微机原理与接口技术

第十章 并行接口

第10章 并行接口

- 教学重点

- 8255A的工作方式和编程
- 8255A的应用
- 简易键盘的扫描程序
- LED数码管的多位显示

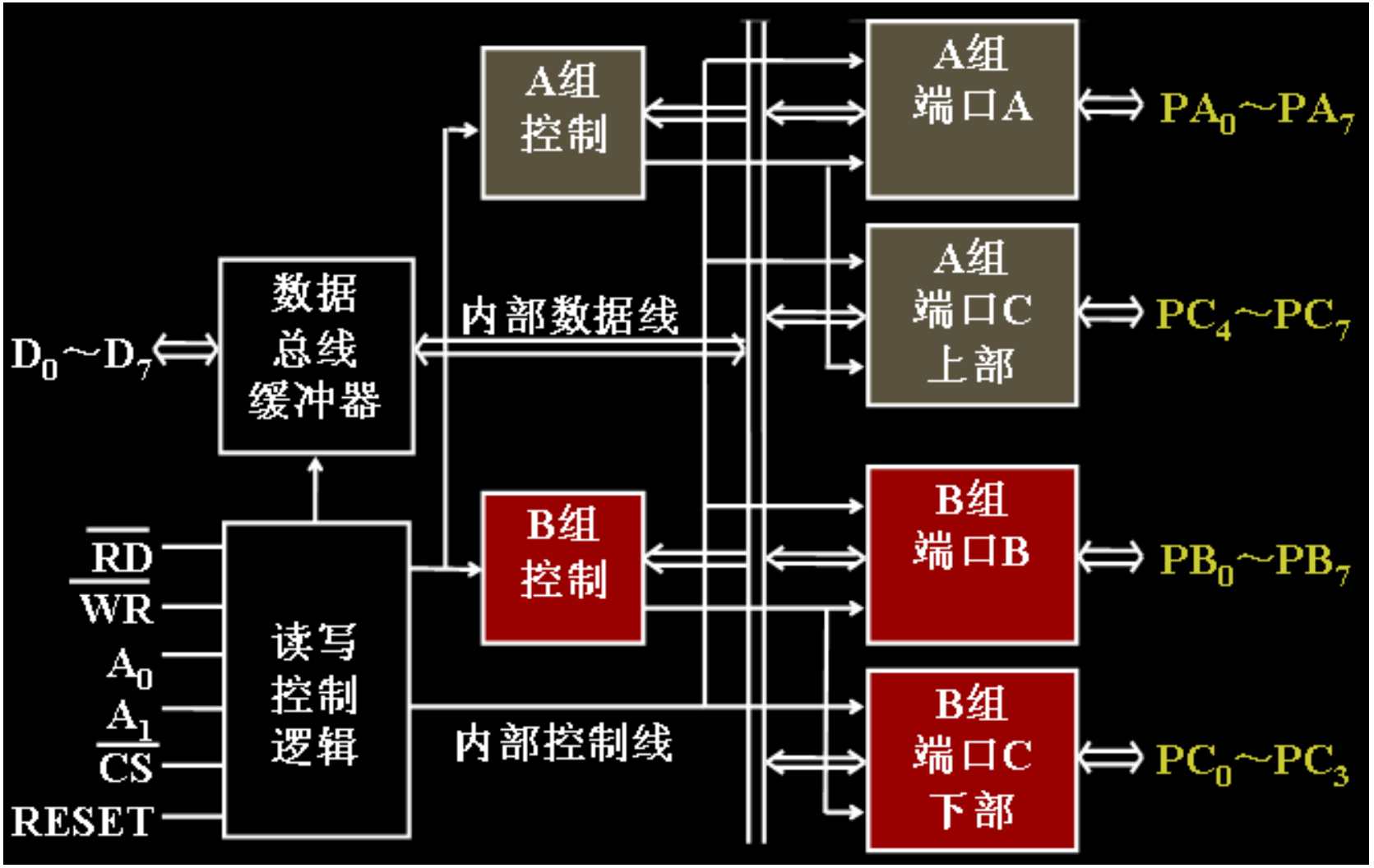
并行数据传输方式

- 以计算机的字长，通常是8位、16位或32位为传输单位，一次传送一个字长的数据
- 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
  - 例如：微机与并行接口打印机、磁盘驱动器
- 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

10.1 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间、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
- 共三种输入输出工作方式

10.1.1 8255A的内部结构和引脚



1. 外设数据端口

- 端口A： PA0～PA7
  - A组，支持工作方式0、1、2
- 端口B： PB0～PB7
  - B组，支持工作方式0、1
- 端口C： PC0～PC7
  - 仅支持工作方式0
  - A组控制高4位PC4～PC7
  - B组控制低4位PC0～PC3

1. 外设数据端口

- 端口A：PA0～PA7
  - 常作数据端口，功能最强大
- 端口B：PB0～PB7
  - 常作数据端口
- 端口C：PC0～PC7
  - 可作数据、状态和控制端口
  - 分两个4位，每位可独立操作
  - 控制最灵活，最难掌握

2. 与处理器接口

- D_0 ~ D_7数据线
- RD*读信号
- CS*片选信号

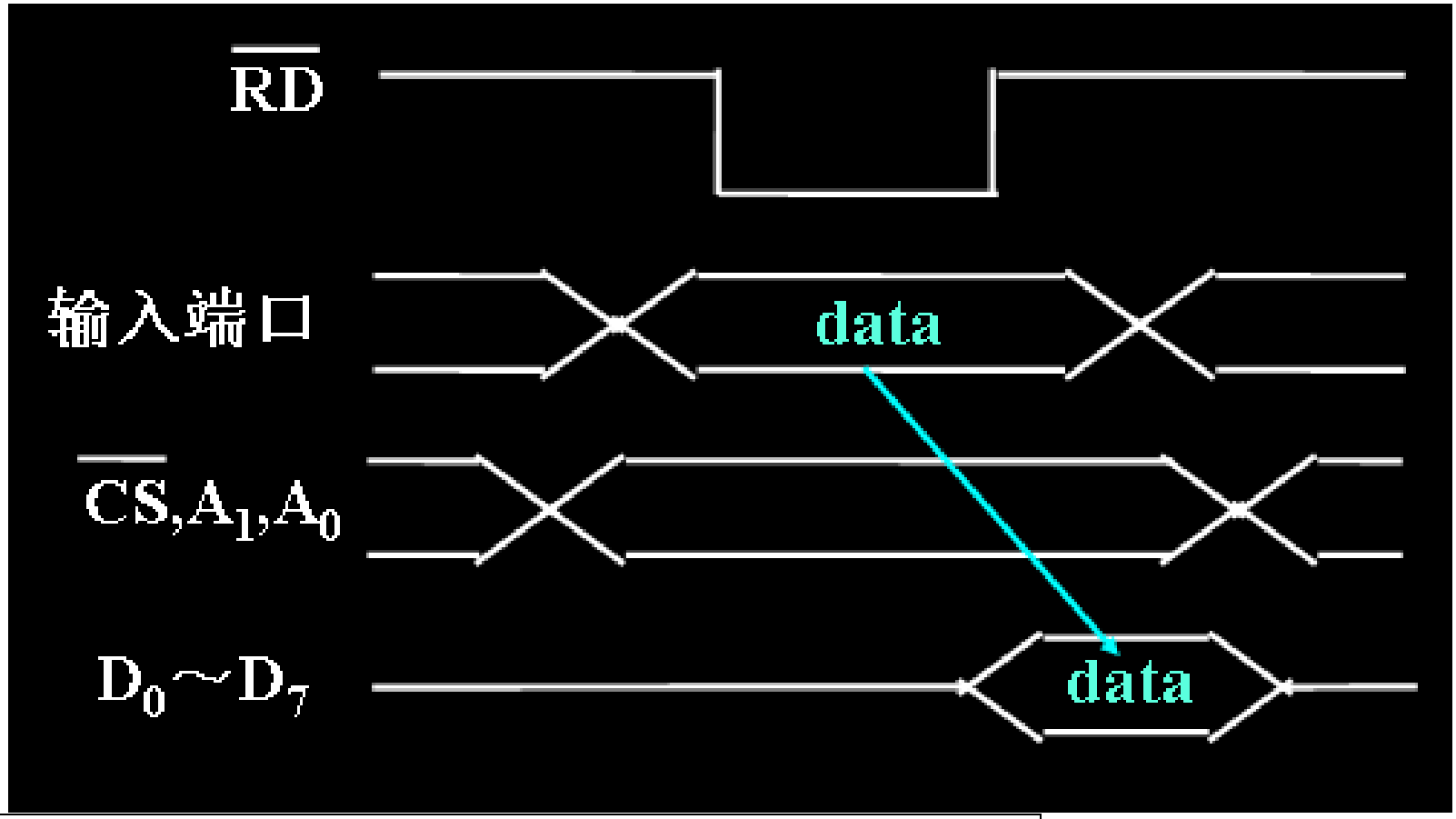
A_0 ~ A_1地址线
WR*写信号
RESET复位信号

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

10.1.2 8255A的工作方式

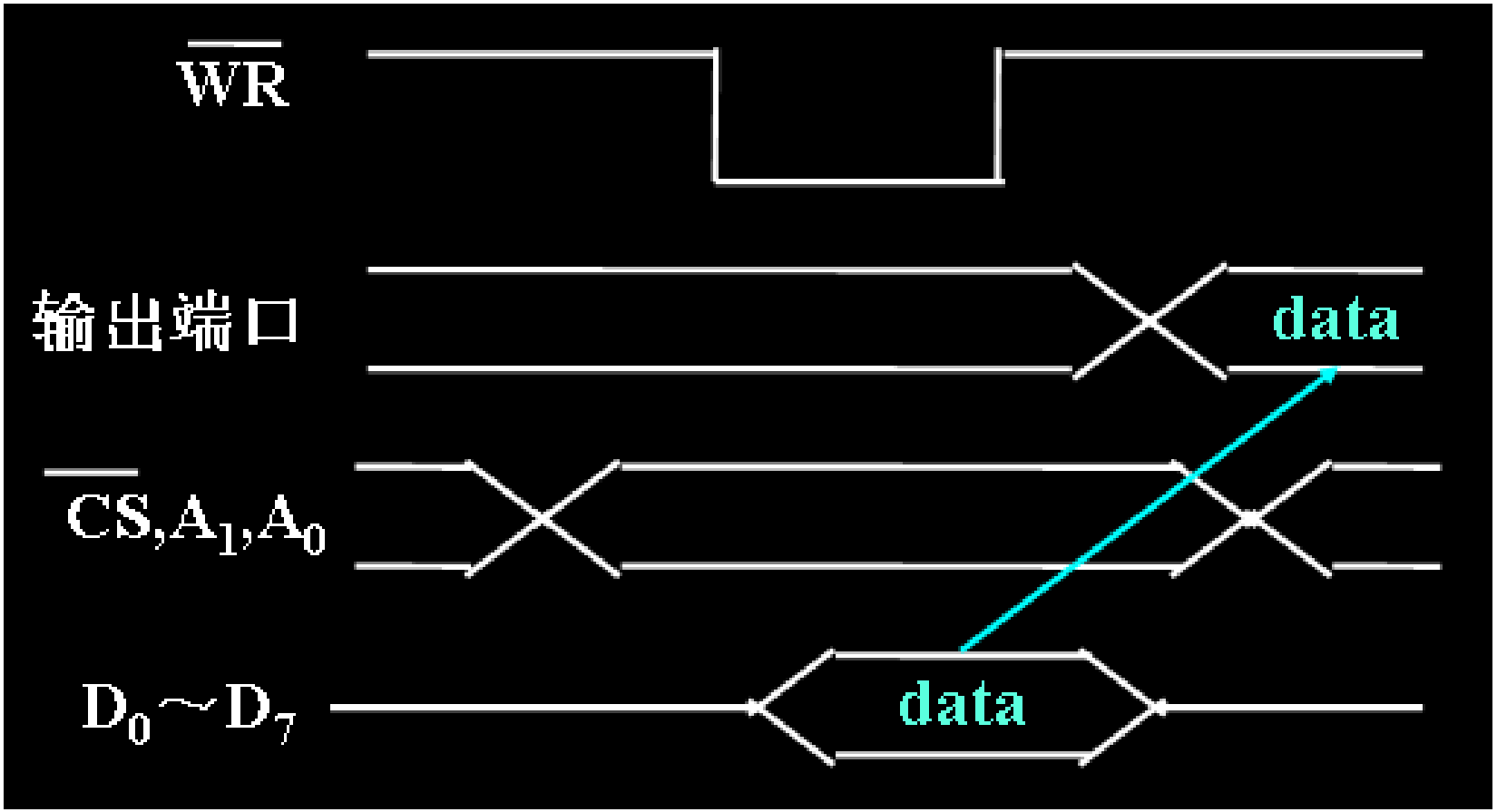
- 方式0：基本输入输出方式
  - 适用于无条件传送和查询方式的接口电路
- 方式1：选通输入输出方式
  - 适用于查询和中断方式的接口电路
- 方式2：双向选通传送方式
  - 适用于与双向传送数据的外设
  - 适用于查询和中断方式的接口电路

