

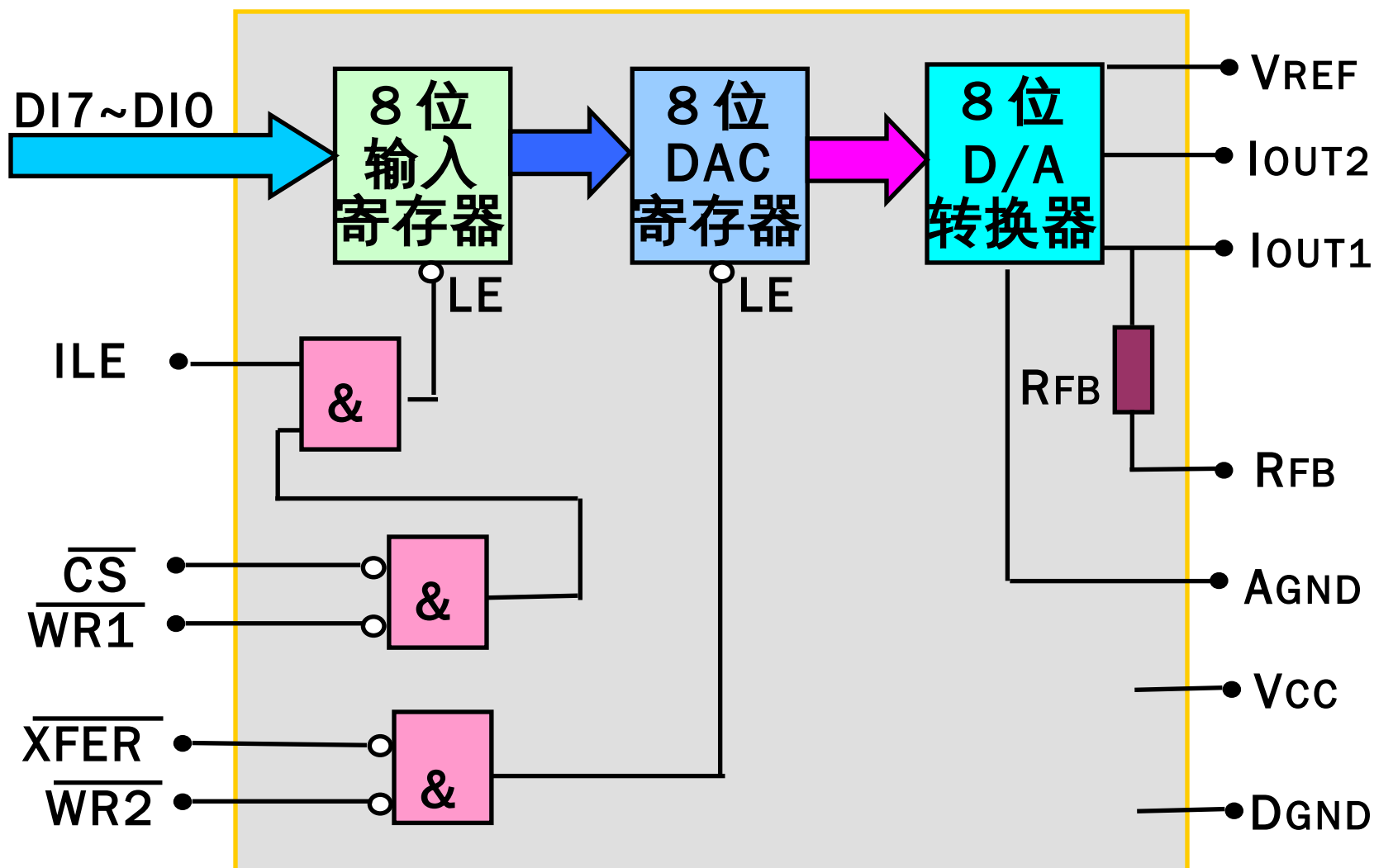
## 四、典型 D/A 转换 DAC0832 芯片

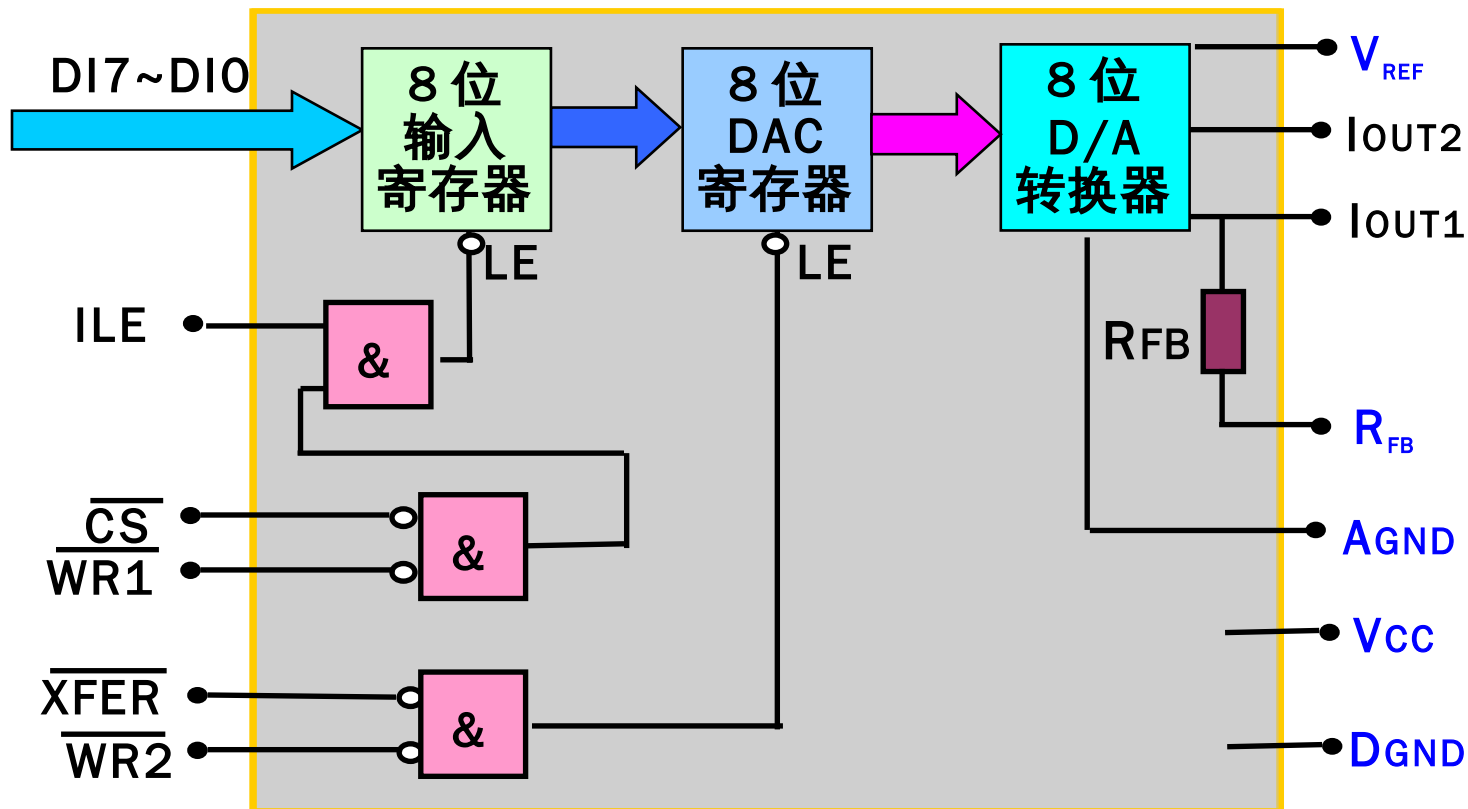
8 位并行、中速（建立时间  $1\mu\text{s}$ ）、电流型、低廉（10~20 元）

- ① 引脚和逻辑结构
- ② DAC0832 与微机系统的连接
- ③ 应用举例

# 1. 引脚和逻辑结构

20 个引脚、双列直插





$V_{CC}$

芯片电源电压，

$+5V \sim +15V$

$V_{REF}$

参考电压，

$-10V \sim +10V$

$R_{FB}$   
端

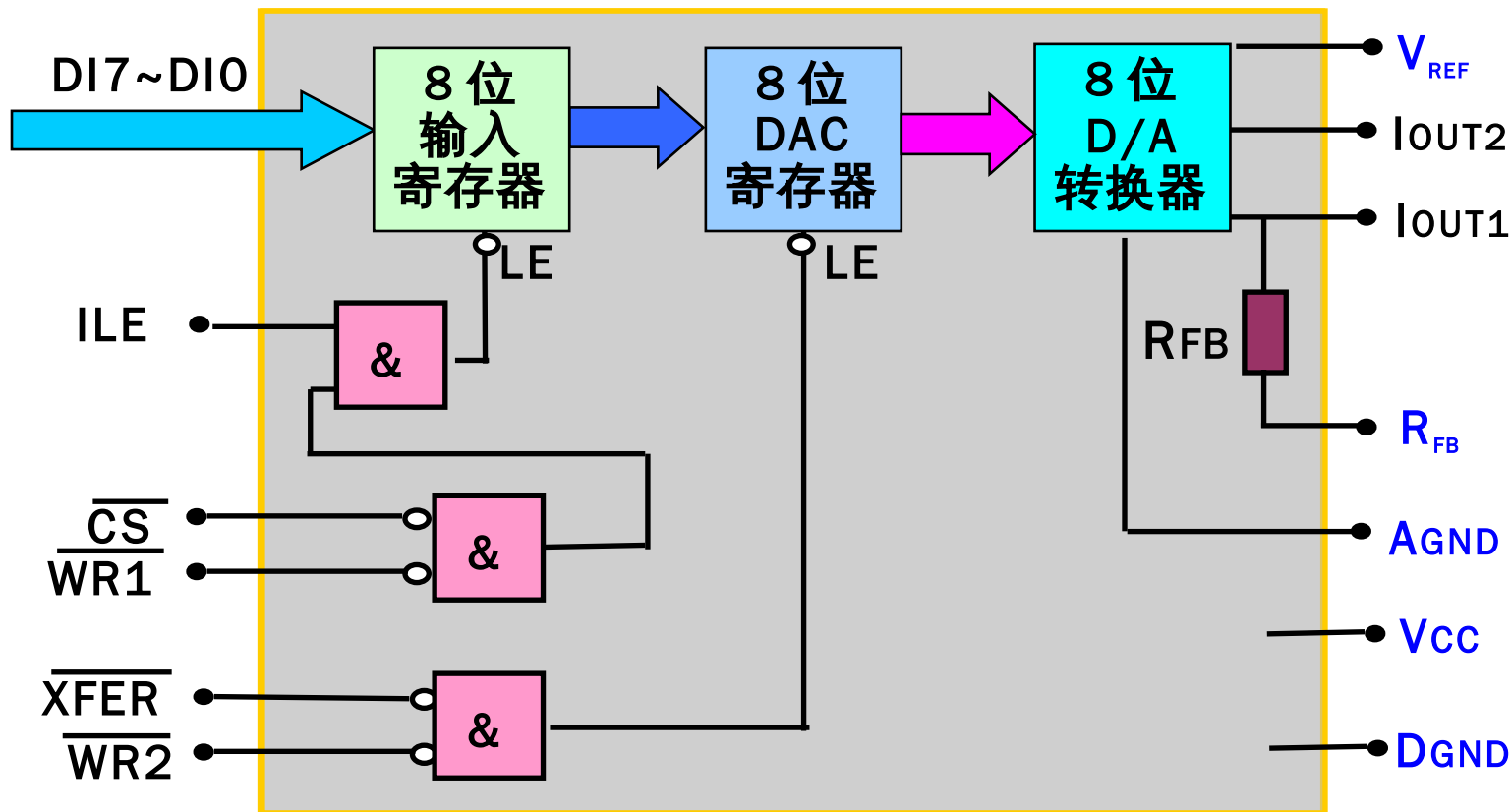
反馈电阻引出端，此端可接运算放大器输出

$AGND$

模拟信号地

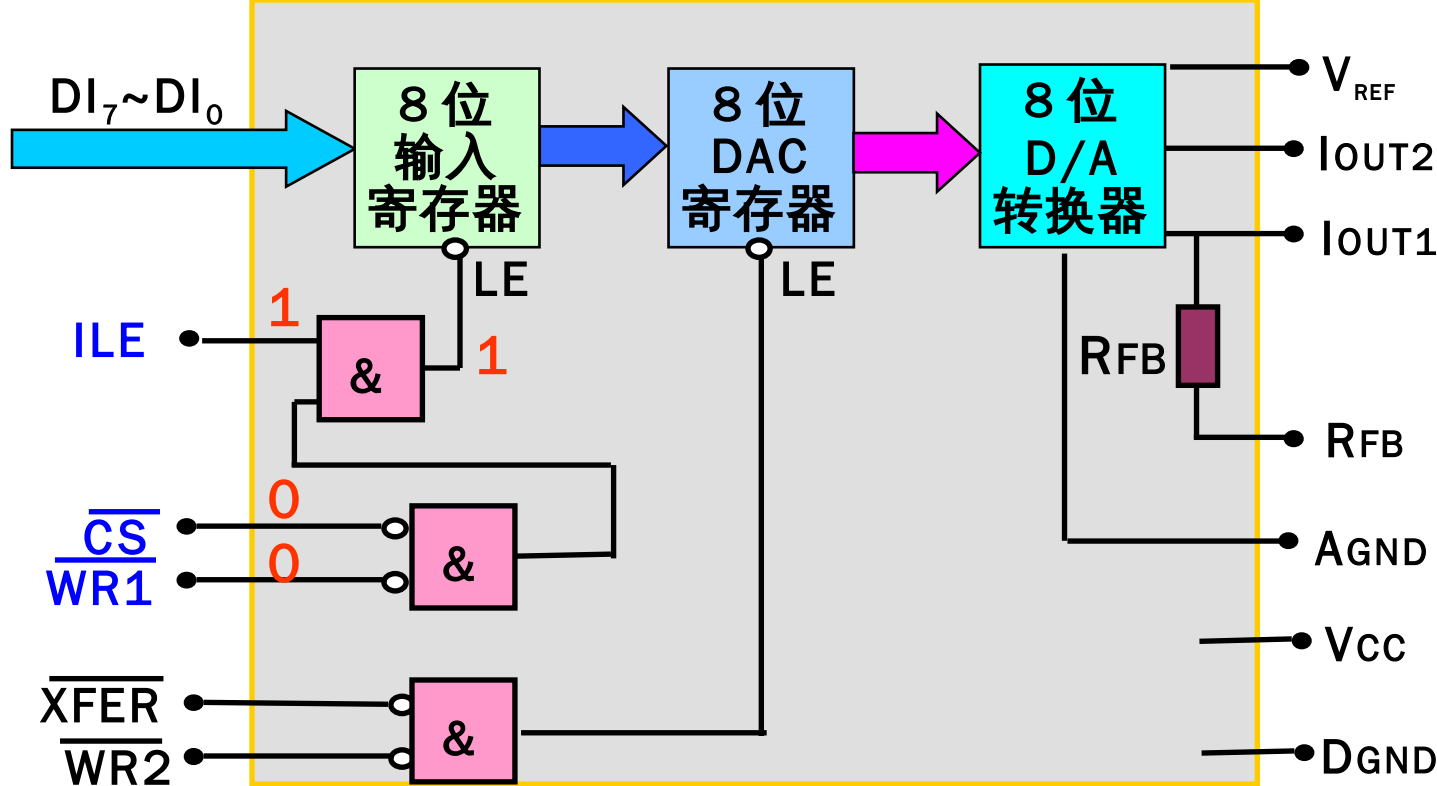
$DGND$

数字信号地



$DI_7 \sim DI_0$       数字量输入信号


其中：  $DI_0$  为最低位，  $DI_7$  为最高位



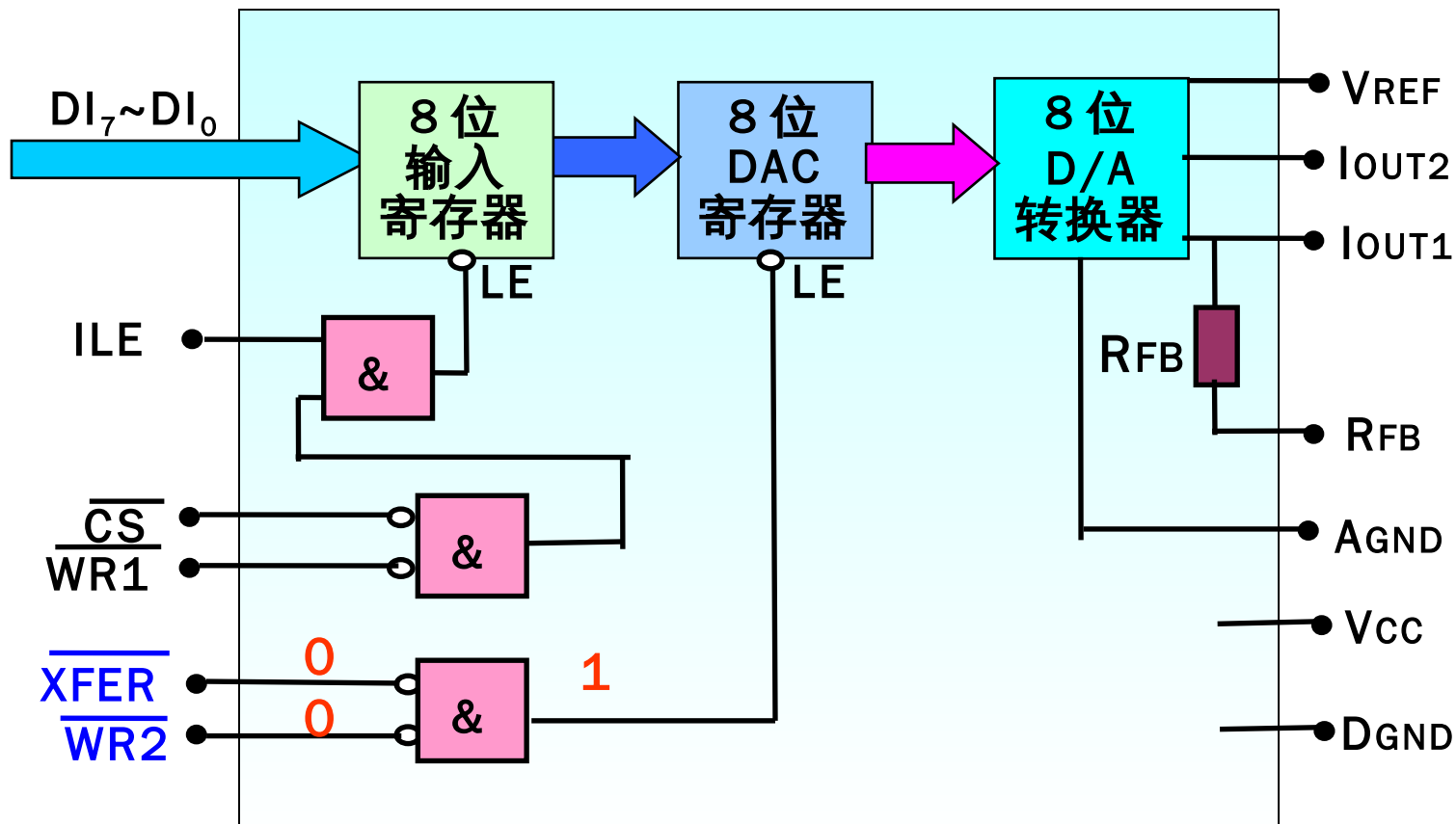
$ILE$  输入锁存允许信号，高电平有效  
 $\overline{CS}$  片选信号，低电平有效  
 $\overline{WR1}$  写信号 1，低电平有效

