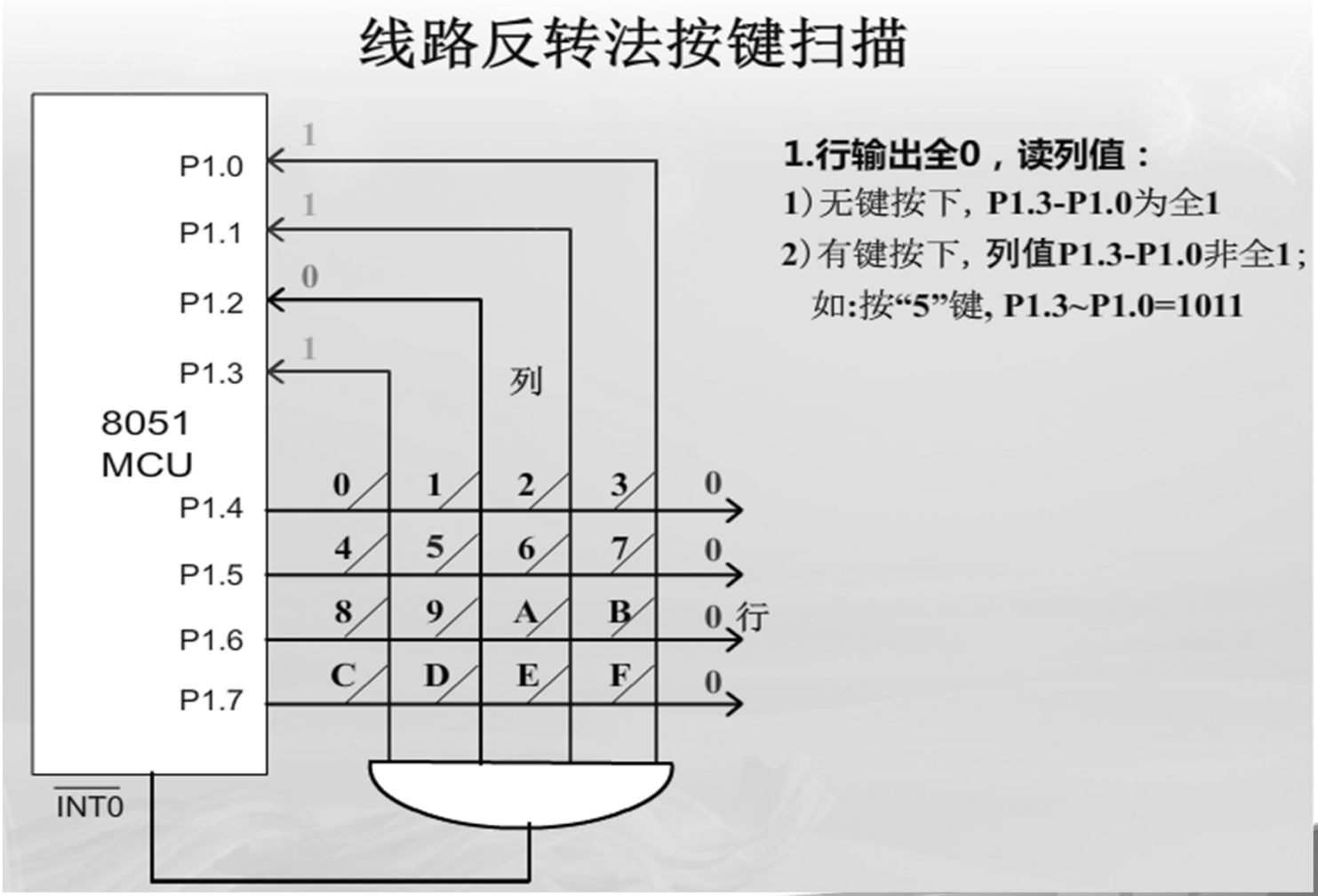
矩阵式键盘接口技术

行扫描法程序（汇编）：

```
FINDKEY:JB      P1.3，NEXTP12     ；依次判断对应行上哪一个键按下
        SJMP    FINDWT
NEXTP12:JB      P1.2，NEXTP11
        INC     R3
        SJMP    FINDWT
NEXTP11:JB      P1.1， NEXTP10
        INC     R3
        INC     R3
        SJMP    FINDWT
NEXTP10:JB      P1.0，Nokey
        INC     R3
        INC     R3
        INC     R3
FINDWT:MOV      P1，#0FH                ；等待释放
        MOV     A，P1
        ANL     A，#0FH
        CJNE    A，#0FH，FINDWT         ；键值保存到A
        MOV     A，R3
Nokey:  RET
```

矩阵式键盘接口技术

2.线路反转法



矩阵式键盘接口技术

2.线路反转法

1）行输出、列输入。令行线P1.7-P1.4输出全“0”，读入列线P1.3-P1.0的状态。设图中某E键被按下，此时读入的P1.3-P1.0为1101，根据“0”的位置可判断出该键在第2列上。

2）线路反转，即列输出、行输入。令列线P1.3-P1.0输出全“0”，读入列线P1.7-P1.4的状态。对于E键，读入的P1.7-P1.4为0111，其中“0”的位置对应该键行的位置，为第3行。

