

*Electronics Information Engineering*  
**Sichuan University EI**

*Principle of Microcomputer and Interface Technology*

# 微机原理与接口技术

## 第8章 并行接口



# 教学重点

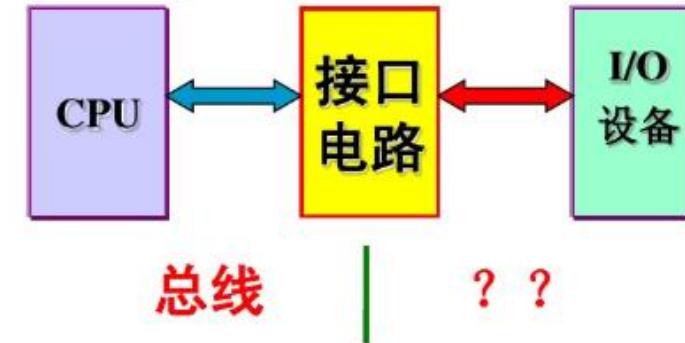
● 并行传输方式及并行接口

● 8255的工作方式

● 8255的编程控制及应用

## 8.1 什么是通信？

- **通信：**计算机中主机与主机，  
主机与外设之间信息的传送。



- 一般分为**并行通信**和**串行通信**。
- **并行通信：**接口与外设通过**多根**（一般为计算机字长）**数据线同时传送多位数据**。
  - 效率高，但需要考虑线间干扰。 不适用于远传
- **串行通信：**接口与外设通过**一根数据线逐位**传送数据。
  - 可靠性高，但（以前）效率低。

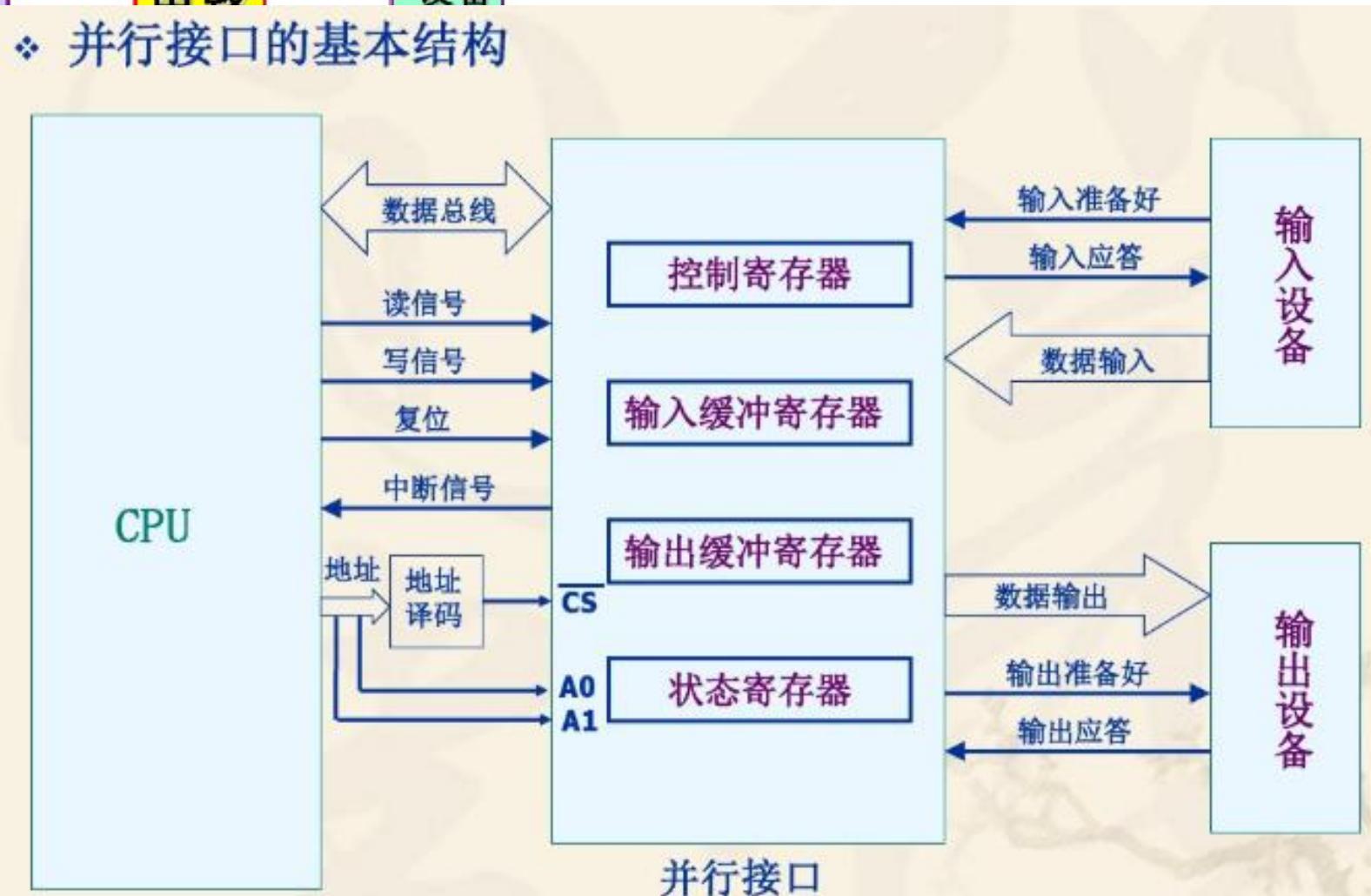
- 适合于外部设备与微机之间进行近距离、大量和快速的信息交换。
- 微机系统中最基本的信息交换方法
  - 例如：系统板上各部件之间，接口电路板上各部件之间。

# 并行接口的功能



◆ 并行接口的基本结构

- 具有一
- 每个端
- 功能。
- CPU与
- 信息。



## 8.2 并行接口电路8255A

- 具有多种功能的可编程并行接口电路芯片
  - 最基本的接口电路“三态缓冲器”和“锁存器”
  - 与CPU、与外设间的接口电路：状态寄存器和控制寄存器
  - 还有端口的译码和控制电路、中断控制电路
- 分3个端口，共24个外设引脚，2组控制
- 共三种输入输出工作方式



## 8.2.1 8255A的外部引脚

- 40引脚双列直插，+5V，24

条可编程I/O引脚

- 除电源和地线外，其他外部

引脚可分为两组：

- 一组面向CPU的信号； $DB_{0-1}$ ,  $AB_{0-1}$

- 一组面向外设的信号。 $\overline{RD}$ ,  $\overline{WR}$

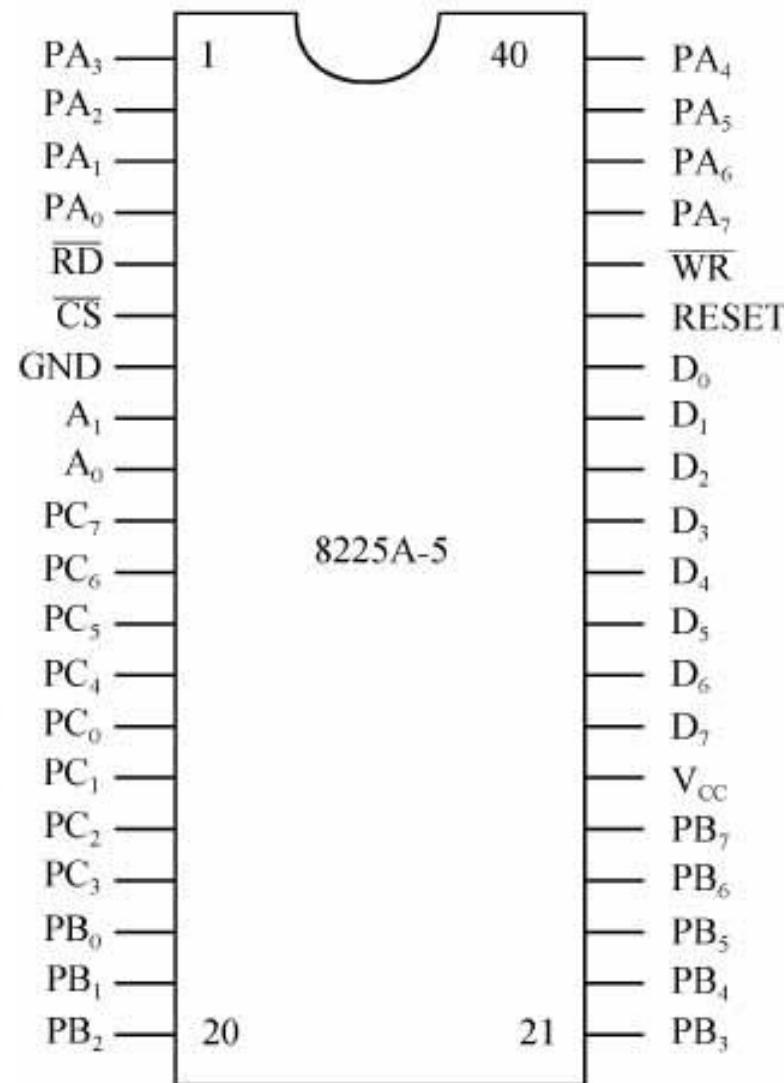


## 8.2.1 8255A的外部引脚

- 连接CPU系统端的主要引线：

- $D_0 \sim D_7$  数据线
- $A_0 \sim A_1$  地址线
- $\overline{CS}$  片选信号
- $\overline{RD}$  读信号
- $\overline{WR}$  写信号
- RESET 复位信号