方式0输入时序



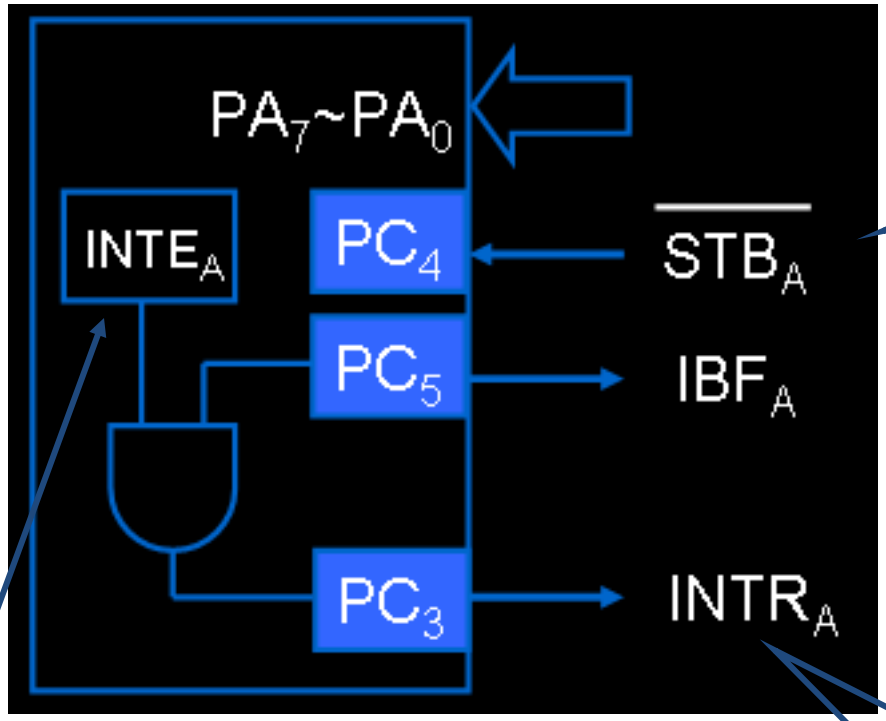
请体会这里8255A的数据缓冲作用

方式0输出时序



8255A对CPU通过它输出给外设的数据进行锁存

方式1输入引脚：A端口



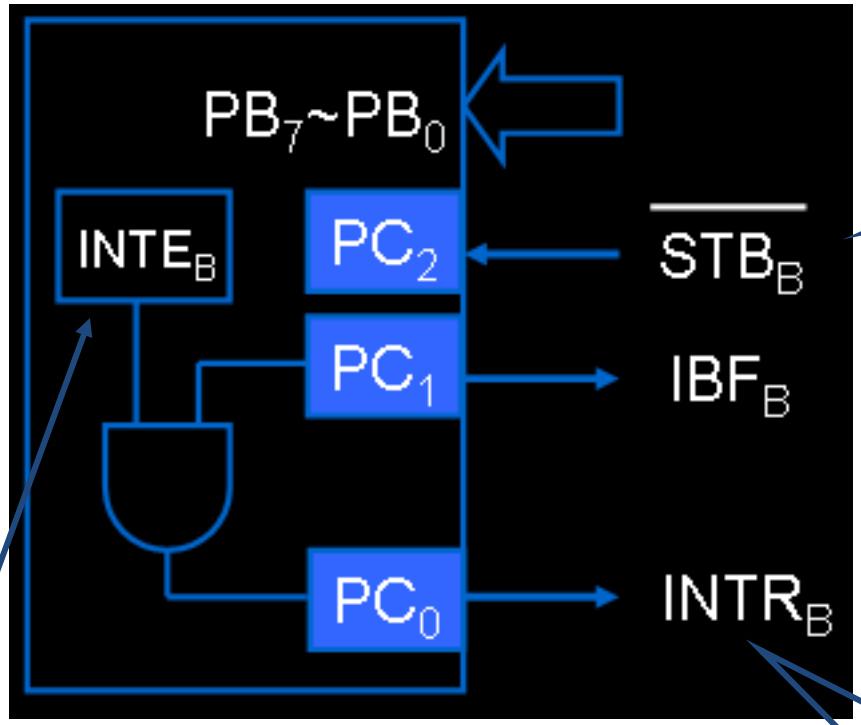
数据选通信号
表示外设已经准备好数据

输入缓冲器满信号
表示A口已经接收数据

中断请求信号
请求CPU接收数据

中断允许触发器

方式1输入引脚：B端口



数据选通信号
表示外设已经准备好数据

输入缓冲器满信号
表示B口已经接收数据

中断请求信号
请求CPU接收数据

中断允许触发器

方式1输入联络信号

- **方式1需借用端口C用做联络信号**
- **同时还具有中断请求和屏蔽功能**

- 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，将输入设备送来的数据锁存至8255A的输入锁存器

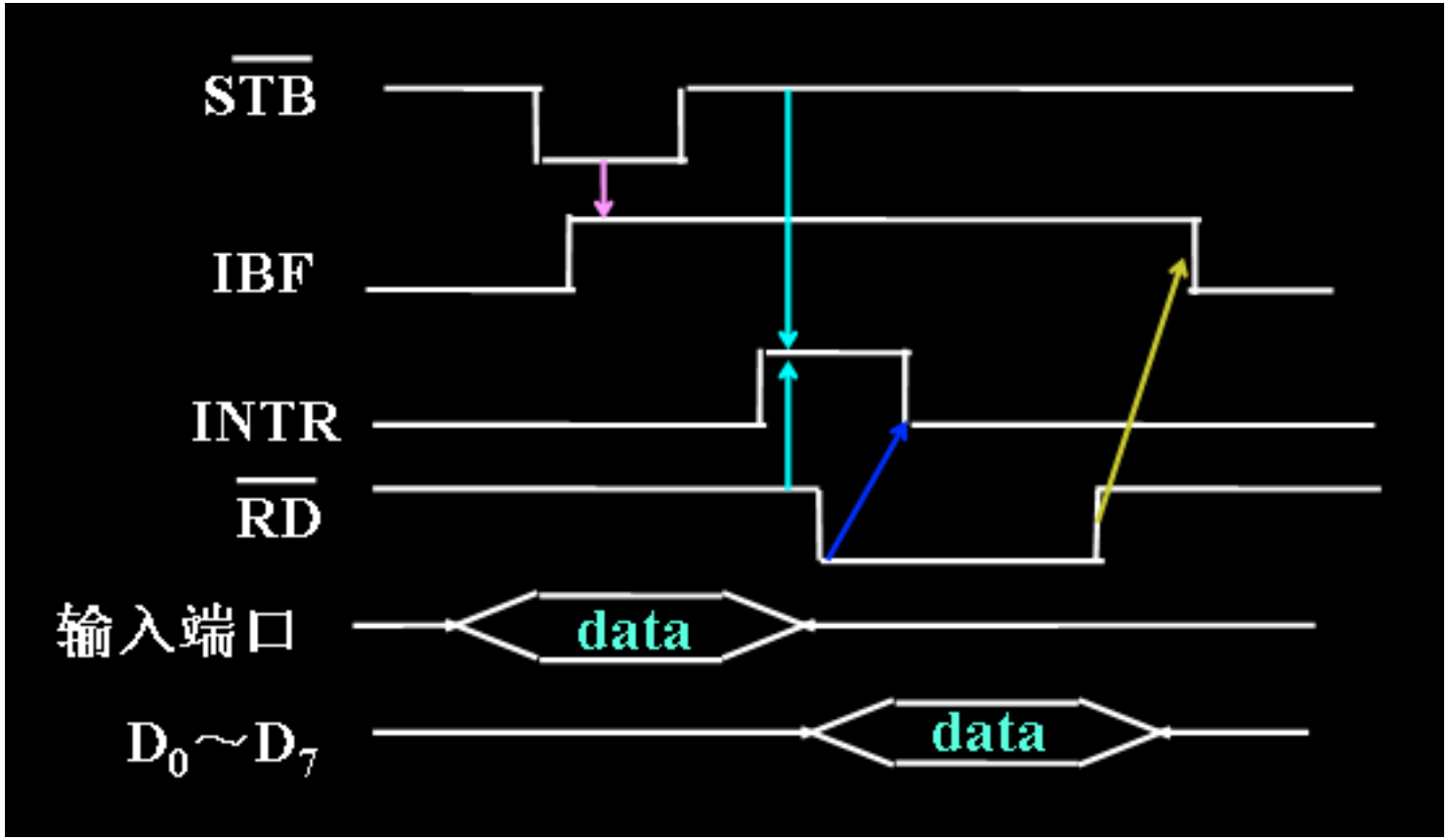
方式1输入联络信号

- IBF——输入缓冲器满信号，高电平有效
  - 8255A输出的联络信号。当其有效时，表示数据已锁存在输入锁存器
- 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

方式1输入联络信号

- **STB*和IBF是外设和8255A间的一对应答联络信号，其目的在于完成可靠的数据输入**

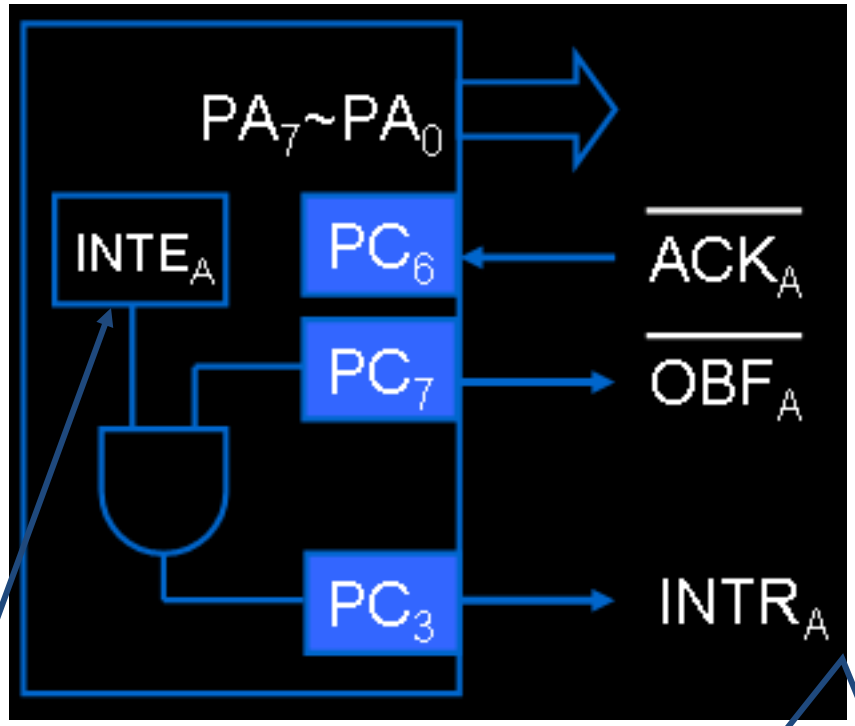
方式1输入时序



方式1中断控制

- 8255A的中断由中断允许触发器INTE控制
  - 置位允许中断，复位禁止中断
- 对INTE的操作通过写入端口C的对应位实现，INTE触发器对应端口C的位是作应答联络信号的输入信号的哪一位，只要对那一位置位/复位就可以控制INTE触发器
- 选通输入方式下
  - 端口A的INTEA对应PC4
  - 端口B的INTEB对应PC2

