

《微型计算机原理与接口技术》

第5版

第10章

模数转换和数模转换



本章主要内容:

§10.1 概述

§ 10.2 D/A转换器

§ 10.3 A/D转换器



§10.1 概 述

10.1.1 一个实时控制系统

10.1.2 采样、量化和编码

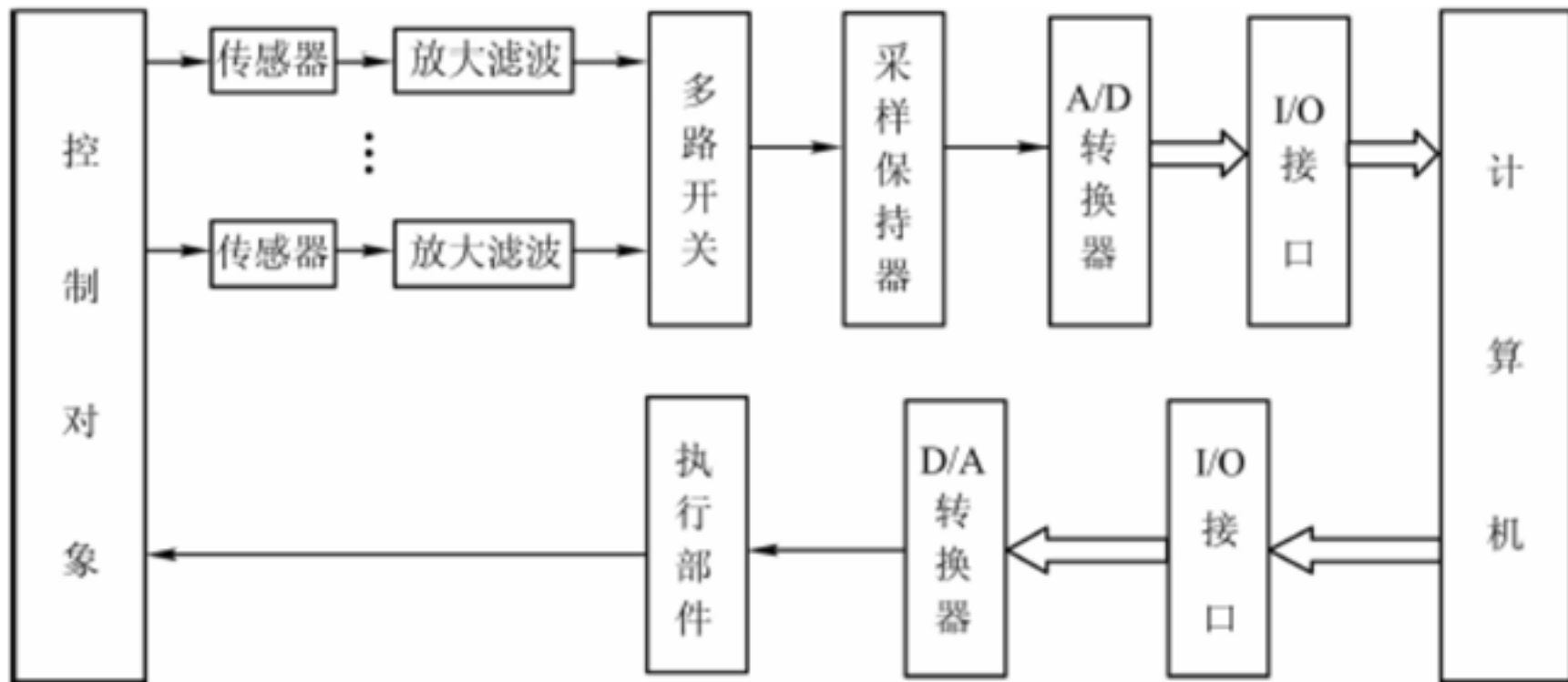
10.1.3 采样保持器



- ❖ 计算机构成数据采集或过程控制系统时，要采集的外部信号或被控制对象的参数，往往是温度、压力、流量、声音和位移等连续变化的模拟量。
- ❖ 模拟量必须由A/D 转换器（Analog to Digital Converter, ADC）变成数字量后，才能送入计算机进行处理。
- ❖ 计算机处理后的结果，要经过D/A 转换器（Digital to Analog Converter, DAC），转换成模拟量后，才能显示、描记或驱动执行部件，达到控制的目的。



10.1.1 一个实时控制系统



一个包含A/D和D/A的实时闭环控制系统

若只有A/D转换通道，便是一个多路数据采集系统

若只包含D/A转换通路，就是一个程序控制系统

模拟量输入通路

- ◆ **传感器：**将压力、流量、速度等非电模拟信号，转换成电流或电压信号。
- ◆ **放大器：**用高输入阻抗运放将 μV 或 mV 量级的模拟信号放大到一定幅度（几V）。
- ◆ **滤波器：**选取信号中一定频率范围内的成分，去掉各种干扰和噪声。
- ◆ **多路模拟开关：**切换多路输入信号，使共用一个ADC，降低成本，减小体积和功耗。也可选用内部有多路开关的ADC，如ADC0809。
- ◆ **采样保持器：**信号较缓慢，可直接加到ADC；如变化较快，还要用采样保持器来提高A/D的正确性。



§10.2 D/A转换器

10.2.1 数/模转换器原理

10.2.2 数/模转换器的主要性能指标

10.2.3 数/模转换器AD7524、 DAC0832和DAC1210



10.2.1 数/模转换器原理

- ◆ D/A转换器(DAC)是把输入数字量转换为与输入量成比例的模拟信号的器件。
- ◆ 多数D/A转换器把数字量（如二进制编码）变成模拟电流，如要转换成模拟电压，还要使用电流/电压转换器（I/V）。
- ◆ 少数DAC内部有I/V变换电路，可直接输出模拟电压值。I/V转换电路由运算放大器构成。
- ◆ D/A转换器原理较简单，大部分ADC内部含有D/A转换电路，因此先学习D/A。



10.2.1 数/模转换器原理

10.2.2 数/模转换器的主要性能指标

10.2.3 数/模转换器DAC 0832



2. 模数转换器DAC0832

1) 性能指标

- ◆ NSC公司（National Semiconductor Corporation）生产的DAC0832，是带有数据输入寄存器的8位DAC，低功耗CMOS工艺，采用R-2R梯形电阻网络实现模数转换，转换结果以差动电流 I_{OUT1} 和 I_{OUT2} 输出。它可直接与8088、8086等微处理器总线相连。