$A_1$	$A_0$	选择
0	0	端口A
0	1	端口B
1	0	端口C
1	1	控制寄存器



## 8.2.1 8255A的外部引脚

- 连接外设的引脚：

- PA<sub>0</sub>~PA<sub>7</sub>

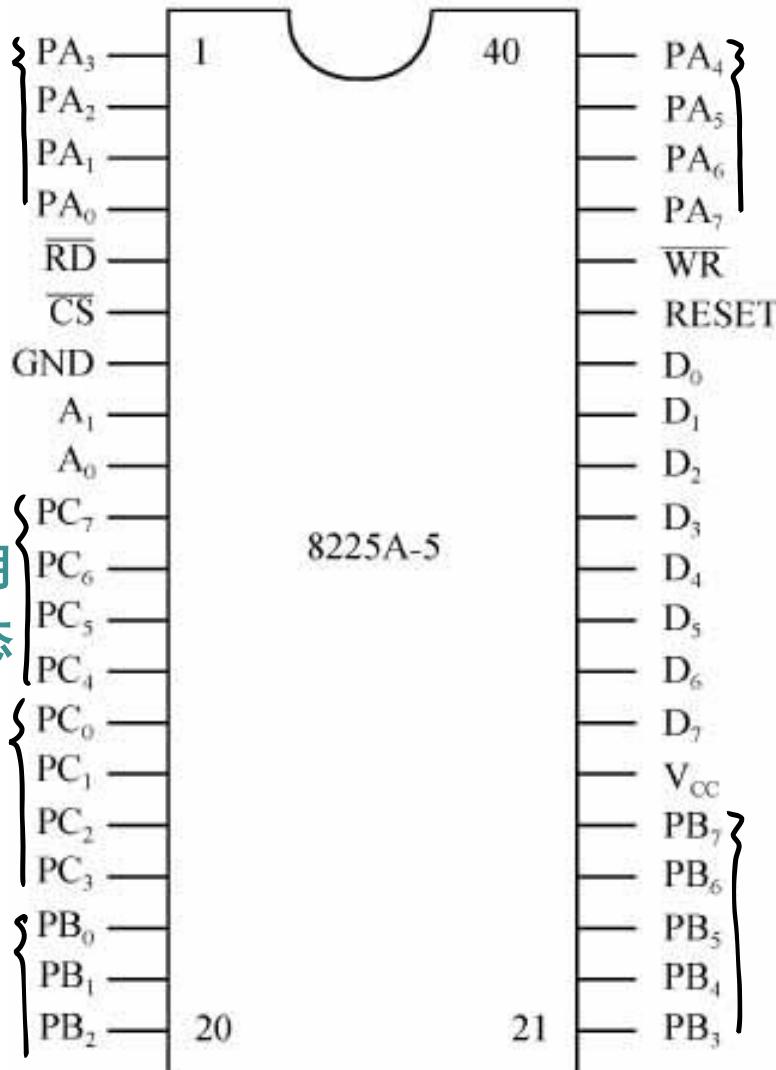
- PB<sub>0</sub>~PB<sub>7</sub>

- PC<sub>0</sub>~PC<sub>7</sub>

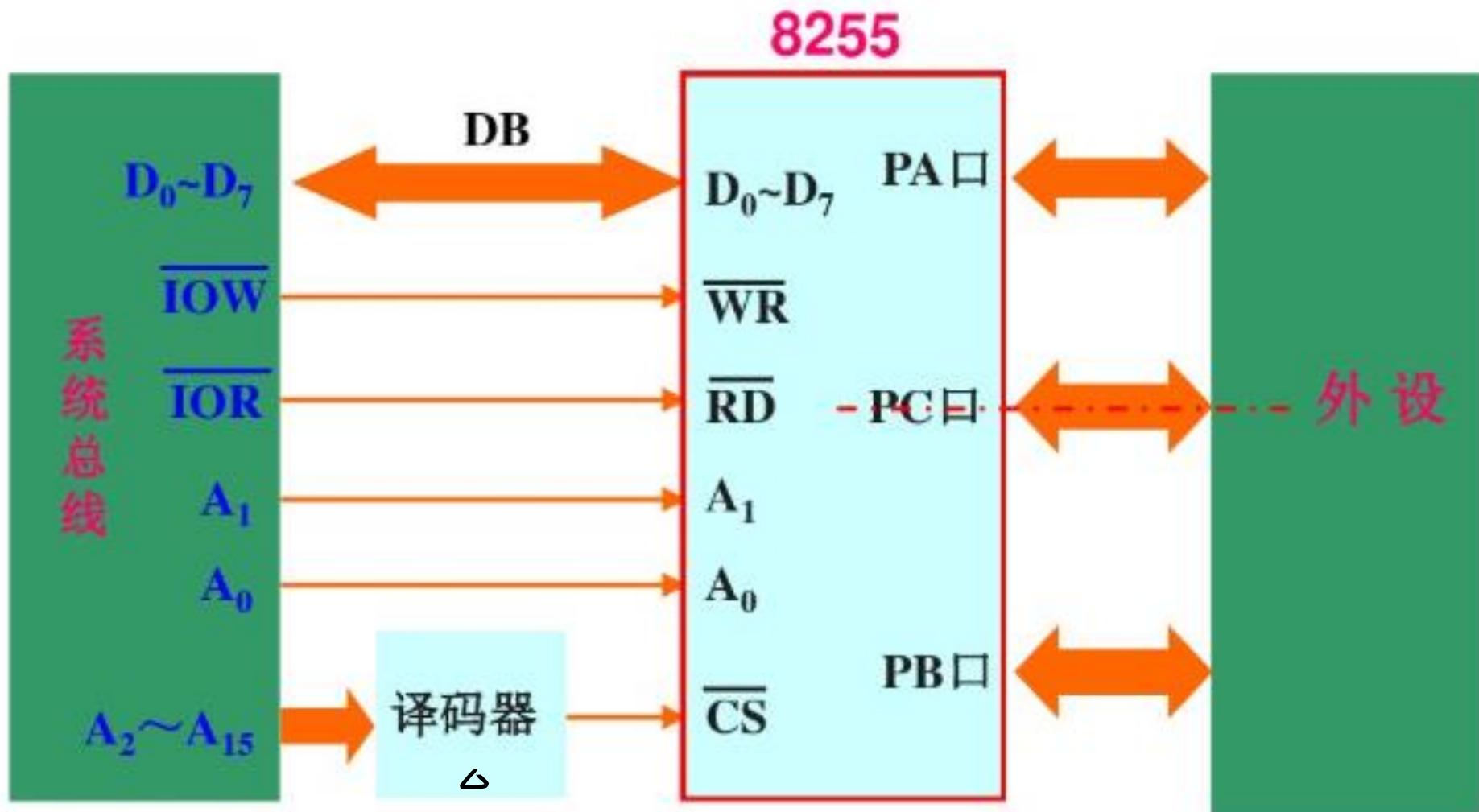
- 三个端口可通过编程分别制定为输入或输出口。

其中，C口即可用作独立的输入/输出口，也可用作A、B口的握手联络信号（控制信号输出或状态信号输入）

CS A1 A0	I/O地址	读操作RD	写操作WR
0 0 0	60H	读端口A	写端口A
0 0 1	61H	读端口B	写端口B
0 1 0	62H	读端口C	写端口C
0 1 1	63H	非法	写控制字

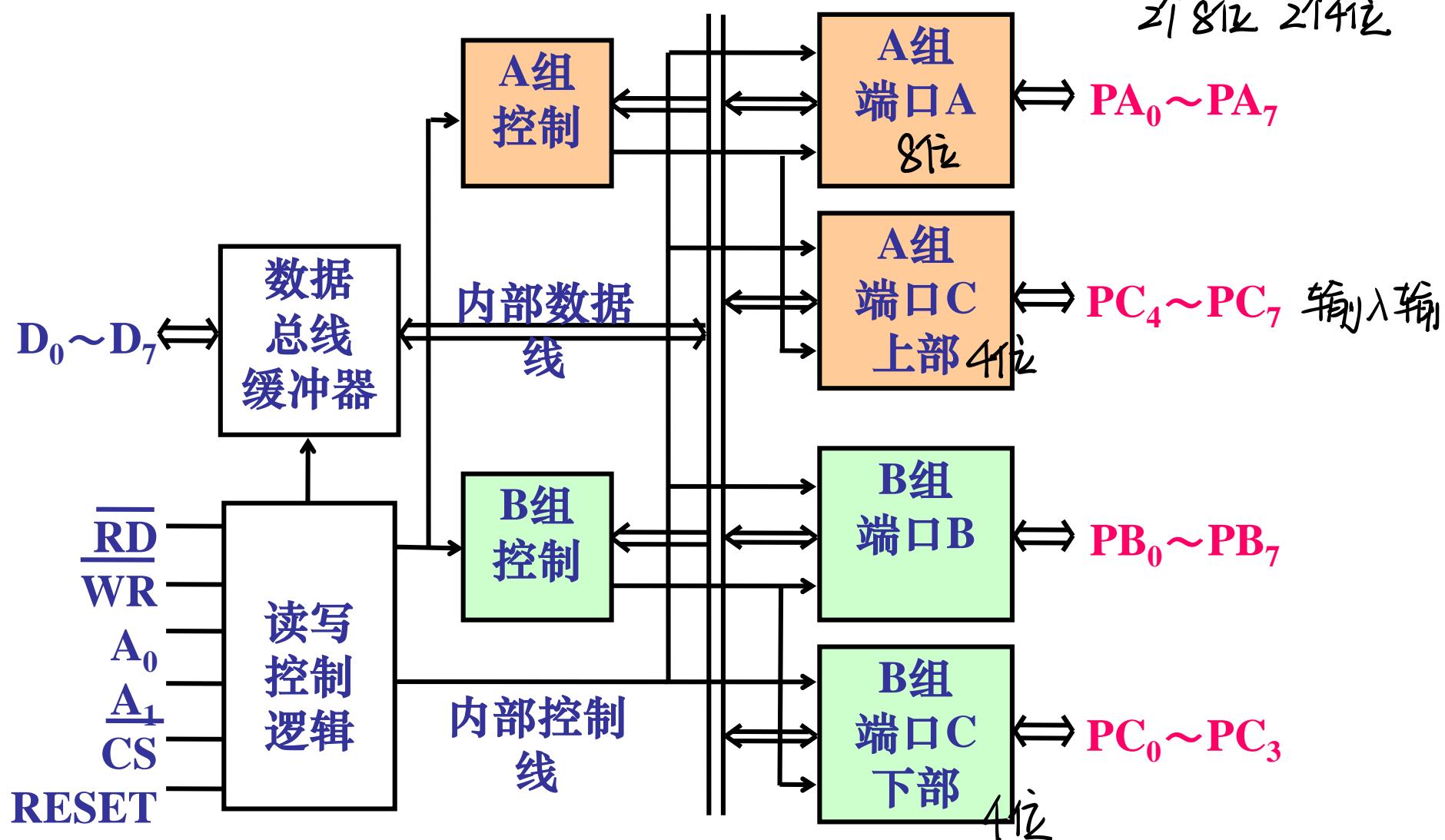


## 8.2.1 8255A与系统的连接图



## 8.2.2 8255A的内部结构

2组控制



## 8.2.2 8255A的内部结构

- 8255A内部分为A、B两组分别进行控制管理
- A组 {
  - 端口PA
  - 端口PC的高四位
- B组 {
  - 端口PB
  - 端口PC的低四位
- PA口和PB口的输入/输出都具有数据锁存能力；PC口输出有锁存能力，而输入没有锁存能力（仅有缓冲能力）。

## 8.2.3 8255A的外设数据端口

A,B主要传输数据

- 端口A:  $PA_0 \sim PA_7$ 
  - A组, 支持工作方式0、1、2
  - 常作数据端口, 功能最强大
- 端口B:  $PB_0 \sim PB_7$ 
  - B组, 支持工作方式0、1
  - 常作数据端口
- 端口C:  $PC_0 \sim PC_7$ 
  - 仅支持工作方式0
  - A组控制 $PC_4 \sim PC_7$ , B组控制 $PC_0 \sim PC_3$
  - 分成两个4位端口
  - 可作数据、状态和控制端口
  - 控制最灵活, 最难掌握

