

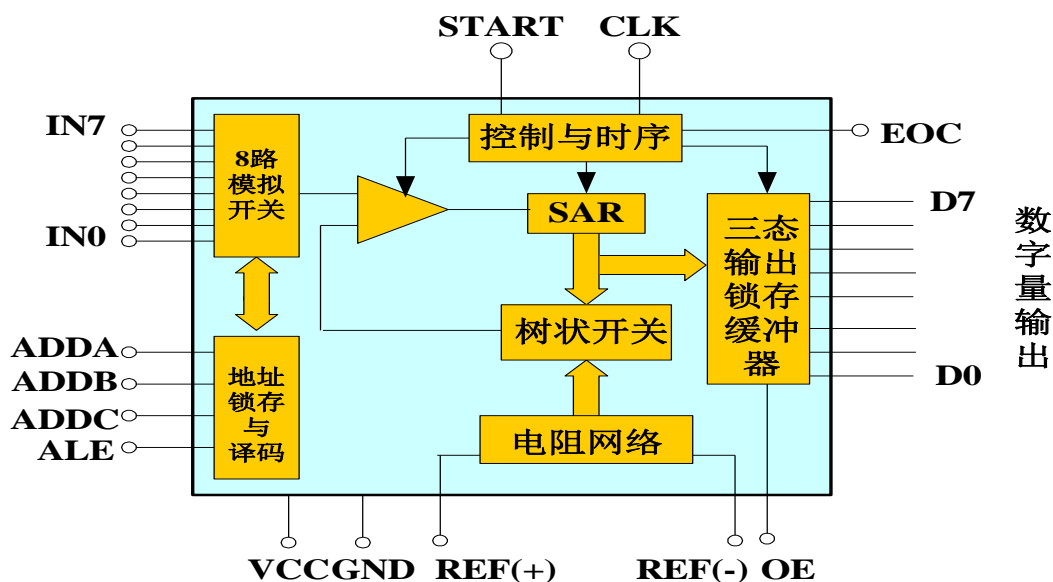
0809 实验

一、实验内容

- 1.将 INT0—INT7 分别连接到 2 个可变电阻器上，每秒采集一次 0809 转换的数字量，将转换后的数字量和路号在显示器 CRT 上显示，并将转换后的数字量送往 8 段显示，采集 10 个循环后结束。
- 2.将 INT0 连接到 1 个可变电阻器上，每秒采集一次 0809 转换的数字量 n ，要求将电压值 U 显示在 8 段显示器。

二、0809 知识

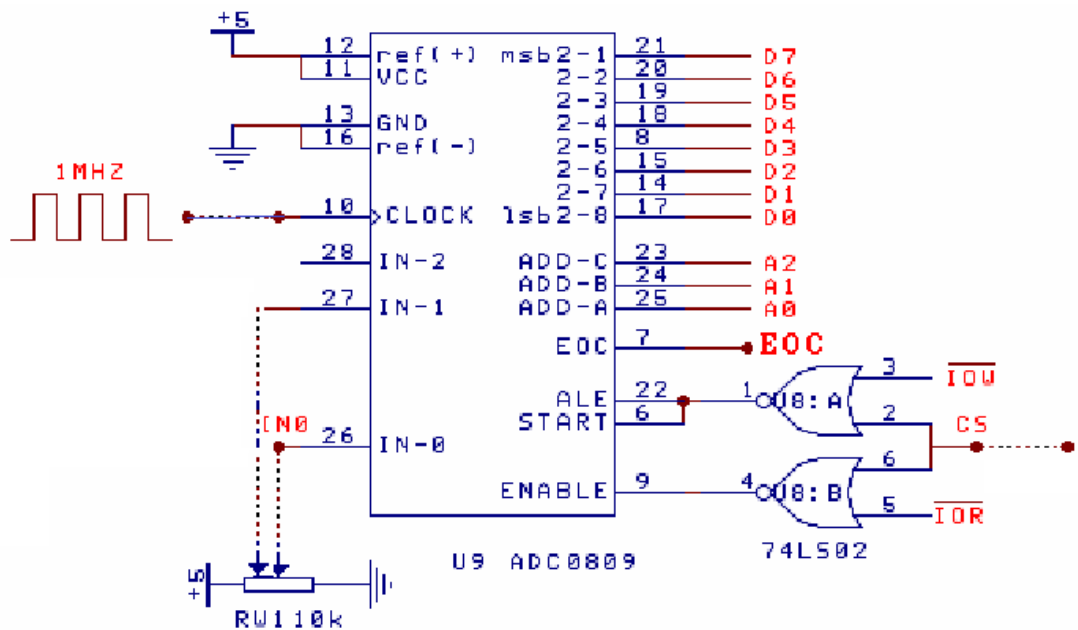
1. ADC0809 的工作过程：



- 第一步：首先确定 ADDA、ADDB、ADDC 三位地址，决定选择哪一路模拟信号。
- 第二步：使 ALE 端接收一正脉冲信号，使该路模拟信号经选择开关达到比较器的输入端。
- 第三步：使 START 端接收一正脉冲信号，START 的上升沿将逐次逼近寄存器复位，下降沿启动 A/D 转换。
- 第四步：EOC 输出信号变低，指示转换正在进行。
- 第五步：A/D 转换结束，EOC 变为高电平，指示 A/D 转换结束。此时，数据已保存到 8 位锁存器中。
- 第六步：OE 信号变为高电平，则 8 位三态锁存缓冲器的三态门被打开，转换好的 8 位数字量数据被输出到数据线上。