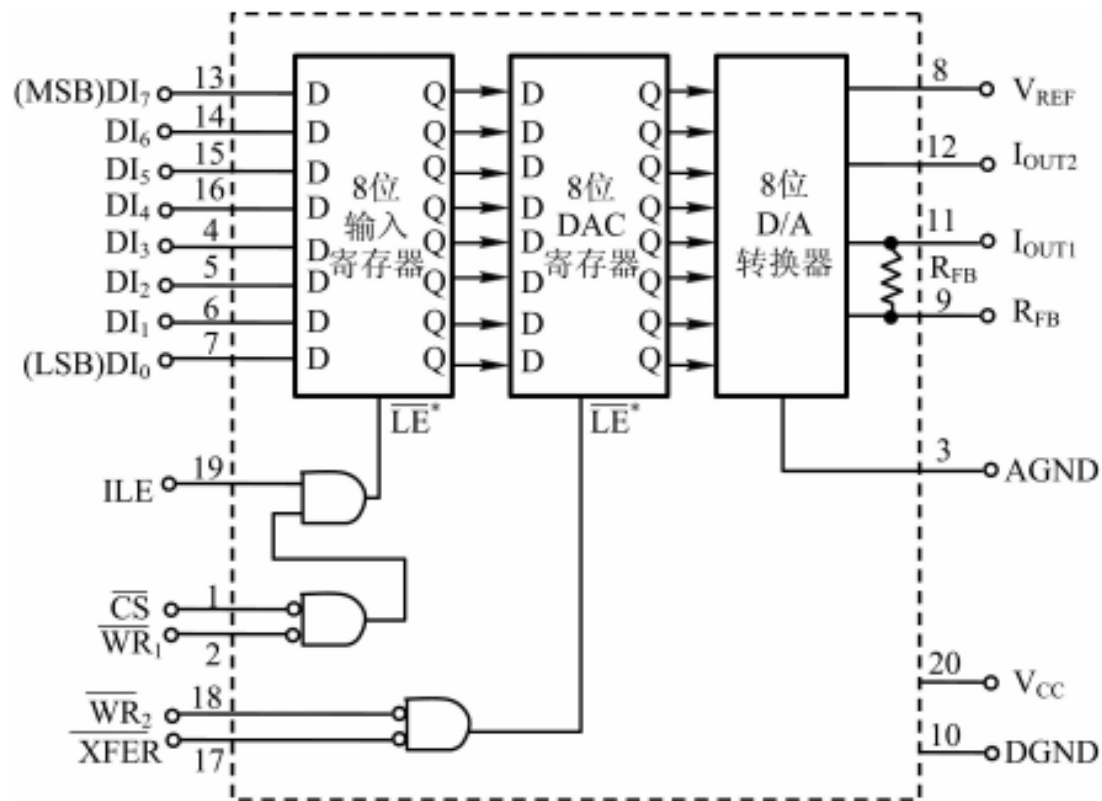
主要参数：

- 分辨率 8位
- 转换时间 $1\mu s$
- 满量程误差 1LSB
- 参考电压 10V
- 单电源 $+5V \sim +15V$

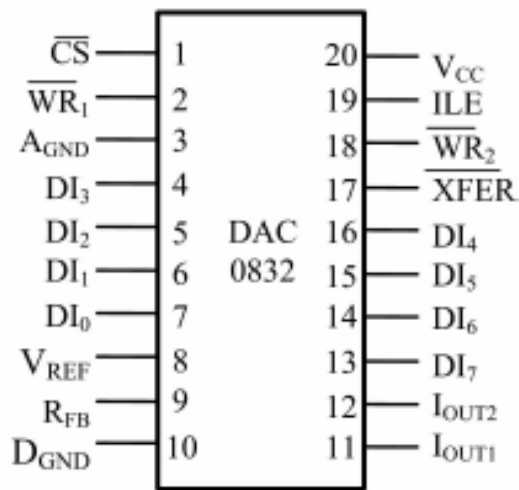


2) 内部结构和引脚功能

20引脚DIP封装。内含1个8位输入寄存器和1个8位DAC寄存器，可分别选通。CPU送来的数据，可先打入输入寄存器，在需要D/A转换时，再选通DAC寄存器，实现转换，即双缓冲工作方式。



(a) DAC0832的内部结构图



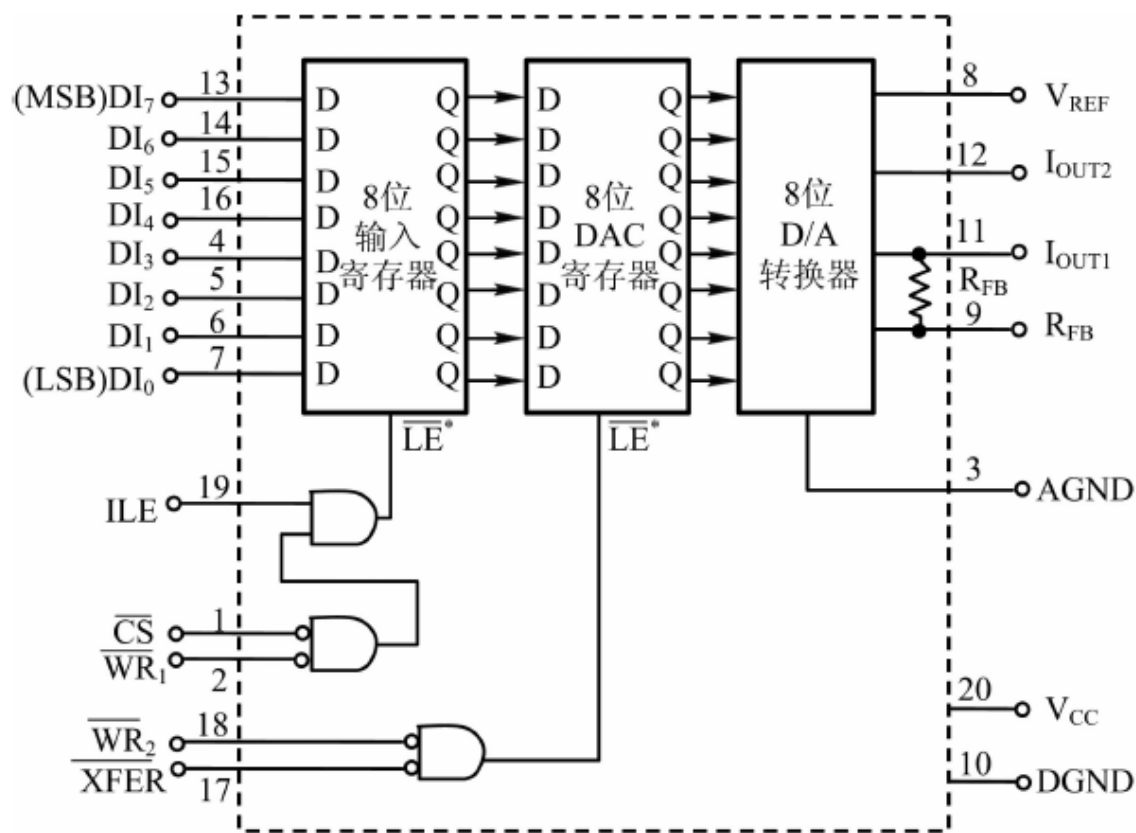
(b) DAC0832的引脚图

各引脚的功能：

- V_{REF} 参考电压输入端。它是转换的基准，要求数值正确，稳定性好。
- V_{CC} 电源电压。
- A_{GND} 模拟地， D_{GND} 数字地。应将电路板上的所有模拟地和数字地各自连在一起，然后连到一个公共接地点，提高系统抗干扰能力。
- $DI_7 \sim DI_0$ 8位数据输入。
- I_{OUT1} 和 I_{OUT2} 互补的电流输出端。 $I_{OUT1} + I_{OUT2} = \text{常数}$
- R_{FB} 片内反馈电阻引脚。
- ILE 输入锁存使能信号输入端，高电平有效。
- \overline{CS} 片选信号输入端。