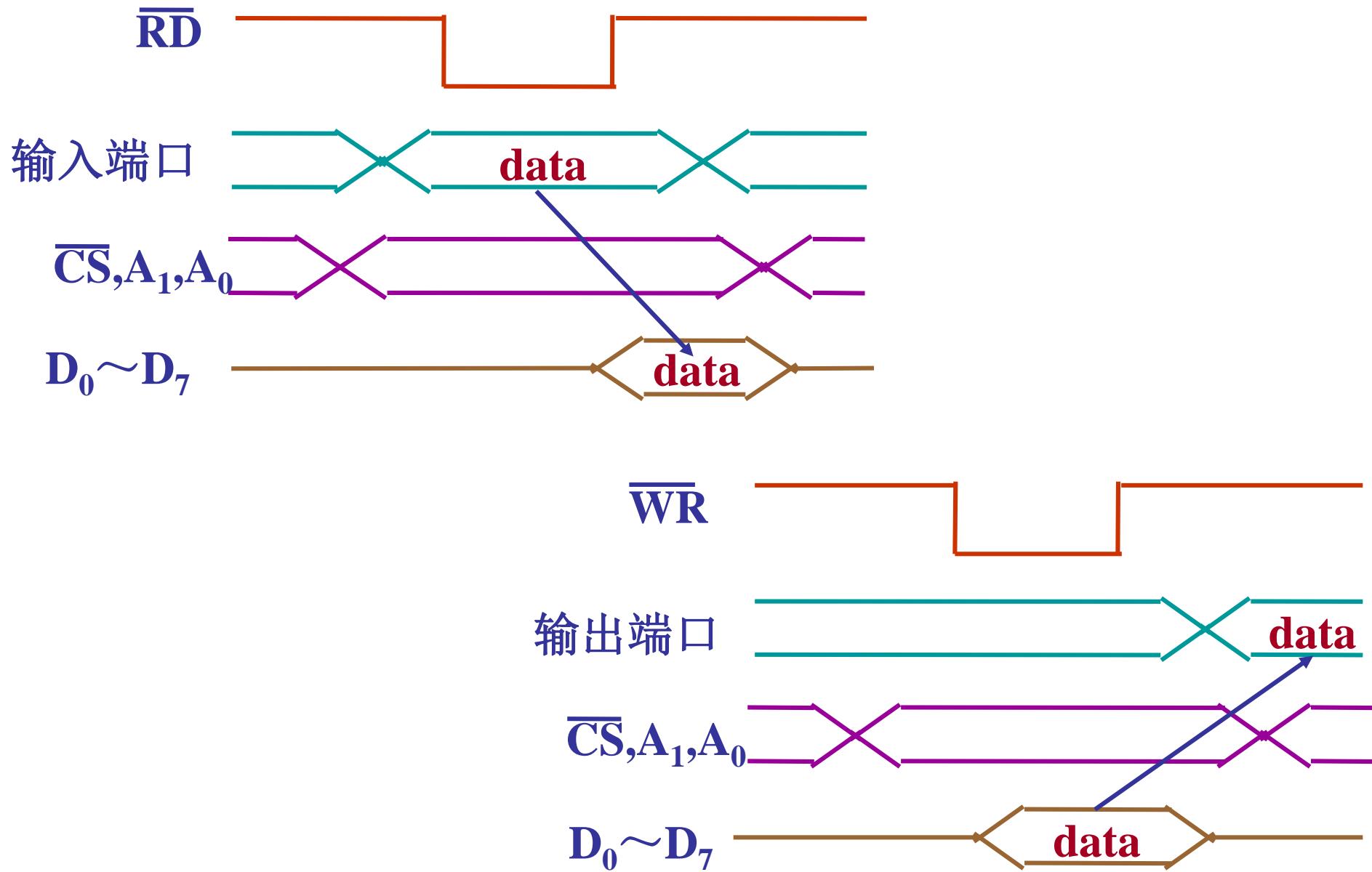
## 8.3 8255A的工作方式

- 方式0：基本输入输出方式
  - 适用于无条件传送和查询方式的接口电路
- 方式1：选通输入输出方式
  - 适用于查询和中断方式的接口电路
- 方式2：双向选通传送方式
  - 适用于与双向传送数据的外设
  - 适用于查询和中断方式的接口电路

## 8.3.1 工作方式0—基本输入输出工作方式

- 8255A的24条I/O线全部由用户分配功能。
- 端口信号线之间无固定的时序关系。
- 输出锁存，输入只有缓冲能力而无锁存功能。
- 单向 I/O，一次初始化只能指定某一端口作输入或作输出，不能使该端口同时既作输入又作输出。
- 分成彼此独立的两个8位和两个4位并行口。
- 只能把C口的高4位为一组或低4位为一组同时输入或输出。

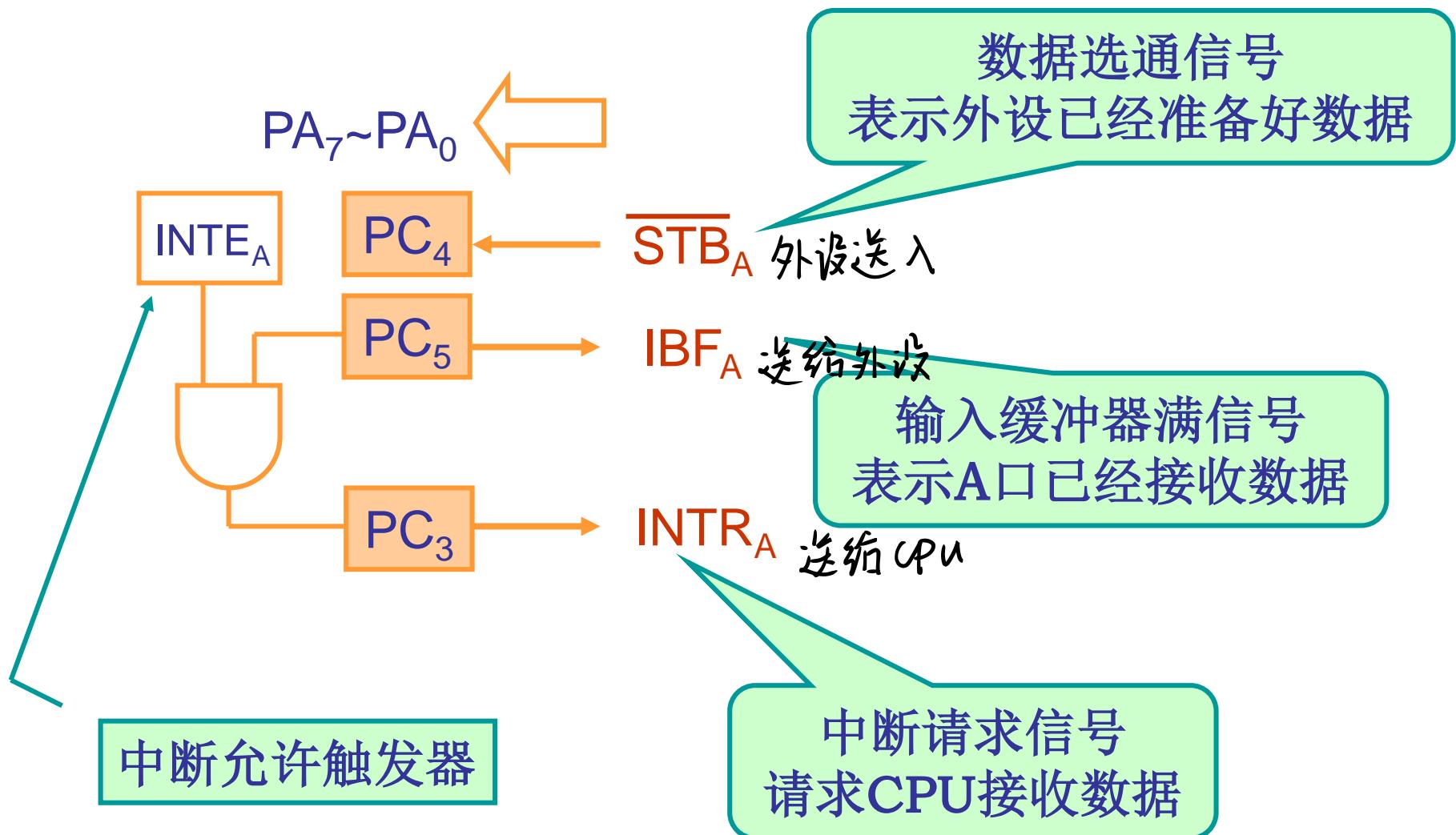
## 8.3.1 方式0时序



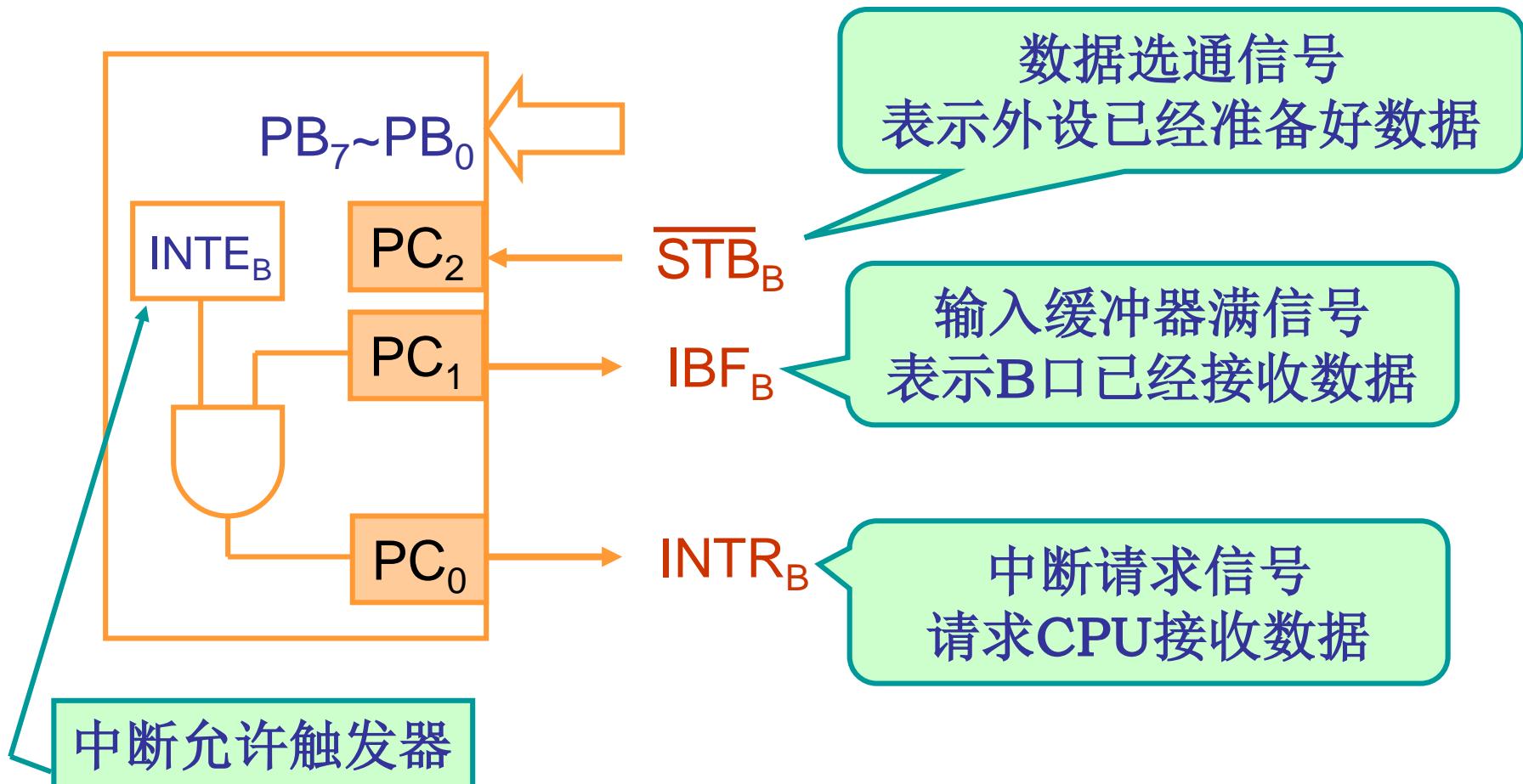
## 8.3.2 工作方式1—选通输入输出方式

- 在面向I/O设备的24根线中，设置专用的中断请求和联络信号线。
- 数据的输入输出都有锁存能力。
- PA和PB为数据口，而PC口的大部分引脚分配作联络信号用，用户对这些引脚不能再指定其它用途。
- 各联络信号线之间有固定的时序关系，传送数据时要严格按照时序进行。
- 输入/输出操作产生确定的状态字，这些状态信息可作为查询或中断请求之用。

