



第10章 并行I/O接口

1

可编程并行I/O接口芯片8255A

2

8255A的应用举例

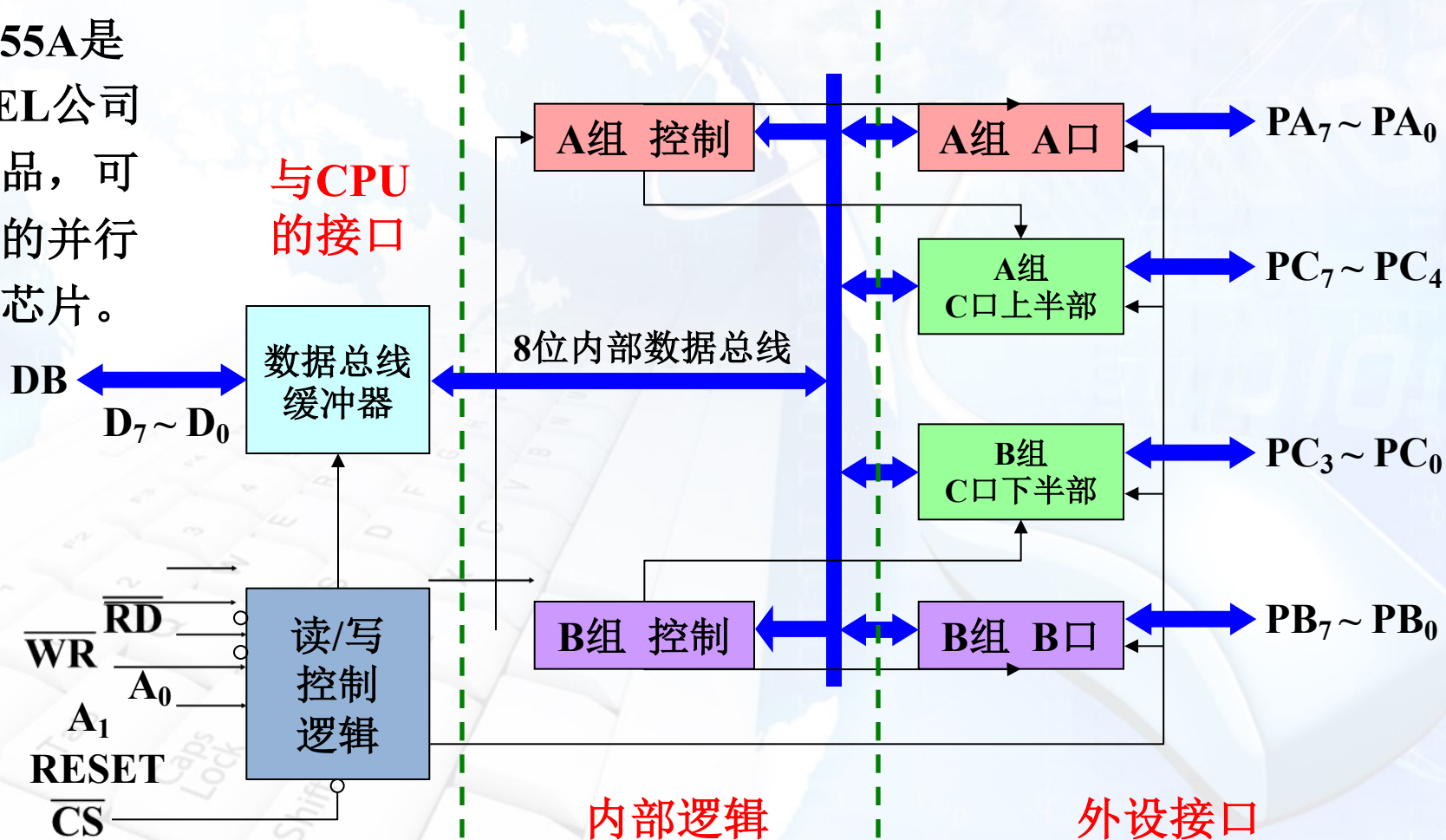
10.1 可编程并行I/O接口芯片8255A



一、8255A的内部结构及外部引脚

1. 8255A的内部结构

8255A是INTEL公司的产品，可编程的并行接口芯片。



2. 8255A的外部引脚

8255A为双列直插式，40引脚：

1) **24根端口数据线**——接外设：

$PA_7 \sim PA_0$ 为A口数据线；

$PB_7 \sim PB_0$ 为B口数据线；

$PC_7 \sim PC_0$ 为C口数据线。

2) **8根系统数据线**——接CPU数据总线 $D_7 \sim D_0$ 。

3) **电源线**：+5V，地线

4) **6根输入控制线**：

RESET：复位信号， $RESET=1$ 时，8255A内部复位，所有内部寄存器清零，A、B、C三个端口自动为输入口；

\overline{CS} ：片选信号， $\overline{CS}=0$ 时，该芯片被选中；

\overline{RD} ：来自CPU的I/O读命令；

\overline{WR} ：来自CPU的I/O写命令；

A_1 、 A_0 ：通常接CPU的地址总线 A_1 、 A_0 。

3. 8255A的端口编址与读/写操作

1) 端口编址

8255A中有4个端口寄存器:

$A_1 A_0$ | 选中...

0 0 | A口数据寄存器

0 1 | B口数据寄存器

1 0 | C口数据寄存器

1 1 | 控制口寄存器

控制口寄存初始化控制字

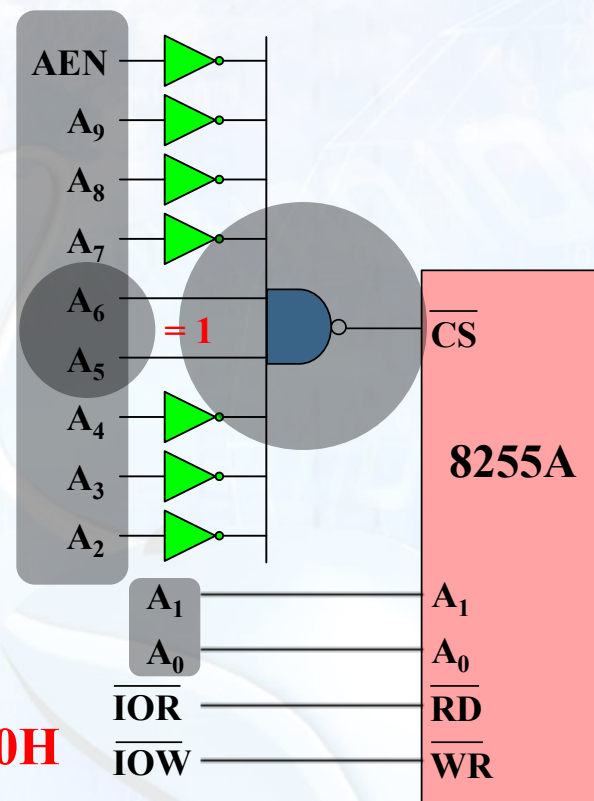
A数据口地址 = 60H

B数据口地址 = 61H

C数据口地址 = 62H

控制口地址 = 63H

系统产生片选信号的译码电路如图, 则:



10.1 一、8255A的内部结构及外部引脚



2) 读/写操作

$\overline{\text{CS}}$	A_1	A_0	$\overline{\text{WR}}$	$\overline{\text{RD}}$	完成操作
0	0	0	1	0	读A口数据 → CPU
0	0	1	1	0	读B口数据 → CPU
0	1	0	1	0	读C口数据 → CPU
0	0	0	0	1	CPU数据 → A口数据寄存器
0	0	1	0	1	CPU数据 → B口数据寄存器
0	1	0	0	1	CPU数据 → C口数据寄存器
0	1	1	0	1	CPU送来的命令字 → 控制寄存器

注意： 对控制寄存器不能进行读操作！

10.1 可编程并行I/O接口芯片8255A



二、8255A的工作方式简介

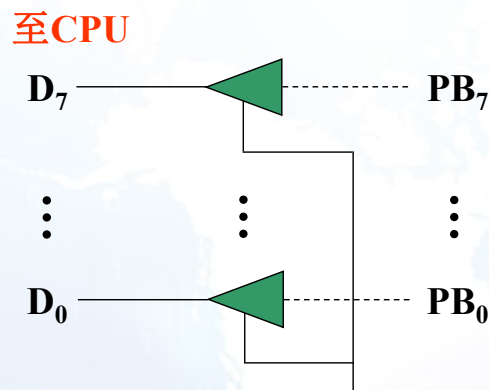
工作方式	适用端口
方式0： 基本型输入/输出	A口、B口、C口
方式1： 选通型输入/输出	A口、B口
方式2： 双向传输	A口

A口可以工作在**方式0**、**方式1**或**方式2**；

B口可以工作在**方式0**或**方式1**；

C口只能工作在**方式0**。

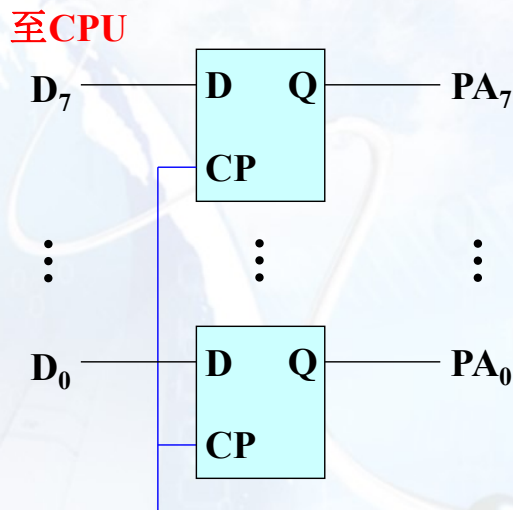
什么是基本型输入？（以B口为例）



当B口定义为基本型输入时，相当于一个输入缓冲器。

对B口执行一条IN指令打开控制门。一条IN指令，就把输入数据 → CPU。

什么是基本型输出？（以A口为例）



当A口定义为基本型输出时，相当于是一个输出锁存器。

CPU对A口执行一条OUT指令把数据锁存在端口。

对A口执行一条OUT指令锁存数据。

总结：基本型输入/输出时，8255A和外设之间没有联络信号！

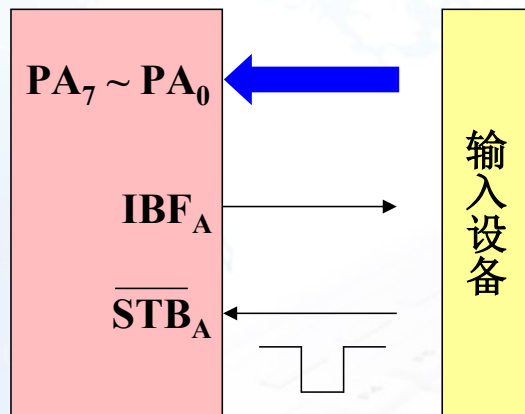
什么是选通型输入？ (以A口为例)

A口定义为选通型输入时，端口和外设之间有两类信号线：

1) 数据线 $PA_7 \sim PA_0$

2) 联络线 IBF_A 、 \overline{STB}_A

8255A A口



IBF (Input Buffer Full)：输入缓冲器满——状态线；

\overline{STB} (Strobe)：输入选通信号。

当外设把数据放在端口线上时必须伴随一个选通信号。

端口收到数据后，通过**IBF**状态告知CPU可读。

IBF=1，通知外设输入缓冲器已满，请不要再送数据；

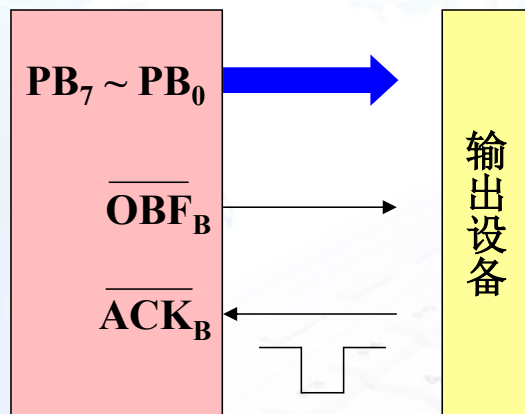
只有在**IBF=0**时，外设才能写入数据。

什么是选通型输出？ (以B口为例)

当端口定义为选通型输出时，端口和外设之间也有两类信号线：

- 1) 数据线 $PB_7 \sim PB_0$
- 2) 联络线 \overline{OBF}_B 、 \overline{ACK}_B

8255A B口



\overline{OBF} (Output Buffer Full)：输出缓冲器满——状态线；

\overline{ACK} (Acknowledge)：确认应答。

CPU对B口执行OUT指令，把一个数 \rightarrow 8255A。

$\overline{OBF}_B = \text{低电平}$ ，通知外设输出缓冲器已满（即数据线上信息可用了）；
外设取走数据之后，使 $\overline{ACK}_B = \text{低电平}$ ，通知CPU端口数据已取走，可再送一个数。