当  $ILE$ 、 $\overline{CS}$ 、 $\overline{WR1}$  同时有效时， $LE=1$   $LE=1$

输入寄存器的输出随输入而变化

$\overline{WR1}$  ， $LE=0$ ，将输入数据锁存到输入寄存器



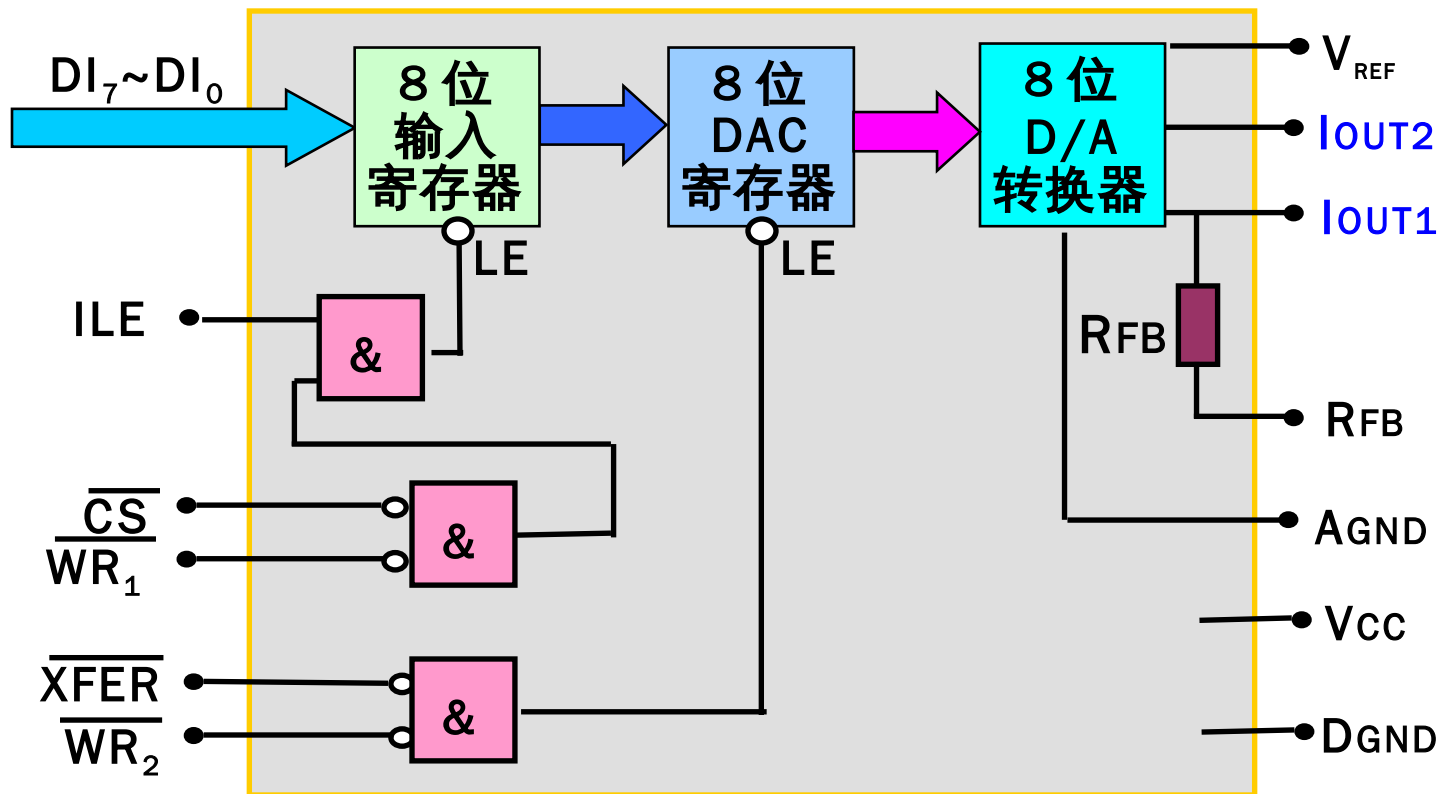


XFER 转移控制信号，低电平有效

WR2 写信号 2，低电平有效

- 当 XFER、WR2 同时有效时，LE2=1，DAC 寄存器输出随输入而变化；
- WR1↑，LE=0，将输入数据锁存到 DAC 寄存器，





**$I_{OUT1}$  模拟电流输出端 1**

当输入数字为全“1”时，输出电流最大，约为： $\frac{255V_{REF}}{256R_{FB}}$   
 全“0”时，输出电流为 0

**$I_{OUT2}$  模拟电流输出端 2**

$I_{OUT1} + I_{OUT2} = \text{常数}$

## 2. DAC0832 与微机系统的 连接

### 1) 单缓冲工作方式

一个寄存器工作于直通状态，  
另一个工作于受控锁存器状态

### 2) 双缓冲工作方式

两个寄存器均工作于受控锁存器状态

,



## 1) 单缓冲工作方式：

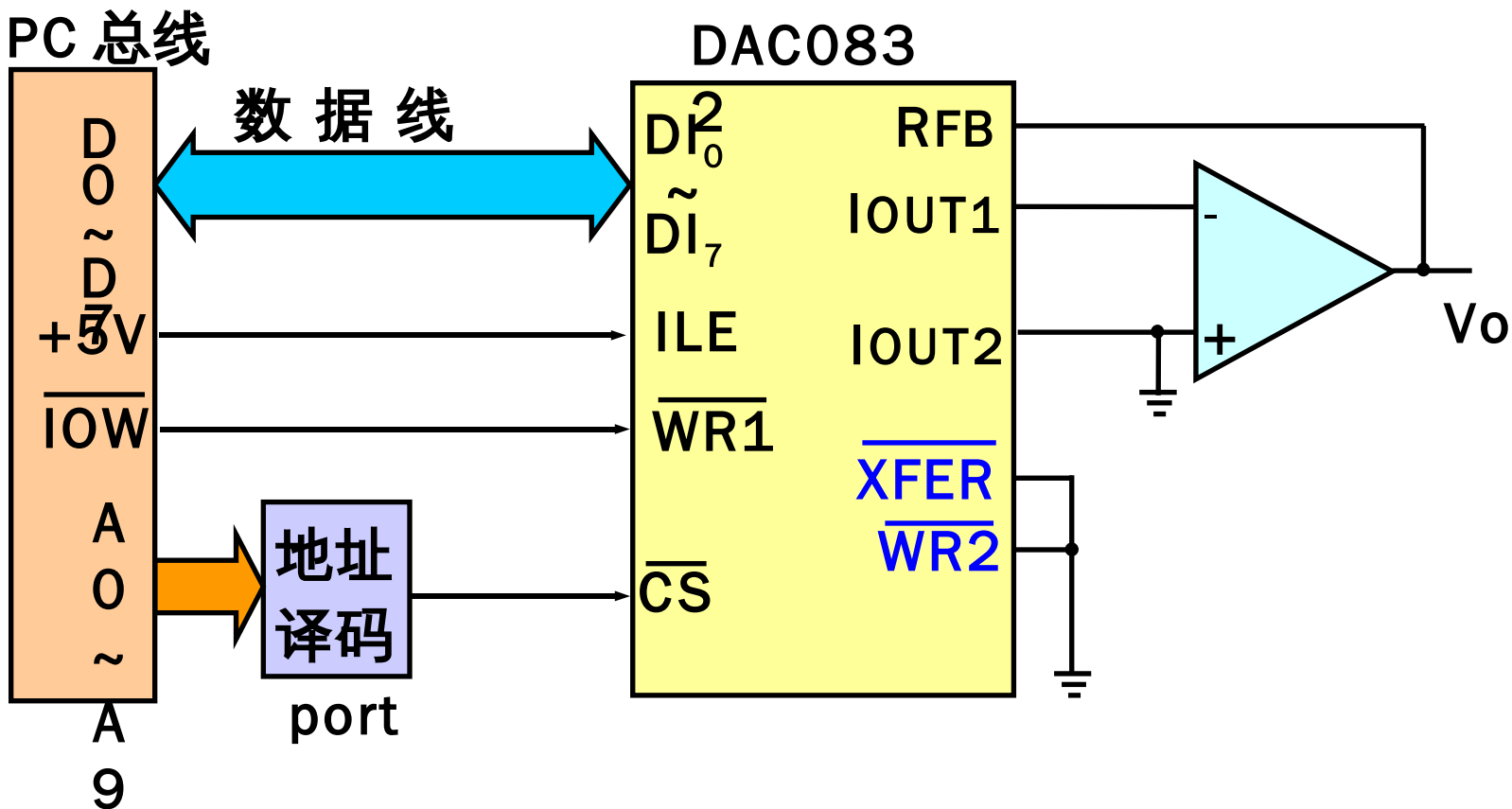
一个寄存器工作于直通状态，

一个工作于受控锁存器状态

在不要求多相 D/A 同时输出时，可以采用单缓冲方式，

此时只需一次写操作，就开始转换，

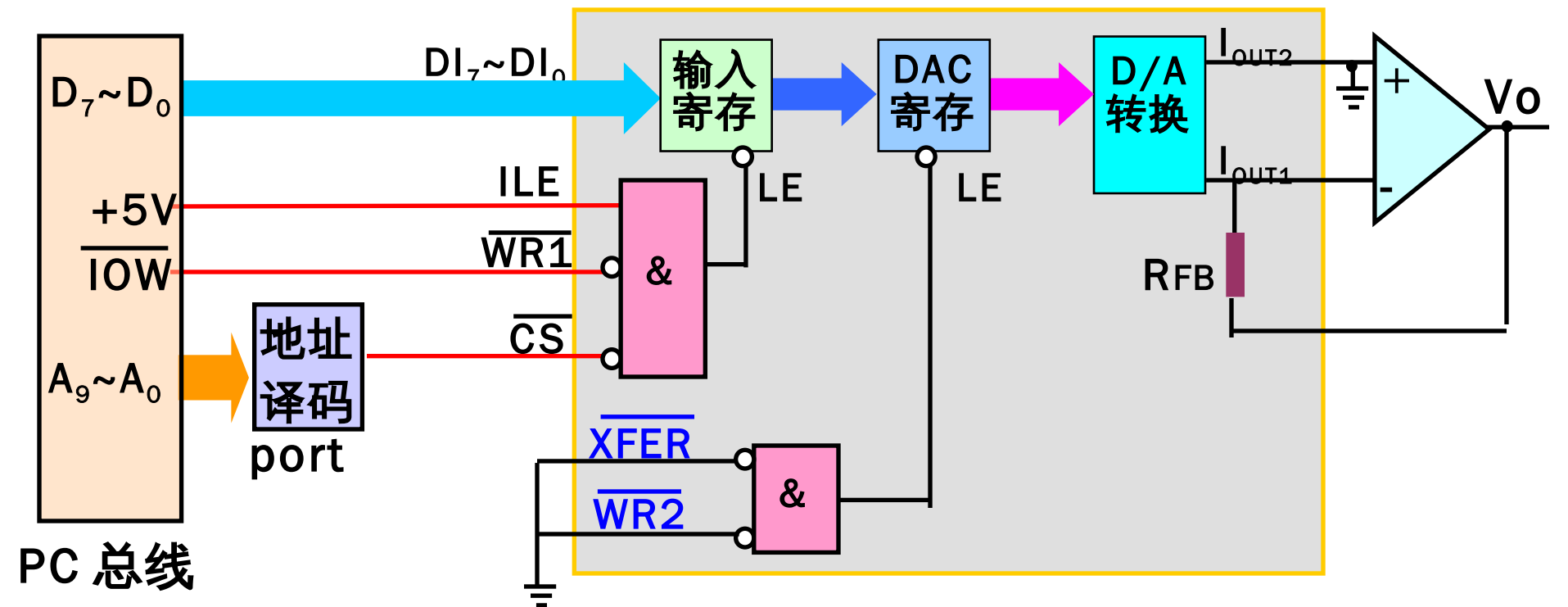
可以提高 D/A 的数据吞吐量。



单缓冲工作方式：

输入寄存器工作于受控状态

DAC 寄存器工作于直通状态

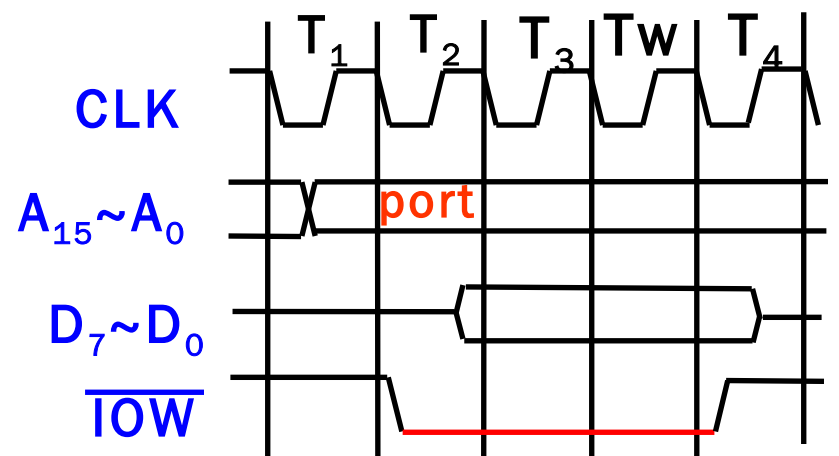


转换一个数据的程序段：

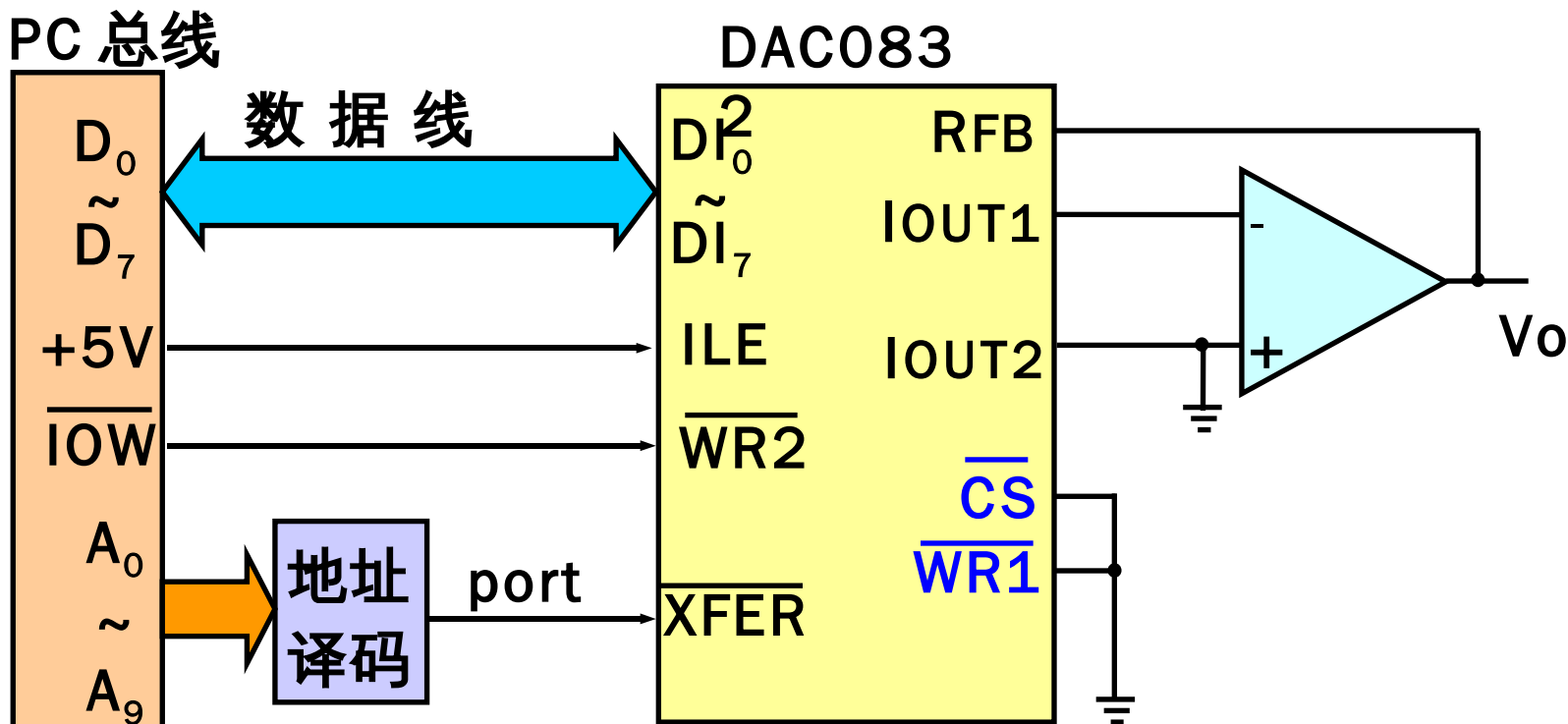
MOV AL, data ; 取数字量

MOV DX, port

OUT DX, AL



PC 序 总线 I/O 写时

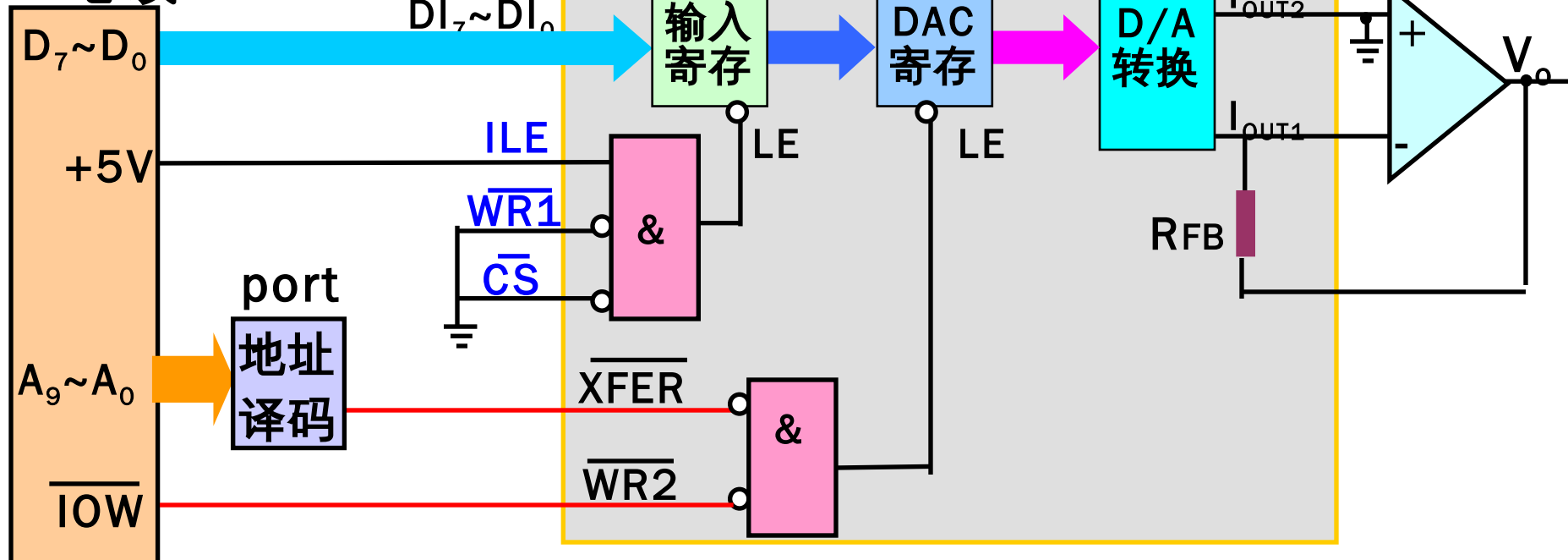


单缓冲工作方式：

输入寄存器工作于直通状态

DAC 寄存器工作于受控状态

PC 总线



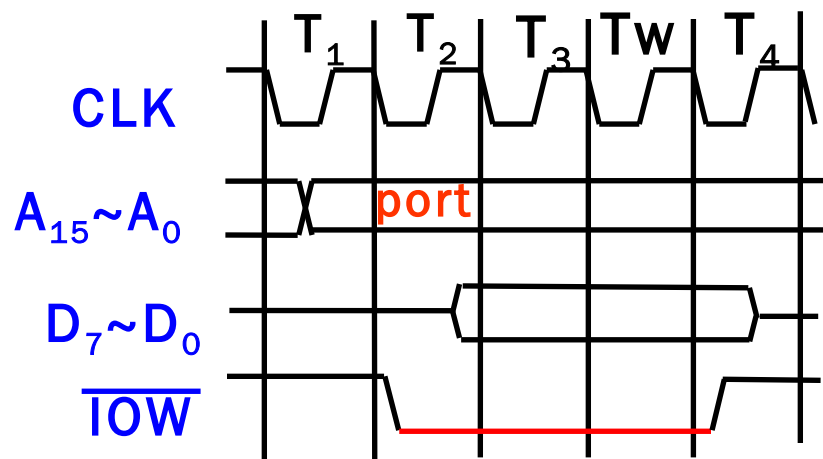
转换一个数据的程序段：

MOV AL, data ; 取数字

量

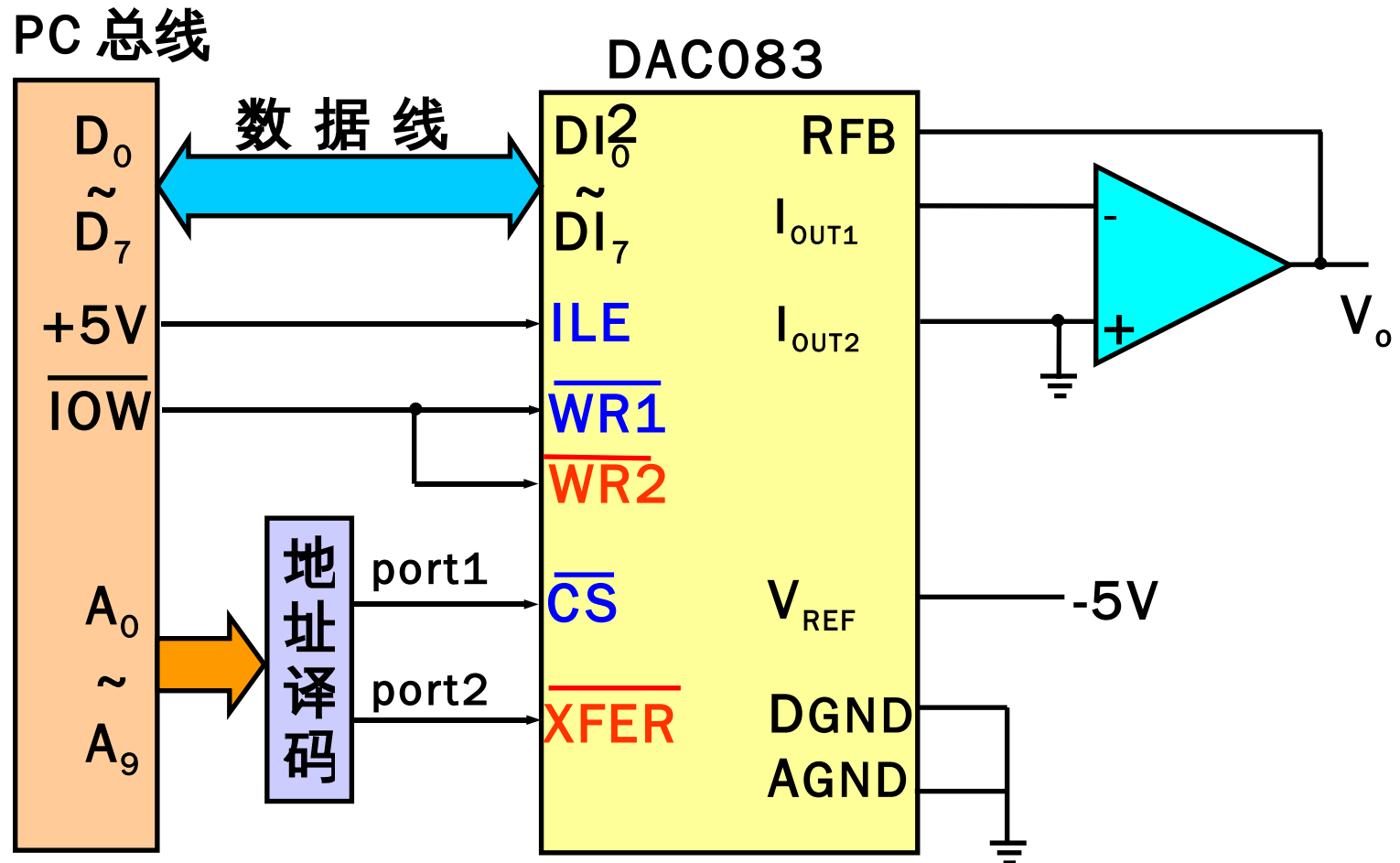
MOV DX, port

OUT DX, AL

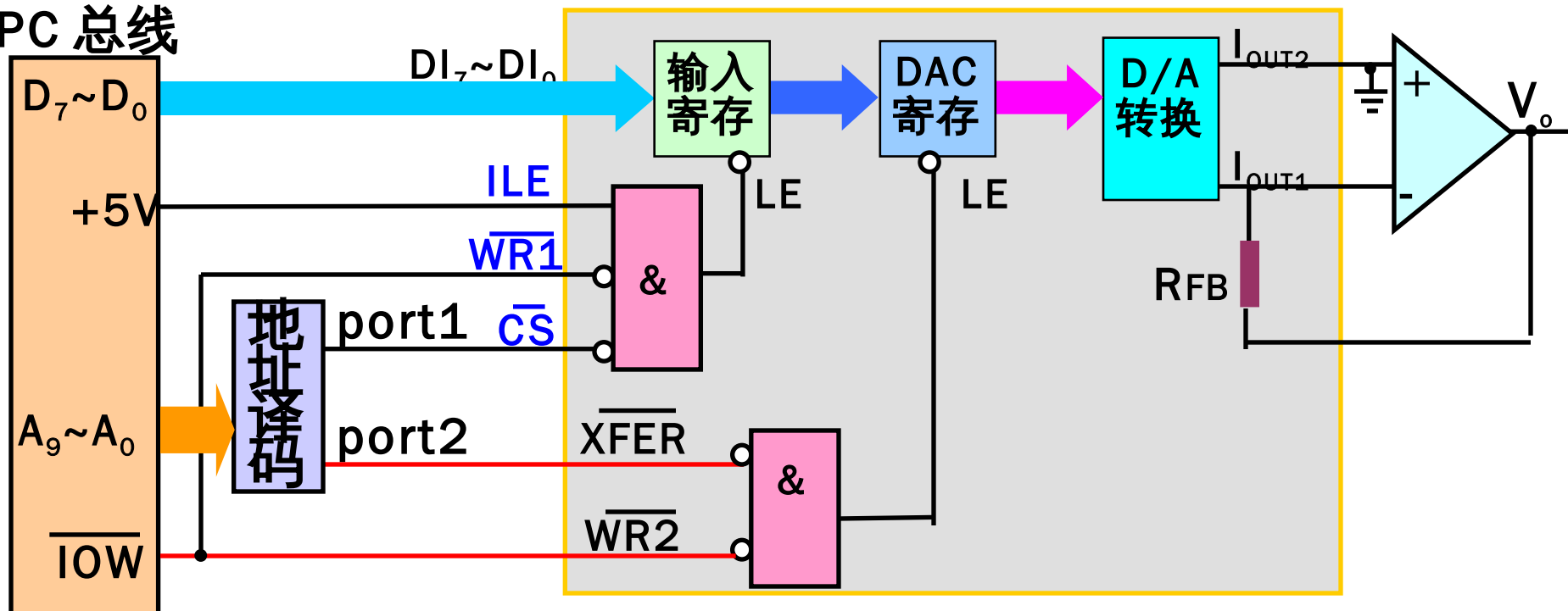


PC 总线 I/O 写时序

## 2) 双缓冲工作方式：两个寄存器均工作于受控锁存器状态



## PC 总线

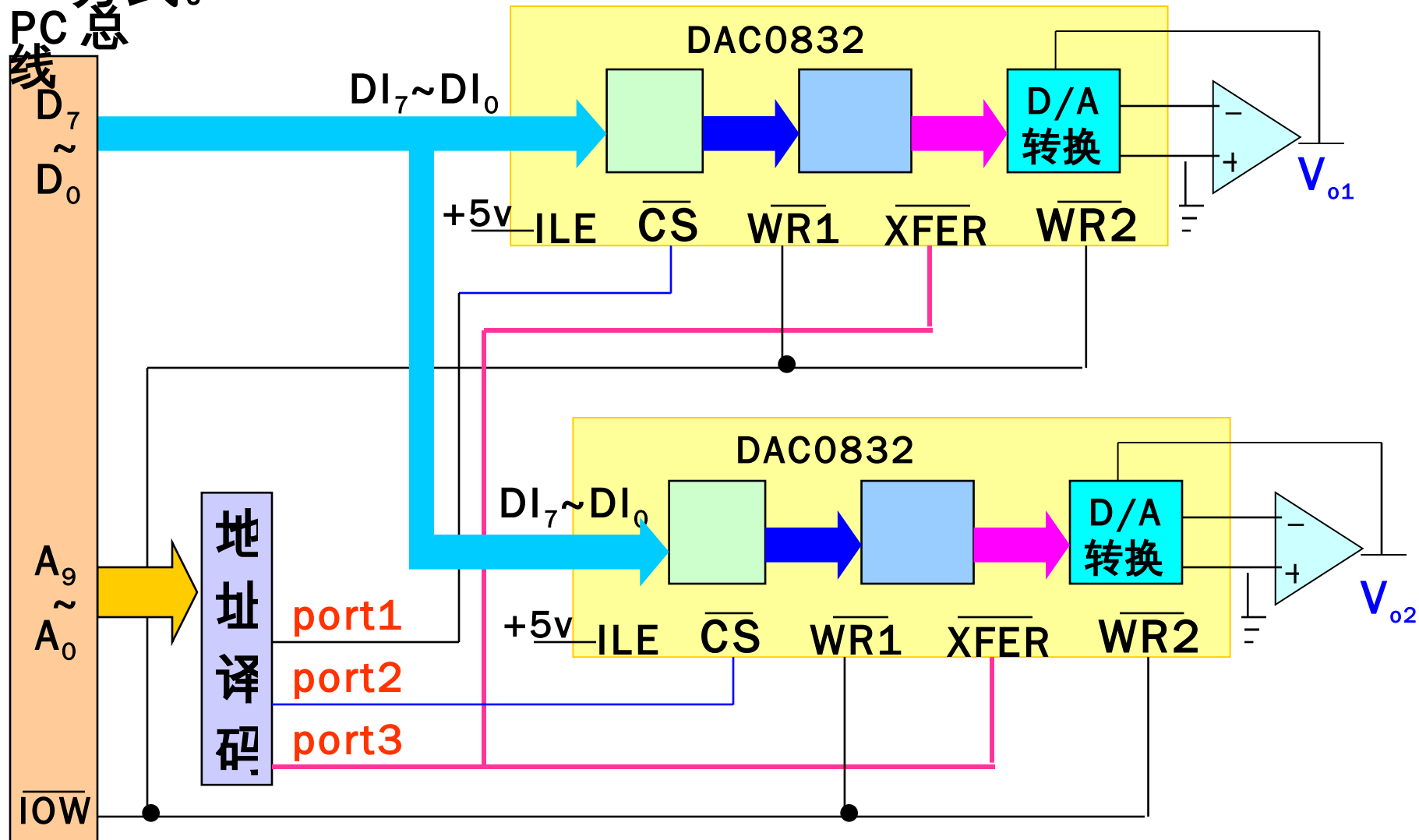


## 转换一个数据的程序段：

```

MOV  AL, data    ;取数字量
MOV  DX, port1
OUT  DX, AL      ;打开第一级锁存
MOV  DX, port2
OUT  DX, AL      ;打开第二级锁存
    
```

当要求多个模拟量同时输出时，可采用双重缓冲方式。



**思考：**相应的程序如何编写？

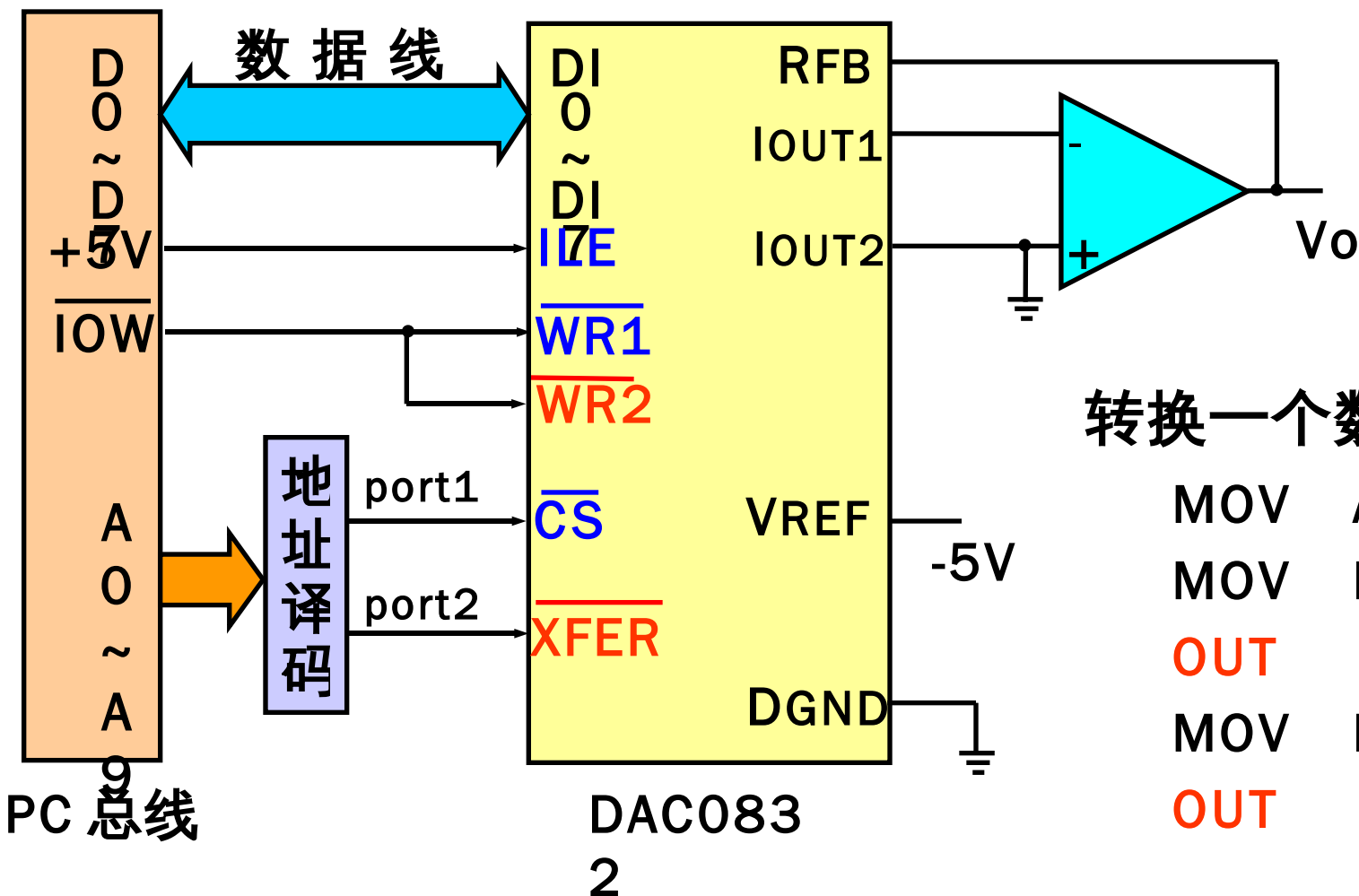


编程code利用上图，将datav1和datav2处的两组数据，  
——对应转换成模拟量同时输出。

```
datav1    DB    11h, 12h, 13h, 14h, 15h, 16h, 17h, 18h, 19h, 1Ah
datav2    DB    21h, 22h, 23h, 24h, 25h, 26h, 27h, 28h, 29h, 2Ah
start:    MOV     AX, code
          MOV     DS, AX
          LEA     SI, data_v1