## 8.3.2.1 方式1输入引脚: A端口



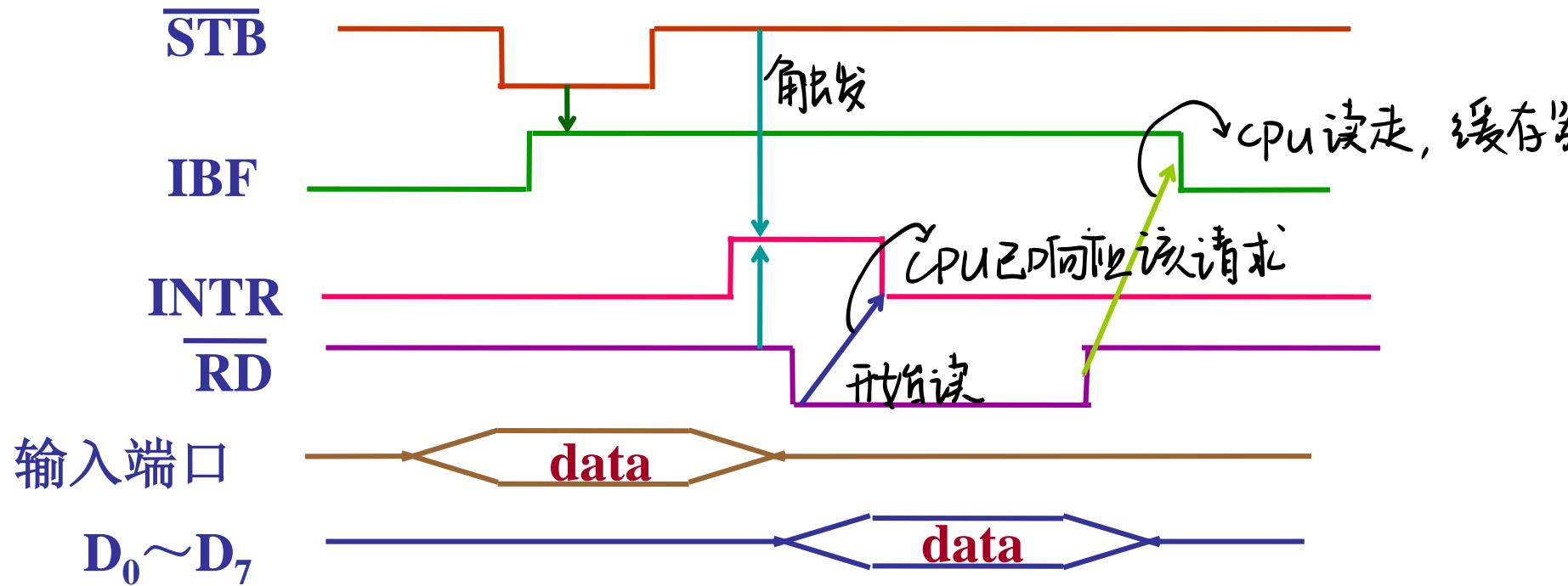
## 8.3.2.1 方式1输入引脚：B端口



### 8.3.2.1 方式1输入联络信号

- STB——选通信号, 低电平有效
  - 由外设提供的输入信号;
  - 当其有效时, 将输入设备送来数据保存至8255A的输入缓冲器。
- IBF——输入缓冲器满信号, 高电平有效
  - 8255A输出的联络信号;
  - 有效时, 表示数据已保存在输入缓冲器。
- INTR——中断请求信号, 高电平有效
  - 8255A输出的信号, 可用于向CPU提出中断请求, 要求CPU读取外设数据。

## 8.3.2.1 方式1输入时序



**STB**和**IBF**是外设和8255A间的一对对应答联络信号，目的是为可靠地输入数据。

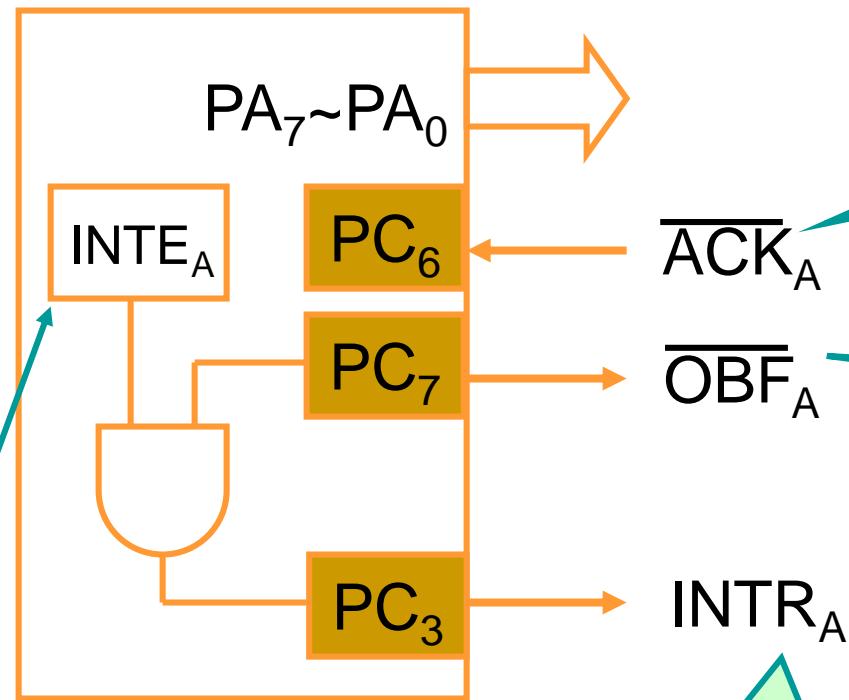


若CPU采用**查询**方式，则通过查询状态字中的INTR位或IBF位是否置位来判断有无数据可读

## 8.3.2.1 方式1输入时序说明——以端口A为例

- ① 当输入设备已准备好一个新数据时, 首先检测IBF线的状态, 若为低(表示输入锁存器空), 则输入设备将数据放入 $\text{PA}_7 \sim \text{PA}_0$ 。
- ② 然后发出选通信号 $\overline{\text{STB}}$ ,  $\overline{\text{STB}}$ 将 $\text{PA}_7 \sim \text{PA}_0$ 上的数据置入数据缓冲器, 此时8255A会使IBF有效, 指示输入缓冲器已满。
- ③  $\overline{\text{STB}}$ 变为无效后, 8255A使INTR由低变高, 通过8259A向CPU发出中断请求, CPU响应请求并进入相应的中断服务程序。
- ④ 在中断服务程序中, CPU读端口A, RD的下降沿使INTR无效。
- ⑤ RD的上升沿使IBF变无效, 指示输入缓冲器的数据已传送给CPU, 处于"空"状态, 准备接收新的数据。