方式1输出引脚：A端口



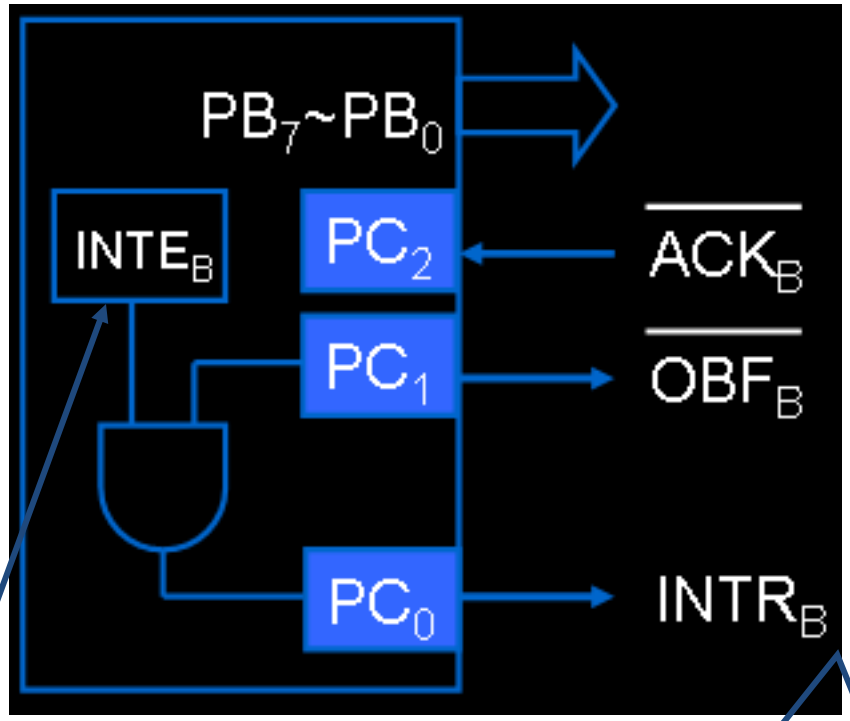
外设响应信号
表示外设已经接收到数据

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
请求CPU再次输出数据

方式1输出引脚：B端口



外设响应信号
表示外设已经接收到数据

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
请求CPU再次输出数据

方式1输出联络信号

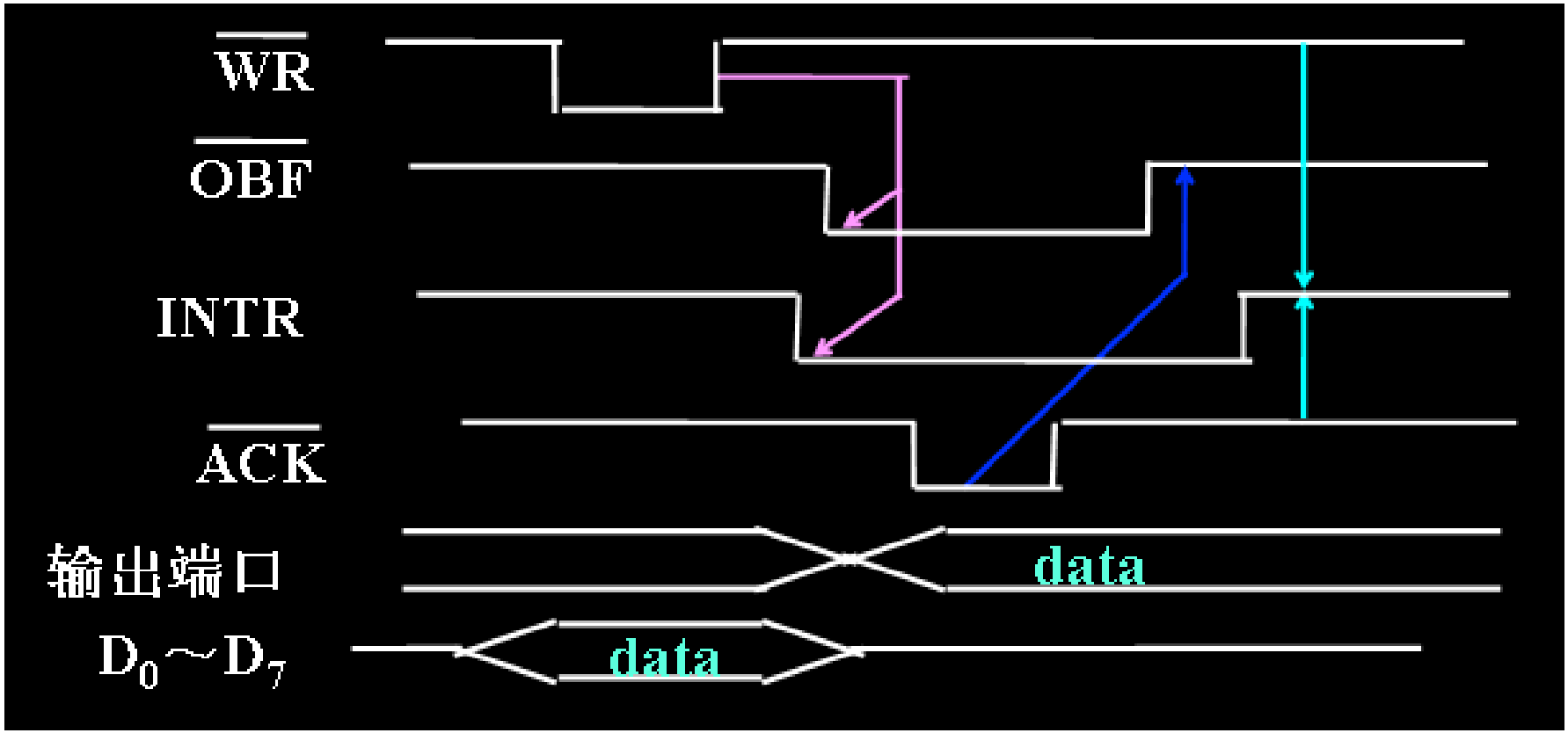
- OBF*——输出缓冲器满信号，低有效
  - 8255A输出给外设的一个控制信号，当其有效时，表示CPU已把数据输出给指定的端口，外设可以取走
- ACK*——响应信号，低有效
  - 外设的响应信号，指示8255A的端口数据已由外设接受
- INTR——中断请求信号，高有效
  - 当输出设备已接受数据后，8255A输出此信号向CPU提出中断请求，要求CPU继续提供数据

方式1输出联络信号

- **OBF*和ACK*是外设和8255A间的一对应答联络信号，目的在于完成可靠的数据输出**

- **端口A的INTEA对应PC6**
- **端口B的INTEB对应PC2**

方式1输出时序


$\overline{WR}$
$\overline{OBF}$
INTR
$\overline{ACK}$
输出端口
$D_0 \sim D_7$
data
data