如上所述，EOC 信号变为高电平表示 A/D 转换完成，EOC 可作为中断申请信号，通知 8086CPU 取走数据。在查询传送方式中，EOC 可以作为 8086 查询外设的状态信号

2. 实验电路图



三、参考程序

;PC7--0809 的 EOC;PC0--8253 的 out1;PA--7 段数码管；PB4--数码管的 S0；PB5--数码管的 S1

DATA SEGMENT

```
TABLE_LED DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H
            DB 7FH,6FH,77H,7CH,58H,5CH,79H,71H
            DB 80H
```

```
TABLE_CRT DB 30H,31H,32H,33H,34H,35H,36H,37H
            DB 38H,39H,41H,42H,43H,44H,45H,46H
```

```
TABLE_KONG DW 0,290H,291H,292H,293H,294H,295H,296H,297H
```

```
NUM1 DB 10H
```

```
NUM2 DW 1010H
```

DATA ENDS

```

CODE  SEGMENT
    ASSUME CS: CODE, DS:DATA
START:
    MOV AX, DATA
    MOV DS, AX

    CALL Init_8253; 地址: 288-28BH,初始化 8253, 使之产生 1s 方波
    CALL Init_8255; 地址: 280-283H,初始化 8255, A 口输出, B 输出, C 口高位输入, C 低入
    CALL WAIT_INSPACE; 等待空格键
    MOV CX, 11
    MOV SI, 0
INIT:
    MOV DI, 0
READ_INPUT:
    CALL LIGHT_LED ;8 段显示
    MOV DX,282H ;读 C 口
    IN AL, DX
    Test AL,01H ;PC0=0?
    JZ READ_INPUT ; PC0=0

    INC DI ; 通道号
    CMP DI, 9
    JZ NEXT_INPUT

    PUSH DI
    SHL DI, 1;乘以 2
    MOV DX, TABLE_KONG[DI]
    POP DI
    OUT DX, AL
    CALL DELAY
    IN AL, DX
    MOV AH, AL ;高位 AH 低位 AL 中
    PUSH CX
    MOV CL,4 ;读高 4 位 AH 中
    SHR AH,CL ;读高 4 位 AH 中
    POP CX
    AND AL,0FH ;低位 AL 中

```

```
MOV NUM2, AX
MOV AX, DI
MOV NUM1, AL
CALL CRT ; CRT 显示
```

WAIT_TIMEOUT:

```
MOV DX,282H ;读 C 口
IN AL,DX
AND AL, 01H
JNZ WAIT_TIMEOUT ;PC0=1
JMP READ_INPUT
```

NEXT_INPUT:

```
DEC CX
CMP CX,0
JZ FINAL ;CX=0?
JMP INIT
```

FINAL:

```
MOV AH,4CH
INT 21H
```

;=====函数具体实现=====

DELAY PROC NEAR

```
PUSH DX
```

AGAIN:

```
MOV DX, 282H;读 C 口
IN AL,DX
test AL,80H ;PC7=0?
JZ AGAIN ;PC7=0
POP DX
RET
```

DELAY ENDP

LIGHT_LED PROC NEAR

```
MOV AL,0H ;熄灭
MOV DX,281H
OUT DX,AL
```

```
MOV AX, NUM2
```

```
LEA  BX,TABLE_LED;取低位
XLAT  TABLE_LED
MOV  DX,280H
OUT  DX,AL
```

```
MOV AL,10H;PB4=1， 点亮
MOV DX,281H;
OUT DX,AL
```

```
MOV AL,0H ;熄灭
MOV DX,281H
OUT DX,AL
```

```
MOV AX,NUM2
MOV AL,AH
XLAT TABLE_LED;取高位
MOV DX,280H
OUT DX,AL
```

```
MOV AL,20H;送 PB5=1,点亮
MOV DX,281H
OUT DX,AL
```

```
MOV AL,0H ;熄灭
MOV DX,281H
OUT DX, AL
RET
```

```
LIGHT_LED  ENDP
```

```
CRT  PROC NEAR
MOV AL,NUM1
LEA BX,TABLE_CRT
XLAT TABLE_CRT;取低位
MOV DL,AL
MOV AH,02H
INT 21H
MOV DL,' '
```

```
MOV AH,02H
INT 21H

MOV AX,NUM2
MOV AL,AH
LEA BX,TABLE_CRT;取高位
XLAT TABLE_CRT
MOV DL,AL
MOV AH,02H
INT 21H
MOV AX,NUM2
XLAT TABLE_CRT;取低位
MOV DL,AL
MOV AH,02H
INT 21H
MOV DL,' '
MOV AH,02H
INT 21H
MOV DL,' '
MOV AH,02H
INT 21H
RET
```

CRT ENDP

Init_8253 PROC;

```
MOV DX,28BH ;8253 控制字
MOV AL,00100101B
OUT DX,AL
MOV DX,288H
MOV AL,10H
OUT DX,AL
MOV DX,28BH
MOV AL,01100111B
OUT DX,AL
MOV DX,289H
MOV AL,10H
OUT DX,AL
```

RET

Init_8253 ENDP

Init_8255 PROC

MOV AL,10001001B;控制字 A 出， B 出， C 高入， C 低入

MOV DX,283H

OUT DX,AL

RET

Init_8255 ENDP

WAIT_INSPACE PROC

NEXT:

MOV AH,01H

INT 21H

CMP AL,20H

JNZ NEXT

RET

WAIT_INSPACE ENDP

CODE ENDS

END START