

北京科技大学

University of Science & Technology Beijing



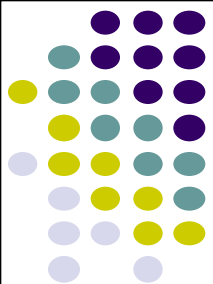
汇编语言与接口技术

北京科技大学



北京科技大学
University of Science and Technology Beijing

计算机与通信工程学院



第7章 串并行接口技术

微机系统中多采用接口芯片作为接口电路；实
现微机与外设间的数据传送

本节主要介绍

并行通信接口8255A



并行与串行通信

串行通信：把一个字符的n个数位，用1条

一进制位组成

特别提示：

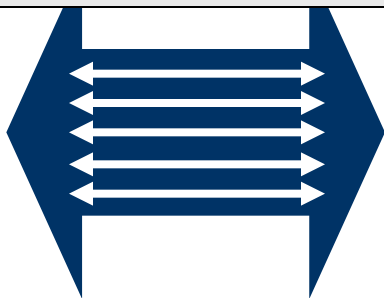
“并行”指的是接口与I/O设备一侧的数据线是并行的

“串行”指的是接口与I/O设备一侧的数据线是串行的

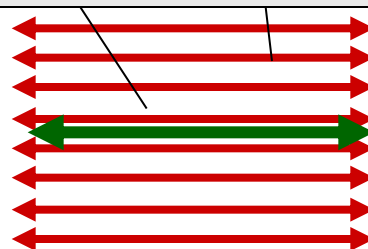
接口与总线一侧的连接总是并行的

接口芯片应支持无条件、查询、中断三种传送方式

CPU




接口
电路
芯片



I/O
外设



8255A的基本性能

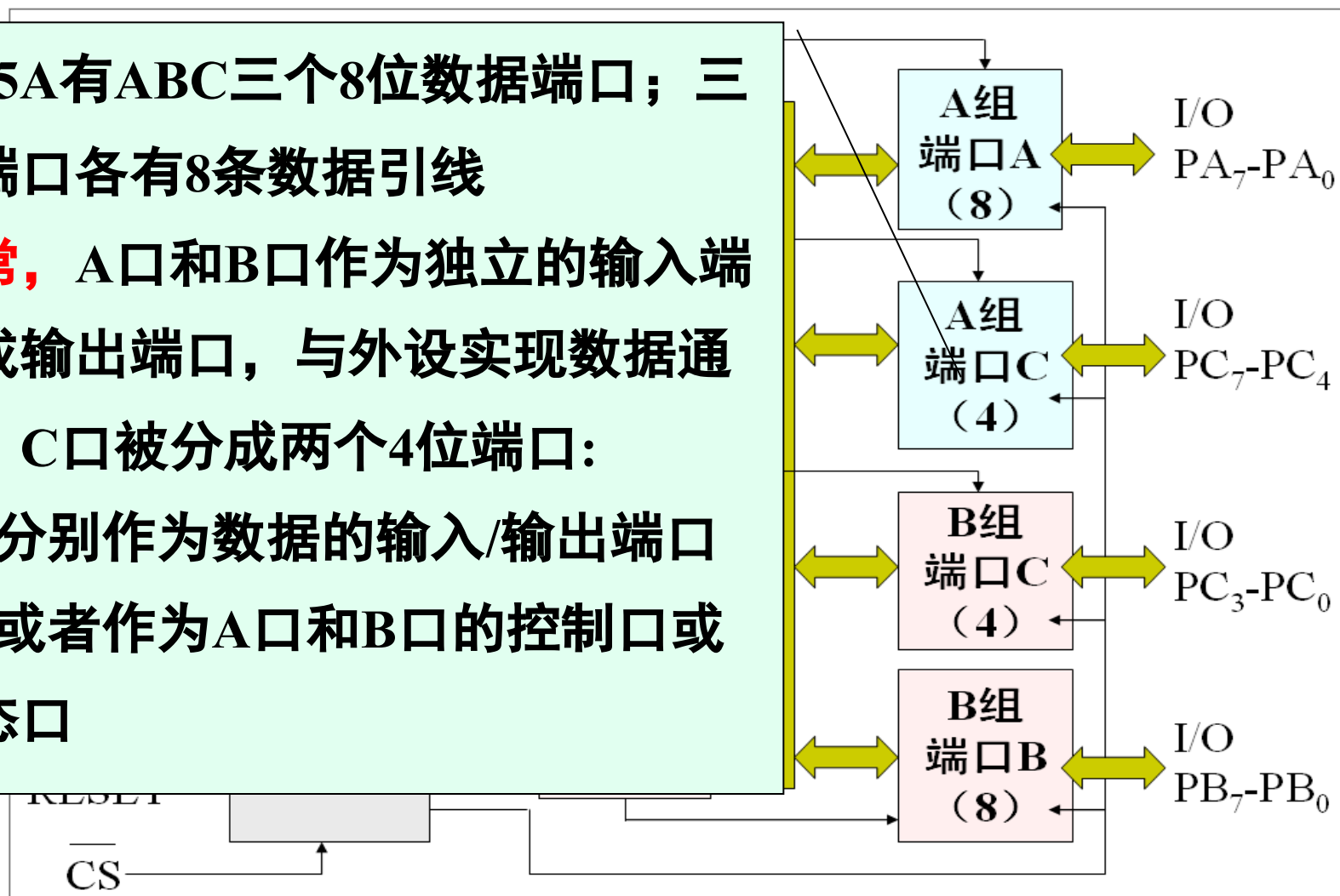
- **可编程**通用并行输入/输出接口电路
 - 通用性强，使用灵活，通过它CPU可直接与外设相连，可实现数据的**无条件、查询、中断**等I/O传送方式
 - 具有40个引脚
 - 具有三个相互独立的输入/输出数据端口
 - A端口、B端口、C端口
 - 三个端口可单独使用,也可组合使用；可传送数据信号,也可传送状态信号和控制信号
 - **三个端口各 有多种工作方式，可编程选择**
- 

8255A的内部结构

8255A有ABC三个8位数据端口；三个端口各有8条数据引线

通常，A口和B口作为独立的输入端口或输出端口，与外设实现数据通信；C口被分成两个4位端口：

- 分别作为数据的输入/输出端口
- 或者作为A口和B口的控制口或状态口



读/写控制逻辑电路中的引脚信号

8255A的片内地址信号 A_1 、 A_0 和 \overline{CS} 以及控制信号 \overline{RD} 、 \overline{WR} 、RESET的**逻辑组合**，形成相应的操作命令，发送到A口、B口、C口，以控制并实现数据和状态信息的正常传输

写选通信号，有效时，CPU
可以把数据或控制字写到
8255A

8255A
所有
个数

端口片内地址信号；用于寻址
8255A的三个数据端口和一个控制字寄存器(也称为**控制口**)：
A口地址为**00**(也称8255的基地址)；B口地址为**01**；C口地址为**10**；控制口地址为**11**



8255A的信号组合与基本操作

\overline{CS}	\overline{RD}	\overline{WR}	A_0	A_1	操 作	数据传送方向
0	0	1	0	0	读A口	A口→数据总线
0	0	1	0	1	读B口	B口→数据总线
0	0	1	1	0	读C口	C口→数据总线
0	0	1	1	1	无操作	$D_0 \sim D_7$ 为三态
0	1	0	0	0	写A口	数据总线→A口
0	1	0	0	1	写B口	数据总线→B口
0	1	0	1	0	写C口	数据总线→C口
0	1	0	1	1	写控制口	数据总线→控制口
0	1	1	×	×	无操作	$D_0 \sim D_7$ 为三态
1	×	×	×	×	禁止	$D_0 \sim D_7$ 为三态



8255A的外部引脚

8255A是40根引脚，双列直插式引脚可分成：

(1) 与外设连接的引脚

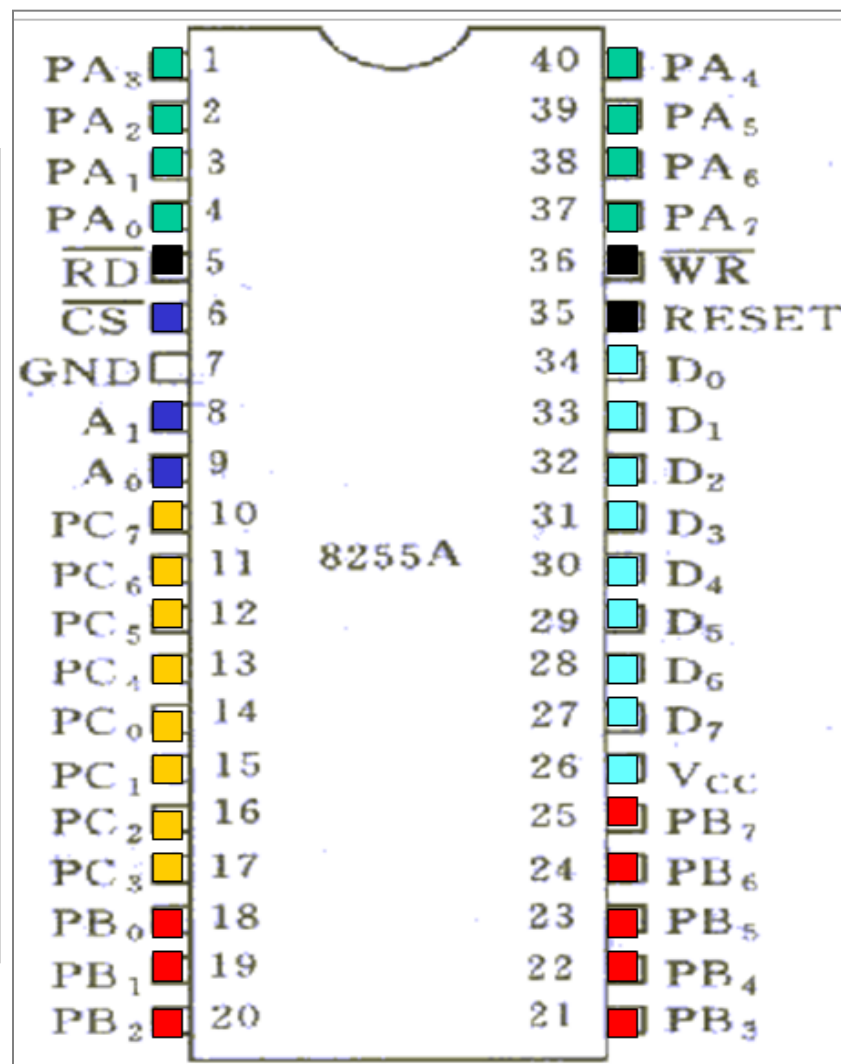
PA₀~PA₇：A口数据线

PB₀~PB₇：B口数据线

PC₀~PC₇：C口数据线

(2) 与CPU连接的引脚

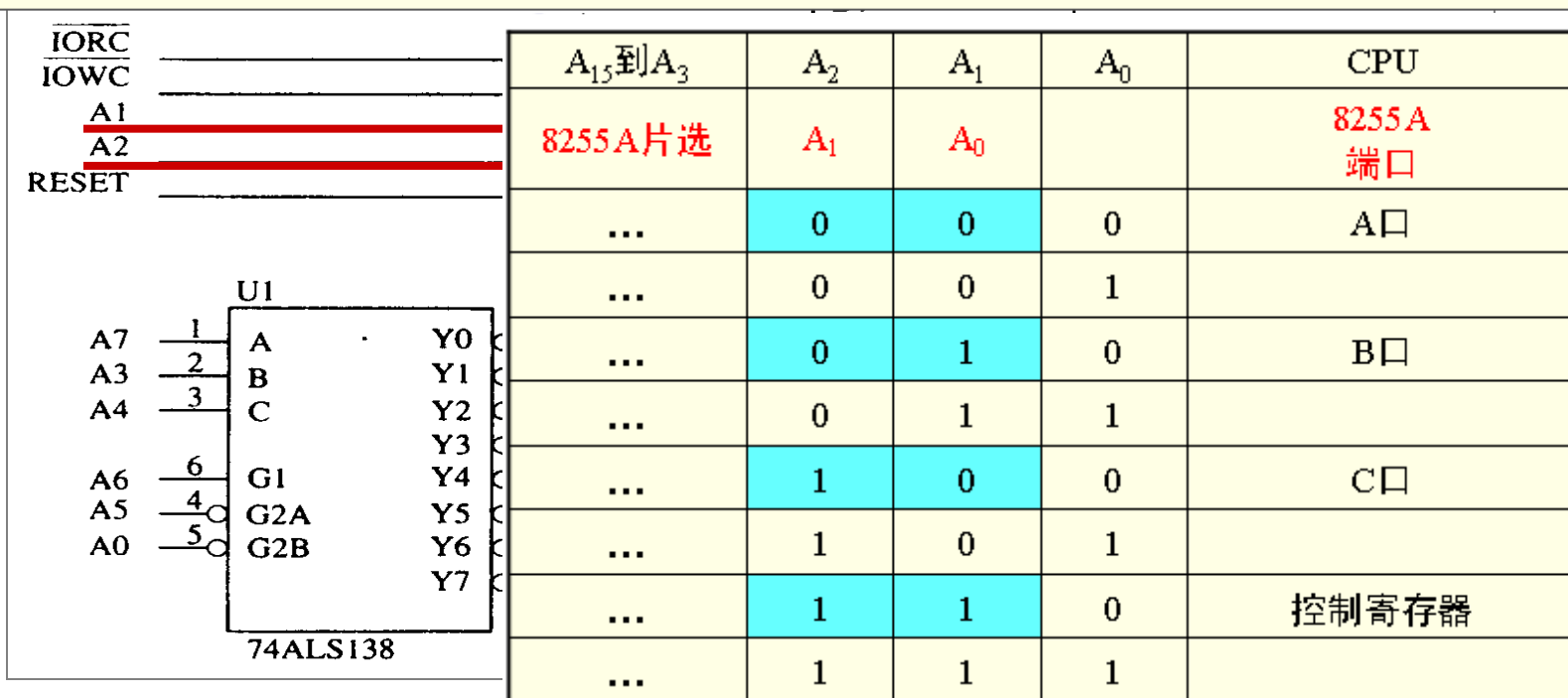
D₇~D₀、A₀、A₁、 $\overline{\text{CS}}$ 、 $\overline{\text{RD}}$ 、 $\overline{\text{WR}}$ 、RESET



8255A与CPU地址线的连接问题

如8255A数据线与8086相连,则要求必须与8086CPU低8位数据线相连; CPU则需使用偶地址访问

此时,则需要将8255A的A₀和A₁分别与CPU的A₁和A₂相连,即CPU需要使用连续的四个偶数地址寻址8255四个端口





8255A的编程与控制字

8255A可通过控制字编程；控制字分为**两类**：

均写入控制口（ $A_0=1$ 、 $A_1=1$ ）

一类是各数据端口的**工作方式选择控制字**

特征位 $D_7=1$;放在程序的开始部分

一类是**C口的位控(置位/复位)控制字**

特征位 $D_7=0$;放在初始化程序以后的任何地方





工作方式及其端口约定

- 8255A有三种基本的工作方式，即：
 - **方式0** 基本的输入/输出方式----A口、B口和C口
 - **方式1** 选通的输入/输出方式----A口和B口
 - **方式2** 双向传输方式-----只能是A口
- 在方式1或2时，C口通常配合A口或B口工作，为A口、B口的数据传输提供**对外设的控制信号和外设的状态信号**；而剩余的位仍可以在方式0下工作





工作方式选择控制字

▲ 方式选择控制字用于设置各数据端口的工作方式

D ₇ =1	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
方式标志位	A口方式选择 00=方式0 01=方式1 1× =方式2		A口I/O 0=输出 1=输入	C口高4位I/O 0=输出 1=输入	B口方式选择 0=方式0 1=方式1	B口I/O 0=输出 1=输入	C口低4位I/O 0=输出 1=输入