10.1 可编程并行I/O接口芯片8255A



三、8255A控制字与初始化编程

1. 8255控制字

1) 方式选择控制字

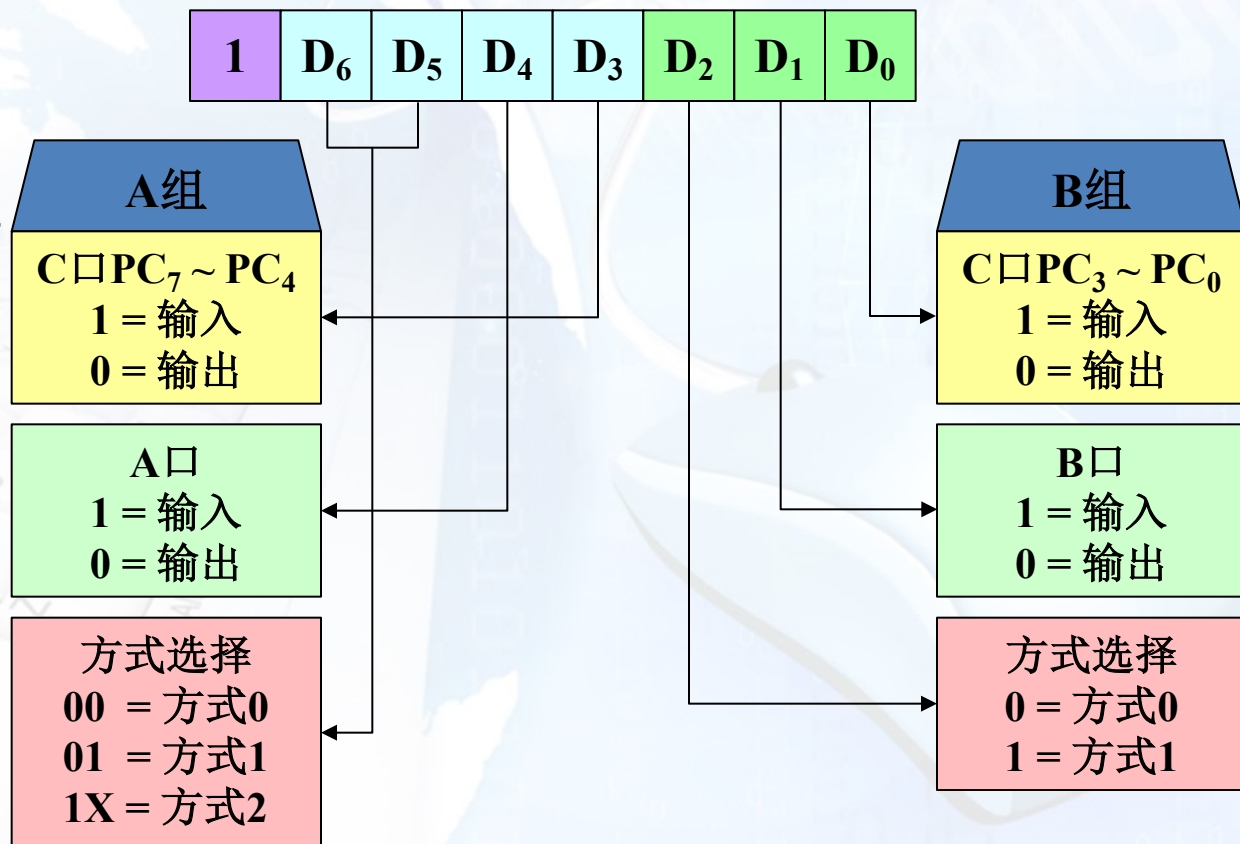
8255有2个控制字，方式选择控制字和C口按位置0/置1命令字，控制字必须写入控制口才能有效。

注意：

A口工作在方式1、方式2，B口工作在方式1时，D₃、D₀只能定义部分PC线的输入/输出。

A口工作在方式2时，D₄不起作用。

C口上、下半部可以选择不同的输入/输出（都是方式0）。



2) C口按位置0/置1控制字



注意： C口按位置0/1控制字
必须写入控制口，写入C口无
效！

2. 8255A的初始化编程

编程步骤:

1) **方式选择控制字** → 控制口

目的: 选择某一口的工作方式;

2) **根据需把C口按位置0/置1控制字** → 控制口

目的: 禁止/允许某一口提出中断请求。

完成初始化编程后, CPU可以用IN指令/OUT指令通过8255A和外设交换信息。

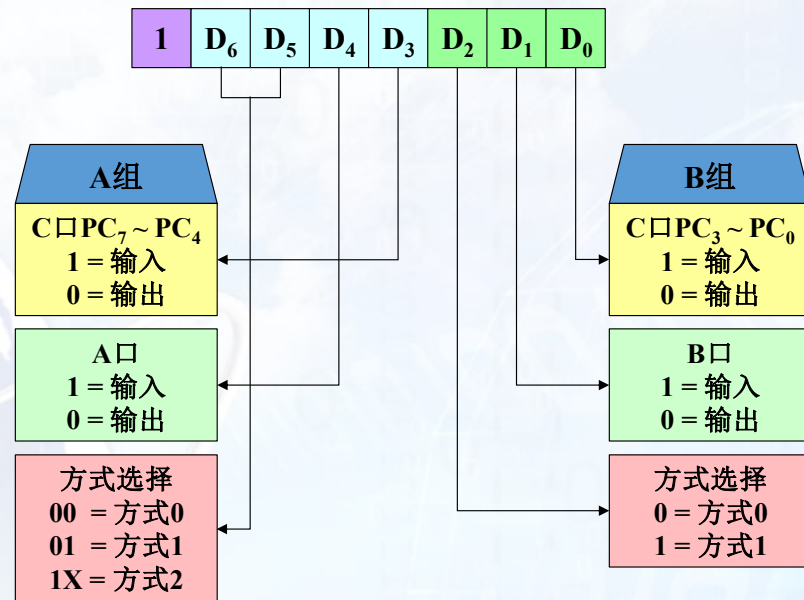
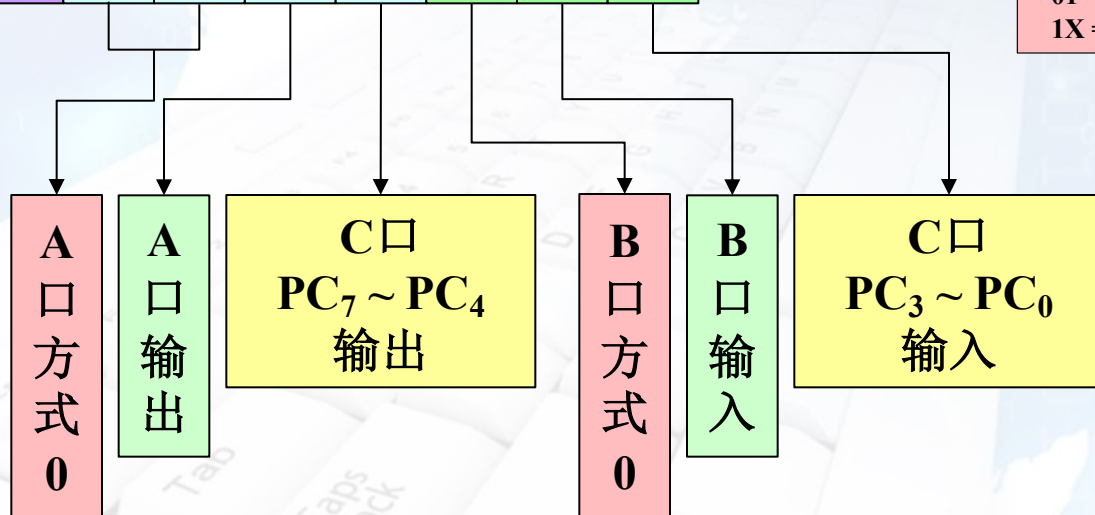
10.1 三、8255A控制字与初始化编程



例：要求置A口为方式0输出，B口为方式0输入，PC₇~PC₄为输出，PC₃~PC₀为输入。

可以写出方式选择控制字：

1 0 0 0 0 0 1 1 = 83H



MOV AL, 83H
OUT 控口地址, AL

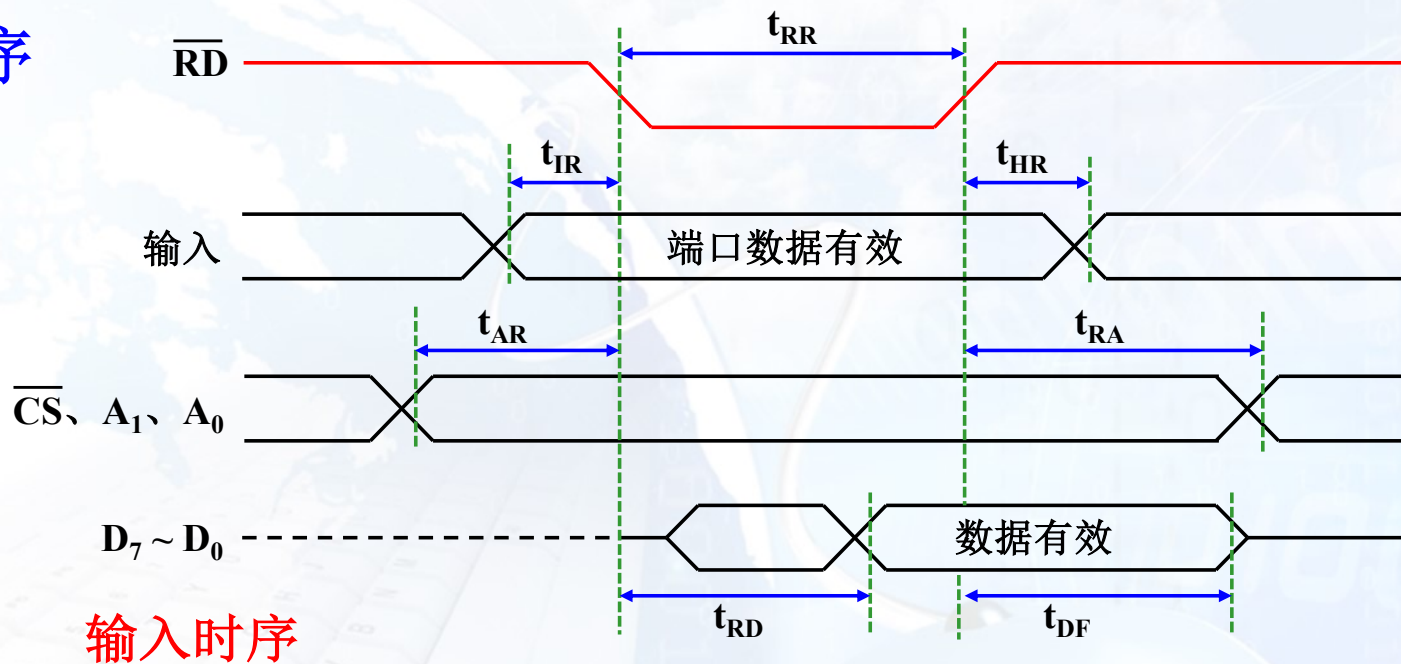
10.1 可编程并行I/O接口芯片8255A



四、8255A工作方式与时序

1. 方式0与时序

8255A工作在
方式0输入时，
CPU在读取数据
之前，端口数据
必须准备好。



CPU对端口执行输入指令， \overline{CS} 、 A_1 、 A_0 有效，8255A被选中，随后 \overline{RD} 信号有效，读取端口数据。

经 t_{RD} 时间延迟，端口数据被送到系统的数据总线上，完成一次输入操作。

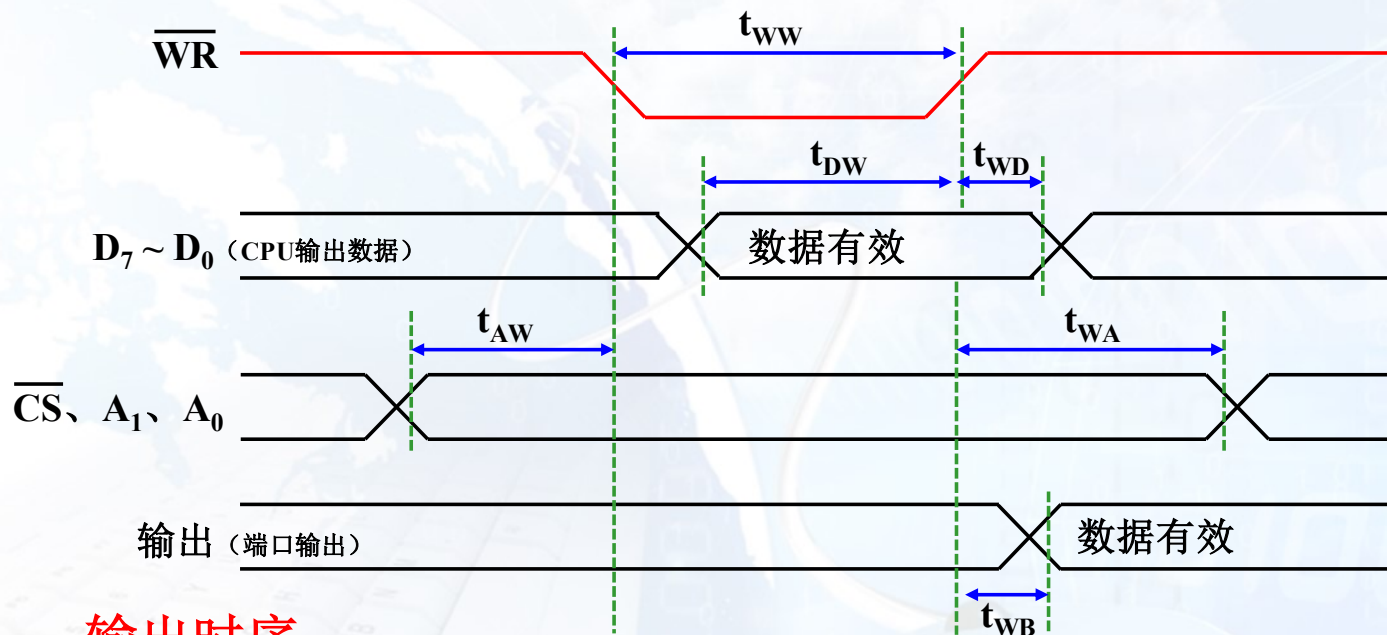
\overline{RD} 有效期间应保持地址信号有效，端口数据应保持到读信号结束后才能消失。