          LEA     BX, data_v2
          MOV     CX, 10
next:     MOV     AL,    [SI]        ; 取 V1 的数据
          OUT     port1,  AL        ; 打开第一片 0832 第一级锁
存
          MOV     AL,    [BX]        ; 取 V2 的数据
          OUT     port2,  AL        ; 打开第二片 0832 第一级锁
存
          OUT     port3,  AL        ; 打开两片 0832 的第二级锁
存
          INC     SI
          INC     BX
          LOOP    next
          MOV     AH, 4CH
          INT     21H
code      ENDS
```

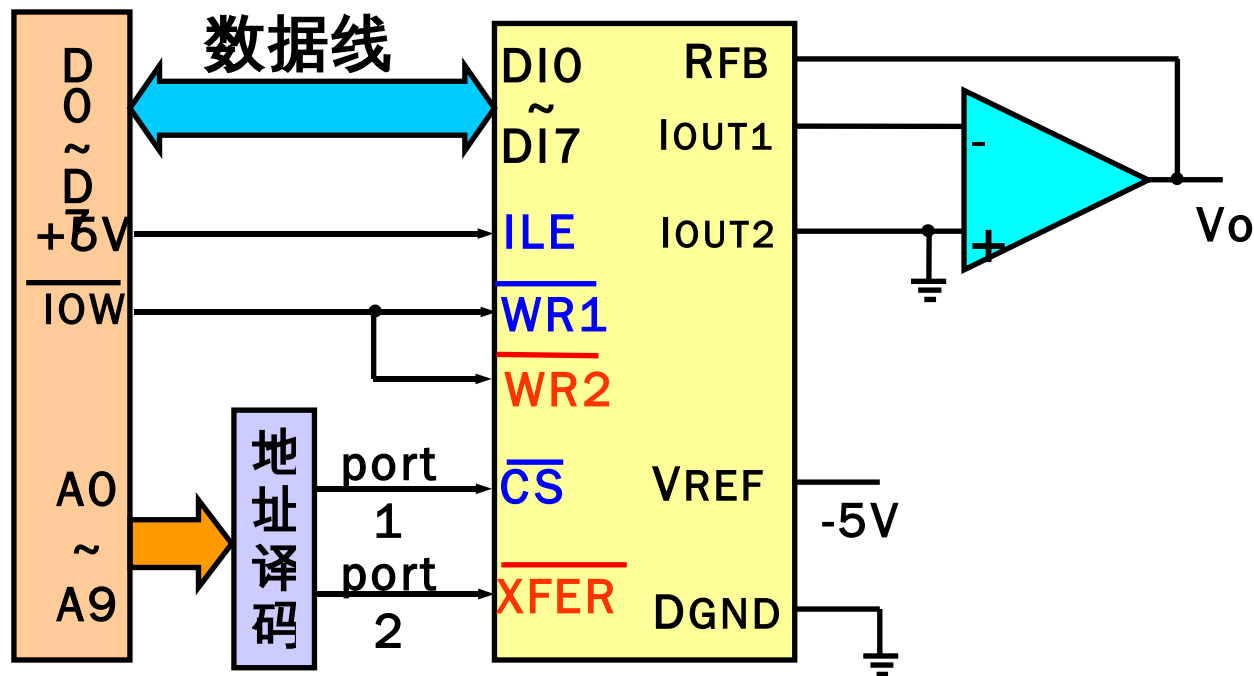
### 3. 应用举例 (调幅)

**例 1** 连线如图，计算当数字量为 0CDH 时的输出  $V_o$ 。



转换一个数据的程序段

```
MOV    AL, 0CDH
MOV    DX, port1
OUT    DX, AL
MOV    DX, port2
OUT    DX, AL
```



调幅分析：当数字量为  $0FFH=255$  时，

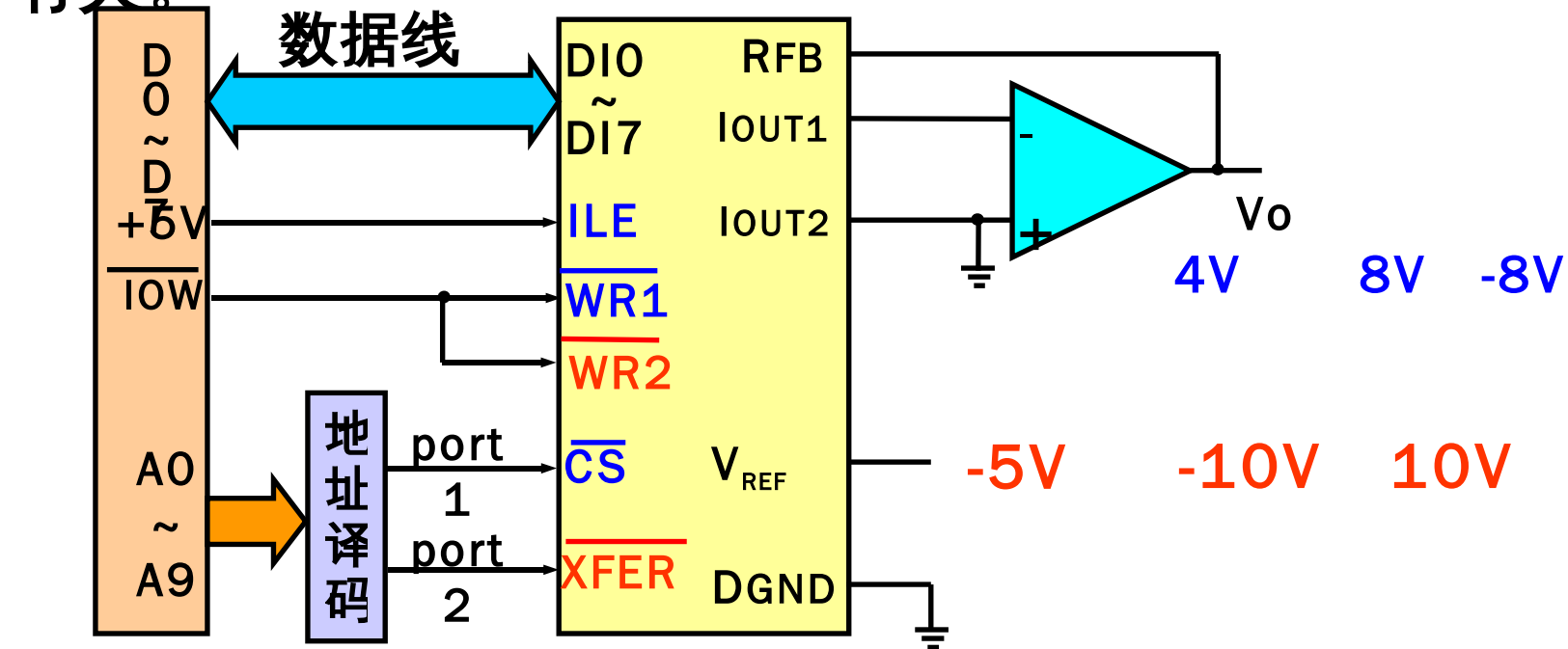
$$I_{OUT1} = \frac{255V_{REF}}{256R_{FB}}$$

$$V_o = - I_{OUT1} \times R_{FB} = - \frac{255}{256} V_{REF}$$

所以：当数字量为  $0CDH=205$ ， $V_{REF} = -5V$  时：

$$V_o = - \frac{205V_{REF}}{256} = 4V$$

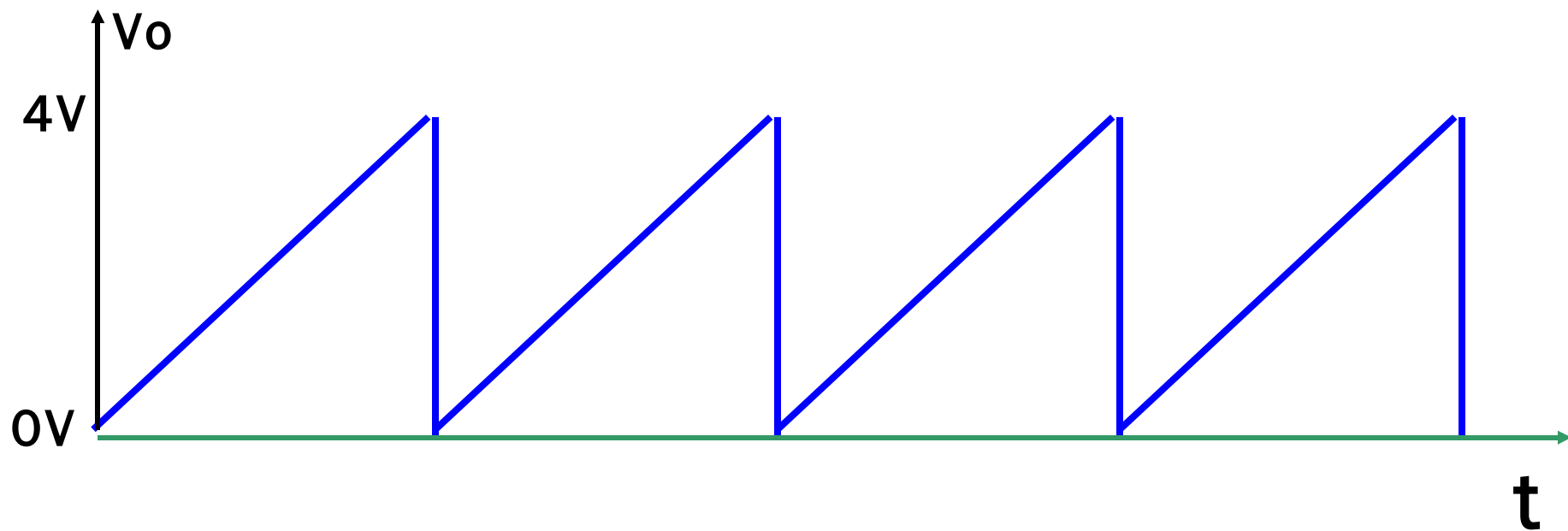
**注意：**  $V_o$  的输出与参考电压  $V_{REF}$ 、  
以及输出的连接方法（同相还是反相）  
有关。



上例中，若  $V_{REF}$  接的是  $-10V$ ，则  $V_o = 8V$

若  $V_{REF}$  接的是  $10V$ ，则  $V_o = -8V$

**例 2** 利用上例连线图，编程输出一锯齿波。

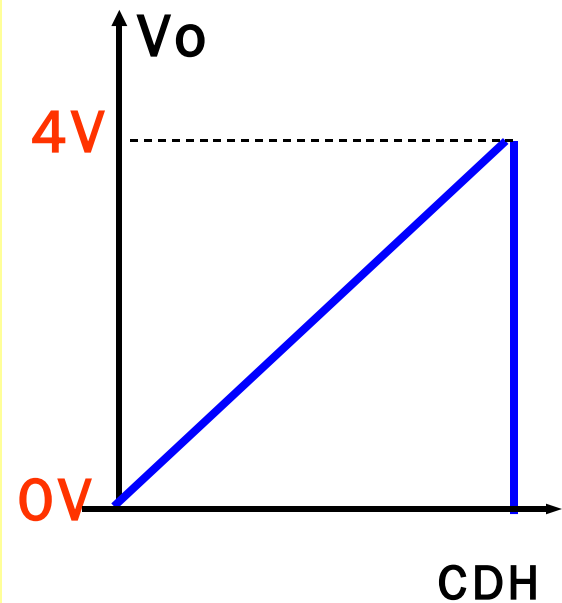


调频：

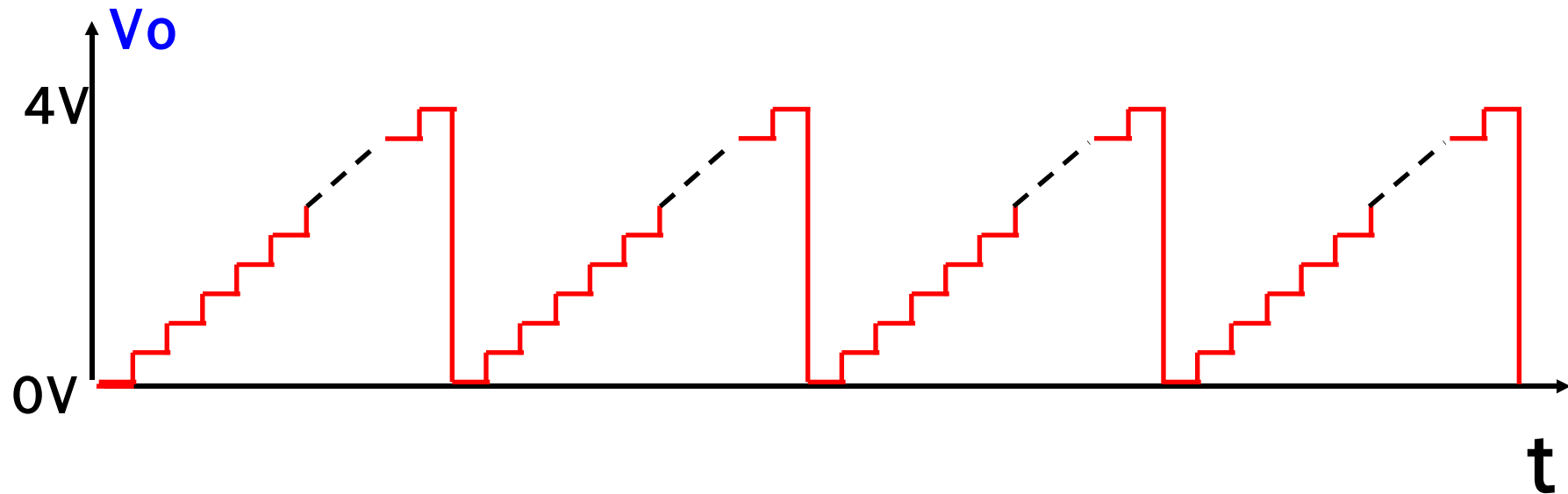
```
code    SEGMENT
        ASSUME  CS:code

start:  MOV     CX, 8000H    ; 波形个数
        MOV     AL, 0       ; 锯齿谷值
next:    MOV     DX, port1   ; 打开第一级锁存
        OUT     DX, AL
        MOV     DX, port2   ; 打开第二级锁
        OUT     DX, AL
        CALL    delay       ; 控制锯齿波的
                                周期
        INC     AL          ; 修改输出值
        CMP     AL, 0CEH    ; 比较是否到锯齿峰
                                值
        JNZ     next       ; 未到跳转
        MOV     AL, 0       ; 重置锯齿谷值
        LOOP    next       ; 输出个数未到跳
                                转
        MOV     AH, 4CH     ; 返回 DOS
        INT     21H
```

