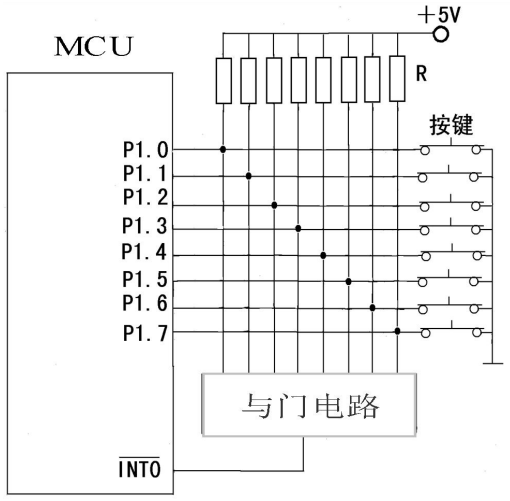
5.5　键盘接口

5.5.1 键盘接口的工作原理

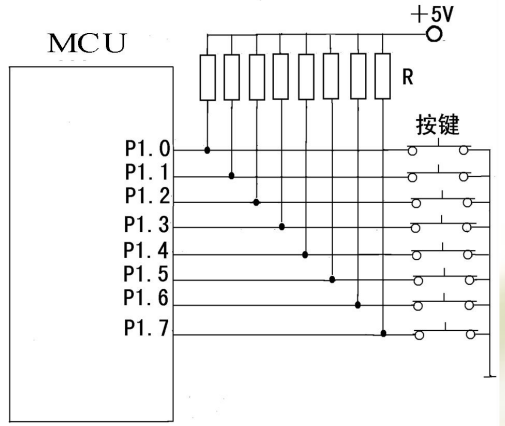
1.键盘连接方式

单片机的键盘连接方式分为两类：即独立式键盘和矩阵式键盘。

（1）独立式键盘（又称线性键盘）是指最简单的键盘电路，各键相互独立，每个按键独立地与一根单片机的I/O端口线相连接的方式构成的键盘电路。



(a) 中断方式



(b)查询方式

（2） 矩阵式键盘

是指把所有按键排列成行列矩阵形式的键盘。如图5.22所示，设选用P1端口中的P1.0~P1.3为四根行线，P1.4~P1.7为四根列线，行线和列线的交叉处放置一按键，当健按下时行列线接通，构成一个4×4的矩阵键盘，可定义16个按键。