将两个步骤读入的状态合成一个代码，称为按键的特征码。如：E的特征码为01111101（7DH）。

经过两步便能获得键值，速度较快；要求行、列接口均为双向I/O接口。

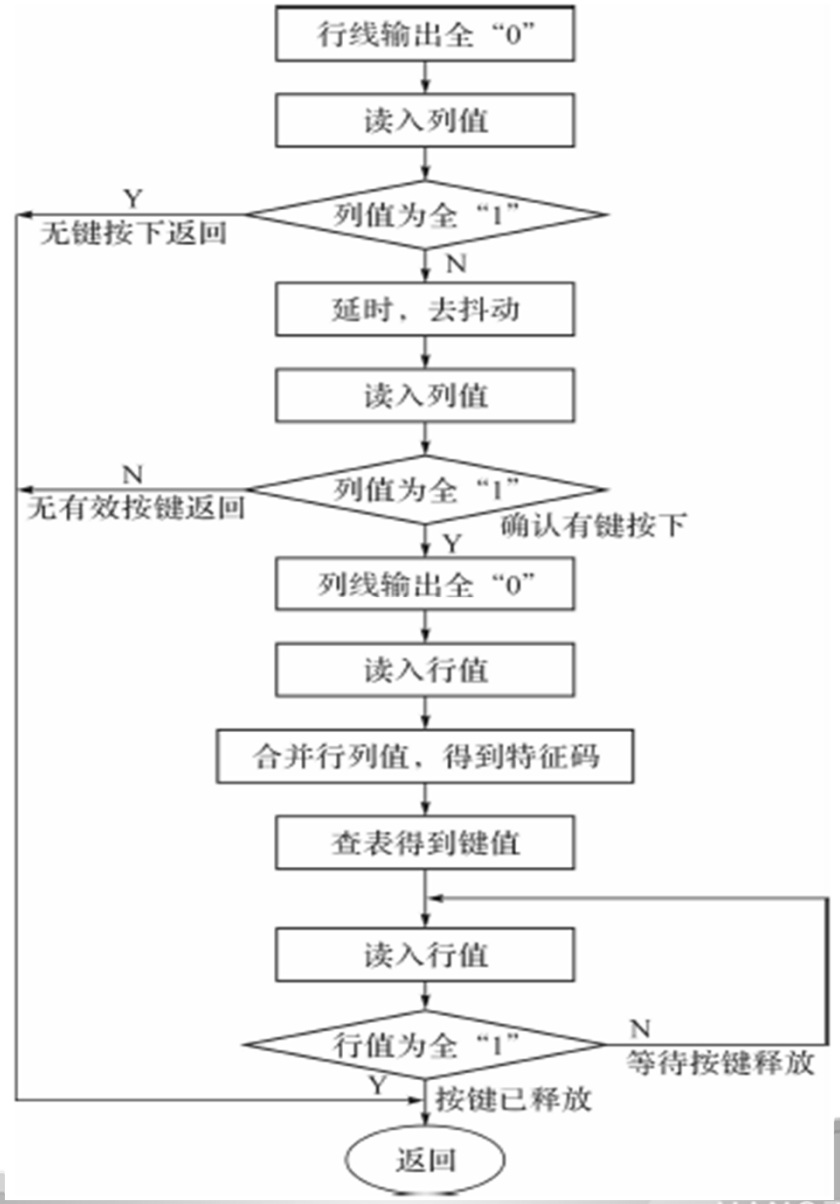
2.线路反转法

键盘的硬件连接确定后，每个按键就有一个**确定的特征码**；可建立定义的键值与特征码的转换关系。通常FFH定义为无键按下的特征码。

键名	特征码	键值	键名	特征码	键值
S0	E7H	00H	S8	B7H	08H
S1	EBH	01H	S9	BBH	09H
S2	EDH	02H	SA	BDH	0AH
S3	EEH	03H	SB	BEH	0BH
S4	D7H	04H	SC	77H	0CH
S5	DBH	05H	SD	7BH	0DH
S6	DDH	06H	SE	7DH	0EH
S7	DEH	07H	SF	7EH	0FH
			空键	FFH	无按键

矩阵式键盘接口技术

线路反转法流程



矩阵式键盘接口技术

线路反转法程序（汇编）： 扫描得到的键值在A中

```
KEYI:  MOV    P1，#0FH          ;行输出全“0” (P1高4位)
       MOV    A，P1;
       ANL    A，#0FH           ;读入列值（P1的低4位)
       MOV    B，A              ;列值送入B
       MOV    P1，#0F0H         ;线路反转；列输出全“0” (P1低4位)
       MOV    A，P1;
       ANL    A，#0F0H          ;读入行值（P1的高4位)
       ORL    A，B              ;合成特征码
       CJNE   A，#0FFH，KEYI1
       RET                      ;特征码=FFH，未按键返回
KEYI1: MOV    B，A              ;特征码保存到B
       MOV    DPTR，#KEYCD
       MOV    R3，#0FFH         ; 键值初始化为0FFH
```

矩阵式键盘接口技术

```
KEYI2：INC   R3
       MOV   A，R3
       MOVC  A，@A+DPTR
       CJNE  A，B，KEYI3          ;不等，继续
KEYI4: MOV    A,P1                ;找到键值在R3中，等待按键释放
       ANL    A,#0F0H
       CJNE   A,#0F0H,KEYI4       ;按键没有释放，继续等待
       MOV   A，R3                ; 已释放，存到A
       RET
KEYI3: CJNE  A，#0FFH，KEYI2     ;未查完，继续
       MOV   A，#0FFH             ;无键按下处理
       RET
KEYCD: DB  0E7H，0EBH，0EDH，0EEH
       DB  0D7H，0DBH，0DDH，0DEH
       DB  0B7H，0BBH，0BDH，0BEH
       DB  77H，7BH，7DH，7EH
       DB  0FFH
```

Thank you!