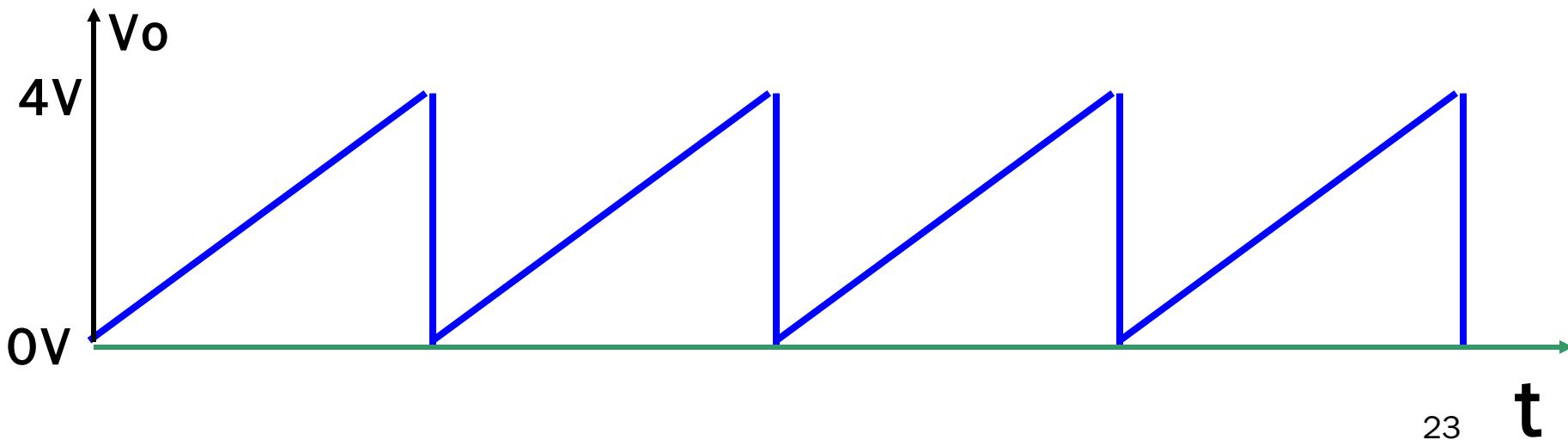
: 子程 delay (略)



# 实际输出的波形图



不是

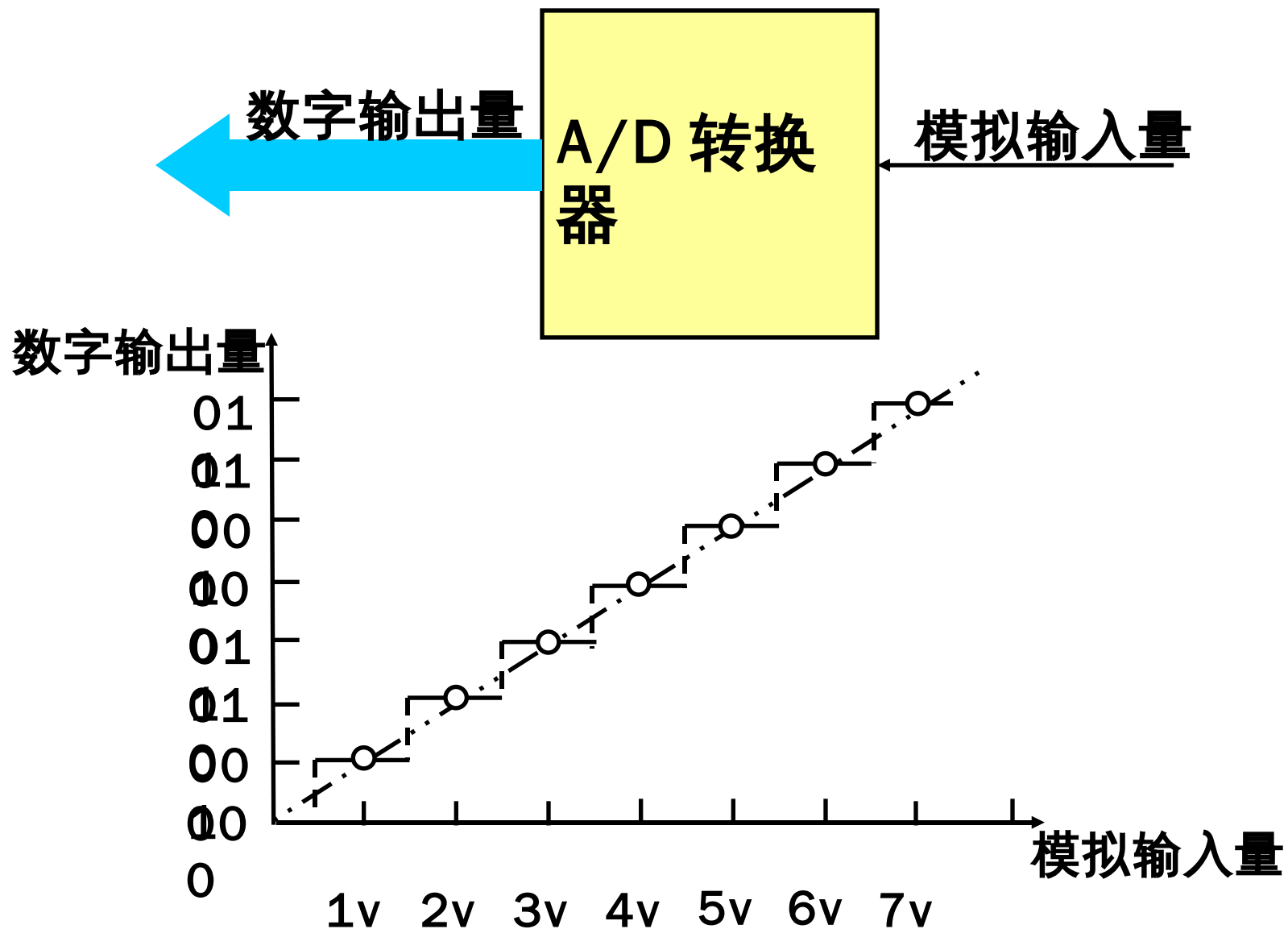


## 第三节 模 / 数转换器

- 一、A/D 转换器的基本原理（自学）
- 二、A/D 转换器的技术指标（自学）
- 三、A/D 转换器及其连接
- 四、典型 A/D 转换器



# 一、A/D 转换器的基本原理（自学）



工作原理	特点
计数式	结构简单、原理清楚 转换速度慢、精度低，实际少用
双积分式	精度高、转换速度慢
逐次逼近式	转换速度较快、精度较高 实际常用
高速并行式	转换速度快, 价格高

□ 计数式 A/D 转换由 8 位 D/A 转换器、8 位计数器和比较器组成。

C 为计数器控制端

:

C=1 , 开始计

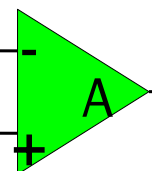
数 ;

C=0 , 停止计

数。 C

模拟输入电压

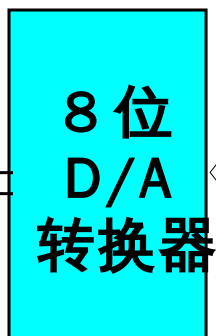
$V_i$



比较器

D/A 转换器输出电压

$V_0$



8 位  
D/A  
转换器

$D_7 D_0$

8 位  
计数器

计数时钟

CLK

CLR

数字量输出  $D_0 \sim D_7$

开始转换

S

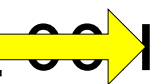
转换结束

EOC

- 启动信号 S :

S 端  : 使 8 位计数器清“0” ,

S 端  : 计数器准备计数。

- 8 位 D/A 转换器 : 数字量  0V 电压输出  $V_o$ 。

当  $V_i > V_o$  时 ,  $C=1$  , 计数器从 0 开始计数 ,

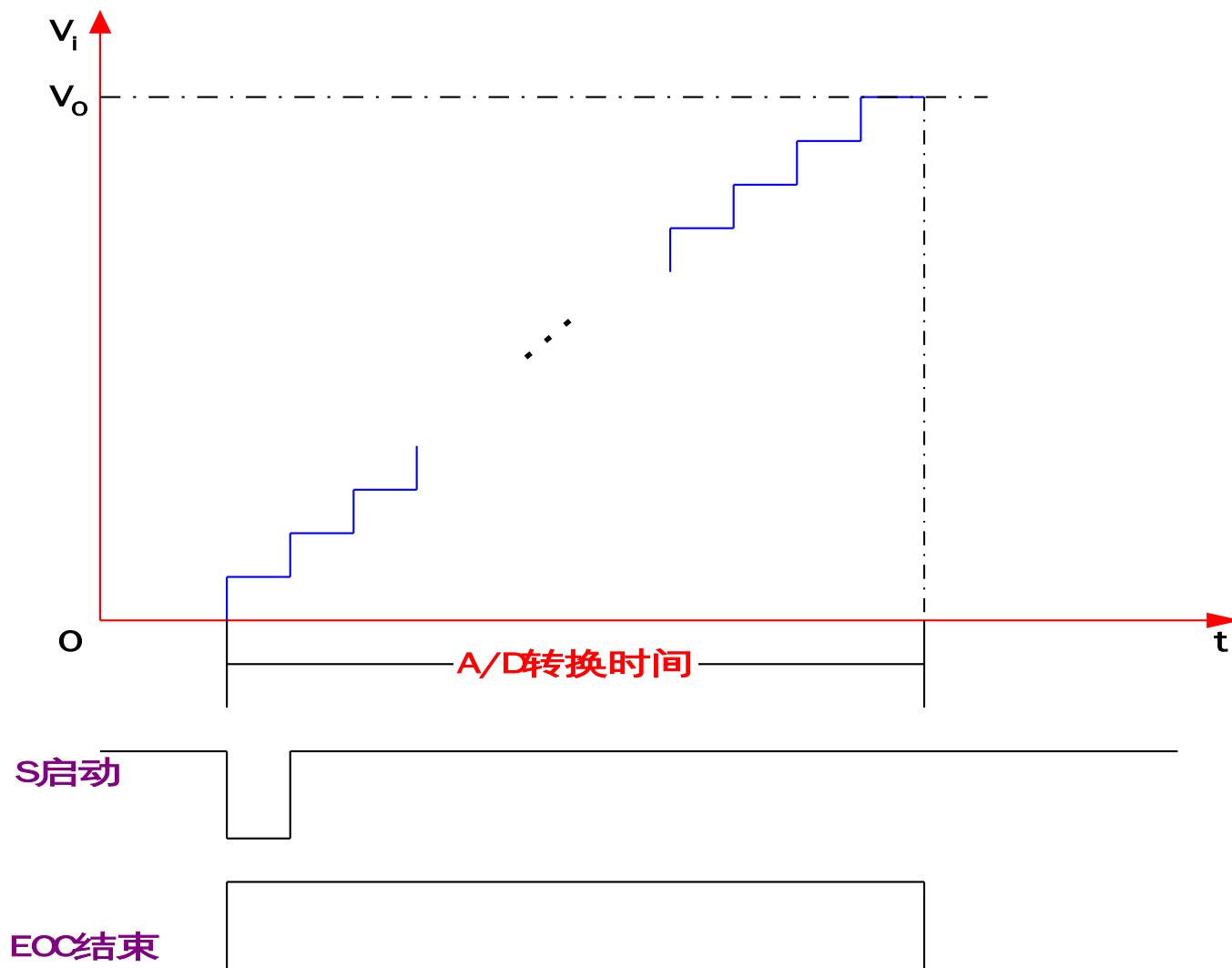
只要  $V_i > V_o$  ,  $C=1$  , 计数器不断计数 ,

当  $V_o \geq V_i$  时 ,  $C=0$  , 计数器停止计数。

-  D7-D0 为  $V_i$  所对应的数字量。实现了 A/D 转换。

C 的  表示 A/D 转换结束 ,

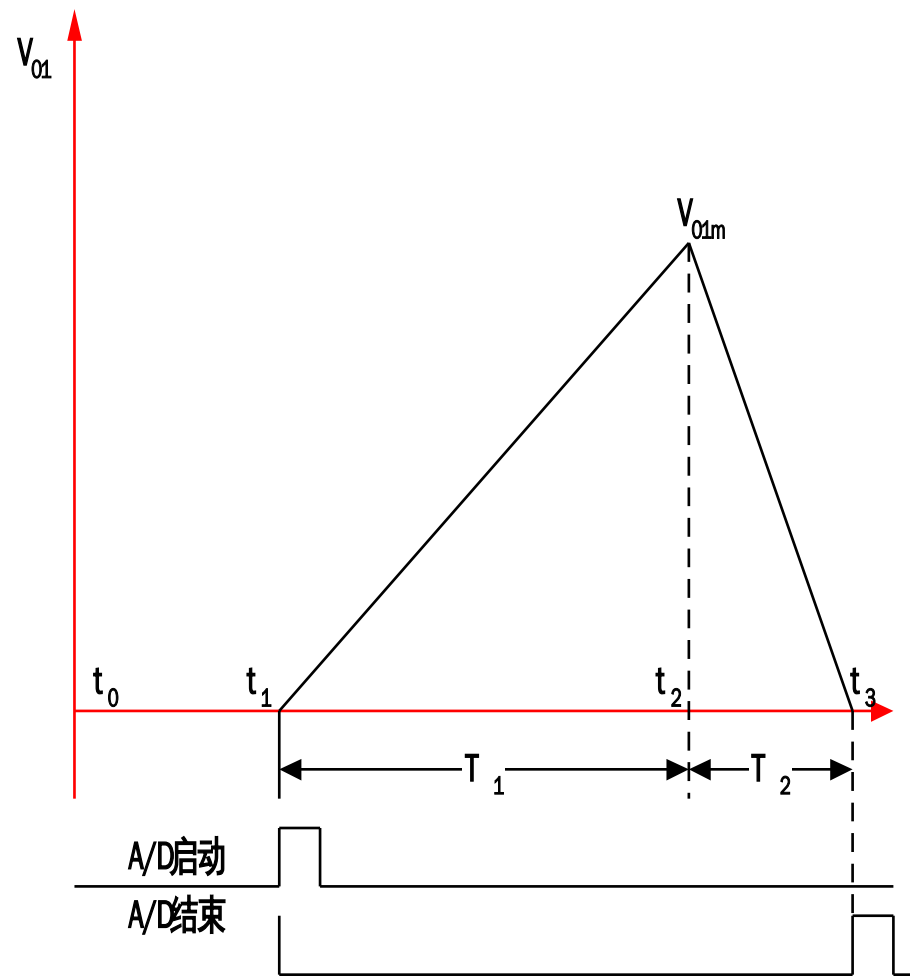
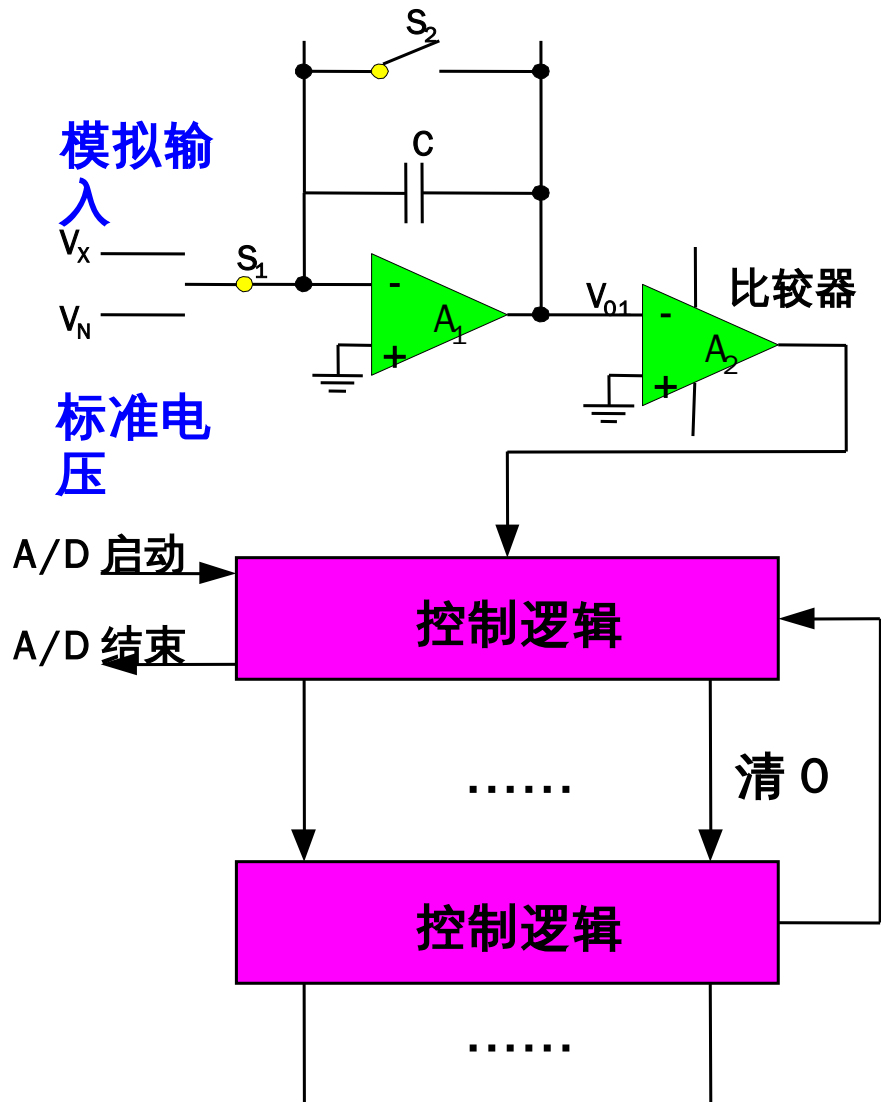
# 计数式 A/D 转换时间图



## □ 双积分式 A/D 转换

- 积分法 A/D 转换种类：  
双斜率、单斜率、多斜率三种。  
仅介绍双斜率法（又称为双积分法）。
- 双积分式 A/D 转换器组成：  
积分器 A1 ；  
零电压比较器 A2 ；  
计数器 ；  
控制逻辑 ；  
标准电压等。

# 双积分式 A/D 转换

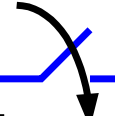



# 双积分式 A/D 转换

- A/D 转换通过采样和测量进行二次积分来完成的。
- 工作过程如下：采样和测量
- 计数器清“0”，

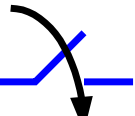
启动脉冲将开关  $S_2$   瞬时接通，积分器 A1 输出  $V_{o1}=0V$ ，

- 采样：

开关  $S_1$   接通模拟输入  $V_x$ ， $S_2$   断开，积分器（ $V_x$  为负）进行正向积分，采样开始，

积分器  $V_{o1}$  稍高于地几个毫伏，比较器  $A_2$   输出 1，计数器开始计数，

- 测量 计数器产生溢出，计数器各位清“0”，采样结束。

• 将  $S_1$   倒向标准电压  $V_N$ ，测量开始，

- $V_N$  为正电压，进行反向积分。

•  $V_{o1}$  高于地电位，比较器输出为 1，计数器又开始计数，直到  $V_{o1}=0$  为止，输出 A/D 结束信号。



□ 采样阶段的正向积分。

设正向积分时间为  $T_1$ ，则积分输出：

$$V_{01} = -\frac{1}{RC} \int_{t_1}^{t_2} (V_X) dt = \frac{T_1}{RC} \frac{1}{T_1} \int_{t_1}^{t_2} (V_X) dt$$

$$\overline{V_X} = \frac{1}{T} \int_{T_1}^{T_2} (-V_X) dt$$

$$\text{当} \rightarrow t=t_2 \text{ 时, } V_{01m} = \frac{1}{RC} \overline{V_X} T_1$$

□ 分析测量阶段反向积分：

设反向积分时间为  $T_2$ ，则：

$$V_{01} = V_{01m} + \left[ \frac{1}{RC} \int_{t_2}^{t_3} V_N dt \right] \rightarrow \rightarrow$$

• 当  $\rightarrow t=t_3$  时， $V_{01}=0$ ， $\rightarrow$  所以  $0 = \frac{V_{01m}}{RC} \int_{t_2}^{t_3} V_N dt$

即  $\frac{T_1}{RC} \overline{V_X} = \frac{V}{RC} T_2$ ,  $\overline{V_X} T_2 = T_1 V_N$