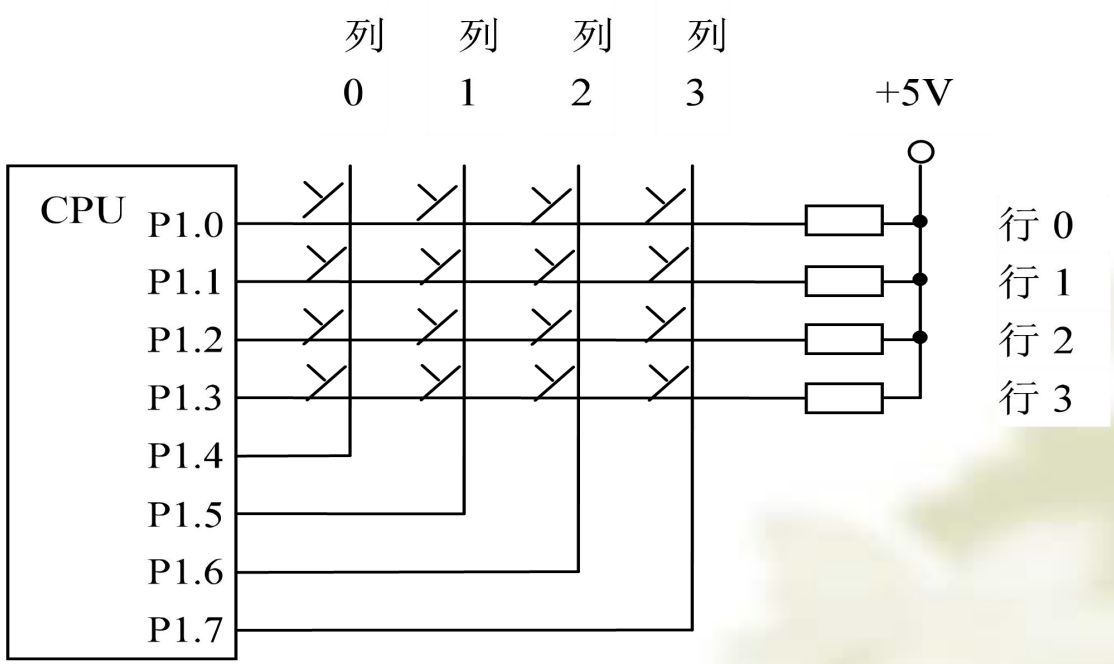


图5.22 4×4矩阵式键盘结构图

2.键盘扫描过程

1）.判断是否有键按下

2）.键盘消抖

3）.再次判断是否有键按下

4）.识别键码

5.5.2 键盘接口电路

80C51的I/O口具有输出锁存和输入缓冲的功能，因而用它们组成键盘电路时，可以省掉输出锁存器和输入缓冲器，图5.25所示，它是由MCS-51单片机本身的P1口来构成4×4矩阵式键盘。

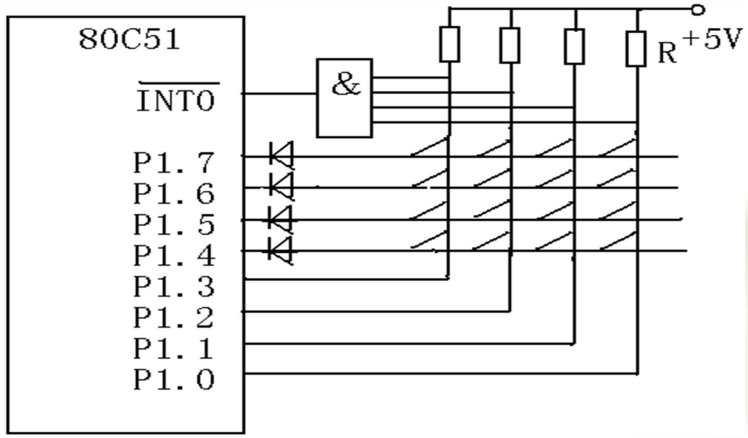
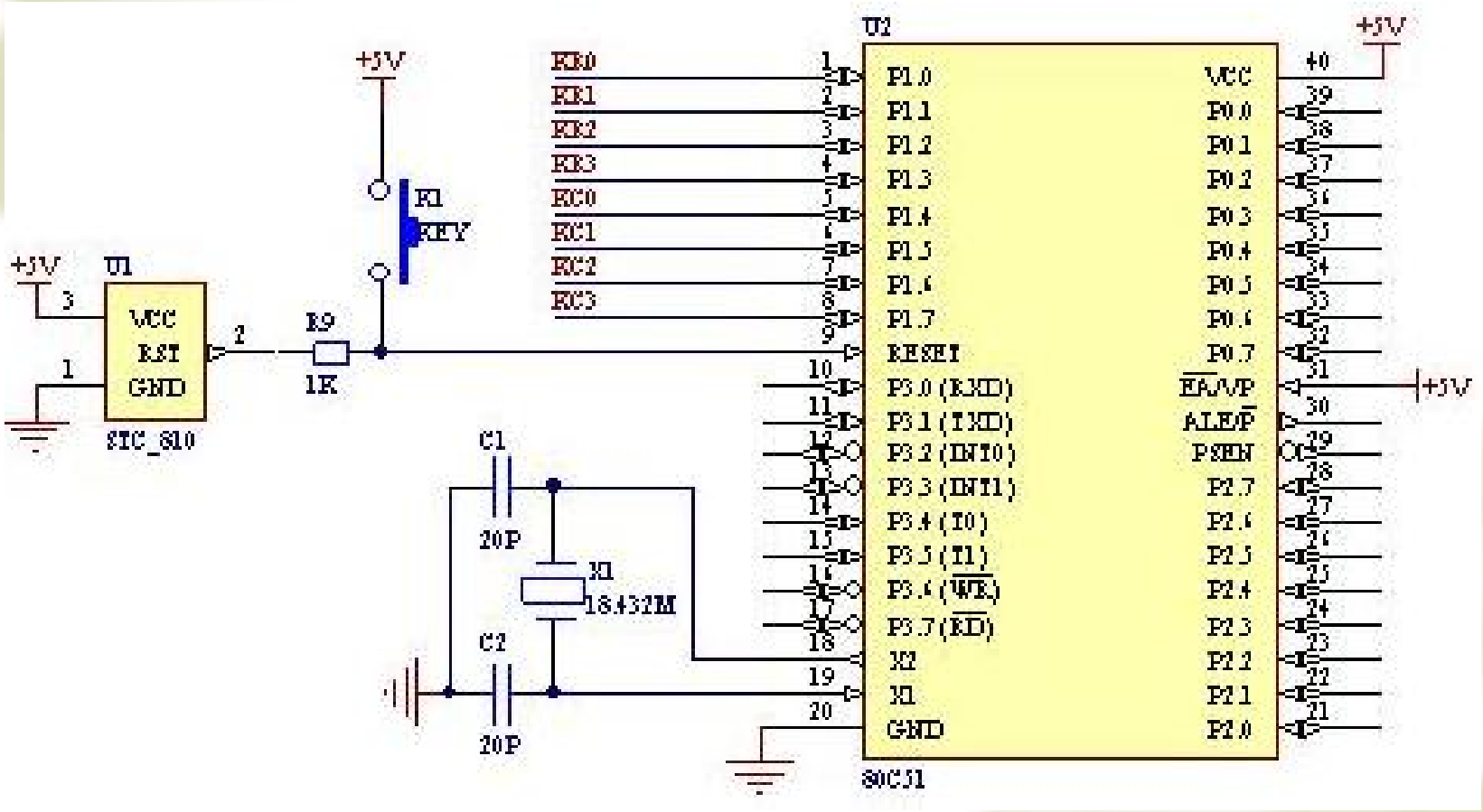


