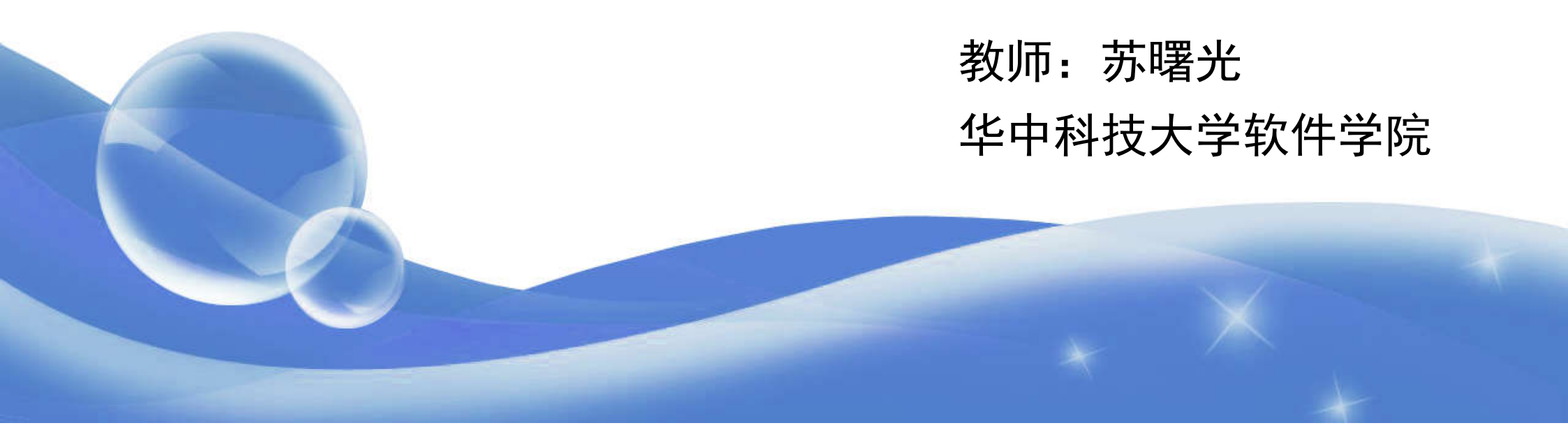




《微机原理与接口》

## 第8章 并行接口和8255A

教师：苏曙光  
华中科技大学软件学院



## ● 教学内容

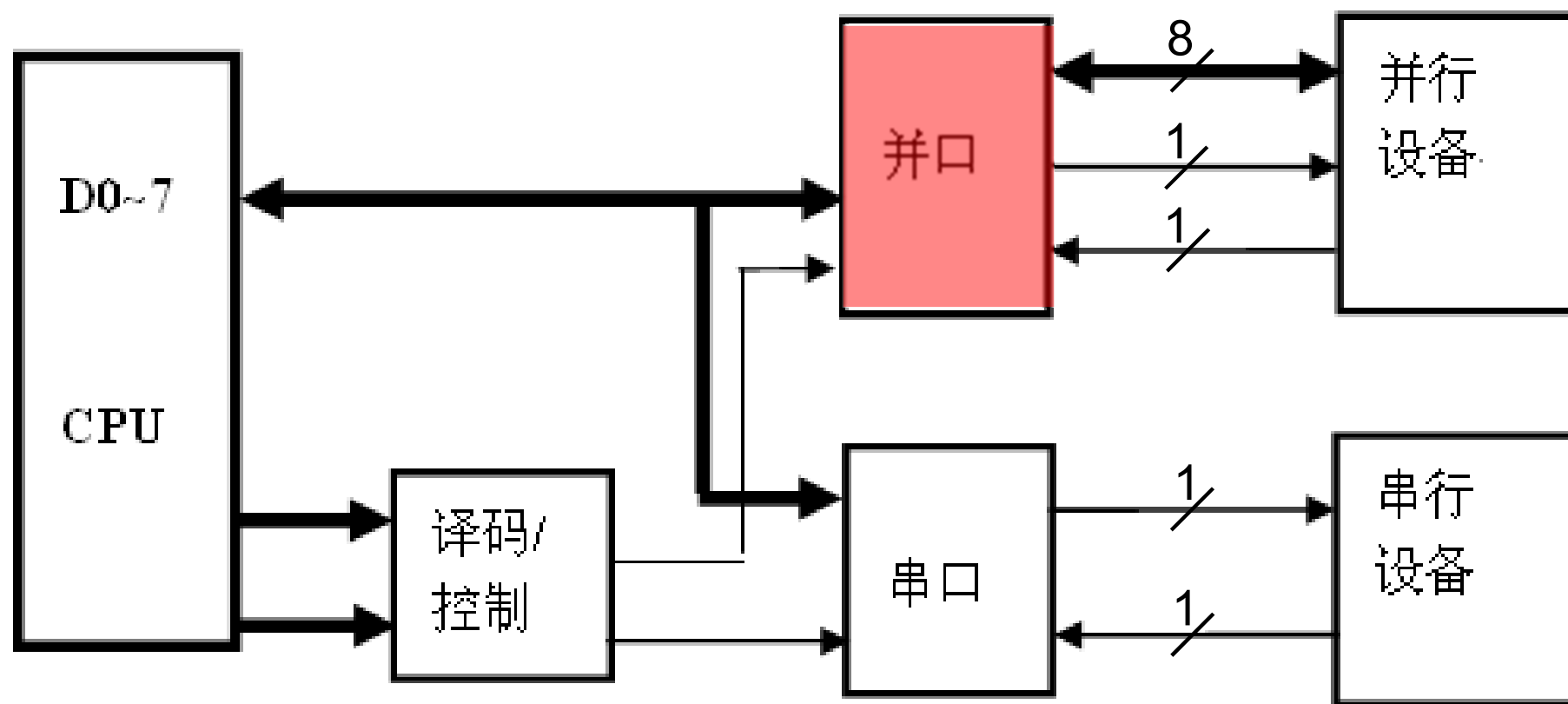
- 第1节 并行接口的概念
- 第2节 8255A的外部引脚和内部结构
- 第3节 8255A的控制字
- 第4节 8255A的工作方式



## 第1节 并行接口的概念

# 并行接口特点

- 并口的位置



# 并行接口的特点

- 1) 数据线有多根，所有数据线一起传输。
  - 典型数据线的根数：8根（字节）或16根（字），也可自定
  - 对比串行接口：通过1根线传送信息
  - 8位或16位一起传输（即使只用到其中1位）。
  - 并行：特指接口与I/O设备之间。

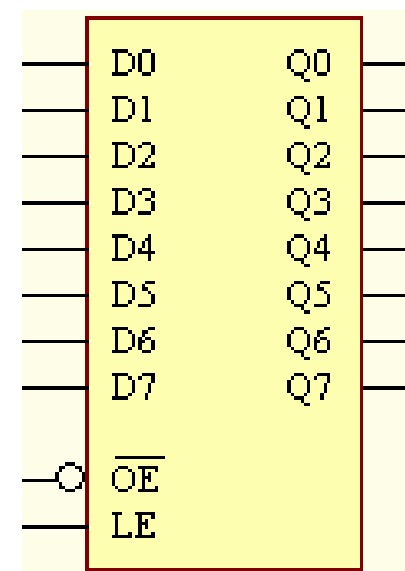
- 3) 数据格式不受限制
- 4) 分类: 硬连接并口和可编程并口。

■ 硬连接并口:

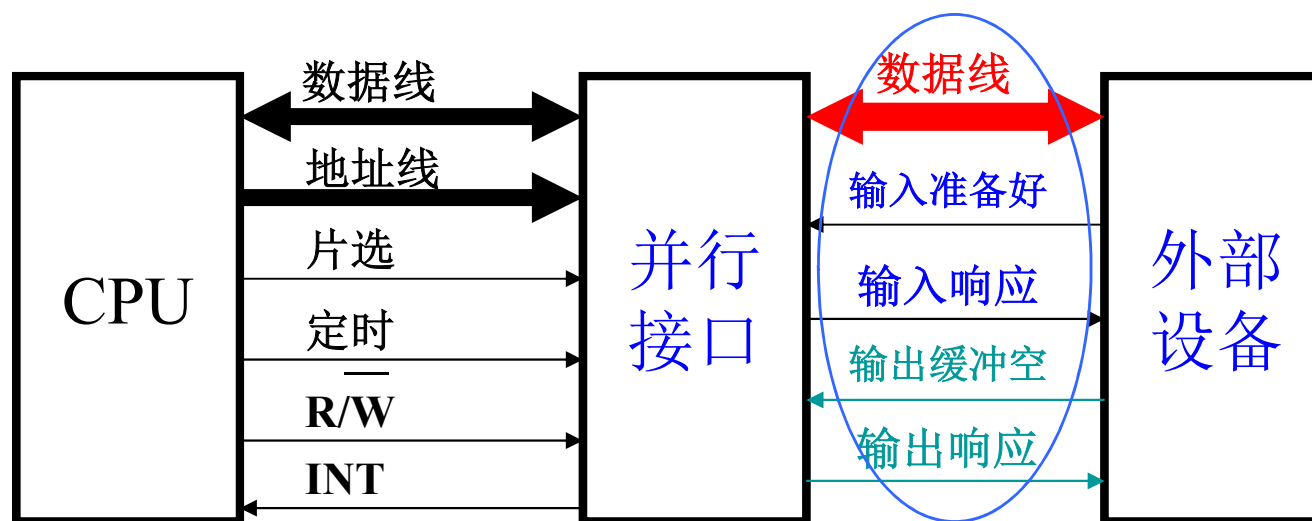
◆ 工作方式及引脚功能固定(例373,244)

■ 可编程并口

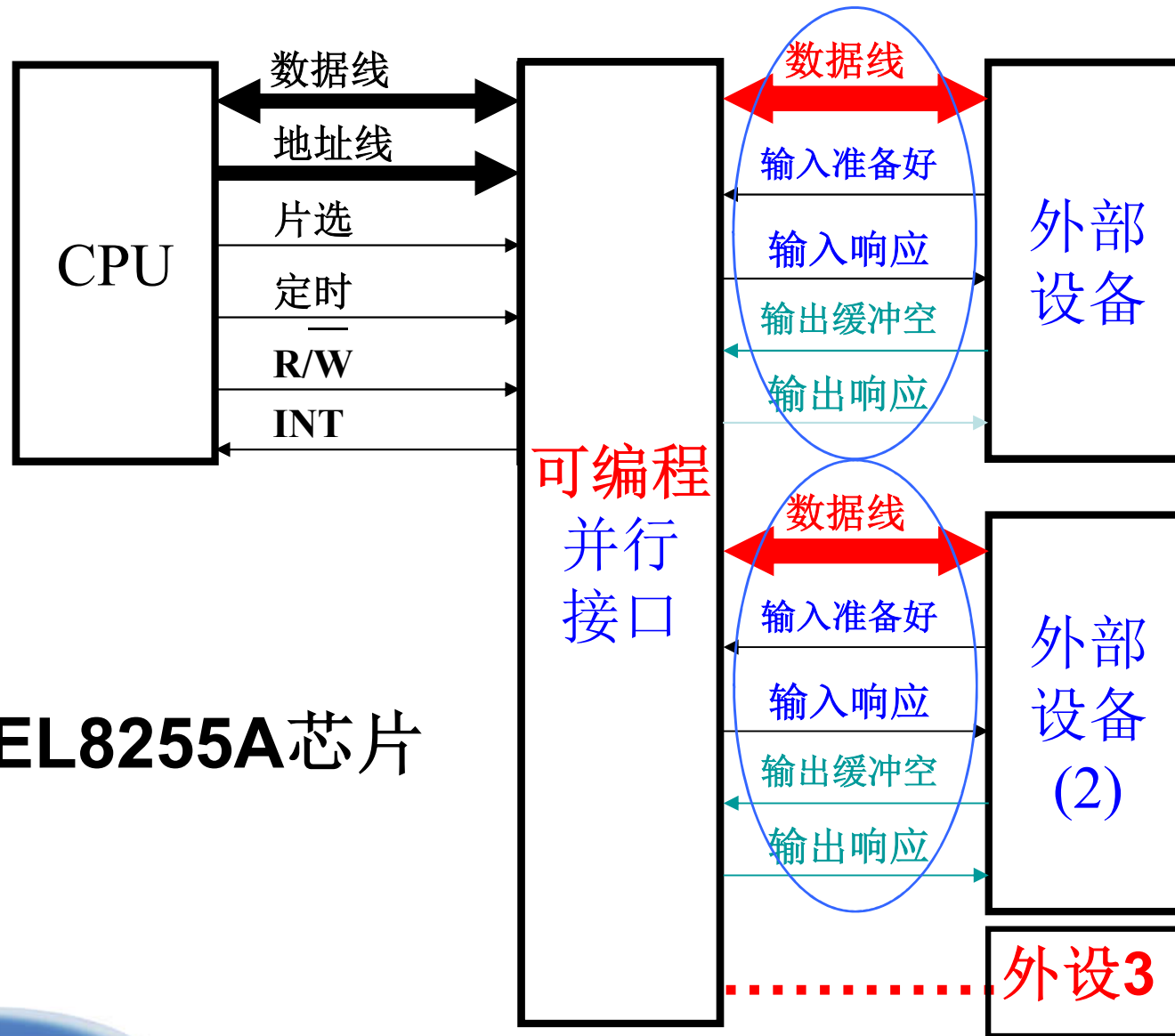
◆ 工作方式及引脚功能可编程设定 (例8255A)