工作方式选择控制字的使用例

▲例，一个8088系统通过一个8255A芯片与外设交换信息；经片选后8255的基地址为 **234H**，即四个端口地址分别为：

A口——234H B口——235H

C口——236H 控制字寄存器——237H

▲如果设定8255A的各端口处于如下工作方式：

A口 方式0，输入

B口 方式1，输出

PC₇-PC₄ 输出， PC₃-PC₀ 输入

工作方式控制字应为： **10010001B=95H**

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	0	1	0	1





8255A初始化问题

▲8255A的初始化过程就是使用输出指令，将初始化控制字写入**控制字寄存器**的过程


具体到本例，采用如下几条典型指令实现：

MOV AL, 95H

MOV DX, 237H

OUT DX, AL

方式控制字确定后，就可以通过编程用指令把这个控制字写入8255A的控制字寄存器中，实现8255A的初始化，然后才可进行数据传送的具体操作



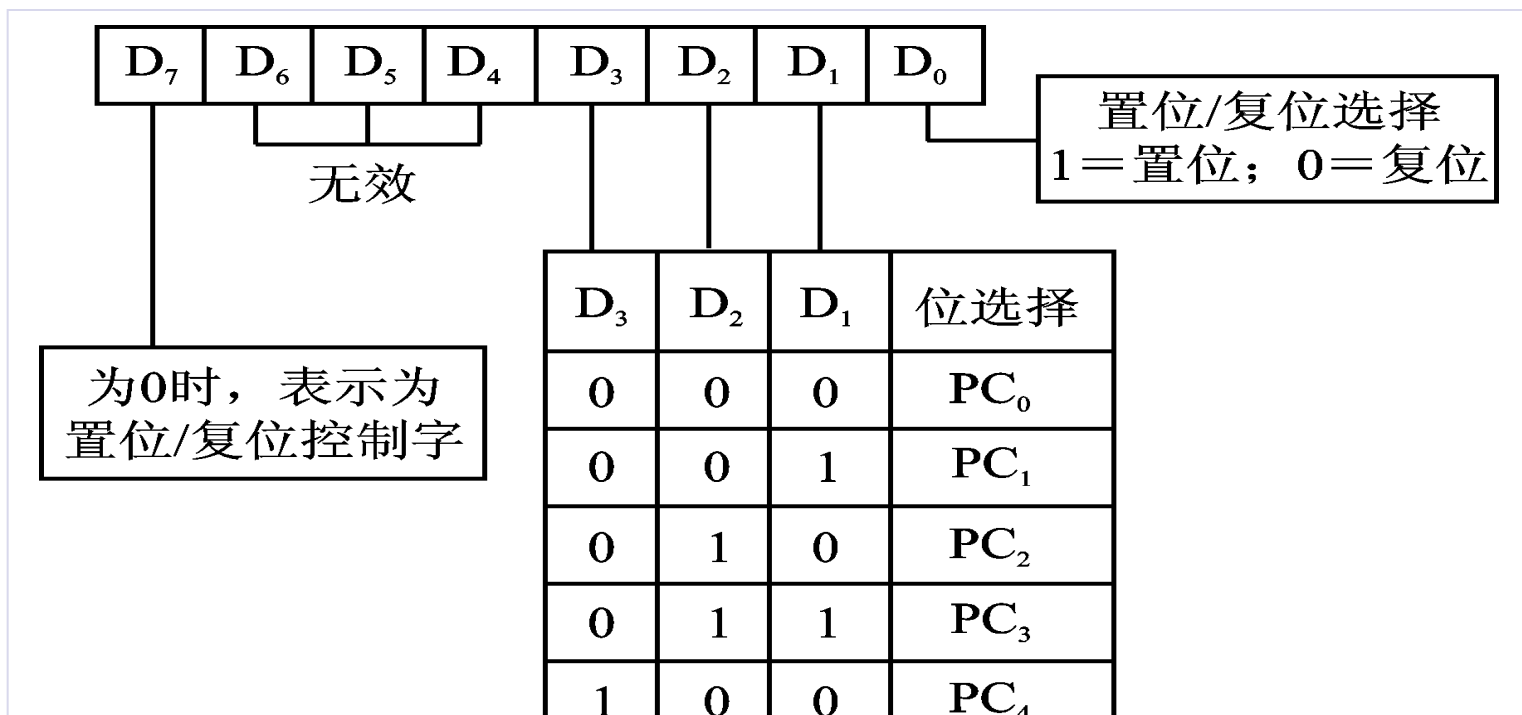


C口位置位/复位控制字

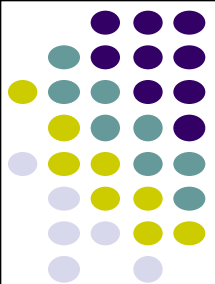
- ▲ C口作为数据口时，可以用输入、输出指令实现数据的传送
- ▲ **但**更多的情况下，C口通常用作A口和B口输入输出的控制信息位或状态信息位
- ▲ **此时，需要对“位”进行置位1或复位0操作，以表示不同的状态信息或控制信号**
- ▲ 任何一位的置位或复位，并不影响其它位的状态



C口位控控制字的格式

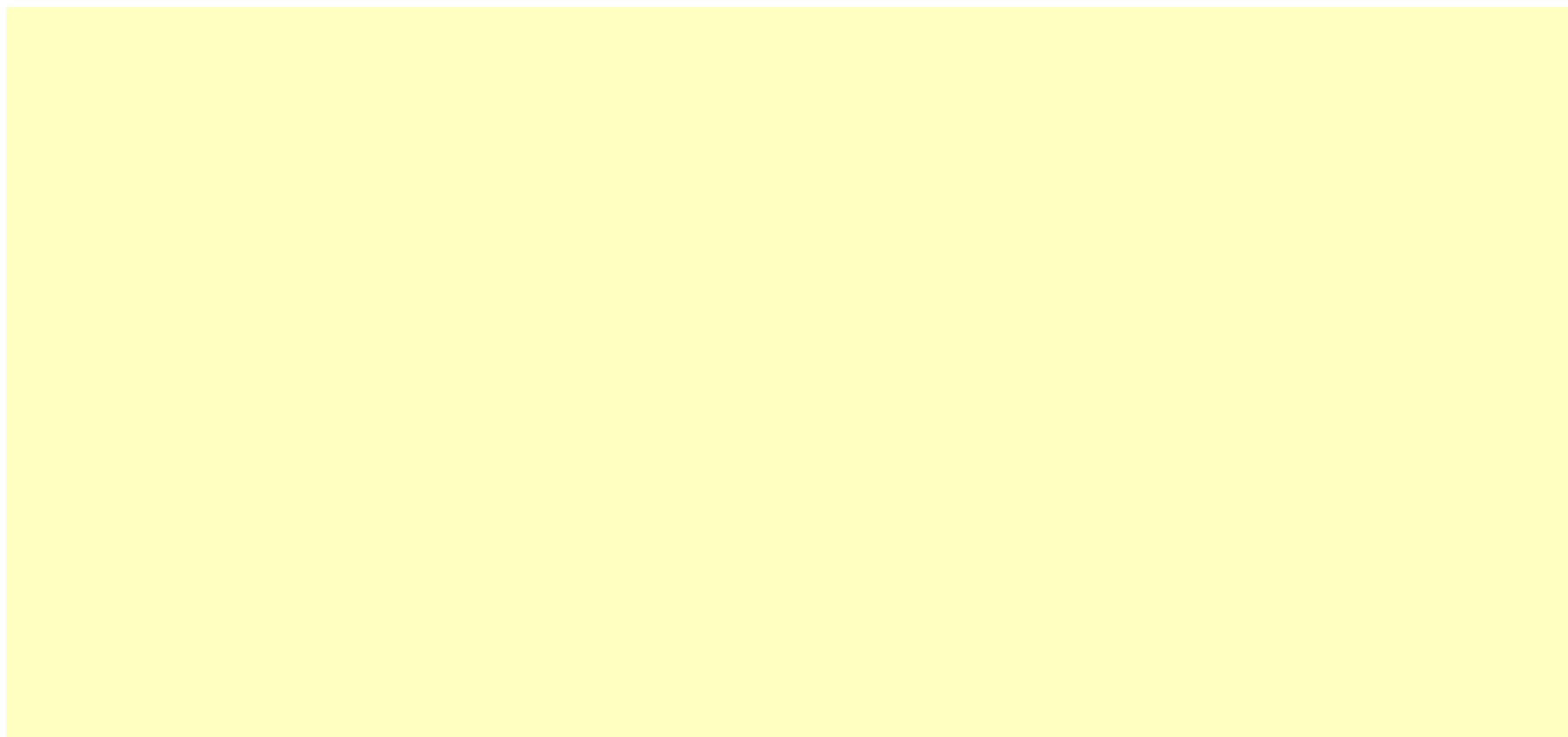


▲ 位控控制字同方式选择控制字一样，也将被送往控制字寄存器，区别在于标志位D₇的不同



C口位控控制字的例

例；如要使C口的PC₅置位，那么控制字应为多少？





C口位控控制字的例

▲ 假设8255A的控制端口地址为0F3H，要对PC₅进行置位，需要进行如下操作：

MOV AL,0BH ; 0000 1011B=0BH

OUT 0F3H,AL

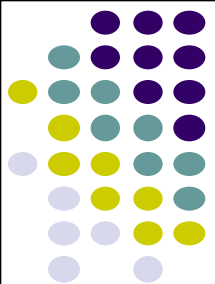
▲ 同样，若要对PC₂进行复位，需要进行如下操作：

MOV AL,04H ; 0000 0100B=04H

OUT 0F3H,AL

▲ 复位PC₂时，并不影响前面已经置位的PC₅和其他各位





8255A的工作方式

- △ 8255A共有三种工作方式;每种工作方式都有其自身的功能和工作特点
- △ 可通过编程，确定8255A的工作方式





工作方式0

▲ 方式0-----基本输入输出方式

▲ 特点：参加数据传输的有A和B两个8位端口和C的两个4位端口；均可独立地由方式选择控制字规定为输入口或输出口

▲ 均不需要固定的**联络信号**，传输的数据只需要通过选定的端口，直接地写入或读取即可



IO8255BASE EQU 200H

IO8255 CODE SEGMENT

IO8255 ASSUME CS:CODE

IO8255 START MOV DX,IO8255K

MOV AL,8BH

OUT DX,AL

L1: MOV DX,IO8255C

IN AL,DX

MOV DX,IO8255A

OUT DX,AL

状态，并根

MOV AH,01H

INT 16H

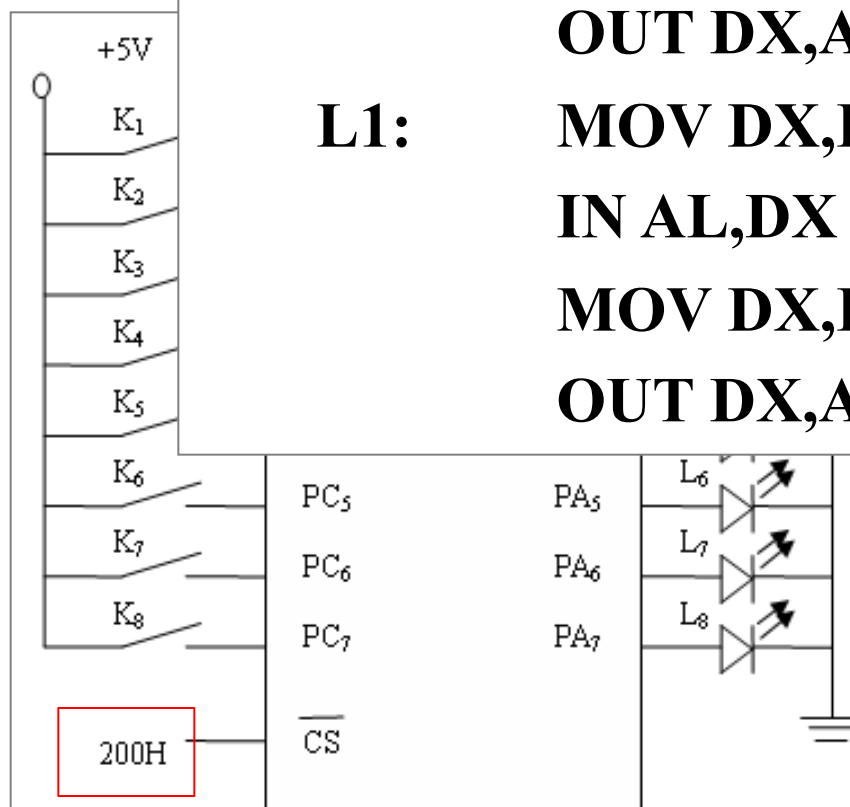
JZ L1

MOV AH,4CH

INT 21H

CODE ENDS

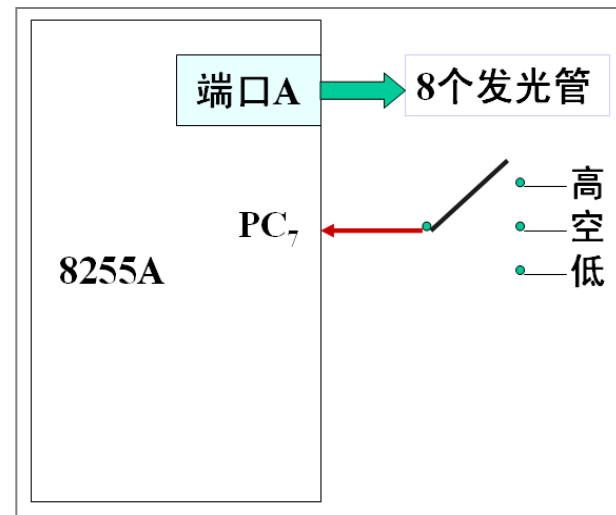
END START



A口的地址称为芯片的基地址

方式0的典型举例

- A口接8个发光二极管，PC₇连接一个开关
 - 开关接通高电平时，8管全亮(点亮时为1)
 - 开关接通低电平时，8管循环点亮
 - 假设端口地址为 60H—63H



- 问题分析
 - 二极管和开关属简单外设，永远就绪，无需关心其状态如何
 - A口可选择工作方式0，设置为输出口
 - C口的上半部为输入口（用于输入开关的高或低电平数据）
 - B口和C口的下半部不用 (方式字位均可取0)
 - 方式字为：1000 1000=88H



方式0的典型应用举例程序代码

```
MAIN: MOV AL, 88H      ; 8255初始化
      OUT 63H, AL      ; 方式字写入控制字寄存器
NEXT: IN AL, 62H        ; 读C口
      TEST AL, 80H     ; 测PC7
      JNZ LIGHT        ; PC7=1, 转全亮
      MOV BL, 01       ; PC7=0, 设置循环点亮的起始位
CONT: MOV AL, BL
      OUT 60H, AL      ; 写A口, 点亮第一个管
      CALL DIS         ; 延时子程序
      SHL BL, 1        ; 左移一位, 准备点亮下一个管
      JNZ CONT         ; BL不为0时, 转去点亮下一个管
      JMP NEXT         ; BL=0, 循环亮结束
LIGHT: MOV AL, 0FFH    ; 设置全亮数据位
      OUT 60H, AL      ; 写A口, 点亮全部管
      JMP NEXT
```