• 若计数时钟频率为  $f$ ，则根据计数  $N$  可以求得计数时间  $T$ 。

$$T_1 = N_1 / f, \quad T_2 = N_2 / f \overline{V_X}$$

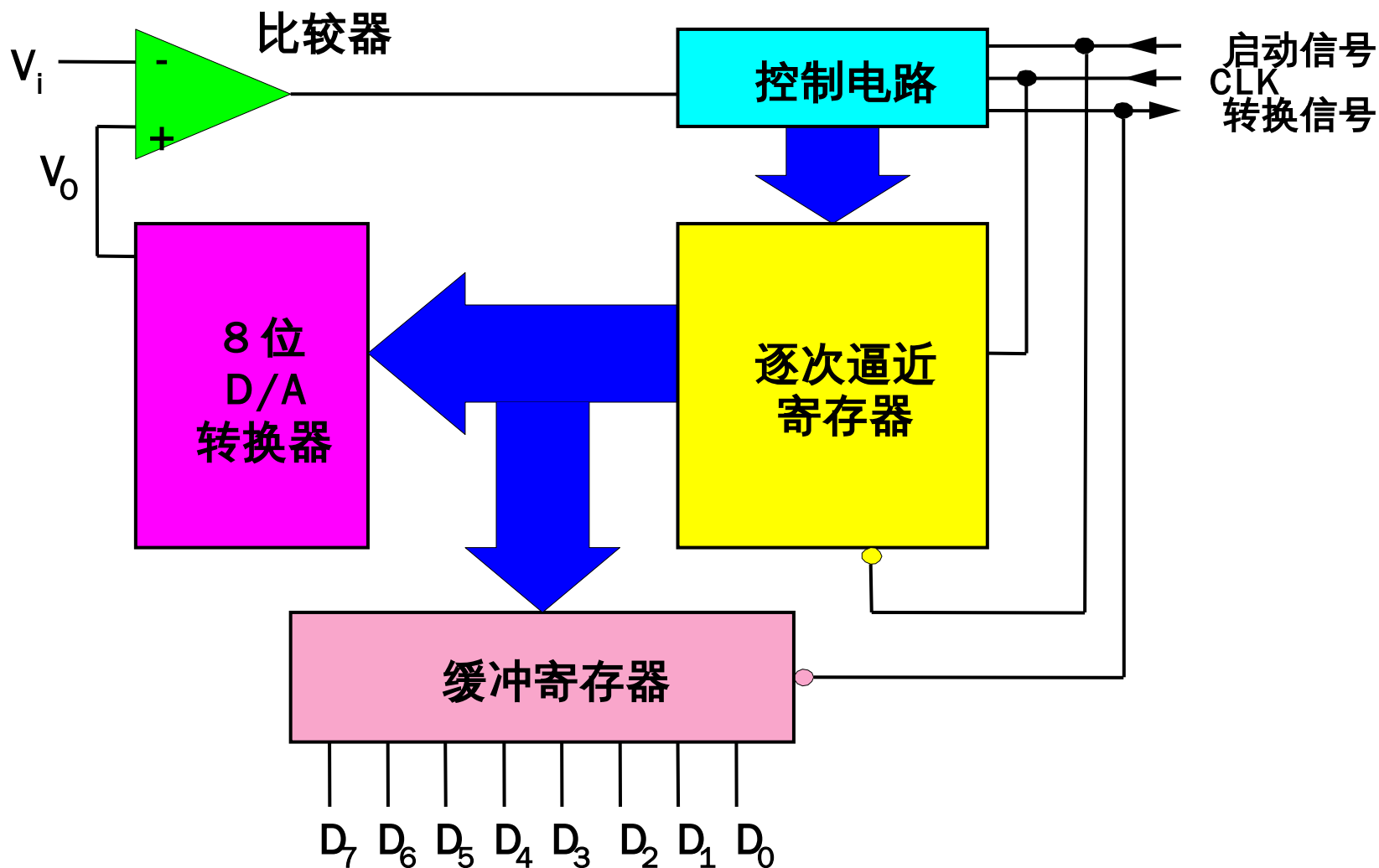
$N_1 \overline{V_X} V_N$  已知，

$$N_2 = N_1$$

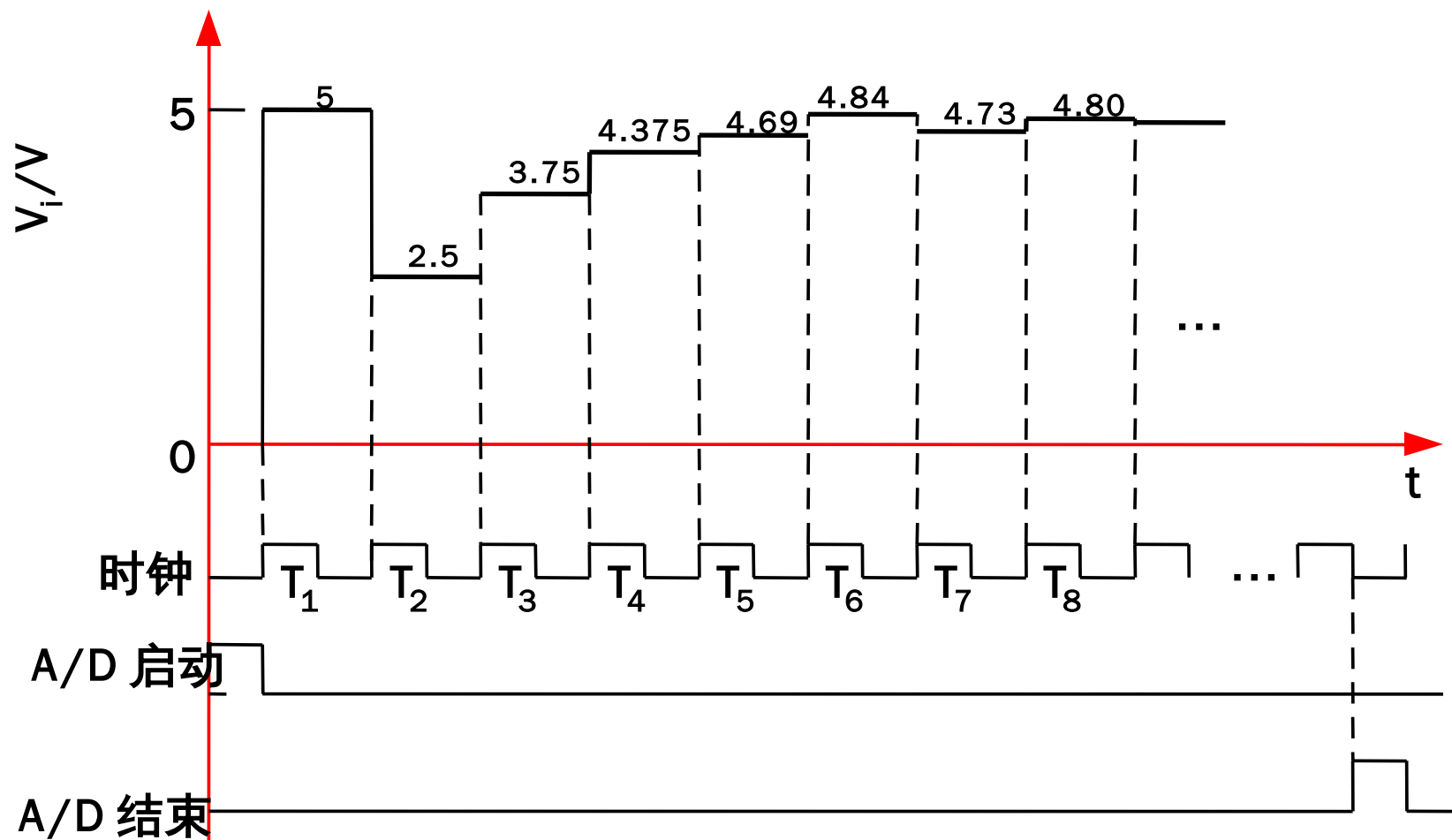
# □ 逐次逼近式 A/D 转换

- 逐次逼近式 A/D 转换是用得最多的一种方法。
- 组成：  
8 位 D/A 转换器、比较器、控制逻辑，逐次逼近寄存器。
- 工作过程：  
从最高位开始通过试探值逐次进行测试，  
直到试探值经 D/A 转换器输出  $V_o$  与  $V_x$  相等或达到允许误差范围为止。则该试探值就为 A/D 转换所需的数字量。

# 逐次逼近式 A/D 转换工作原理图




# 逐次逼近式 A/D 转换



如：实现模拟电压 4.80V 相当于数字量 123 的 A / D 转换。

具体过程如下：


x 当出现启动脉冲  时，逐次逼近寄存器清“0”；

x 当第一个 T1 到来，逐次逼近寄存器 **最高位**  $D_7$  置“1”，

8 位 D/A 转换器输入为 10000000B，

输出  $V_o$  为满度的一半 5V，即满量值的 128/255。

若  $V_o > V_i$ ，比较器输出低电平，

x 当第二个   $T_2$  到来，逐次逼近寄存器  $D_6$  位置“1”，  
D/A 转换器的数字量输入为  $01000000B$ ，  
输出电压为  $2.5V$ ， $V_o < V_i$ ，比较器输出高电平，  
将  $D_6$  位的“1”保留（否则，将  $D_6$  位置“0”）；

□ 第三个   $T_3$  时钟脉冲来，又将  $D_5$  位置“1”……

x 重复上述过程直到  $D_0$  位置“1”，再与输入比较。

x 经过 8 次以后，

x 逐次逼近寄存器中得到的数字量就是转换结果。

x 过程用下表表示。

# 逐次逼近式 A/D 转换

设定试探值	D/A输出电压 $V_0$ (V)	$V_0$ 与 $V_i$ 比较	结果
10000000	5.0	$V_0 > V_i$ , $D_7=0$	0
01000000	2.5	$V_0 < V_i$ , $D_6=1$	64
01100000	3.75	$V_0 < V_i$ , $D_5=1$	$64+32=96$
01110000	4.375	$V_0 < V_i$ , $D_4=1$	$64+32+16=112$
01111000	4.69	$V_0 < V_i$ , $D_3=1$	$64+32+16+8=120$
01111100	4.84	$V_0 > V_i$ , $D_2=0$	$64+32+16+8=120$
01111010	4.76	$V_0 < V_i$ , $D_1=1$	$64+32+16+8+2=122$
01111011	4.80	$V_0 < V_i$ , $D_0=1$	$64+32+16+8+2+1=123$



# 比较三种 A/D 转换方式

- 计数式 A/D 转换速度慢，价格低，适用于慢速系统；
- 双积分式 A/D 转换分辨率高，抗干扰性好，但转换速度较慢，适用于中速系统。
- 逐次逼近型 A/D 转换精度高、转换速度快、易受干扰。
- 微机系统中大多数采用逐次逼近型 A/D 转换方法。

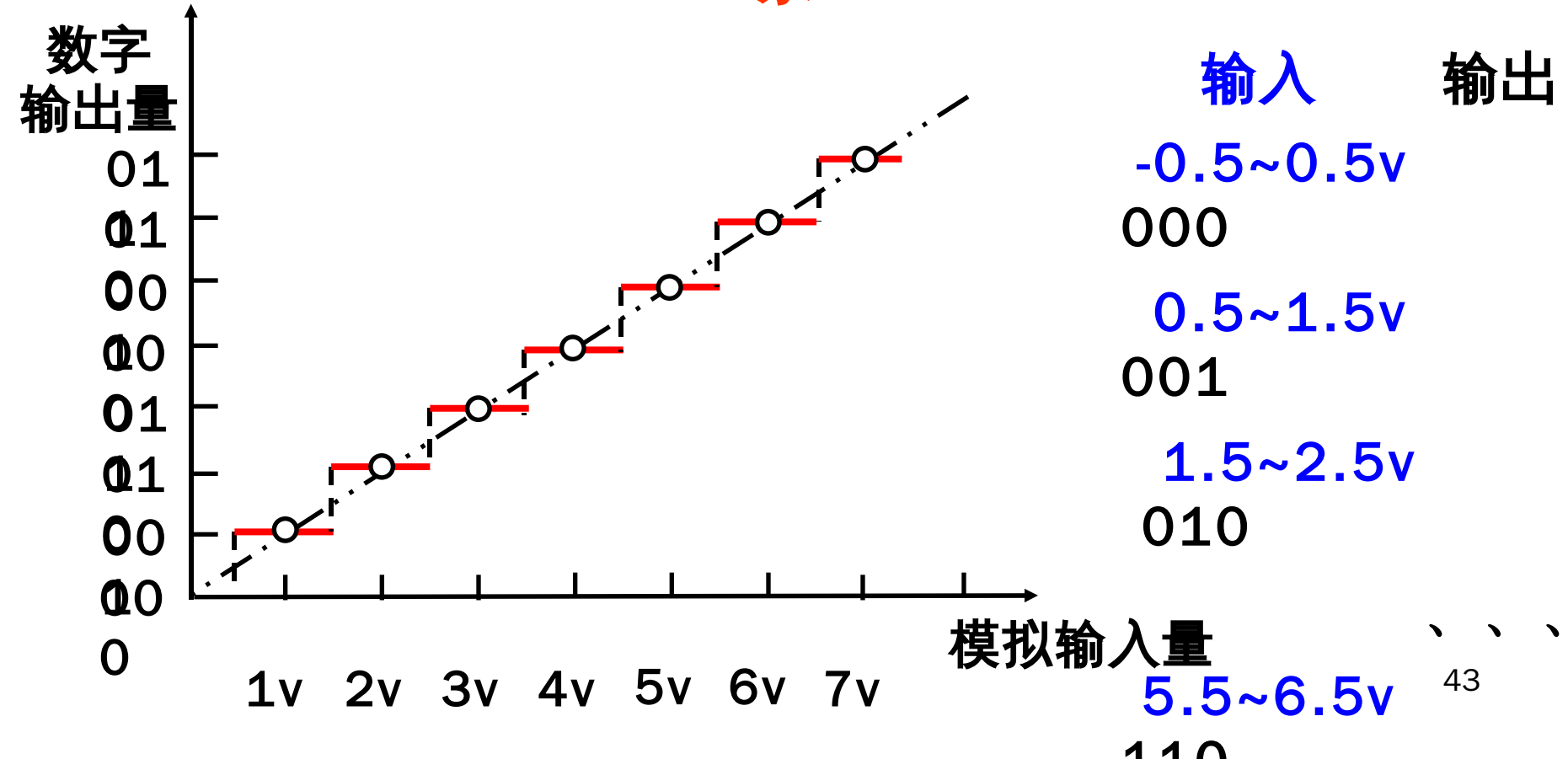
## 二、A/D 转换器的技术指标 (自学)

1. 分辨率
2. 转换精度
3. 转换时间和转换率

# 1 . 分辨率

指 A/D 转换器所能分辨的最小模拟输入量，  
或指转换器满量程模拟输入量被分离的级数。

在 ADC 中，模拟量和数字量之间不是一一对应的关系



A/D 分辨率通常用能转换成的数字量位数表示。

如：8 位 A/D 转换器的分辨率为 8 位。

10 位 A/D 转换器的分辨率为 10 位。

## 2 . 转换精度

指在输出端产生给定的数字量，  
实际输入的模拟值与理论输入的模拟值之间的偏差。

反应 ADC 的实际输出接近理想输出的精确程度

。

由于在一定范围内的模拟值产生相同的数字量，  
取该范围内的中间模拟值计算。

常用数字量最低有效位 **LSB** 对应模拟量  $\Delta$  的几分之几表示。

若 ADC 在  $\pm \frac{1}{2} \Delta$  范围内产生相对应的**唯一数字量**，  
称其精度为 **0LSB**

在此基准上定义 ADC 精度：

ADC 在  $\pm \frac{3}{4} \Delta$  范围内产生相对应的唯一数字量，  
称其精度为  **$\pm \frac{1}{4} \text{LSB}$**

ADC 在  $\pm \frac{1}{2} \Delta$  范围内产生相对应的唯一数字量，  
称其精度为  **$\pm \frac{1}{2} \text{LSB}$**

### 3 . 转换时间和转换率

**转换时间**指完成一次 A/D 转换所需的时间，  
从启动信号开始到转换结束，得到稳定数字量的时间。

**转换率**是转换时间的倒数。

## 三、A/D 转换器及其连接

### 1. A/D 转换器分类

### 2. A/D 转换器与系统的连接



# 1. A/D 转换器分类

- 按工作原理分
- 按输入方式分
- 按输出方式分
- 按性能特点分
- 按输出是否带三态缓冲分

- 按工作原理分

计数式 ADC 、

双积分式 ADC

逐次逼近式 ADC 、 并行式 ADC

- 按模拟量输入方式分

单极性 ADC 、 双极性 ADC

- 按数字量输出方式分

并行 ADC 、 串行 ADC

## 性能特点分

### 按分辨率分

4 位、6 位、8 位、10 位、12 位、14 位、16 位、

### 按转换速度分

低速、中速、高速、超高速

转换时间分别为  $\geq 1s$  、  $\leq 1ms$  、  $\leq 1\mu s$  、  $\leq 1ns$  )

### 按转换精度分

低精度、中精度、高精度、超高精度

- 按输出是否带三态缓冲分

带可控三态缓冲 ADC

如： ADC0809

不带可控三态缓冲 ADC

如：

AD570、ADC1210

## 2. A/D 转换器及其连接

- 1) A/D 转换器的典型信号
- 2) A/D 转换器各信号与系统的连接

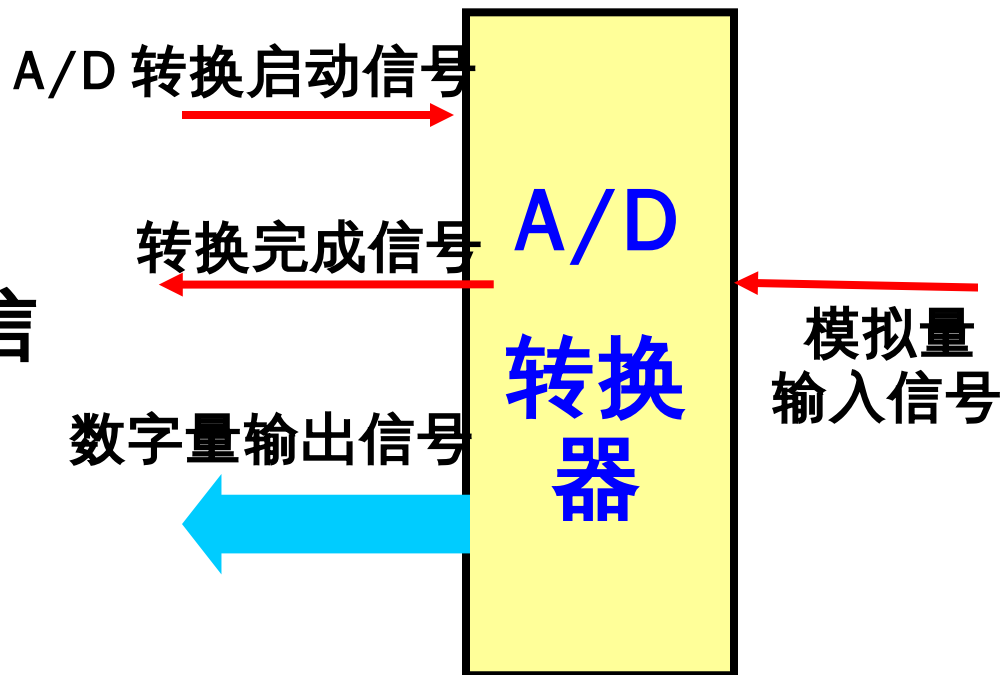
# 1) A/D 转换器的典型信号

① 模拟量输入信号

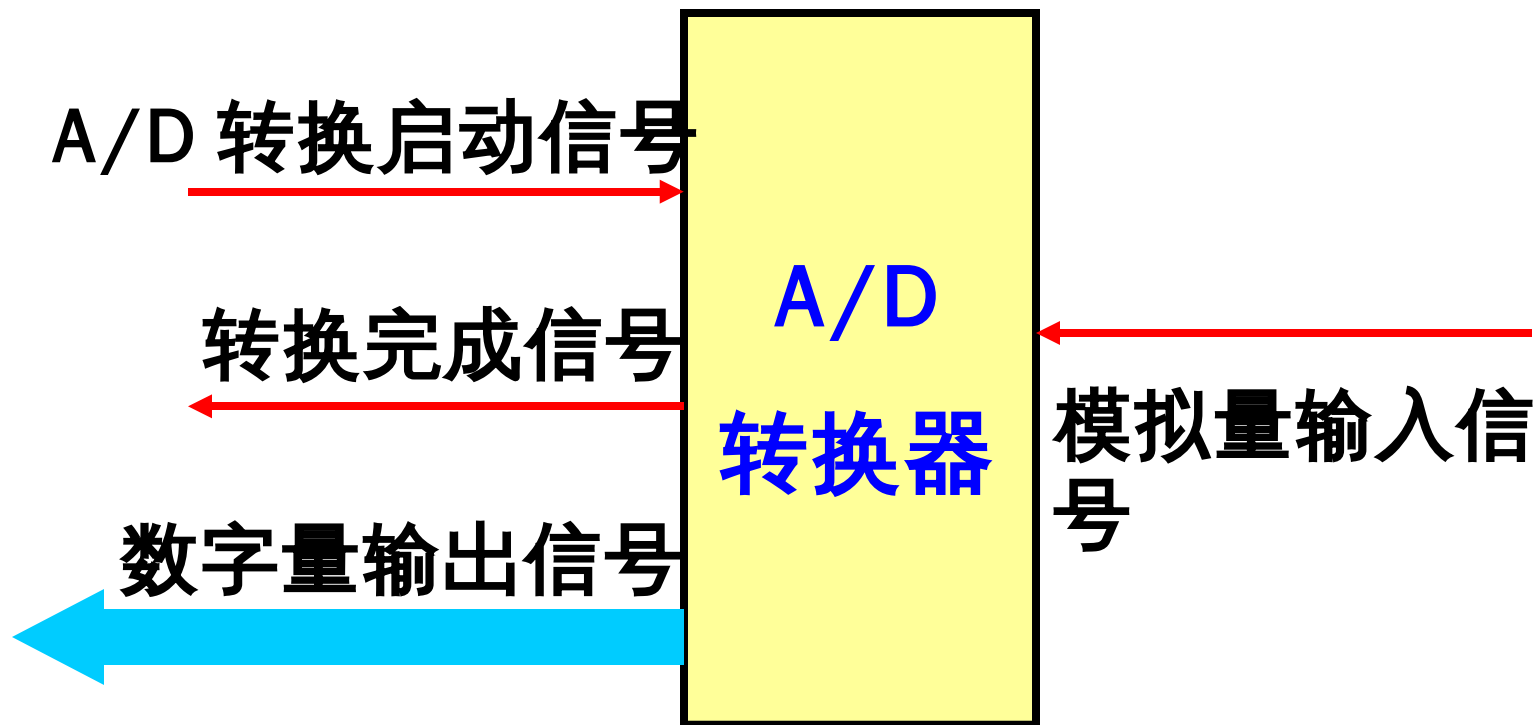
② A/D 转换启动信号

③ 转换完成（结束）信号

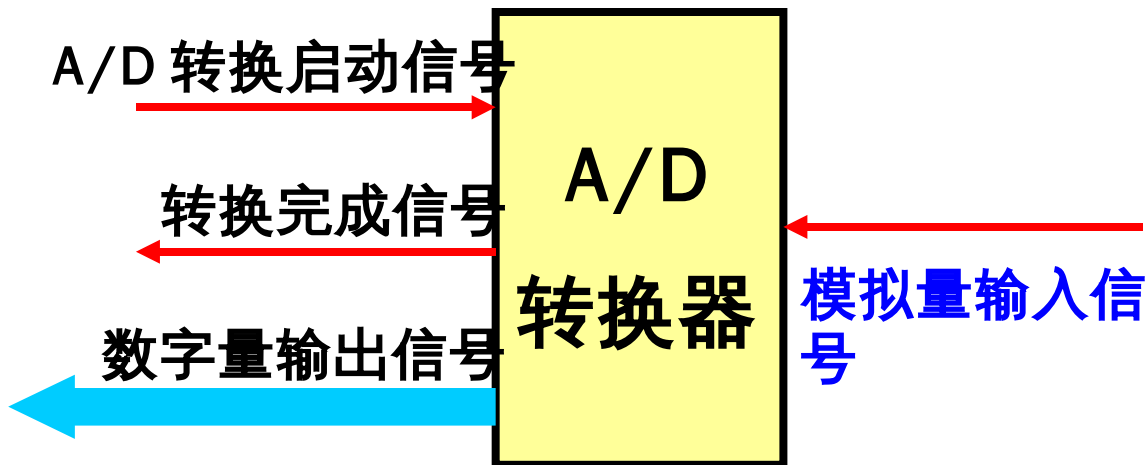
④ 数字量输出信号



## 2) A/D 转换器各信号与系统的连接



## ① 模拟量输入信号



**x** 注意 A/D 转换器允许输入的模拟值范围，不要超出范围

**□** 为充分发挥 A/D 转换器的分辨率，输入量应与转换量程相称。

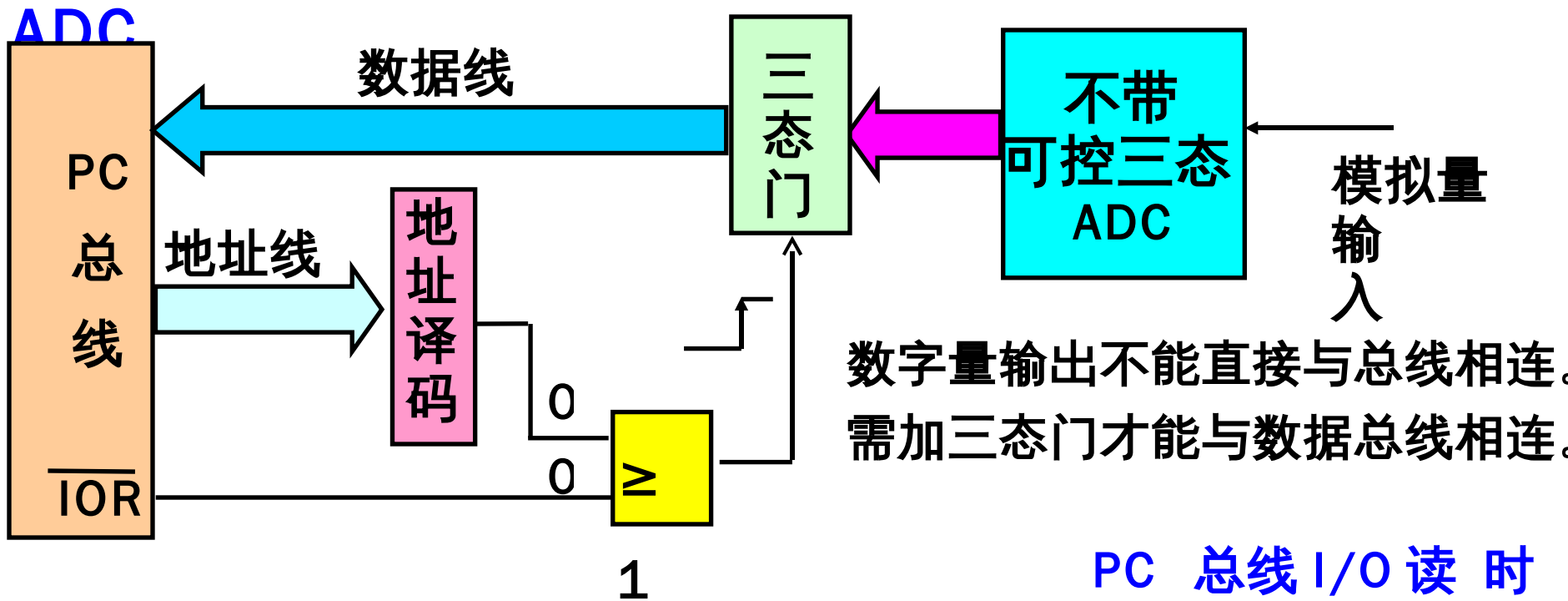
**例如** 某 A/D 转换的范围为  $0\sim 10V$ ，输入的模拟信号为  $0\sim 5V$   
则应将输入信号放大 2 倍，再送入 A/D 进行转换。