- **WR1** 和 **WR2** 两个写命令输入。
- **XFER** 传输控制信号输入端，低电平有效。
- 当 $\text{ILE} = 1$, **CS** 和 **WR1** 均为0时，8位数字量可到达输入寄存器；当 **CS** 或 **WR1** 由0变1时，数据被锁存在其输出端。



- 当 **WR2** 和 **XFER** 均为0, 输入寄存器锁存的8位数据进 **WR2** 寄存器。当 **WR2** 或 **XFER** 由0变1时, 该数据被锁存在DAC寄存器输出端, 即加到了D/A转换器, 进行转换。

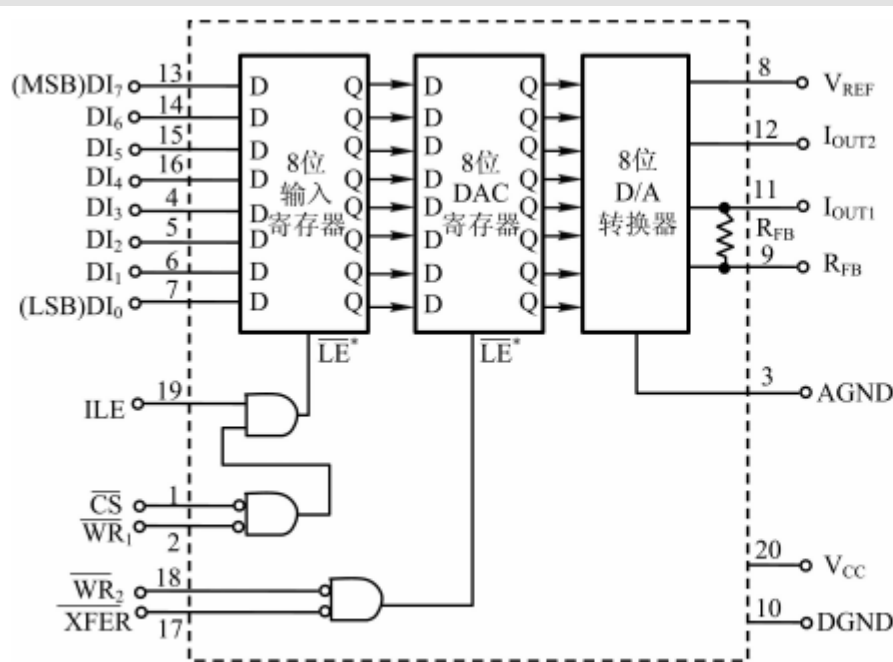
- ◆ 不论DAC寄存器输出的数据是否变化，D/A转换器都不停地转换并输出，就是说D/A转换器的输出是无法被禁止的。所以在控制系统的应用中，应在系统初始化时就将DAC0832设置一个案例状态，以避免执行机构的误操作。



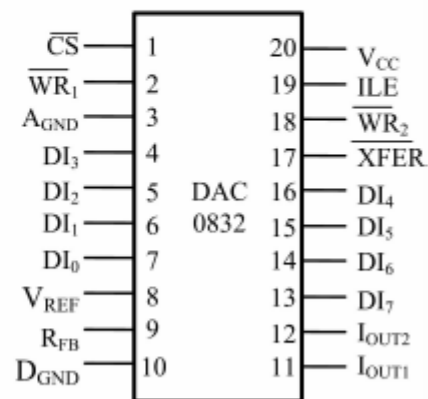
3) 三种工作方式

(1) 直通方式

- ❖ ILE接高电平， $\overline{\text{CS}}$ 、 $\overline{\text{WR}}_1$ 、 $\overline{\text{WR}}_2$ 和 $\overline{\text{XFER}}$ 都接数字地，两个内部寄存器均选通，8位数字量一到达输入端，就立即加到 8位DAC，被转换成模拟量。
- ❖ 这种方式DAC0832始终处于D/A转换状态。这种工作方式下DAC0832不能直接与CPU的数据总线相连，因为这种方式不使用缓冲器，故可以接一片8255作为缓冲



(a) DAC0832的内部结构图



(b) DAC0832的引脚图



3) 三种工作方式

(2) 单缓冲方式-----一个直通、一个受控

- ◇ 把两个寄存器中的任一个接成直通方式，而用另一个锁存数据，就是单缓冲方式。
- 一般将 $\overline{WR_2}$ 和 \overline{XFER} 都接地，使DAC寄存器直通。
- 将 \overline{ILE} 接高电平， \overline{CS} 接I/O译码， $\overline{WR_1}$ 接 \overline{IOW} ，当执行

，将数据锁存到输入寄存器中，实现

【例8.1】如图8-5所示，设DAC0832端口地址为port1，待转换数据在数据段的3000H单元中，完成转换的程序如下：

```
MOV AL, [3000H] ; 取数据
MOV DX, port1    ; 取端口地址
                  ; 址送DX
OUT DX, AL       ; 输出数据
                  ; D / A转换
HLT
```