方式0典型示例

程序段:

```
MOV AL, 10010000B
```

```
MOV DX, 038BH
```

```
OUT DX, AL
```

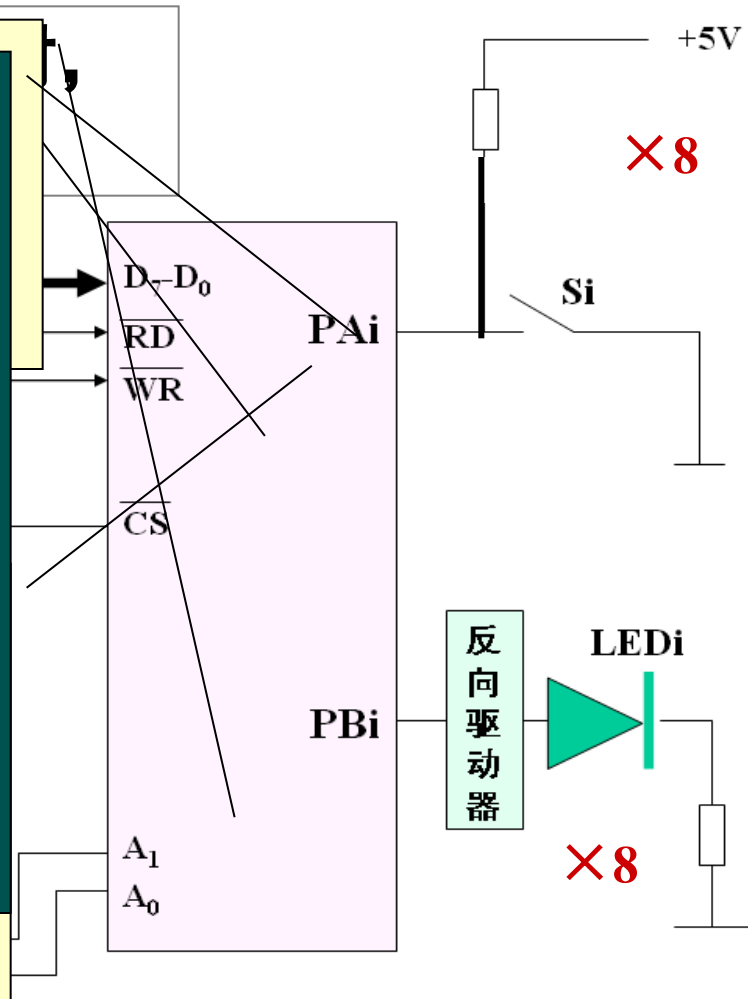
```
MOV DX, 0388H
```

```
IN AL, DX
```

```
MOV DX, 0389H
```

```
OUT DX, AL
```

无需再取反





方式0的应用特点

- 方式0的典型特点是：不提供也不需要提供**专门**的状态信息和控制信息的端口数据线，各端口均可独立使用
- 方式0的典型应用是：**无条件传送方式**下的数据传送；直接执行IN指令或OUT指令
- **也适用于**查询传送方式下的数据传送
 - 首先查询外设是否“就绪”
 - 可选择C口的某些位，作为外设的状态信息位和对外设的选通控制位(但**不确定专用位，可任意选择**)
 - 当检测“就绪”后，再执行IN指令或OUT指令



方式0用于查询方式传送的图示

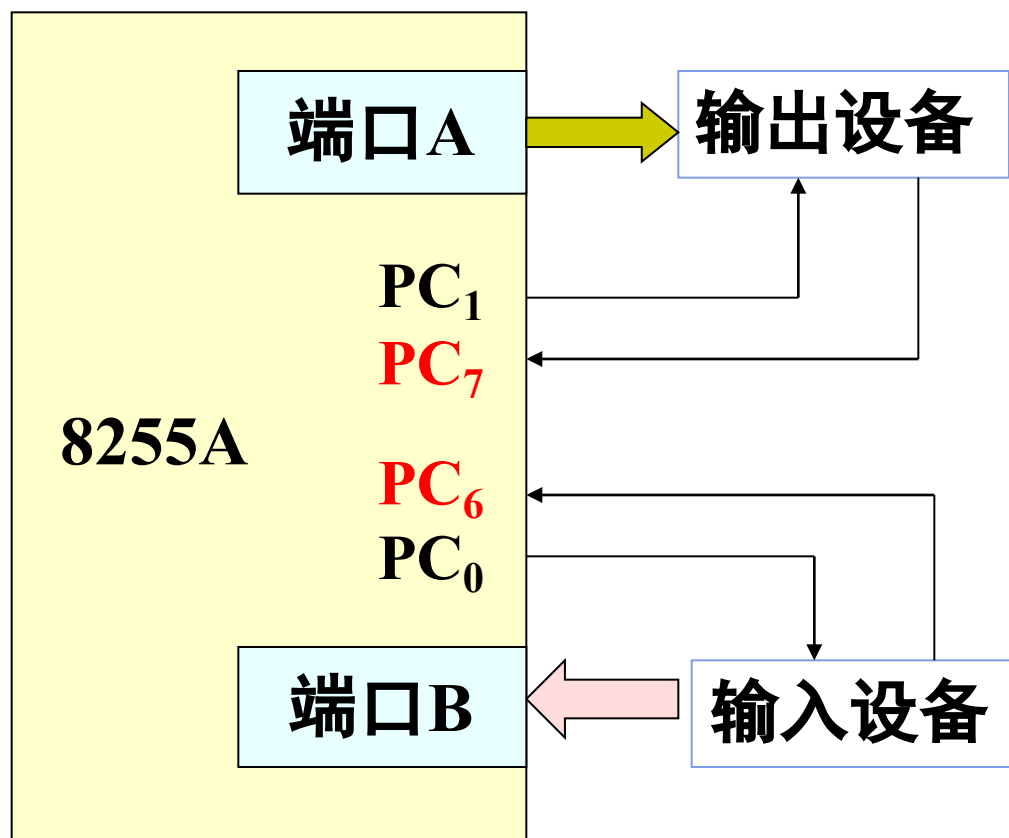
工作方式字为：

1	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---

首先从C口读取**状态信息字**，测试PC₇或PC₆

“**就绪**”后，用PC₁或PC₀选通外设，发送**控制字**

从A口输出；从B口输入



控制信息字

×	×	×	×	×	×	1/0	1/0
---	---	---	---	---	---	-----	-----

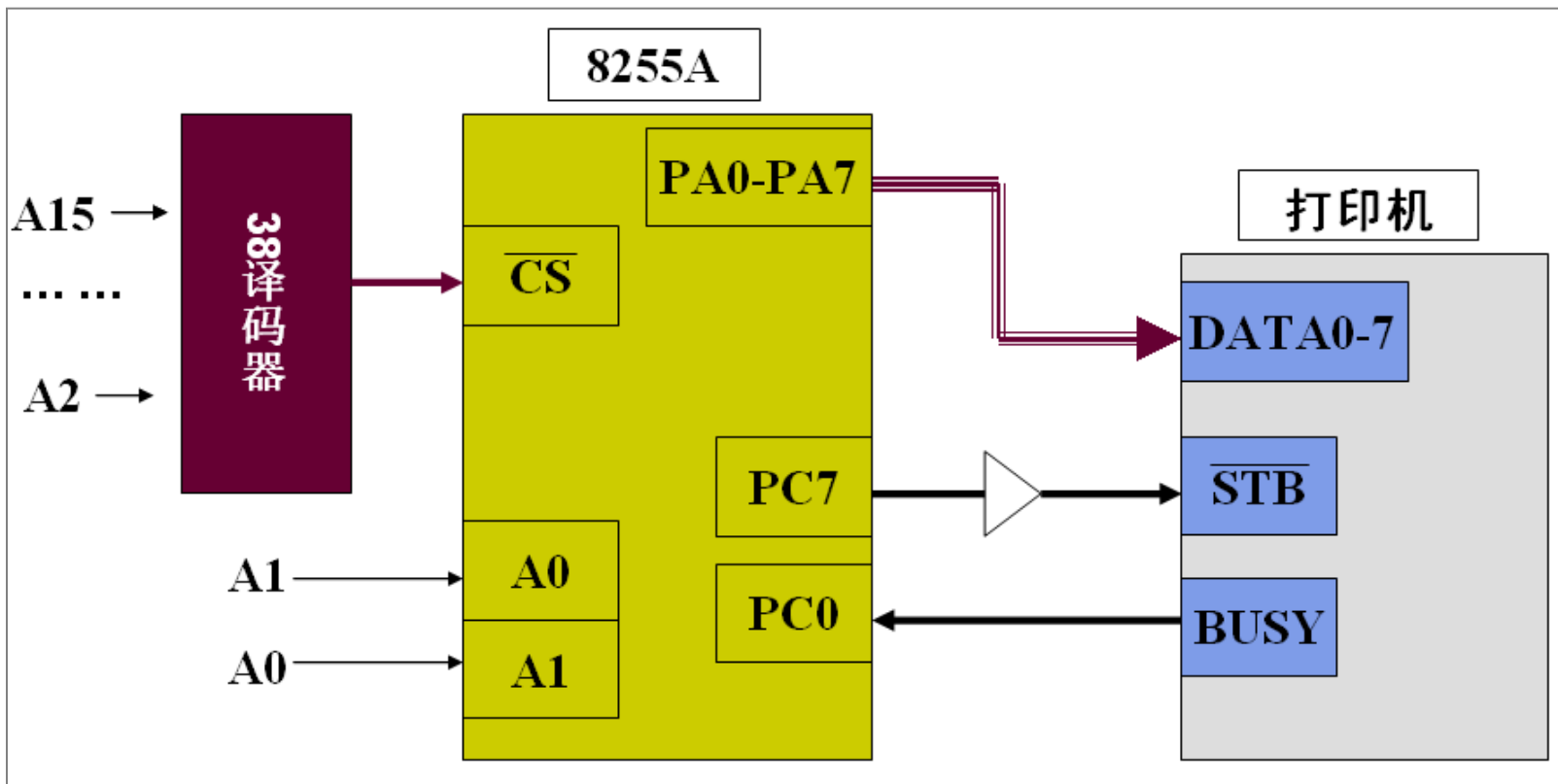
方式0用于查询方式传送的例

按
(1)
表
(2)
开
(3)
BU
(4)
示

A-PORT
B-PORT
C-PORT
CTRL-PORT
MOV
MOV
OUT
MOV
OUT
MOV
MOV

```
LOOP:  MOV DX,C-PORT          ;PC端口地址
        IN AL, DX              ;PC1 =0 ? ( ACK / BUSY =0 ? )
        AND AL,02H
        JNZ LOOP
        MOV DX,A-PORT          ;PA口地址
        MOV AL,[SI]            ;取数据
        OUT DX,AL              ;数据送PA口
        MOV DX, CTRL-PORT      ;8255A控制端口
        MOV AL,00001100B       ;PC6 置低(STB =0)
        OUT DX,AL
        NOP                    ; 延时约1微秒,
        NOP                    ; ACK / BUSY =1, 打印机读数据
        NOP                    ; 读取完毕将回送ACK / BUSY =0信号
        MOV AL,0001101B        ;PC6 置高(STB =1)
        OUT DX,AL
        INC SI                  ;地址加1
        DEC CX                  ;字符个数减1
        JNZ LOOP               ;未传送完, 继续
```

综合练习题

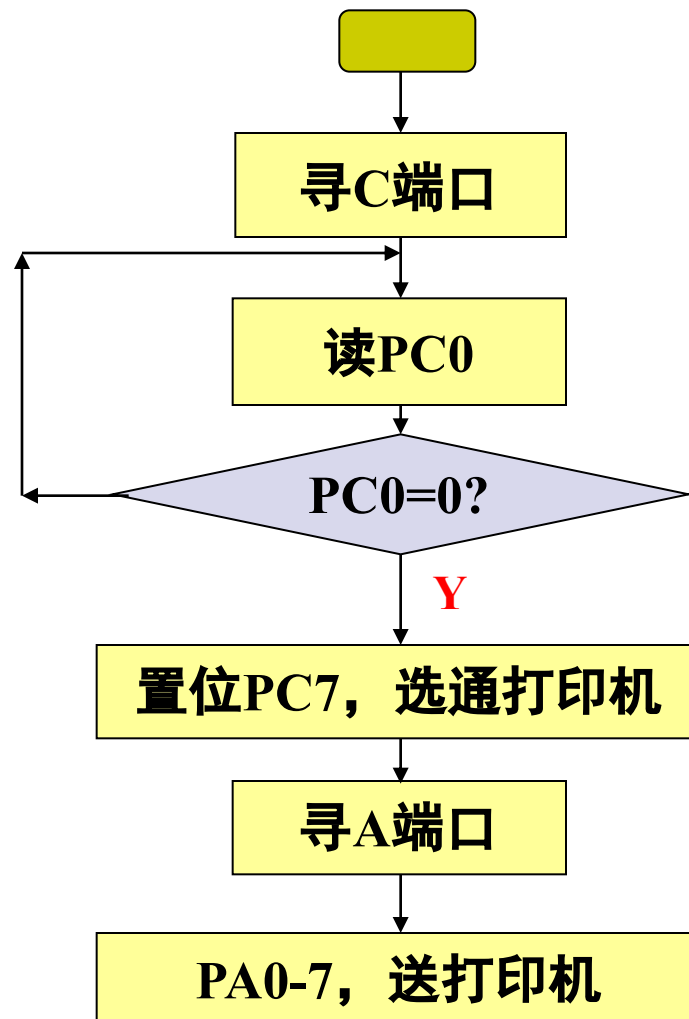


8255A与打印机的连接如图示

假设当A15-A2的取值为**0010 0100 1100 01**时，选中该8255A

综合练习题

- **MOV DX,---**
- **LOOP IN AL,DX**
- **TEST AL,01**
- **JNZ LOOP**
- **MOV AL, 0FH**
- **MOV DX,---**
- **OUT DX,AL**
- **MOV DX,---**
- **MOV AL,BUF**
- **OUT DX,AL**





工作方式1

▲ 方式1也称为**选通输入/输出方式**

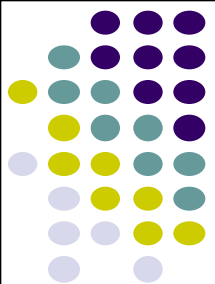
▲ A口和B口可以工作在方式1

▲ A口和B口都可以分别作为数据端口传送数据

▲ A口和B口都可以作为输入口或输出口

▲ 方式1工作的端口可以是两个，也可以只设一个；其余的输入输出端口，还可以工作在其它方式下





工作方式1

- 方式1的典型特点是：
 - **固定**了接口与外设的联络信号线； A、B两端口分别被固定了C口的 **3位联络线**
 - AB两端口**接口组态是固定的**，用户不能更改
- 方式1的典型应用是：中断方式的数据传送，也适用于查询方式下的数据传送



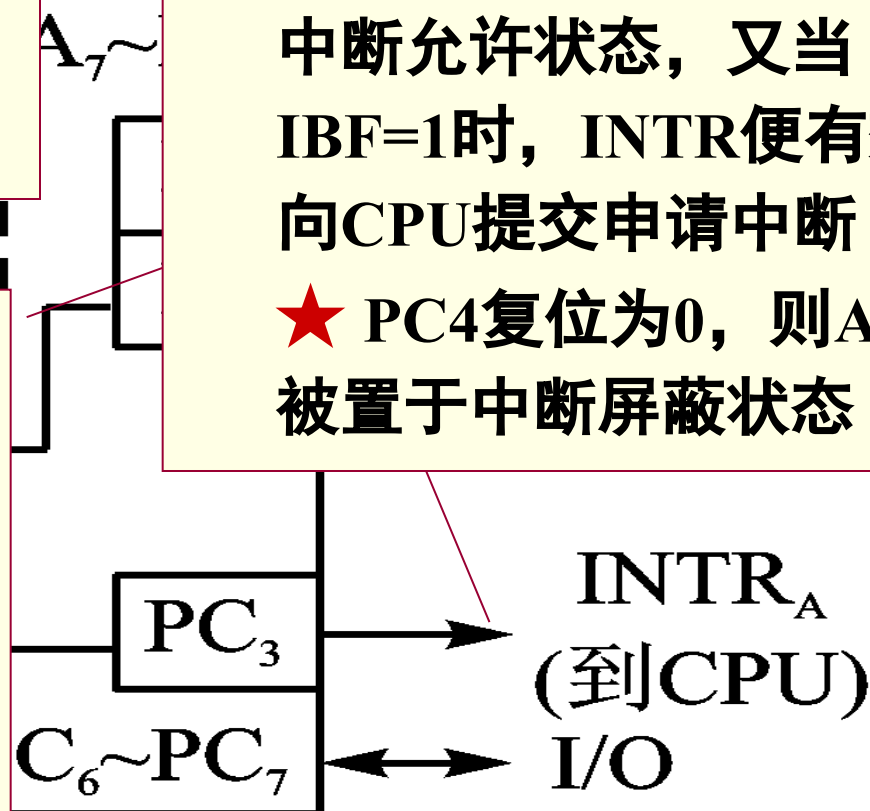
INTR_A 是 8255A 送往 CPU 的中断请求信号，高电平有效。当端口数据就绪后，如果中断允许，8255A 将把 INTR_A 送出请求 CPU 的中断服务；A 口的中断请求从 PC_3 送出