图5.25 采用中断申请方式的矩阵式键盘

5.5.3 键盘扫描实例



有一个4×4键盘，行的引出线分别为KR0~KR3,列的引出线为KC0~KC3,
利用单片机的P1口完成键盘的扫描

```
#include "SignalGen.h"

/* 键扫描函数 */
uchar keyscan(void)
{
        uchar scancode,tmpcode;
        P1 = 0xf0;                      // 发全0行扫描码
        if ((P1&0xf0)!=0xf0)            // 若有键按下
        {
                delay();                // 延时去抖动
                if ((P1&0xf0)!=0xf0)            // 延时后再判断一次，去除抖动影响
                {
                        scancode = 0xfe;
                        while((scancode&0x10)!=0)// 逐行扫描
                        {
                                P1 = scancode;      // 输出行扫描码
                                if ((P1&0xf0)!=0xf0)            // 本行有键按下
                                {
                                  tmpcode = (P1&0xf0)|0x0f;
                                /* 返回特征字节码，为1的位即对应于行和列 */
                                return((~scancode)+(~tmpcode));
                                }
                                else scancode = (scancode<<1)|0x01; // 行扫描码左移一位
                        }
                }
        }
        return(0);                                              // 无键按下，返回值为0
}
```

```
/* 主程序 */
void main()
{
        uchar key;
        while(1)
          {
                key = keyscan();              // 调用键盘扫描函数
                delay();
                switch(key)
                {
                case 0x11:                    // 第1行第1列，选择正弦波输出
                        A0 = 0;
                        A1 = 1;
                        break;
                    case 0x21:                // 第1行第2列，选择矩形波输出
                        A0 = 0;
                        A1 = 0;
                        break;
                    case 0x41:                // 第1行第3列，选择三角波输出
                        A0 = 1;
                        A1 = 0;
                        break;
                    default:break;
                }
```

5.6 A/D转换器

5.6.1 概述

A/D转换器是一种用来将连续的模拟信号转换成适合于数字处理的二进制数的器件，它能够将一个模拟信号值编制成与其大小相对应的二进制码的编码器。

D/A转换器则可以理解为一个解码器。模拟量与数字量之间有一一对应的关系，设D为N位二进制数字量，U_A为电压模拟量，U_{REF}为参考电压，无论A/D或D/A，其转换关系为：