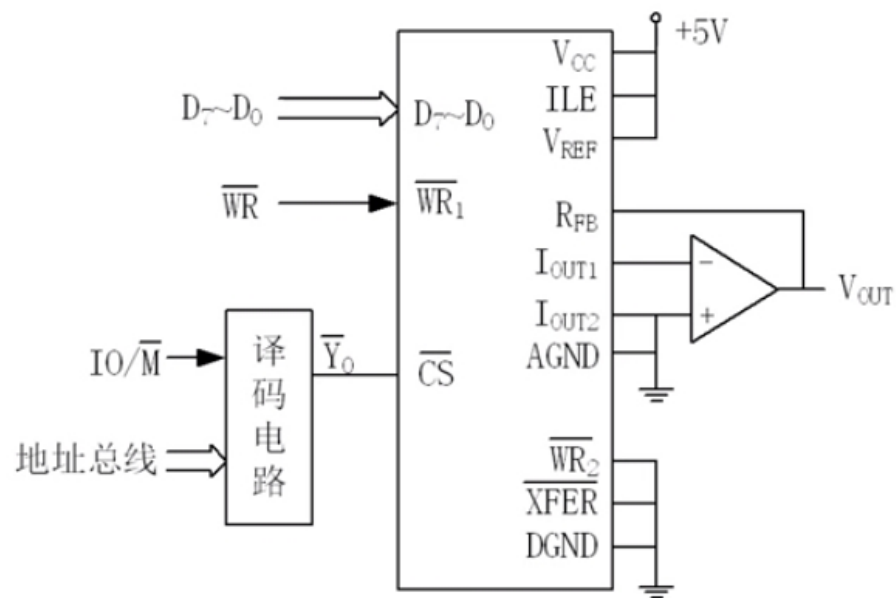


图8-5 DAC0832单缓冲方式下的电路

(3) 双缓冲方式-----2个寄存器均受控

以下两种情况需要用双缓冲式的D/A转换：

一、先把要转换数据打入输入寄存器，再在某个时刻启动D/A转换。这样可对某数据转换的同时，输入下个数据，提高运行速度。

- 可将ILE接高， $\overline{WR1}$ 和 $\overline{WR2}$ 接 \overline{IOW} ， \overline{CS} 和 \overline{XFER} 分别接两个不同的I/O地址译码信号。执行OUT指令时

$\overline{WR1}$ 和 $\overline{WR2}$ 均为0。这样，可先用OUT指令，把数据写入输入寄存器；再执行第二条OUT指令，把输入寄存器中数据写入DAC寄存器，实现D/A转换。

例10.3 要求DAC0832工作于双缓冲方式，与8位CPU相连，试画出硬件连线图，并编写相关的程序。



设计的硬件电路如图。 \overline{CS} 的口地址 320H, \overline{XFER} 的口地址 321H。把1个数据两次锁存, 通过DAC0832输出的程序段:

```
MOV    DX, 320H    ; 指向输入寄存器
MOV    AL, DATA    ; DATA为被转换的数据
OUT    DX, AL       ; 数据打入输入寄存器
INC    DX           ; 指向DAC寄存器
OUT    DX, AL       ; 选通DAC寄存器启动D/A转换
```

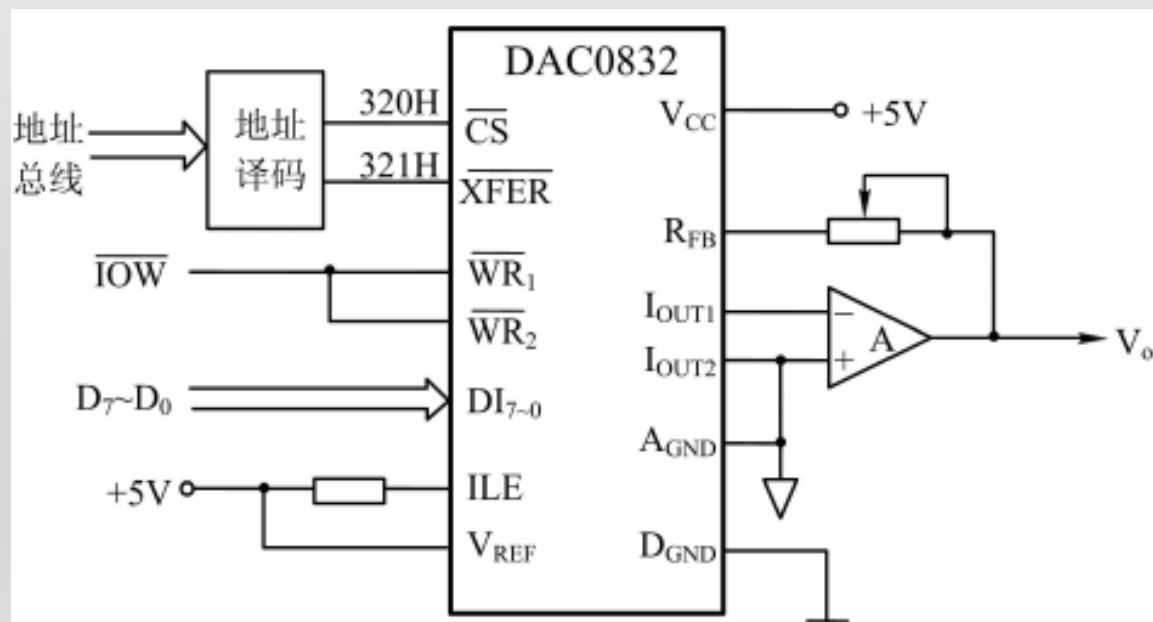


图 10.11 DAC0832 与 8 位数据总线微机的连线图



§10.3 A/D转换器

10.3.1 模/数转换器原理

10.3.2 模/数转换器ADC0809 和AD574A



10.3.1 模/数转换器原理

- ❖ 实现A/D转换的方法有十几种，常采用计数法、逐次逼近法、双斜积分法和并行转换法等。
- ❖ 逐次逼近式A/D转换速度快、分辨率高，芯片成本较低，因此在数据采集系统中广泛应用。仅介绍逐次逼近式A/D转换器的原理和它们的使用。
- ❖ 这类ADC的转换原理以逐次逼近原理为基础，即把输入电压 V_i 与一组从参考电压分层得到的量化电压比较，从最大量化电压开始，由粗到细逐次进行，根据每次比较的结果，确定相应的位是1还是0。不断比较，不断逼近，直到两者的差别小于某一误差范围时，即完成了一次转换。