方式2双向方式

- 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

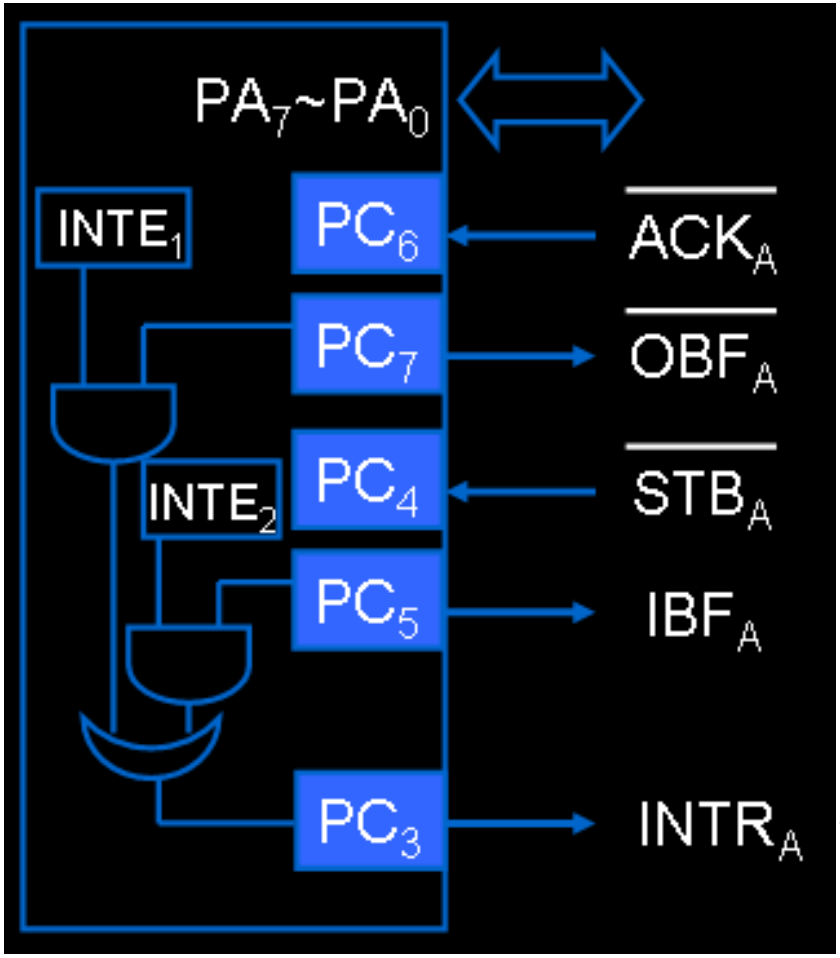
- 只有端口A可以工作于方式2，需要利用端口C的5个信号线，其作用与方式1相同

方式2双向方式

- 方式2的数据输入过程与方式1的输入方式一样

- 方式2的数据输出过程与方式1的输出方式有一点不同：数据输出时8255A不是在OBF*有效时向外设输出数据，而是在外设提供响应信号ACK*时才送出数据

方式2双向引脚



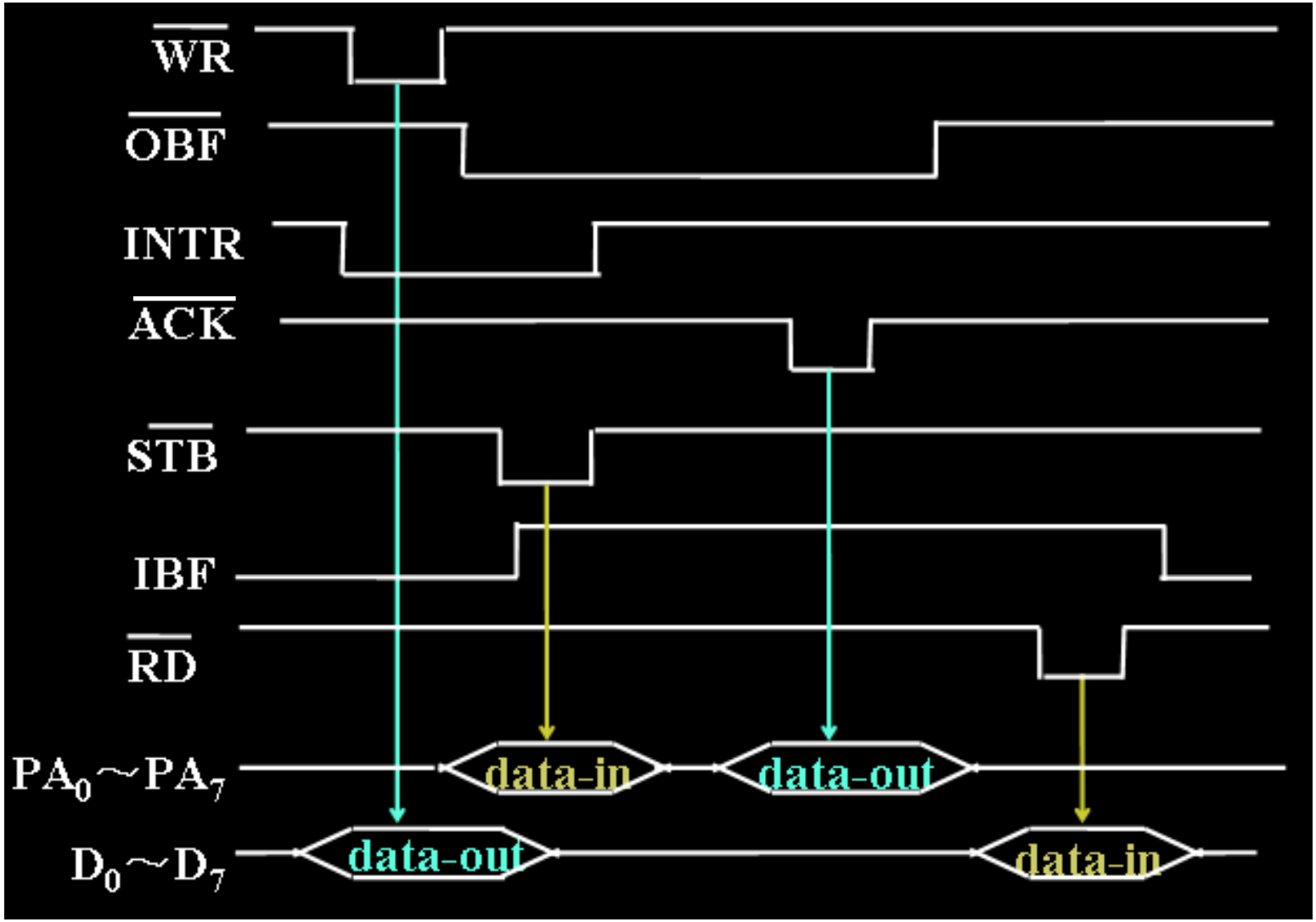
用PC_6设置$INTE_1$（输出）

用PC_4设置$INTE_2$（输入）

输入和输出中断通过

或门输出$INTR_A$信号

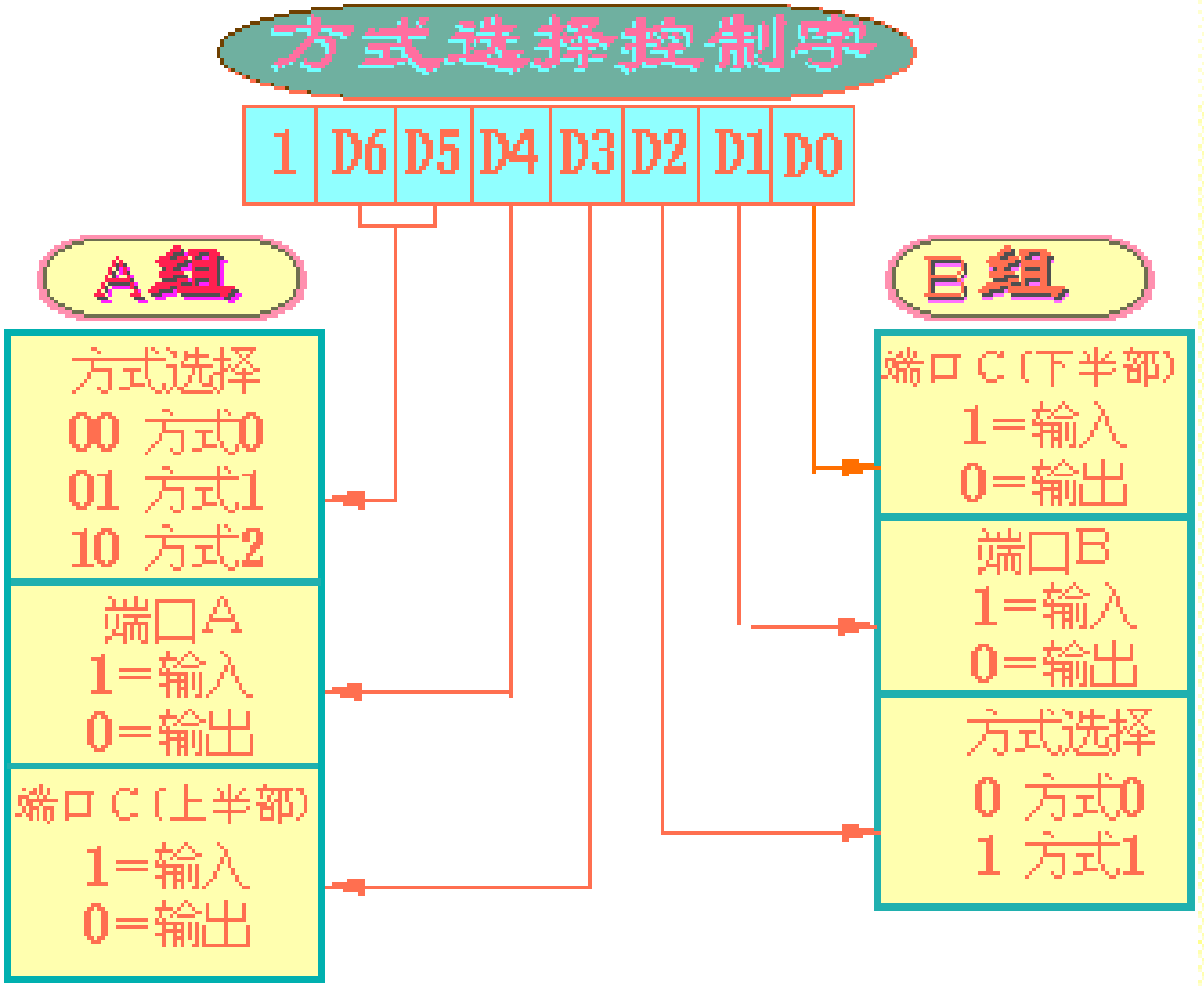
方式2双向时序


WR
OBF
INTR
ACK
STB
IBF
RD
PA0~PA7
data-in
data-out
D0~D7
data-out
data-in

10.1.3 8255A的编程

- 初始化编程：一个方式控制字
  - 采用控制I/O地址：A_1A_0＝11
- 工作过程中：通过数据端口对外设数据进行读写
  - 数据读写利用端口A、B和C的I/O地址，A_1A_0依次等于00、01、10
- IBM PC/XT机上，端口A、B、C和控制端口的I/O地址为60H、61H、62H和63H

1. 写入方式控制字：控制字格式



1. 写入方式控制字：示例

- 要求：
  - A端口：方式1输入
  - C端口上半部：输出，C口下半部：输入
  - B端口：方式0输出
- 方式控制字：10110001B或B1H
- 初始化的程序段：

```
mov dx,0fffeh     ;假设控制端口为FFFEH
mov al,0b1h       ;方式控制字
out dx,al   ;送到控制端口
```

2. 读写数据端口

- 初始化编程后：
  - 当数据端口作为输入接口时，执行输入IN指令将从输入设备得到外设数据
  - 当数据端口作为输出接口时，执行输出OUT指令将把CPU的数据送给输出设备
- 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
  - 不是读取外设数据
  - 读取的是上次CPU给外设的数据

2. 读写数据端口：示例

- 利用8255A的输出锁存能力，可实现按位输出控制
- 对输出端口B的PB7位置位的程序段：

```
mov dx,0fffah ;B端口假设为FFFAH
in al,dx   ;读出B端口原输出内容
or al,80h;使PB7＝1
out dx,al ;输出新的内容
```

3. 读写端口C

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

- **在控制上，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**

3. 读写端口C

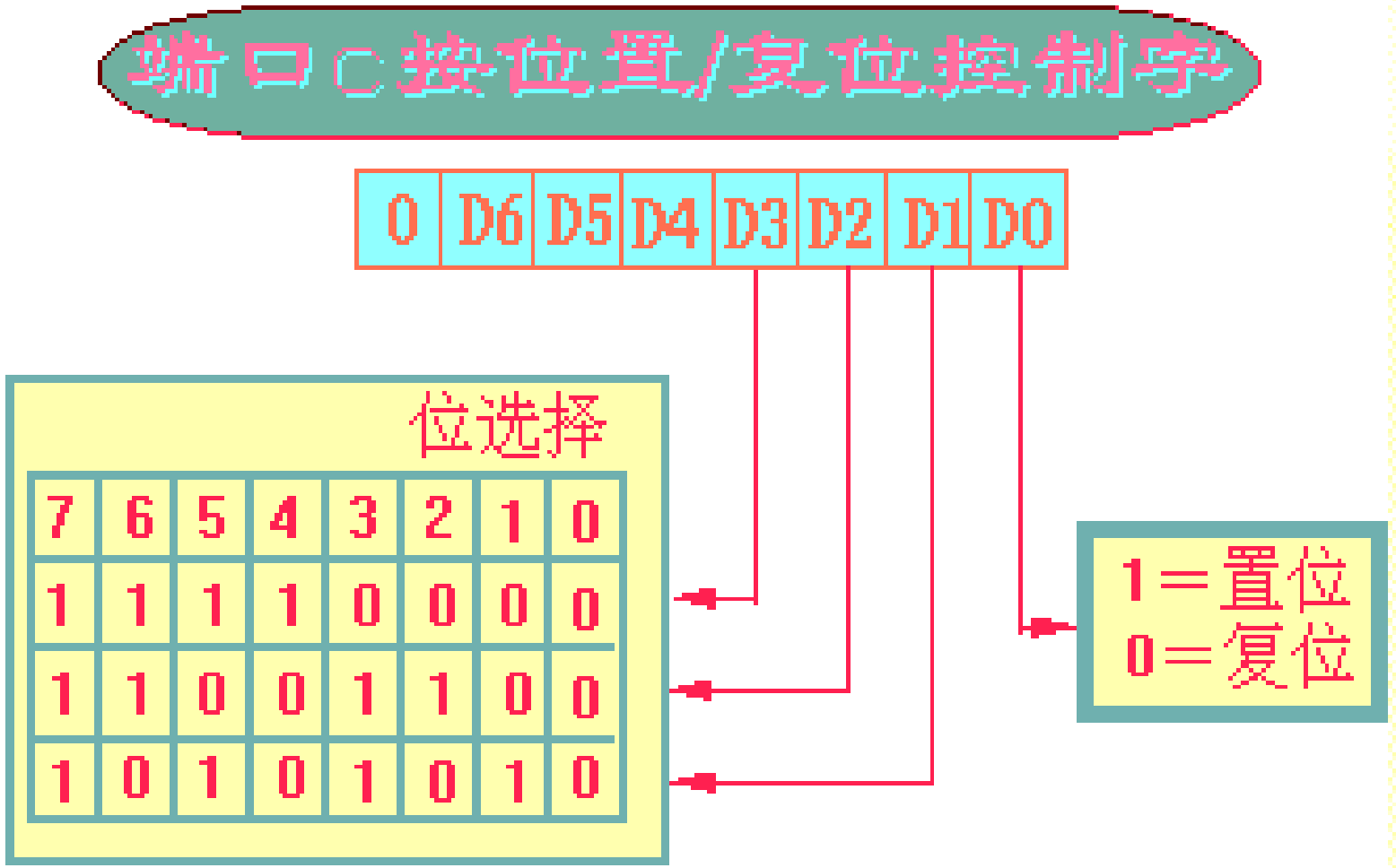
- 当A和B端口工作在方式1或方式2时，C端口的部分或全部引脚将被占用

- **其余引脚仍可设定工作在方式0**

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。这一数据被写进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端口写入位控字，使C端口的某个引脚输出1或0，或置位复位内部的中断允许触发器

端口C的位控制字


端口C按位置/复位控制字
0 D6 D5 D4 D3 D2 D1 D0
位选择
7 6 5 4 3 2 1 0
1 1 1 1 0 0 0 0
1 1 0 0 1 1 0 0
1 0 1 0 1 0 1 0
1＝置位
0＝复位

端口C的位控制字

- 位控制字写入控制端口
- 特别便于置位复位内部中断允许触发器INTE

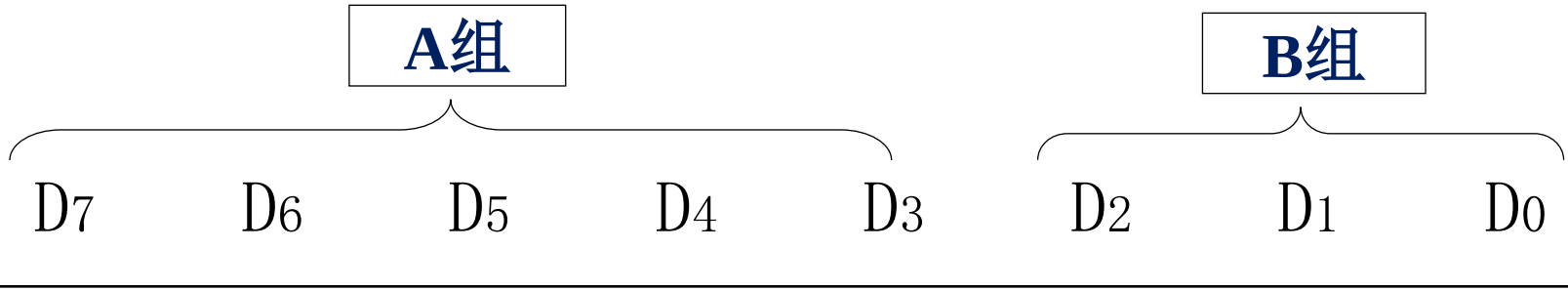
3. 读写端口C

- 读取的C端口数据有两种情况

- 未被A和B端口占用的引脚：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端口占用作为联络线的引脚：将读到反映8255A状态的状态字

端口C的状态字



方式1输入

I/O	I/O	IBFA	INTEA	INTRA	INTEB	IBFB	INTRB

方式1输出

$\overline{\text{OBFA}}$	INTEA	I/O	I/O	INTRA	INTEB	$\overline{\text{OBFB}}$	INTRB

方式2双向

$\overline{\text{OBFA}}$	INTE1	IBFA	INTE2	INTRA	×	×	×

10.2 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

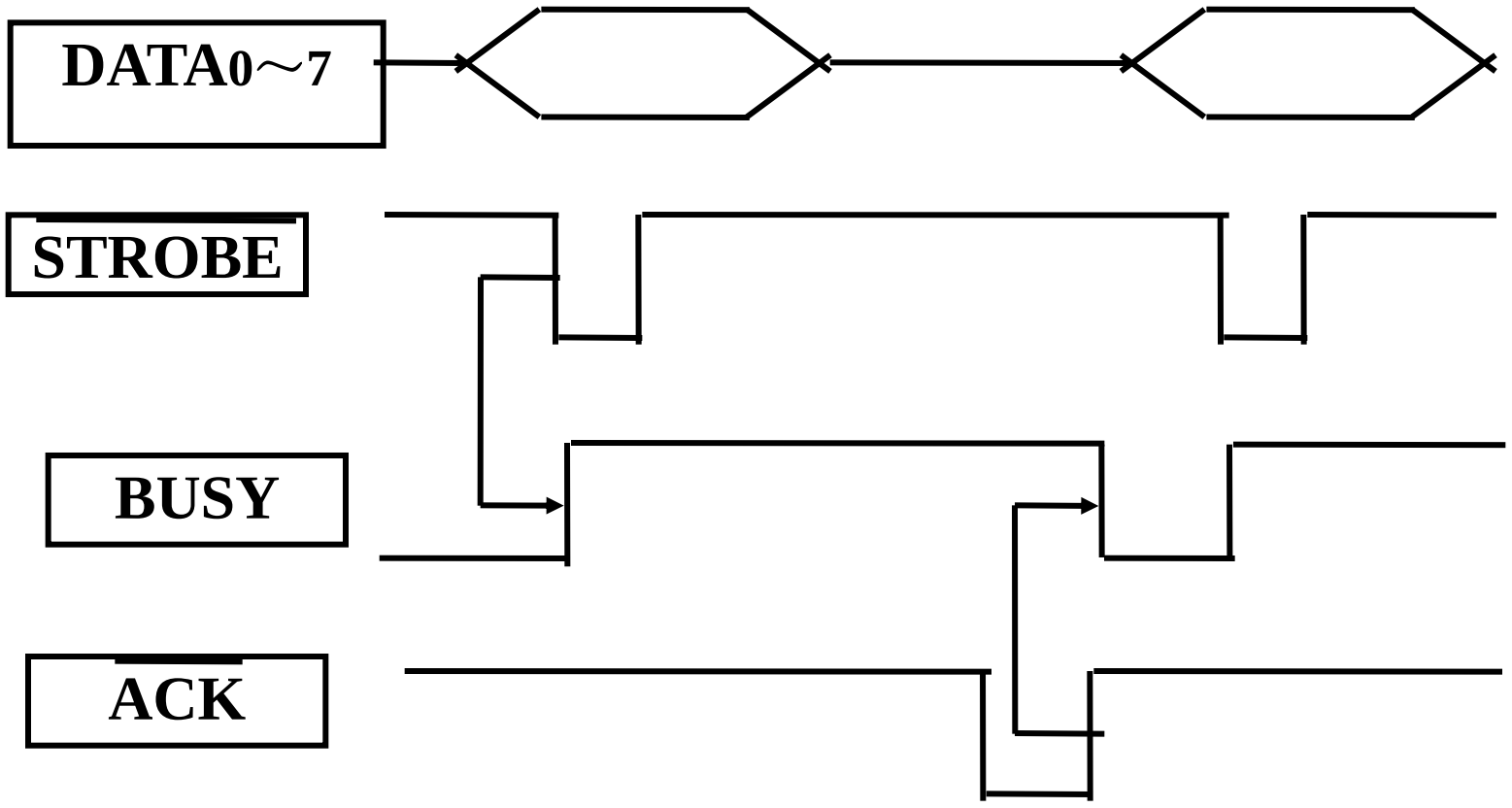
- 应用在IBM PC/XT微机上
- 应用于打印机接口电路
- 连接简易键盘
- 驱动LED数码管
- ……