## 8.3.2.2 方式1输出引脚：A端口



外设响应信号  
表示外设已经接收到数据 外设已

输出缓冲器满信号  
表示CPU已经输出了数据

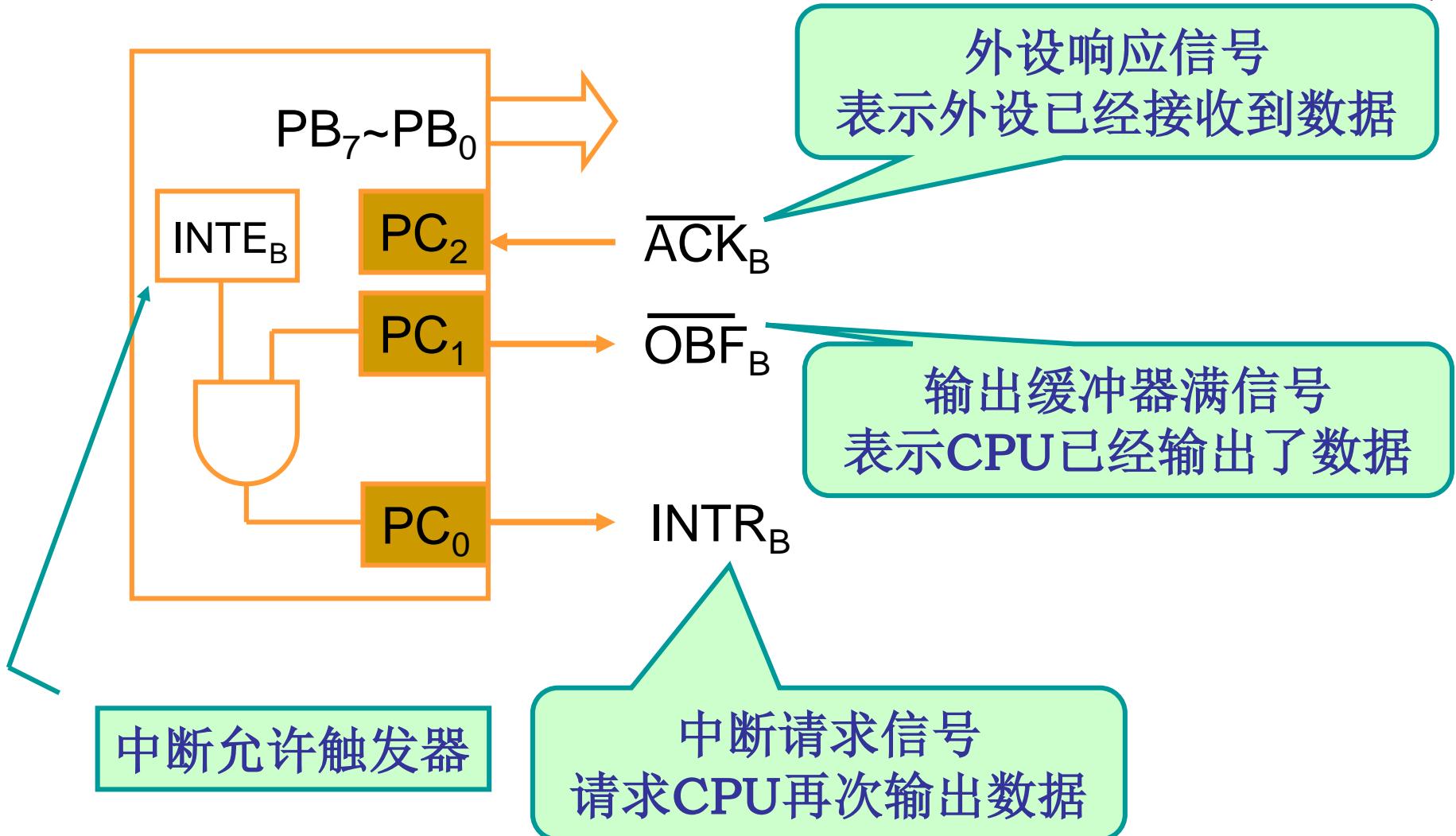
中断允许触发器

中断请求信号 空时  
请求CPU再次输出数据

A可以同时入

## 8.3.2.2 方式1输出引脚：B端口

B不可同时入



### 8.3.2.2 方式1的输出联络信号

- OBF——输出缓冲器满信号, 低有效

- 8255A输出给外设的控制信号;
- 当其有效时, 表示CPU已把数据输出给指定的端口, 外设可以取走。

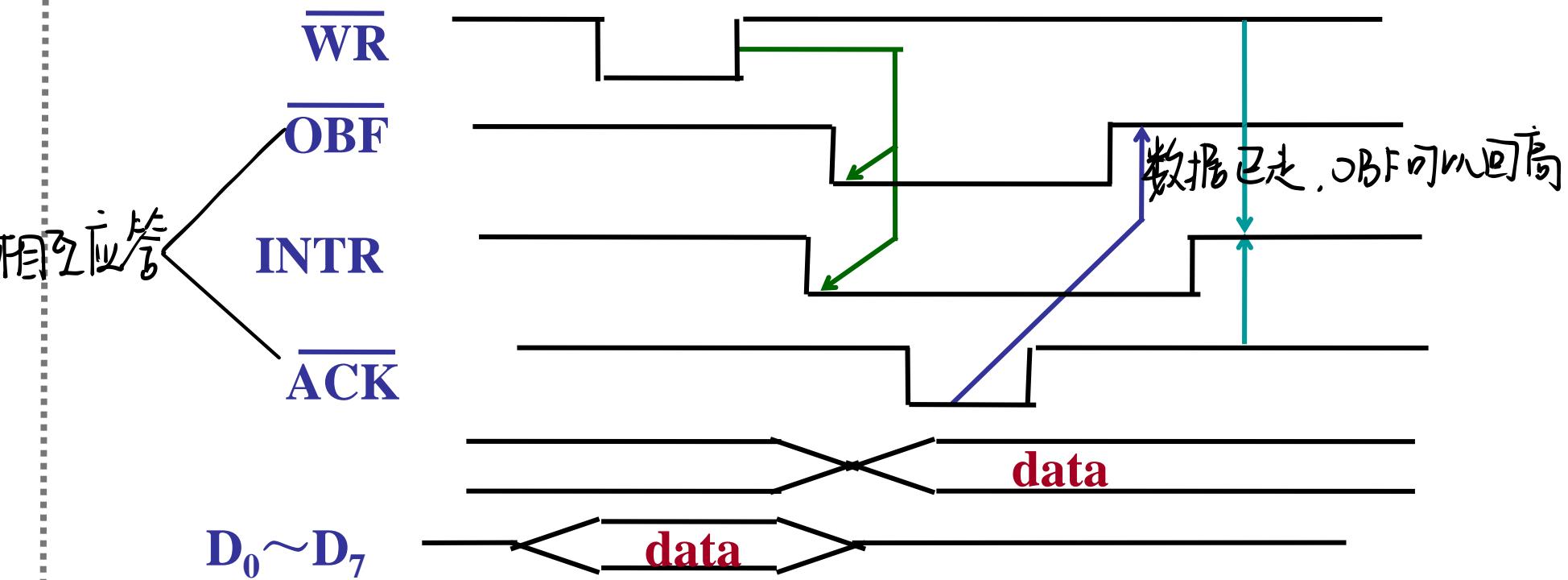
- ACK——响应信号, 低有效

- 外设的响应信号, 指示8255A的端口数据已由外设接受。

- INTR——中断请求信号, 高有效

- 当输出设备已接受数据后, 8255A输出此信号向CPU提出中断请求, 要求CPU继续提供数据。

## 8.3.2.2 方式1输出时序



OBF和ACK是外设和8255A间的一对对应答联络信号，目的是为可靠地输出数据。



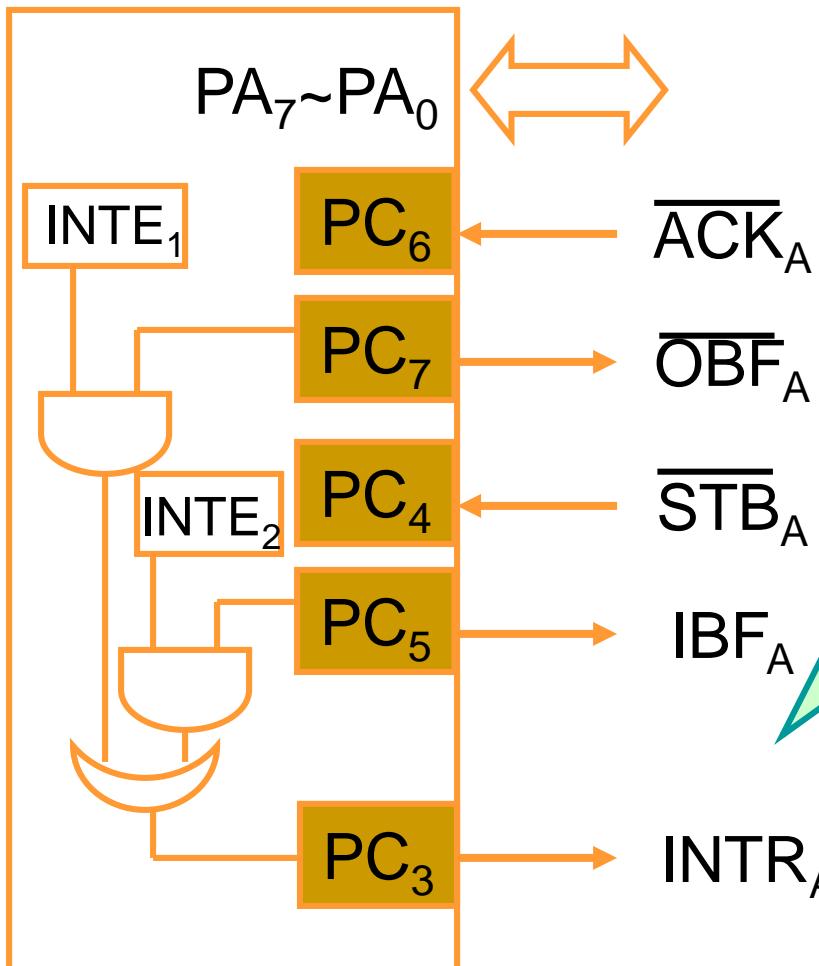
## 8.3.2.2 方式1输出时序说明——以端口A为例

- ① 以端口A为例, 当输出设备接收了前一次数据后, 8255A通过8259A向CPU请求中断。
- ② 在中断服务程序中, CPU将数据写入指定的端口,  $\overline{WR}$ 变为有效, 于是数据写入输出缓冲器并出现在 $PA_7 \sim PA_0$ 。
- ③ 写信号 $\overline{WR}$ 的上升沿使OBF变为有效, 指示输出设备 $PA_7 \sim PA_0$ 上已有新的数据, 同时还使INTR变为无效。
- ④ 输出设备通过 $PA_7 \sim PA_0$ 接收数据。
- ⑤ 输出设备接收到数据后, 向8255A发回信号 $\overline{ACK}$ ,  $\overline{ACK}$ 的上升沿通常表示输出设备已准备好再接收新的数据。
- ⑥ 8255A在 $\overline{ACK}$ 出现上升沿之前使OBF变为无效, 若INTE=1, 则还使INTR变为有效, 再次请求中断。

### 8.3.3 方式2：双向选通输入输出方式

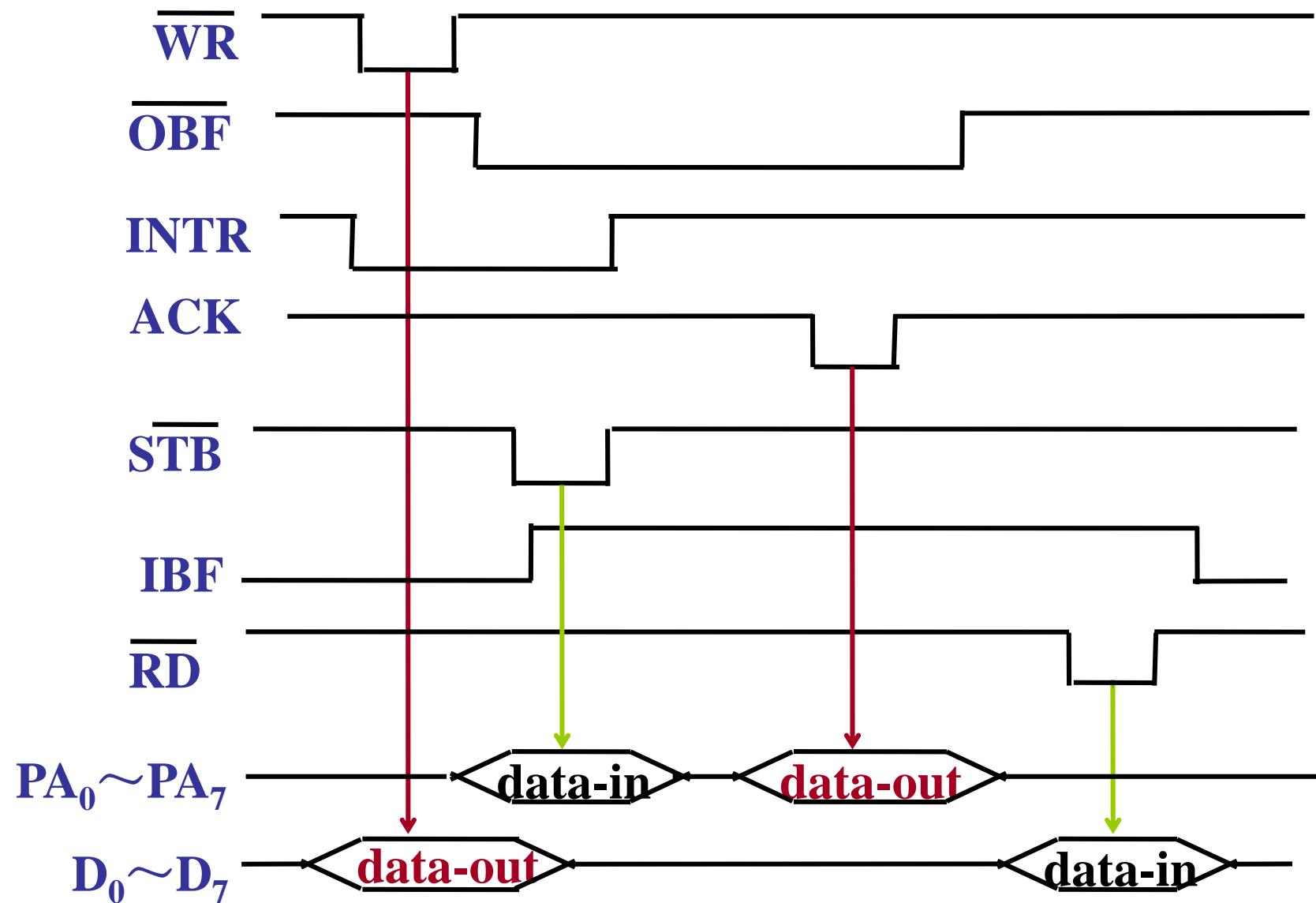
- 方式2将方式1的选通输入输出功能组合成一个双向数据端口，可以发送数据和接收数据；
- 只有端口A可以工作于方式2，需要利用端口C的5根信号线，其作用与方式1相同；
- 时序关系基本上是方式1下输入和输出两种操作的组合。

### 8.3.3 方式2双向引脚



用PC<sub>6</sub>设置INTE<sub>1</sub>（输出）  
用PC<sub>4</sub>设置INTE<sub>2</sub>（输入）  
输入和输出中断通过或门输出INTR<sub>A</sub>信号

### 8.3.3 方式2时序



## 8.3.4 8255A工作方式总结

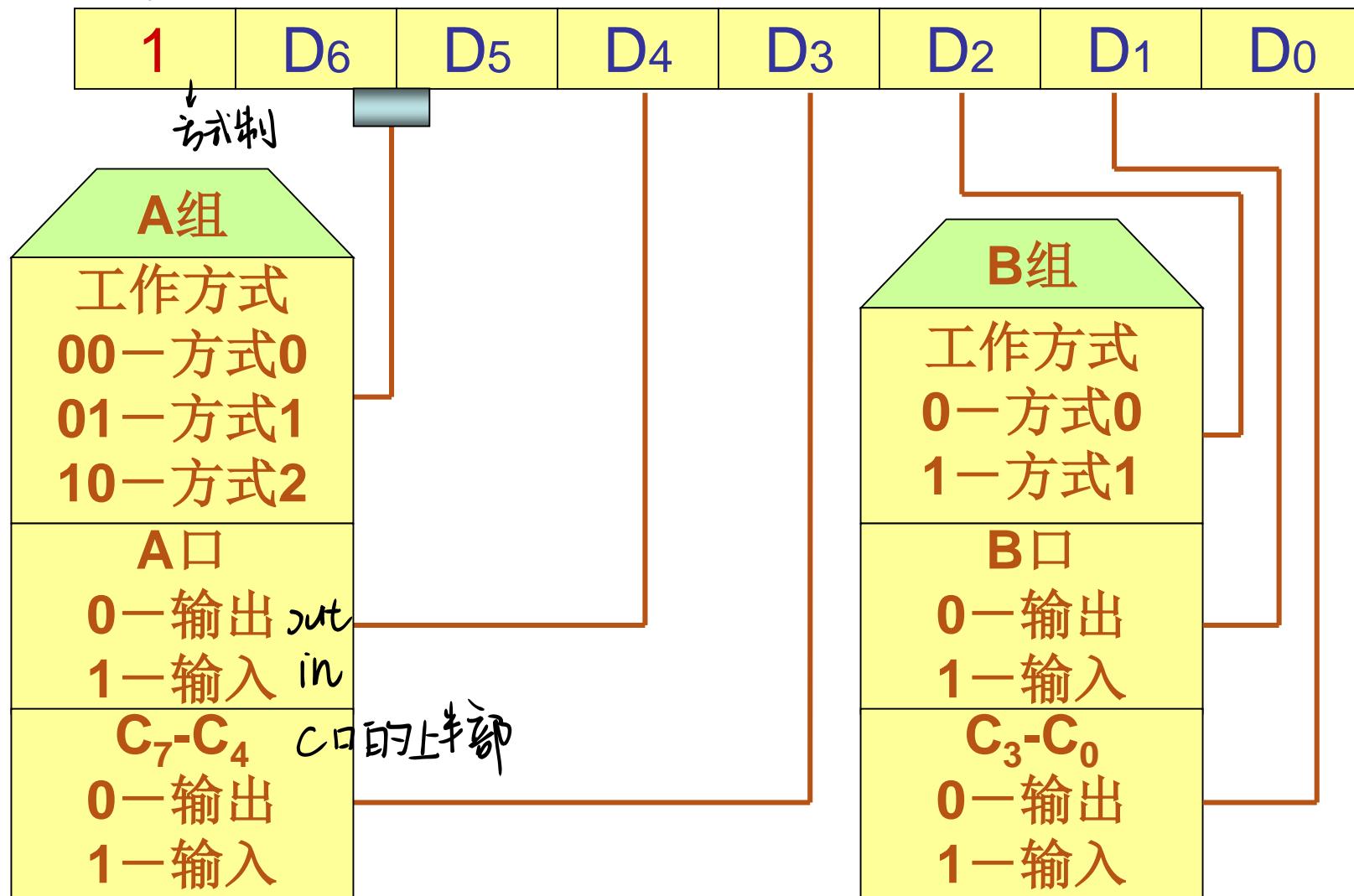
	方式0	方式1	方式2
名称	基本输入输出	选通输入输出	双向选通输入输出
传送方式	无条件	查询、中断	查询、中断
锁存	输入缓冲/输出锁存	输入、输出锁存	输入、输出锁存
联络线	无	有	有
状态字	无	有	有
传送方向	单向	单向	双向
可工作端口	A、B、C	A、B	A
连接方法	A、B、C口分别连接I/O线	A组： $PA_{0-7}$ 连I/O线 $PC_{3-7}$ 连联络线 B组： $PB_{0-7}$ 连I/O线 $PC_{0-2}$ 连联络线	A组： $PA_{0-7}$ 连I/O线 $PC_{3-7}$ 连联络线