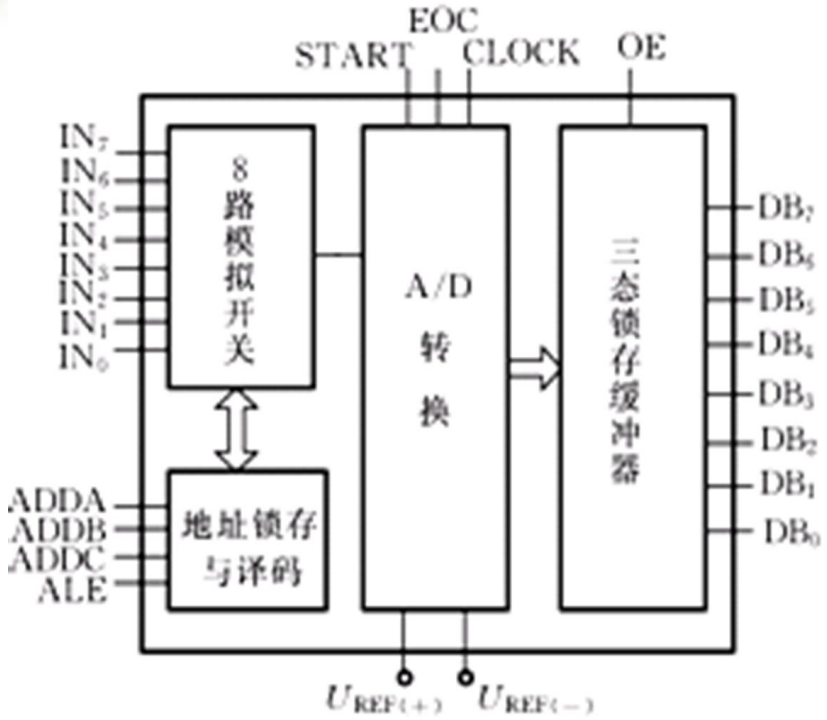
$$U_A = D \times U_{REF} / 2^N \quad （其中：D=D_0 \times 2^0 + D_1 \times 2^1 + \ldots + D_{N-1} \times 2^{N-1}）$$

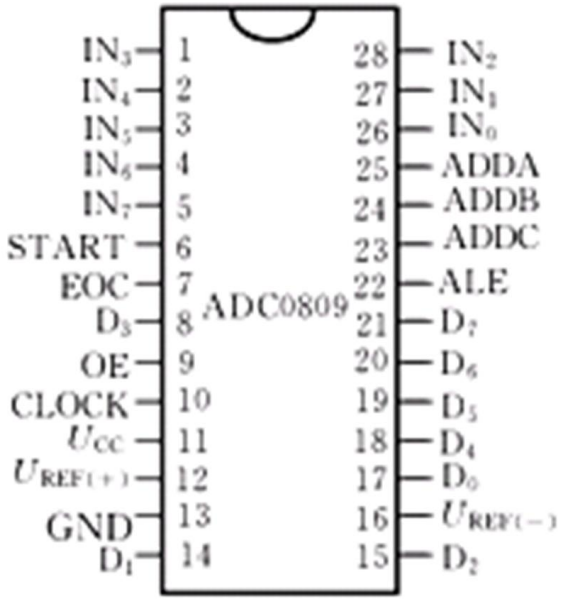
5.6.2典型A/D转换器芯片—ADC0809

1. ADC0809简介

ADC0809的分辨率为8位，不必进行调零和调满量程；最大不可调误差小于±LSB ； 输入电压范围0～+5.000V（5V单电源供电） 。转换时间为100μs左右；输出八位二进制数字量，与TTL电平兼容；具有锁存控制的8路多路选择模拟开关（8选1）

2.0809结构及引脚


EOC
START
CLOCK
OE
IN7
IN6
IN5
IN4
IN3
IN2
IN1
IN0
8路模拟开关
A/D转换
三态锁存缓冲器
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
ADDA
ADDB
ADDC
ALE
地址锁存与译码
UREF(+)
UREF(−)