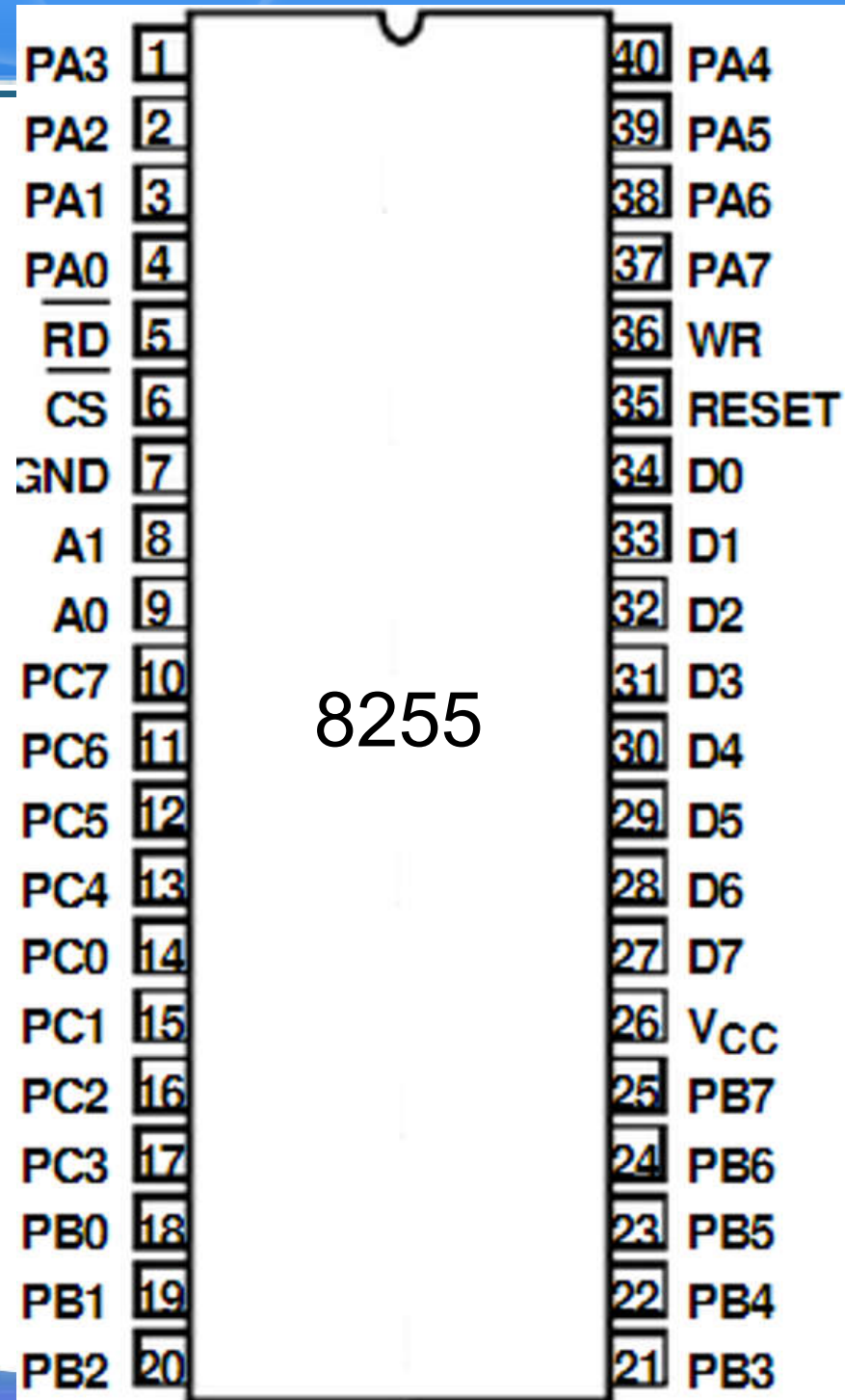
74LS373



# 可编程并口作用：扩充数据总线的数量和适应性



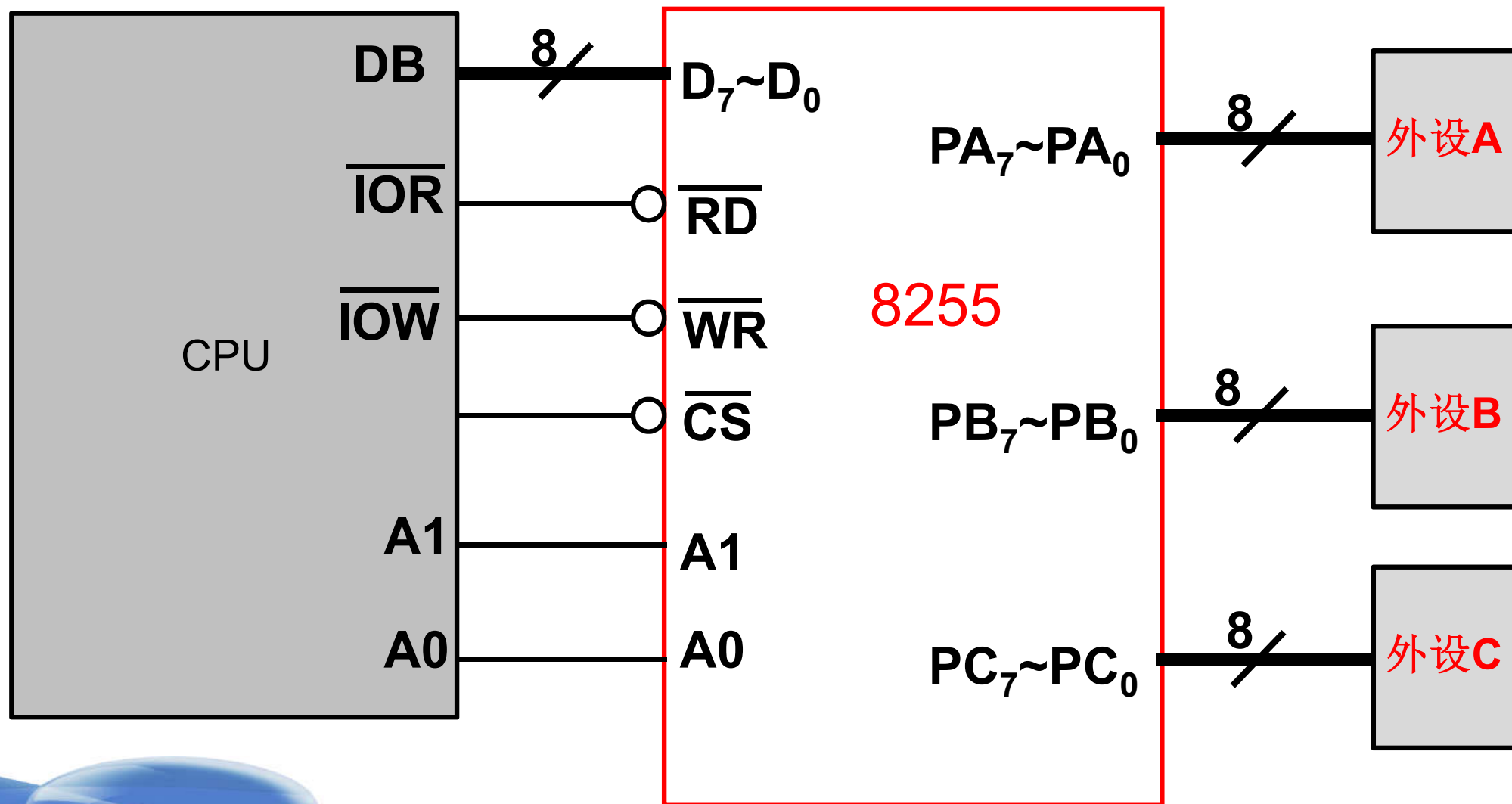
## ● INTEL8255A芯片





# 8255A与PC微机的典型连接方式

## ● (DB被扩充: 1→3)





## 第2节 8255A的外部引脚和内部结构

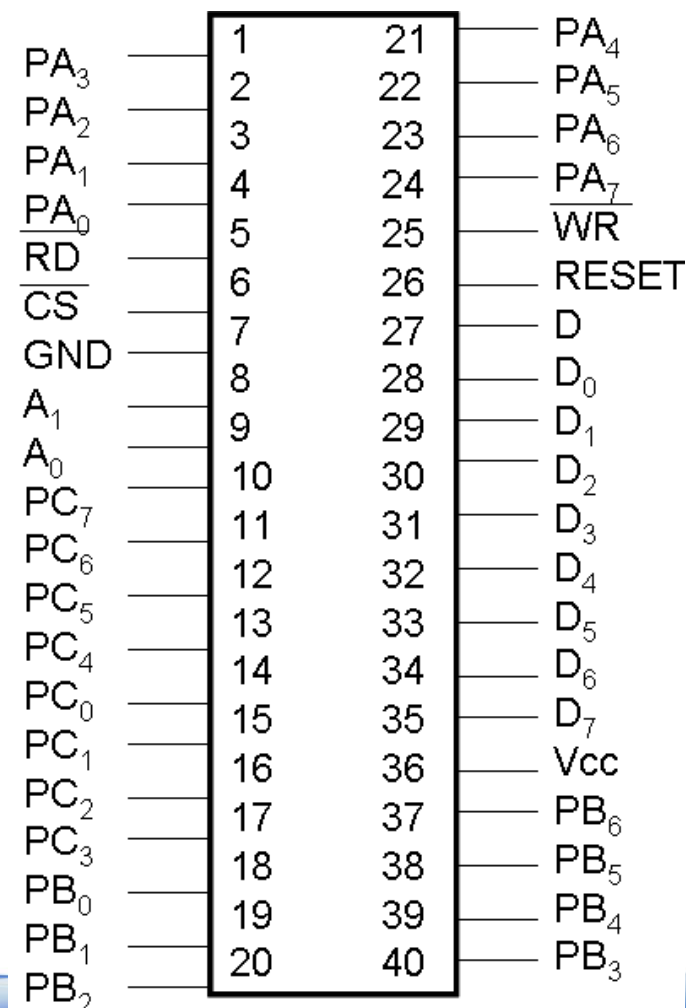
# 8255A的基本特点

## ● 功能

- 扩充数据总线的数量（1→3）： PA,PB,PC
- 适应输入/输出，查询，中断等多种应用场合
- 具有3个输入/输出并口（PA,PB,PC）

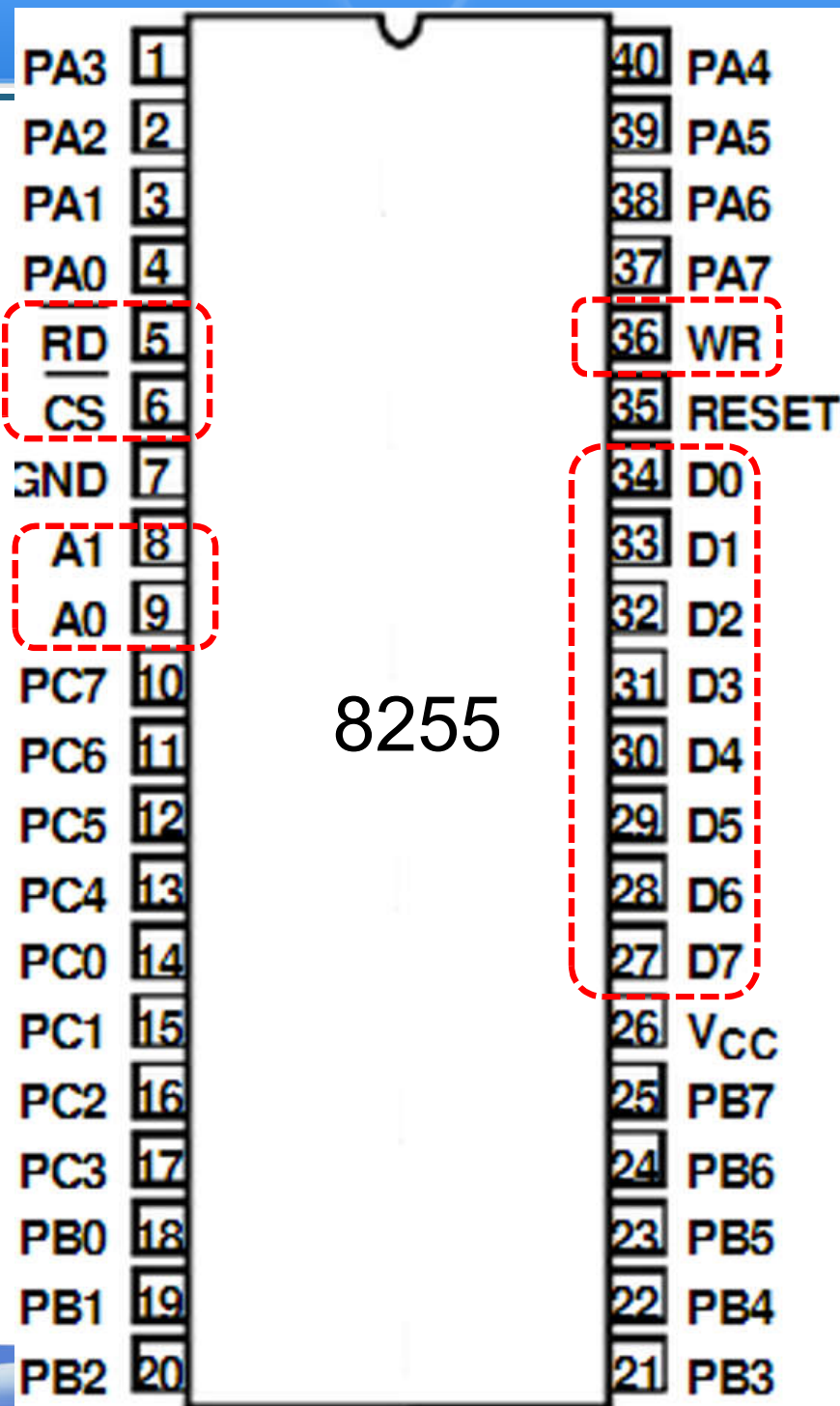
## ● 特点

- 可选择某特定并口连接CPU与外设；
- 每个并口可以指定输入/输出的方向；
- 与CPU之间的数据传送方式可选择；
- 有4个8位端口：
  - ◆ PA端口， PB端口， PC端口， 控制端口



# 8255的外部引脚 (面向CPU侧)

- 数据总线: D0~D7
  - 命令、数据、状态等数据。
- 地址总线: A1, A0 ( CS )
  - 4个端口: PA, PB, PC, 控制口
- 控制总线: RD, WR
  - 读, 写信号
- 其他总线: RESET, GND, VCC
  - RESET , 高电平有效。清除控制寄存器并将三个端口置输入方式并且屏蔽中断请求。



# 8255的外部引脚 (面向外设侧)

- 三组并口线

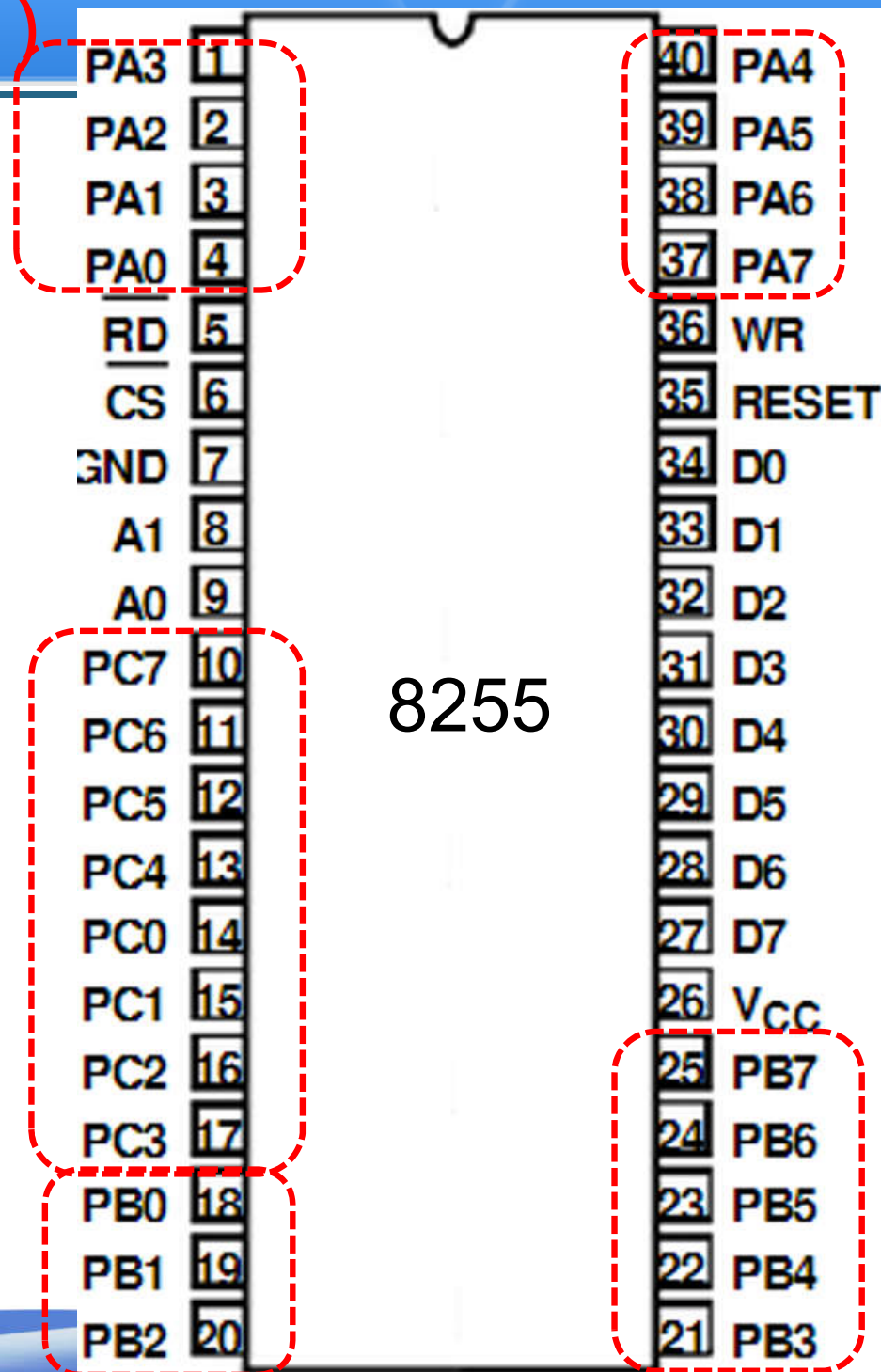
- PA0~PA7: 对应PA端口

- PB0~PB7: 对应PB端口

- PC0~PC7: 对应PC端口

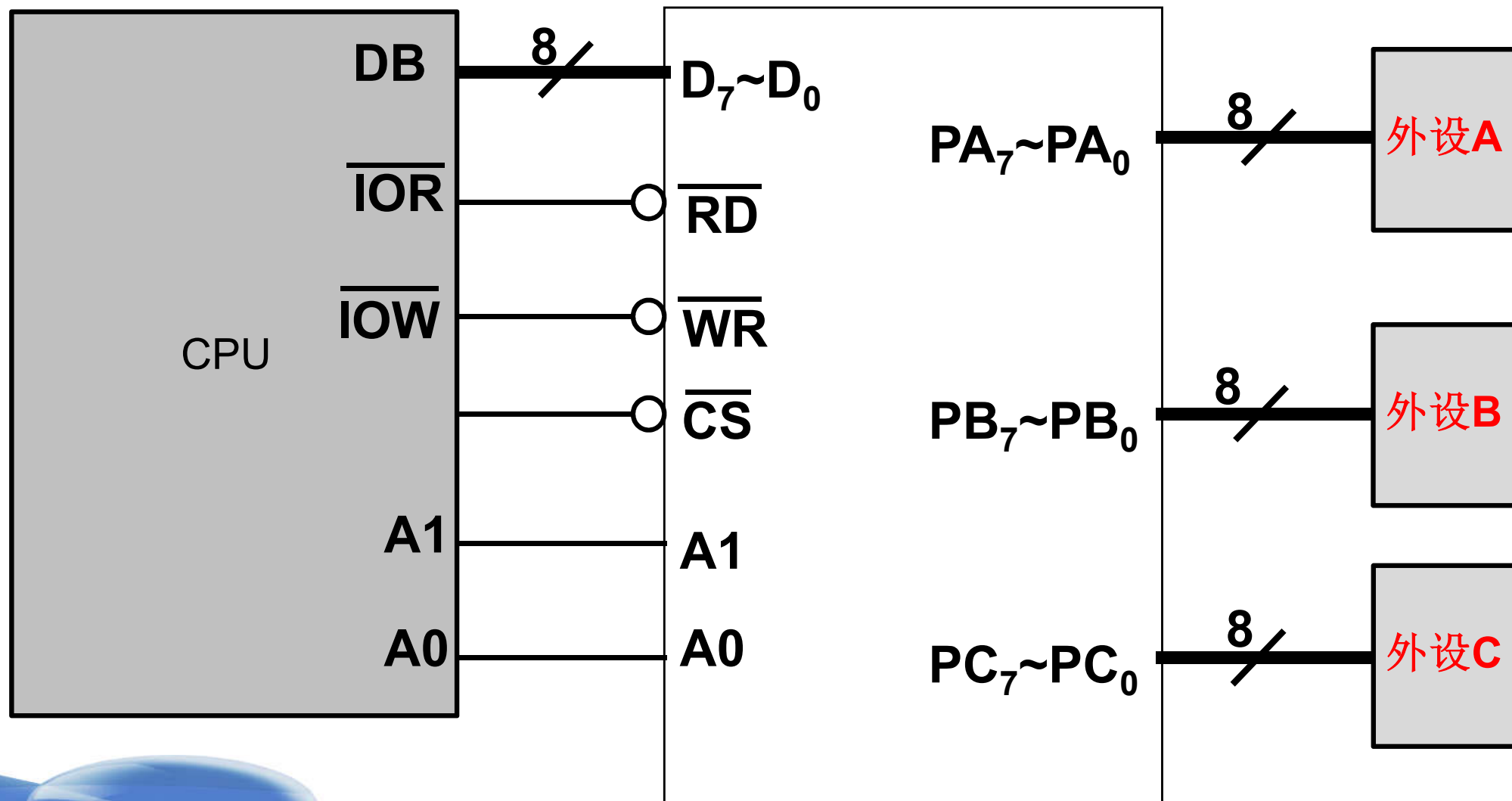
- 说明

- PA, PB与PC的作用和方式可以编程: 可以做8位并行数据线, PC还可以按位控制作控制线来用。



# 8255A与PC微机的典型连接方式

## ● (DB被扩充: 1→3)



## ● 8255A的操作与端口

■ 4个连续端口：PA,PB,PC,控制口

$\overline{\text{CS}}$	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\text{A}_1$	$\text{A}_0$	操作
0	1	0	0	0	写端口A
0	1	0	0	1	写端口B
0	1	0	1	0	写端口C
0	1	0	1	1	写控制口
0	0	1	0	0	读端口A
0	0	1	0	1	读端口B
0	0	1	1	0	读端口C
0	0	1	1	1	无操作