§10.3 A/D转换器

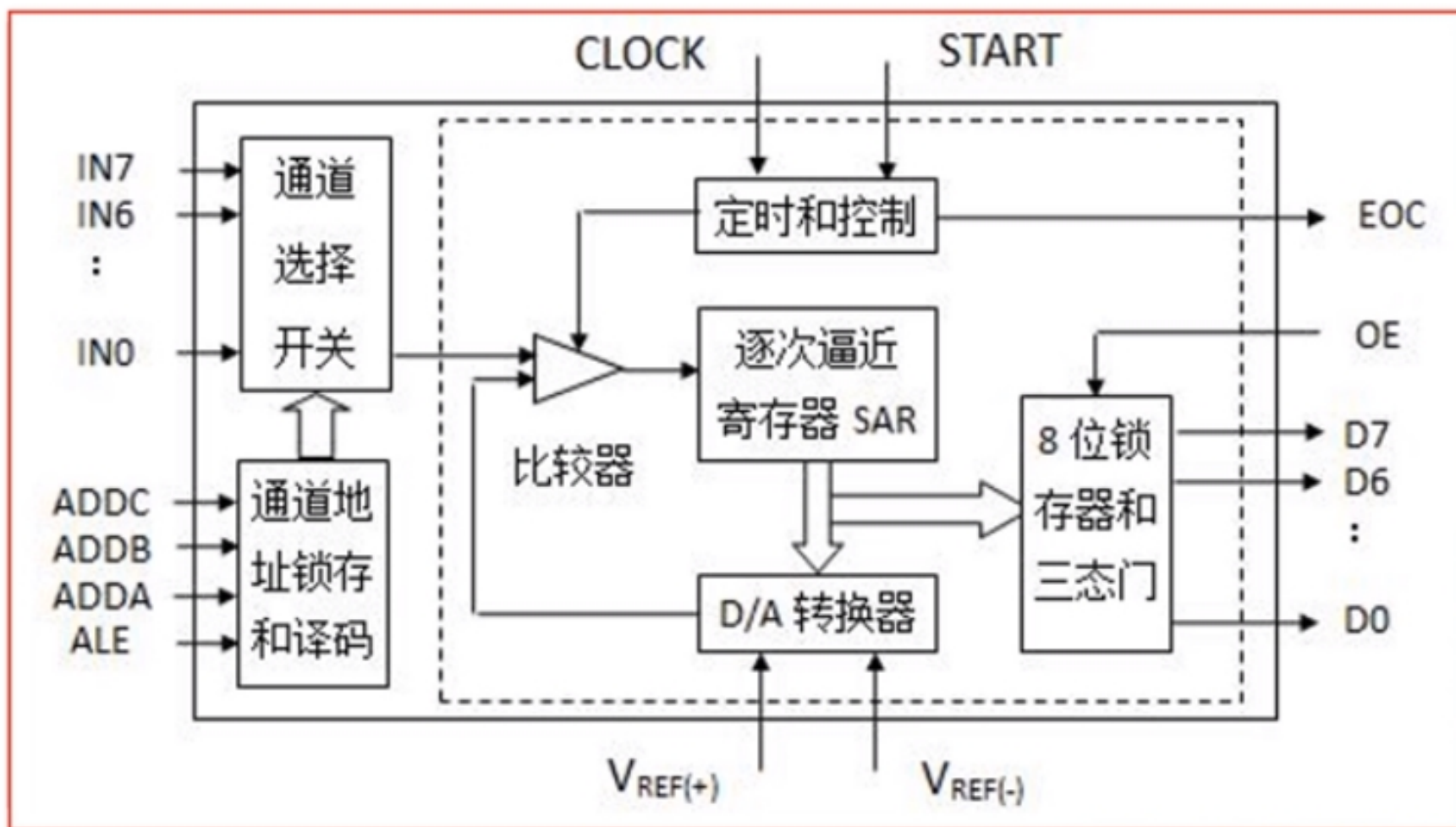
10.3.1 模/数转换器原理

10.3.2 模/数转换器ADC0809



1. 模/数转换器ADC0809

- ADC0809是NSC公司的8通道8位A/D转换器，将多路模拟开关和8位ADC集成在一个芯片内，便于构成多通道数据采集系统。



1. 模/数转换器ADC0809

- ADC0809是NSC公司的8通道8位A/D转换器，将多路模拟开关和8位ADC集成在一个芯片内，便于构成多通道数据采集系统。

1) 引脚

- $IN_7 \sim IN_0$ 8通道模拟量输入端
- $D_7 \sim D_0$ 结果数据输出端
- **START** 启动转换命令输入端
- **EOC** 转换结束指示脚。转换开始后为低电平，转换结束即变回高电平。
- **OE** 输出使能端。高电平打开输出缓冲器三态门，读出数据

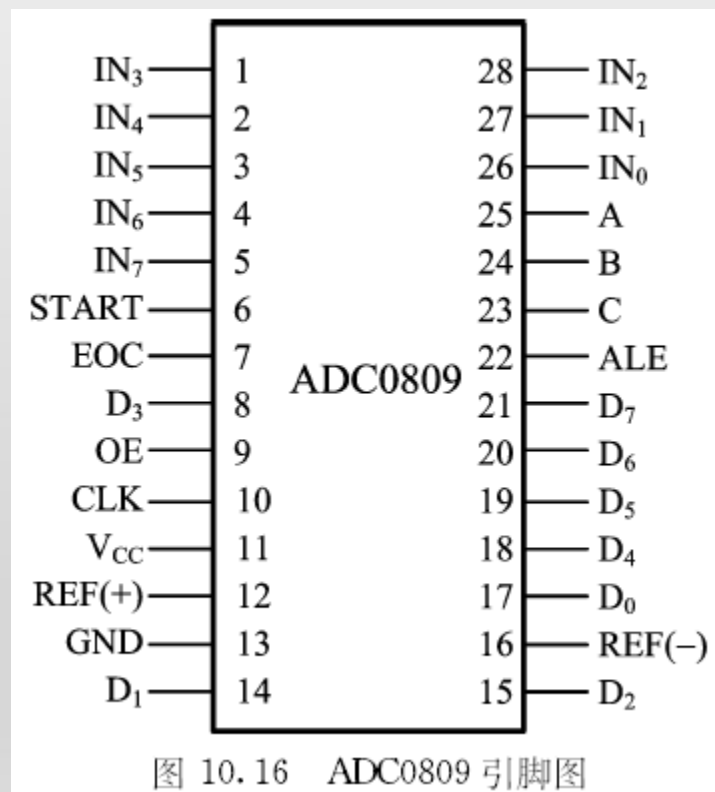


图 10.16 ADC0809 引脚图

1) 引脚

- C、B和A 通道号选择输入端。所加电平的编码为000~111时分别选通模拟输入通道 $IN_0 \sim IN_7$ 。
- ALE 通道号锁存控制端。高电平锁存CBA脚上的通道号选择码，接通相应通道的模拟开关。常把它与START连一起，由启动信号同时锁存通道号。
- CLK 外接时钟接入端。当 $V_{CC}=+5V$ 时，典型时钟640kHz， $t_C=100\mu s$ 。
- REF(+), REF(-) 参考电压输入脚。通常REF(-)接模拟地，若REF(+)=+5V，输入范围为0~+5V。