## 8.4 8255A的应用编程

- 8255A的控制寄存器控制着它的3个端口的工作方式和数据传送方式。
- 8255A的编程：
  - 初始化编程：设置工作方式，写入控制端口
    - ( $A_1 A_0 = 11$ )
  - 工作过程中：读写数据端口
    - 数据读写利用端口A、B和C的I/O地址,  $A_1 A_0$ 依次为00、01、10。

## 8.4.1 8255A的初始化编程——方式控制字

方式控制字



## 8.4.1 8255A的初始化编程——方式控制字写入示例

- 要求

- A端口：方式0，输入
- B端口：方式1，输出
- C端口上半部：输入
- C端口下半部：输出

- 方式控制字：10011100B或9BH

- 初始化的程序段：

```
mov dx, 63h          ; 假设控制端口为63H  
mov al, 9bh          ; 方式控制字  
out    dx, al        ; 送到控制端口
```

## 8.4.2 读写数据端口——通过各个数据端口进行

- 初始化编程后：
  - 当数据端口作为输入接口时, 执行输入IN指令得到外设数据;
  - 当数据端口作为输出接口时, 执行输出OUT指令将数据送给输出设备。
- 8255A具有锁存输出数据的能力
  - 对输出方式的端口同样可以输入;
  - 但不是读取外设数据, 而是读取上次CPU给外设的数据。

## 8.4.2 读写数据端口---示例

- 要求：对输出端口B的PB<sub>5</sub>位**置位**，假设端口B的地址为301H。
- 程序：

```
mov dx, 301h
```

```
in al, dx          ; 读出B端口原输出内容 即 PB5 内容
```

```
or al, 20h         ; 使PB5=1
```

```
out dx, al        ; 输出新的内容
```

## 8.4.3 读写端口C

- C端口被分成两个4位端口。
- 这两个端口只能以方式0工作，可分别选择输入或输出。

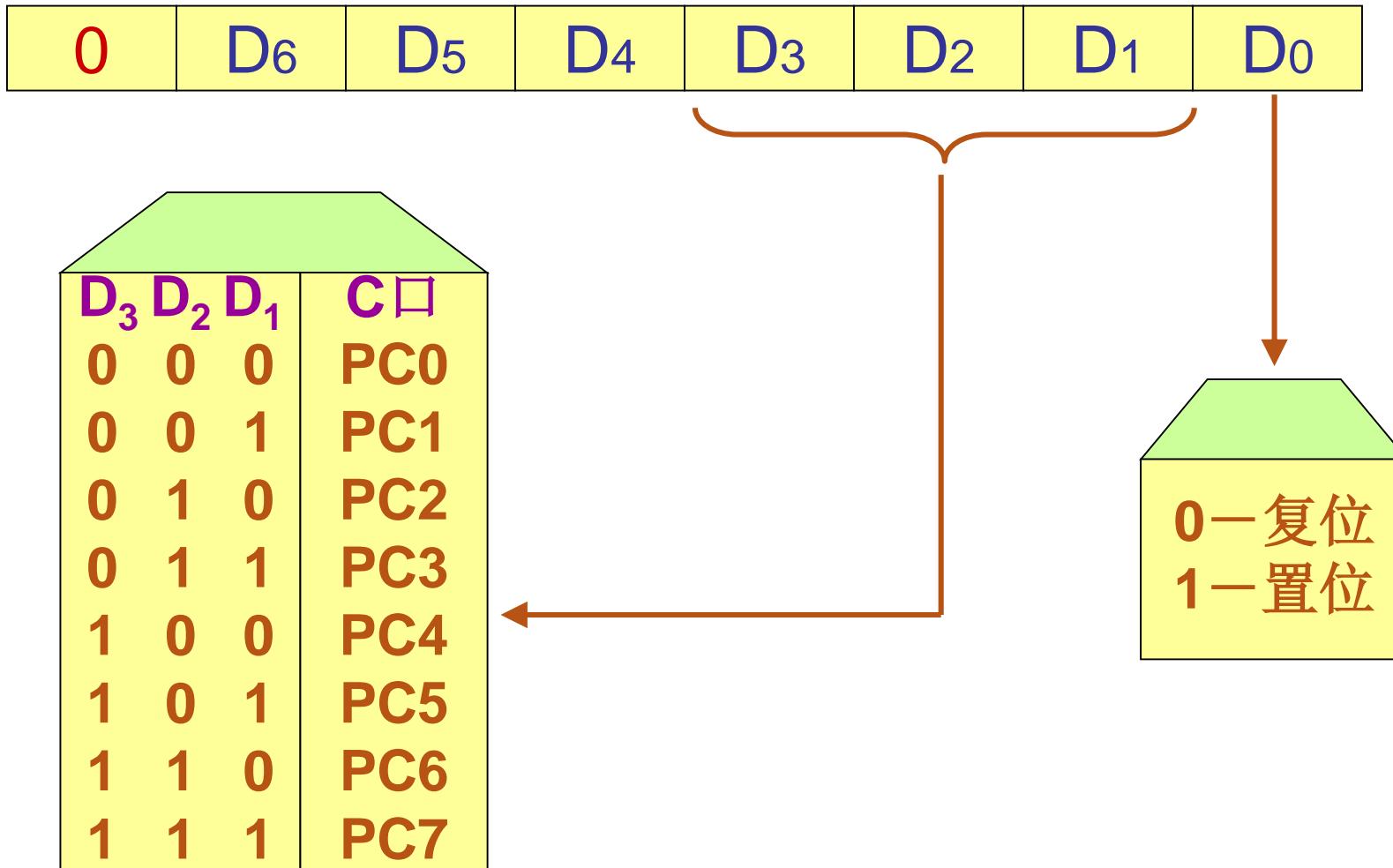
C端口上半部和A端口编为A组，  
C端口下半部和B端口编为B组。



## 8.4.3 读写端口C

- 对端口C的数据输出有两种办法。
  - 通过端口C的I/O地址
    - 向C端口直接写入字节数据
    - 数据被写进C端口的输出锁存器，并从输出引脚输出，但对设置为输入的引脚无效。
- 通过控制端口
  - 向C端口写入按位置位/复位控制字，使C端口的某个引脚输出1或0。

### 8.4.3 C端口的按位置位/复位控制字



- 按位置位/复位控制字写入控制端口( $A_1A_0=11$ )
- 特别便于置位/复位内部中断允许触发器

## 8.4.3 C端口的按位置位/复位控制字—示例

- 假设8255A的控制端口地址为303H，若要把PC<sub>3</sub>引脚置高（置位），则编程

- MOV DX, 303H ; 8255A命令口
- MOV AL, 00000111B; PC<sub>3</sub>=1
- OUT DX, AL PC<sub>3</sub> 置位

## 8.4.3 读写端口C

- 读取的C端口数据有两种情况。

- 未被A和B端口征用的引脚

- 将从定义为输入的端口读到引脚输入信息；
- 将从定义为输出的端口读到输出锁存器中的信息。

- 被A和B端口征用的引脚

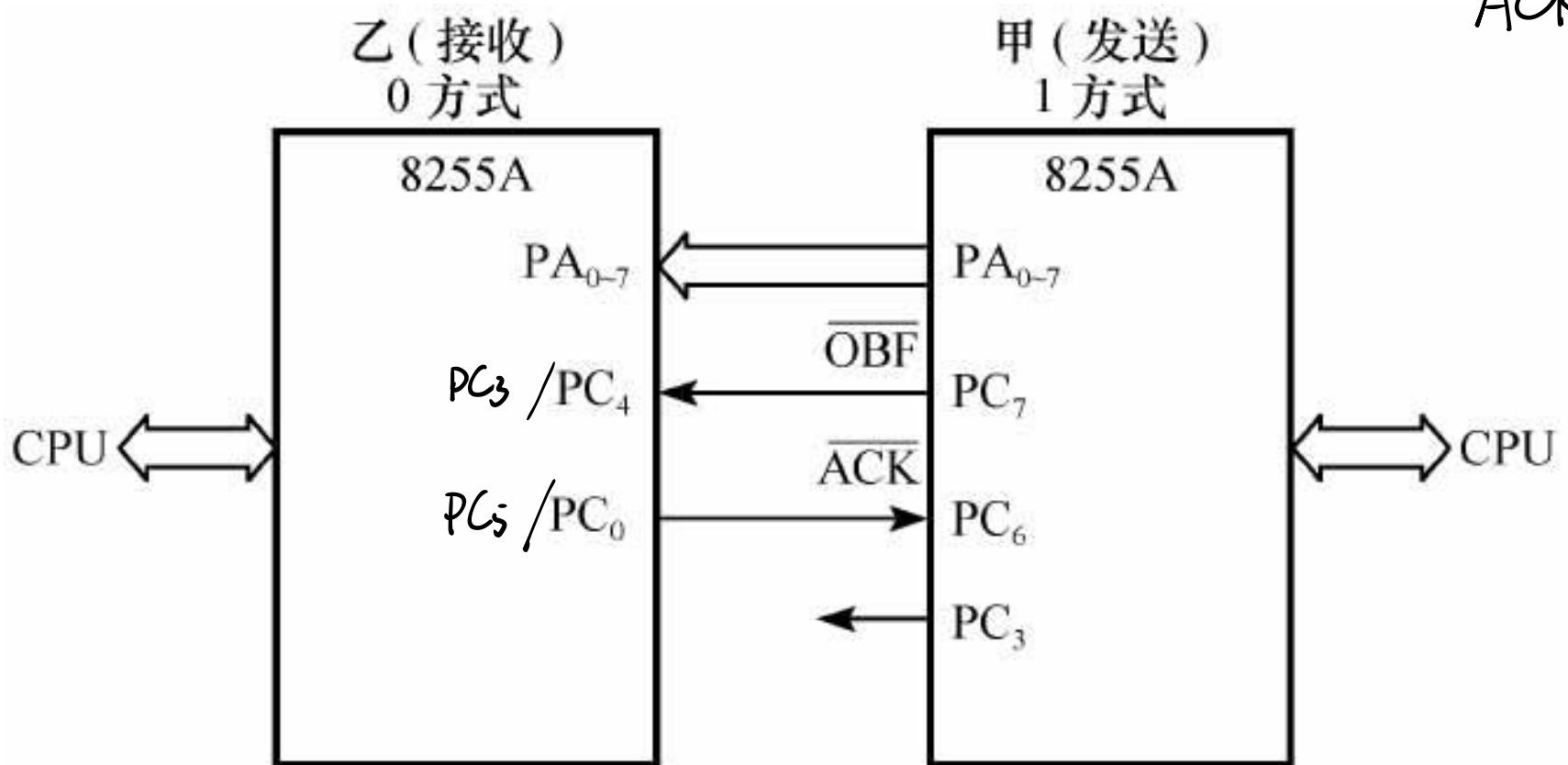
- 将读到反映8255A状态的状态字。

AMAZARASHI

## 8.3.5 8255A的应用

- 作为通用的并行接口电路芯片, 8255A具有广泛的应用:
  - 应用在IBM PC/XT微机上
  - 应用于打印机接口电路
  - 连接简易键盘
  - 驱动LED数码管
  - .....