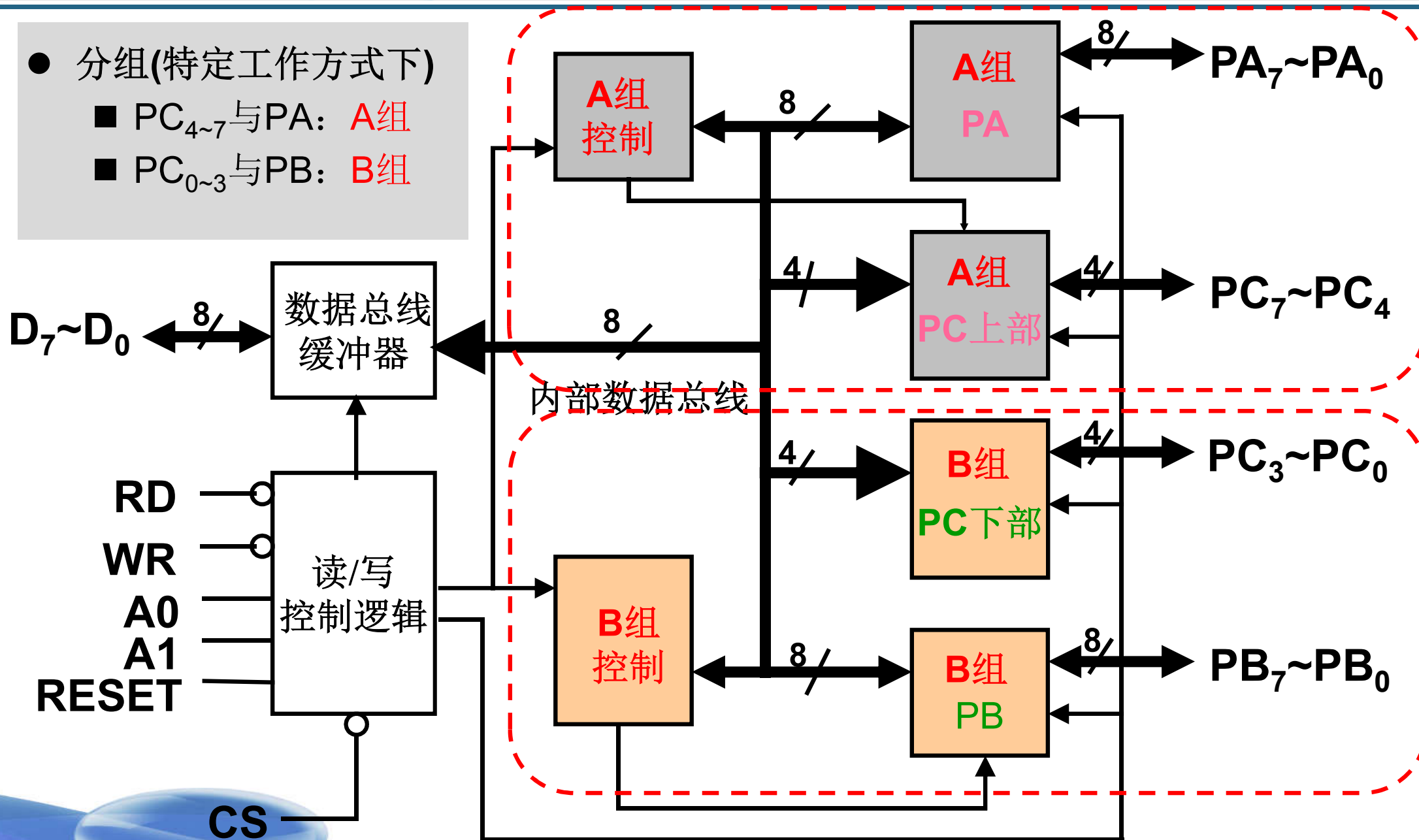


# 8255A的内部结构

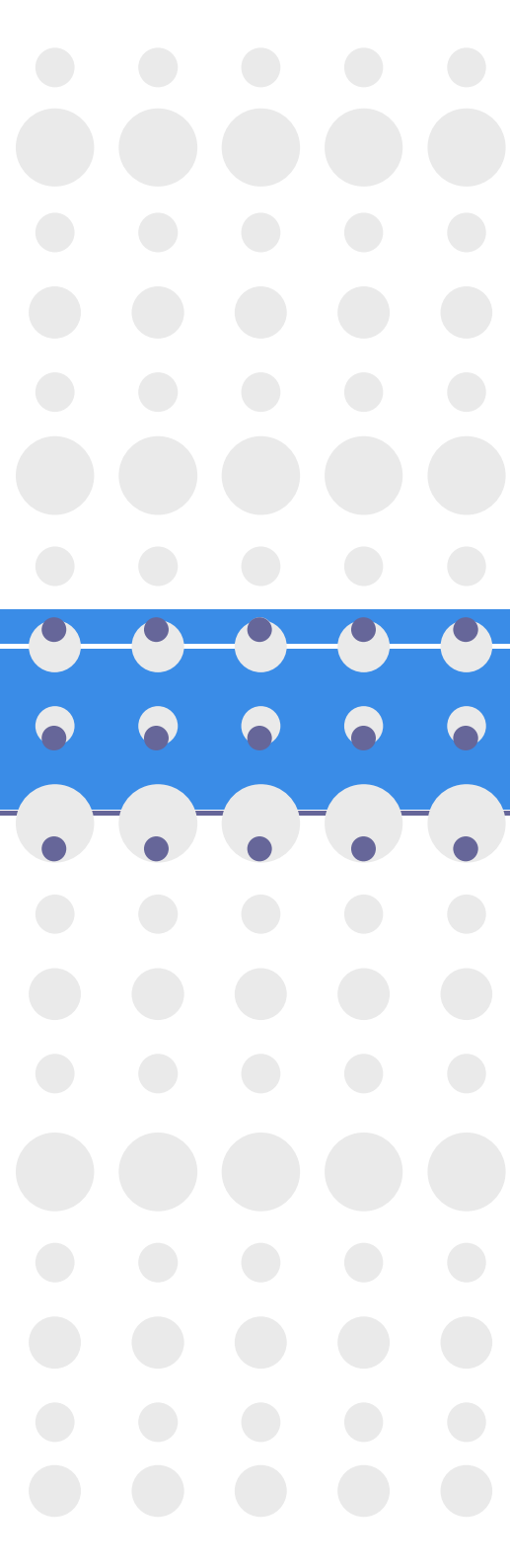
- 分组(特定工作方式下)

- $PC_{4\sim7}$ 与PA: **A组**

- $PC_{0\sim3}$ 与PB: **B组**







## 第3节 8255A的控制字

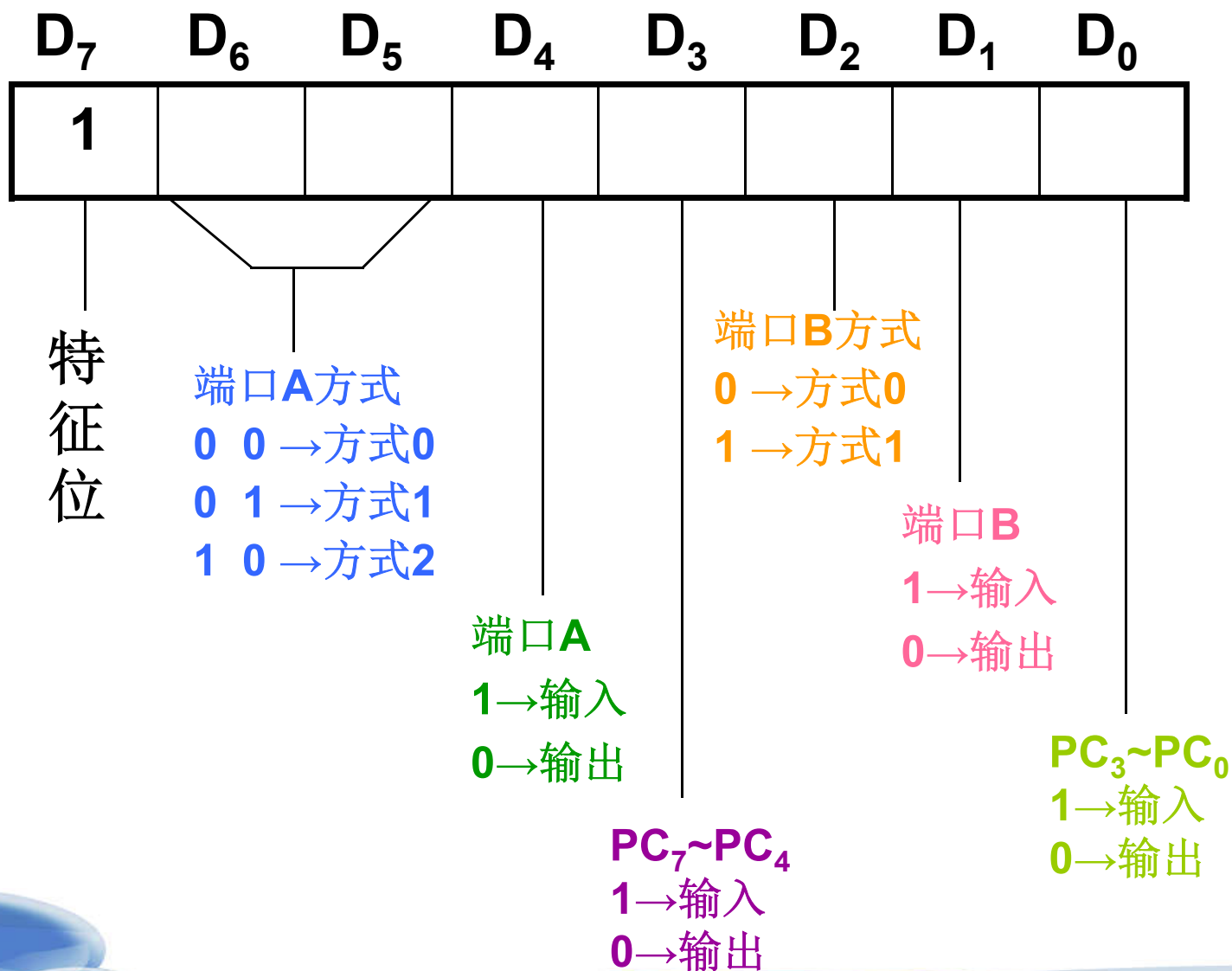
## ● 8255A的常用控制字

- 方式选择控制字：设置工作方式
- 按位置位/复位字：按位设定PC某位的值。
- 注意：控制字必须写在控制端口

## ● 8255的3种工作方式

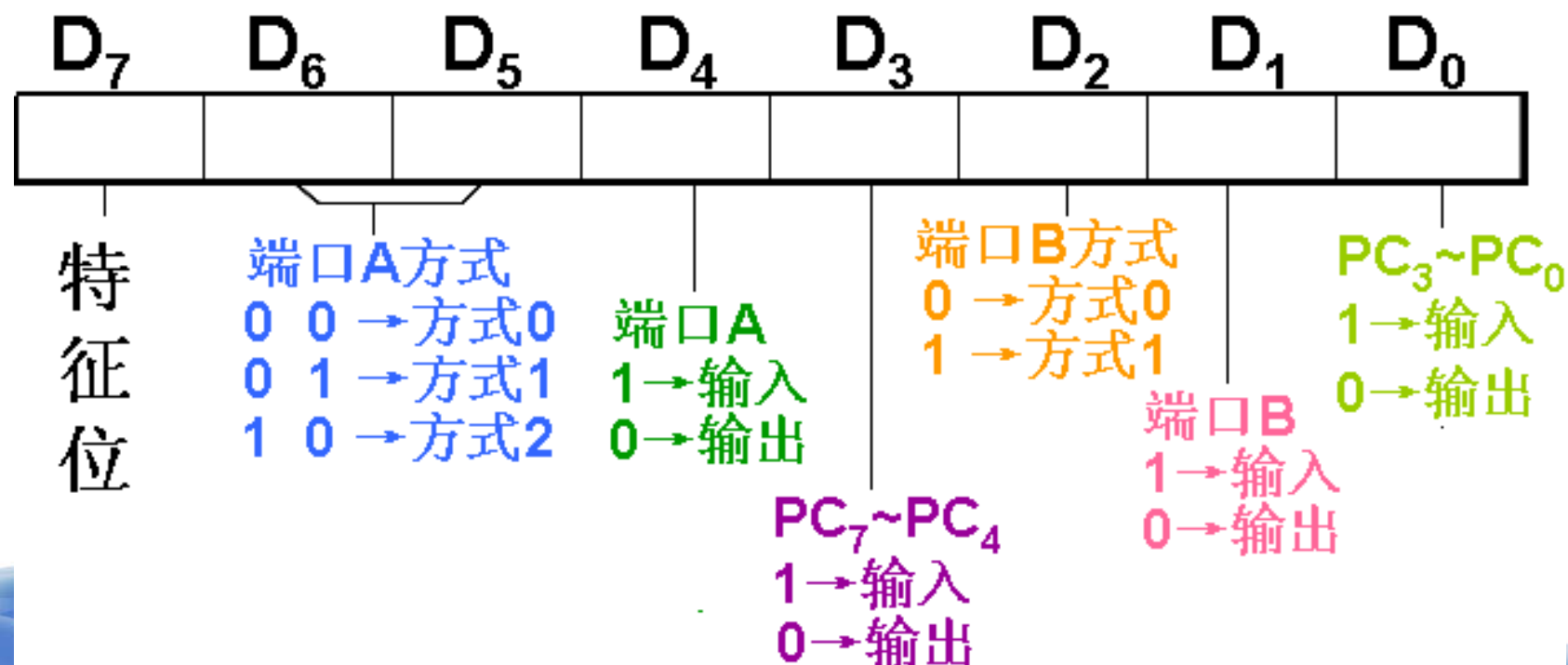
- 方式0(Mode 0) ——基本输入输出
- 方式1(Mode 1) ——选通输入输出
- 方式2(Mode 2) ——双向传送
- 三个并口（端口）可以独自设置不同的工作方式。
  - ◆ PA: 方式0、方式1、方式2;
  - ◆ PB: 方式0、方式1;
  - ◆ PC: 方式0
  - ◆ 传输方向可以指定
- 工作方式由方式选择控制字设定。

## ● 方式选择控制字



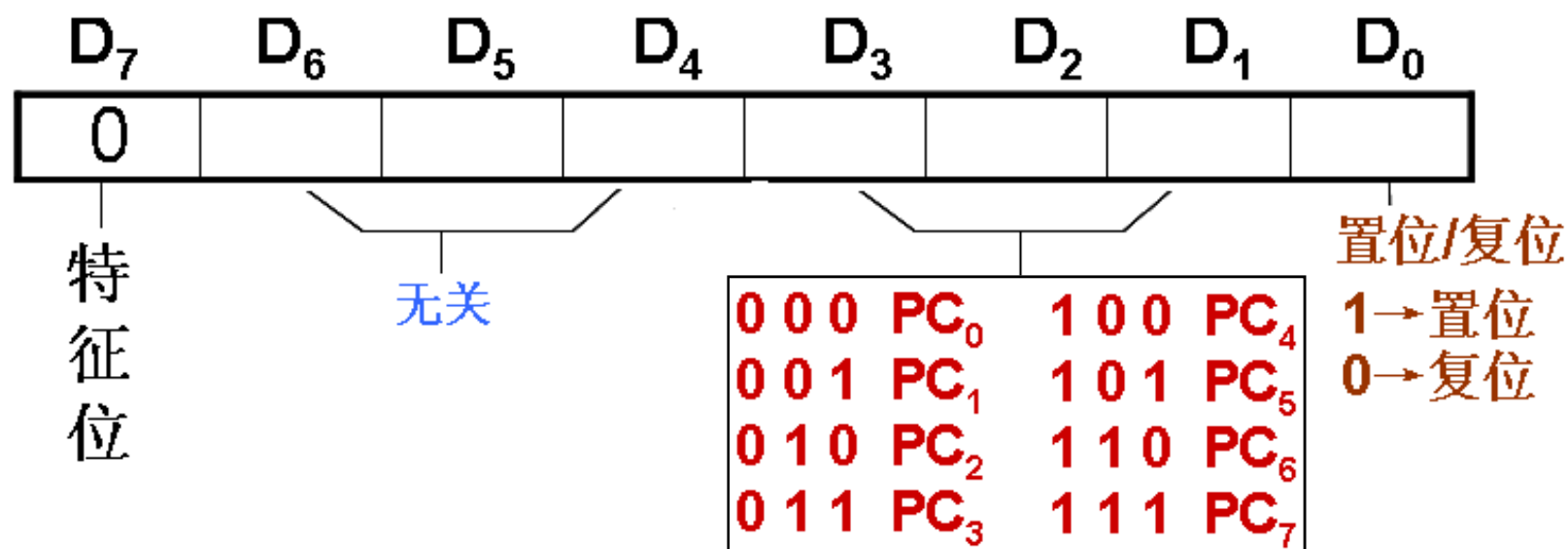
- 例子：初始化8255：A口方式1，输入；C口上部输出；B口方式0，输出；C口下部输入。8255地址：300H~303H

```
MOV DX, 303H    ; 8255A命令口地址
MOV AL, B1H     ; 初始化命令
OUT DX, AL      ; 命令口
```



## ● 按位置位/复位控制字

- 作用：按位设定PC某一位电平的高或低。
- 格式：8位：最高位固定为0。



■ 例：PC<sub>3</sub>置位：00000111B

■ 例：PC<sub>3</sub>复位：00000110B

- 例：把PC<sub>2</sub>置成高电平。命令字：0000**0101**B或05H

MOV DX, 303H ;8255A命令口地址

MOV AL, **05H** ;使PC<sub>2</sub>=1的命令字

OUT DX, AL ;送到命令口

；若置PC<sub>2</sub>低电平：0000**0100**B或**04H**

**区别**如下置PC<sub>2</sub>高电平？

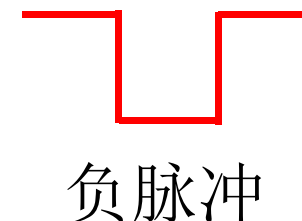
MOV DX, **302H** ; PC口的地址

MOV AL, 00000**100** ; **PC<sub>2</sub>置为高电平**

OUT DX, AL

- 例子：利用**PC<sub>7</sub>**产生**负脉冲**作打印机接口数据选通信号。  
;假定PC7初始缺省为高电平

```
mov dx, 303H      ;8255A命令口地址
mov al, 00001110B ;使PC7=0的命令字
out dx, al        ;送到命令口
nop               ;维持低电平
nop               ;维持低电平
nop               ;维持低电平
nop               ;维持低电平
mov al, 00001111B ;置PC7=1
out dx, al
```