2) 工作过程

- 图10.17是ADC0809的定时图。

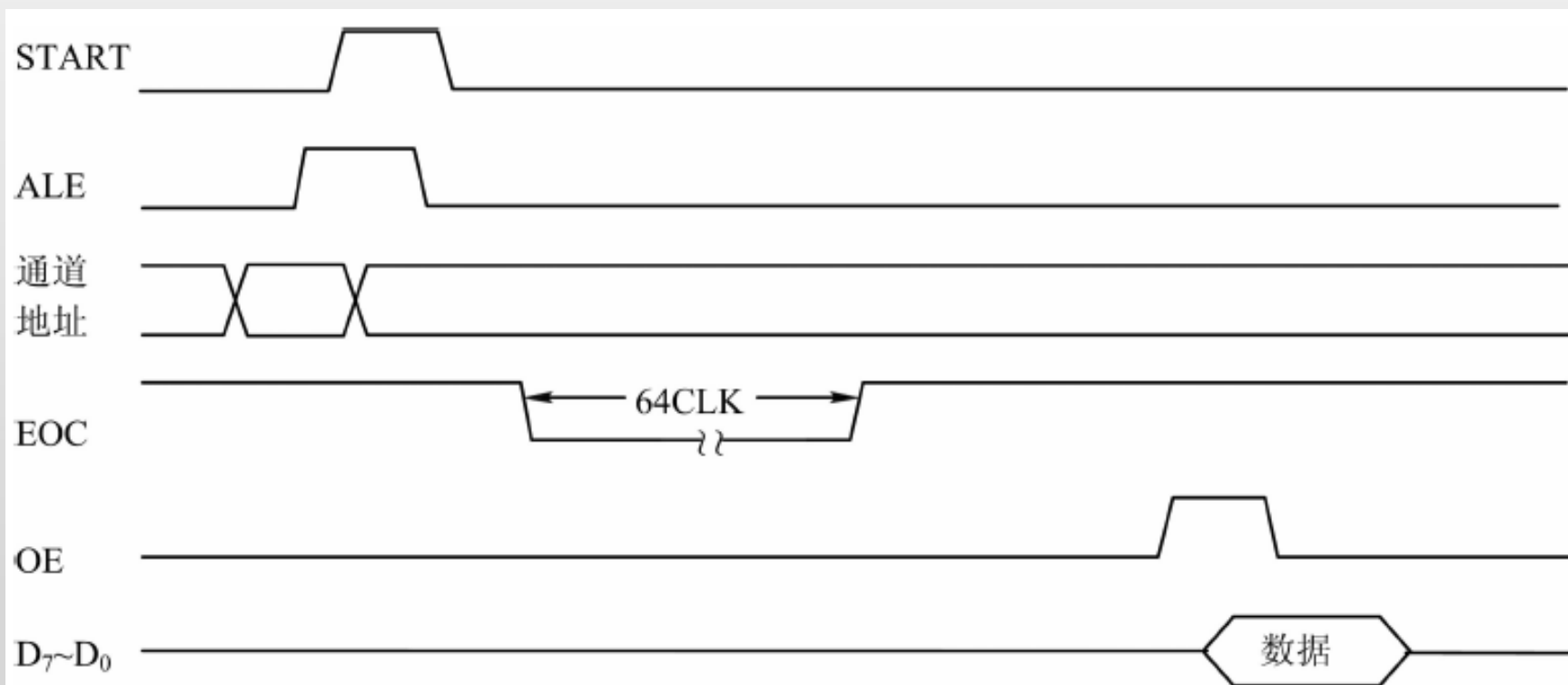


图 10.17 ADC0809 的定时图

- **对指定的通道采集一个数据的过程：**

- (1) 选择当前转换的通道，即将通道号编码送到C、B和A引脚上。
- (2) 在START和ALE脚上加一个正脉冲，将通道选择码锁存并启动A/D转换。可执行OUT指令来产生负脉冲，经反相形成正脉冲，也可由定时电路或可编程定时器提供启动脉冲。
- (3) 转换开始，EOC变低，经64个时钟周期后，转换结束，EOC变高。
- (4) 转换结束后，可通过执行IN指令，设法在OE脚上形成高电平脉冲，打开输出缓冲器三态门，让转换结果出现在数据总线上，并被读入累加器。



- **采样率控制：**
可采用软件延时、定时中断、周期脉冲等方法
- **转换结束判定：**
延时等待、查询EOC电平、用EOC 正跳变请求中断
- **转换通道选择：**
 - 先从数据总线送出通道号，用锁存器将它们锁存在CBA引脚上后，再启动转换。
 - $A_2 \sim A_0$ 不参加I/O译码，而连到CBA脚，执行OUT指令启动各通道转换时，同时将I/O地址中的通道号送出。
- **多通道数据采集：**
若8个通道均接模拟输入，可从通道0开始，启动转换并等转换结束后读取数据；然后启动下通道的转换并读取数据；8个通道全部转换一次称完成了一遍扫描，至少需要8倍的转换时间。



3) 多通道数据采集方案

(1) 定时中断控制采样率，采用地址信号选择通道的方案

- 用ADC0809设计一块插入PC/XT的扩展槽的8通道数据采集卡；
- 以200Hz速率对每个通道均采集1024个数据，也就是5ms对8通道扫描一遍，采来的数据存到数据段中以DBUF为始址的缓冲区中；
- 按通道号次序存放数据，即按通道0→7依次存入各通道的第1个数据，接着存入各通道的第2个数据，直到各通道都存满1024个数据为止。



- 用8253来产生定时脉冲控制采样率
 - 在8253的CLK₀加频率为1MHz时钟脉冲，编程使通道0工作于方式2，时间常数取 $1\text{MHz}/200\text{Hz}=5000$ ，便可从OUT₀输出200Hz的负脉冲序列，即每隔5ms从OUT₀输出1个正跳变脉冲。
- 用中断服务程序实现8个通道的转换
 - 将该脉冲加到PC机上为用户保留的IRQ₂中断请求端，即加到系统板上8259A的IR₂引脚上，在8259A控制下定时向CPU发中断请求，由中断服务程序实现采样。
 - 在中断服务程序中用OUT指令启动通道0的转换，然后查询EOC脚，当EOC=1时转换结束，用IN指令读入结果并存储。接着启动通道1的转换并读取数据，8个通道依次转换结束便完成一次扫描。