10.1 四、8255A工作方式与时序



CPU将数据
经数据总线，
传送到8255A的
端口数据线上。

在CPU执行
输出指令之前，
端口数据线必
须是空闲的。



输出时序

当CPU对端口执行一条输出指令时， \overline{CS} 、 A_1 、 A_0 有效，待输出的数据在系统数据线上，当 \overline{WR} 信号结束后，最长经过 t_{WB} ，端口数据线上出现有效数据。

CPU写入的数据在整个写过程期间要保持有效，当 \overline{WR} 结束后，还需至少保持 t_{WD} 。

2. 方式1输入（选通型输入）及时序

1) A口方式1输入的预置

方式选择控制字

1 0 1 1 1/0 X X X = B0H

A口
方式1

1 = PC_{7~6}为输入线
0 = PC_{7~6}为输出线

输入

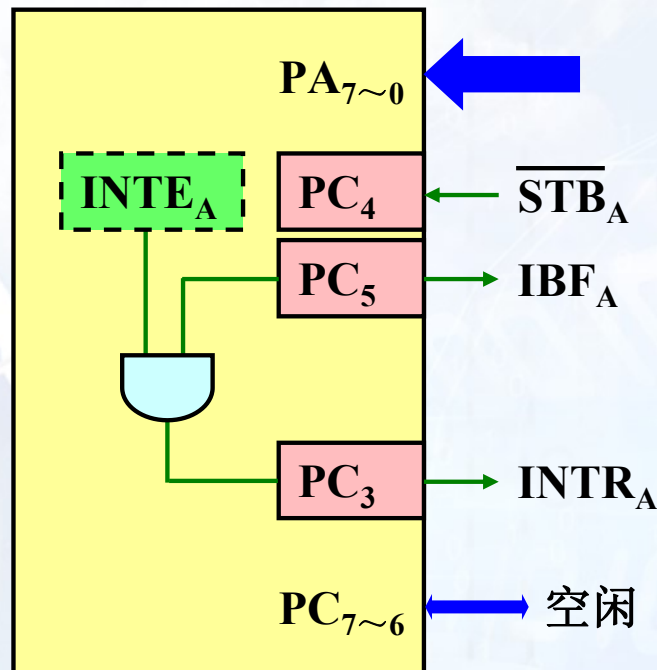
当方式选择控制字**B0H**写入控制口之后，A口即工作在**选通型输入**。

此时：

PA_{7~0}即为输入线。

PC₇、PC₆空闲，输出线 or 输入线？受方式选择控制字D₃控制。

A口选通型输入



PC₄自动定义为输入线，称为 \overline{STB}_A ；

PC₅自动定义为输出线，称为IBF_A；

PC₃自动定义为输出线，称为INTR_A；

它们是输出线 or 输入线？不再受方式选择控制字的D₃、D₀控制！

2) 方式1输入时A口的中断管理

$INTE_A$ 为A口的中断允许触发器，

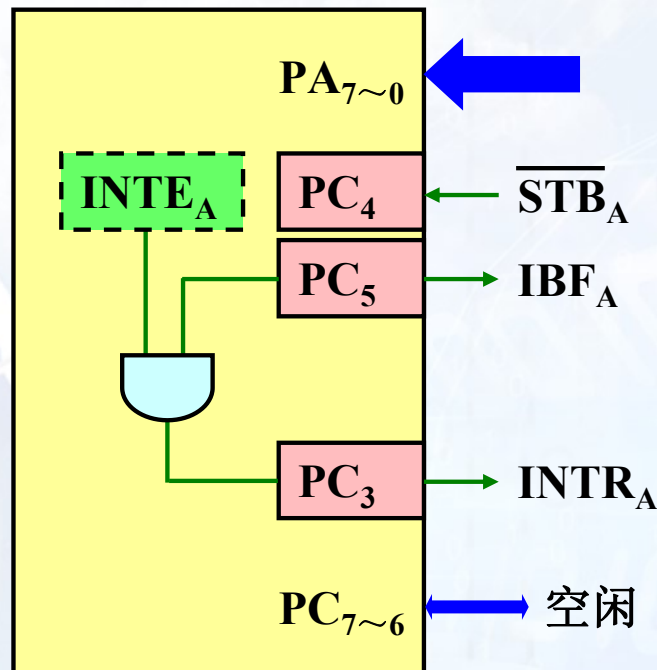
当A口定义为方式1输入时：

- 用C口置0/置1控制字使 $PC_4=1$ ，则 $INTE_A=1$ ，允许A口中断；
- 用C口置0/置1控制字使 $PC_4=0$ ，则 $INTE_A=0$ ，禁止A口中断。

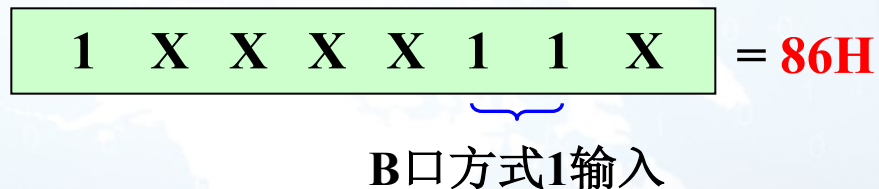
当 $INTE_A=1$ 之后，若 $IBF_A=1$ ，则 $INTR_A=1$ ，A口提出中断请求；

注意：中断允许触发器只能受CPU控制，外设信号 \overline{STB}_A 不能使其置0/1。

A口选通型输入



3) B口方式1输入的预置 方式选择控制字

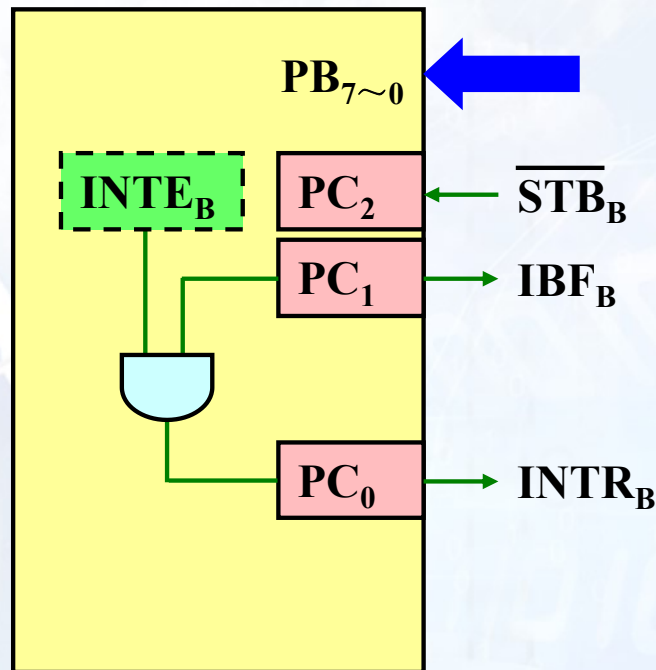


当方式选择控制字**86H**写入控制口之后，B口即工作在**选通型输入**。

此时：

PB₇ ~ PB₀即为输入线。

B口选通型输入



PC₂自动定义为输入线，称为 \overline{STB}_B ；

PC₁自动定义为输出线，称为 IBF_B ；

PC₀自动定义为输出线，称为 $INTR_B$ ；

它们是输出线 or 输入线？不再受方式选择控制字的D₀控制！

4) 方式1输入时B口的中断管理

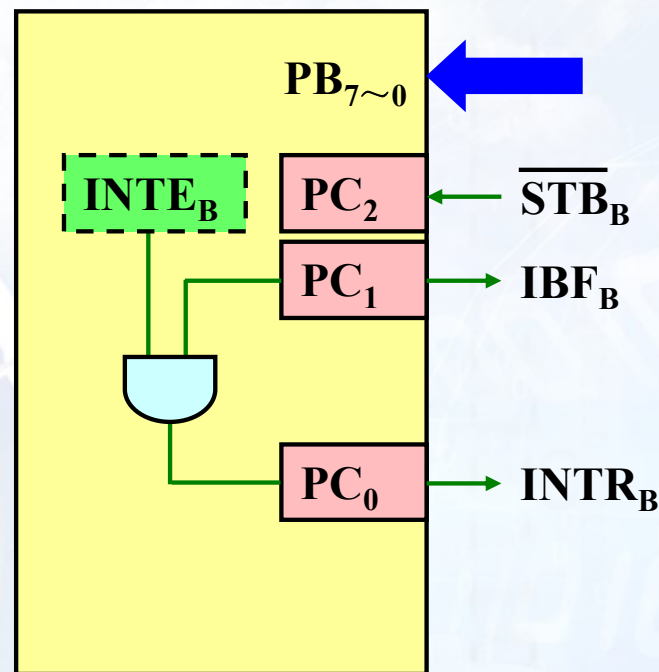
$INTE_B$ 为B口的中断允许触发器，
当B口定义为方式1输入时：

- 用C口置0/置1控制字使 $PC_2=1$ ，
则 $INTE_B=1$ ，允许B口中断；
- 用C口置0/置1控制字使 $PC_2=0$ ，
则 $INTE_B=0$ ，禁止B口中断。

当 $INTE_B=1$ 之后，若 $IBF_B=1$ ，则 $INTR_B=1$ ，B口提出中断请求；

注意：中断允许触发器只能受CPU控制，外设信号 \overline{STB}_B 不能使其置0/1。

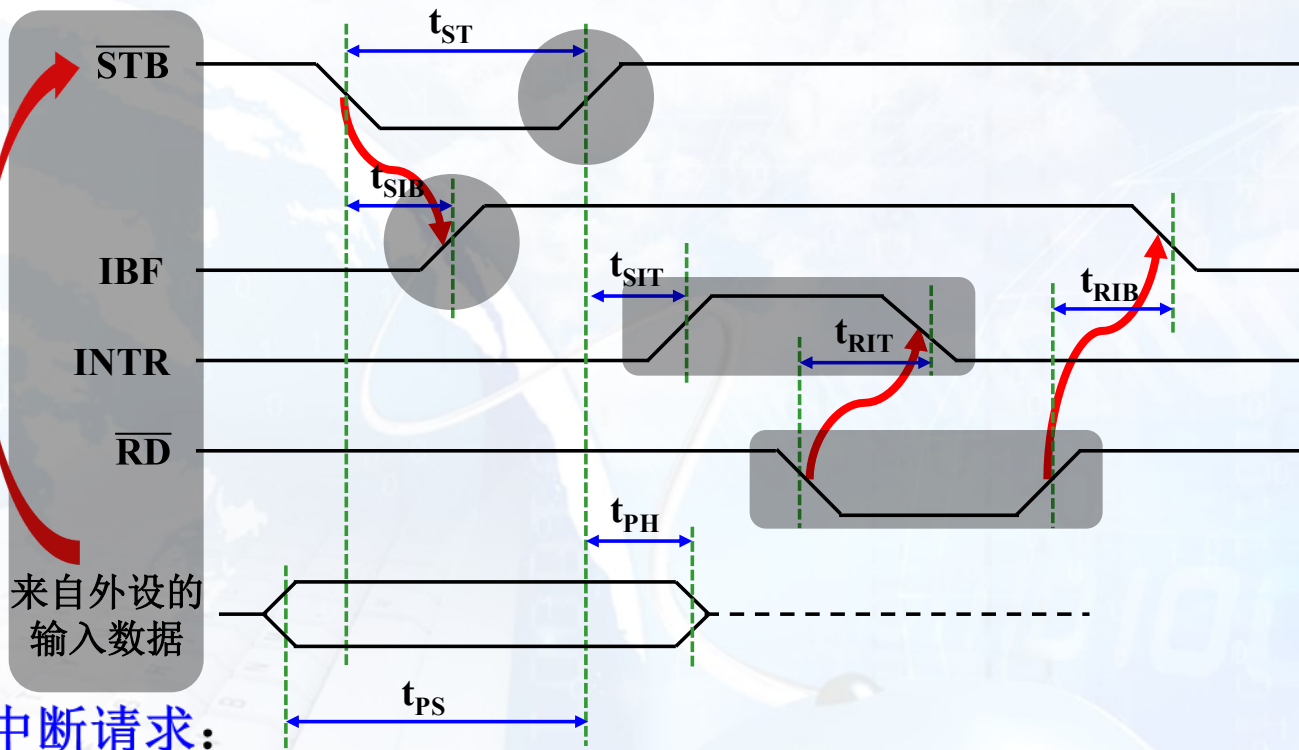
B口选通型输入



5) 方式1输入的时序图

方式1输入操作
是由 $\overline{\text{STB}} = \text{低电平}$
引发的。

$\overline{\text{STB}}$ 有效后，
外设数据 \rightarrow
8255A，
 $\text{IBF} = \text{高电平}$ 。



满足下列条件，提出中断请求：

- ① $\overline{\text{STB}}$ 信号结束；
- ② $\text{IBF}=1$ （外设数据已经锁存）；
- ③ $\text{INTE}=1$ （中断允许触发器置1）。

服务程序执行IN指令后， $\overline{\text{RD}}$ 信号前沿使 $\text{INTR} = \text{低电平}$ ，后沿使 $\text{IBF} = \text{低电平}$ ，从而结束一次数据输入过程。