## ● 按位置位/复位字与方式控制字的使用

- 1、D7位区分两个命令
- 2、按位置位/复位字在方式选择控制字之后写入控制端口。
- 3、按位置位/复位写入命令口，不能写入PC口
- 4、按位置位/复位不影响工作方式



## 第4节 8255的工作方式

## ● 0方式（基本I/O方式）

- 指无条件传送或查询传送。

- ◆无条件传送：不用联络信号（状态信息）。

- ◆查询传送：联络信号可由任意空闲引脚充当。

- 没有特定的引脚充当联络信号

- 联络信号的时序没有特别约定

- PA, PB和两个PC4位都可以设定为0方式

- 可以由用户指定输入或者输出方向。

- 单向输入或输出

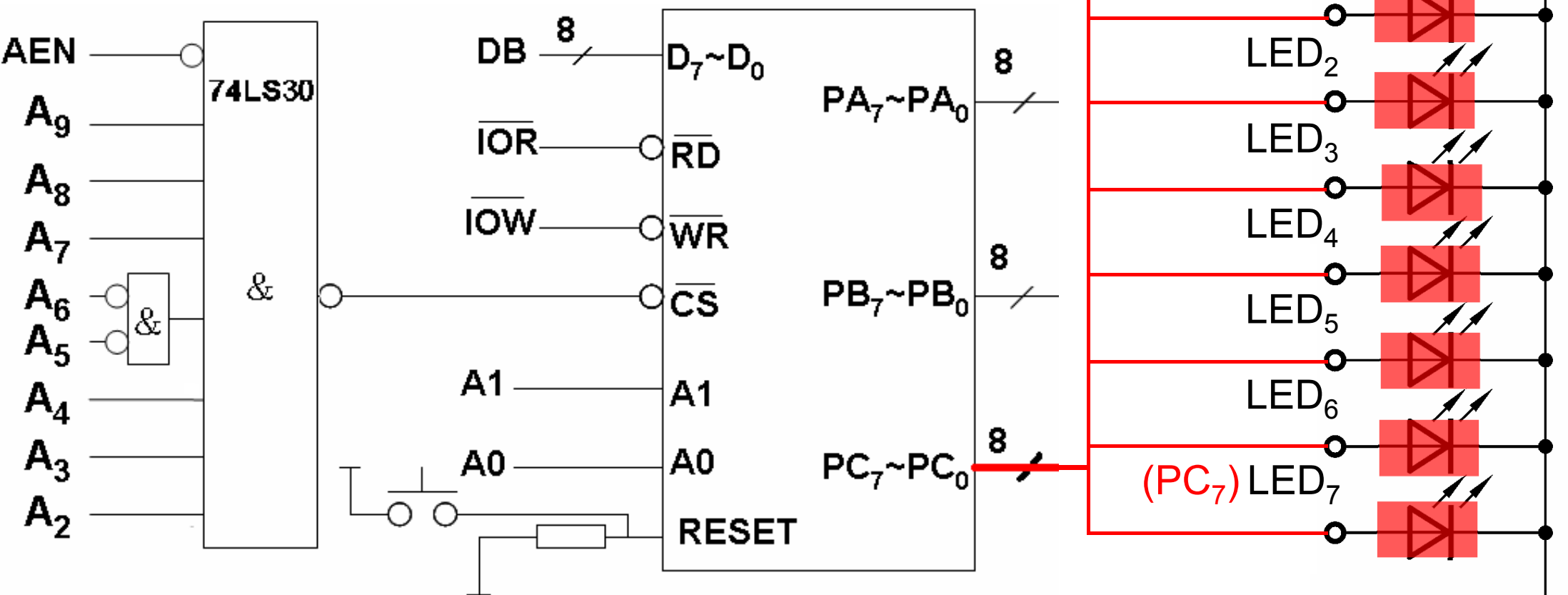
- 输出有锁存能力，输入有缓冲能力而无锁存能力。

思考（1）如何设置方式字？（2）程序流程？  
（3）8255的地址=？

## ● 方式0例子

■ 8255PC接8只LED，依次亮灭

◆  $PC_i = 1$  亮；  $PC_i = 0$  灭



注意：8255A地址: 39CH

MOV DX,39FH ;控制端口地址

MOV AL,80H

OUT DX,AL

MOV DX,39EH ;PC端口地址

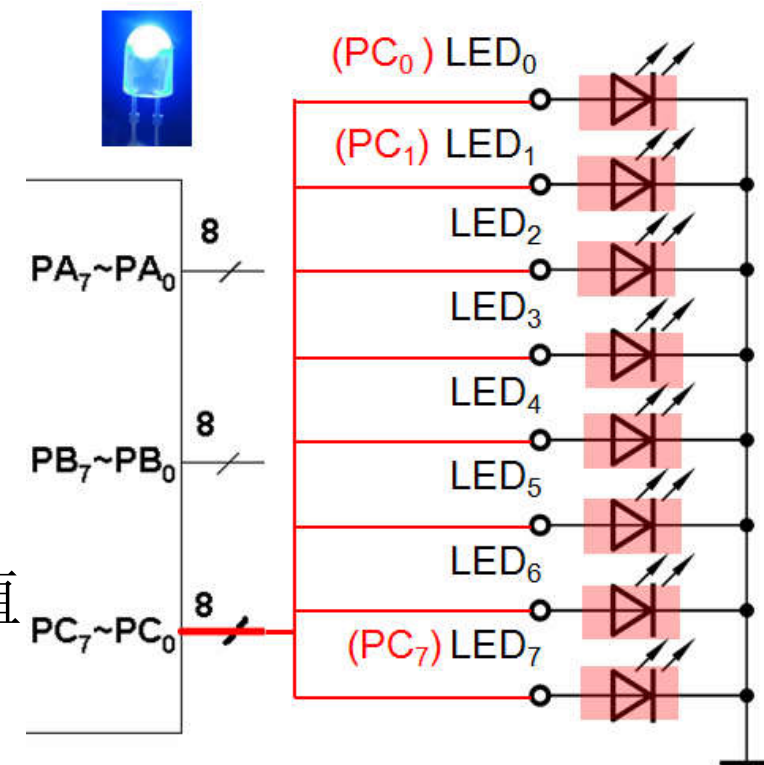
MOV AL,0000 0001B ;设定PC的值

AGAIN:OUT DX,AL ;点亮LED

DELEY\_1000MS ;延时1秒

ROL AL,1 ;逻辑左移 (LED下行)，改变PC值

JMP AGAIN



# 课堂练习：方式0的应用（查询方式）

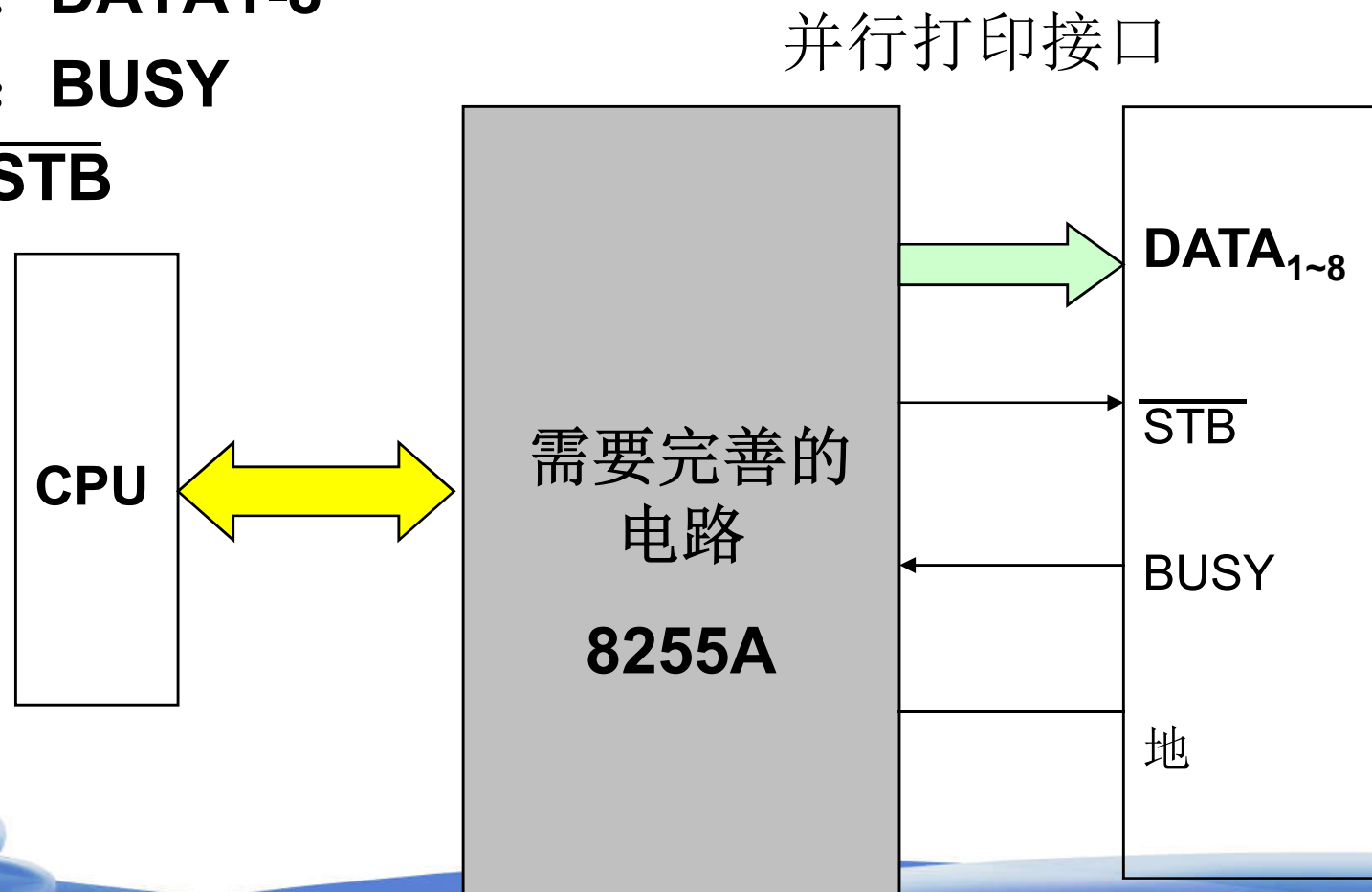
## ● 例子：并行打印机接口设计（查询方式）

■ 设计一个并行打印接口，CPU采用查询方式把存放在BUF缓冲区的256个字符（ASCII码）送去打印机进行打印。

■ 数据线：DATA1-8

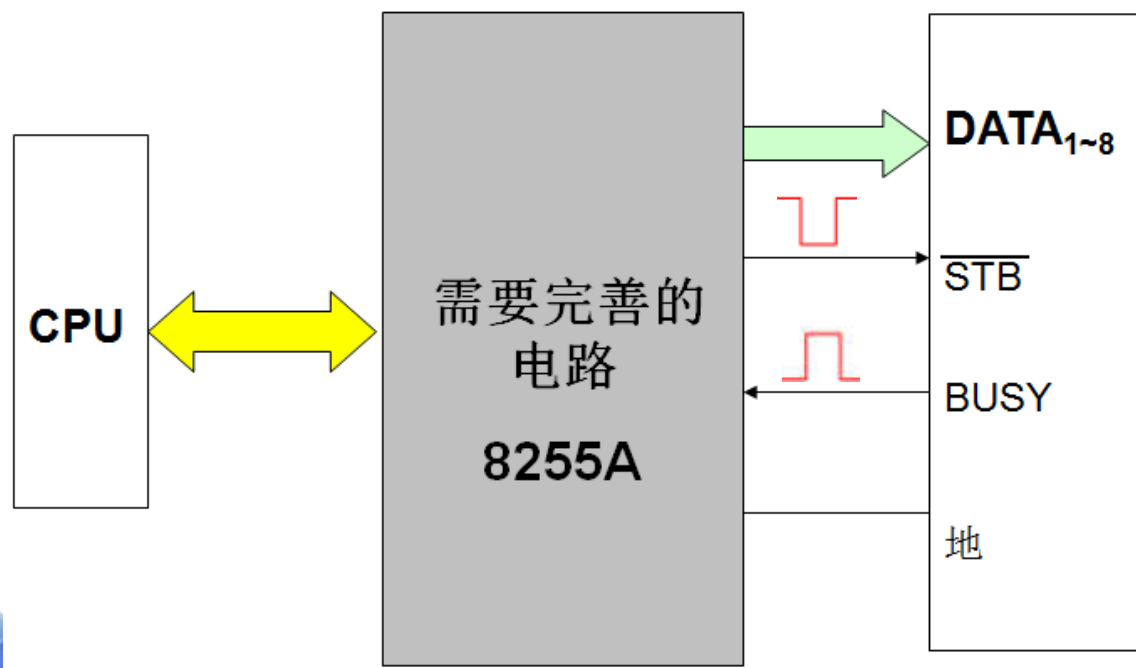
■ 状态忙：BUSY

■ 选通： $\overline{STB}$



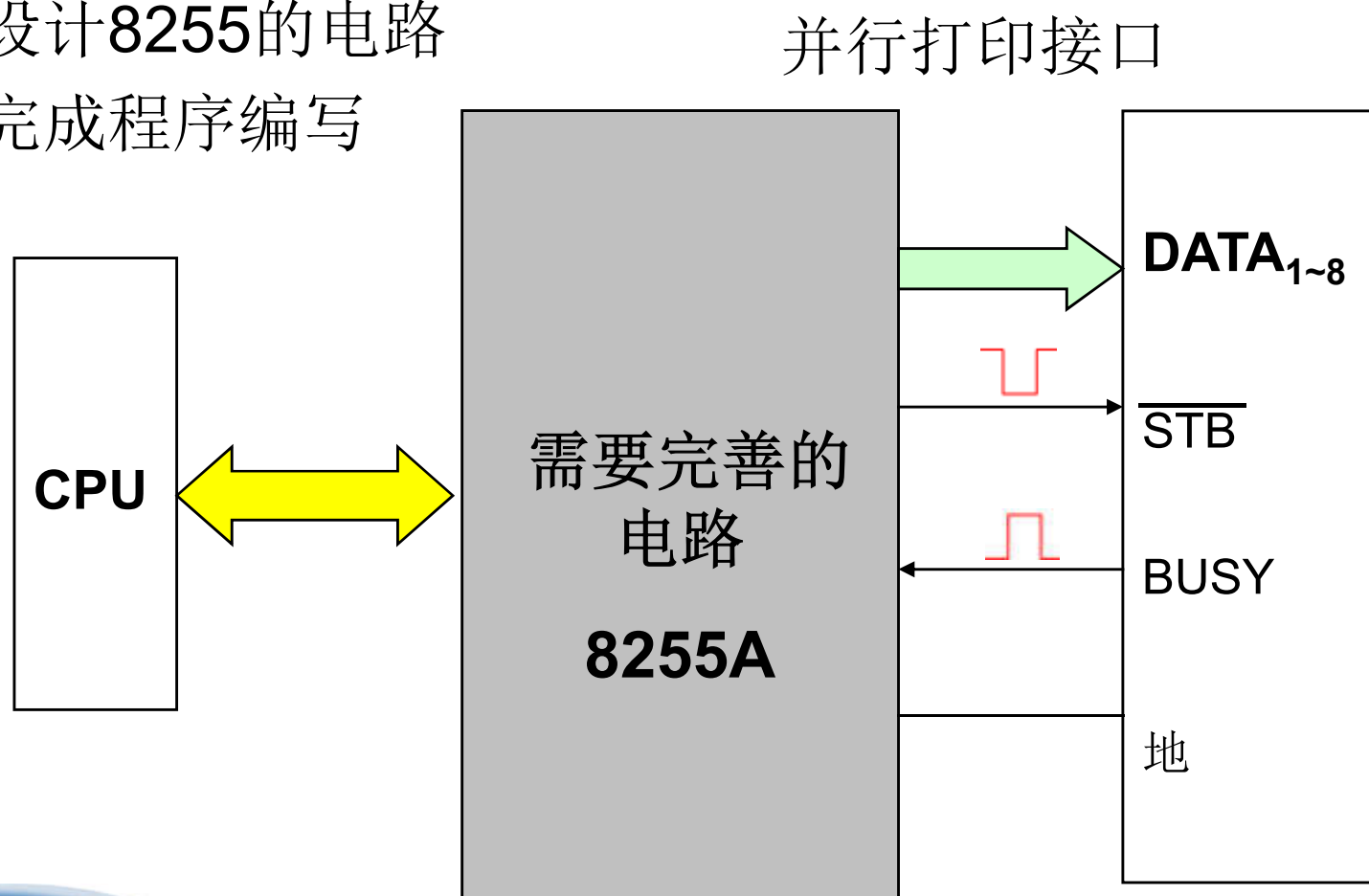
# 打印机与CPU之间传送数据的过程

- ① **CPU**首先查询**BUSY**。
  - 若BUSY=1，表示打印机忙，则等待；
  - 若BUSY=0，表示打印机不忙，则通过DATA<sub>1-8</sub>送新的打印数据。
- ②通过接口把数据送到**DATA<sub>1~8</sub>**上（**PA**端口）
- ③ **STROBE**(负脉冲 )把数据打入到打印机数据缓冲器。
- ④ 打印机收到数据后，发出“忙”（**BUSY=1**）信号
- ⑤每当一个字符打印完毕，撤消“忙”信号，置**BUSY=0**。
- 重复上述工作，直到把全部字符打印出来。