10.2.1 8255A在IBM PC/XT上的应用

- 工作在基本输入/输出方式0
  - 端口A为方式0输入，用来读取键盘扫描码
  - 端口B工作于方式0输出，例如控制扬声器等
  - 端口C为方式0输入，读取系统状态和配置
- 系统的初始化编程：

```
mov al,10011001b  ;方式控制字99H
out 63h,al
```

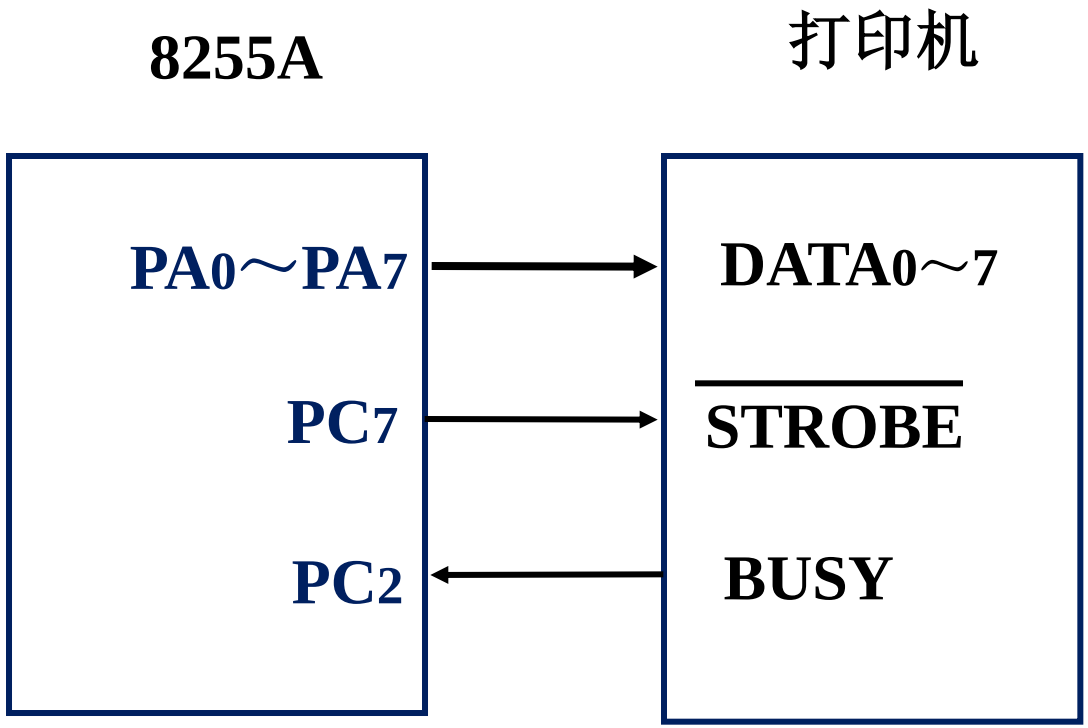
打印机接口的信号与时序



打印机接口的信号与时序

- 主机把数据送给引脚DATA0～DATA7
- 同时送出数据选通信号STROBE*
- 打印机在BUSY信号线上发出忙信号
- 打印机处理好输入的数据时
  - 撤消忙信号
  - 同时又送出一个响应信号ACK*

10.2.2 用8255A方式0与打印机接口



8255A的初始化 例10.1

- **mov dx,0fffeh**
- **;控制端口地址：FFFEH**
- **mov al,10000001B**
- **;方式控制字：81H**
- **out dx,al**
- **;A端口方式0输出，C端口上输出、下输入**
- **mov al,00001111B**
- **;端口C的复位置位控制字，使PC7＝1**
- **out dx,al**

打印子程序：查询 例10.1

```
printc    proc
          push ax
          push dx
prn:      mov dx,0fffch    ;读取端口C
          in al,dx         ;查询打印机状态
          and al,04h ;PC2＝BUSY＝0？
          jnz prn
          ;PC2＝1，打印机忙，则循环等待
```

打印子程序：输出 例10.1

- **mov dx,0fff8h**
- **;PC2＝0，打印机不忙，则输出数据**
- **mov al,ah**
- **out dx,al　;将打印数据从端口A输出**

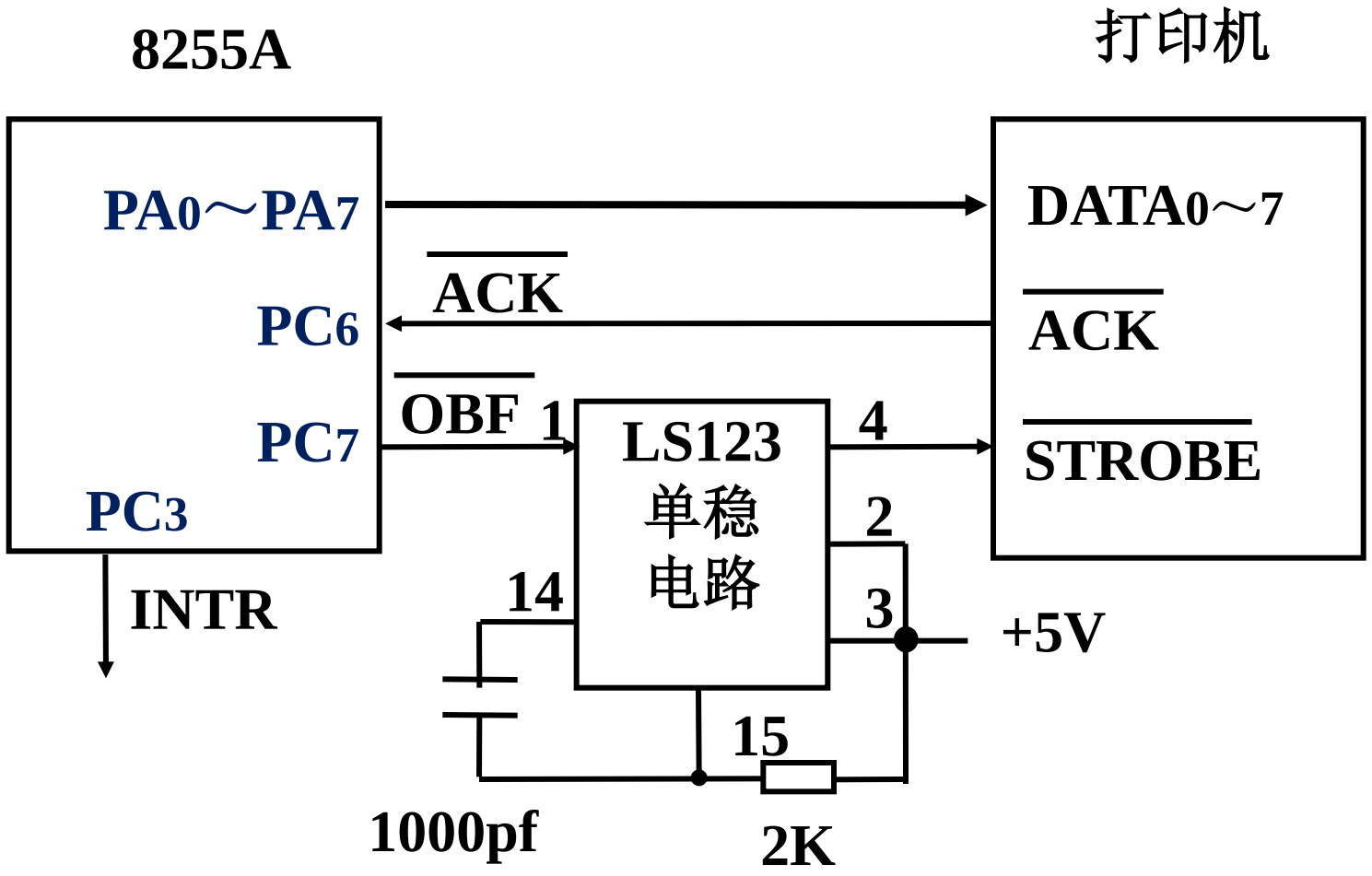
打印子程序：打印 例10.1

- **mov dx,0fffeh ;从PC7送出控制低脉冲**
- **mov al,00001110B ;置STROBE*＝0**
- **out dx,al**
- **nop ;产生一定宽度的低电平**
- **nop**
- **mov al,00001111B ;置=1**
- **out dx,al**
- **;最终，STROBE*产生低脉冲信号**

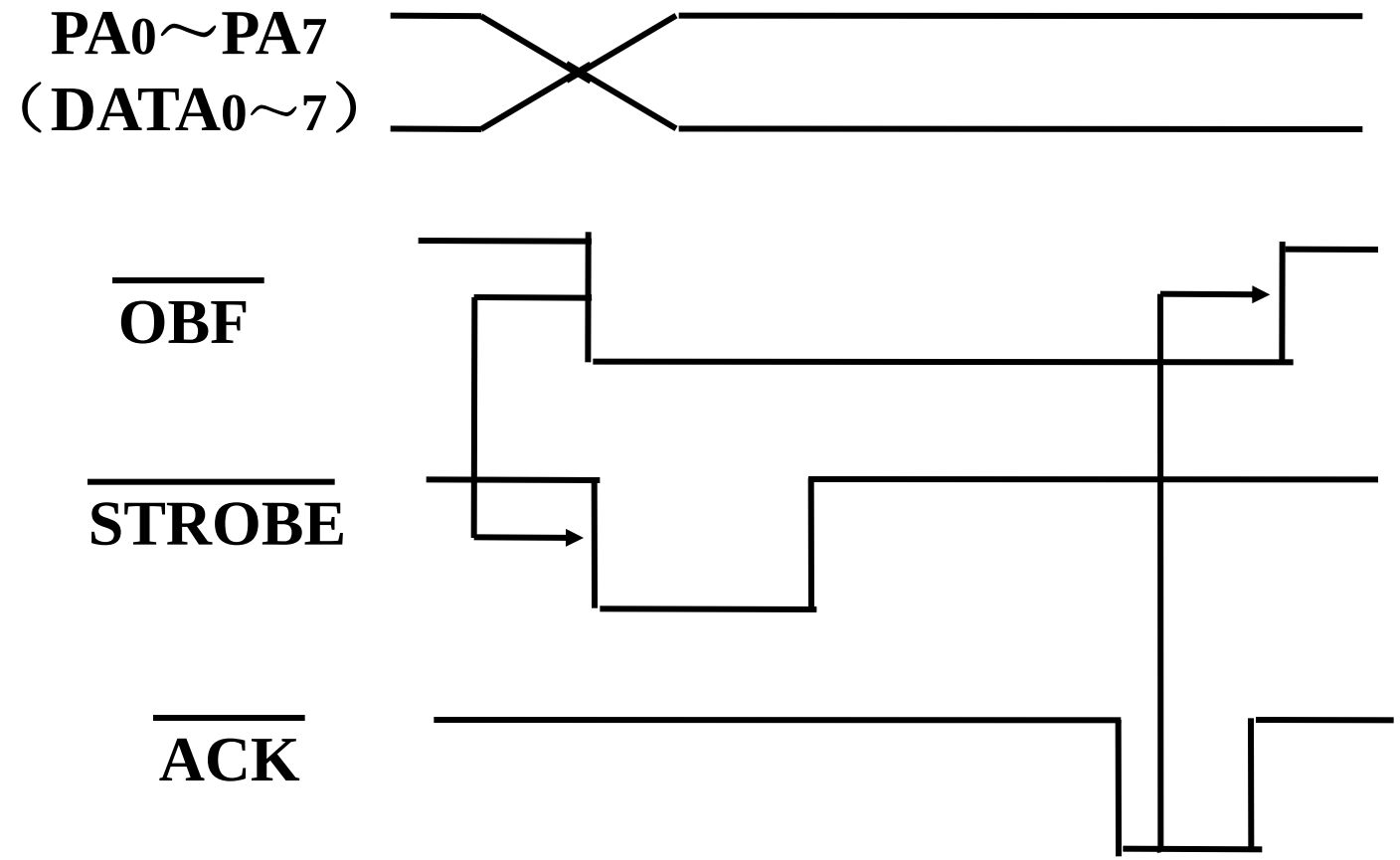
打印子程序：返回 例10.1

- **pop dx**
- **pop ax**
- **ret**
- **printc endp**

10.2.3 用8255A方式1与打印机接口



8255A方式1与打印机接口时序配合



8255A的初始化 例10.2

- **mov dx,0fffeh**
- **mov al,0a0h**
- **out dx,al**
- **mov al,0ch**
- **;使INTEA（PC6）为0，禁止中断**
- **out dx,al**
- **……**
- **mov cx,counter ;打印字节数送CX**
- **mov bx,offset buffer ;取字符串首地址**
- **call prints ;调用打印子程序**

打印子程序：输出 例10.2

```
prints    proc
          push ax     ;保护寄存器
          push dx
print1:   mov al,[bx]        ;取一个数据
          mov dx,0fff8h
          out dx,al   ;从端口A输出
```