STB 是外设数据输入选通信号，低电平有效；由外设送给 8255A；STB 与 PC_4 相连；有效时，外设“通知”端口并开始送入数据到数据缓冲器

INTE_A 是对中断请求屏蔽或允许的内部控制信号；**只能**通过对 PC_4 的置位/复位实现控制

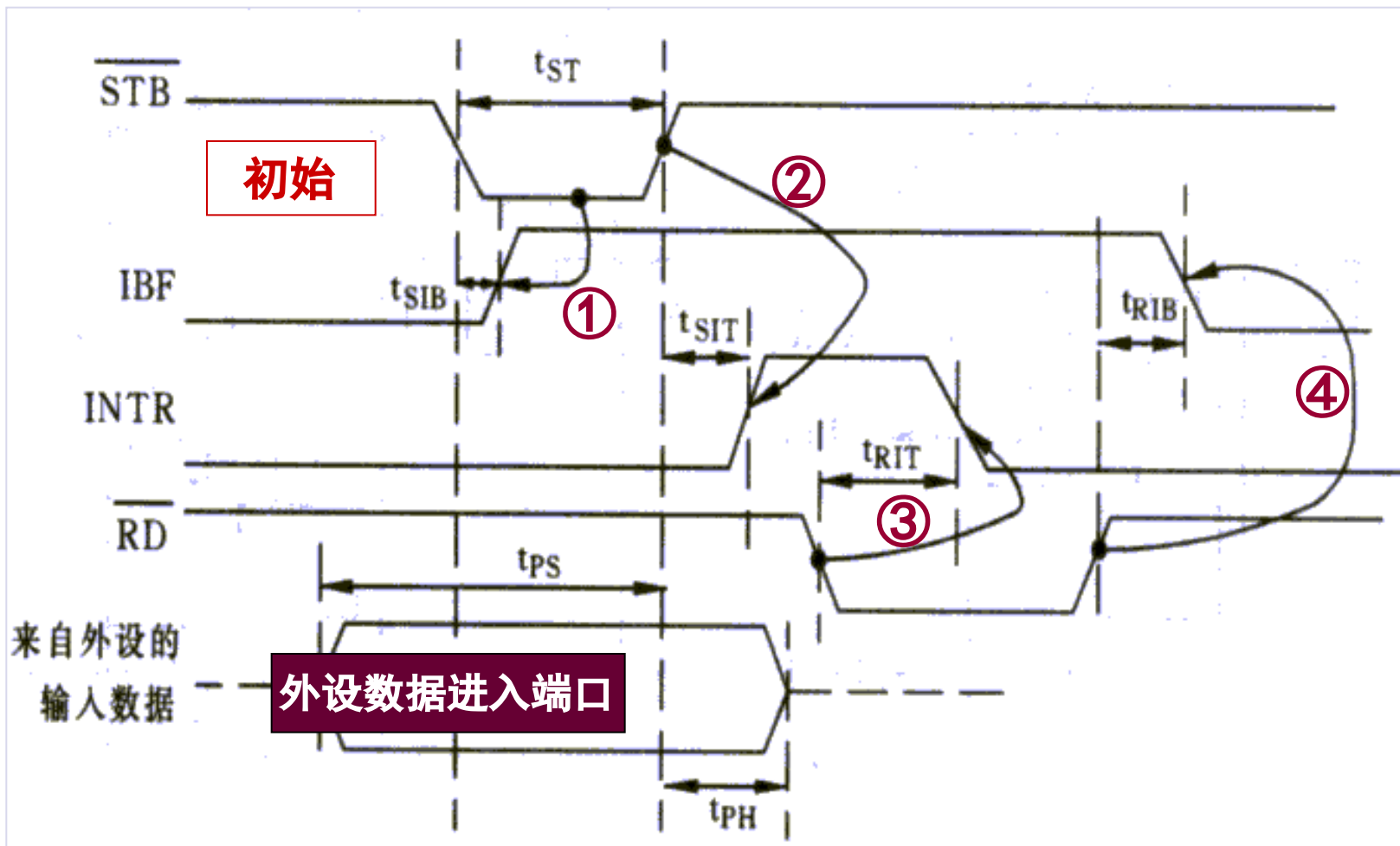
★ PC_4 置位为 1，A 口处于中断允许状态，又当 $\text{IBF}=1$ 时， INTR 便有效，向 CPU 提交申请中断

★ PC_4 复位为 0，则 A 口被置于中断屏蔽状态



初始时, STB=1(未开始传), 将IBF
IBF=0(已空), RD=1(未选通), 可进
读), INTR=0(无请求)

及时序



工作在方式1时输入B端口的信号

注意与A口的异同

基本信号交换和工作原理相似

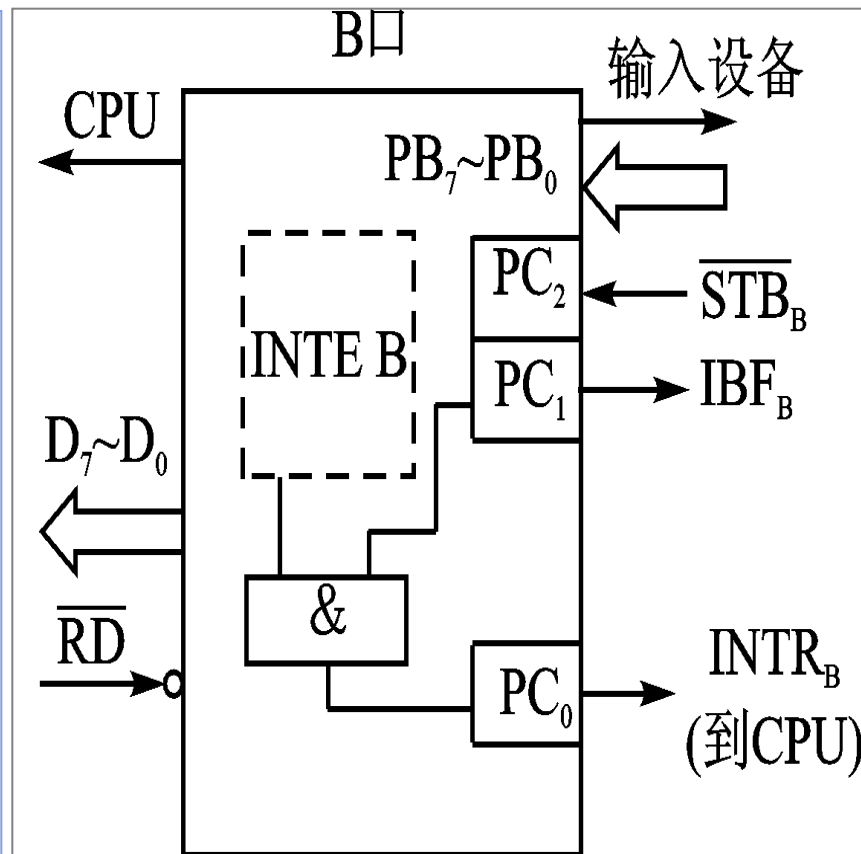
C口各联络位不同

B口的STB信号与PC₂相连

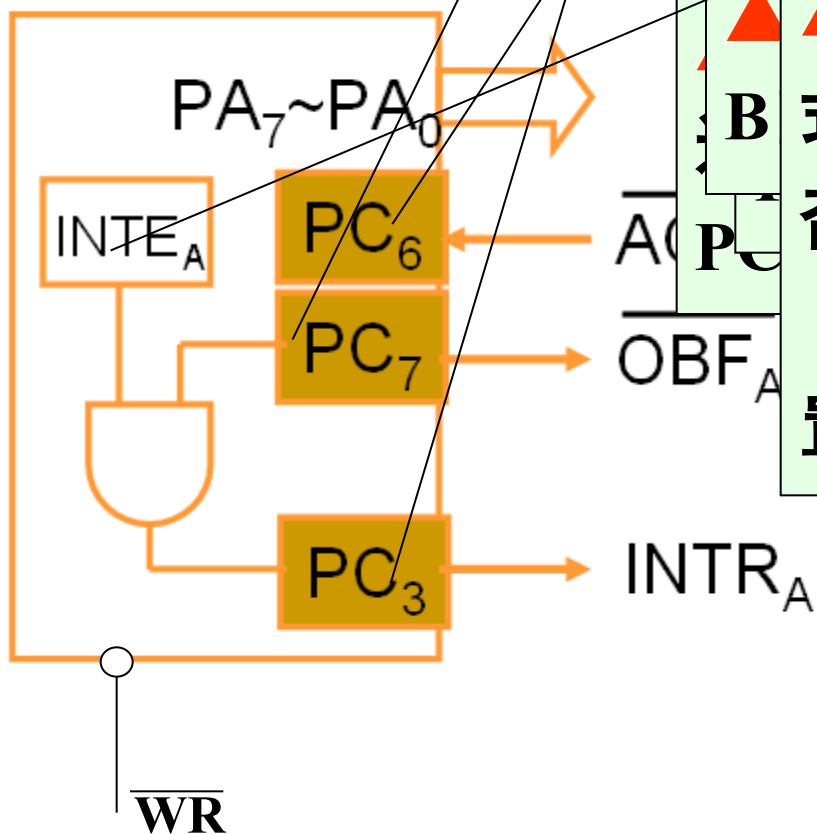
B口的IBF信号由PC₁输出

B口的中断请求由PC₀送出

对PC₂的置位/复位指令，实现对中断屏蔽或允许的控制



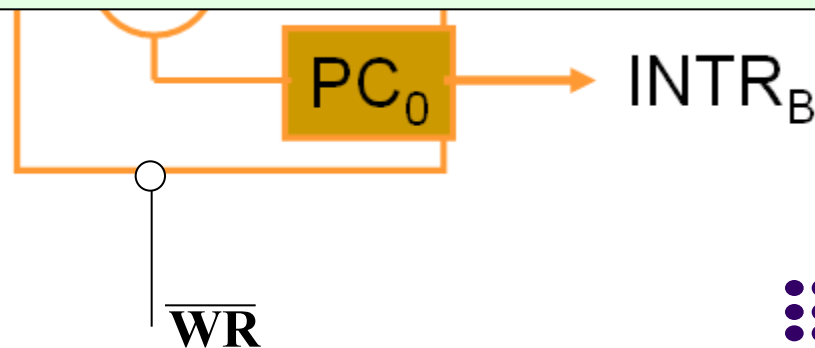
工作在方式



▲ $INTR$ 为中断请求信号，高电平有效。当外设取走数据后，端口已“空”，发出中断请求信号，请求CPU继续下一次送来数据

▲ $INTE_A$ 和 $INTE_B$ 的意义与方式1的输入情况时相似，决定是否屏蔽中断请求

▲ 分别(只能)通过 PC_6 和 PC_2 的置位/复位实现





方式1下的控制字设置例

要求： 将A口设置为方式1输入；PC₇和PC₆作为输出；

B口设置为方式1输出；A口不允许中断，B口允许

MOV DX, 0263H ; 控制字寄存器端口号为0263H

MOV AL, 0B4H ; 方式控制字

OUT DX, AL

MOV AL, 08H ; 设置A口不允许中断

OUT DX, AL

MOV AL, 05H ; 设置B口允许中断

OUT DX, AL



方式1下数据的中断传送方式

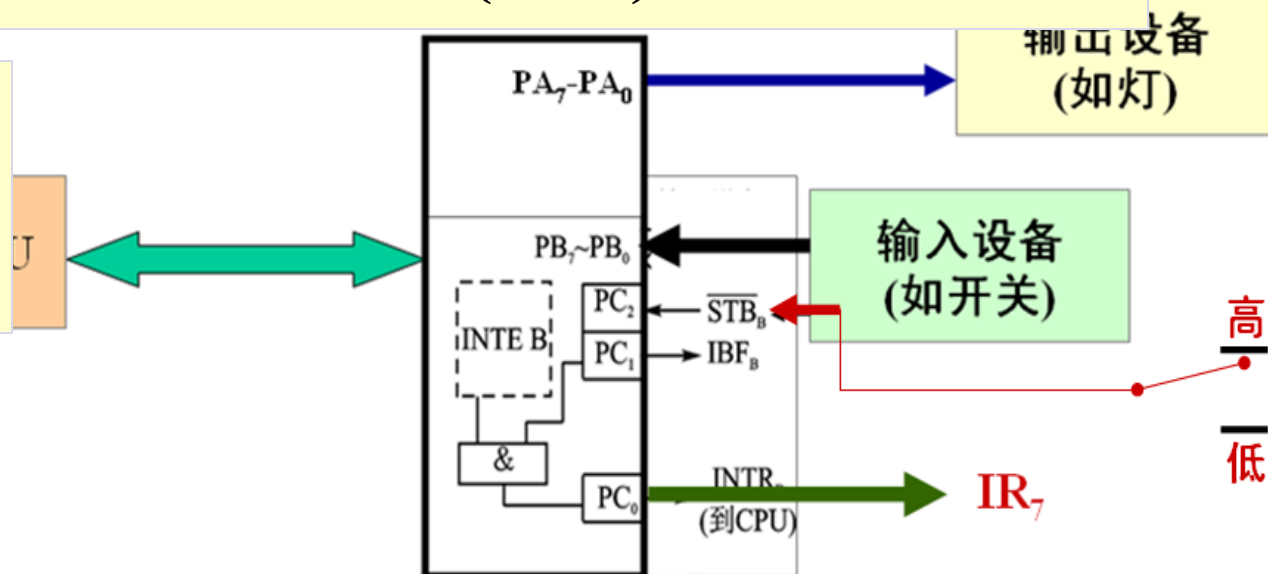
- 方式1的最典型应用是用于**中断传送方式**
 - 通过**A**口的**PC₆**和**B**口的**PC₂**的置位/复位控制字使相应的 $INTE=1$ ，即允许内部中断
 - 通常由**PC₃**和**PC₀**送出的**INTR**信号接到8259A的某个**IR_i**引脚