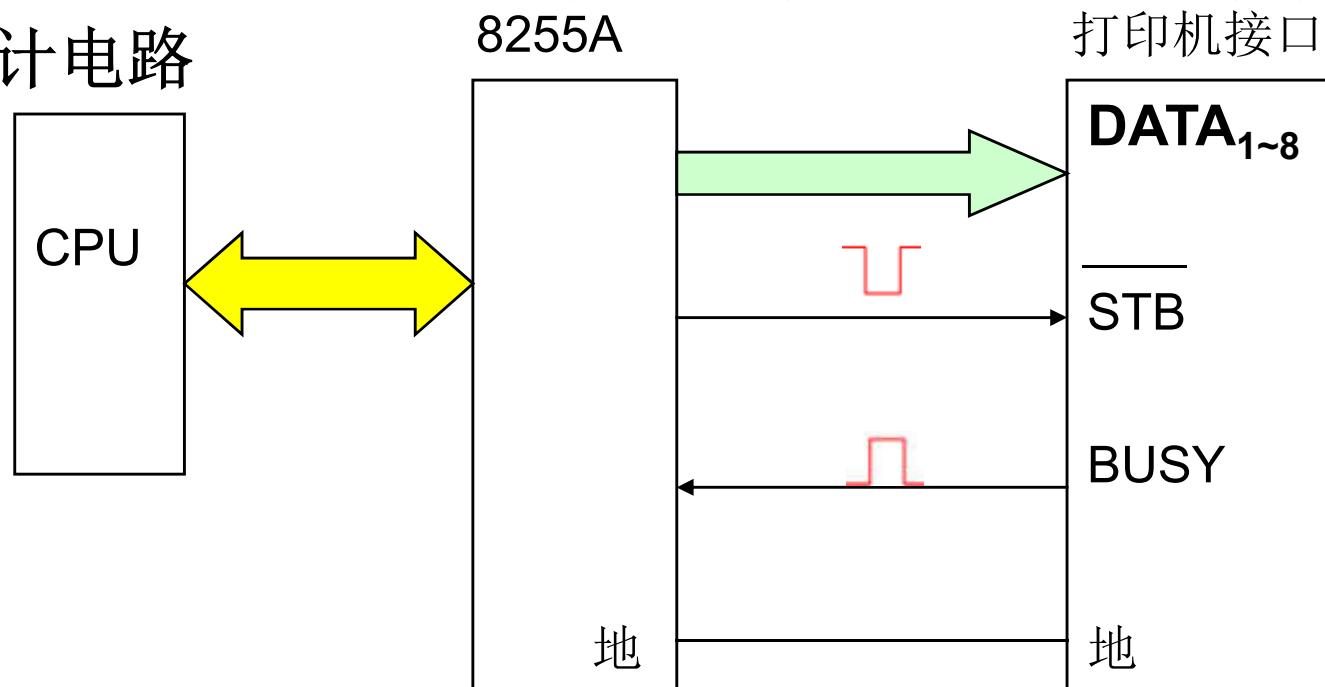
## ● 需要完成的工作

- 选定8255工作方式
- 设计8255的电路
- 完成程序编写





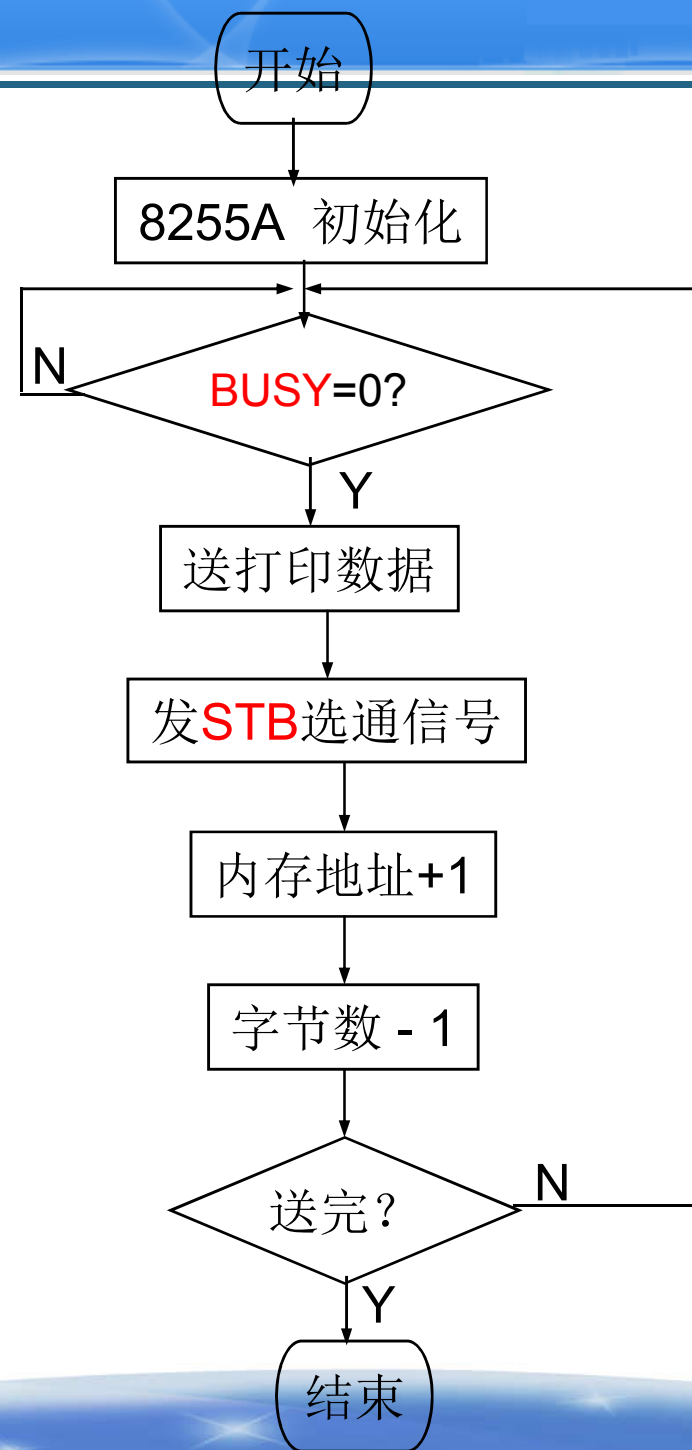
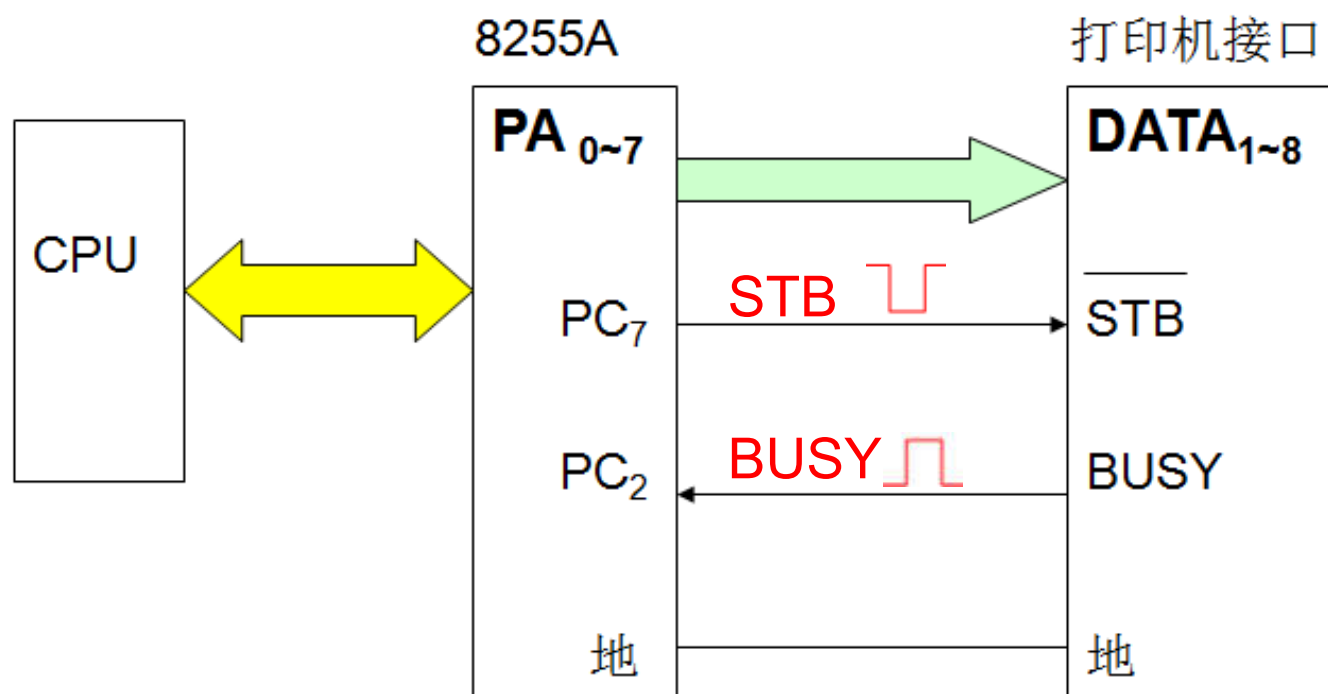
- 工作方式0：8位并行数据输出+2个联络信号（1个输出1个输入）
- 硬件设计电路



- PA输出，输出8位打印数据；
- PC<sub>7</sub>输出。数据选通信号（ $\overline{\text{STB}}$ ）；
- PC<sub>2</sub>输入。接收忙信号(BUSY)。
- 注意：PC<sub>7</sub>和PC<sub>2</sub>并非固定作控制用。这是0方式特点。

# 软件实现【8255地址300H ~ 303H】

## ● 流程



# 软件实现【8255地址300H ~ 303H】

MOV DX,303H ;8255A命令口

MOV AL,10000001B ;工作方式字

OUT DX,AL ;A口0方式输出, C<sub>4~7</sub>输出, C<sub>0~3</sub>输入

MOV AL,00001111B ;PC<sub>7</sub>置高, 使 $\overline{STB}$ =1

OUT DX,AL

MOV SI,OFFSET BUF ;打印字符的内存首址

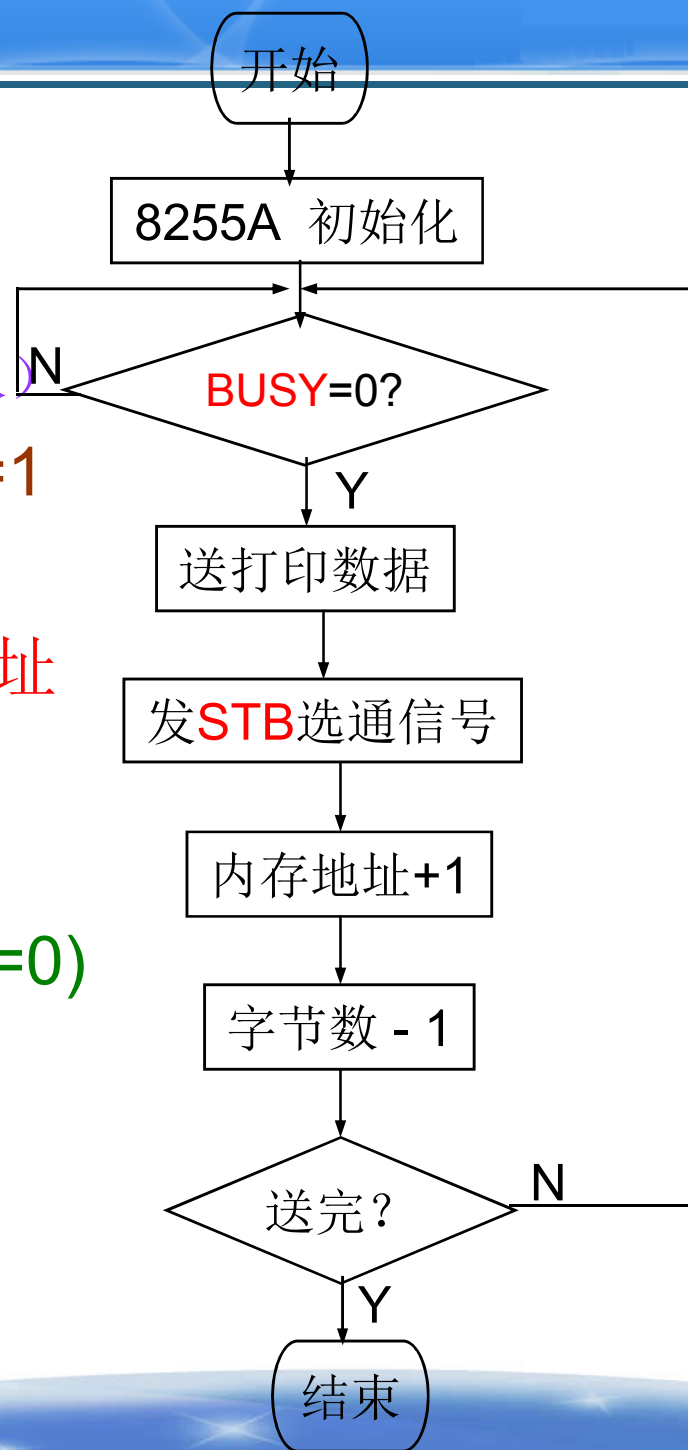
MOV CX,0FFH ;打印字符个数

**BUSY: MOV DX,302H ;PC口地址**

**IN AL,DX ;查BUSY=0? (PC<sub>2</sub>=0)**

AND AL,04H ;测试PC<sub>2</sub>位是否为0

JNZ **BUSY** ;忙, 则等待



# 软件实现(续)

；不忙，则向A口送数

MOV DX,300H

；PA口地址

MOV AL,[SI]

；从内存取数

OUT DX,AL

；送数到A口

MOV DX,303H

；8255A命令口

MOV AL,00001110B

；置 $\overline{STB}$ 为低 ( $PC_7=0$ )

OUT DX,AL

DELEY\_500MS ;负脉冲宽度 (延时)

MOV AL,00001111B

；置 $\overline{STB}$ 为高 ( $PC_7=1$ )

OUT DX,AL

INC SI

；内存地址加1

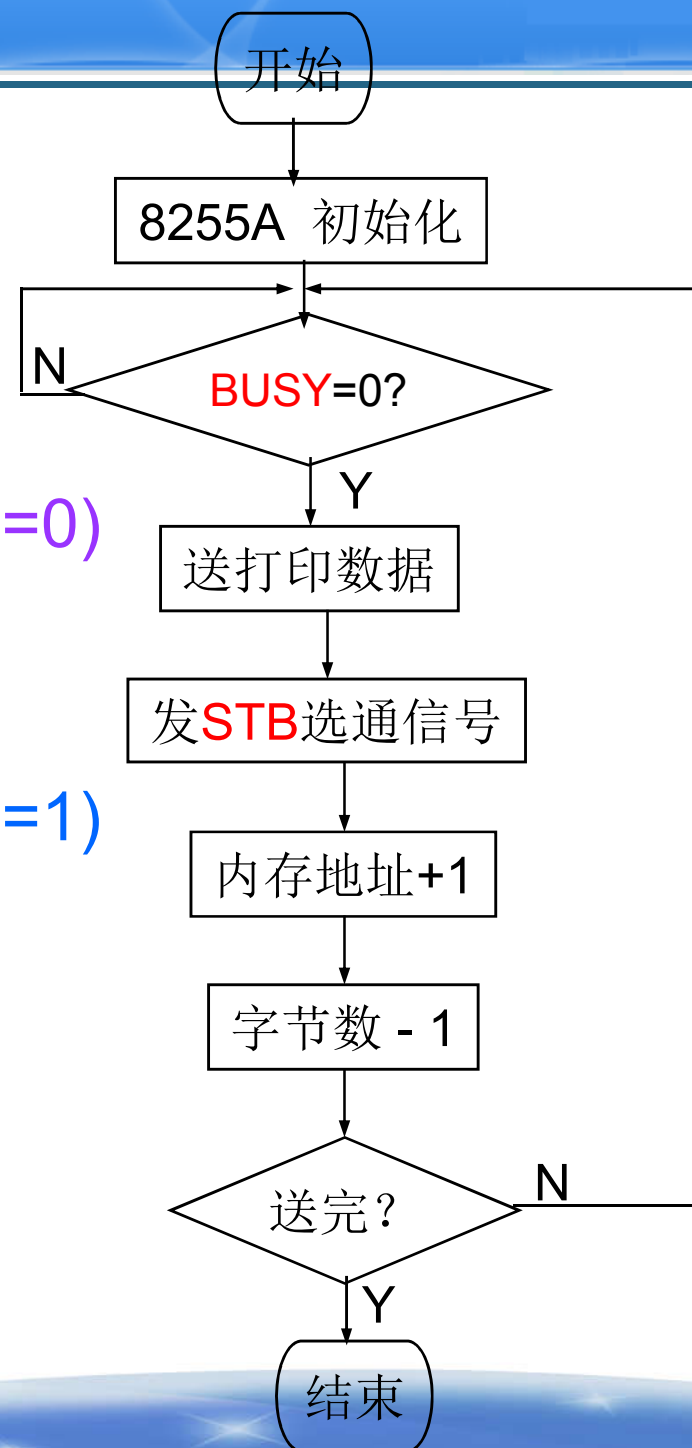
DEC CX

；字符数减1

JNZ BUSY

；未完，继续

BUF DB “256个ASCII字符代码……”