6) 方式1输入时，CPU与8255A交换信息的方式

- 可以用中断方式;
- 可以用查询方式，**输入查询应查询IBF，不应该查询 \overline{STB} !**

∴ **\overline{STB}** 脉冲是外设送来的，它最小可小到500ns，而在500ns的时间内正好运行CPU的查询指令是不可能的。

3. 方式1输出（选通型输出）及时序

1) A口方式1输出的预置

方式选择控制字

1	0	1	0	1/0	X	X	X
---	---	---	---	-----	---	---	---

 = **A0H**

A口
方式1

1 = PC_{5~4}为输入线
0 = PC_{5~4}为输出线

输出

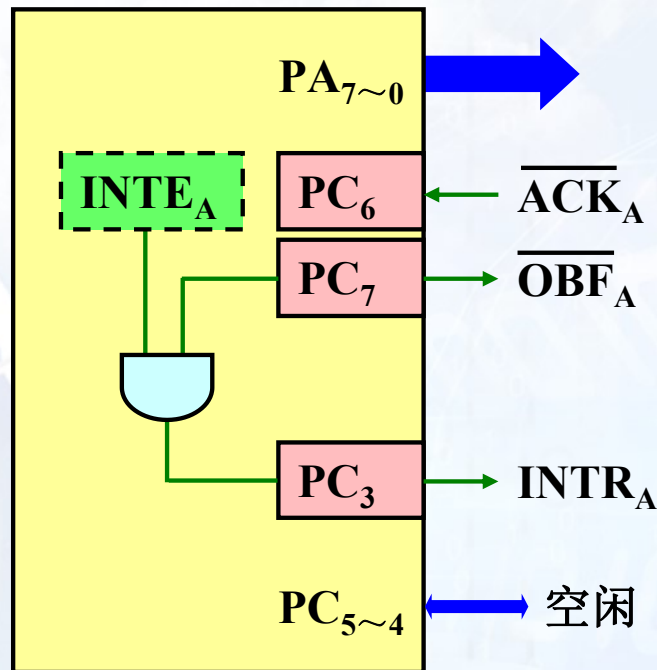
当方式选择控制字**A0H**写入控制口之后，A口即工作在**选通型输出**。

此时：

PA_{7~0}即为输出线。

PC₅、PC₄空闲，输出线 or 输入线？受方式选择控制字D₃控制。

A口选通型输出



PC₆自动定义为输入线，称为 \overline{ACK}_A ；

PC₇自动定义为输出线，称为 \overline{OBF}_A ；

PC₃自动定义为输出线，称为 $INTR_A$ ；

它们是输出线 or 输入线？不再受方式选择控制字的D₃、D₀控制！

2) 方式1输出时A口的中断管理

$INTE_A$ 为A口的中断允许触发器，

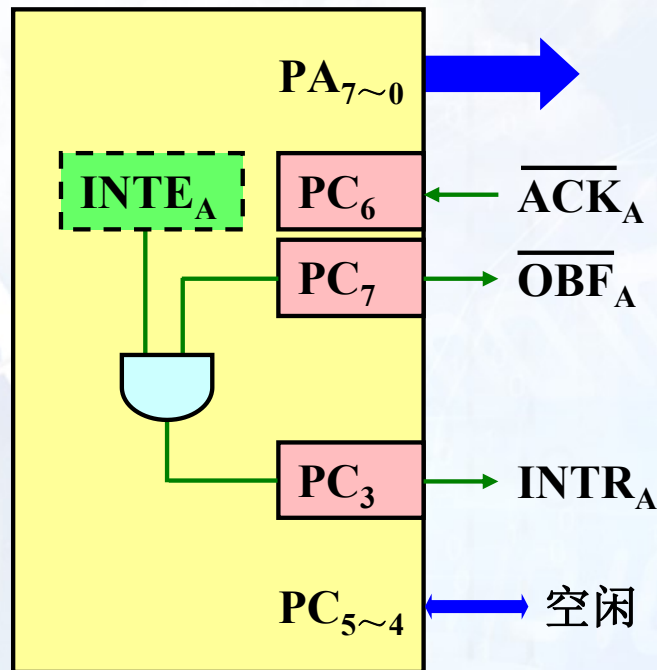
当A口定义为方式1输出时：

- 用C口置0/置1控制字使 $PC_6=1$ ，则 $INTE_A=1$ ，允许A口中断；
- 用C口置0/置1控制字使 $PC_6=0$ ，则 $INTE_A=0$ ，禁止A口中断。

当 $INTE_A=1$ 之后，若 $\overline{OBF}_A=1$ ，则 $INTR_A=1$ ，A口提出中断请求；

注意：中断允许触发器只能受CPU控制，外设信号 \overline{ACK}_A 不能使其置0/1。

A口选通型输出



3) B口方式1输出的预置 方式选择控制字

1	X	X	X	X	1	0	X
---	---	---	---	---	---	---	---

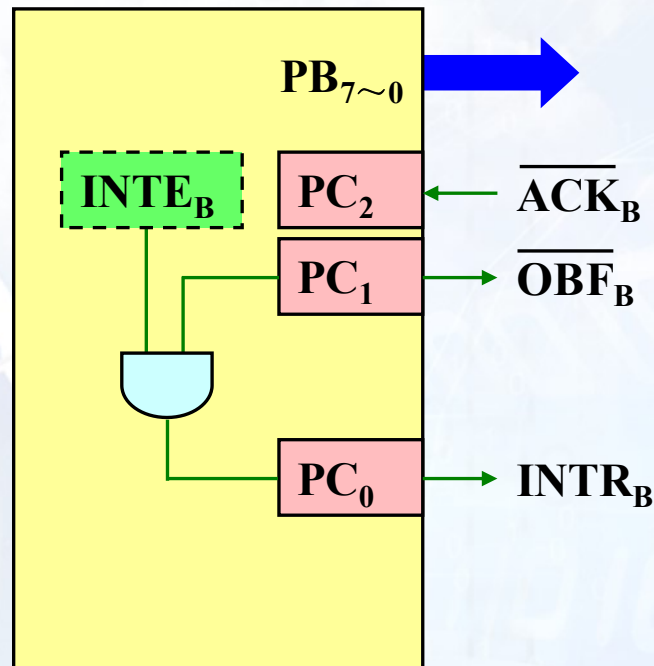
$\underbrace{\hspace{10em}}_{\text{B口方式1输出}} = 84\text{H}$

当方式选择控制字**84H**写入控制口之后，B口即工作在**选通型输出**。

此时：

$\text{PB}_7 \sim \text{PB}_0$ 即为输出线。

B口选通型输出



PC_2 自动定义为输入线，称为 $\overline{\text{ACK}}_B$ ；

PC_1 自动定义为输出线，称为 $\overline{\text{OBF}}_B$ ；

PC_0 自动定义为输出线，称为 INTR_B ；

它们是输出线 or 输入线？不再受方式选择控制字的 D_0 控制！

4) 方式1输出时B口的中断管理

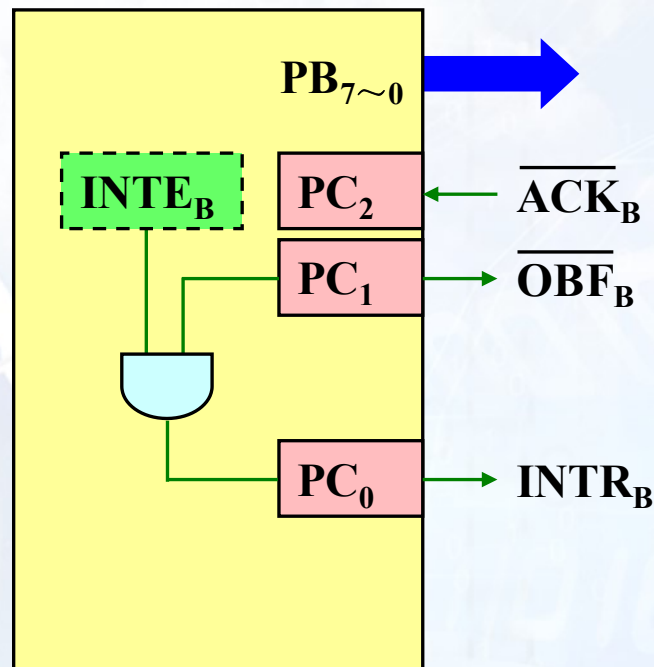
$INTE_B$ 为B口的中断允许触发器，
当B口定义为方式1输出时：

- 用C口置0/置1控制字使 $PC_2=1$ ，
则 $INTE_B=1$ ，允许B口中断；
- 用C口置0/置1控制字使 $PC_2=0$ ，
则 $INTE_B=0$ ，禁止B口中断。

当 $INTE_B=1$ 之后，若 $\overline{OBF}_B=1$ ，则 $INTR_B=1$ ，B口提出中断请求；

注意：中断允许触发器只能受CPU控制，外设信号 \overline{ACK}_B 不能使其置0/1。

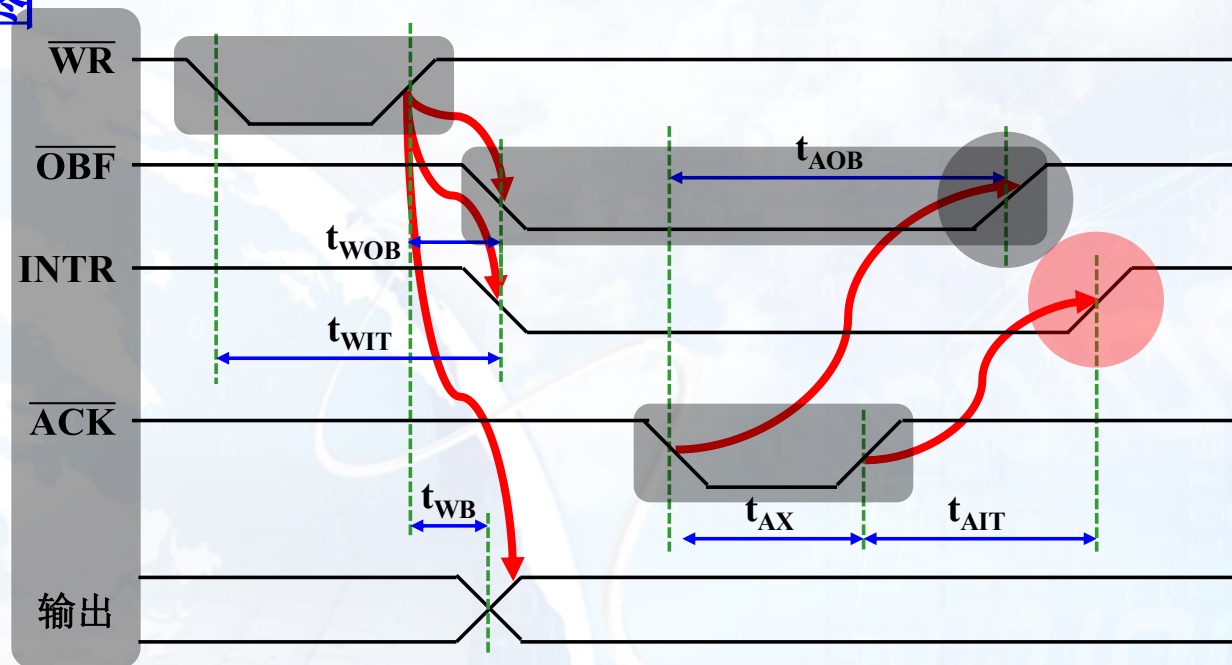
B口选通型输出



5) 方式1输出的时序图

- 方式1输出时，CPU针对8255A数据口执行OUT指令， $\overline{\text{WR}}$ 信号有效。

- $\overline{\text{WR}}$ 信号结束后，数据出现在外设数据线上：



① $\overline{\text{OBF}}$ 有效（向外设通报其数据线信息可用）；

② 使INTR无效，撤销中断请求。

- 外设取走数据后，发来ACK，ACK信号前沿使 $\overline{\text{OBF}}$ 无效，后沿提出中断。

- 工作在方式1输出时端口提出中断请求的条件：

① 外设发出ACK信号且使 $\overline{\text{OBF}}=1$ ；

② 中断允许触发器置1。

6) 方式1输出时，CPU与8255A交换信息的方式

- 可以用中断方式;
- 可以用查询方式，**输出查询应查询 $\overline{\text{OBF}}$ ，不应该查询 $\overline{\text{ACK}}$!**

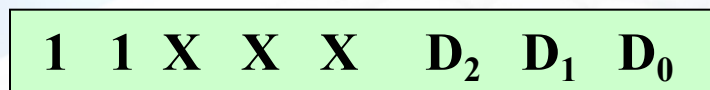
当 $\overline{\text{OBF}} = 1$ （不满）时可送下一个数据。