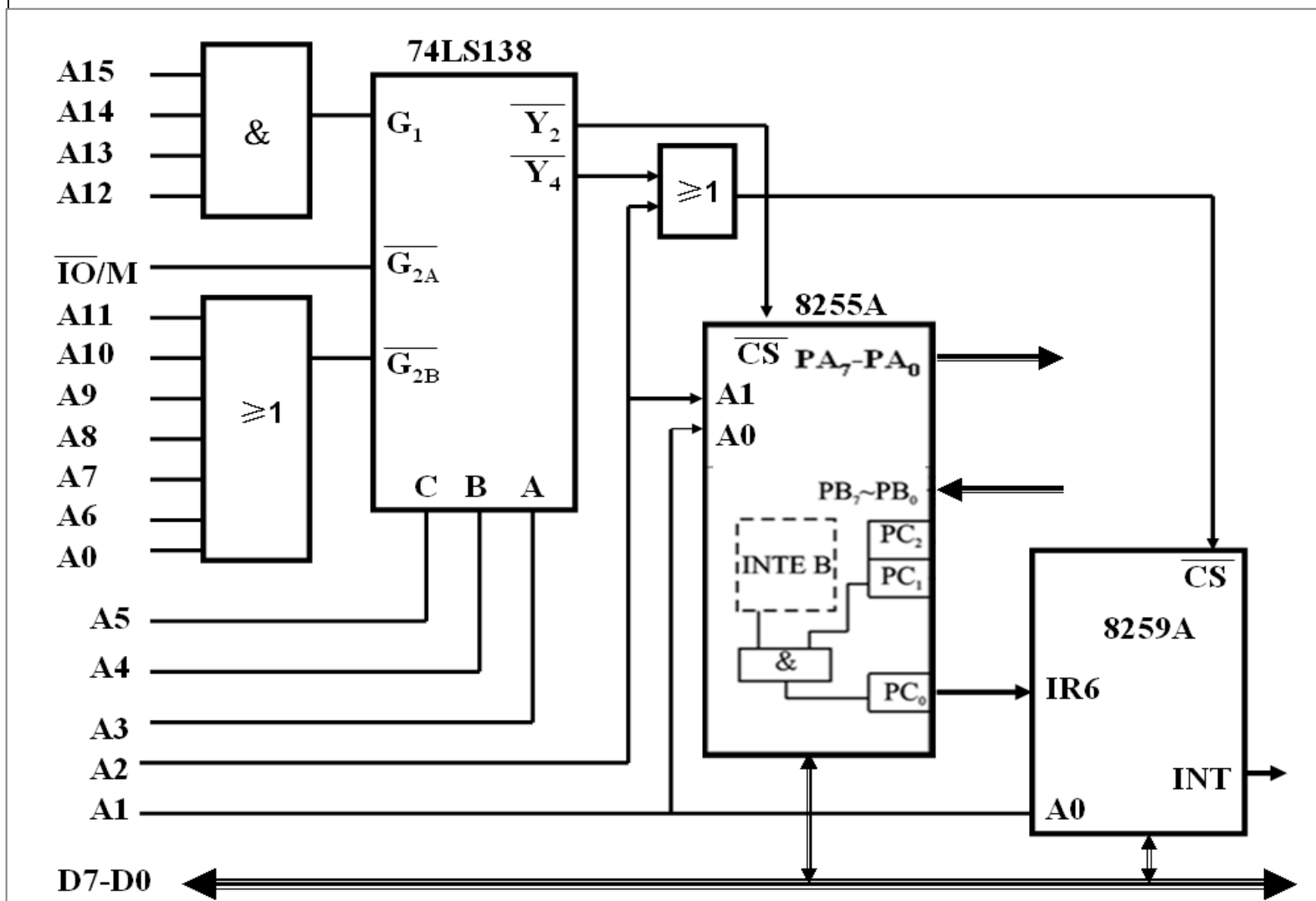
实验操作过程 (将STB引脚接一个开关)

- 初始时, $STB = 1$; $IBF = 0$
- 使 $STB = 0$, 开关闭合的状态数据将进入端口(使 $IBF = 1$)
- 再使 $STB = 1$, 恢复初始状态
- $IBF = 1$ 后, 8255A将向8259A发送中断请求 $INTR$
- 8259A判优后, 将送CPU; CPU响应后读取并输出数据; 之后 $IBF = 0$, 恢复初始状态
- CPU无操作时, 处于暂停状态(HLT)

A1: STI
HLT
JMP A1



综合练习题





综合练习题

- 通过8255A的B口接收外设数据；数据就绪即通过8259A向8086申请中断；CPU响应后，执行中断服务程序，由中断服务程序读取数据；进行左移一位处理后，通过A口输出到输出设备
 - 假设A口工作在方式0，输出；B口工作在方式1，输入；允许中断；送出的中断请求信号连接到8259A的IR6引脚
 - 假设8259A为单片使用，上升沿触发，非缓冲连接，非自动中断结束，采用普通嵌套；同时假设中断服务程序的中断向量地址指针为00198H
-
-



综合练习题

- A-PORT EQU 0F01
- B-PORT EQU 0F02
- C-PORT EQU 0F03
- CTRL-PORT EQU 0F04
- PORT_0 EQU 0F05
- PORT_1 EQU 0F06
- INTR_IVADD EQU 0F07

中断服务序段

```
IRQ6: MOV DX, B-PORT
      IN AL, DX
```

```
      STI
      MOV DX, A-PORT
      OUT DX, AL
      JMP A1
```

```
      MOV DX, PORT_0
      MOV AL, 20H
      OUT DX, AL
      IRET
```

方式1下数据的查询传送方式

- 方式1，也可以用于查询传送方式
 - 8255A提供一个供CPU查询的方式1**状态字**
 - CPU通过**C口读取该状态字**，检测判断相应的状态标志位
 - 确认就绪后，启动数据传送
 - 输入**时，首先查询 IBF，1为就绪(端口数据区满)
 - 输出**时，首先查询 $\overline{\text{OBF}}$ ，1为端口数据区“空”，空闲

PC ₇	PC ₆	PC ₅	PC ₄	PC ₃	PC ₂	PC ₁	PC ₀
I/O	I/O	IBF _A	INTE _A	INTR _A	INTE _B	IBF _B	INTR _B
$\overline{\text{OBF}}_A$	INTE _A	I/O	I/O	INTR _A	INTE _B	$\overline{\text{OBF}}_B$	INTR _B

方式1下的查询传送

8255A

PA口

PB口

```
MOV DX,202H
L1: IN AL,DX
TEST AL, 0010 0000B
JZ L1
MOV DX,200H
IN AL,DX
```

;读取PC端口内容
;判断PC5是否为1

数据

```
MOV DX,202H
L1: IN AL,DX
TEST AL, 00000010B
JZ L1
MOV AL,BUF
MOV DX,201H
OUT DX,AL
```

;读取PC端口内容
;判断PC1是否为1



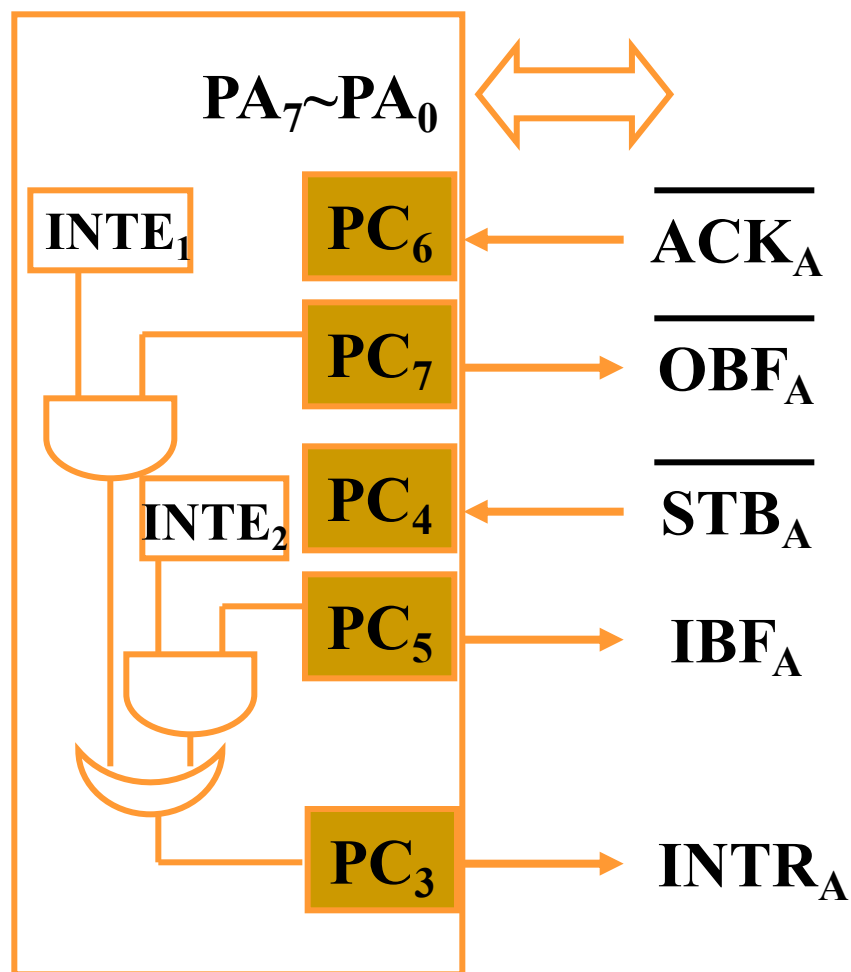


工作方式2

- ▲ 方式2也称双向传输方式，该方式只适用于A口
- ▲ 双向方式下，外设可以利用**A口的8位数据线**和CPU进行双向传送，既能发送数据，又能接收数据
- ▲ 与方式1相似，端口C也**自动地**为方式2提供相应的控制信号；即C口的5个数据位将自动配合A口作与外设之间的控制信号和状态信号
- ▲ 方式2相当于方式1下A口输入和输出的组合



方式2的控制信号及状态控制字



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

1	1	x					
---	---	---	--	--	--	--	--



方式2与其它方式的组合说明

- ◆ 当A口工作在方式2时，B口可以工作在方式0或方式1，可以作为输入端口也可以作为输出端口
 - ◆ C口由于其 PC_7-PC_3 为方式2服务，作为A口的联络控制信号，余下的只有 $PC_2\sim PC_0$ 可另作它用
 - ◆ 如B口工作于方式0下， $PC_2\sim PC_0$ 也工作于方式0，作为输入位或输出位
 - ◆ 如B口工作在方式1下， $PC_2\sim PC_0$ 恰好作为B口的联络信号，具体的规定与方式1相似
- 



例

8255A芯片的基地址为200H，PA端口用于方式2，PB
端口用于方式1输入

```
MOV DX,203H  
MOV AL,11000110B  
OUT DX,AL
```





工作方式初始化综合示例

(3) 将 A 口设置成方式2； B 口为方式0，输出；
PC0~PC2 为输入； 允许 A 口中断

```
MOV AL, 11000001B
```

```
OUT 63H, AL
```

```
MOV AL, 00001001B
```

```
OUT 63H, AL
```

```
MOV AL, 00001101B
```

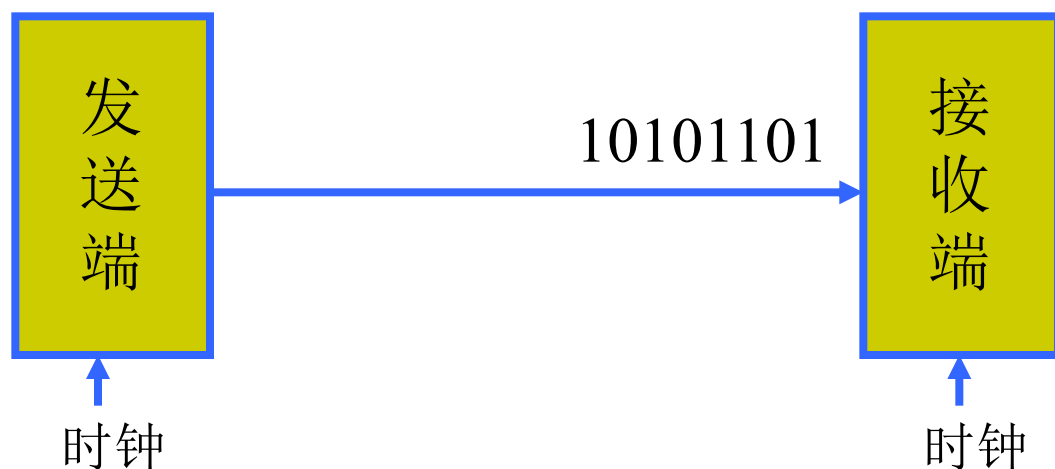
```
OUT 63H, AL
```



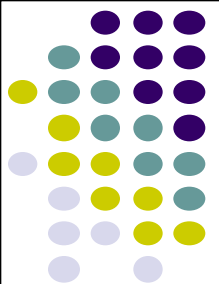
■ 串行通信

特点：每个字符的所有位是按位传送。

字符的每一位放在传输线上和采样传输线的状态，随着系统的不同而不同。



通常，一个接收端在一个很短的时间采样一个传输线的状态，观察它是1还是0，如果在从1到0的变化期间采样传输线，就可能得到不正确的结果。采样一位的理想地方是位的中间。



传送时钟

数据

接收时钟

每个时钟传送一个位数据

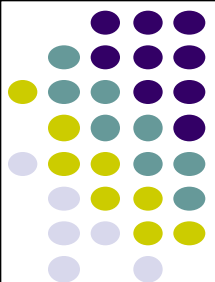
传送时钟

数据

接收时钟

每二个时钟传送一位数据





实际上，如果在发送端和接收端每边都采用一个时钟，时钟速率就可能稍有不同，尽管这种差异很小，但随着时间的持续，它们最终要失去同步，因此必须周期性地对它们进行同步。

1) 位同步

要解决的问题：

- (1) 接收端何时询问传输线，并把到达的数据位取离传输线；
- (2) 发送端应当每隔多少时间把要传送的数据位放在传输线上。

2) 字符同步

要解决的问题：

- (1) 一个字符由几位组成；
- (2) 哪一位是字符的第一位。

因而在串行通信中分为二种传送方式：

- I 异步传送方式
- II 同步传送方式

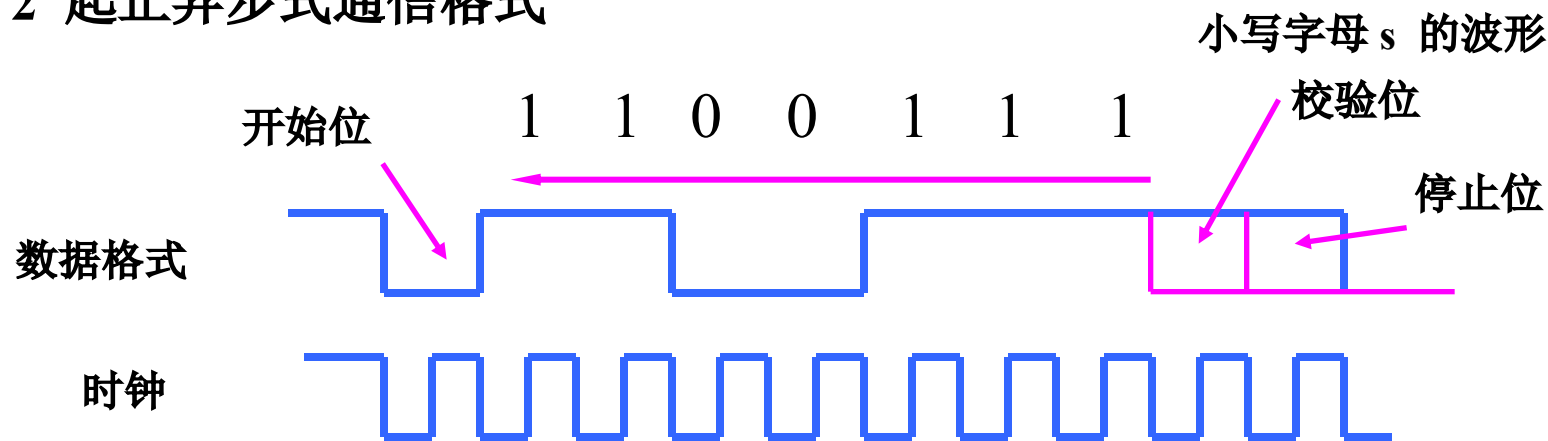


I 异步传送方式

1 特点:

- 1) 每次传送一个字符;
- 2) 在二个要传送的字符间**不必有固定的时间关系**;
- 3) 接收端为**每个字符重复建立同步信号**;
- 4) 每个字符的前面要有一个脉冲, 用于告诉接收端一个字符的开始;
- 5) 每个字符结束时传送一个停止脉冲使接收端在另一个字符传来之前稳定下来。

2 起止异步式通信格式



奇校验: 使数据中1的个数为奇数。

如: 0 1110011

偶校验: 使数据中1的个数为偶数。

如: 1 1110011



II 同步传送方式

每次传送全部数据块;

1 面向字符的同步传送方式

特点:

把若干个字符组成一个信息块-----帧, 一起传送

为了区分帧的头部和尾部, 使用ASCII码中的某些位来加以控制。

SYN	SYN	SOH	标题	STX	数据块	ETB/ETX	块校验
-----	-----	-----	----	-----	-----	---------	-----

SYN	同步字符	ASCII码16H
SOH	标题开始	ASCII码01H
STX	正文开始	ASCII码02H
ETB	一组数据结束	ASCII码17H
ETX	全部数据结束	ASCII码03H
块校验	采用CRC校验	



2 面向比特的同步传送方式

特点：

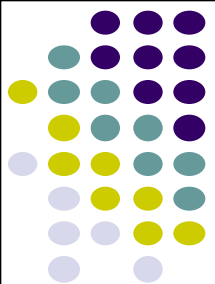
- 1) 每-帧中所传送的数据可以含有任意数量的比特位；
- 2) 为了区分帧的头部和尾部，用特定的比特模式来定义帧的开始和结束（而不是使用ASCII码特定字符）

01111110	地址	控制	数据	校验	01111110
----------	----	----	----	----	----------

地址：指出接收端的地址

控制：用于通信双方交换命令和状态

校验：用于错误校验



串行通信的传输率

串行异步通信中，每秒传输的位数称波特率。

如：1200，2400，4800，9600，19200

大多数接口的接收波特率和发送波特率可以分别设置，可通过编程制定

这边貌似
有问题



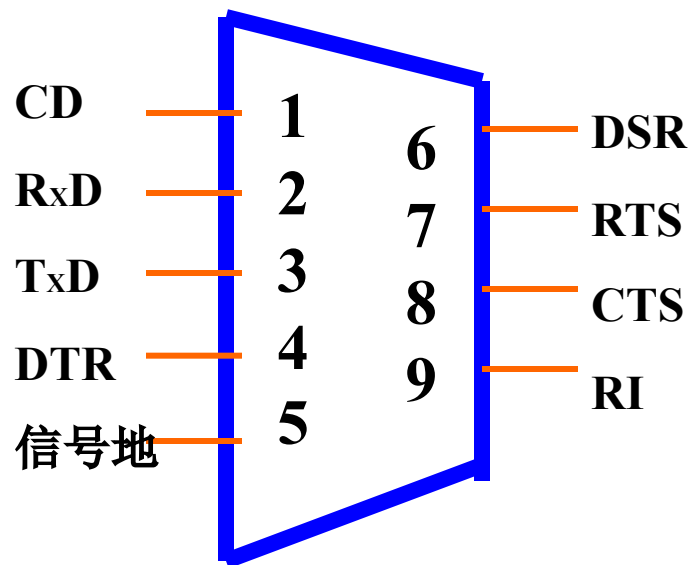
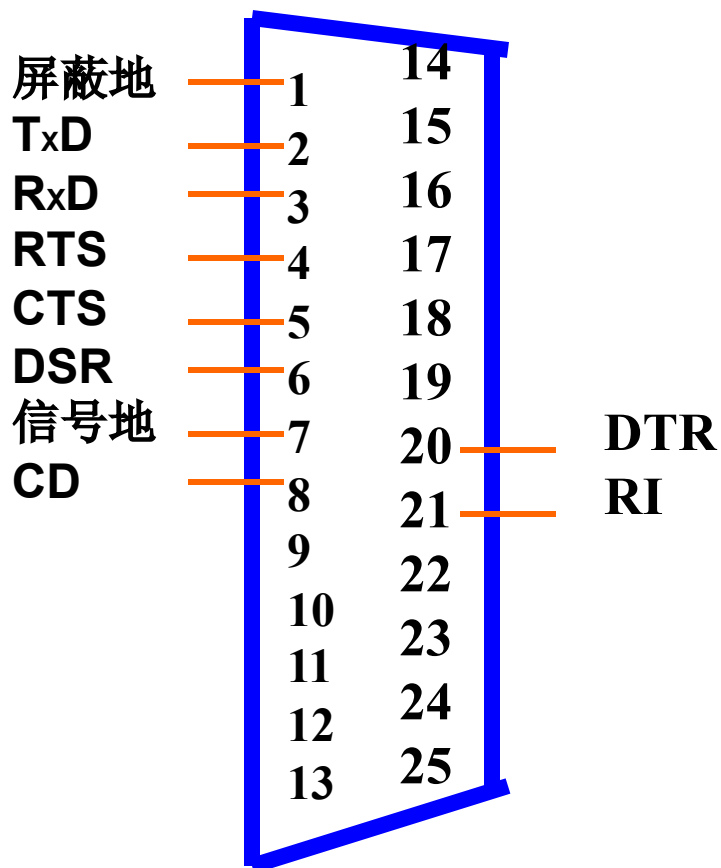
串行通信的接口标准

RS-232串行接口

该接口有两种连接器，一种DB-25; 另一种DB-9

数据线采用负逻辑规定逻辑电平;