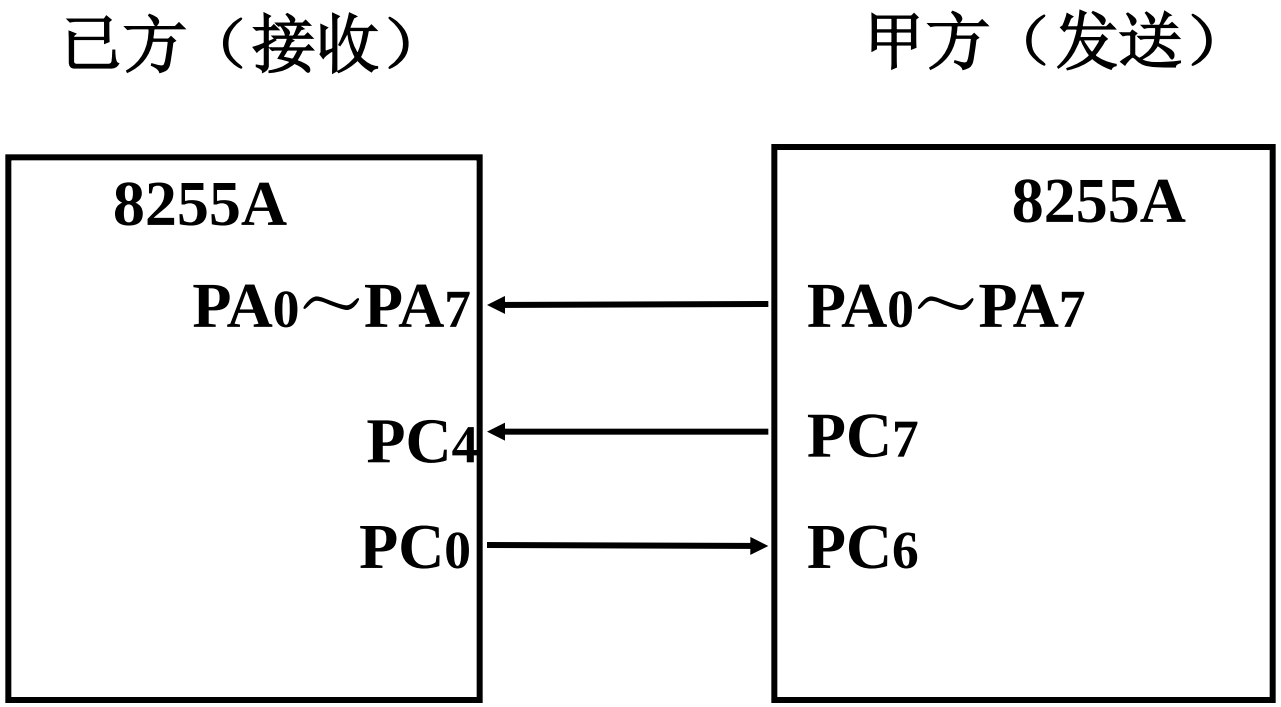
打印子程序：查询 例10.2

- **mov dx,0fffch**
- **print2: in al,dx**
- **test al,80h**
- **;检测（PC7）为1否?**
- **jz print2**
- **;为0，说明打印机没有响应，继续检测**

打印子程序：返回 例10.2

```
        inc bx
    ;为1，说明打印机已接受数据
        loop print1
    ;准备取下一个数据输出
        pop dx      ;打印结束，恢复寄存器
        pop ax
        ret    ;返回
prints  endp
```

10.2.4 双机并行通信接口



甲机的初始化 例10.3

- **mov dx,0fffeh**
- **mov al,0a0h**
- **out dx,al**
- **;工作方式字：端口A方式1输出**
- **mov al,0dh**
- **;使PC6（INTEA）＝1，允许中断**
- **out dx,al**

甲机发送程序 例10.3

- **trsmt:　mov dx,0fffch**
- **in al,dx**
- **;查询PC3（INTRA）＝1？**
- **and al,08h**
- **jz trsmt**
- **mov dx,0fff8h　;发送数据**
- **mov al,ah**
- **out dx,al**

乙机的初始化 例10.3

- **mov dx,0fffeh**
- **mov al,98h**
- **out dx,al**
- **;工作方式字：端口A方式0输入**
- **mov al,01h**
- **;使PC0（ACK*）＝1，因尚未收到数据**
- **out dx,al**

乙机:查询接收 例10.3

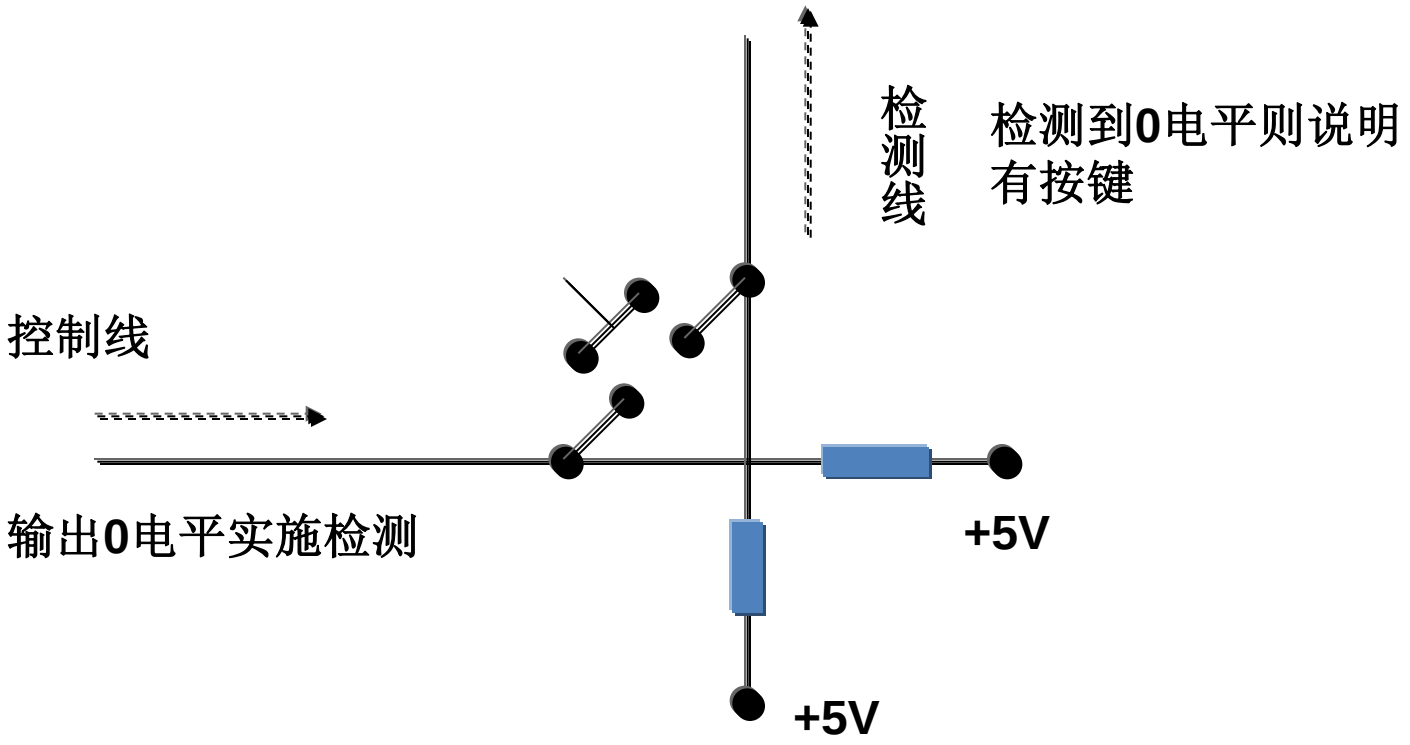
- **receive: mov dx,0fffch**
- **in al,dx**
- **;查询PC4（OBF*）＝0？**
- **and al,10h**
- **jnz receive**
- **mov dx,0fff8h ;接收数据**
- **in al,dx**
- **mov ah,al**

乙机：接收响应 例10.3

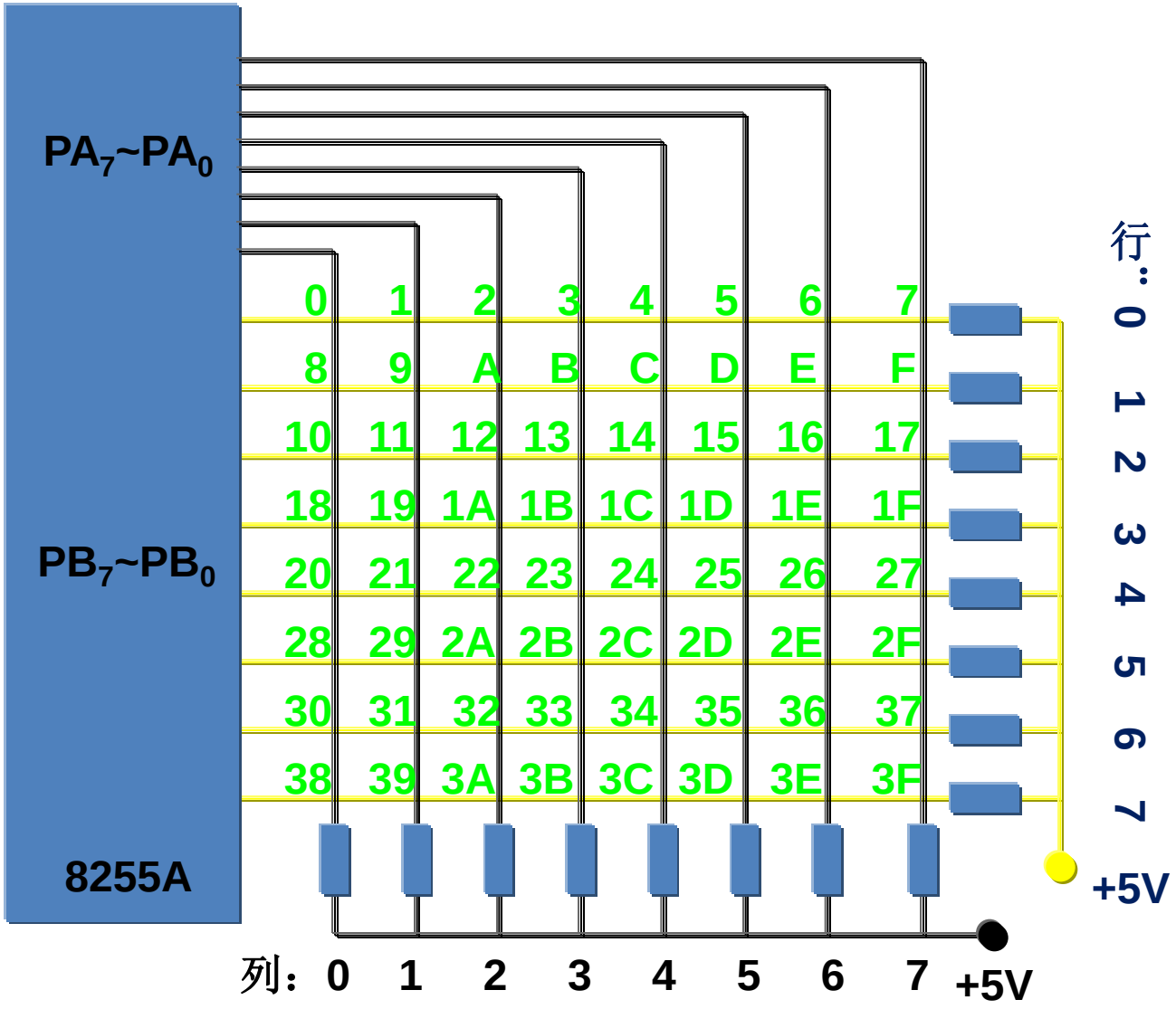
- **mov dx,0fffeh**
- **mov al,00h　　;使PC0（ACK*）＝0**
- **out dx,al**
- **nop**
- **;适当延时，产生一定宽度的低脉冲**
- **nop**
- **mov al,01h　　;使PC0（ACK*）＝1**
- **out dx,al　;产生低脉冲ACK*信号**

10.3 键盘及接口

- 简易键盘工作原理：



与8255连接的简易矩阵键盘



判断是否有按键

```
Key0:  mov  al, 00h
       mov  dx, rowport   ；行线端口地址
       out   dx, al
       mov  dx, colport    ；列线端口地址
       in      al, dx
       ；判断是否存在列线为低电平
       cmp   al, 0ffh
       jz      key0
       call    delay          ；防抖动延时，20ms
      ；防抖动，判断延时后按键是否仍然存在
       in      al, dx
       cmp   al, 0ffh
       jz      key0
       ……
```

判断按键是否松开

```
Key1:  mov  al, 00h
       mov  dx, rowport   ；行线端口地址
       out   dx, al
       mov  dx, colport    ；列线端口地址
       in      al, dx
       ；判断是否存在列线为低电平
       cmp   al, 0ffh
       jnz      key1
       call    delay          ；防抖动延时，20ms
      ；防抖动，判断延时后按键是否仍然存在
       in      al, dx
       cmp   al, 0ffh
       jnz      key1
       ……
```

识别按键（扫描法）

```
；上接按键判断程序片段
；若有按键，执行以下程序
        mov  cx, 8
        mov  ah, 0feh
key2: mov   al, ah
        mov   dx, rowport
        out     dx, al                  ；仅扫描一行
        mov   dx, colport
        in      al, dx                  ；读取此行列状态
```