IN3 1
IN4 2
IN5 3
IN6 4
IN7 5
START 6
EOC 7
D3 8
OE 9
CLOCK 10
Vcc 11
VREF(+) 12
GND 13
D1 14
ADC0809
28 IN2
27 IN1
26 IN0
25 ADDA
24 ADDB
23 ADDC
22 ALE
21 D7
20 D6
19 D5
18 D4
17 D0
16 VREF(−)
15 D2

引脚说明：

- IN0~IN7：8路模拟量输入通道，可以由ADDA、ADDB和ADDC三根线的状态进行通道选择；
- D0~D7：8位数据输出线，为三态缓冲输出形式，可以和单片机的数据总线直接相连（D0为最低位，D7为最高位）
- EOC：转换结束信号脉冲输出端，当EOC=1时表示A/D转换结束,EOC=0，表示正在转换，在与单片机联络中常用于中断或查询信号的判断。
- OE：输出允许信号，用于控制三态输出锁存器向单片机输出转换得到的数据。OE=0，输出数据线呈高阻；OE=1，输出转换得到的数据。
- START：A/D启动控制信号，上升沿复位ADC转换器，下降沿启动AD转换，常简称ST。
- ALE：地址锁存控制信号端，当它为1时，锁存某输入通道，确保A/D转换时采集的数据为指定通道上的数据，常与START共同使用。
- ADDA- ADDB- ADDC：8路模拟开关的三位地址选通输入端，以选择对应的输入通道，
- CLK：外部时钟输入端，其频率范围为10~1280KHZ
- V_{REF+}、V_{REF-}：参考电压输入端
- VCC、GND：＋5V工作电源、电源地