## ● 1方式（选通I/O方式、应答方式）

- 常用于查询（条件）传送或中断传送。

- ◆ 需设置**专用**联络线，联络线指定固定引脚且有固定时序。

- PA或/和PB为数据口，PC部分引脚固定充当联络信号

- ◆ A组：

- PA和PC<sub>3</sub>，PC<sub>4</sub>，PC<sub>5</sub> 3位联络信号

- PA和PC<sub>3</sub>，PC<sub>6</sub>，PC<sub>7</sub> 3位联络信号

- ◆ B组

- PB和PC<sub>0</sub>~PC<sub>2</sub>

- 传送过程会产生**状态字**，可作查询或中断用。

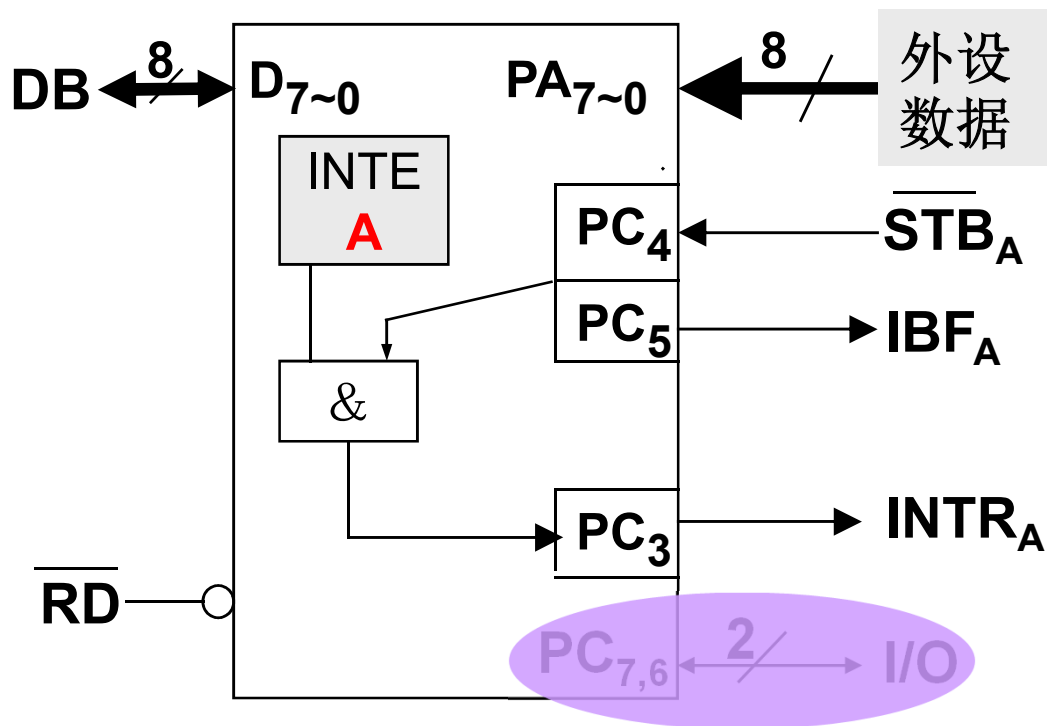
- ◆ 从**PC口**读取**状态字**。

- **单向**传送

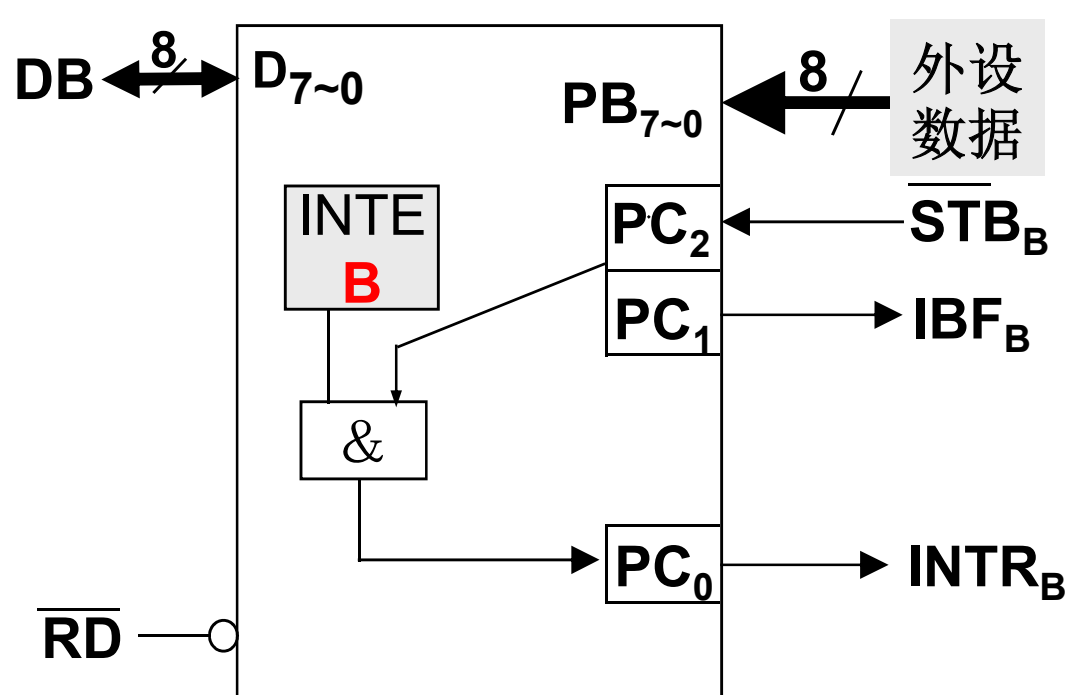
# 1方式输入的联络信号定义及时序

- 输入：外设数据 → 送到8255 → CPU读取
- PA用PC<sub>3~5</sub>做专用联络线，PB利用PC<sub>0~2</sub>做专用联络线。
- 注意：PC<sub>7,6</sub>在方式1输入模式时是自由I/O引脚

1方式输入（组A）



1方式输入（组B）



## ● 1方式输入时的联络信号

■ **STB**: **STROBE**, 选通信号, 低有效。

◆ 外设将数据锁存到PA口或PB口。

■ **IBF** (**I**nter**B**uffer **F**ull), 输入缓冲器满, 高有效

◆ 8255给外设: 数据未被CPU取走。

■ **INTR** (**I**nterrupt **R**equest), 高电平有效

◆ 8255给CPU的“中断”信号。INTR变高3个必要条件

□ STB变高;

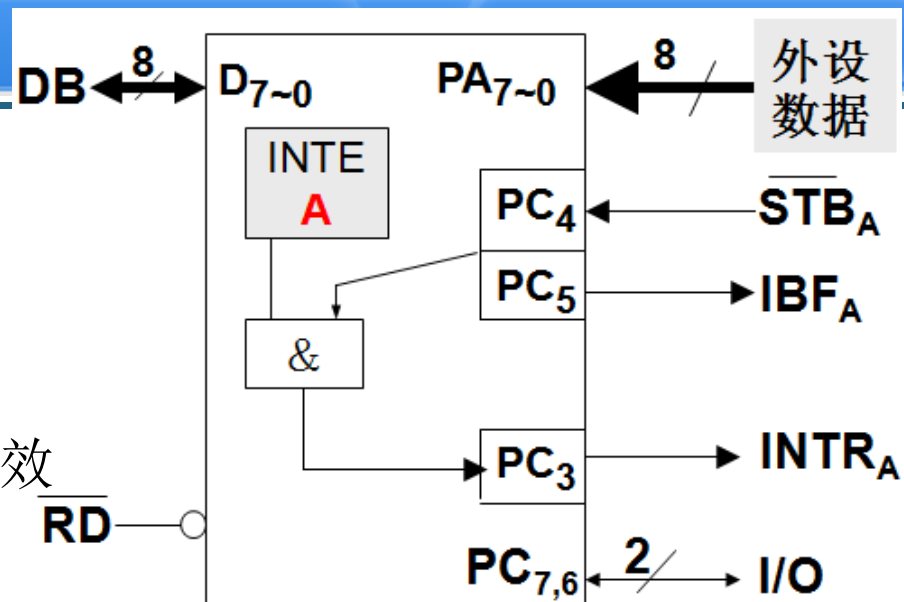
□ IBF变高;

□ 中断请求被允许 ( INTE = 1)

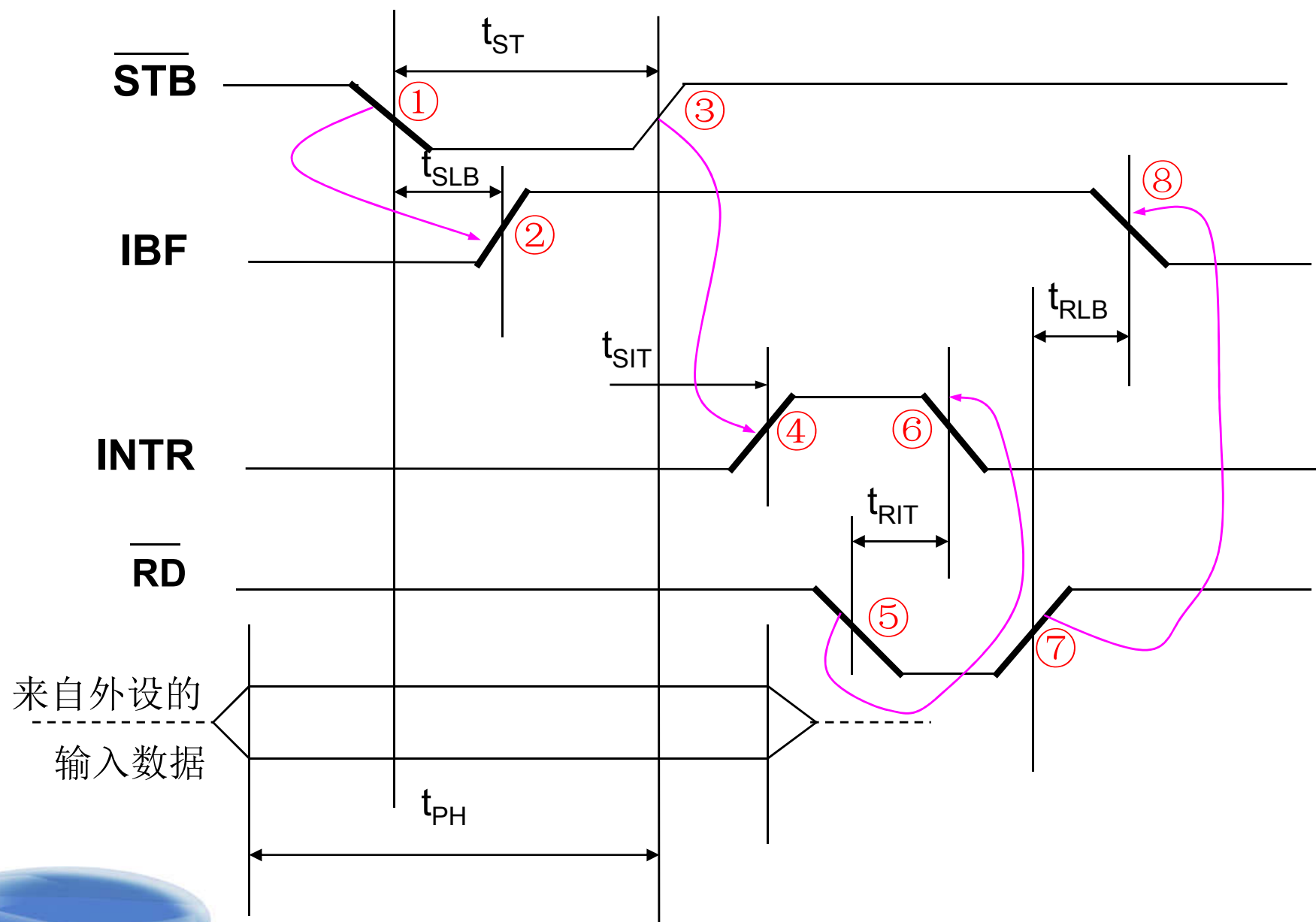
■ **INTE**: 中断允许 ( **INTE<sub>A</sub>**, **INTE<sub>B</sub>**允许PA或PB产生中断? )

◆ A组INTE<sub>A</sub>由PC<sub>4</sub>按位置位/复位来控制。

◆ B组INTE<sub>B</sub>由PC<sub>2</sub>按位置位/复位来控制。



# 1方式输入时的时序



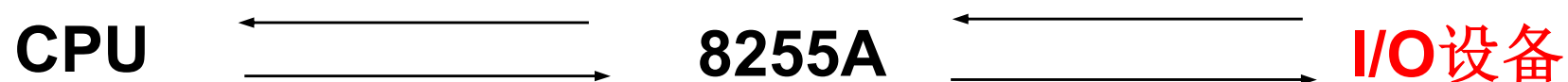


## ● 1方式输入时的时序

- 1、外设准备好数据，发送 $\overline{\text{STB}}$ 信号，把数据输入8255。
- 2、在 $\overline{\text{STB}}$ 下降沿，数据已锁存到8255缓冲器后，引起IBF变高，表示8255A“输入缓冲器满”，禁止输入新数据。
- 3、在 $\overline{\text{STB}}$ 上升沿后，在中断允许（ $\text{INTE}=1$ ）的情况下IBF的高电平产生中断请求，使INTR上升变高，通知CPU，接口中已有数据，请求CPU读数。CPU接受中断请求后，转到相应的中断子程序。在子程序中执行IN指令，将缓冲器中的数据取走。
- 4、CPU读操作（ $\overline{\text{RD}}$ ）信号的下降沿使INTR复位，撤消中断请求，为下一次中断请求作好准备。
- 5、 $\overline{\text{RD}}$ 信号的上升沿延时一段时间后清除IBF使其变低，表示接口的输入缓冲器变空，允许外设再次输入新数据。

## ● 1方式输入时的时序

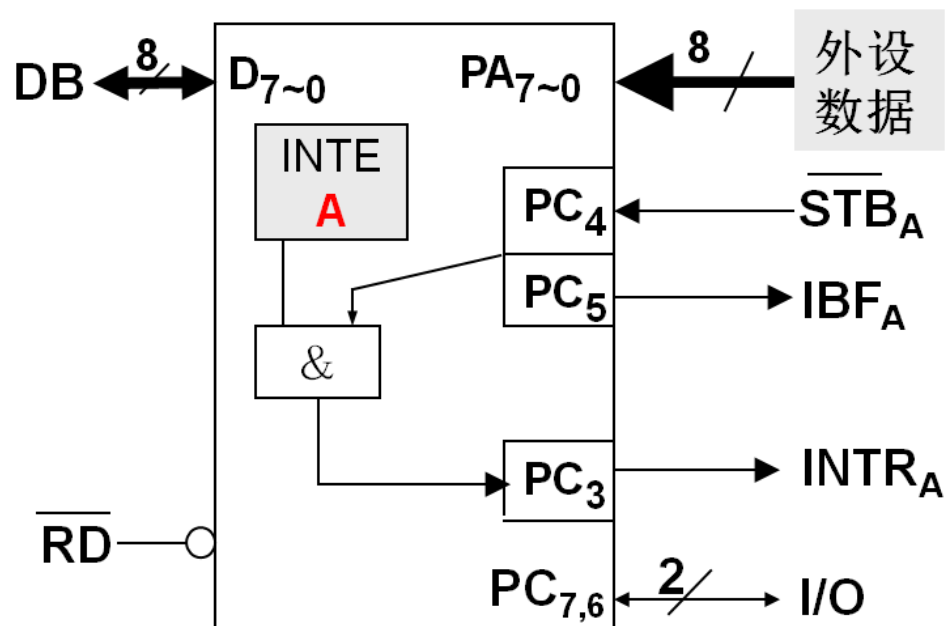
- 数据从I/O设备发出，通过8255A，送到CPU。



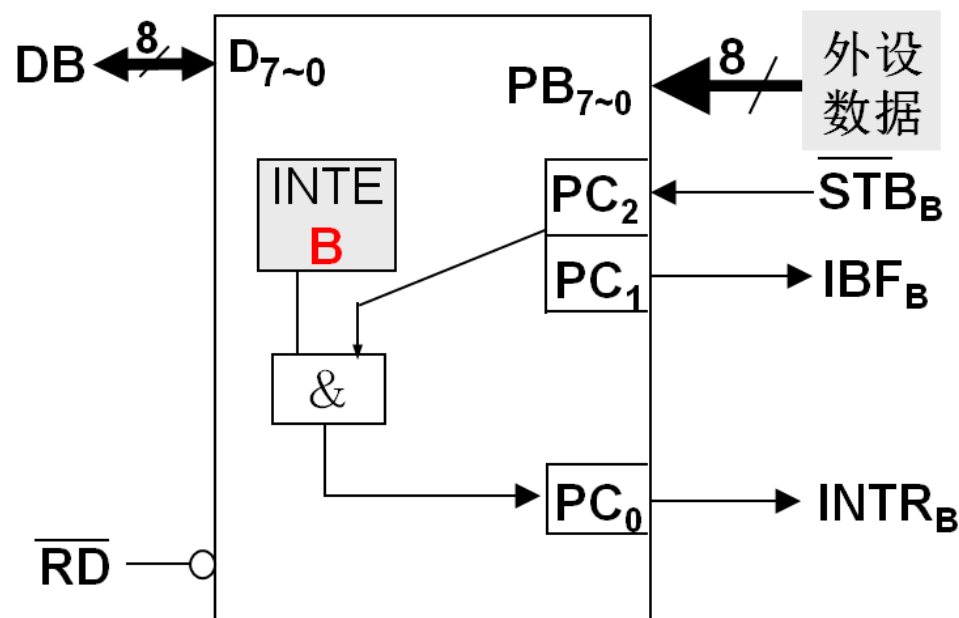
# 补充说明

- 若 **A组**和**B组**都工作于方式1，则**PC**还留下**PC<sub>6-7</sub>**2位，可以做普通**I/O**用。
  - 这2位可以由**方式控制字D3位**指定作为输入或输出。
  - 也具有**按位置位/复位**功能。
- 若**A组**或**B组**只有一组工作于**方式1**，则余下的**13位**可以工作在**方式0**，可以做普通**I/O**用。