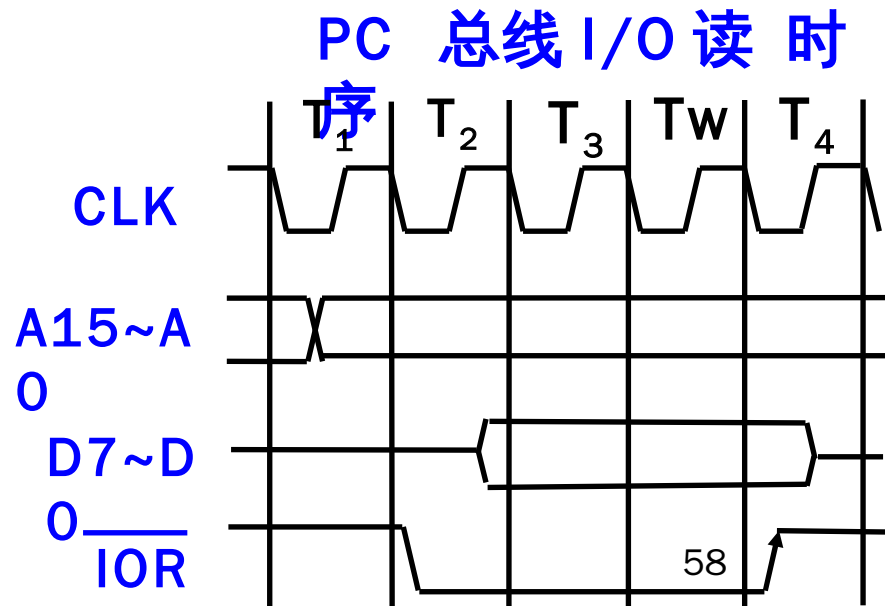
## ② 数字量输出信号

- 输出不带可控三态缓冲器的 ADC
- 输出带可控三态缓冲器的 ADC
- 输出位数超过微机数据总线的 ADC

## □ 输出不带可控三态缓冲器的

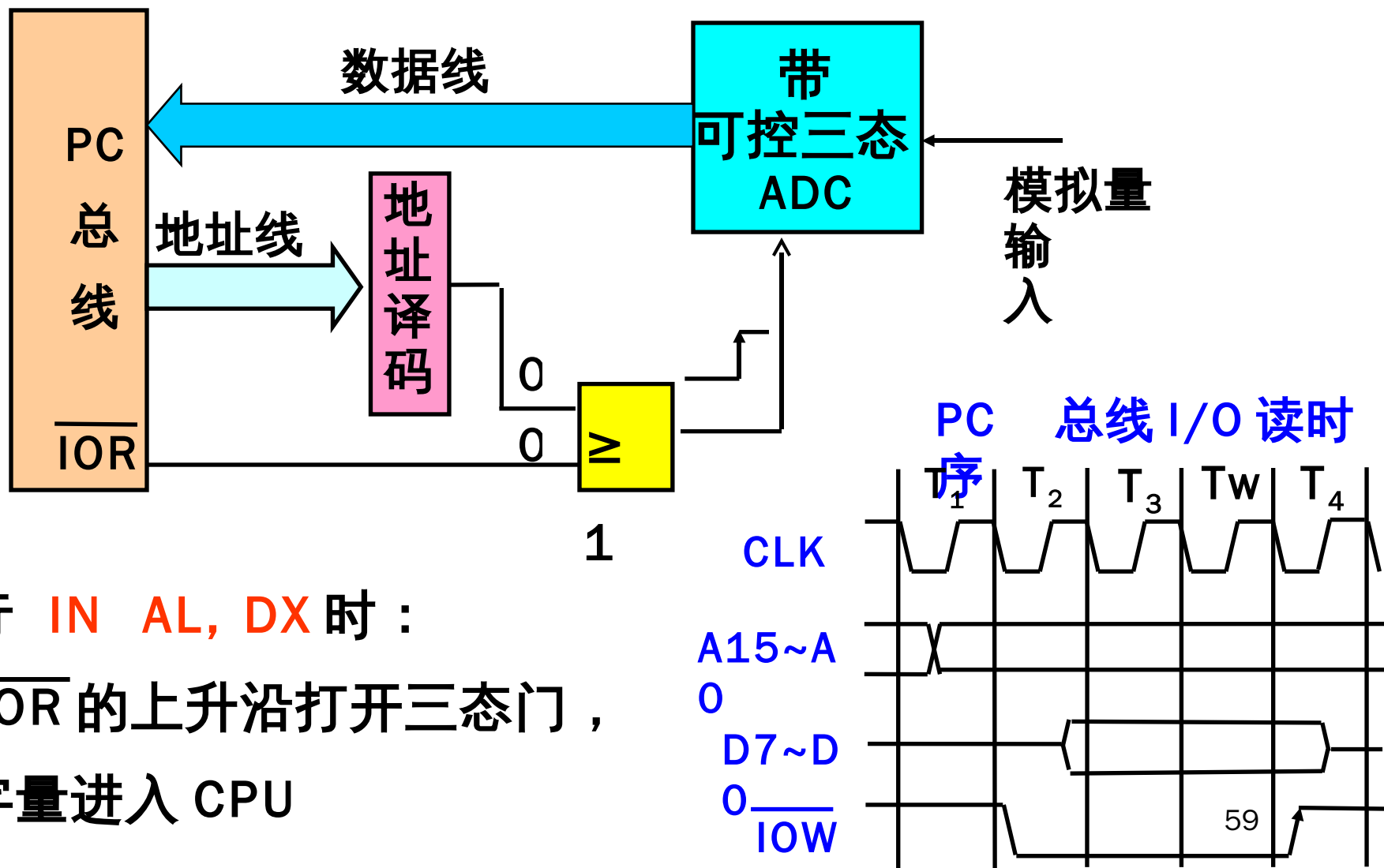


执行 **IN AL, DX** 时：  
在  $\overline{IOR}$  的上升沿控制三态门，  
数字量进入 CPU



## □ 输出带可控三态缓冲器的 ADC

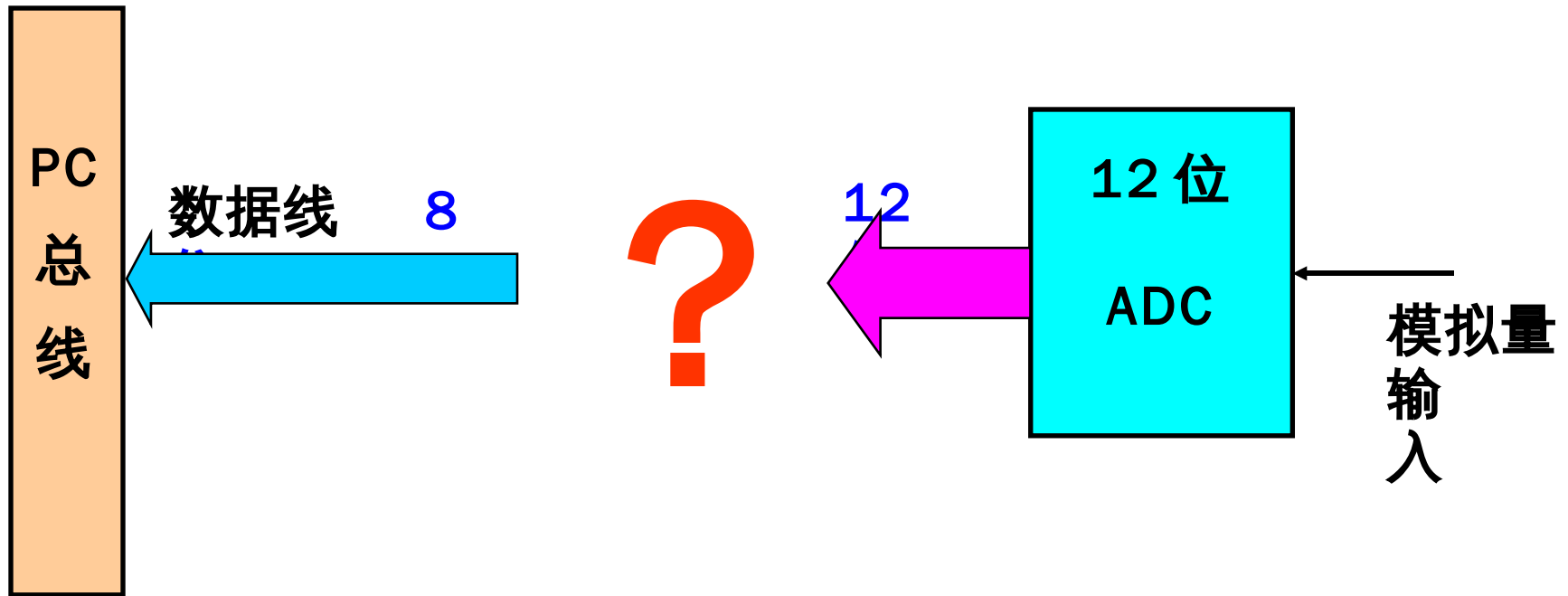
其数字量输出可直接与微机的数据总线相连。



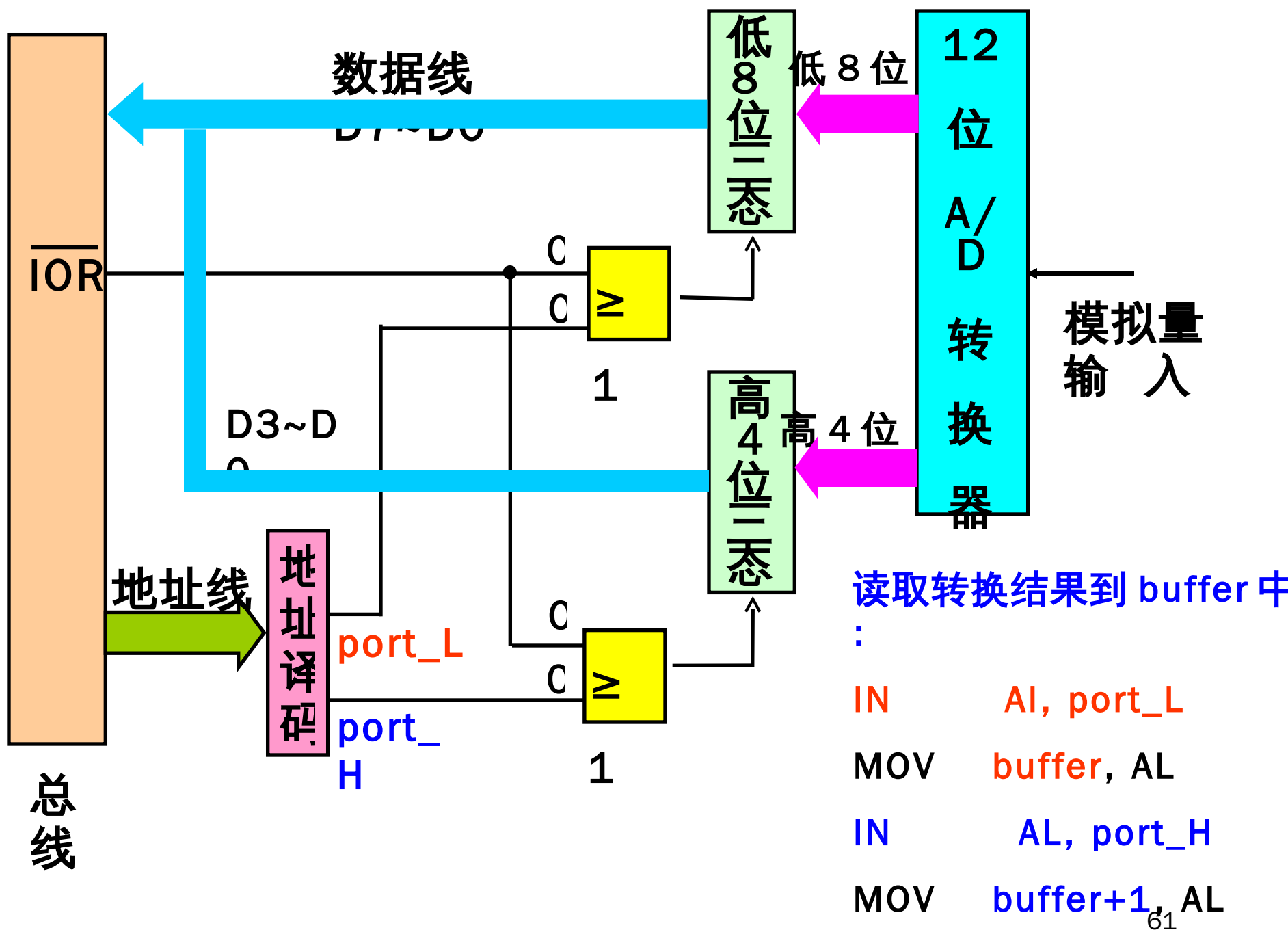
执行 **IN AL, DX** 时：

在  $\overline{IOR}$  的上升沿打开三态门，  
数字量进入 CPU

## ❑ 输出数字量位数超过微机数据总线的 ADC

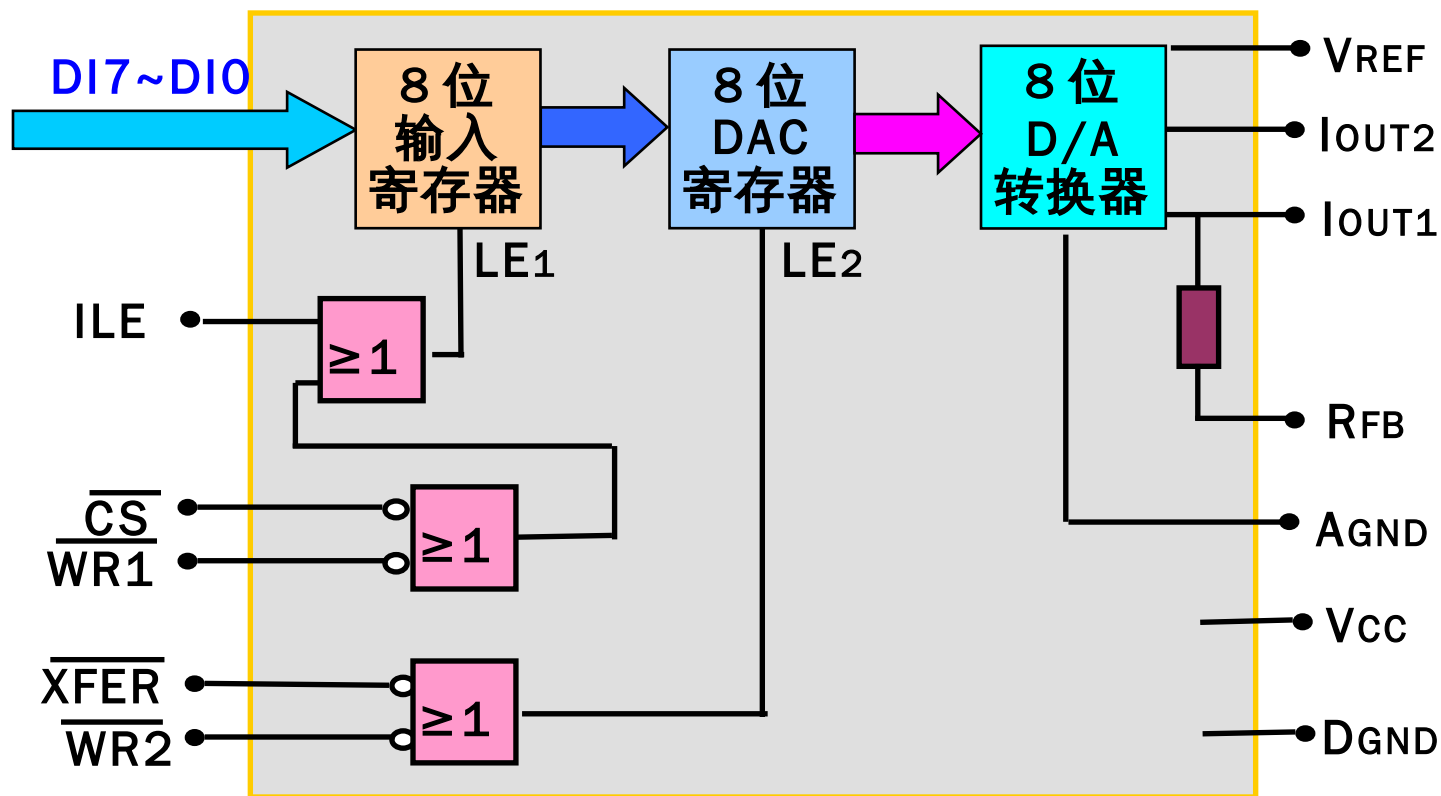


ADC 的转换结果不能一次进入 CPU ，需按字节分多次读取。

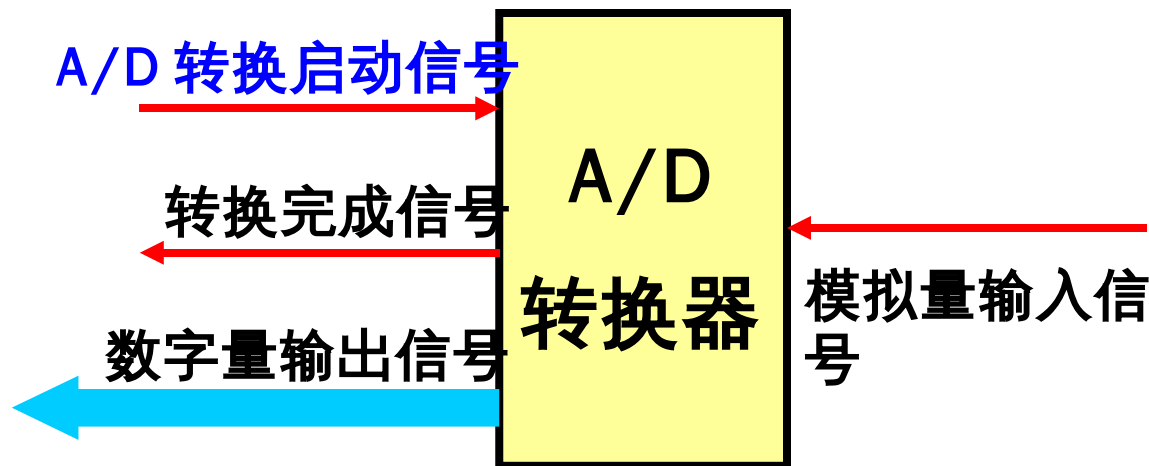


### ③ A/D 转换启动信号

对 D/A 芯片，只要数字信号进入转换电路，就开始 D/A 转换，无启动信号。



- 而 A/D 芯片，每进行一次数据转换，均受启动信号控制，在启动信号有效之后，才开始一次 A/D 转换，得到一个数字量。



- 对一个连续的模拟信号进行 A/D 转换时，在一个数据转换完成之后，应再发启动信号，开始下一个数据的转换。

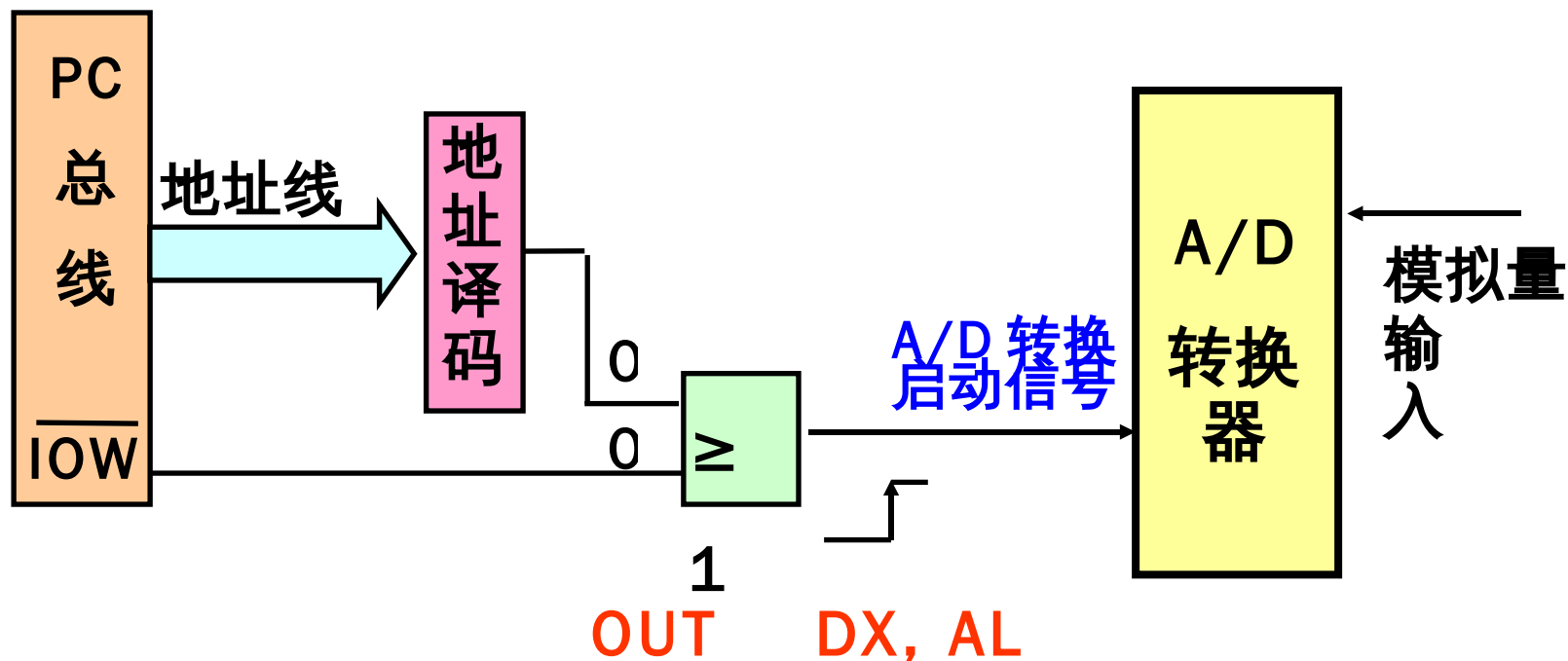
# A/D 启动信号的形式有电平启动和脉冲启动

## ◇ 脉冲启动

对脉冲启动的 ADC，

如 ADC0804、ADC0809、ADC1210

可用 CPU 执行输出指令时发出的片选信号和写信号组合得到

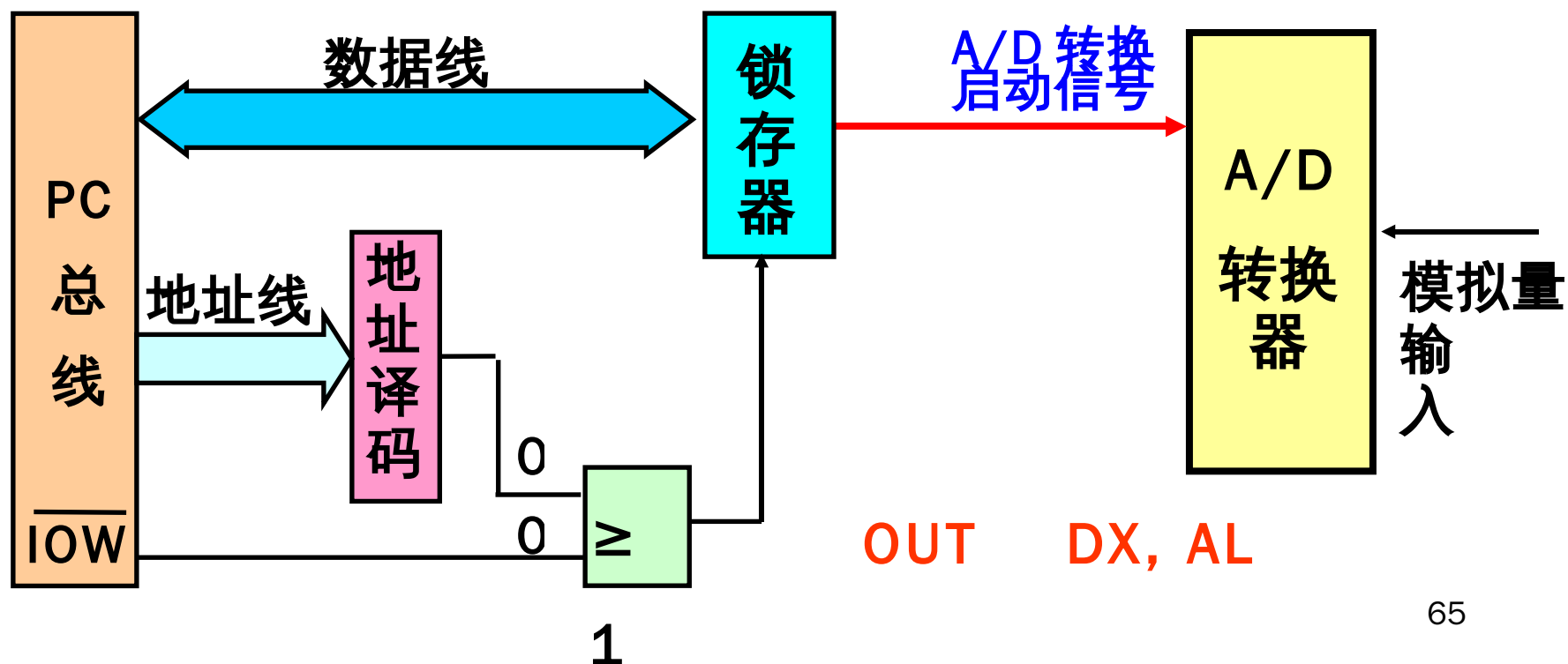