## 8.3.5. 18255A应用：双机通信



## 8.3.5. 18255A应用：双机通信

- **要求：**在甲乙两台微机之间并行传送1K字节数据。
- 甲机发送，乙机接收。
- 甲机一侧的8255A采用**方式1**工作。
- 乙机一侧的8255A采用**方式0**工作。
- 两机的CPU与接口之间都采用**查询**方式交换数据。

## 8.3.5. 18255A应用：双机通信

- **甲机8255A是方式1，发送**

- 因此, PA口为输出, 发送数据, 而PC7和PC6引脚分别固定做为联络线OBF和ACK。

- **乙机8255A是方式0，接收**

- 故PA口为输入, 接收数据, 引脚PC4和PC0做为联络线。

- **注意：**

- 甲机8255A是1方式, 其联络线是固定的不可替换。
  - 乙机的8255A是0方式, 其联络线是不固定的, 可选择, 比如可选择PC4、PC1或PC7、PC2等组合。

组6

# 甲机发送程序

```
MOV DX, 303H          ; 8255A命令口
MOV AL, 10100000B      ; 初始化工作方式字
OUT DX ,AL
MOV AL ,00001101B      ; 置发送中断允许INTEA=1
OUT DX ,AL              ; PC6=1
MOV AX ,0030H          ; 发送数据内存首址
MOV ES ,AX
MOV BX ,0 } 寻内存中的地址
MOV CX ,0400H          ; 发送字节数
MOV DX ,300H          ; 向A口写第1个数,产生第1个OBF信号
MOV AL ,ES:[BX]         ; 送给对方,以便获取对方的ACK
OUT DX ,AL
INC BX                 ; 指针加1
DEC CX                 ; 字节数减1
```

# 甲机发送程序

L:

MOV DX,302H	; 8255A状态口
IN AL,DX	; 查发送中断请求 INTRA=1?
TEST AL,08H	; PC3=1?
JZ L	; 若无中断请求,则等待
MOV DX,300H	; 8255A PA口地址
MOV AL,ES:[BX]	; 从内存取数
OUT DX,AL	; 通过A口向乙机发送第二个数据
INC BX	; 内存地址加 1
DEC CX	; 字节数减1
JNZ L	; 字节未完,继续
MOV AH,4CH	; 已完,退出
INT 21H	; 返回 DOS

# 乙机接收程序

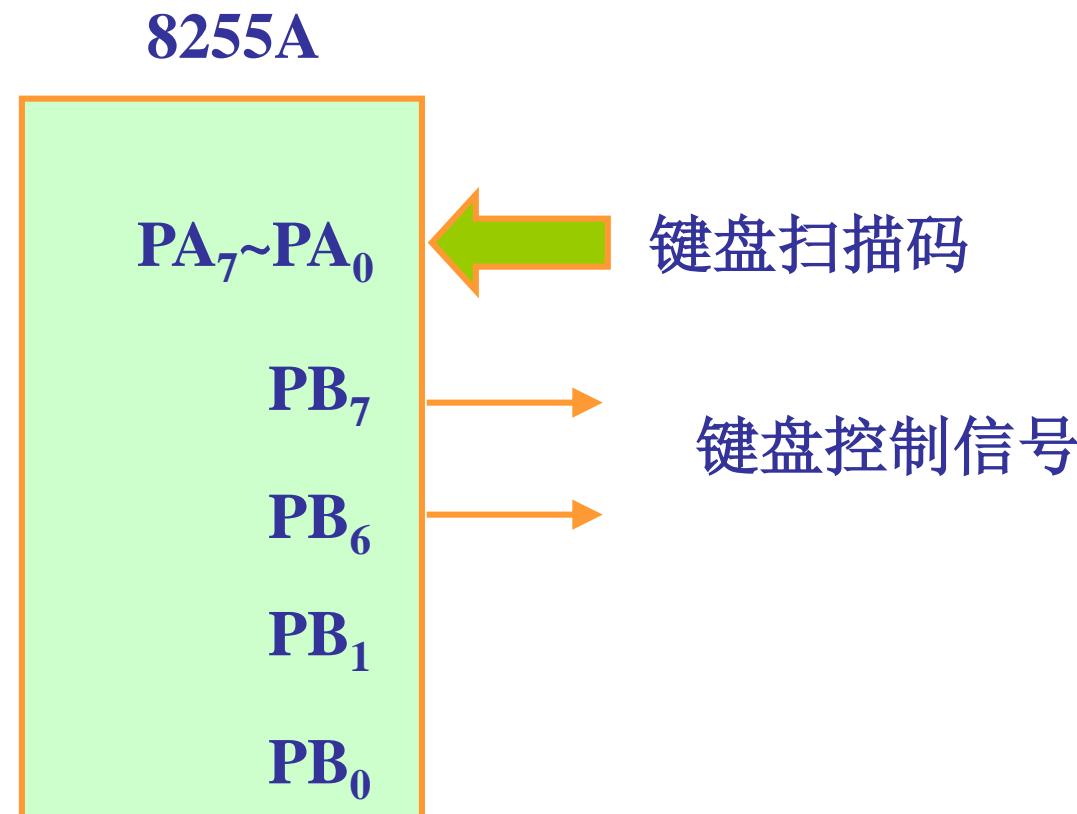
```
MOV DX,303H          ; 8255A命令口
MOV AL,10011000B      ; 初始化工作方式字
OUT DX,AL
MOV AL,00000001B      ; 置ACK=1 (PC0=1)
OUT DX,AL
MOV AX,0040H          ; 接收数据内存首址
MOV ES,AX
MOV BX,00H
MOV CX,400H          ; 接收字节数
L1: MOV DX,302H      ; 8255A PC口
IN AL,DX              ; 查甲机的OBF=0? (PC4)
AND AL,10H            ; 即查甲机是否有数据发来
JNZ L1                ; 若无数据发来，则等待
                      ; 若有数据，则从A口读数
```

# 乙机接收程序

<b>MOV DX,300H</b>	; 8255A PA口地址
<b>IN AL,DX</b>	; 从A口读入数据
<b>MOV ES:[BX],AL</b>	; 存入内存
<b>MOV DX,303H</b>	; 产生ACK信号,发回给甲机
<b>MOV AL,00000000B</b>	; PC0置“0”
<b>OUT DX,AL</b>	
<b>NOP</b>	
<b>NOP</b>	
<b>MOV AL,00000001B</b>	; PC0置“1”
<b>OUT DX,AL</b>	
<b>INC BX</b>	; 内存地址加1
<b>DEC CX</b>	; 字节数减1
<b>JNZ L1</b>	; 字节未完,则继续
<b>MOV AH,4CH</b>	; 已完,退出
<b>INT 21H</b>	; 返回DOS

### 8.5.3.2 8255A应用：键盘接口

8255A在IBM PC/XT上面向键盘的应用：  
2层扫描  
端口A为方式0输入,用来读取键盘扫描码



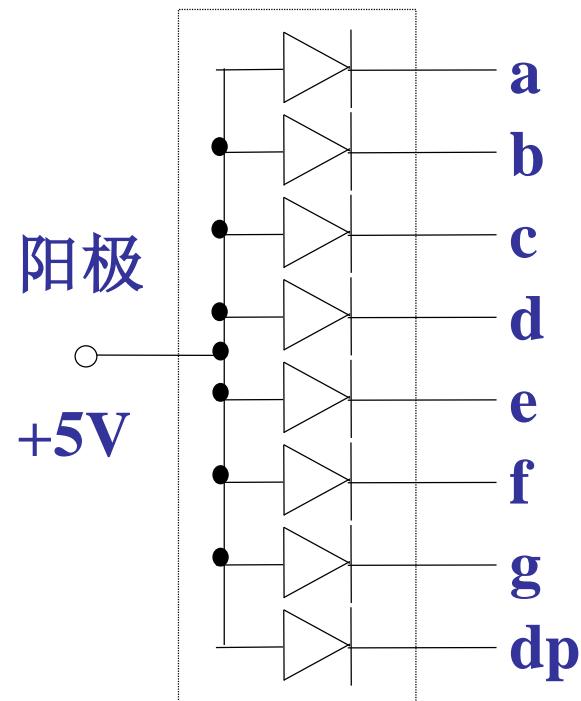
### 8.5.3.3 8255A的应用：LED显示器接口



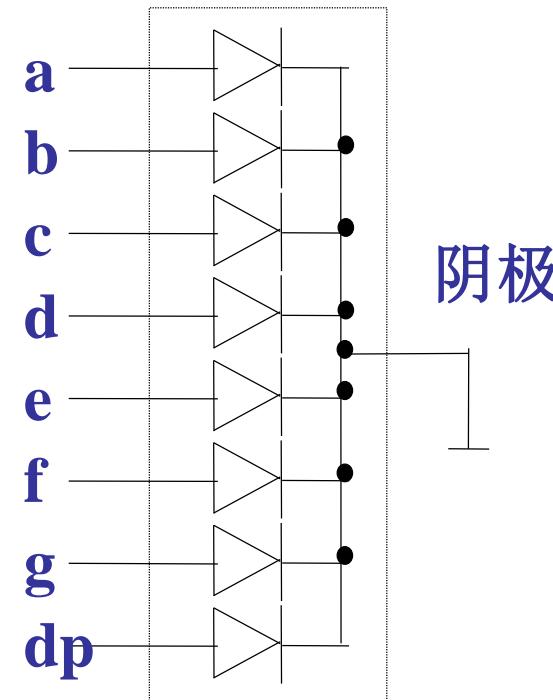
- **发光二极管LED (Light Emitting Diode) 是最简单的显示设备。**
- **由7—8段LED就可以组成的LED显示器。**
- **顺时针分别称为a、 b、 c、 d、 e、 f、 g。**
- **有的产品还附带有一个小数点dp。**
- **通过7或8个发光段的不同组合**
  - 主要显示0~9
  - 可以显示A~F
  - 还可显示个别特殊字符, 如一、P 等
- **LED广泛用于单板微型机、微型机控制系统及数字化仪器中。**

# 8段LED数码管的结构

共阳极



共阴极



- 定义

- 为了达到显示某一字形的目的, 需要从接口中输出不同的数码。

- LED显示器与数据线的连接有如下对应关系:

- a~g分别接数据线D<sub>0</sub>~D<sub>6</sub>, dp接D<sub>7</sub>。

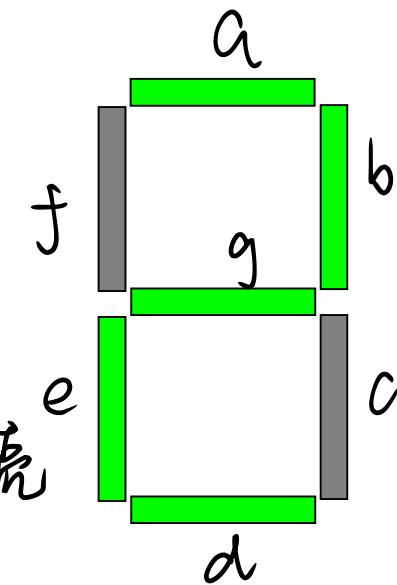
- 例如, 显示“2”

- 采用共阴极接法, 应该使:

- D<sub>7</sub>D<sub>6</sub>D<sub>5</sub>D<sub>4</sub>D<sub>3</sub>D<sub>2</sub>D<sub>1</sub>D<sub>0</sub>=01011011B

段码      ↓ g f e d c b a  
dp.