识别按键（扫描法）

```
        cmp  al, 0ffh
        jnz     key3     ；存在按键则转移
        rol     ah, 1
        loop   key2
        jmp    key0
key3: ……                ；按键处理
        jmp    key0
```

识别按键（反转法）

- ；请自行添加防抖动功能
- key2: mov al, 00h
- mov dx, rowport
- out dx, al
- mov dx, colport
- in al, dx
- cmp al, 0ffh
- jz key2 ；无按键则循环等待

识别按键（反转法）

- push ax ；保护列状态
- push ax
- …… ；交换行列口输入、输出性质
- mov dx, colport
- pop ax ；恢复列状态
- out dx, al ；按列进行扫描
- mov dx, rowport
- in al, dx ；读取行状态

识别按键（反转法）

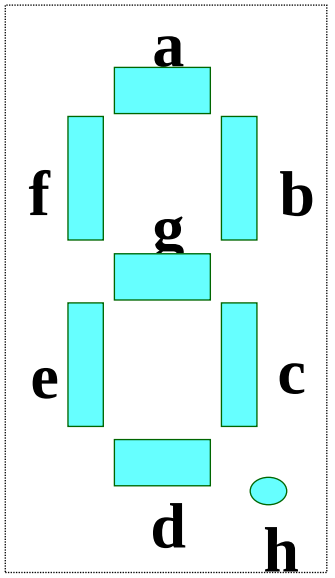
- pop bx ；恢复列状态
- mov ah, bl ；组合行列状态
- …… ；按键处理

10.4 LED数码管及其接口

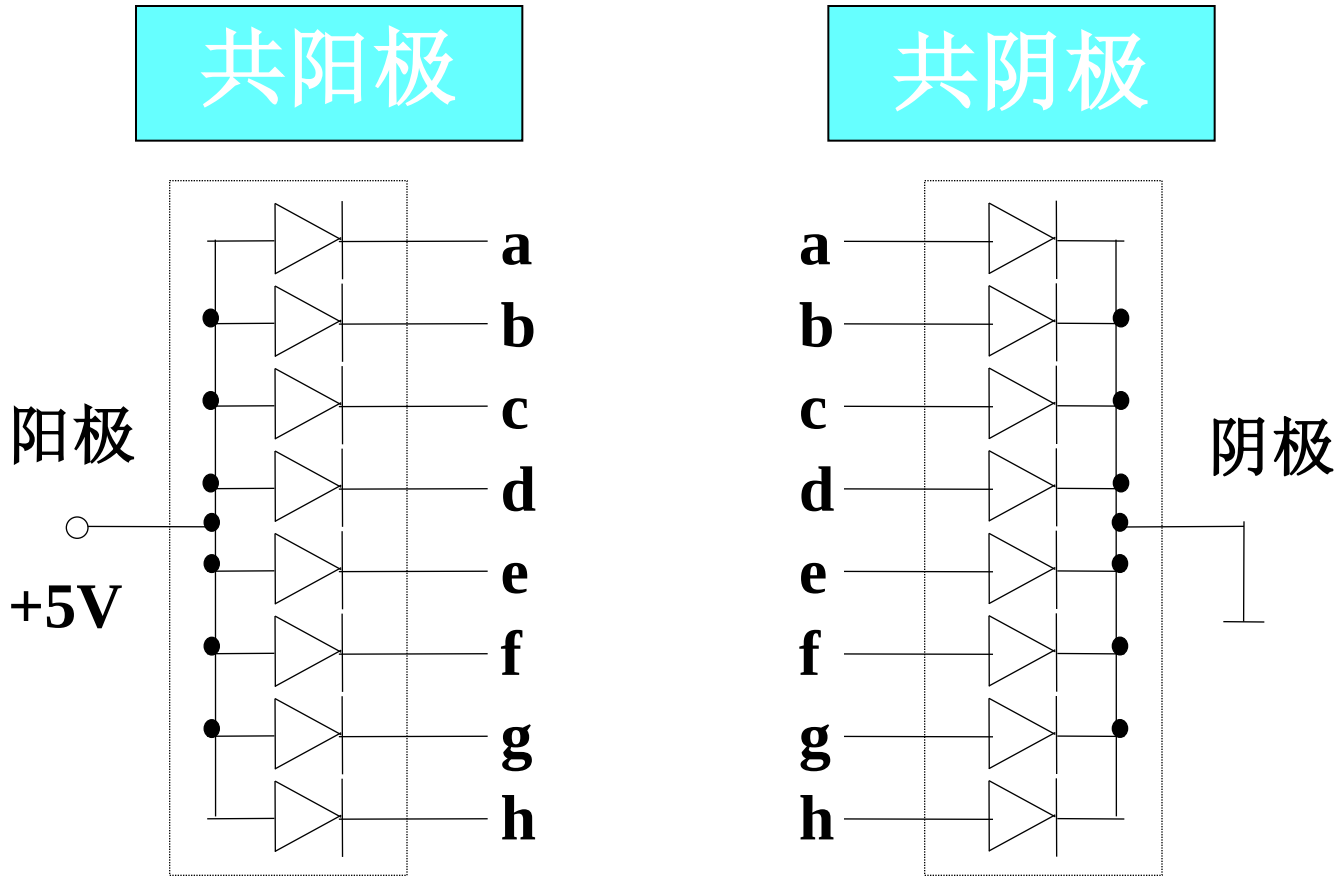
- 发光二极管LED是最简单的显示设备
- 由7段LED就可以组成的LED数码管
- LED数码管广泛用于单板微型机、微型机控制系统及数字化仪器中
- LED数码管可以显示内存地址和数据等

1. LED数码管的工作原理

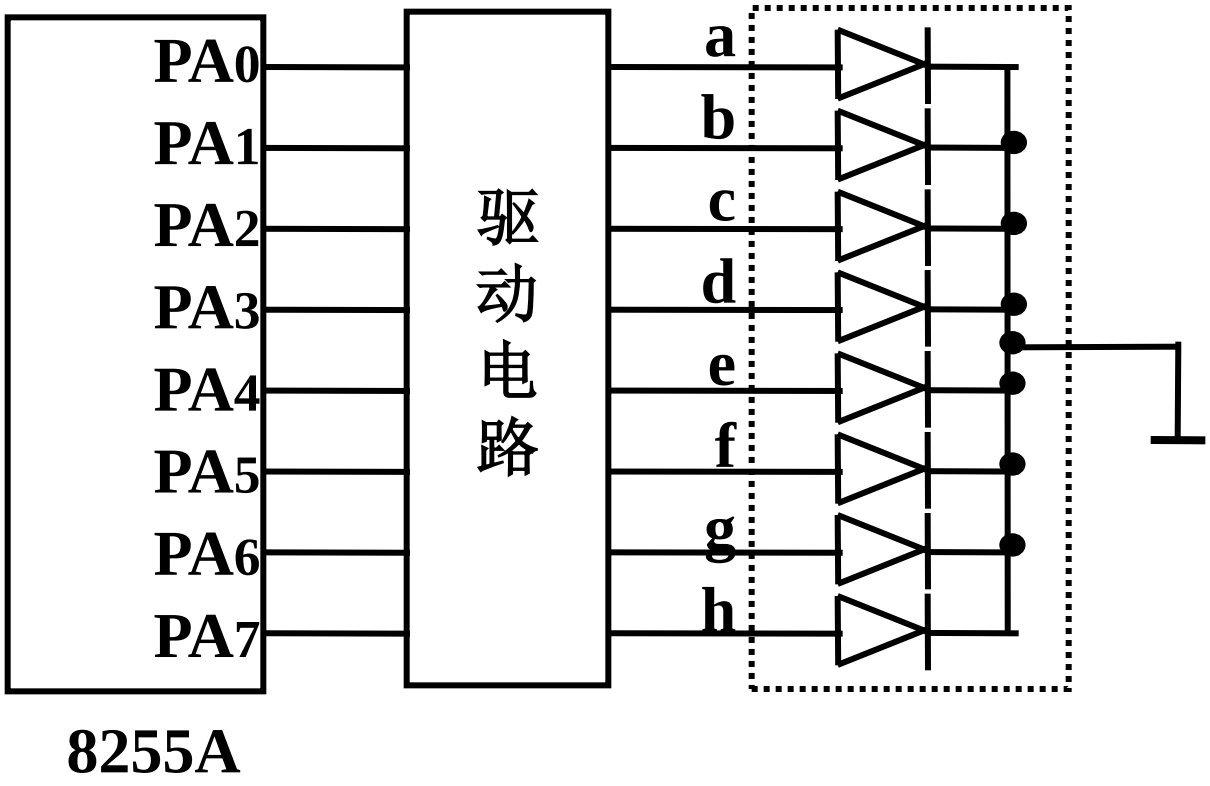
- 主要部分是7段发光管
- 顺时针分别称为a、b、c、d、e、f、g
- 有的产品还附带有一个小数点h
- 通过7个发光段的不同组合
  - 主要显示0～9
  - 也可显示A～F（16进制数）
  - 还可显示个别特殊字符：－、P



LED数码管的结构



2. 单个LED数码管的显示

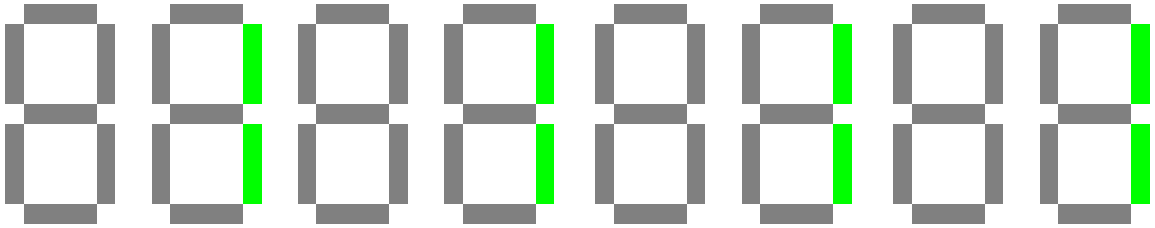


单个数码管的显示 软件译码

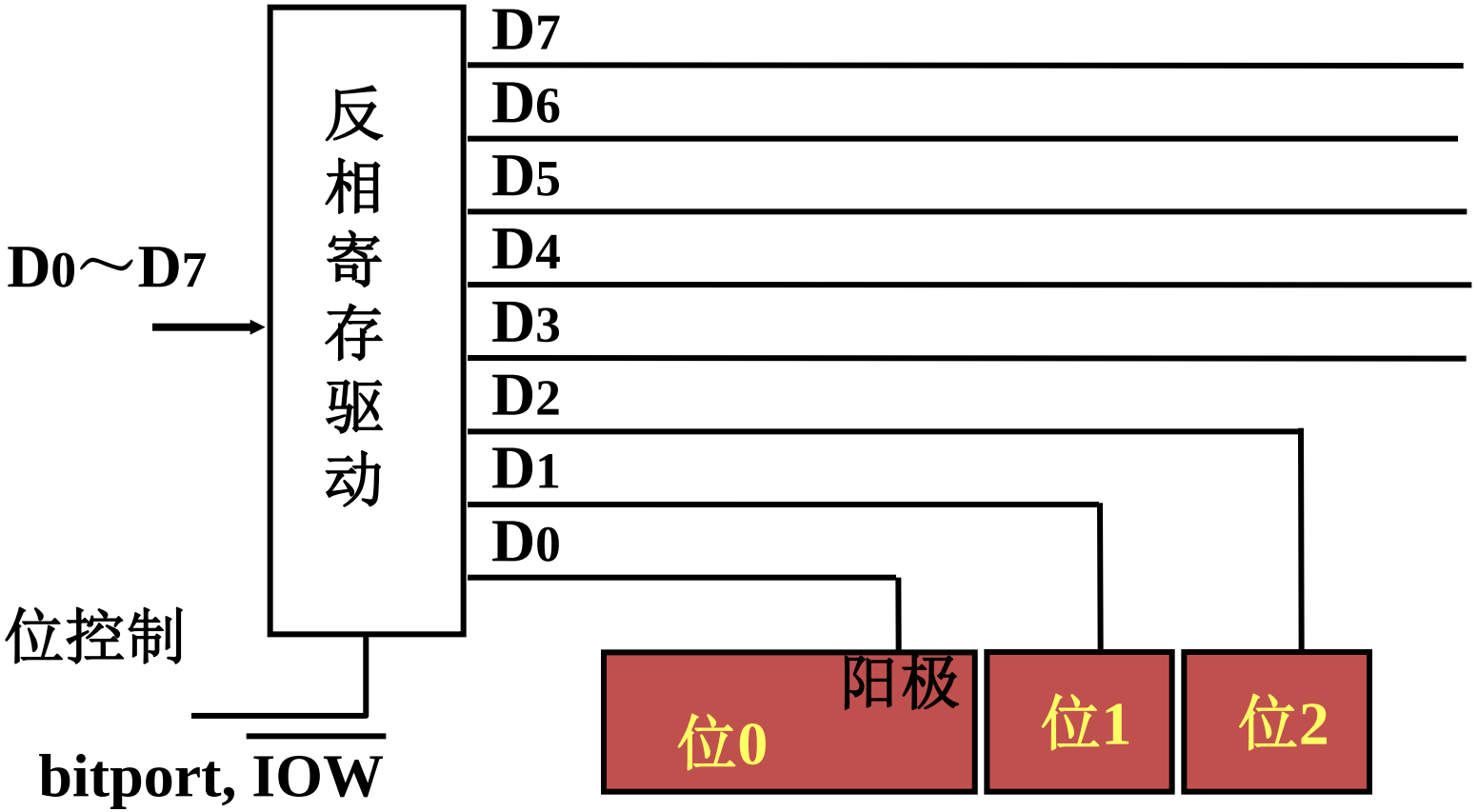
- **LEDtb db 3fh,06h,5bh,……　　;显示代码表**
- **……**
- **mov al,1　　;AL←要显示的数字**
- **mov bx,offset LEDtb**
- **xlat**
- **;换码：AL←DS:[BX＋AL]**
- **mov dx,port**
- **out dx,al　　;输出显示**