## ◇电平启动

对电平启动的 ADC ，如 AD570 、 AD571 、 AD572 ，该信号必须保持到 A/D 转换结束，中途不能撤除；否则会停止转换，得到错误结果。

CPU 可通过并行接口对 ADC 芯片发电平形式的启动信号。

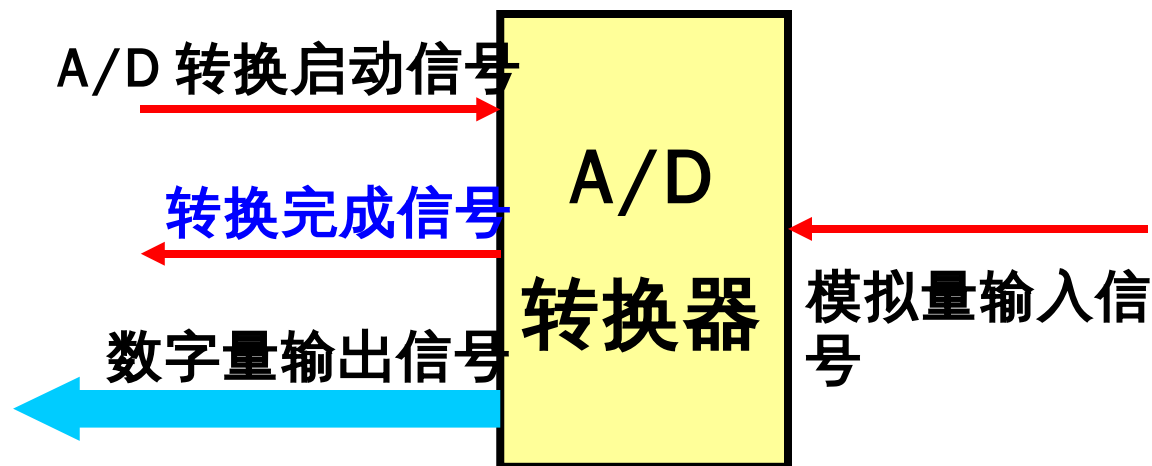


## ④ 转换完成 EOC 信号

A/D 转换需要一定时间，  
在转换完一个数据之后，

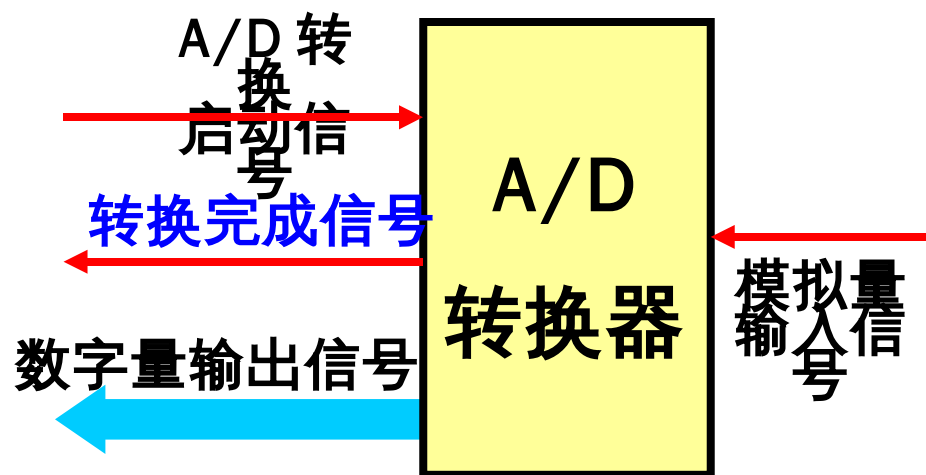
A/D 芯片会发出一个转换完成信号。

( 相当于输入设备的准备好  
信号 )



将 A/D 芯片看作一个输入设备，  
CPU 可采用下列 四种方法，读取 A/D 的转换结果：

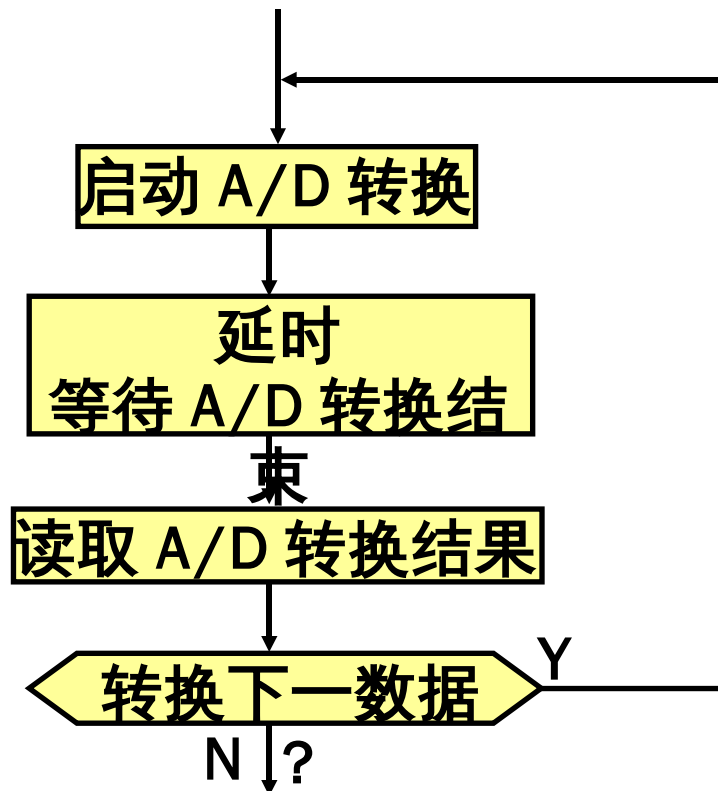
- 程序延时方式（同步方式）
- 程序查询方式
- 中断方式
- 等待方式



四种方式对 EOC 信号的处理各不相同

## □ 程序延时方式（同步方式）

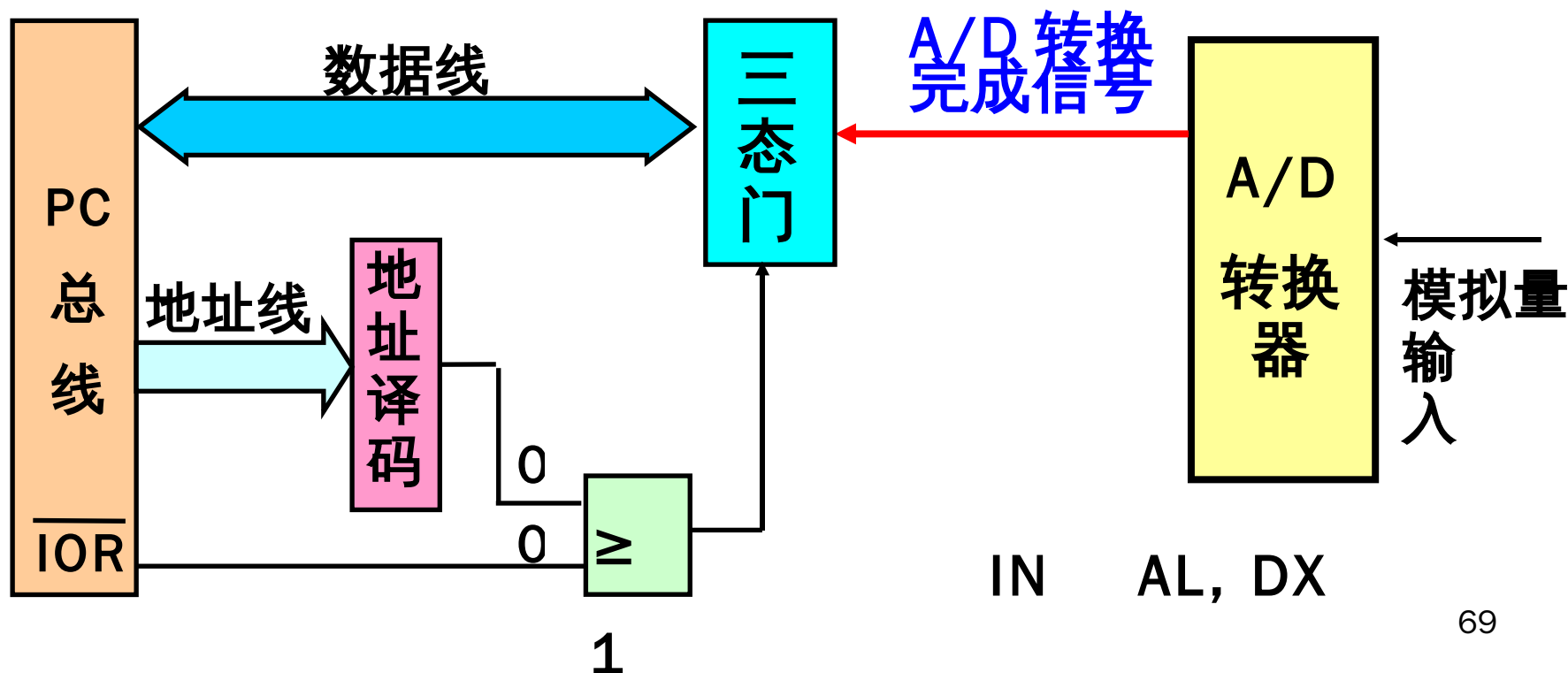
通过查阅手册了解 A/D 转换一个数据所需时间，  
在 CPU 启动 A/D 转换之后，**执行一个固定延时程序**，  
延时应**大于等于** A/D 的转换时间，  
然后 CPU 再读取 A/D 的转换结果。

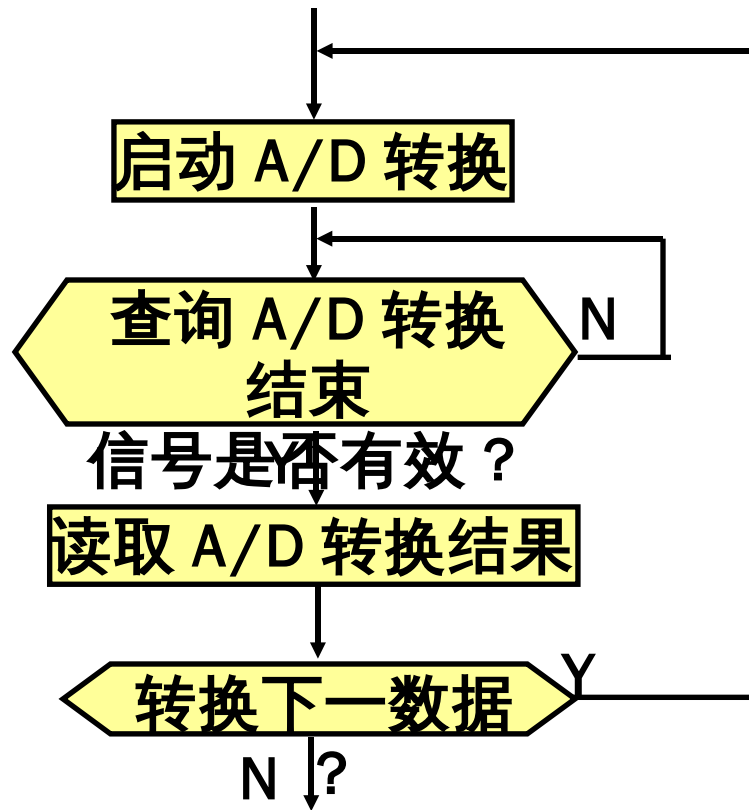


h 程序延时方式下，  
硬件连线上未利用转换完成信号

## □ 程序查询方式

转换完成 EOC 信号通过并行端口，送入 CPU。  
在 CPU 启动 A/D 转换之后，  
CPU 不断查询 A/D 的转换结束信号，  
一旦该信号有效，CPU 读取 A/D 的转换结果。



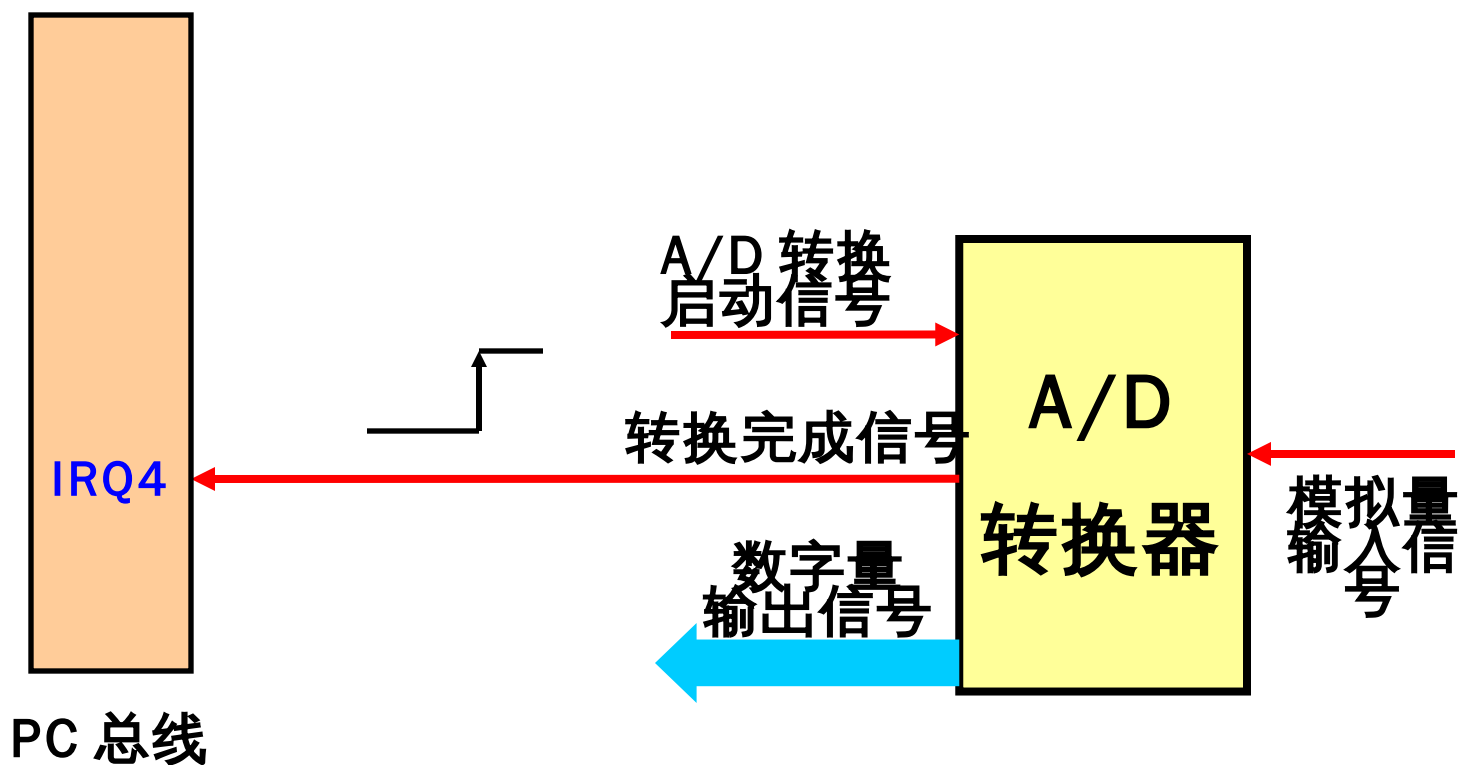


## 程序查询方式流程

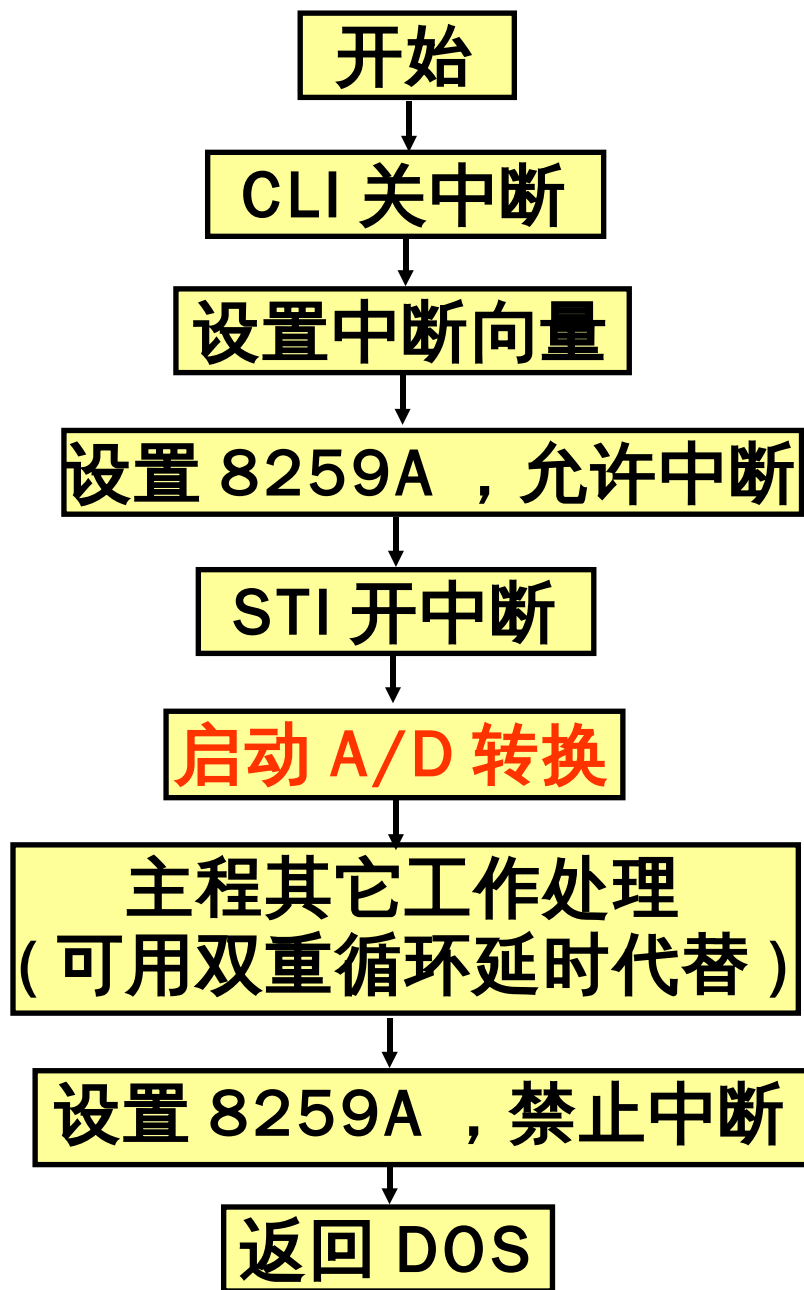


## 中断方式

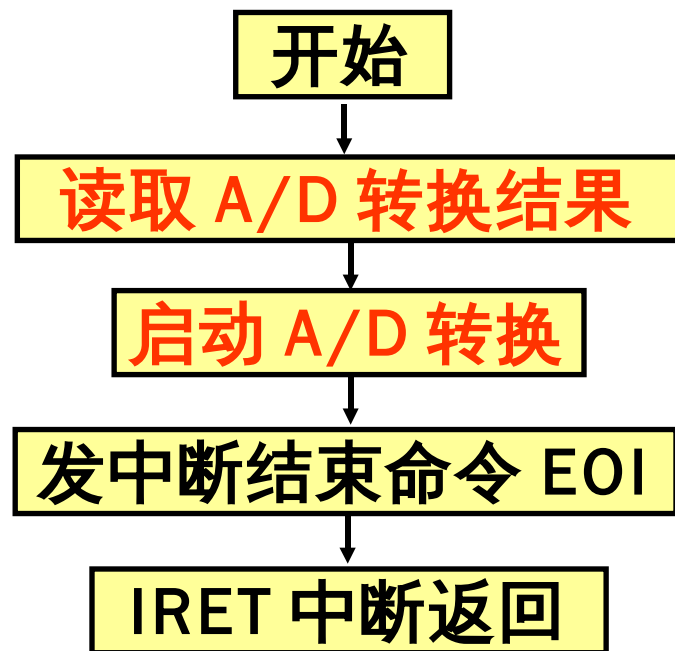
用 A/D 转换结束信号向微机系统发中断申请，  
CPU 采用中断方式读取 A/D 转换结果。