共阳极: 高电位点亮



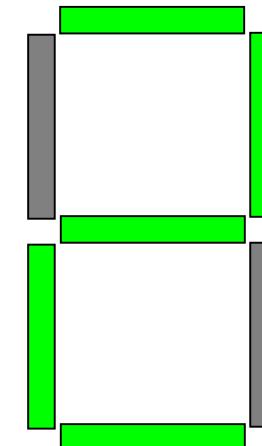
- 对于共阴极和共阳极两种不同的接法, 为了显示同一个字符, 对应的显示段码是不同的。
- 例如, 显示 “2”

- 采用共阴极接法, 应该使:

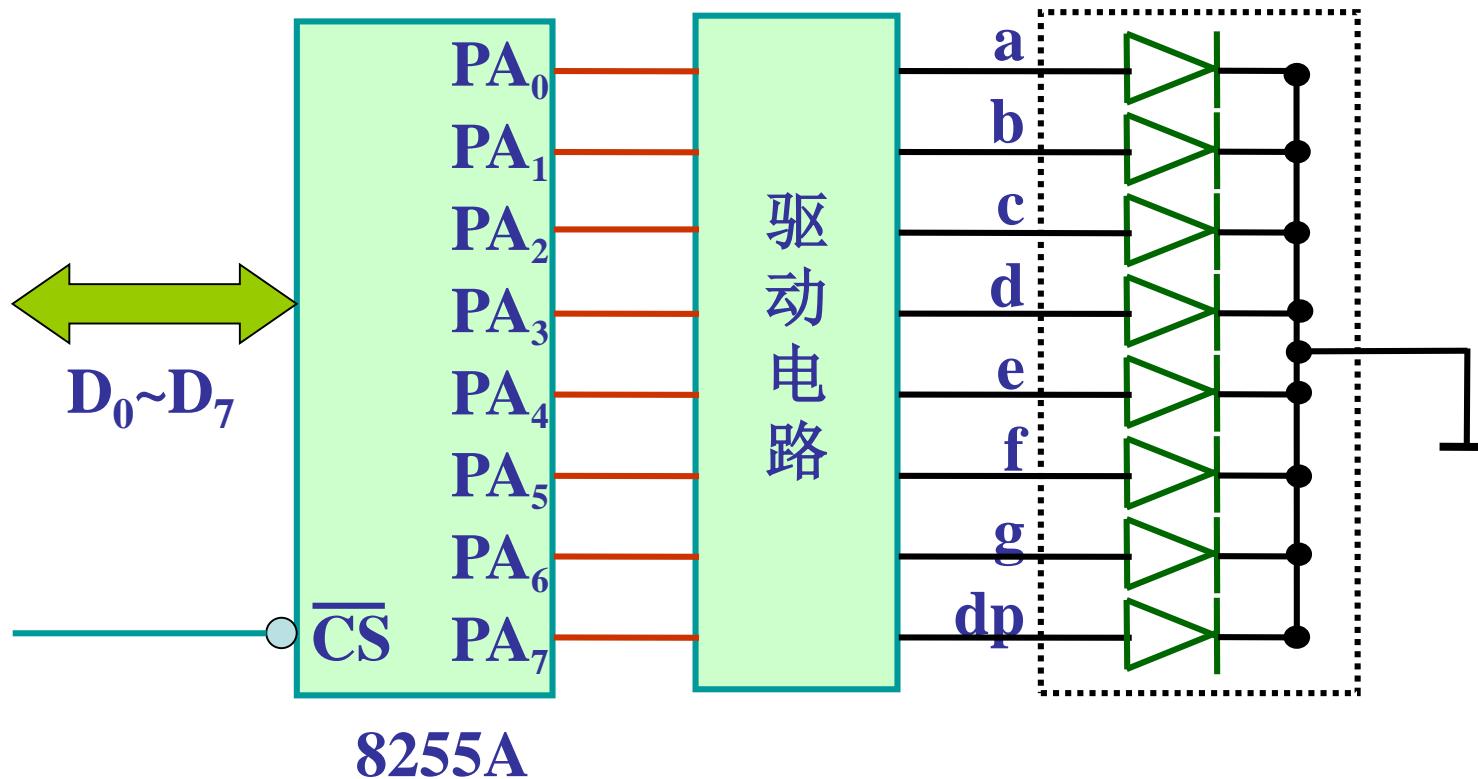
-  $D_7D_6D_5D_4D_3D_2D_1D_0=01011011B$

- 采用共阳极接法, 应该使:

-  $D_7D_6D_5D_4D_3D_2D_1D_0=10100100B$



# 单个LED数码管的显示



**MOV AL, 80H**

**OUT 43H, AL ;设置方式字**

**MOV AL, 6DH ;“5”的段码**

**OUT 40H, AL**

6D即为共阳的段码

# 举例：显示0~F

TABLE DB 3FH,06H,5BH,4FH  
DB 99H,6DH,7DH,07H  
DB 7FH,6FH,77H,7CH  
DB 39H,5EH,79H,71H

0~F. 段码数组表

.....

MOV AL,80H

OUT 43H,AL

MOVCX,16 ; 显示字符循环次数

LEA DI, TABLE ; DI指向段码表首址 TABLE 的偏移地址

DISP:

MOV AL, [DI] ; 将显示段码送从AL

MOV DX, 40H ; 将端口地址送DX

OUT DX,AL ; 输出段码至输出端口

INC DI ; 指向下一个要显示字符的段码

CALL DELAY ; 调显示延时子程序

LOOP DISP ; 0~F未显示完,转 DISP

.....

# 多位LED显示器接口

- 2种基本方法

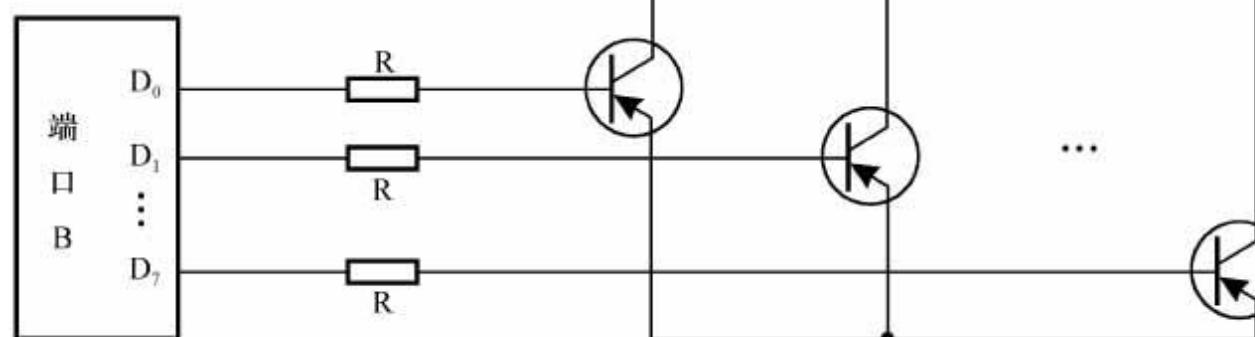
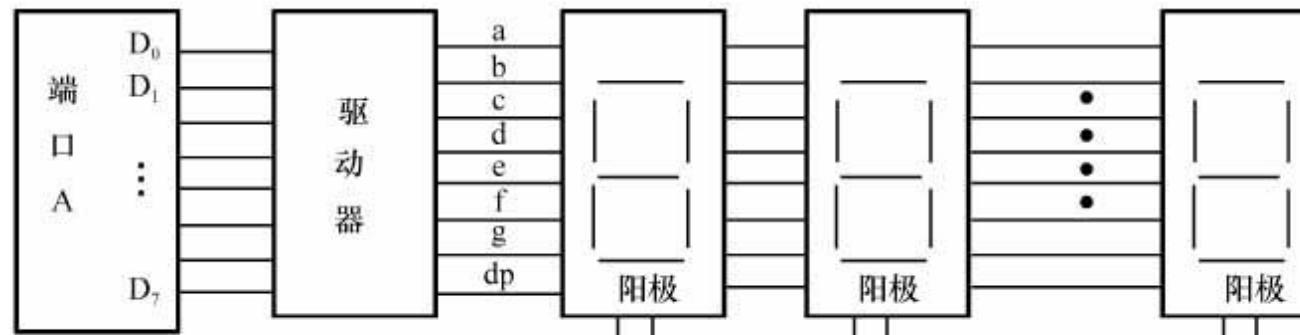


段码控制:  $D_i = 0$ , 相应段发光 此端口由8个数码管共用

- 采用多个独立的并行输出端口来驱动

- 动态扫描、分时循环显示

段码控制端口 (端口A)



显示  
↑  
10111111

位控制码

位控制端口 (端口B)

位7	位6	位5	位4	位3	位2	位1	位0
----	----	----	----	----	----	----	----

位控制:  $D_i = 0$ , 相应位发光

- 利用上述接口电路, 只要设计相应的接口驱动软件加以控制, 就可实现所需要的动态扫描、分时显示。
- 准备工作

- 建立**显示数据缓冲区**, 保存要显示的字符;
- 建立**段码表**, 以供查询要显示字符所对应的段码。

↳ TABLE

```
    } LEA  DI, TABLE  
    { MOV  AL, [DI]  
      MOV  DX, xxH  
      OUT  DX, AL
```

# 显示缓冲区和段码表

1998.10.

ひくじつ XLA7

DISMEM

.....
01H
09H
09H
08H
10H
01H
00H
10H

显示缓冲区

SEGPT

.....
C0H
F9H
.....
8EH
7FH

段码表

## DATA SEGMENT

SEGPT

DB C0H, F9H, A4H, B0H  
DB 99H, 92H, 82H, FBH  
DB 80H, 90H, 88H, 83H  
DB C6H, A1H, 86H, 8EH  
DB 7FH

DISMEM

DB 01H, 09H, 09H, 08H  
DB 10H, 01H, 00H, 10H

DATA ENDS

.....

↓  
小数点

BX, 数组首地址

序号

# 接口程序

**MOV AL,80H**

**OUT 43H,AL**

**LEA DI,DISMEM**

; 指向显示缓冲区首址

**MOV CL,0FEH**

; 指向左端LED显示器

**MOV AL,0FFH**

; 关显示

最左侧的位控码

**OUT PortB,AL**

**DISP:**

**MOV AL, [DI]**

; 取要显示的字符

**LEA BX,SEGPT**

; 段码表首址送BX

**XLAT** ; 将要显示的字符转换为对应的段码存AL

**OUT PortA,AL**

; 将段码送至端口 A

**MOV AL, CL**

; 将位码送端口 B

**OUT PortB,AL**

<b>PUSH</b>	<b>CX</b>	; 保存位码至堆栈
<b>MOV</b>	<b>CX,30H</b>	; 延时一定时间
<b>DELAY:</b>		
<b>LOOP</b>	<b>DELAY</b>	
<b>POP</b>	<b>CX</b>	; 从堆栈取出位码
<b>ROL</b>	<b>CL,1</b>	指向第二位的位码
<b>CMP</b>	<b>CL,0FEH</b>	; 显示至最右端了吗? 若=0FEH 则已显
<b>JZ</b>	<b>DESEND</b>	; 是,转出口
<b>INC</b>	<b>DI</b>	; 否,指向下一位要显示的字符
<b>JMP</b>	<b>DISP</b>	
<b>DISEND:</b>		
.....		

- **教学要求**

- 掌握8255A的结构特点和引脚功能；
- 掌握8255A的工作方式（0、1）、编程及方式0的应用。
- 熟悉键盘工作原理
- 掌握LED显示器工作原理及应用

- **作业**

- P214: 1, 4, 5, 6, 7, 8