5.6.3 MCS-51单片机与ADC0809接口

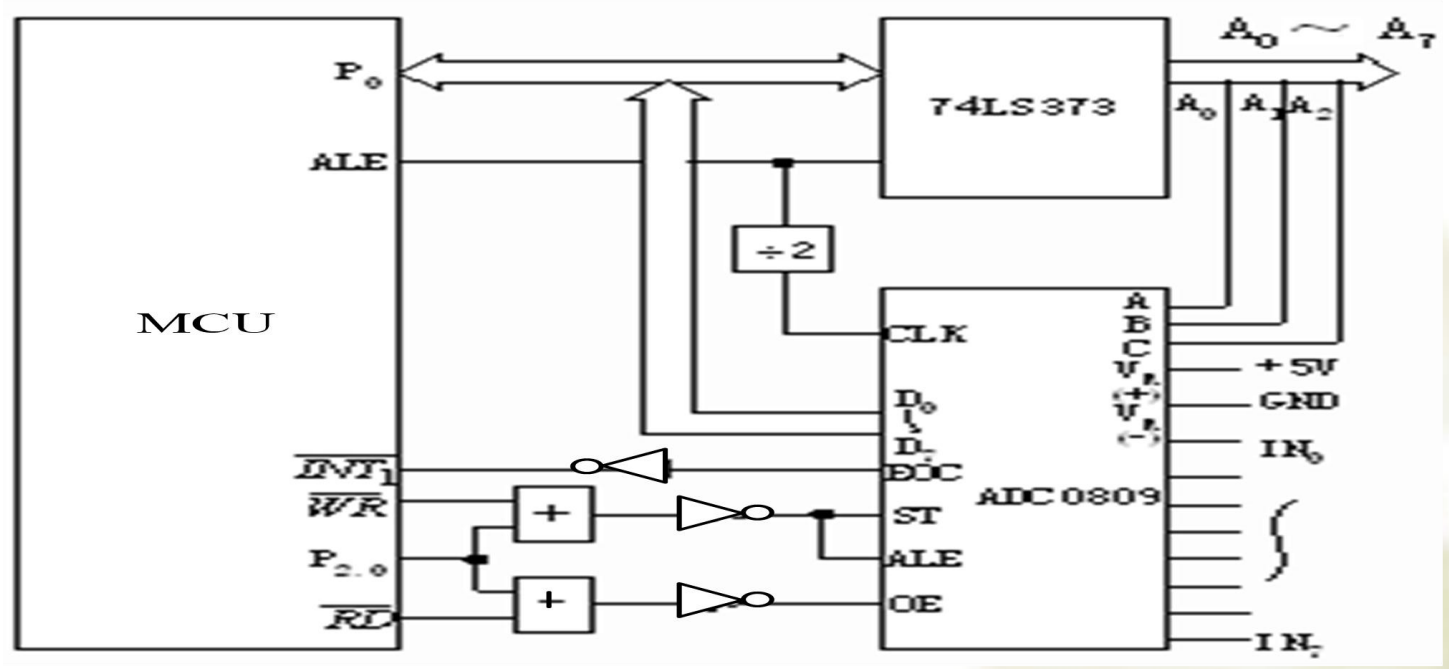
确认A/D转换完成有3种方式：

（1）定时传送方式：对于已选定的A/D转换器，其技术指标（转换时间是技术指标中的一项）是已知的。比如ADC0809转换时间为128μs，当MCS-51单片机的晶振为6MHz时，其转换所需时间为64个机器周期，可据此设计一个延时大于128μs的子程序，A/D转换启动后即调用此子程序，延迟时间到，确定转换结束，读取转换结果并进行处理。

（2）查询方式：一般A/D转换芯片都有一个表征转换结束的状态信号，例如ADC0809的EOC端。因此可以用查询方式，测试EOC的状态，即可判断转换是否结束，并读取转换结果。

（3）中断方式：把表征转换结束的状态信号（EOC）作为外部中断请求信号，以中断方式进行数据读取处理。

下图所示为ADC0809与单片机组成的一个8路模拟量输入的巡回监测系统连接图。



ADC0809地址为0FEF8H~0FEFFH分别对应于不同输入通道IN0~IN7。ADC0809工作过程如下：

① 先清零P2.0=0，令其控制的两个或门开启，然后通过单片机的P3.6（WR）送一次高电平，再送一次低电平，经过反相器后获得一个脉冲的上升沿，到达A/D的启动端ST和通道锁存端ALE，实现A/D转换器复位和通道锁存的功能；

② P3.6再送一次高电平信号，经过反相后形成一个下降沿，完成启动A/D转换的工作；

③ 等待中断信号的产生，一旦检测到外部中断1信号为0，表示有中断产生，则响应中断服务程序；

④ 在中断服务程序中，使P3.7(RD)引脚为低电平，反相后到达A/D的OE端，为高电平，表示允许转换后的数据传入P0口，从P0口读取数据。

主程序：

```
        ORG         0000H
        LJMP        MAIN        ；转到主程序
        ORG         0013H
        LJMP        INT1        ；转到中断服务程序
        ORG         0100H
 MAIN：MOV          R0，#0A0H；指向数据存储区首地址
        MOV         R2，#08H    ；8路计数器
        SETB        IT1         ；边沿触发方式
        SETB        EA          ；中断允许
        SETB        EX1         ；允许外部中断1中断
        CLR         P2.0        ；打开总控制开关
       SETB         P3.7        ；禁止取转换数据
       MOV          DPTR，#0FEF8H；选择A/D转换 器0通道
CONVERT：MOV        @DPTR，A
        SETB        P3.6 ；先产生上升沿复位A/D
        CLR         P3.6
        SETB        P3.6 ；产生下降沿，启动A/D
HERE：SJMP          HERE        ；等待中断
```

中断服务程序：

```
        ORG   0200H
 INT1：PUSH ACC
        PUSH   PSW
         CLR    P3.7        ；设置允许A/D输出数据的控制信号为OE=1
         MOVX  A，@DPTR  ；从选择的端口地址读入转换后的数据
        MOVX   @R0，A                    ；把数据存入指定地址
        INC       DPTR              ；指向下一模拟通道
         INC      R0                  ；指向数据存储器下一单元
       MOVX  @DPTR，A        ；选择下一通道
       DJNZ  R2，ADEND；判是否完成8路转换，未完成则返回主程序
      CLR   EX1                ；完成8路转换，关闭中断
ADEND：POP   PSW
         POP    ACC
         RETI              ；中断返回
```