数据采集电路

如图，仅画出了ADC0809部分的电路，8253部分电路较简单，未画出。

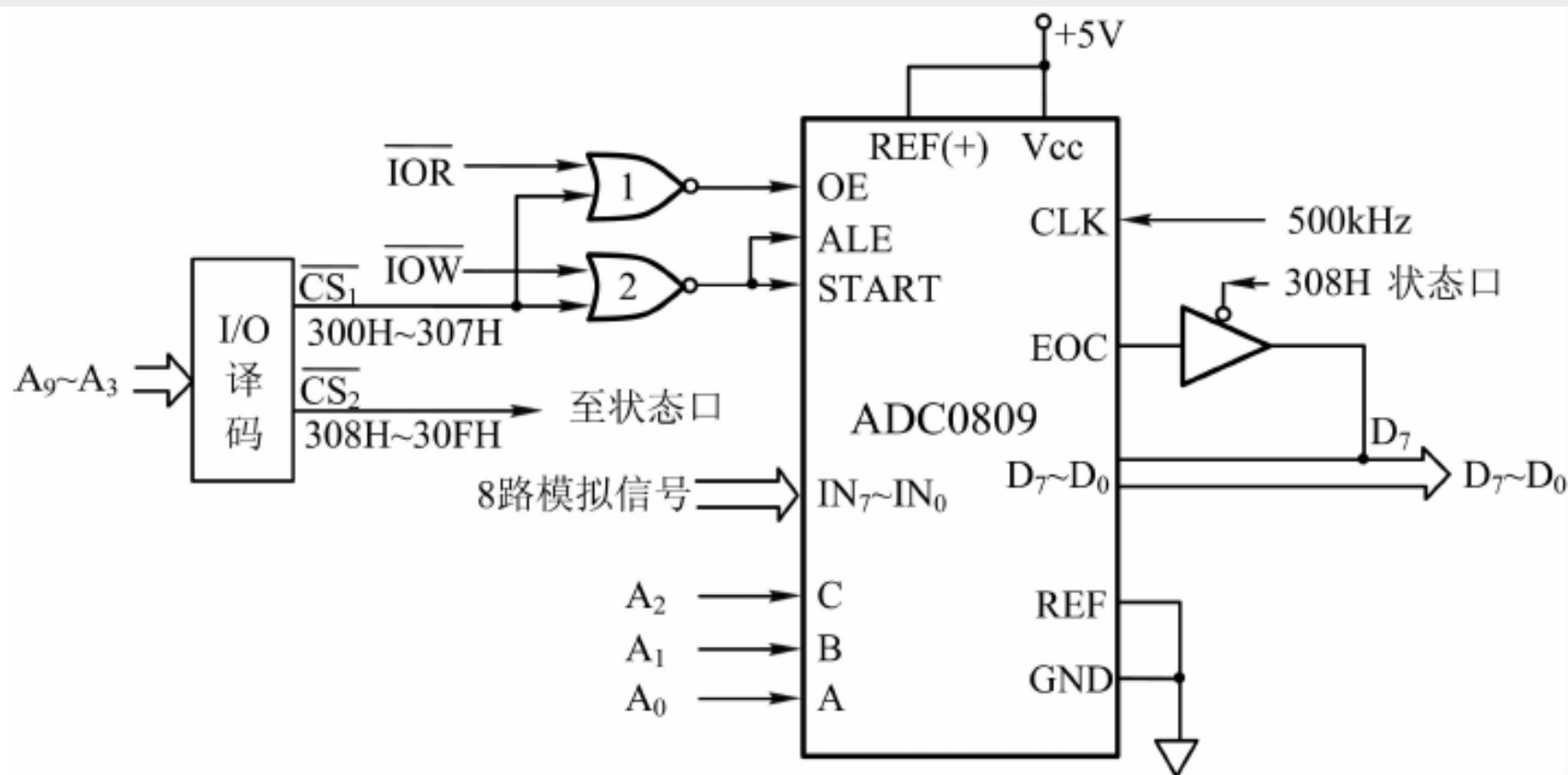


图 10.18 用 ADC0809 设计的多路数据采集电路

■ 数据采集电路

- 地址 $A_9 \sim A_3$ 经I/O译码器形成片选信号 $\overline{CS1}$ 和 $\overline{CS2}$ 。
 $\overline{CS1}$ 选中8个I/O地址300H~307H，地址线 $A_2 \sim A_0$ 接到ADC的CBA脚，每个地址对应1个输入通道。 $\overline{CS2}$ 选中8个I/O地址308H~30FH，用作状态口地址等。
- 接到ADC的时钟信号CLK从系统时钟分频而来，频率500kHz。
- CPU执行OUT指令时，只要端口地址在300H~307H之内， $\overline{CS1}$ 和 \overline{IOW} 更有效，或非门2输出高电平脉冲，加在START和ALE脚上，启动A/D转换，同时还将 $A_2 \sim A_0$ 的编码即通道号锁存，选定待转换的输入通道。
- EOC脚接 D_7 ，构成状态口，地址308H。发出启动脉冲后，查EOC=0？是，已开始转换；再查EOC=1？是，转换已结束。用IN指令读取结果。



- PC/XT 机中8259A口地址为20H/21H，设数据采集卡上8253通道0 和控制口地址为318H和31BH。
- 数据采集程序：

```
DATA          SEGMENT                ; 数据段
DBUFDB 8*1024 DUP (?)                ; 数据区 (8×1024字节)
DATA          ENDS
...                                     ; 堆栈段
```

； 数据采集子程序

```
CODE SEGMENT                ; 代码段
      ASSUME  CS: CODE, DS: DATA
```



AD 8 **PROC** FAR

MOV AX, DATA

MOV DS, AX

; DS指向数据区段址

CLI

; 禁止中断

CLD

; 清方向标志

; 设置0AH号中断矢量, 段址和偏移量ES:DI=0000:
4*0AH

MOV AX, 0

MOV ES, AX

; ES=中断矢量表段址0000

MOV DI, 4*0AH

; DI=中断IR2的偏移地址

MOV AX, OFFSET ADINT

; AX=中断服务子程序偏

址

STOSW

; 放入中断矢量表

MOV AX, SEG ADINT

; 取中断矢量段地址

STOSW

; 放入中断矢量表中



； 对8253进行初始化编程,使通道0的控制字为:方式2,先读写低字节,BCD计数；定时常数n=5000。

```
MOV     DX, 31BH      ; 8253控制口
MOV     AL, 00110101B ; 通道0控制字
OUT     DX, AL        ; 输出控制字
MOV     DX, 318H      ; 8253通道0
MOV     AX, 5000H      ; 时间常数n
OUT     DX, AL        ; 先送低8位
MOV     AL, AH
OUT     DX, AL        ; 后送高8位
```

； 设置屏蔽字,仅允许8259A的IR2和键盘中断,其余禁止

```
MOV     AL, 11111001B ; 屏蔽字
OUT     21H, AL       ; 输出到屏蔽寄存器
```



； 设置数据缓冲区始址到SI中, 计数初值到BX中, 等待中断。每通道采完1024个数据后结束中断。

MOV SI, OFFSET DBUF ; SI=数据缓冲区始址
MOV BX, 1024 ; BX=数据计数器初值
STI ; 开中断, 等待中断

AGAIN:

CMP BX, 0 ; 中断一次BX-1, BX=0?
JNZ AGAIN ; BX≠0, 未采完, 循环等待中断

MOV AL, 1111101B ; 采完, 禁止IR2中断

OUT 21H, AL

MOV AH, 4CH ; 退出中断

INT 21H

RET ; 从子程序返回

AD_8 ENDP ; AD_8过程结束



； 中断服务程序，每通道均采集1个数据，存进DBUF

ADINT **PROC** NEAR

MOV CX, 0008H ; 设置通道计数器初值

MOV DX, 300H ; DX指向ADC通道0

NEXT:

OUT DX, AL ; 启动一次转换

PUSH DX ; 保存通道号

MOV DX, 308H ; DX指向状态口308H

POLL:

IN AL, DX ; 读入EOC状态

TEST AL, 80H ; EOC (D7) =0?即开

始转换了?