3. 多个LED数码管的显示

- 8个数码管：用2个8位输出端口控制
- 硬件上用公用的驱动电路来驱动各数码管
- 软件上用扫描方法实现数码显示



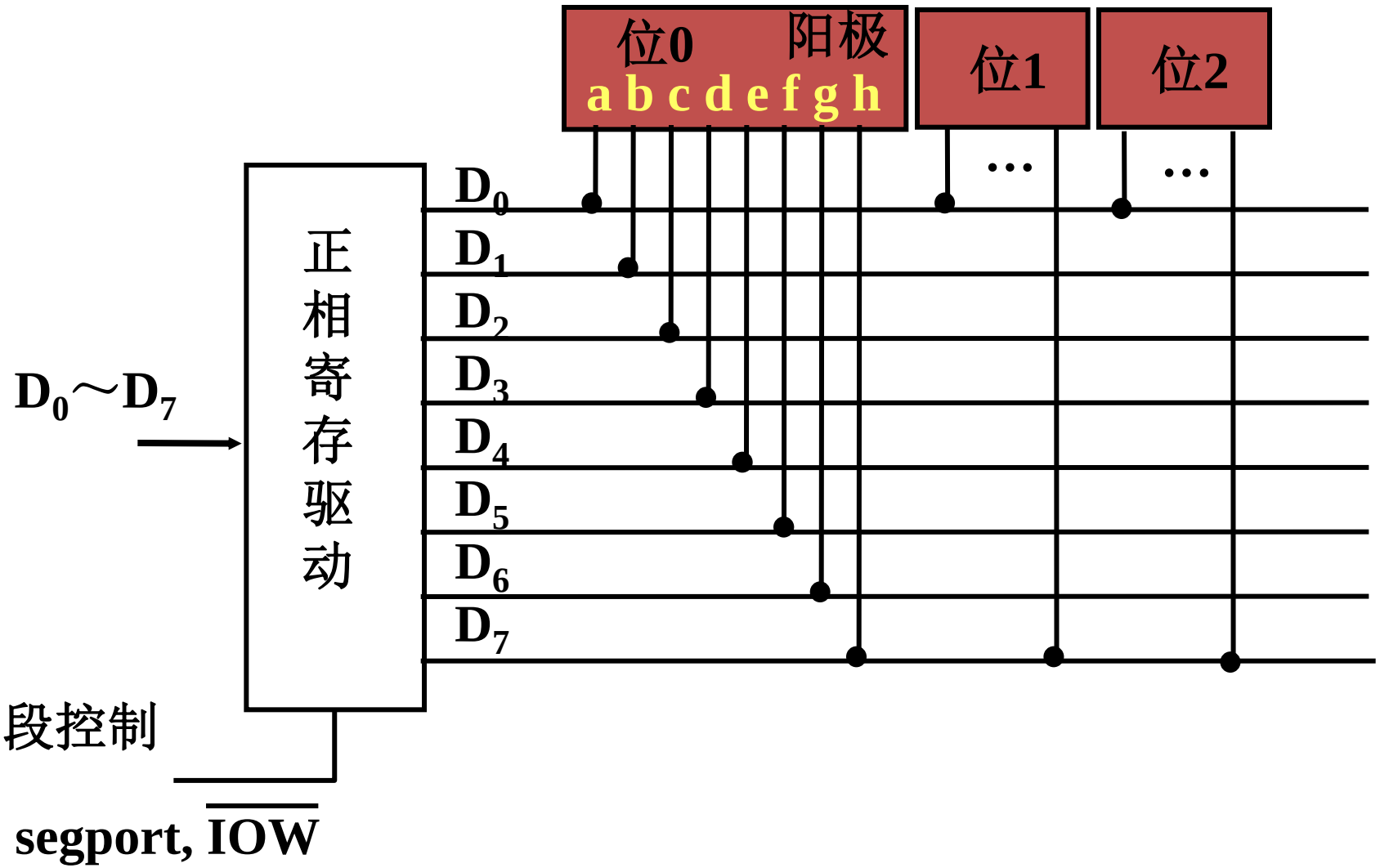
位控制端口电路



位控制端口作用

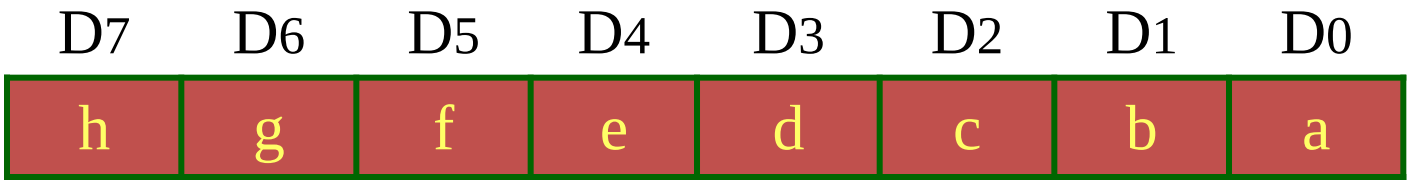
- 控制哪个（位）数码管显示
- 当位控制端口的控制码某位为低电平时，经反相驱动，便在相应数码管的阳极加上了高电平，这个数码管就可以显示数据
- **位控制：Di＝0，相应位发光**

段控制端口电路


位0
阳极
a b c d e f g h
位1
位2
…
…
D0
D1
D2
D3
D4
D5
D6
D7
正相寄存驱动
D0～D7
段控制
segport, IOW

段控制端口作用

- 控制哪个**段**显示，决定具体显示什么数码
- 段控制端口送出显示代码到数码管相应段
- 此端口由8个数码管共用
- **段控制：Di＝0，相应段发光**
- **通过位、段控制端口的共同作用**
- **才能确定哪个数码管显示什么数码**



数码缓冲区 例10.4

- ;数据段
- LEDdt db 8 dup(0) ;数码缓冲区
- ;主程序
- mov si,offset LEDdt
- call LEDdisp ;调用显示子程序

获取显示代码 例10.4

- **LEDdisp proc**
- **push ax**
- **push bx**
- **push dx**
- **mov bx,offset LEDtb**
- **mov ah,0feh ;指向最左边数码管**
- **LED1: lodsb ;取出要显示的数字**
- **xlat cs:LEDtb**
- **;得到显示代码：AL←CS:[BX＋AL]**

数码显示 例10.4

- **mov dx,segport ;segport为段控制端口**
- **out dx,al ;送出段码**
- **mov al,ah ;取出位显示代码**
- **mov dx,bitport ;bitport为位控制端口**
- **out dx,al ;送出位码**
- **call delay ;实现数码管延时显示**

显示下位数码 例10.4

```
        rol ah,1        ;指向下一个数码管
        cmp ah,0feh;最右边的数码管？
        jnz LED1        ;显示下一个数字
        pop dx
        pop bx
        pop ax
        ret     ;8位数码管都显示
LEDtb   db 0c0h,0f9h, ……
LEDdisp endp
```

软件延时 例10.4

```
timer       = 10   ;延时常量
delay       proc
            push bx
            push cx
            mov bx,timer
      ;外循环：timer确定的次数
delay1:     xor cx,cx
delay2:     loop delay2
      ;内循环：65536次循环
```

软件延时 例10.4

```
        dec bx
        jnz delay1
        pop cx
        pop bx
        ret
delay   endp
；通过控制重复频率和延时时间就
；可以得到各种显示效果
```

10.5 并行打印机接口

- 一般采用Centronics标准接口或其简化接口
- Centronics接口是的一个并行接口协议
- 这个协议规定了36脚簧式插头座和信号含义
- 其中前11条线是关键信号，他们是8条数据线、3条联络线（选通、响应和打印机忙）
- 还有一些特殊控制线、状态线
- PC系列机的并行打印机接口是一个25针插口

1. 控制打印机的输出信号

- SLCTIN*选择输入——相当于打印机选中信号
- INIT*初始化——使打印机被复位成初始状态
- AUTOFEEDXT*自动走纸——使打印机打印后自动走纸一行
- STROBE*选通——用于使打印机接收数据的选通信号。负脉冲的宽度在接收端应大于0.5μs，数据才可靠地存入打印机数据缓冲区

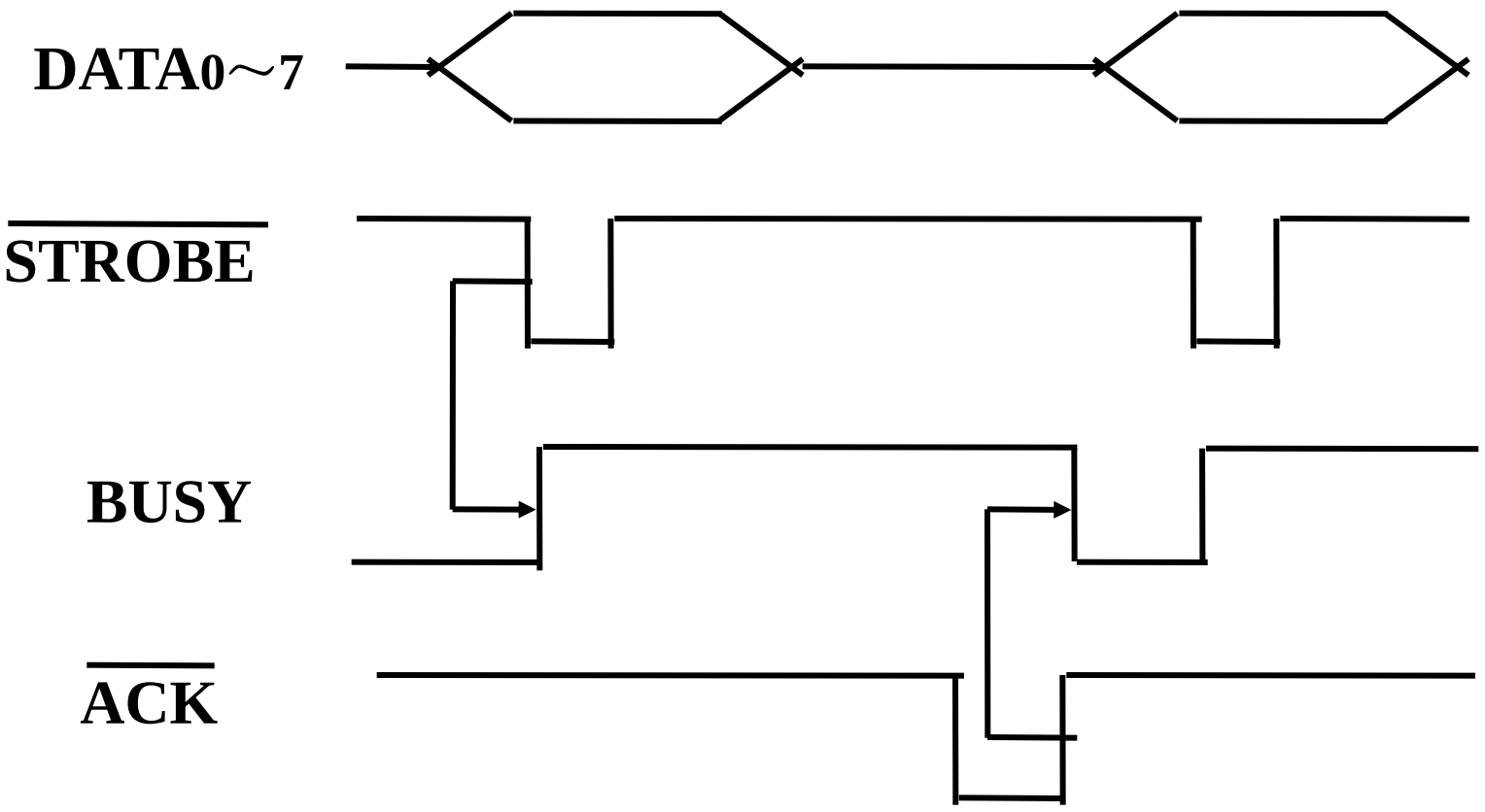
2. 反映打印机状态的输入信号

- BUSY忙——表示打印机不能接收数据
- ACK*响应——打印机接收一个数据字节后就回送一个响应的负脉冲信号（脉宽约为5μs），表示打印机已准备好接收新数据
- PE纸用完——说明打印机无纸
- SLCT选择——表示处于联机选中状态
- ERROR*错误——当打印机处于无纸、脱机或错误状态之一时，这个信号变为低电平

3. 输出数据线

- DATA0～DATA7——8位并行数据信号线
- 打印数据通过它们送至打印机
- 8位数据的可靠输出通过选通STROBE*、响应ACK*和忙BUSY三个联络信号控制

打印机时序



第10章教学要求

- 1. 掌握8255A的结构特点和引脚功能
- 2. 掌握8255A的各种工作方式、编程及方式0/1的应用

第10章教学要求

- 4. 掌握LED数码管的工作原理和多位显示方法