4. 方式2：双向传输

只有A口可以工作在双向方式！

当A口工作在双向方式时，B口可以工作在基本型I/O（不需要联络线），也可以工作在选通型I/O（联络线PC_{2~0}）。

1) 双向方式的设置



A口双向

00 B口方式0输出

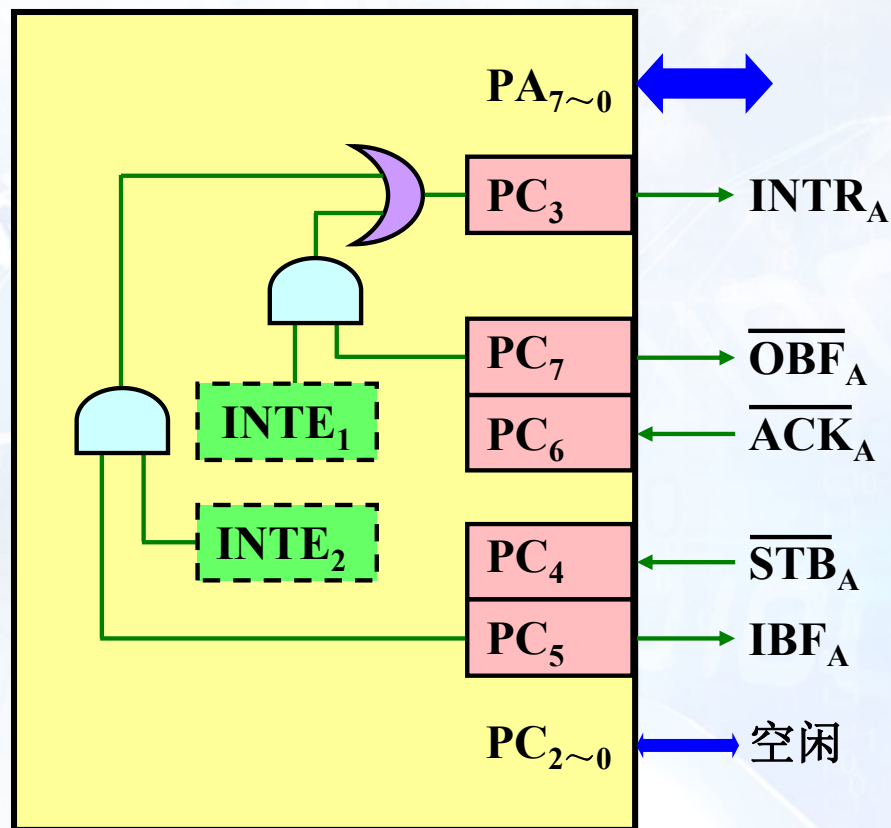
01 B口方式0输入

10 B口方式1输出

11 B口方式1输入

1 = PC_{2~0}为输入线

0 = PC_{2~0}为输出线



10.1 四、8255A工作方式与时序



A口定义为双向方式之后：

$PA_{7\sim0}$ 是双向的数据线，可以分时完成数据输入/输出。

PC_3 自动定义为输出线： $INTR_A$

PC_7 自动定义为输出线： \overline{OBF}_A

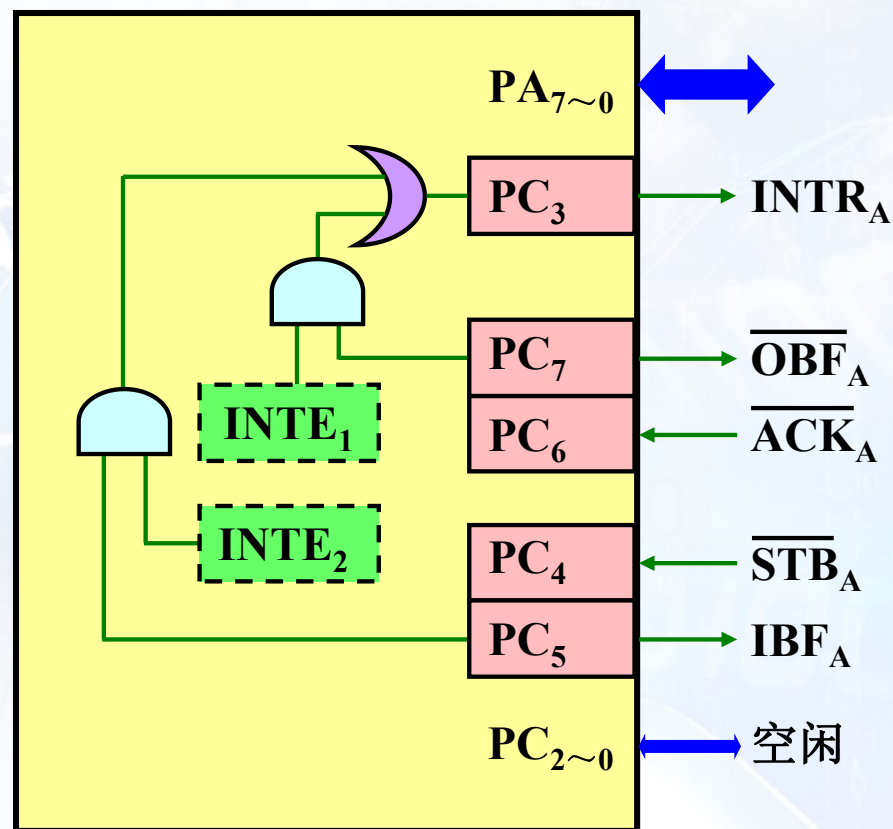
PC_6 自动定义为输入线： \overline{ACK}_A

PC_4 自动定义为输入线： \overline{STB}_A

PC_5 自动定义为输出线： IBF_A

它们是输出线 or 输入线？不再受方式字 D_3 、 D_0 控制！

$PC_{2\sim0}$ 空闲。



\overline{STB}_A 、 IBF_A 是双向方式中的一对输入联络线；

\overline{OBF}_A 、 \overline{ACK}_A 是双向方式中的一对输出联络线。

数据输入过程、数据输出过程与方式1类似。

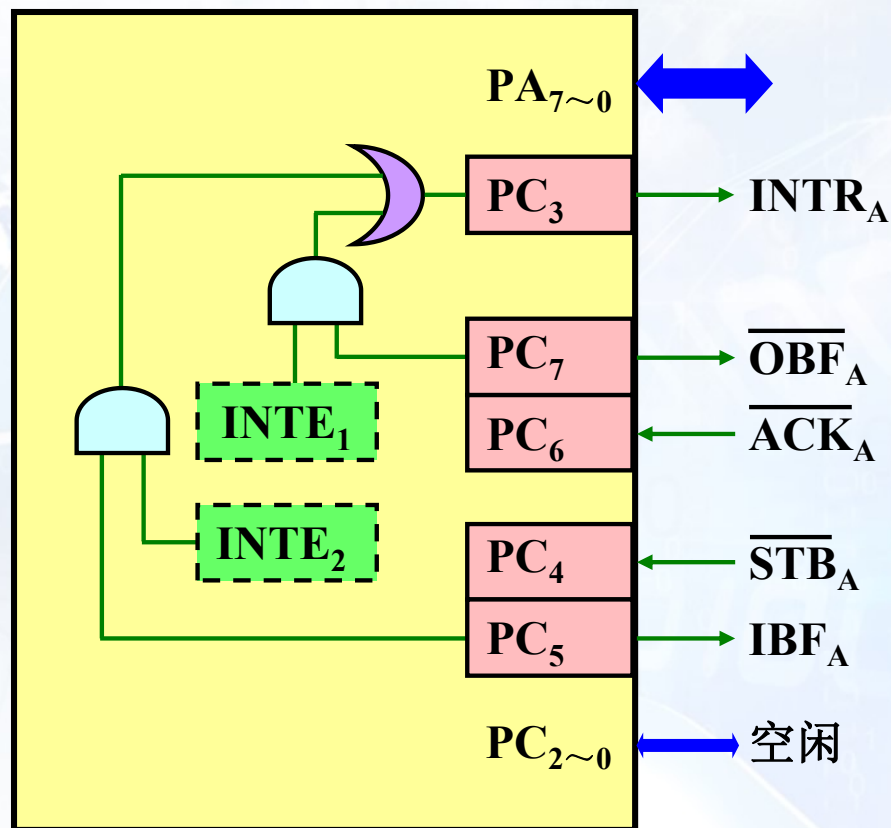
2) A口双向方式的中断管理

**$INTE_1$ 为双向方式中的输出
中断允许触发器。**

用C口置0/1控制字使 PC_6
置0, 则 $INTE_1=0$;

用C口置0/1控制字使 PC_6
置1, 则 $INTE_1=1$ 。

$INTE_1$ 置1后, 若 $\overline{OBF}_A=1$, 则 $INTR_A=1$, A口提出中断请求。



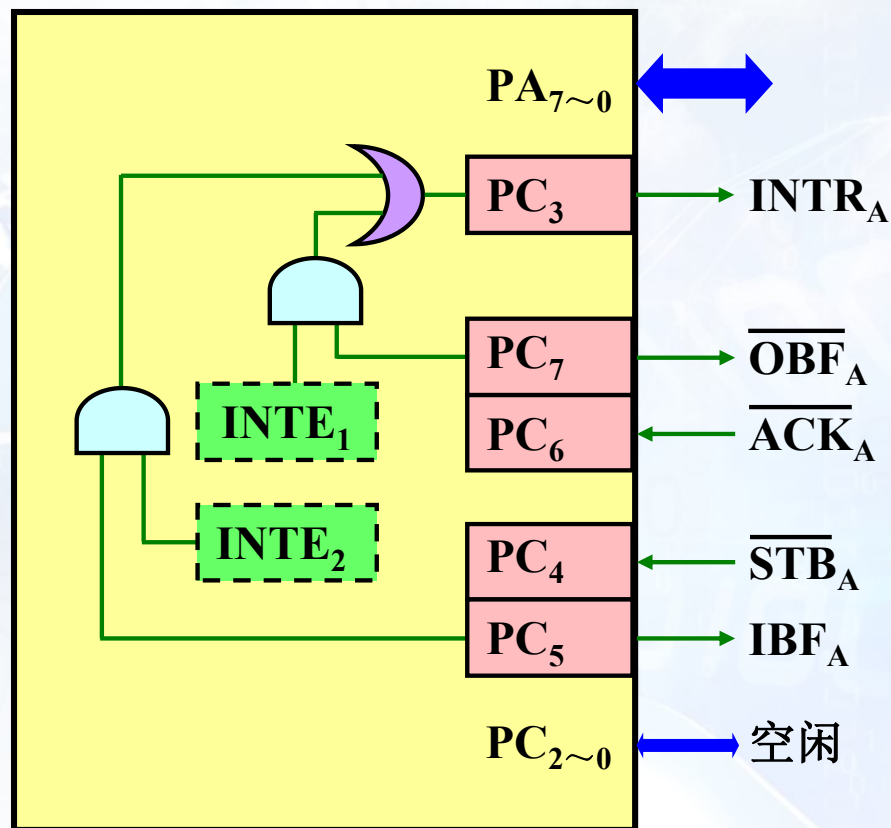
2) A口双向方式的中断管理

INTE₂为双向方式中的输入
中断允许触发器。

用C口置0/1控制字使PC₄
置0，则INTE₂=0；

用C口置0/1控制字使PC₄
置1，则INTE₂=1。

INTE₂置1后，若IBF_A=1，则INTR_A=1，A口提出中断请求。



5. 8255A小结

1) 8255A工作方式

A口可以工作在方式0、1、2，B口可以工作在方式0、1，不能工作在方式2，C口可以工作在方式0；

方式0的特点：**8255A和外设交换信息时，不需要联络线；**

方式1、2的特点：**8255A和外设交换信息时，需要联络线；**

C口分为PC_{7~4}、PC_{3~0}两部分，但程序员对C口的读写是按字节操作的；

A口工作在方式2时，B口只能工作在方式0、1（限制：CPU只能用查询方式而不能用中断方式和B口交换数据）。

2) CPU与8255A交换信息的方式

数据口工作在方式0时，CPU采用无条件传送；

数据口工作在方式1、2时，CPU用：

a. 查询方式，输入查询测试**IBF**，输出查询测试 **\overline{OBF}** ；

b. 中断方式，由于8255A本身不能向CPU回送中断向量，所以**必须用8259A管理**。在系统外扩8255A时，8255A的**INTR**应接到系统机从8259A的**IR₁**，作为用户中断来处理。

3) 中断允许触发器的置0、置1

对程序员来讲：工作在方式1、方式2时，8255A内部有4个中断允许触发器，用“C口专用的置0/1控制字”可以使它们置0/1。

中断允许触发器 \ 控制位 \ 方式	A口方式1输入	A口方式1输出	B口方式1输入	B口方式1输出	A口双向
$INTE_A$	PC ₄	PC ₆			
$INTE_B$			PC ₂	PC ₂	
$INTE_1$					PC ₆
$INTE_2$					PC ₄

如：A口工作在方式1输入时，

执行 **MOV AL, 00001001B**

OUT 8255A控制口, AL ; 即可使 $INTE_A$ 置1, 允许中断

4) 8255A的初始化编程

工作在方式0时:

方式选择控制字 → 控制口;

工作在方式1、方式2时:

a. 方式选择控制字 → 控制口;

b. 根据CPU和8255A之间是用查询方式还是中断方式交换信息, 组织C口置0/置1控制字 → 控制口, 使相应的中断允许触发器置0/置1, 从而达到禁止或开放中断的目的。

10.2 8255A的应用举例

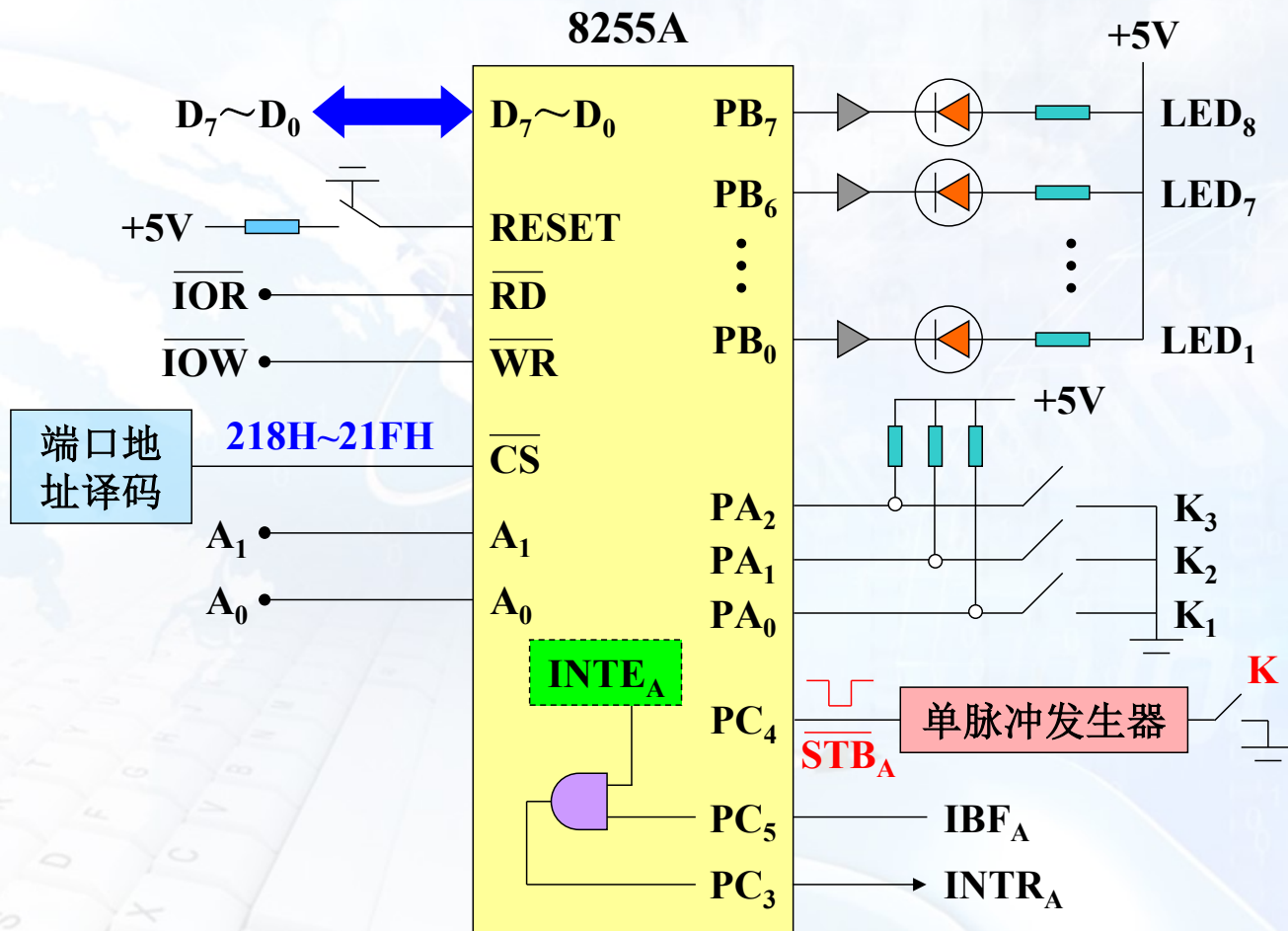


例1:

设系统机外扩了一片8255A，相应的实验电路如图所示。

要求：先预置开关 $K_3 \sim K_1$ 为一组状态，然后按下自复按钮 K 产生一个负脉冲信号输入到 PC_4 。

用发光二极管 LED_i 亮来显示 $K_3 \sim K_1$ 的状态。主机键盘有任意键按下时结束演示。



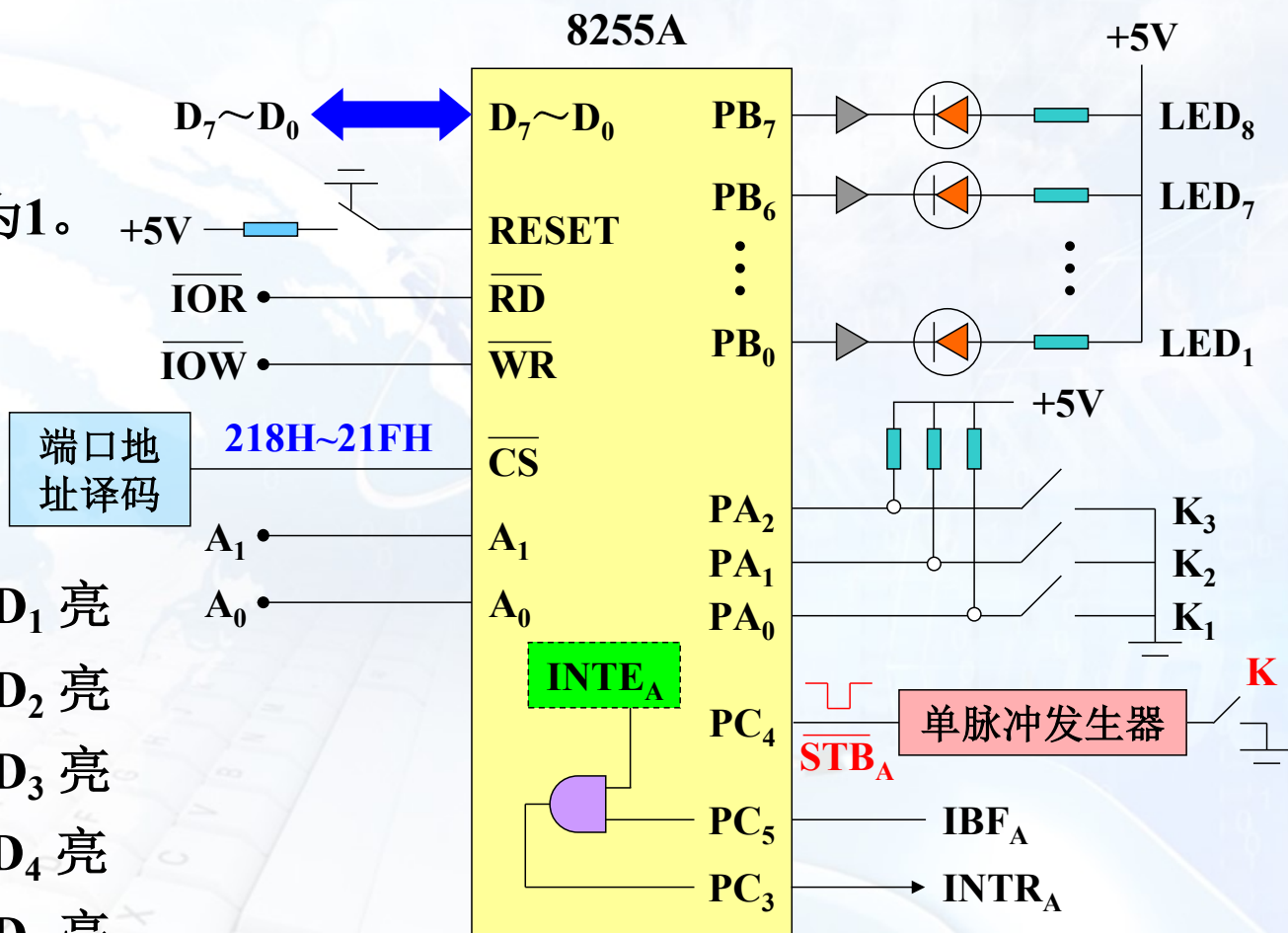
10.2 8255A的应用举例



要求:

$K_3 \sim K_1$ 闭合为0, 断开为1。

$K_3 K_2 K_1 = 000$ 时, LED₁ 亮
 $K_3 K_2 K_1 = 001$ 时, LED₂ 亮
 $K_3 K_2 K_1 = 010$ 时, LED₃ 亮
 $K_3 K_2 K_1 = 011$ 时, LED₄ 亮
 $K_3 K_2 K_1 = 100$ 时, LED₅ 亮
 $K_3 K_2 K_1 = 101$ 时, LED₆ 亮
 $K_3 K_2 K_1 = 110$ 时, LED₇ 亮
 $K_3 K_2 K_1 = 111$ 时, LED₈ 亮



10.2 8255A的应用举例



设计思路:

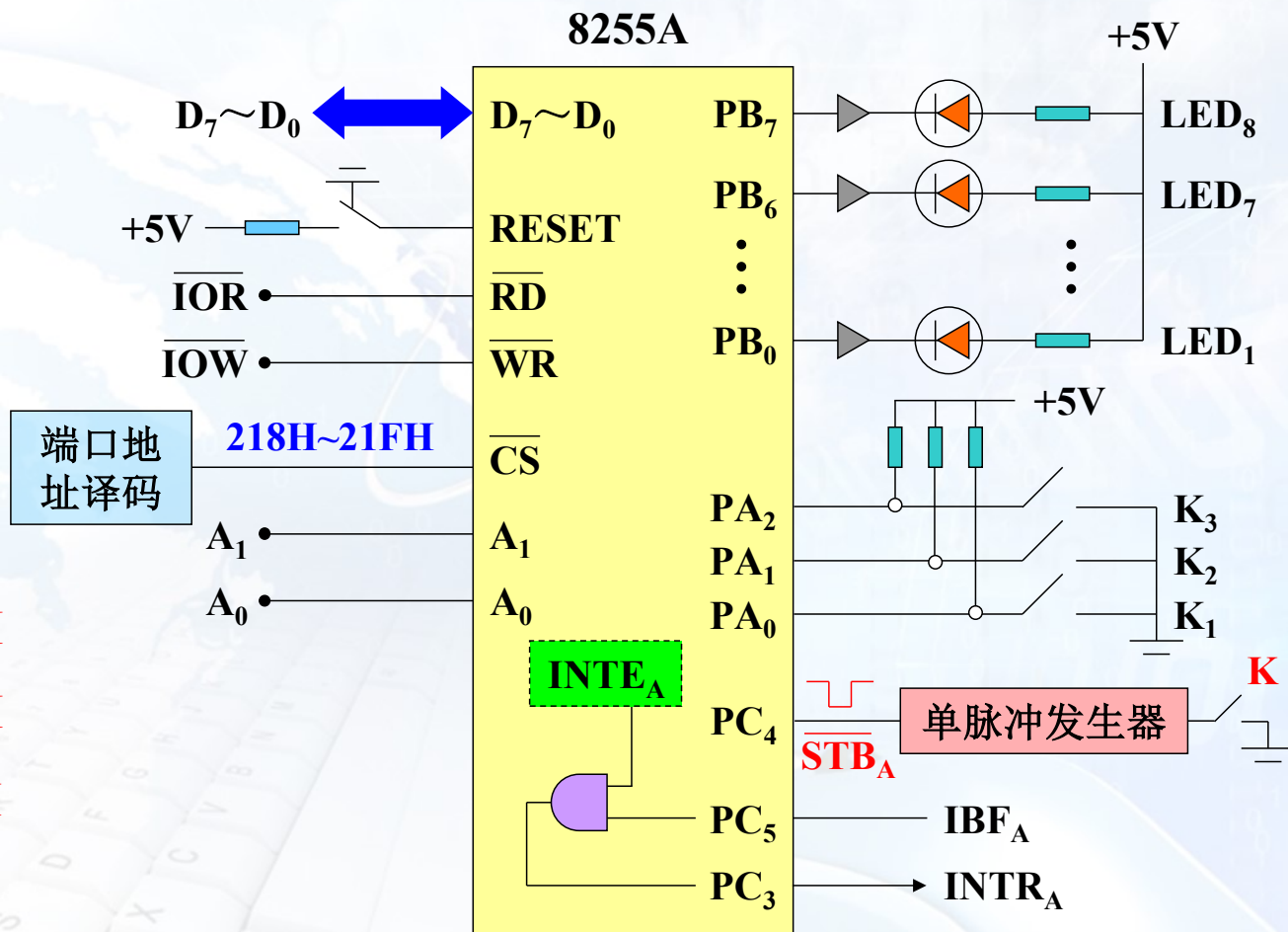
1) 端口地址

A口: **218H、21CH**

B口: **219H、21DH**

C口: **21AH、21EH**

控制口: **21BH、21FH**



A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
1	0	0	0	0	1	1	X	A ₁	A ₀

10.2 8255A的应用举例

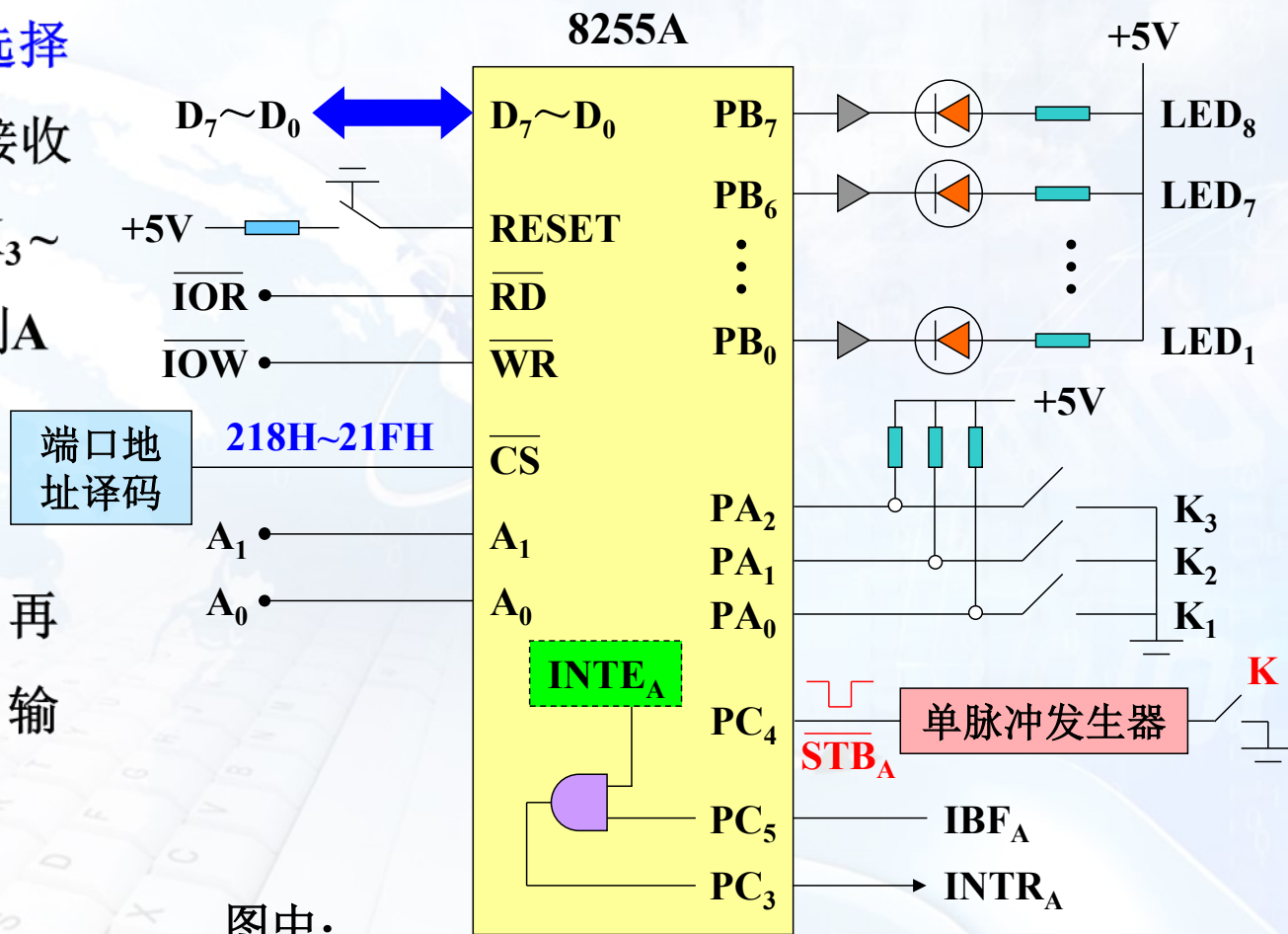


2) 8255A工作方式的选择

当PC₄（即 \overline{STB}_A ）接收到负脉冲信号之后，K₃~K₁的状态信息被锁存到A口数据寄存器PA₂~PA₀中。

CPU读取A口信息，再根据PA₂~PA₀的状态，输出相应数据到B口，使LED_i亮。

因此，**B口应为基本型输出**，**A口应为选通型输入**。



图中：

PB_i接LED_{i+1}；

PA₂~PA₀接开关K₃~K₁；

PC₄接自复按钮K。

10.2 8255A的应用举例



3) 8255A与CPU交换信息的方式

CPU → 8255A B口:

无条件输出

8255A A口 → CPU:

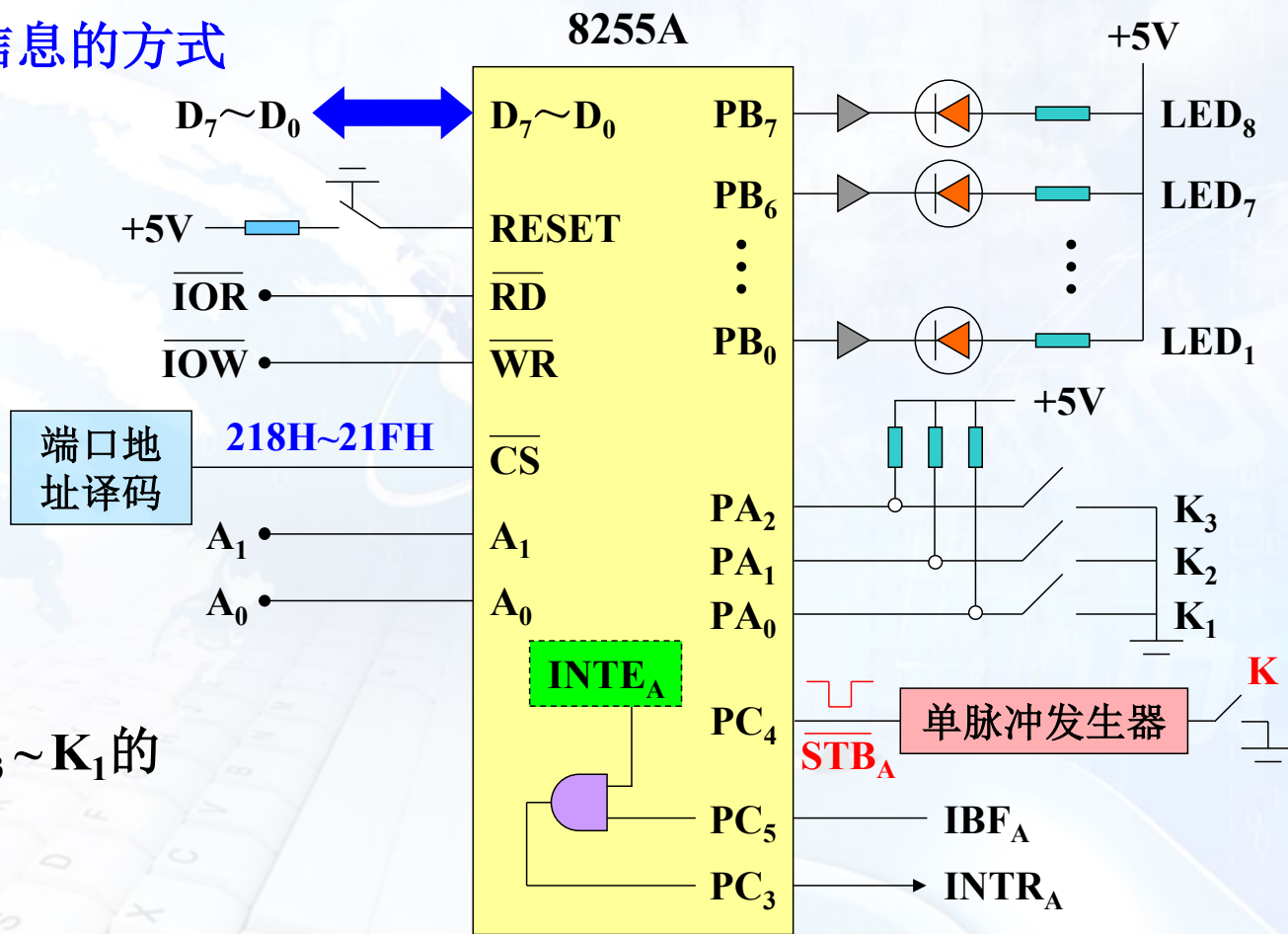
解法①查询方式

当PC₅=1时, 表示K₃~K₁的

值已经写入A口了;

解法②中断方式

PC₃接系统从8259A的IR₁,
按用户中断处理。



A口工作在选通型输入时:

PC₅成为输入线IBF_A

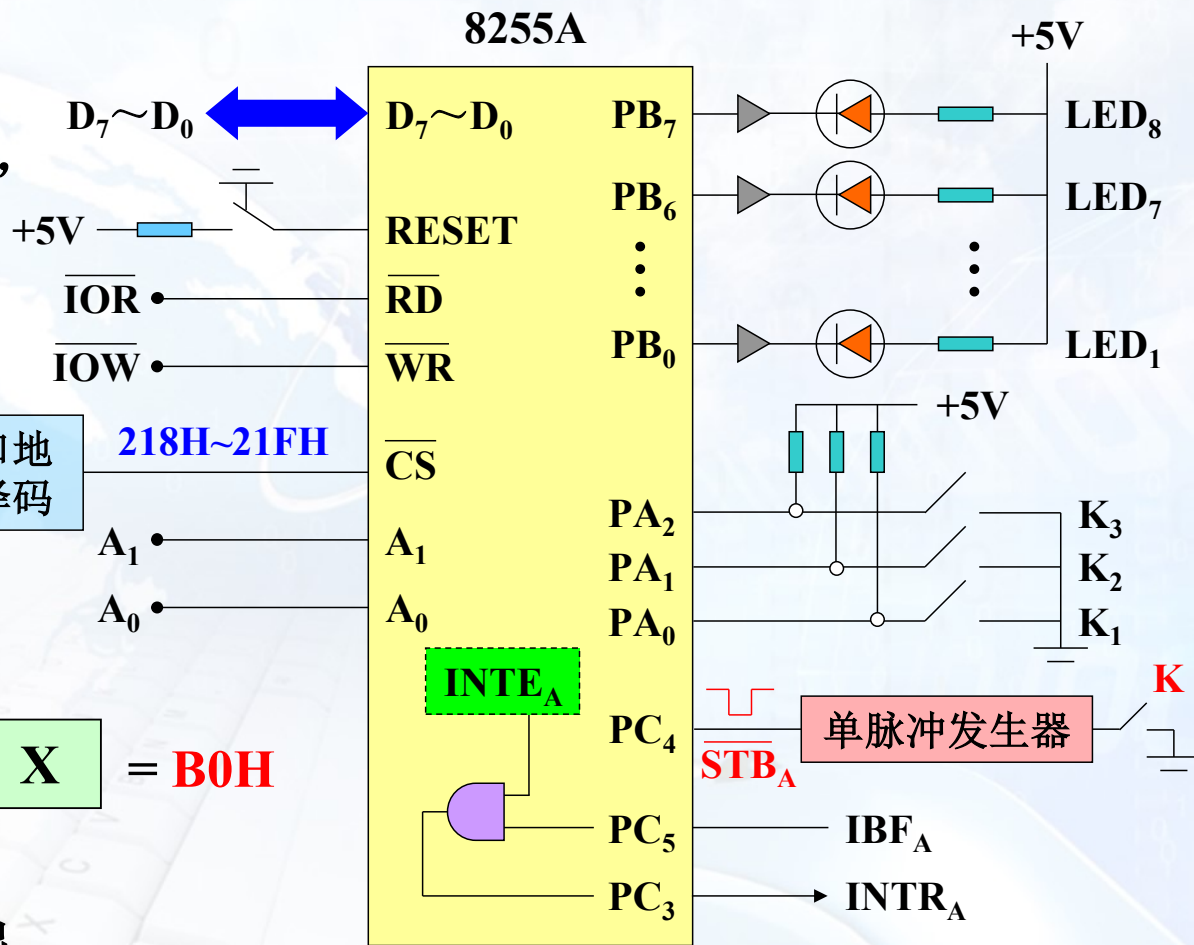
PC₃成为输出线INTR_A

10.2 8255A的应用举例



解法① 查询方式

B口基本型输出（方式0），
A口选通型输入（方式1）。



方式选择控制字:

1 0 1 1 X 0 0 X = B0H

A口
方式1

B口
方式0输出

A口输入

A口禁止中断控制字:

0 X X X 1 0 0 0 = 08H

10.2 8255A的应用举例



[查询方式程序清单]

.486

DATA SEGMENT

MESG DB '8255A READY...', 0DH, 0AH, '\$'

TAB DB 11111110B

DB 11111101B

DB 11111011B

DB 11110111B

DB 11101111B

DB 11011111B

DB 10111111B

XLAT TAB

DB 01111111B

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

BEG: MOV AX, DATA

MOV DS, AX

10.2 8255A的应用举例



CODE SEGMENT

ASSUME CS:CODE, DS:DATA

BEG: MOV AX, DATA

MOV DS, AX

CALL I8255A



;8255A初始化

MOV AH, 9

MOV DX, OFFSET MSG

INT 21H

;给出操作提示

SCAN: MOV AH, 1

INT 16H

;有键入 ?

JNZ LAST

;有, 则程序结束

MOV DX, 21AH

; 21EH

IN AL, DX

;读8255A C口

TEST AL, 00100000B

; PC₅=1 ?

JZ SCAN

; NO

用IN指令去读8255A的C口,测试PC₅这一位是否为1,而PC₅就是IBF信号。

10.2 8255A的应用举例



```
MOV DX, 218H ; 21CH
```

```
IN AL, DX ;读8255A A口
```

```
AND AL, 07H
```

```
MOV BX, OFFSET TAB
```

```
XLAT TAB ;查表
```

```
MOV DX, 219H ; 21DH
```

```
OUT DX, AL ;表项输出到B口
```

```
JMP SCAN
```

```
LAST: MOV AH, 4CH
```

```
INT 21H
```

XLAT指令执行时，BX定义为指向TAB表头的指针，AL的值是相对于表头的偏移量。

$K_3 \sim K_1 = 101$

TAB	11111110	← BX
	11111101	
	11111011	↑ [BX+AL]
	11110111	
	11101111	
	11011111	
	10111111	←
	01111111	

执行OUT指令，将被选中的表项11011111B输出到B口，点亮LED₆。

中断
方式

10.2 8255A的应用举例



I8255A PROC

```
MOV    DX,    21BH    ; 21FH
MOV    AL,    0B0H
OUT    DX,    AL      ;写入工作方式字
MOV    AL,    08H
OUT    DX,    AL      ;PC4=0 (INTEA=0)
MOV    DX,    219H    ; 21DH, B口
MOV    AL,    0FFH
OUT    DX,    AL      ;熄灭LED
RET
```