1方式输入（**组A**）



1方式输入（**组B**）



## ● 1方式输入时的A,B两组的方式控制字

A 组方式控制字

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>
1	0	1	1	I/O	×	×	×

方式 1

端口 A 输入

PC<sub>6,7</sub>  
1=输入  
0=输出

## ● 注意：D3位

■ PC<sub>7,6</sub>

B 组方式控制字

1	×	×	×	×	1	1	×
---	---	---	---	---	---	---	---

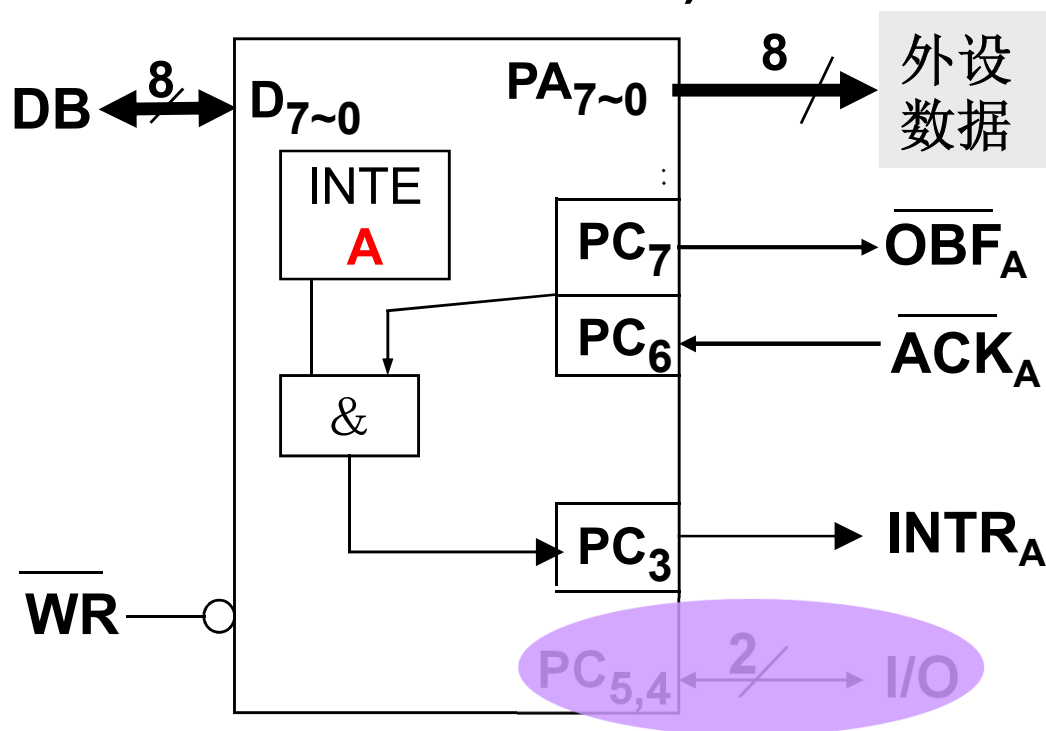
端口 B 输入

端口 B 方式 1

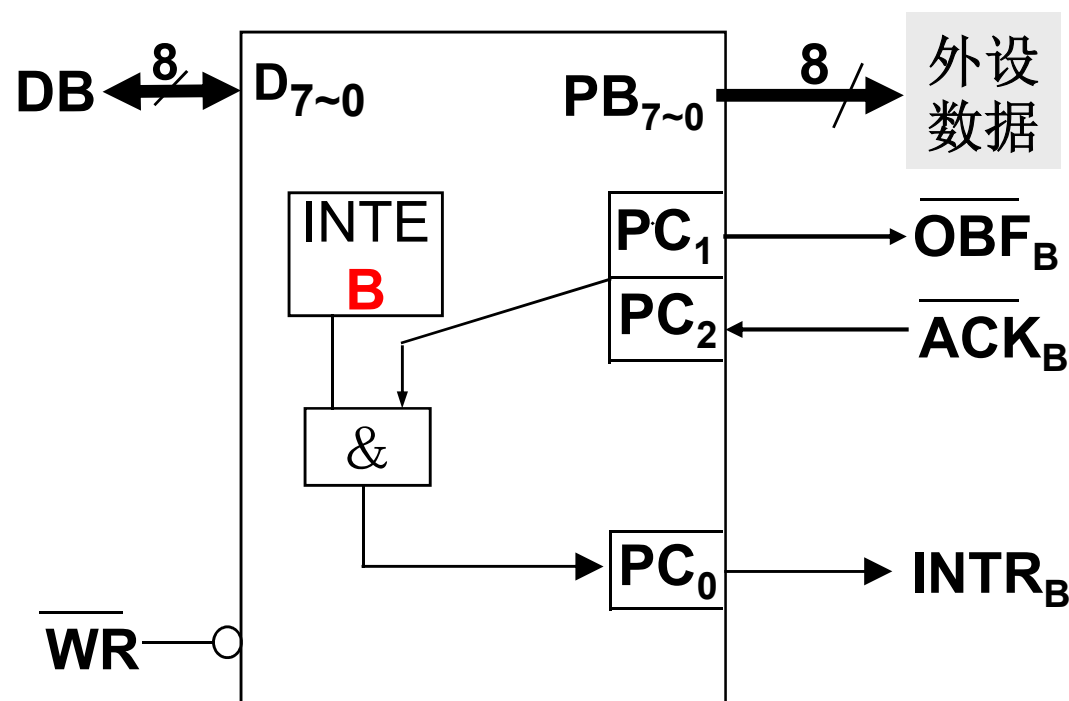
# 1方式输出的联络信号定义及时序

- 输出: CPU数据 → 写入8255 → 外设
- PA用PC<sub>3,6,7</sub> 3位, PB用PC<sub>0~2</sub> 3位做专用联络信号。
- 注意: PC<sub>5,4</sub>是自由I/O引脚

1方式输出 (组A)



1方式输出 (组B)



## ● 1方式输出时的联络信号

■  $\overline{\text{OBF}}$ , “输出缓冲器满”低有效

◆ CPU已将数据写到8255输出口

■  $\overline{\text{ACK}}$ , 外设对8255A应答低有效。

◆ 外设已经取走8255端口的数据。

◆  $\overline{\text{ACK}}$ 下降一段时间后使 $\overline{\text{OBF}}$ 变高, 为下次输出作好准备。

■ INTR, 8255A给CPU的“中断请求”信号, 高电平有效。

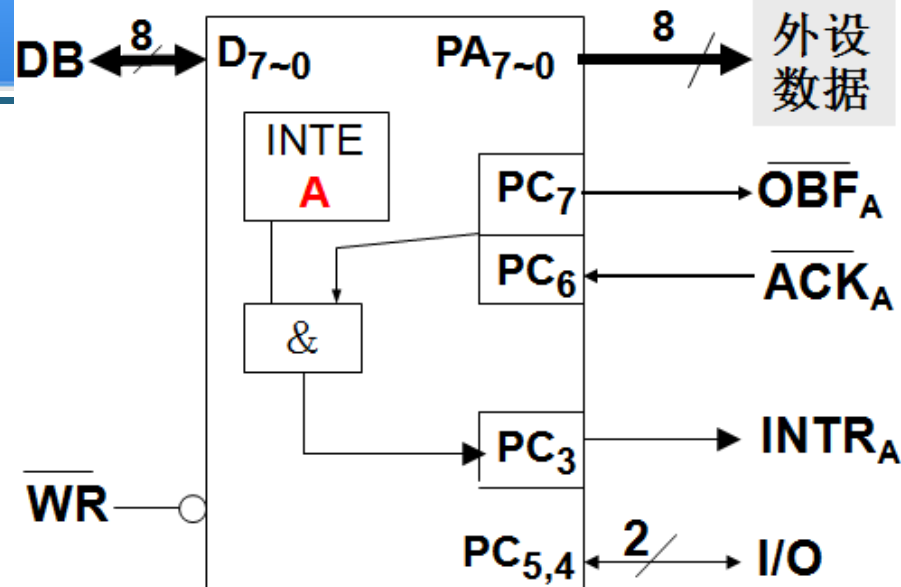
◆ 作为中断请求信号, 请求CPU向8255A输出数据。

◆ INTR变高条件:  $\overline{\text{OBF}}$ 、 $\overline{\text{ACK}}$ 和 INTE都为高电平。

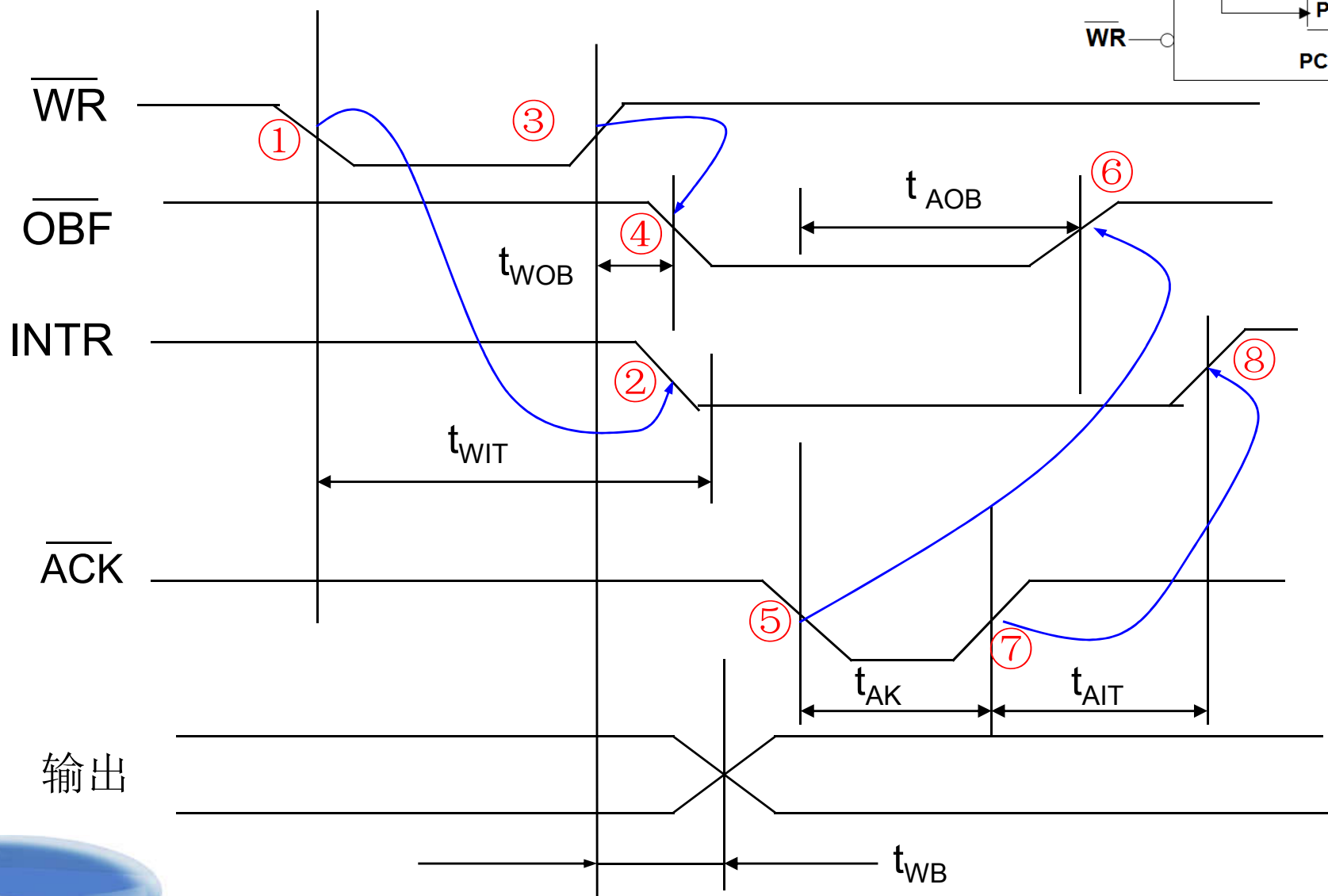
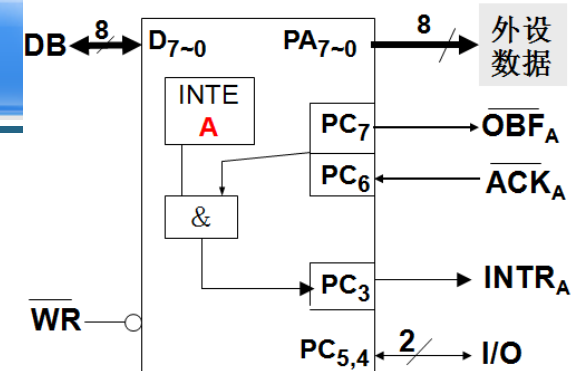
■ INTE: 中断允许

◆ A组 $\text{INTE}_A$ 由 $\text{PC}_6$  按位置位/复位来控制。

◆ B组 $\text{INTE}_B$ 由 $\text{PC}_2$  按位置位/复位来控制。



# ● 1方式输出时的时序（假定在中断方式下）



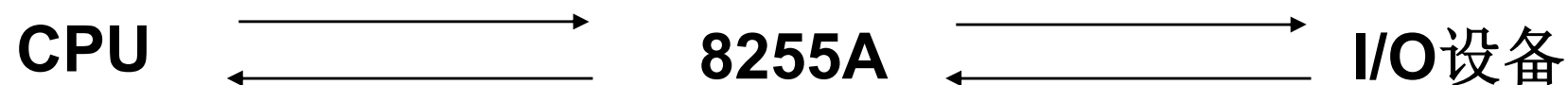
## ● 1方式输出时的时序（假定在中断方式下）

- 1、数据输出时，CPU准备好数据并写到8255输出数据寄存器。当CPU向8255A写完一个数据后， $\overline{WR}$ 的上升沿使 $\overline{OBF}$ 有效，表示8255A的输出缓冲器已满，通知外设读取数据。
- 2、 $\overline{WR}$ 的下降沿同时使中断请求INTR变低，封锁中断请求。
- 3、外设得到 $\overline{OBF}$ 有效的通知后，开始读数。当外设读取数据后，用 $\overline{ACK}$ （低电平）回答8255A，表示数据已收到。
- 4、 $\overline{ACK}$ 下降沿将 $\overline{OBF}$ 置高，表示输出缓冲器变空，为下一次输出作准备。
- 5、在中断允许的情况下 $\overline{ACK}$ 上升沿使INTR变高，产生中断请求。CPU在中断服务程序中，向8255A写(OUT)新数据。



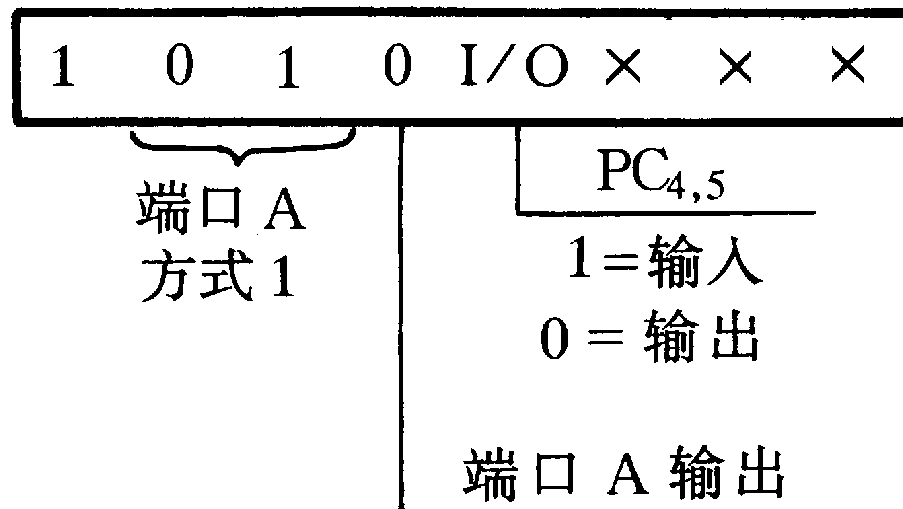
## ● 1方式输出时的时序

■ 数据从CPU，通过8255A，送到I/O设备有如下4步：

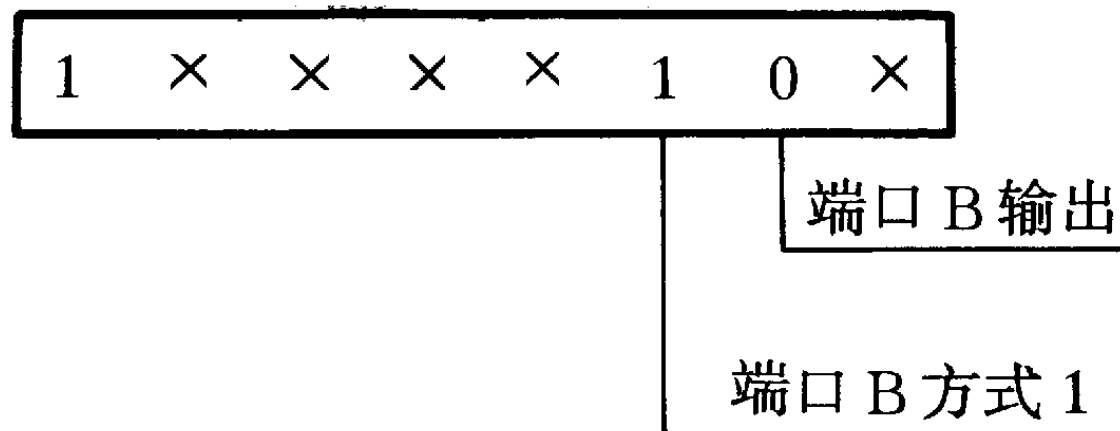


## ● 1方式输出时的A,B两组的方式控制字

A组方式1控制字



B组方式1输出控制字

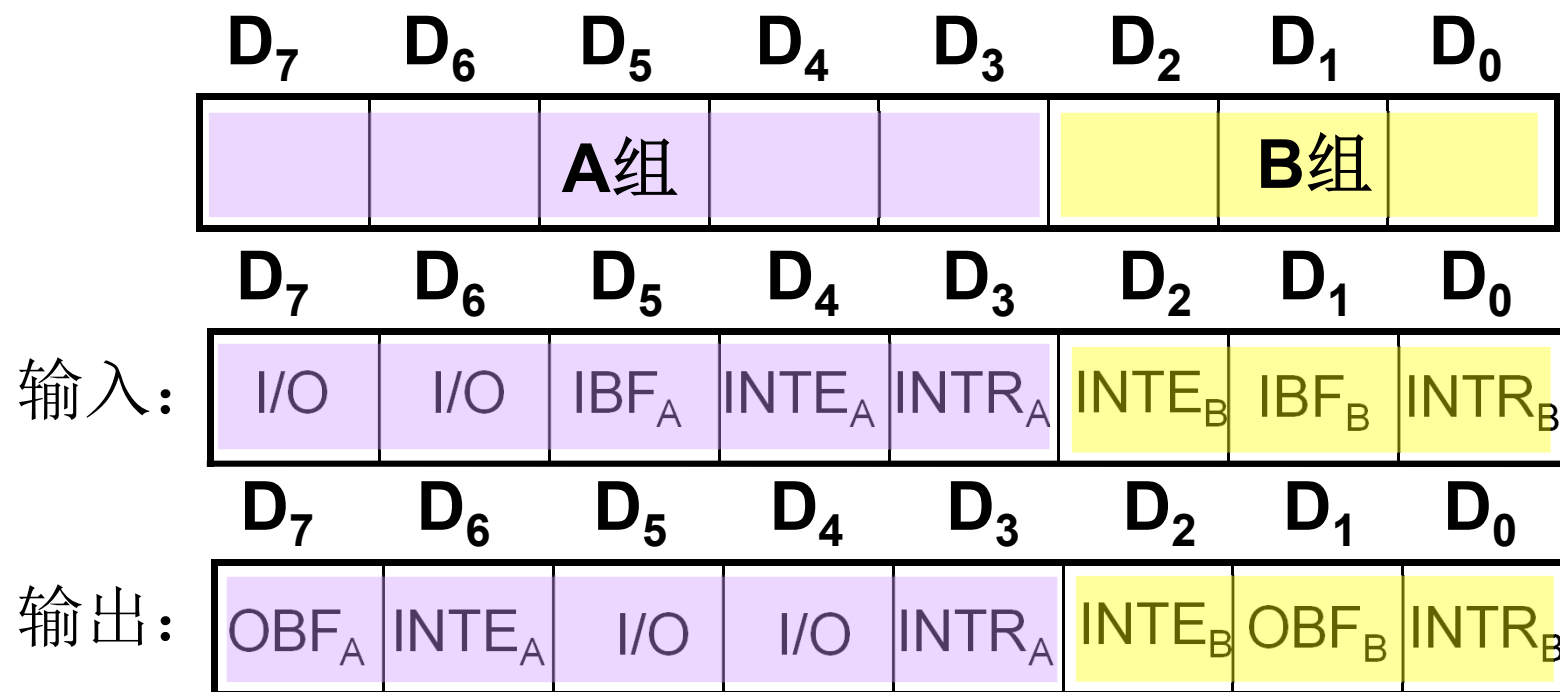


## ● 注意：D3位

■ PC<sub>5,4</sub>

# 1方式的状态字

- 提供8255的状态；辅助CPU识别中断源（不提供中断矢量）
- 状态字从PC口读取，与外部引脚无关。
- 状态字的格式



- 说明

- 供查询的状态位有INTR、IBF、OBF等。
- 在查询方式下，一般查询INTR位（比IBF、OBF 两位可靠）
- INTE位由用户用置位/复位指令设定，而非I/O操作自动产生。