JNZ POLL ; 非0，循环等待

NO_END:

IN AL, DX ; EOC=0，已开始转换

TEST AL, 80H ; 再查EOC是否为1

JZ NO_END ; EOC=0，等待转换结束



POP	DX	;	EOC=1, 恢复通道地址
IN	AL, DX	;	读取结果
MOV	[SI], AL	;	存储到缓冲区中
INC	DX	;	DX指向下个通道
INC	SI	;	指向下个缓存单元
LOOP	NEXT	;	通道计数器-1, ≠0则循环
DEC	BX	;	=0, 缓冲数据计数器
	+1		
MOV	AL, 20H		
OUT	20H, AL		
STI		;	开中断
IRET		;	自中断返回
ADINT	ENDP	;	中断服务程序结束
CODE	ENDS		
	END		



(2) 用8255A控制ADC0809的方法

- ◆ 用8255A控制ADC0809的方案图如图10.19。采用查询法来检测转换是否结束。
- 8255A的A口编程为方式0输入，C口高4位为输入，低4位为输出。
- ADC的输出接到A口，8位数据从A口输入。
- $PC_2 \sim PC_0$ 输出3位通道号地址。
- PC_3 与ADC的START和ALE相连接，编程使 PC_3 发启动信号，并锁存通道号。
- PC_7 与EOC输出相连，查询 PC_7 状态可了解转换进程。
- EOC还与ADC的OE输入相连，转换结束时OE也会变高，使ADC的输出缓冲器打开，数据出现在A口上，由IN指令读入CPU。



- ◆ 设图10.19中，8255A的端口地址为320H~323H，已对它进行了初始化，并将ES和DS置成了相同段基地址。要求把8通道的转换结果，存到段基址为ES，偏移量从DATA_BUF开始的内存中。则ADC0809完成一次8路模拟量采集的子程序AD_SUB如下：

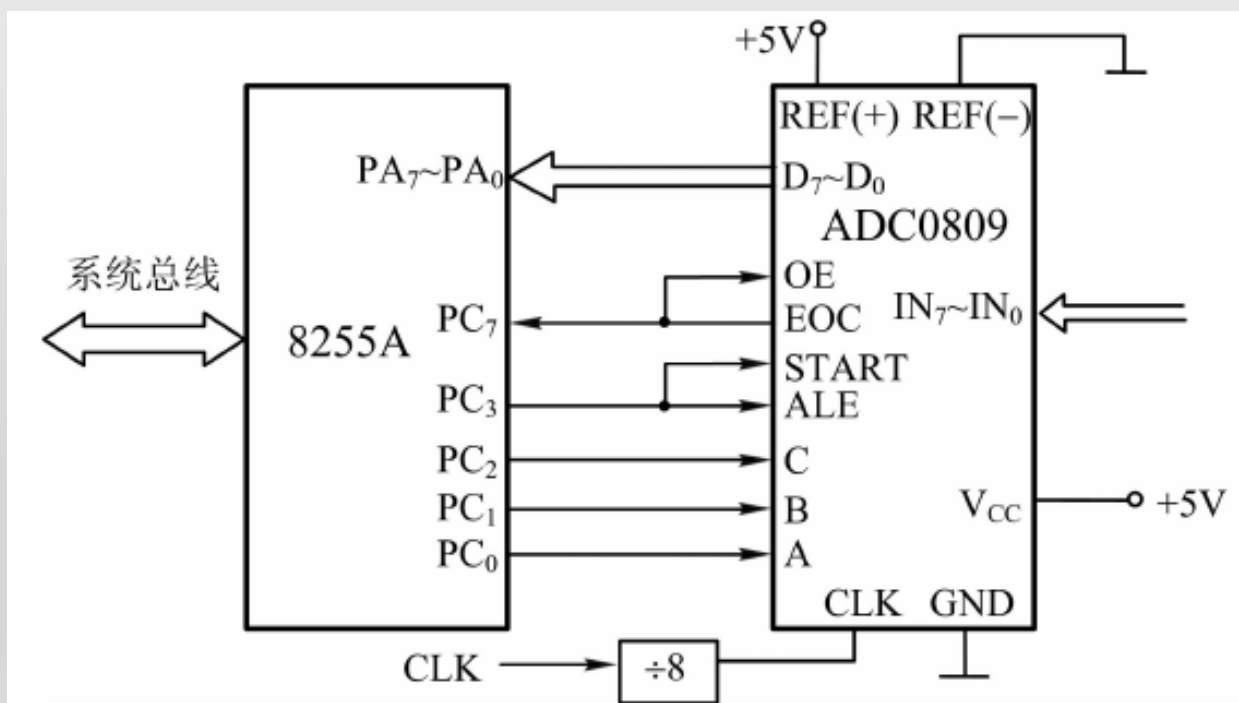


图 10.19 用 8255A 控制 ADC0809 的电路图

AD SUB **PROC** NEAR

MOV CX, 8

; CX作数据计数器

CLD

; 清方向标志

MOV BL, 00H

; 模拟通道号存在

BL中

LEA DI, DATA_BUF

; 缓冲区偏移地址

NEXT_IN:

MOV DX, 322H

; C口地址

MOV AL, BL

OUT DX, AL

; 输出通道号

MOV DX, 323H

; 指向控制口

MOV AL, 00000111B

; PC₃置1

OUT DX, AL

; 送出START信

号

NOP

; 延时

NOP

; 高电平保持一段时间

NOP



MOV	AL, 00000110B	;	PC ₃ =0
OUT	DX, AL	;	使START=0, 结束启动信号
MOV	DX, 322H	;	DX指向C口
NO_CONV:			
IN	AL, DX	;	读入C口内容
TEST	AL, 80H	;	查PC ₇ , 即EOC信号
JNZ	NO_CONV	;	PC ₇ =1, 还未开始转换,
			等待
NO_EOC:			
IN	AL, DX	;	PC ₇ =0, 已启动转换
TEST	AL, 80H	;	再查PC ₇
JZ	NO-EOC	;	PC ₇ =0, 转换未结束,
			等待
MOV	DX, 320H	;	PC ₇ =1, 转换结束, 指向A口
IN	AL, DX	;	读入数据
STOS	DATA_BUF	;	存入ES段的数据缓冲区
INC	BL	;	指向下个通道
LOOP	NEXT_IN	;	尚未完成8路转换则循环
RET		;	已完成, 返回
AD_SUB	ENDP		