I8255A ENDP

CODE ENDS

END BEG

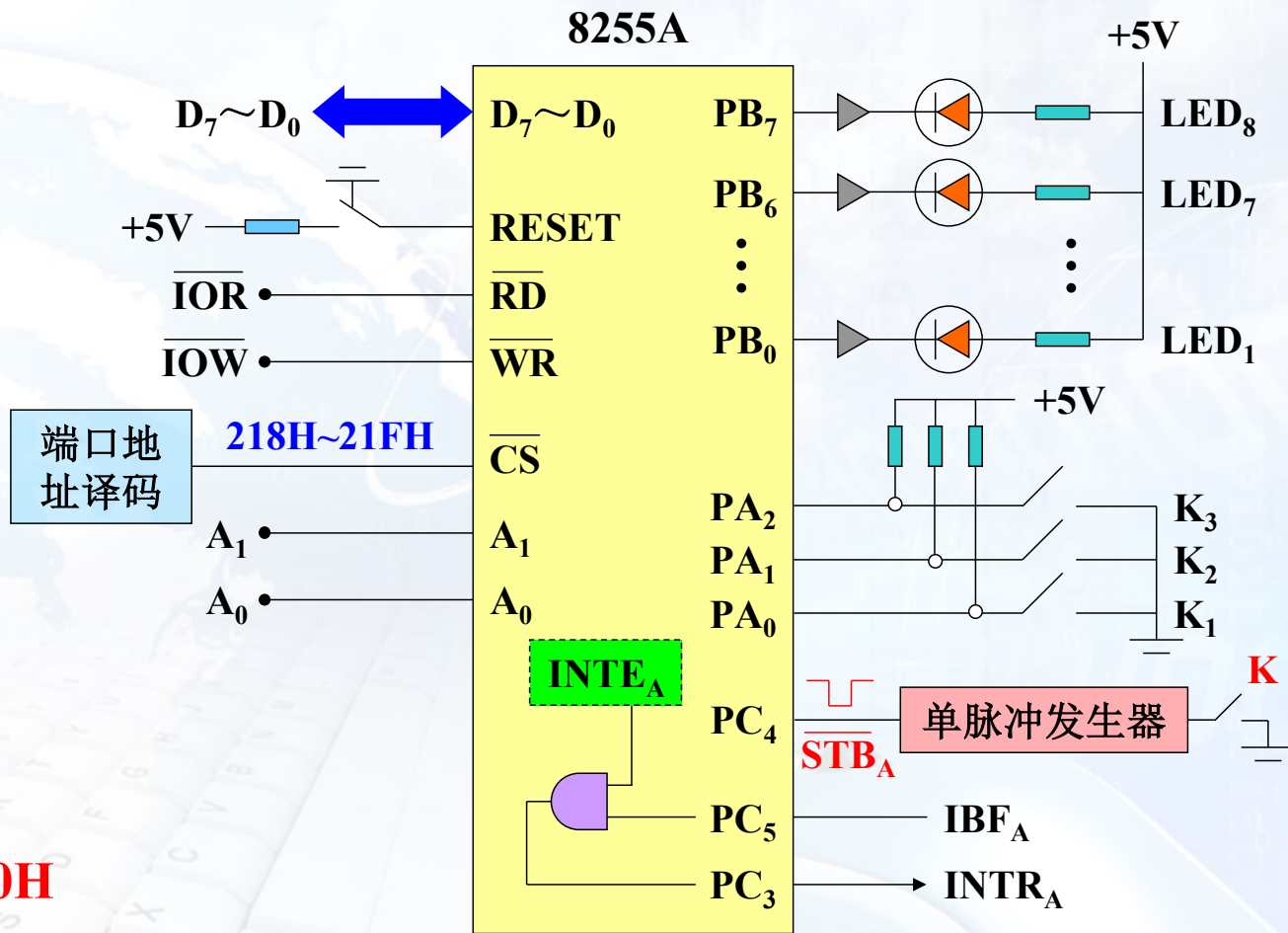
解法② 中断方式

B口方式0输出， A口方式1输入， 中断方式。

方式选择控制字 = B0H

A口允许中断控制字:

0 X X X 1 0 0 1 = **09H**



10.2 8255A的应用举例



[中断方式程序清单]

.486

DATA SEGMENT

MESG DB '8255A READY...', 0DH, 0AH, '\$'

TAB DB 11111110B

DB 11111101B

DB 11111011B

DB 11110111B

DB 11101111B

DB 11011111B

DB 10111111B

DB 01111111B

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DATA

BEG: MOV AX, DATA

MOV DS, AX

10.2 8255A的应用举例



```
CLI
CALL I8255A
CALL WRITE0A
CALL I8259
MOV AH, 9
MOV DX, OFFSET MSG
INT 21H
STI
SCAN: MOV AH, 1
INT 16H
JZ SCAN
...
SERVICE PROC
...
MOV AL, 20H
OUT 20H, AL
POP DS
POP AX
IRET
SERVICE ENDP
```



; 8255A初始化
; 置换0AH型中断向量
; 开放用户中断

; 给出操作提示
; 开中断

; 有键入 ?
; 无转
; 有返回DOS

; 为中断服务
; 中断结束命令
; →主8259

; 中断返回

10.2 8255A的应用举例



I8255A

PROC

MOV DX, 21BH

; 21FH

MOV AL, 0B0H

OUT DX, AL

;写入方式字

MOV AL, 09H

OUT DX, AL

;PC₄=1, (INTE_A=1)

MOV DX, 219H

; 21DH, B口

MOV AL, 0FFH

OUT DX, AL

;熄灭LED

RET



I8255A

ENDP

WRITE0A

PROC

PUSH DS

MOV AX, CODE

MOV DS, AX

MOV DX, OFFSET SERVICE

MOV AX, 250AH

10.2 8255A的应用举例



```
INT    21H
POP    DS
RET
WRITE0A ENDP
I8259  PROC
    IN    AL,    21H
    AND   AL,    11111011B
    OUT   21H,   AL
    IN    AL,    0A1H
    AND   AL,    11111101B
    OUT   0A1H,  AL
    RET
I8259  ENDP
CODE   ENDS
END    BEG
```

;开放用户中断

10.2 8255A的应用举例



例2：8255A方式1与打印机接口

1. 打印机接口信号

数据线： $D_7 \sim D_0$ 。

控制信号：（PC机 → 打印机）

$\overline{\text{STROBE}}$ ： 选通信号；

状态信号：（打印机 → PC机）

BUSY： 忙标志信号

$\overline{\text{ACKNLG}}$ ： 打印机的应答信号；

- 在**BUSY=0**（空闲）时，CPU执行OUT指令，输出一个数据；
- CPU使接口电路发出选通脉冲： **$\overline{\text{STROBE}}=0$** ，即有效。
- 打印机收到 **$\overline{\text{STROBE}}=0$** 之后，立即使**BUSY=1**以示打印机忙；
- 打印机把数据存入内部的数据缓冲器之后，向PC机发出应答信号 **$\overline{\text{ACKNLG}}$** 。
 $\overline{\text{ACKNLG}}$ 有效后，过5us再使BUSY=0，BUSY=0后撤销 $\overline{\text{ACKNLG}}$ 。

10.2 8255A的应用举例



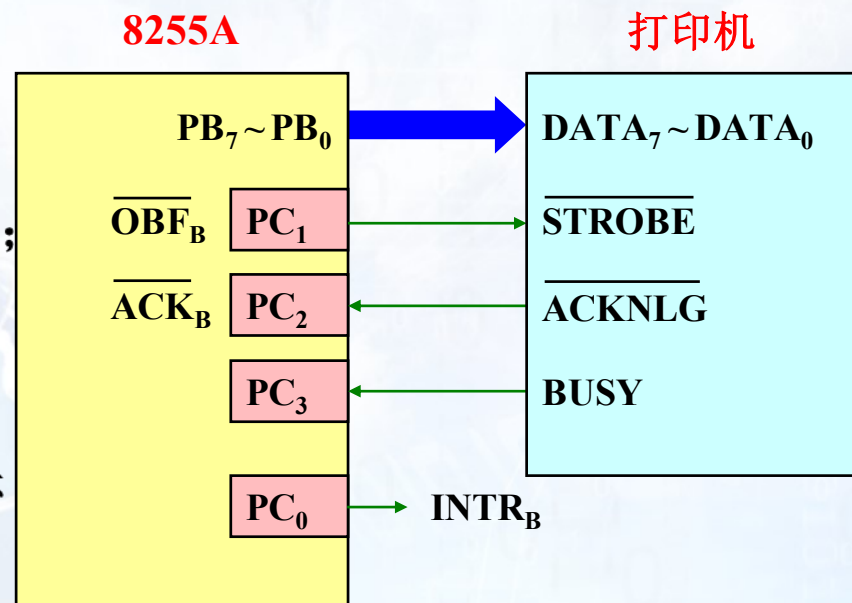
例：采用8255A的端口B工作于选通输出方式，与打印机接口。

此时， PC_1 自动为B口的 \overline{OBF}_B 输出信号；

PC_2 为B口 \overline{ACK}_B 输入信号；

PC_0 为B口INTR输出信号。

另外，通过 PC_2 控制 $INTE_B$ ，决定是否采用中断方式。



当CPU输出数据时，8255A产生低有效 \overline{OBF} 输出信号，它需要一个 \overline{ACK} 响应信号恢复为高；另一方面，打印机需要一个低脉冲 \overline{STROBE} 才能接收数据，并反馈一个 \overline{ACKNLG} 响应信号表示数据接收完毕。

因此在8255A和打印机接口进行连线时，将8255A的 PC_1 与打印机中的 \overline{STROBE} 引脚相连、打印机中的 \overline{ACKNLG} 引脚与8255A中的 PC_2 相连， $BUSY$ 引脚与 PC_3 相连，如图所示。

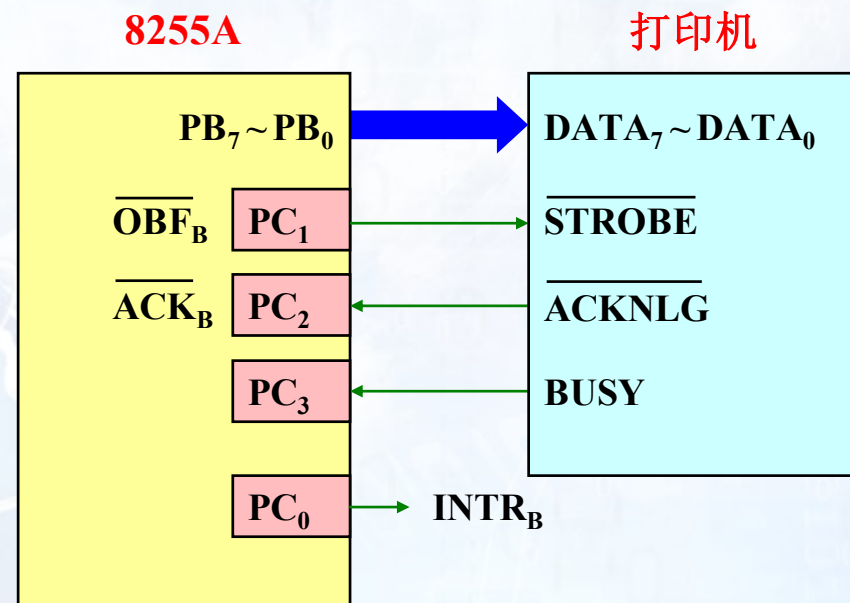
假设8255A的端口地址为80H~83H。

10.2 8255A的应用举例



分析：端口B工作在方式1输出，因此不需要查询打印机的工作状态BUSY信号。

通过读取C口的内容就可以查询打印机数据口的状态，从而了解打印机是否已经接收完数据。



这是为什么呢？

原因就隐藏在打印机工作时序图中，我们已经学习过：

因为BUSY信号的产生，取决于STROBE信号的后沿，而BUSY信号的撤销，取决于应答信号的前沿，完全不需要额外的操作。

对于这个题目，
使用查询方式即可。

10.2 8255A的应用举例



方式选择控制字

1 X X X X 1 0 1 = **85H**

B口方式1输出

PC₃输入

C口置0/置1控制字——B口禁止中断

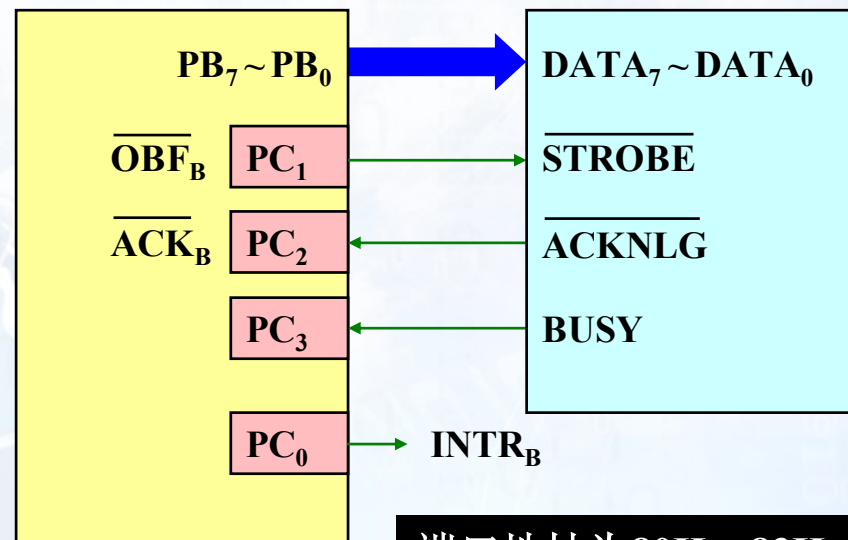
0 X X X 0 1 0 0 = **04H**

.586

```
DATA SEGMENT USE16
    BUF DB 'HELLO' , 0AH, 0DH
LENS EQU $-BUF
DATA ENDS
CODE SEGMENT USE16
    ASSUME CS:CODE, DS:DATA
    BEG: MOV AX, DATA
        MOV DS, AX
        CALL I8255A ;8255A初始化
```

8255A

打印机



端口地址为80H~83H

```
I8255A PROC
    MOV AL, 85H
    OUT 83H, AL
    MOV AL, 04H
    OUT 83H, AL
    RET
I8255A ENDP
```


10.2 8255A的应用举例



```
LEA    BX, BUF
MOV    CX, LENS

NEXT:  MOV    AL, [BX]
      INC    BX
      OUT    81H, AL
SCAN:  IN     AL, 82H
      TEST   AL, 02H
      JZ     SCAN

      LOOP   NEXT
      MOV    AH, 4CH
      INT    21H

CODE   ENDS
      END    BEG
```

00000010B

若 $\overline{\text{OBF}}$ 信号等于0，说明B口还没有收到打印机应答信号，即上一次的字符打印工作尚未完成。

;通过 PC_1 ，查询 $\overline{\text{OBF}}$

; $\overline{\text{OBF}}=0$ ，不能向B口再写数据

; $\overline{\text{OBF}}=1$ ，可向B口继续写数据

8255A的A口工作在方式1查询输入，CPU在读取端口数据之前应先查询引脚()。

- ☐ A PC2
- ☐ B PC3
- ☐ C PC4
- ☒ D PC5

提交

8255A的B口设置在选通型输出方式，则8255A与外设之间的联络信号为ACK和[填空1]，若CPU采用中断方式输出数据，B口允许中断受PC [填空2] 引脚控制。

正常使用填空题需3.0以上版本雨课堂