+5V--- +15V规定为 “0”， --5V---- -15V规定为 “1”。





●信号间的作用如下：

TxD：发送数据

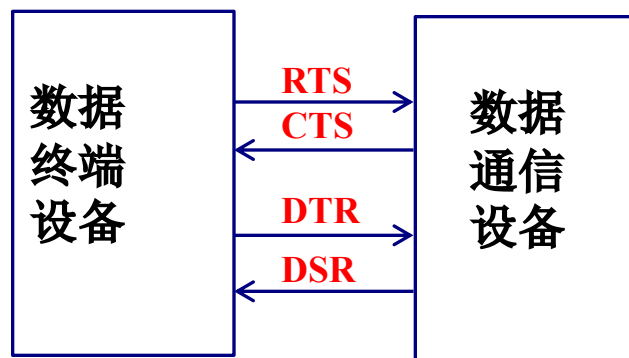
RxD：接收数据

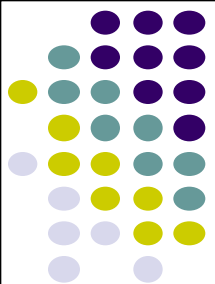
RTS：请求发送，数据终端设备发出，通知数据通信设备准备接收数据。

CTS：清除发送，数据通信设备准备好接收，允许发送。

DTR：数据终端设备准备就绪，准备接收。

DSR：数据通信设备准备就绪。

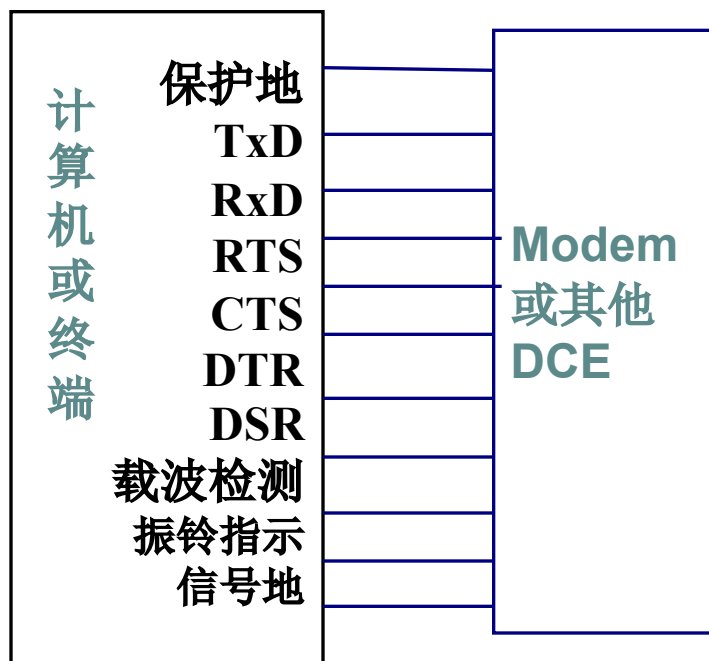


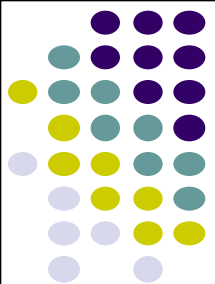


● RS-232连接

1 使用Modem连接

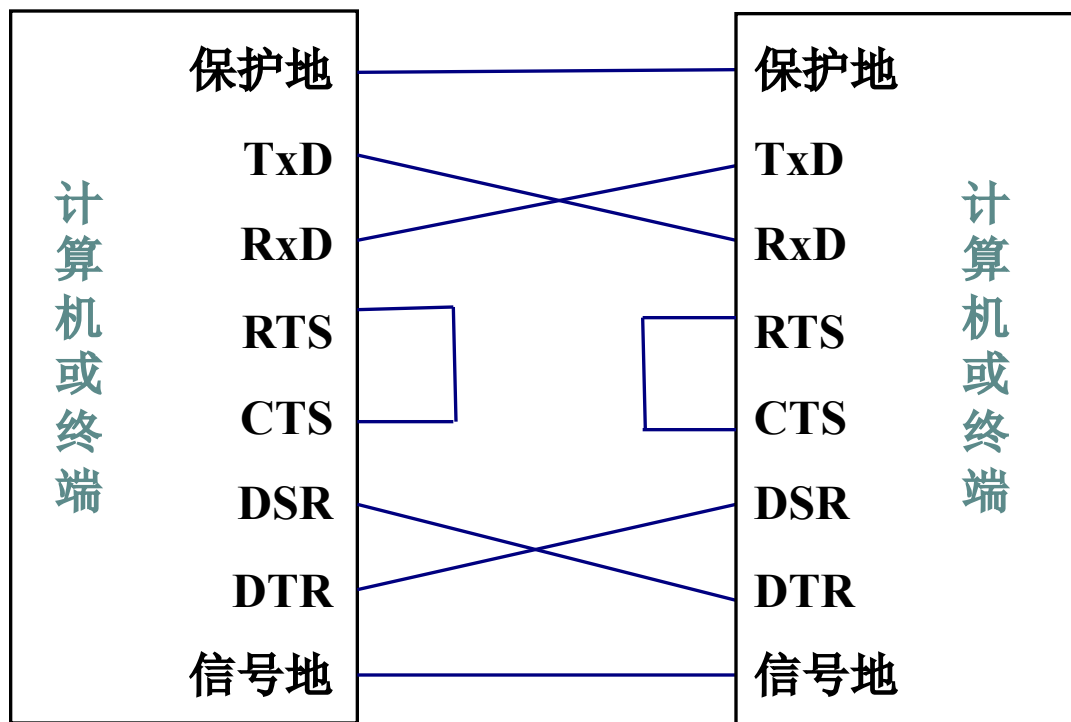
一般只需要9个常用信号

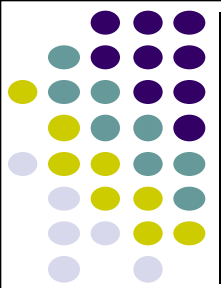




● RS-232连接

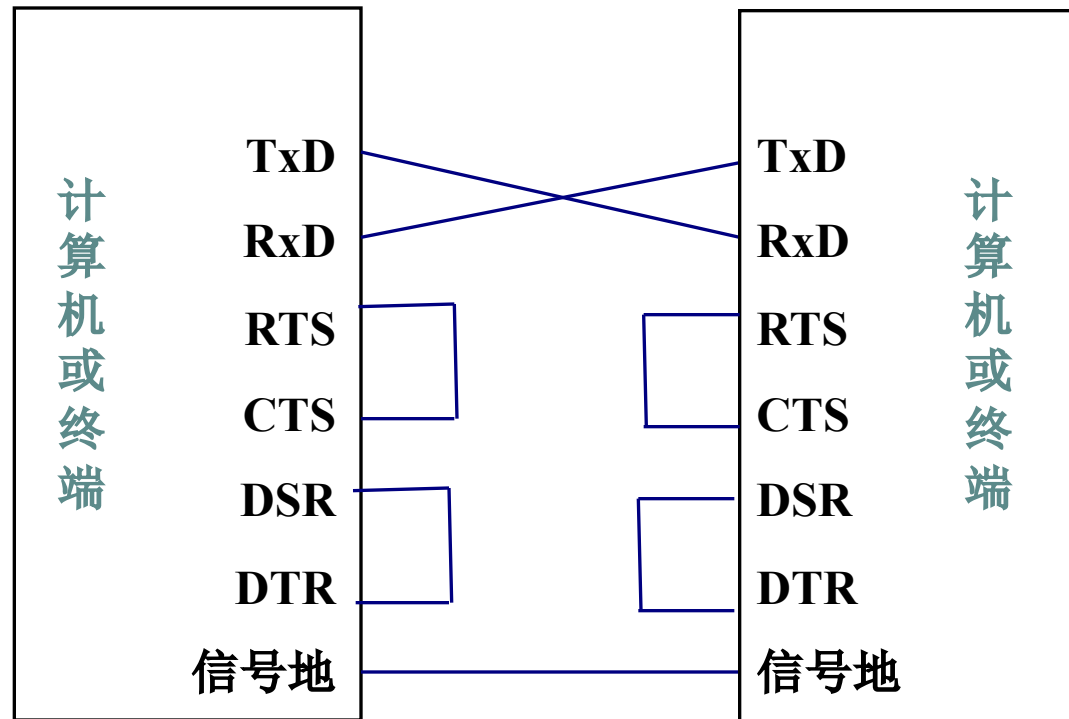
2 直接连接



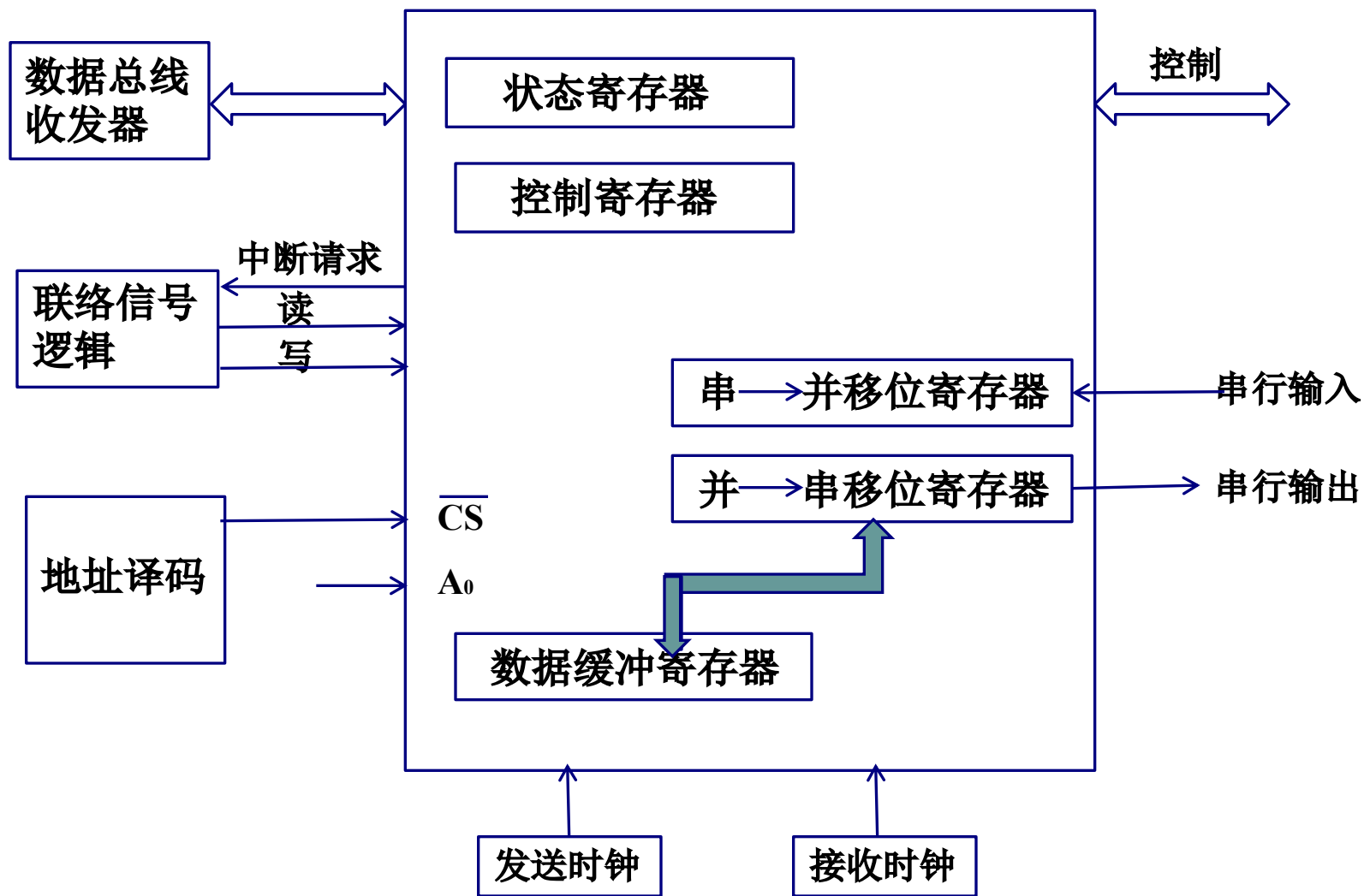


● RS-232连接

3 三线连接法



■ 可编程串行接口的典型结构





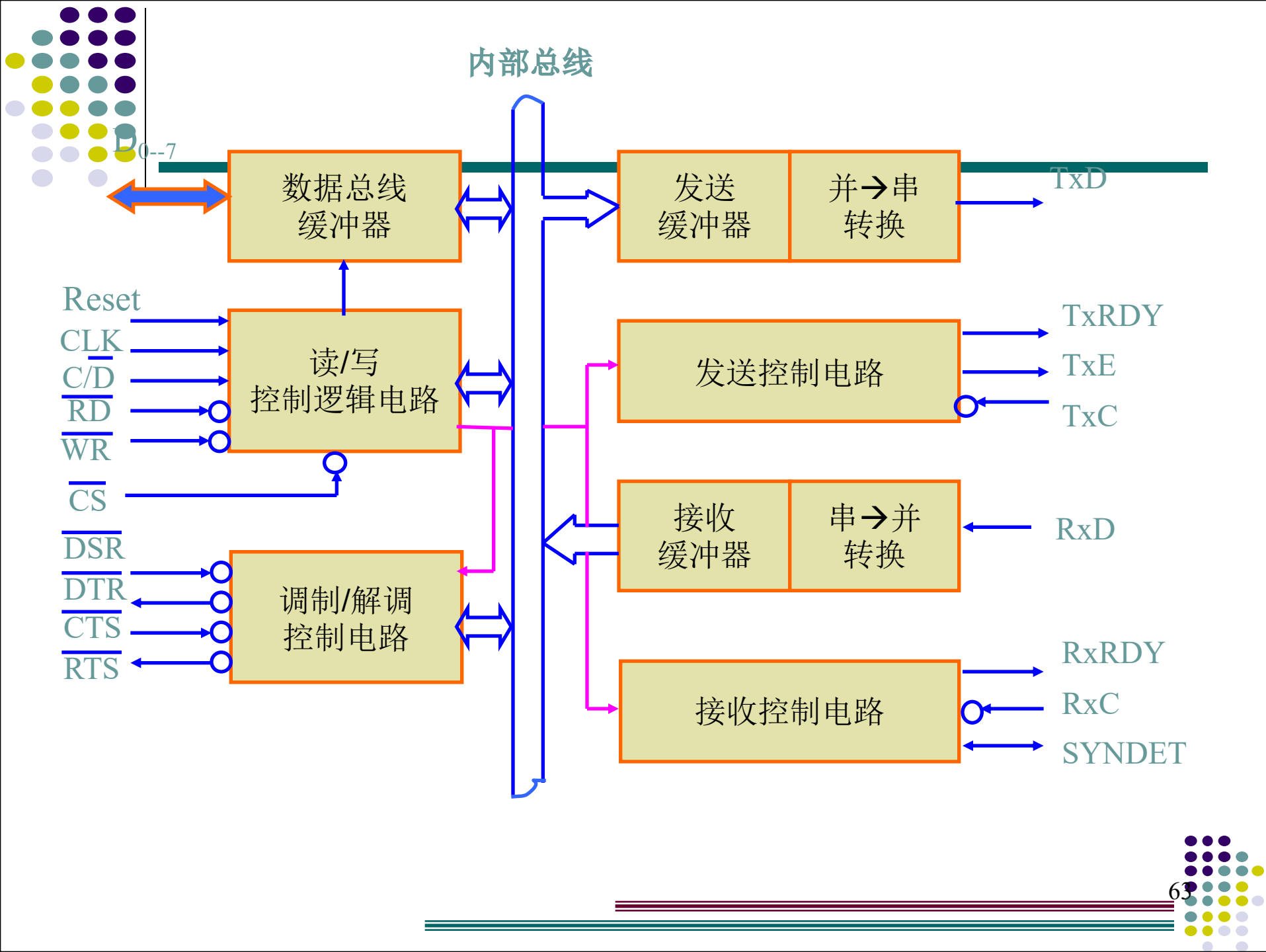
可编程串行通信接口8251A

1 特点

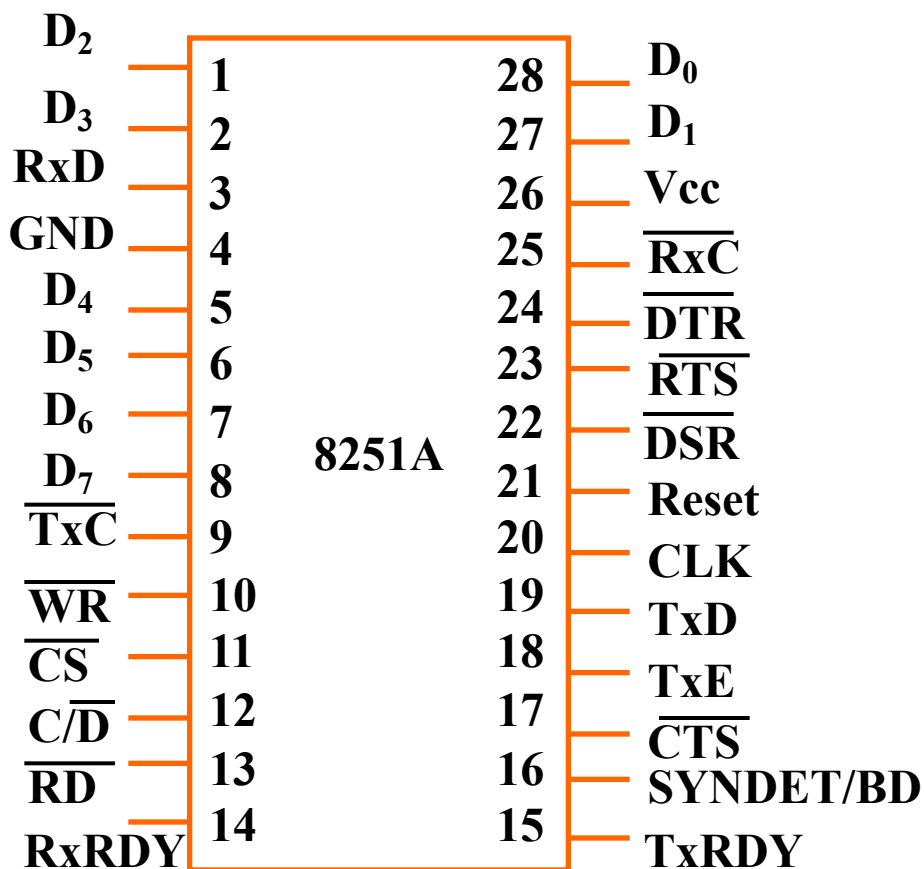
- 1) 可工作在全双工方式的串行通信;
- 2) 可用于同步和异步传送;
- 3) 同步传送: 5---8个比特代表字符, 内部或外部字符同步, 可自动插入一个或二个同步字符;
- 4) 异步传送: 5---8个比特代表字符, 波特率为时钟的1/1、1/16或1/64;
- 5) 具有奇偶、溢出和帧错误等检测电路;
- 6) 可产生1、1.5或2个停止位, 可检测假启动位, 自动检测和处理停止位。

2 内部结构

- 1) 总线缓冲器 用于同CPU交换数据;
- 2) 读/写控制逻辑;
- 3) 调制解调器控制;
- 4) 接收器, 把R_xD来的串行数据变成并行数据;
- 5) 发送器, 把并行数据变成串行数据从T_xD发出。



8251的管脚结构



8251A引脚图

1) D_{0—7}

8位数据总线，用于同CPU交换数据；

2) R_xD

串行接收数据；

3) T_xC

传送时钟；

4) WR, RD

CPU对8251的读写控制信号；

5) CS

片选信号；

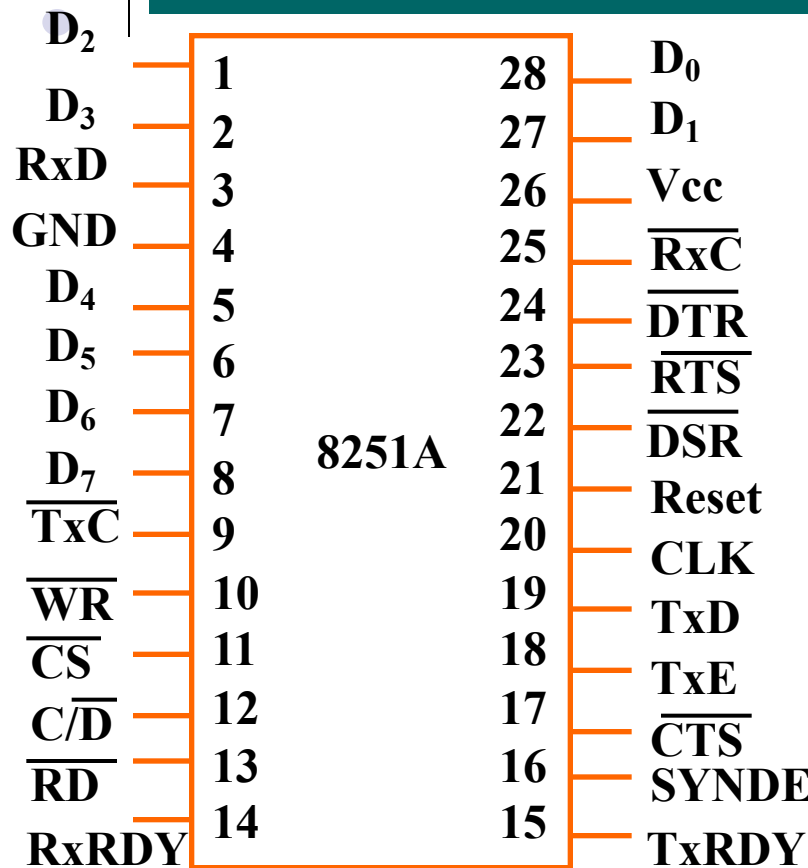
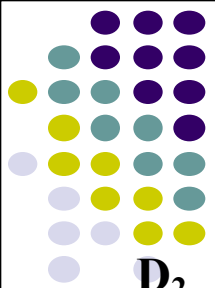
6) C/D

CPU向8251写时，用于选择命令/数据；
CPU从8251读时，用于选择状态/数据；

重点！

8251A的读写操作表

$\overline{\text{CS}}$	$\text{C}/\overline{\text{D}}$	$\overline{\text{RD}}$	$\overline{\text{WR}}$	功能
0	0	0	1	CPU从8251A读数据
0	1	0	1	CPU从8251A读状态
0	0	1	0	CPU写数据到8251A
0	1	1	0	CPU写命令到8251A
0	X	1	1	8251A数据总线缓冲器为高阻状态
1	X	X	X	本片未被选中



8251A引脚图

7) R_xRDY

接收准备好，用于通知CPU，8251已经接收了一个字符，CPU可以取走，8251每收到一个字符就使R_xRDY=1，CPU取走后又变为0；

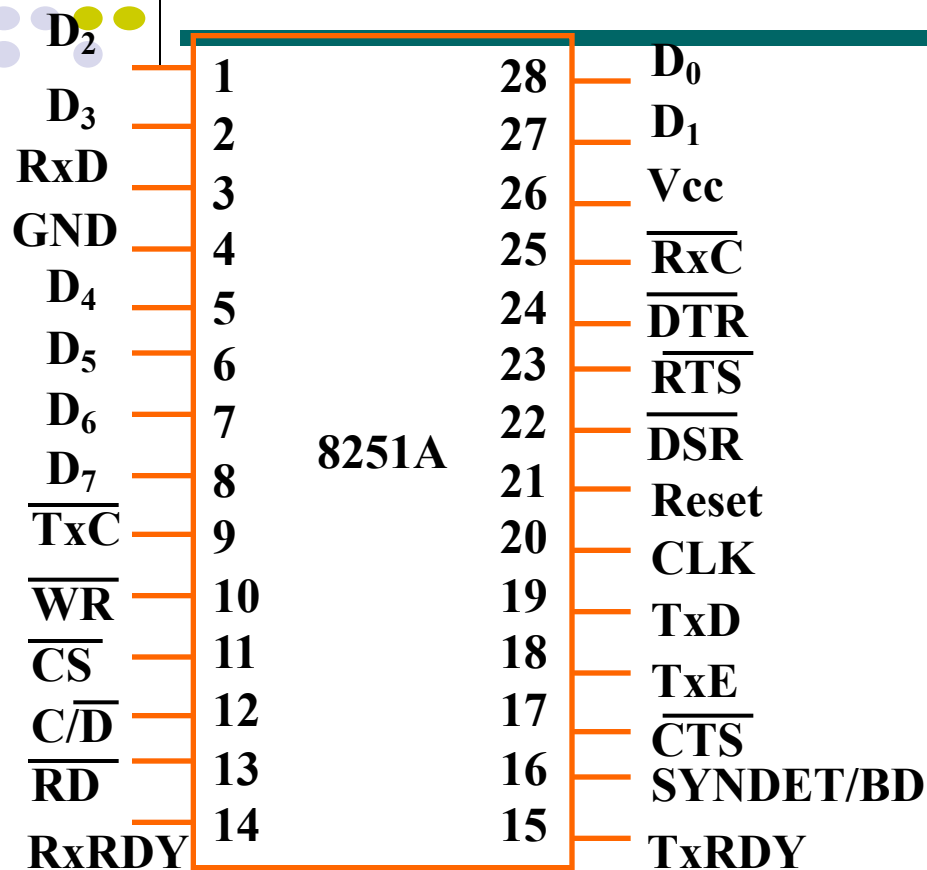
8) T_xRDY

传送准备好，用于通知CPU，8251可以从CPU接收数据；

9) SYNDET

此管脚和状态寄存器中的SYBDET一样；在同步方式时用于表示SYNC字符被发现；在内同步时，高电平输出表示8251已达到同步；在外同步，输入正跳沿使8251在下一个RXC的下降沿开始装配据；