# 1方式的应用和设计

## ● 基本思路

- (1) 确定A组和B组的输入/输出;
- (2) 把PC联络线与外设连好, 与CPU相应引脚连好。
- (3) 确定是采用中断方式还是查询方式和CPU通信

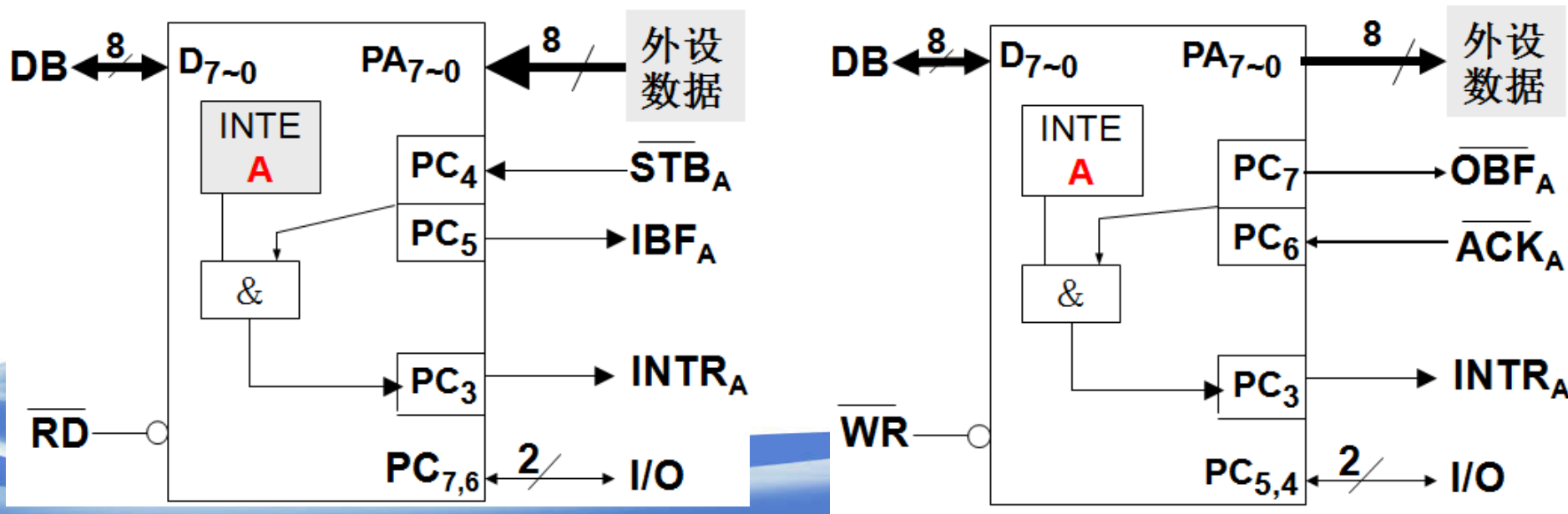
◆ 中断方式: 把INTR线接到CPU或8259A的INTR;

□ 8255不提供中断矢量, 可结合系统的中断控制器使用。

◆ 查询方式, 把INTR线空着, 程序查寻状态字INTR位。

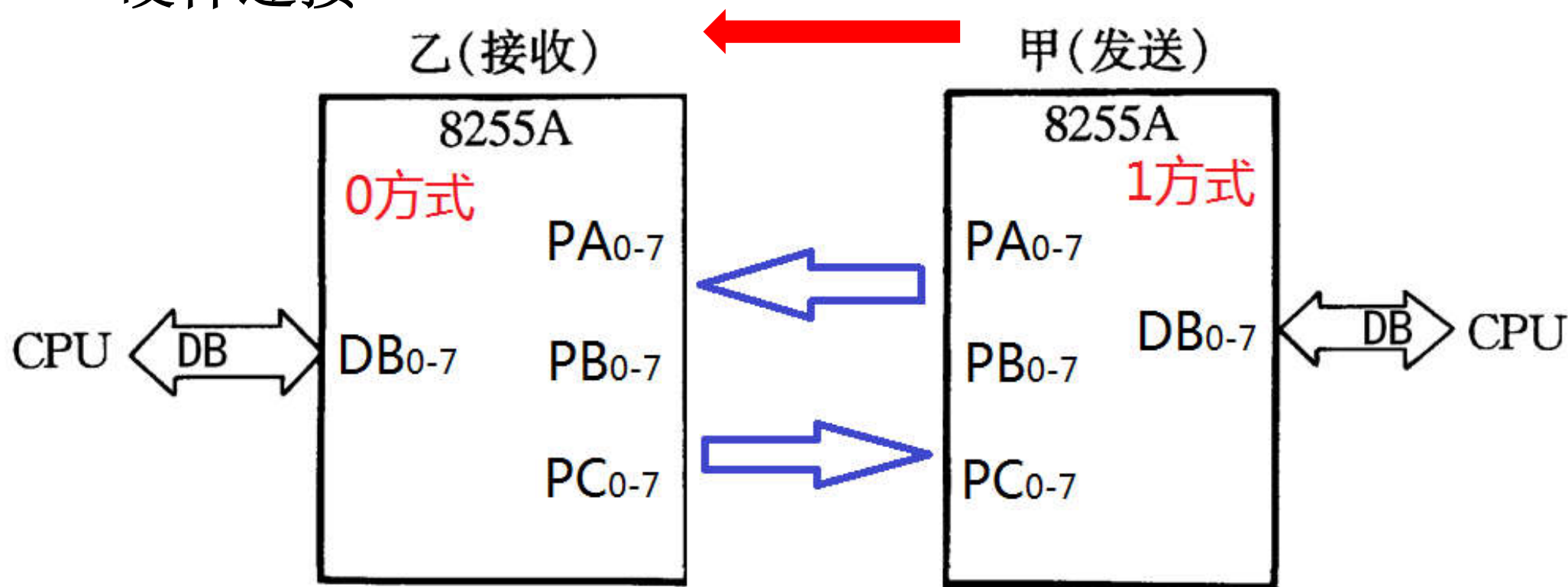
□ 查询INTR位比查询IBF位或OBF位可靠性更高。

□ 注意: 查询INTR位时, 要确保INTE已经置位。



# 1方式应用例子

- 甲乙两机（查询方式）传送**1K**字节[起址**0030:0000**]
  - 甲机发送（**方式1**）→ 乙机接收（**方式0**）。
  - 8255端口地址：300H~303H。
- 硬件连接



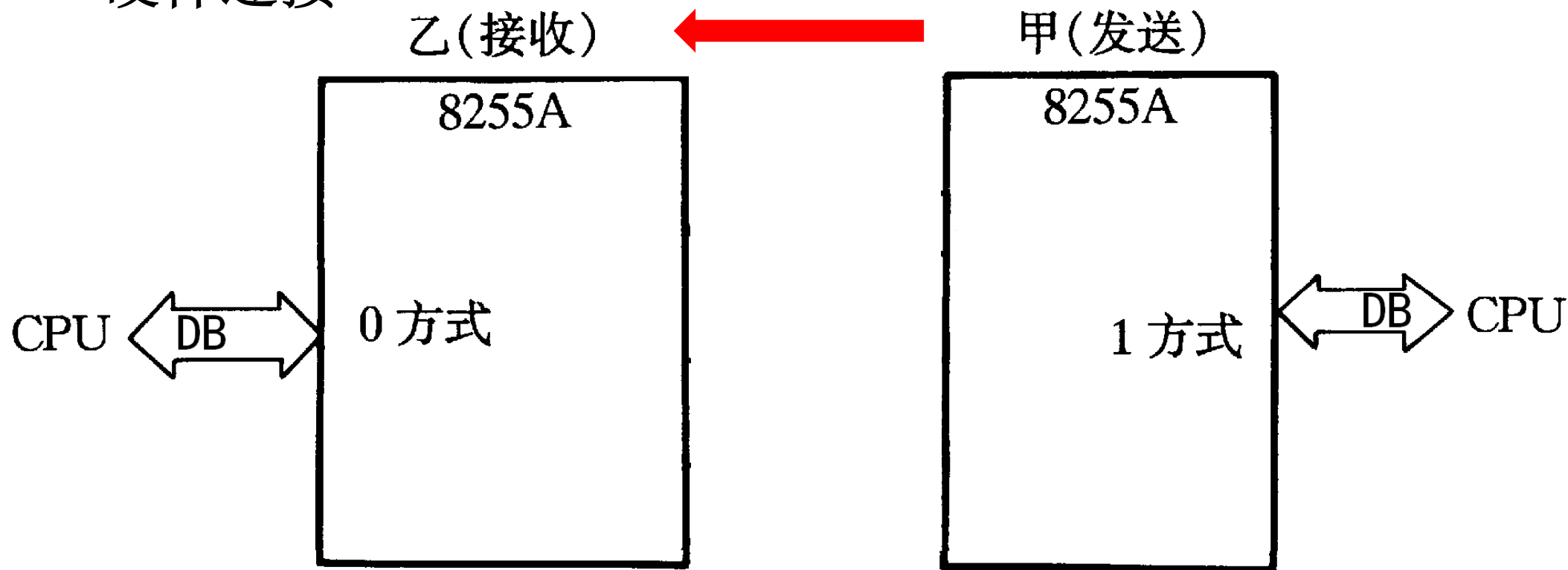
# 1方式应用例子

- 甲乙两机（查询方式）传送**1K**字节[起址**0030:0000**]

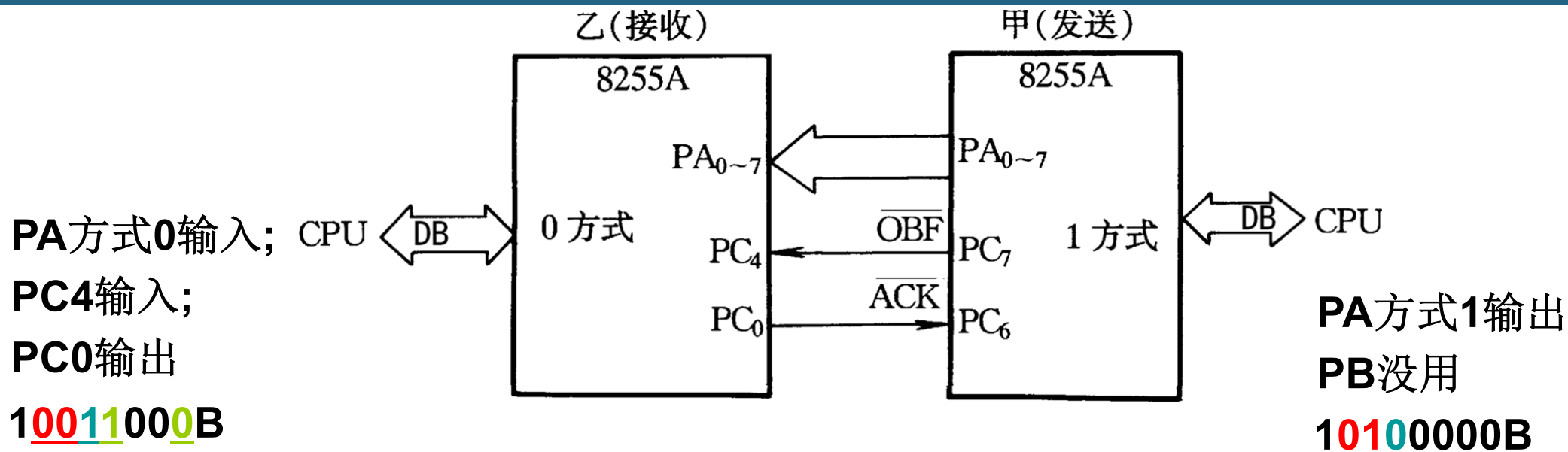
- 甲机发送（**方式1**）→ 乙机接收（**方式0**）。

- 8255端口地址：300H~303H。

- 硬件连接



# 确定方式控制字和软件流程



## 乙机流程:

1. 设置控制字,  $PC_0$  置1, 初始化:  
(初始化数据首址和数量)
2. 查询  $PC_4$  (是否有数据? 负脉冲)  
有: 读PA接收1个数据  
 $PC_0$  产生负脉冲
3. GOTO 2

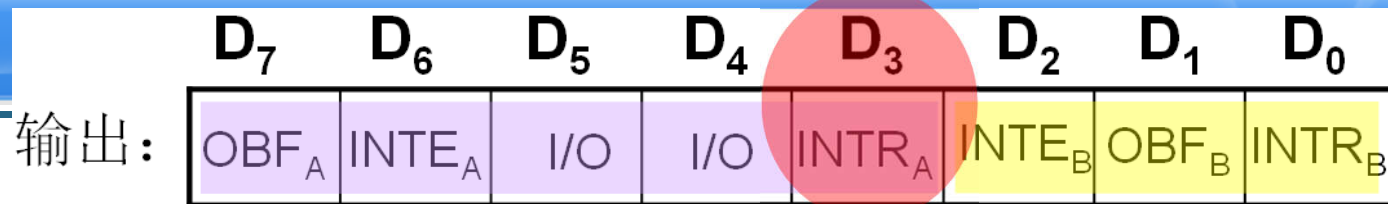
## 甲机流程:

1. 设置控制字, 使能中断, 初始化:  
(初始化数据首址和数量)
2. PA口发送1个数据  
(OBF → 负脉冲)
3. 轮询  $INTR_A$ , 决定是否发新数据  
是: GOTO 2  
否: GOTO 3

## ● 甲机:发送程序（方式1）[起址0030:0000, 1K]

MOV	DX, 303H	; 8255A命令口
MOV	AL, 10100000B	; 方式字: 端口A方式1
OUT	DX, AL	; 输出方式字
MOV	AL, 01000000B;	; 置输出中断使能INTE <sub>A</sub> =1
OUT	DX, AL	; PC <sub>6</sub> 置“1”
MOV	AX, 030H	; 待发送数据的首地址
MOV	ES, AX	
MOV	BX, 00H	; 待发送数据的偏移
MOV	CX, 3FFH	; 置发送字节数
MOV	DX, 300H	; 置8255A数据字地址
MOV	AL, ES:[BX]	; 取第1个待发送的数据
OUT	DX, AL	; 输出第1个数据
INC	BX	; 指向下一个数
DEC	CX	; 字节数减 1





```

L: MOV     DX, 302H           ; 8255A状态口 (即PC)
   IN      AL, DX             ; 输入状态
   AND     AL, 08H           ; 检查INTRA: 是否已被乙方接收?
   JZ      L                  ; 若无中断请求则等待
   MOV     DX, 300H           ; 置数据口地址, 准备输出下一个数
   MOV     AL, ES:[BX]        ; 取数据
   OUT     DX, AL             ; 输出数据
   INC     BX                 ; 指向下一个数
   DEC     CX                 ; 字节数减 1
   JNZ     L                  ; 若未发送完, 继续循环
   MOV     AX, 4C00H
   INT     21H                ; 发送完成, 返回DOS

```

# 乙机接收程序（方式0）[起址0040:0000]

```
MOV    DX, 303H      ; 置8255A命令口地址
MOV    AL, 10011000B ; A方式0、PC4输入、PC0输出
OUT    DX, AL        ; 方式字
```

```
MOV    AL, 00000001B ; PC0置1。按位置位/复位控制字
OUT    DX, AL        ; 使 ACK高电平
```

```
MOV    AX, 040H      ; 准备接收区地址ES:BX
MOV    ES, AX        ; 首址ES:BX = 40:00
MOV    BX, 00H
```

```
MOV    CX, 3FFH      ; 置字节数
```

```
L1:    MOV    DX, 302H ; PC口
        IN     AL, DX   ; 查PC4 (OBF)=0? 是否有数据输入
        AND    AL, 10H
        JNZ    L1      ; 无数据, 等待。继续查询PC4
```

```
MOV    DX, 300H      ; PA数据口
IN     AL, DX         ; 输入数据
```

```
MOV    ES:[BX], AL   ; 存入内存
```

MOV DX, 303H  
MOV AL, 00000000B ; PC<sub>0</sub>置0  
OUT DX, AL ; 产生ACK信号: 低电平

NOP

MOV AL, 00000001B ; PC<sub>0</sub>置1  
OUT DX, AL ; ACK变高

INC BX

DEC CX ; 字节数减1, 判断是否接收完成  
JNZ L1 ; 未接收完, 循环

MOV AX, 4C00H

INT 21H ; 接收完, 返回DOS