## 主程序



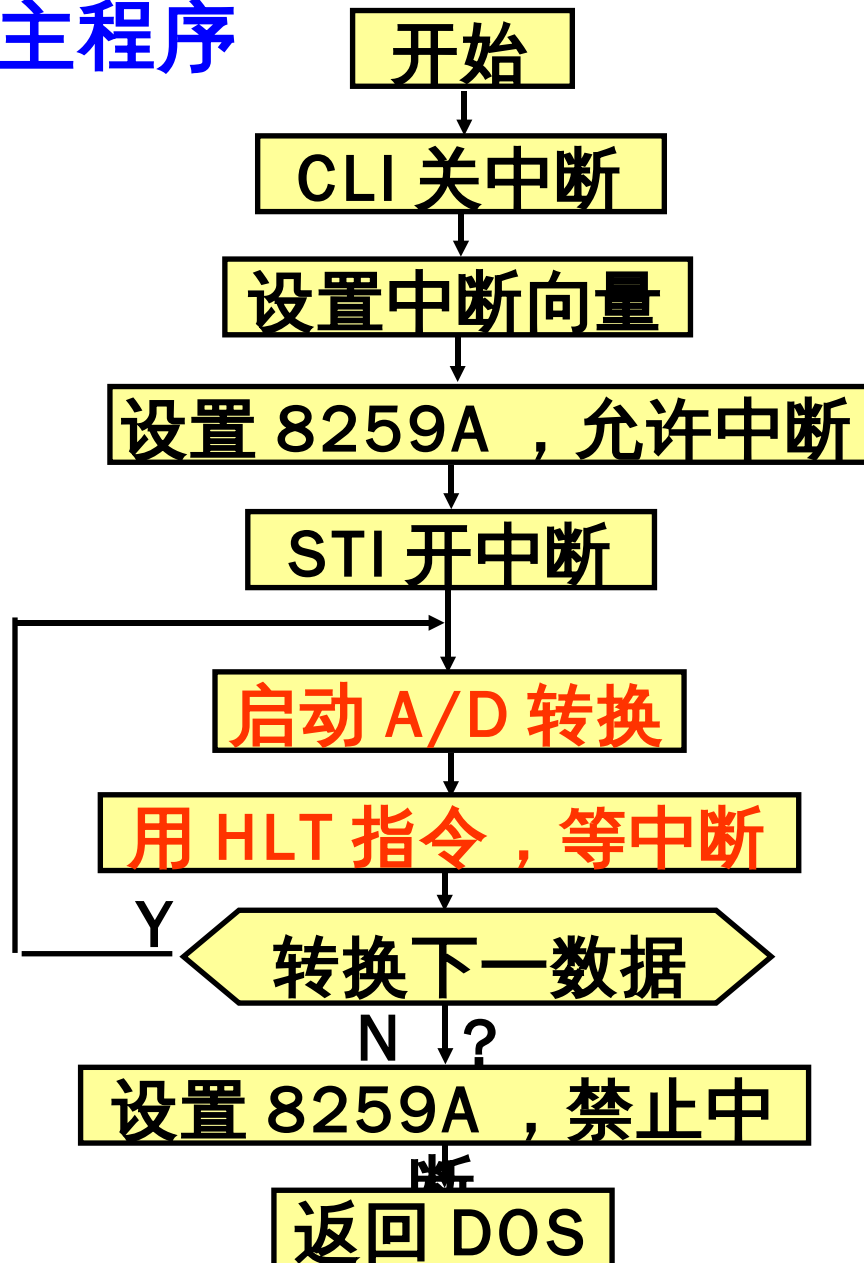
## 中断子程



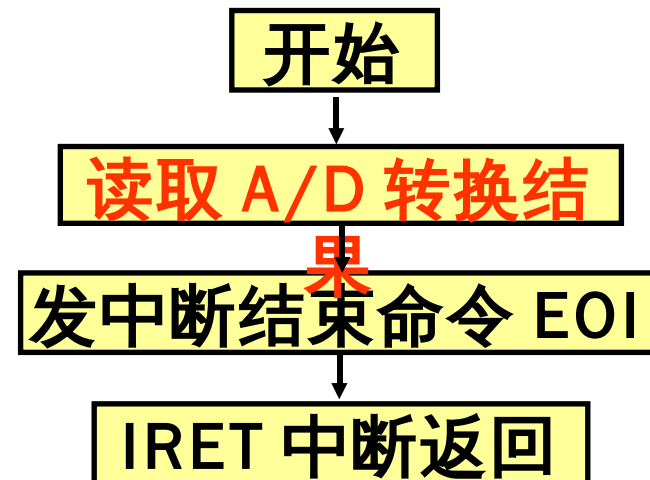


# 或使用 HLT 指令

## 主程序

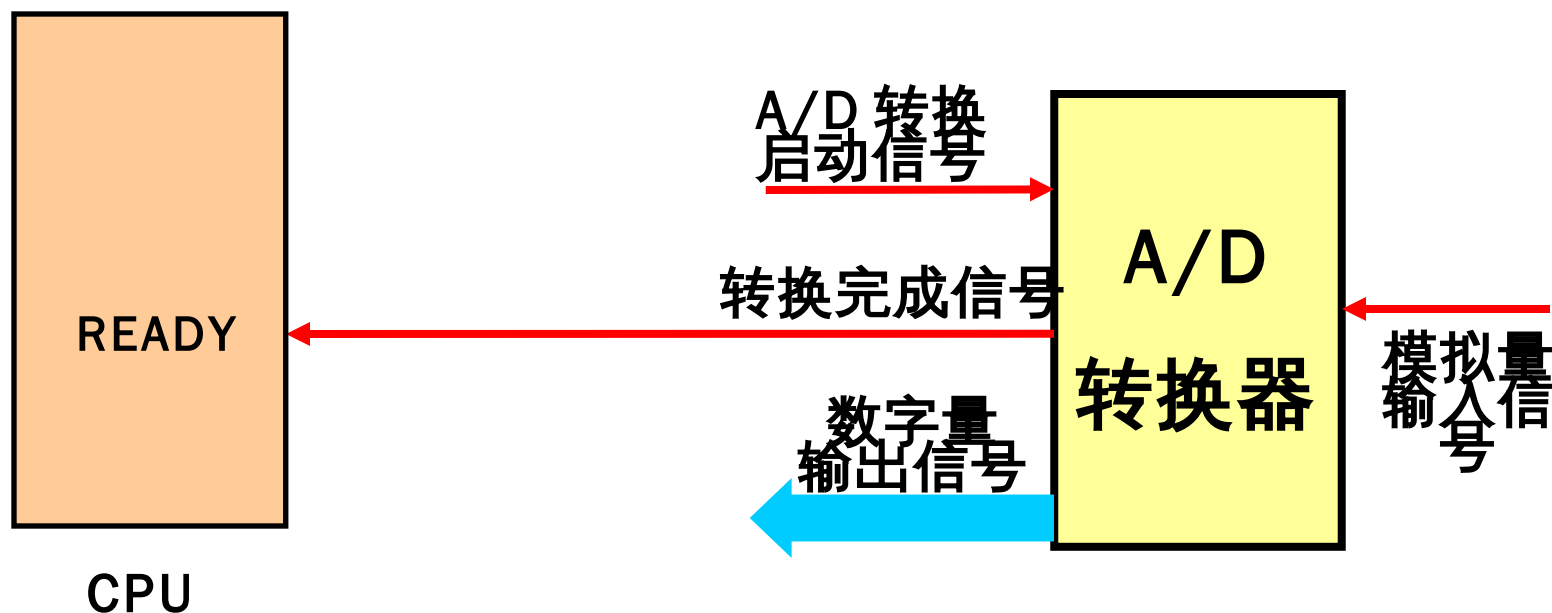


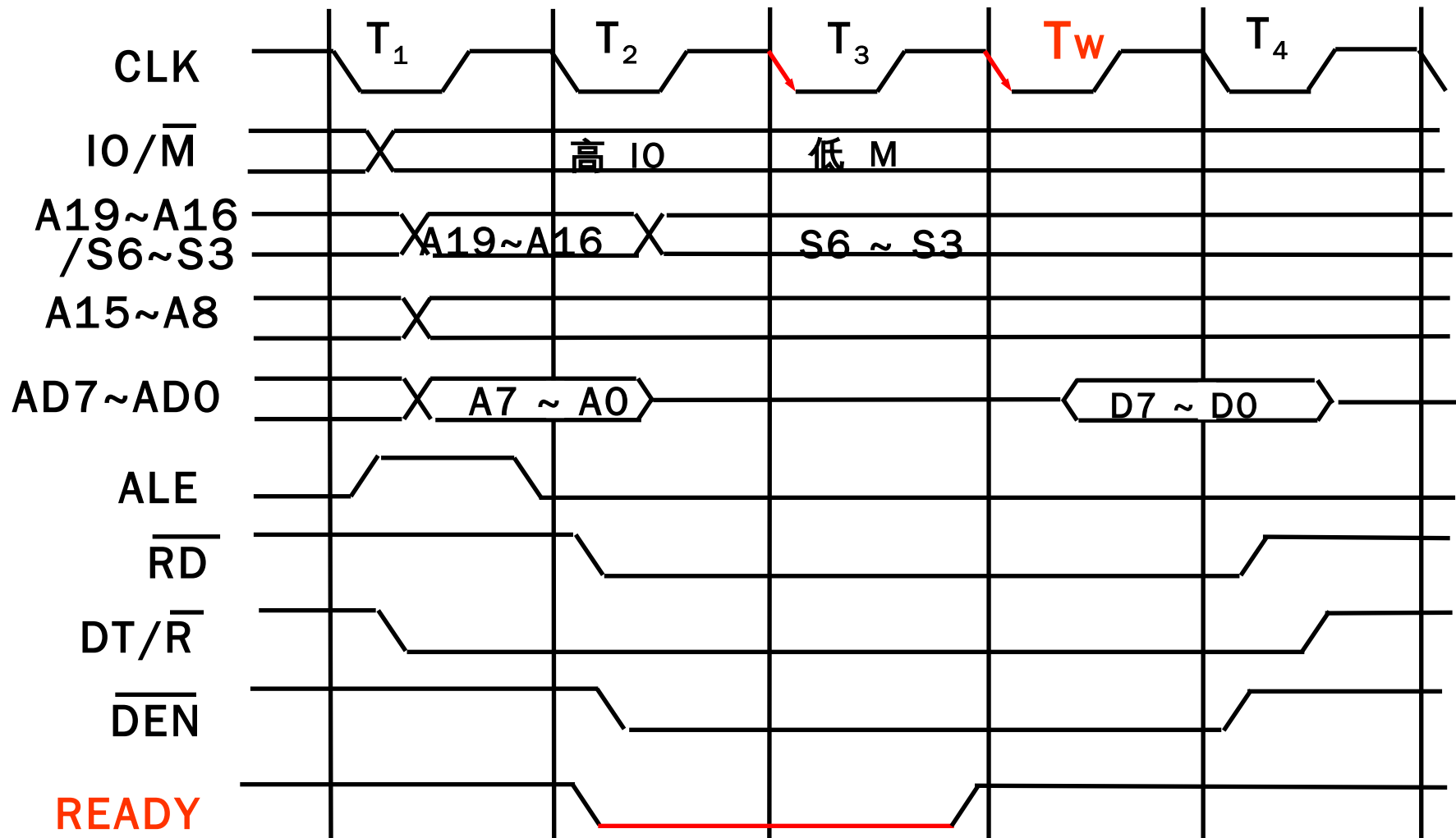
## 中断子程



## □ 等待方式

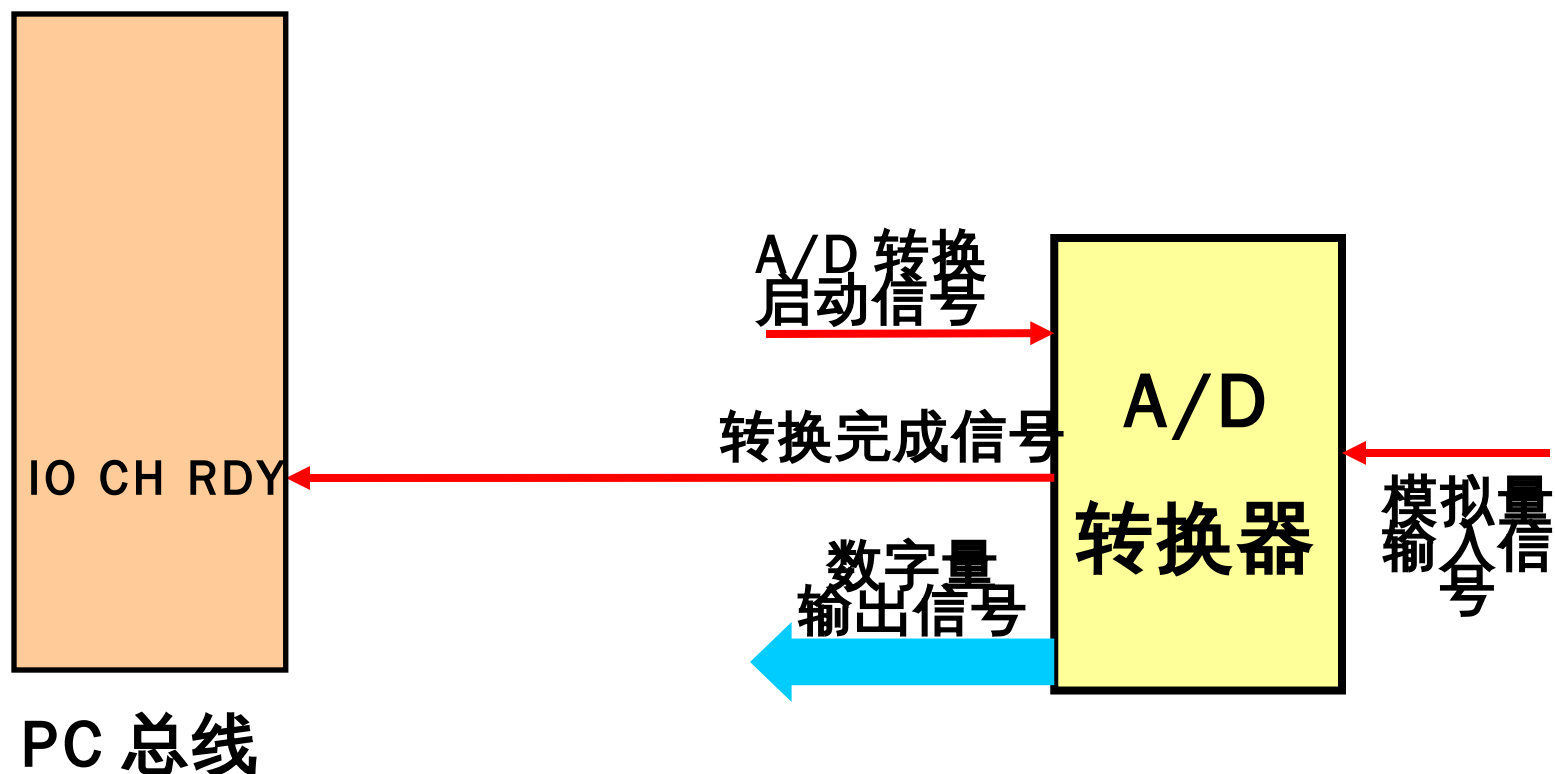
用 A/D 转换结束信号向微机系统发 READY 信号，当 CPU 用 IN 读指令读取结果时，在未转换完成前，READY 信号为低电平，将延长 CPU 的读 I/O 端口周期；待 READY 有效，即 A/D 转换完成之后，再读取转换结果。

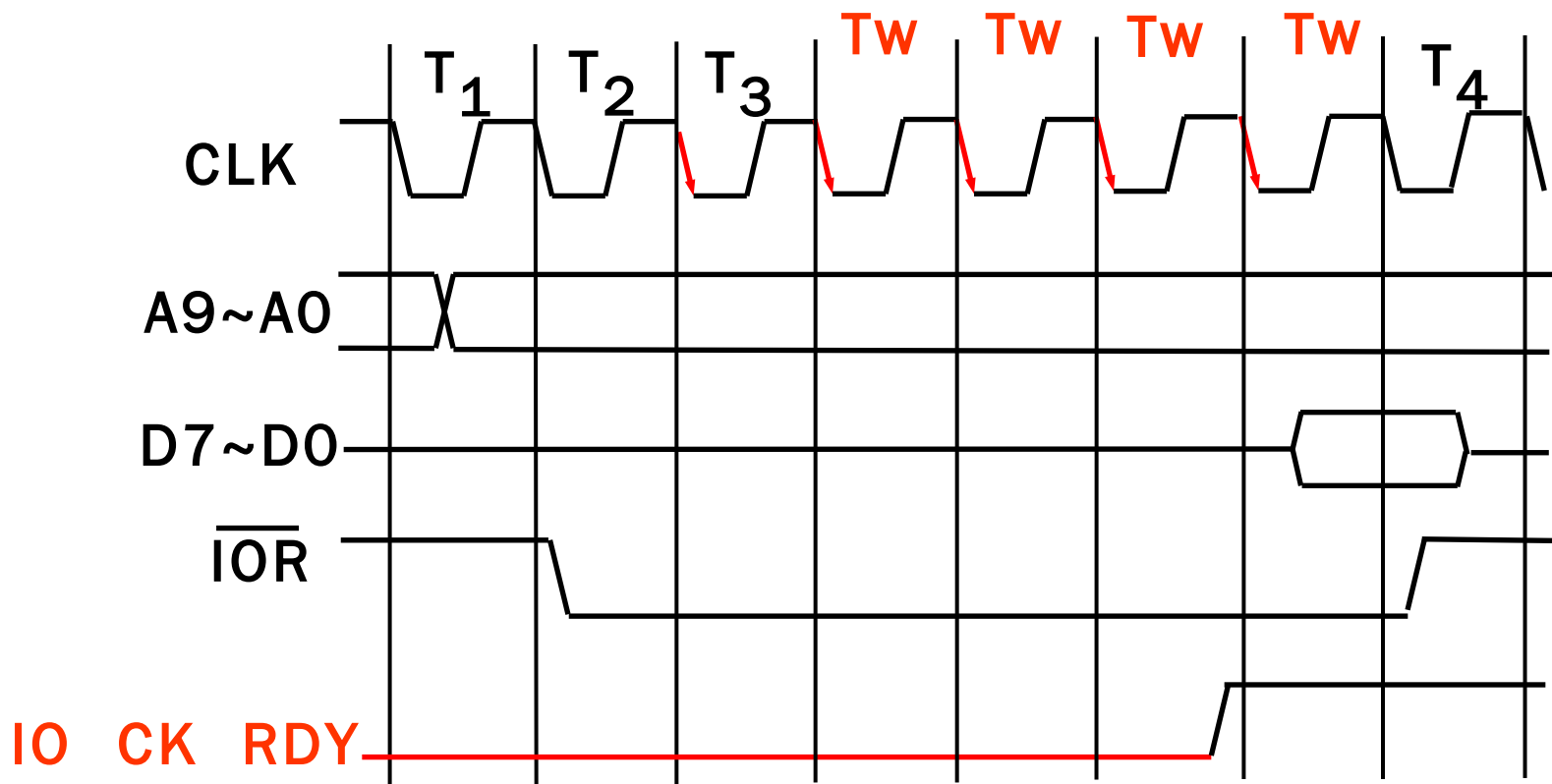




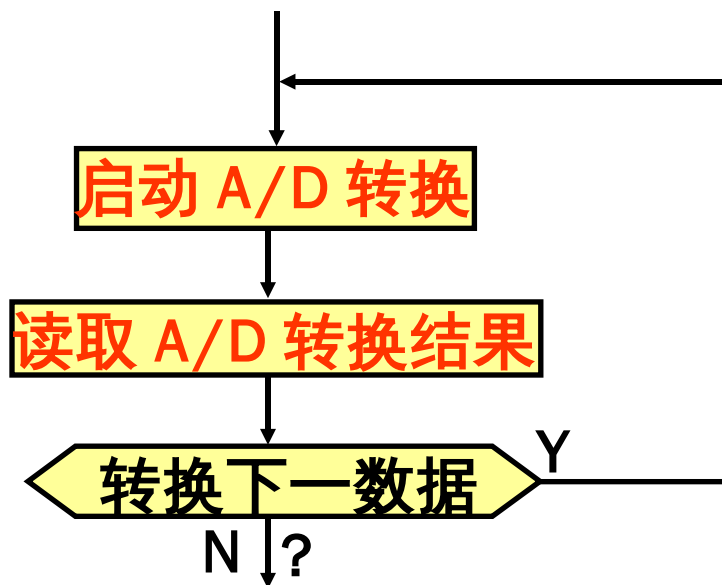
8088CPU 最小模式下，  
 在  $T_3$  状态下降沿，检测 READY 信号，  
 当 **READY 为低电平**，增加  $T_w$  状态；  
 直到 **READY 为高电平**，才进入  $T_4$  状态。

# 利用 IBM PC/XT 总线上的 IO CH RDY 信号 ：

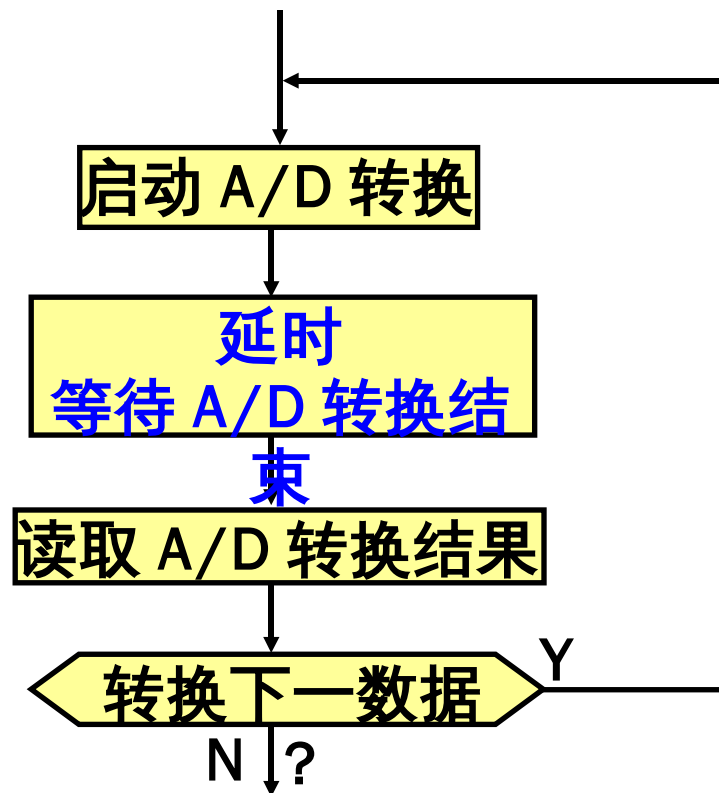




在  $T_3$  状态下降沿，检测 IO CK RDY 信号，  
 当 IO CK RDY 为低电平，增加  $T_w$  状态  
 直到 IO CK RDY 为高电平，才进入  $T_4$  状态



等待方式流程



程序延时方式流程

思考：

程序延时方式和等待方式在硬件连线和软件上的不同之处？