8251A引脚图

10) CTS

允许发送。MODEM用于通知8251
MODEM已准备接收数据；

11) Tx̄E

传送寄存器空。并行/串行寄存器都为空
，在同步方式工作时若CPU来不及输出一个
新的字符，则将使Tx̄E变为高电平，同时发
送器在输出线上自动插入同步字符，以填补
传送空隙；

12) Tx̄D

CPU送来的并行数据被转换成串行数据由
这个输出端输出。

13) CLK

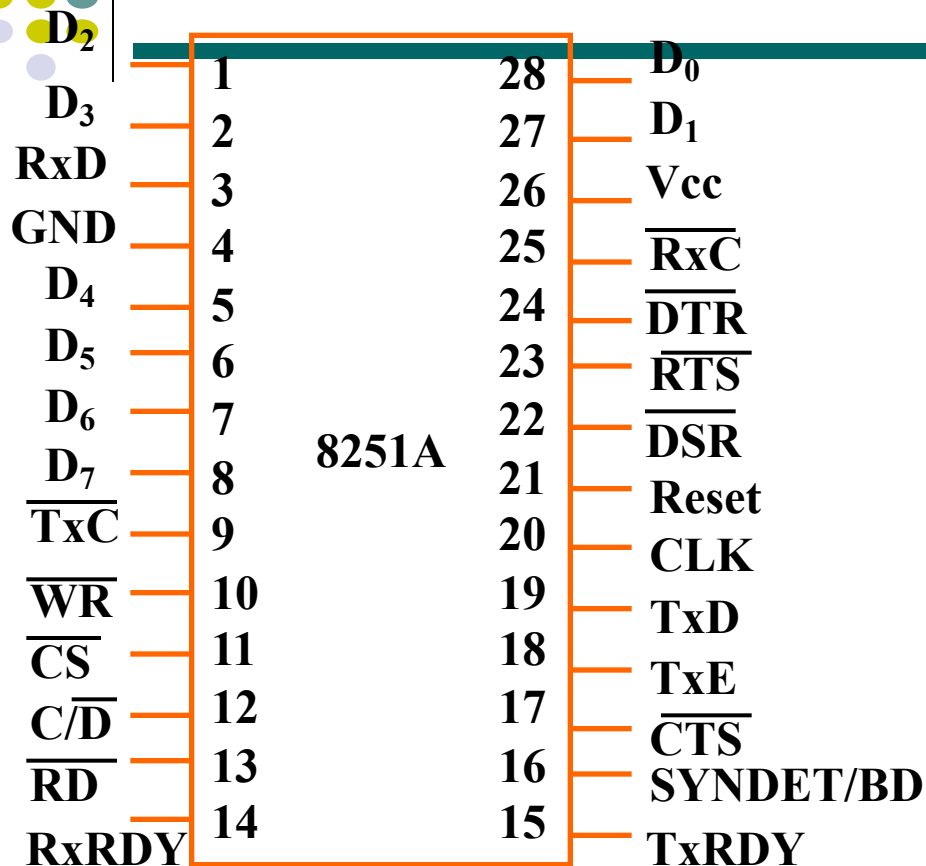
系统时钟；

14) RESET

复位信号；

15) DSR

MODEM来的信号，表示MODEM已准
备好；



16) RTS

8251 输出给MODEM, 用于通知MODEM8251准备传送数据;

17) DTR

8251 输出给MODEM, 用于通知MODEM8251已准备好;

18) RxC

接收时钟;

19) Vcc

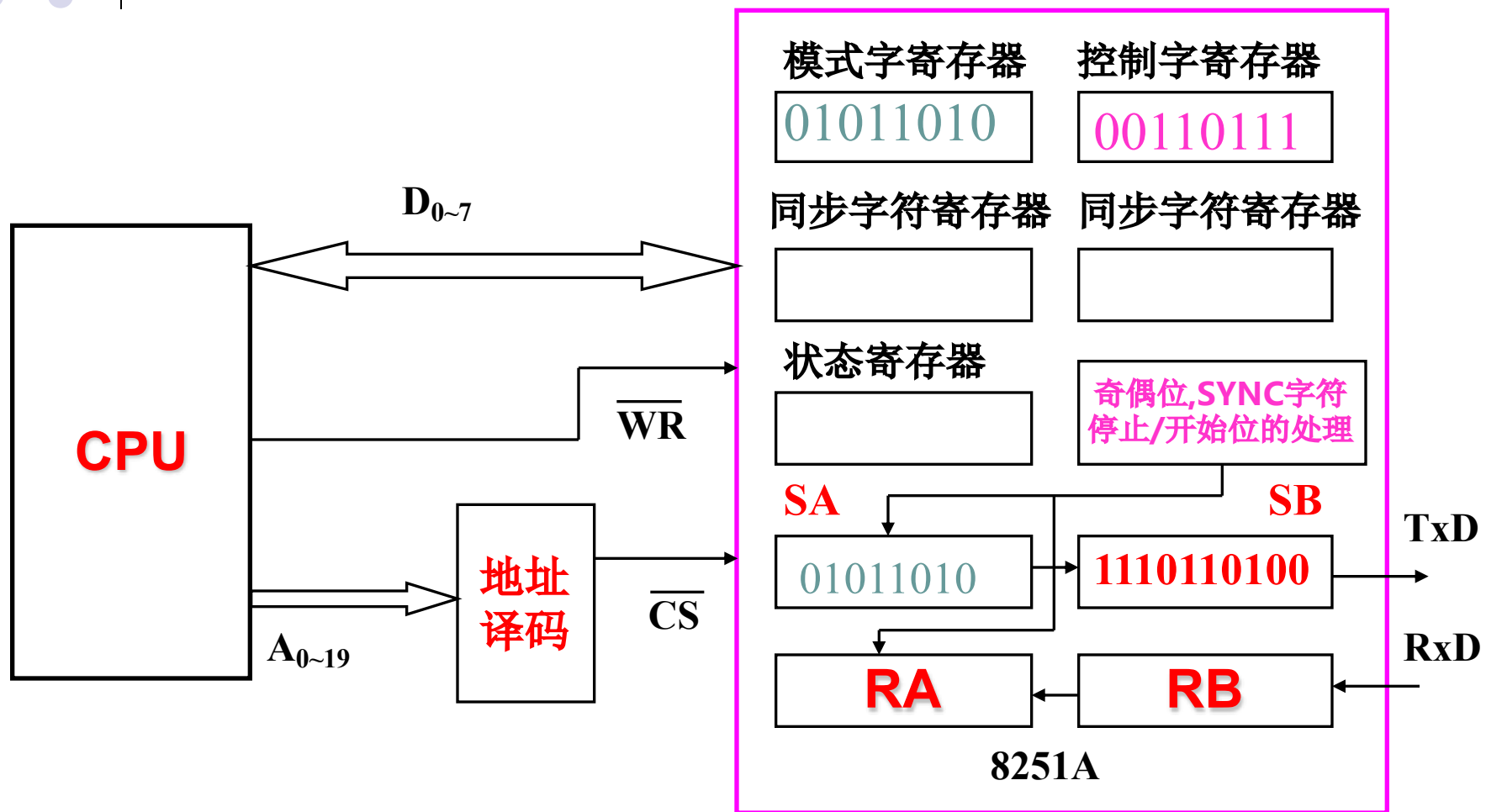
电源;

20) GND

信号地。

8251A引脚图

8251的工作流程





8251的端口地址

模式字寄存器、控制字寄存器和数据寄存器怎么区分？

- 端口地址

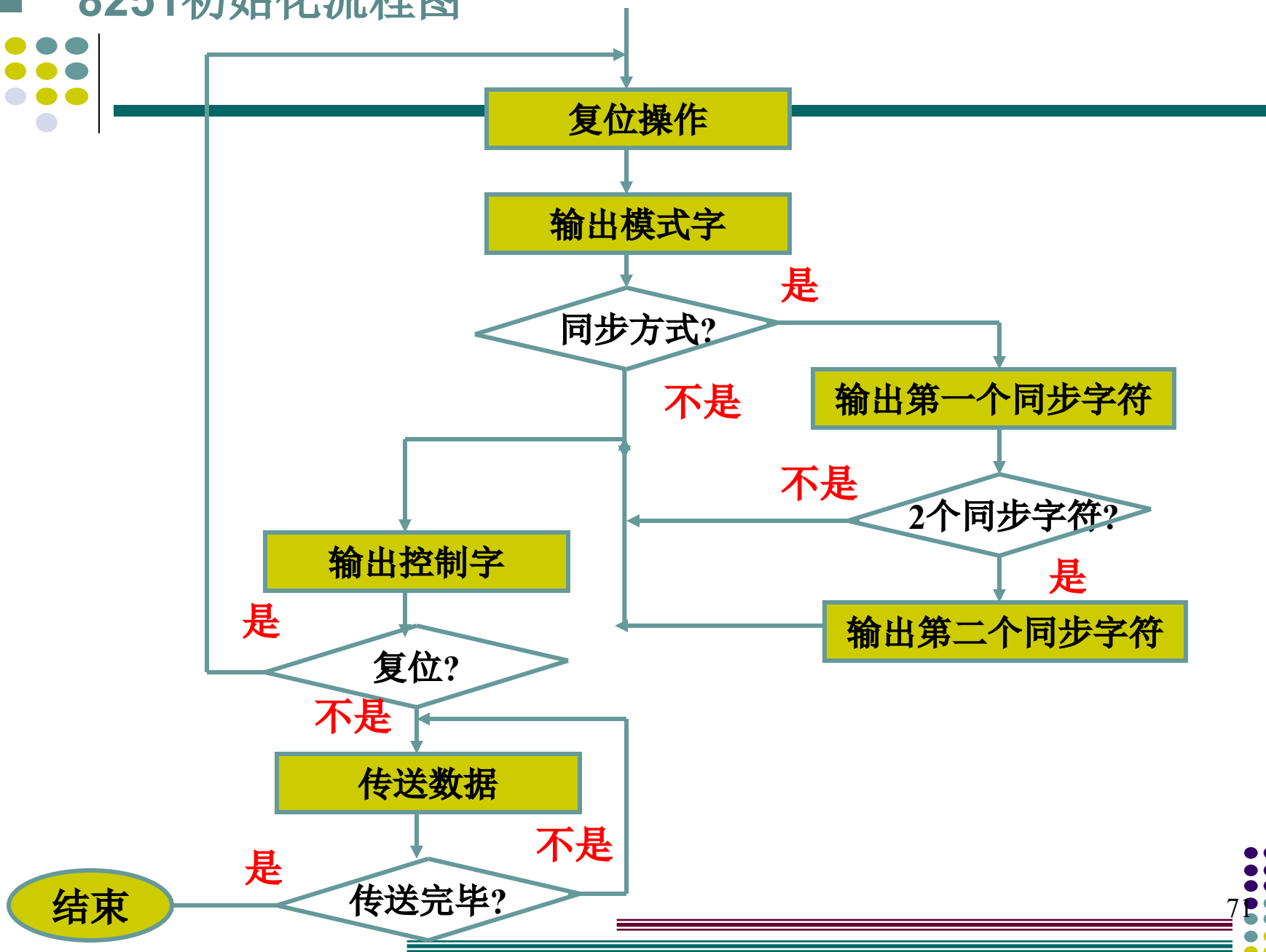
数据输入端口和输出端口合用一个地址

状态端口和控制端口合用一个地址

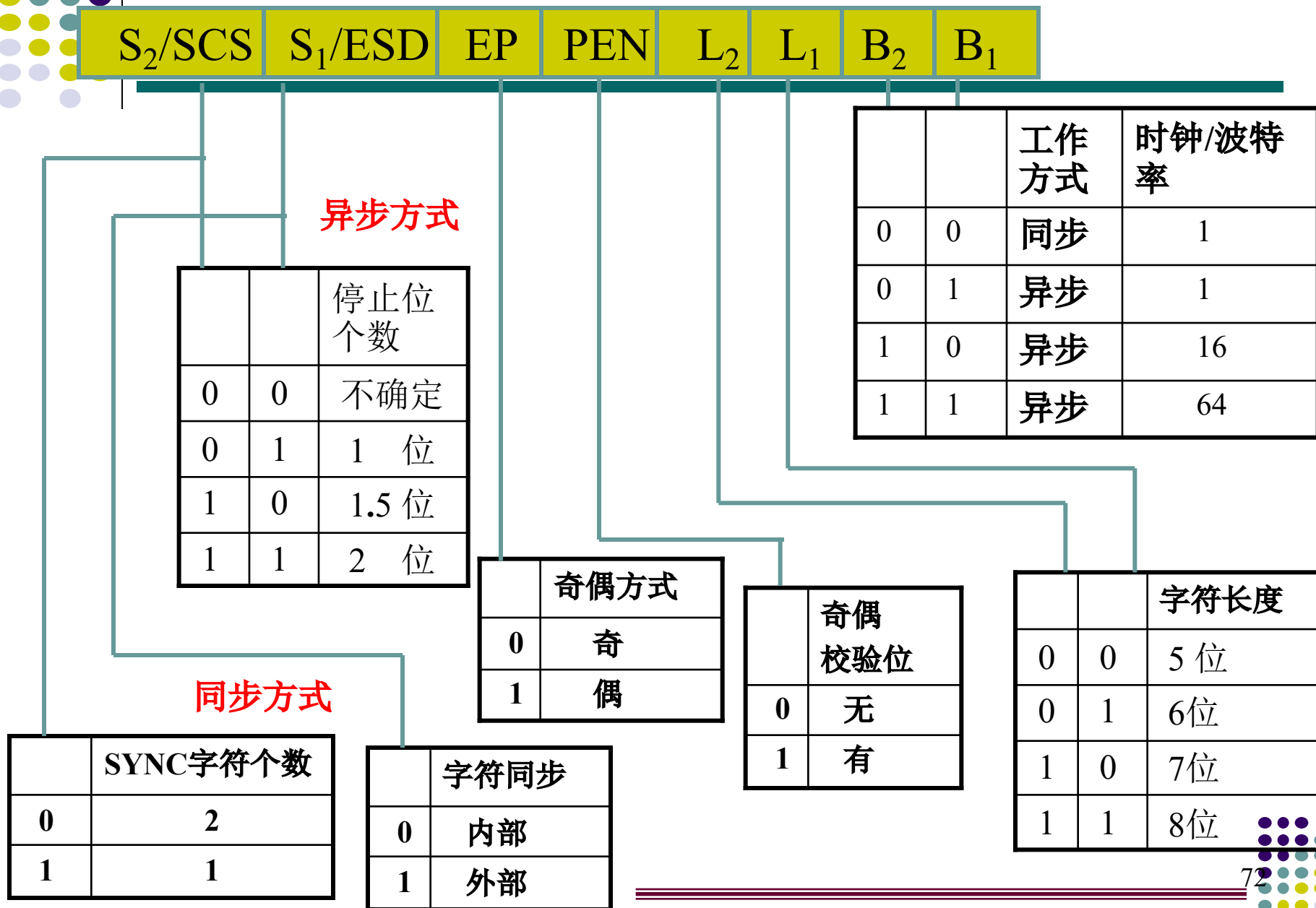
- 用1位地址线区分

- 16位系统中，用A1区分，A1低电平选中偶端口；A1高电平选中奇端口，正好连接C/D信号
- 32位系统中，用A3区分

8251初始化流程图



8251的模式字



8251的控制字

信

EH IR RTS ER SBRK R_xE DTR T_xEN

0	未开始检索同步字符
1	检索同步字符

0	正常工作
1	使 T _x D=0 输出一个空白字符

0	不允许传送
1	允许传送

0	使输出端 <u>DTR</u> = 1
1	使输出端 <u>DTR</u> = 0

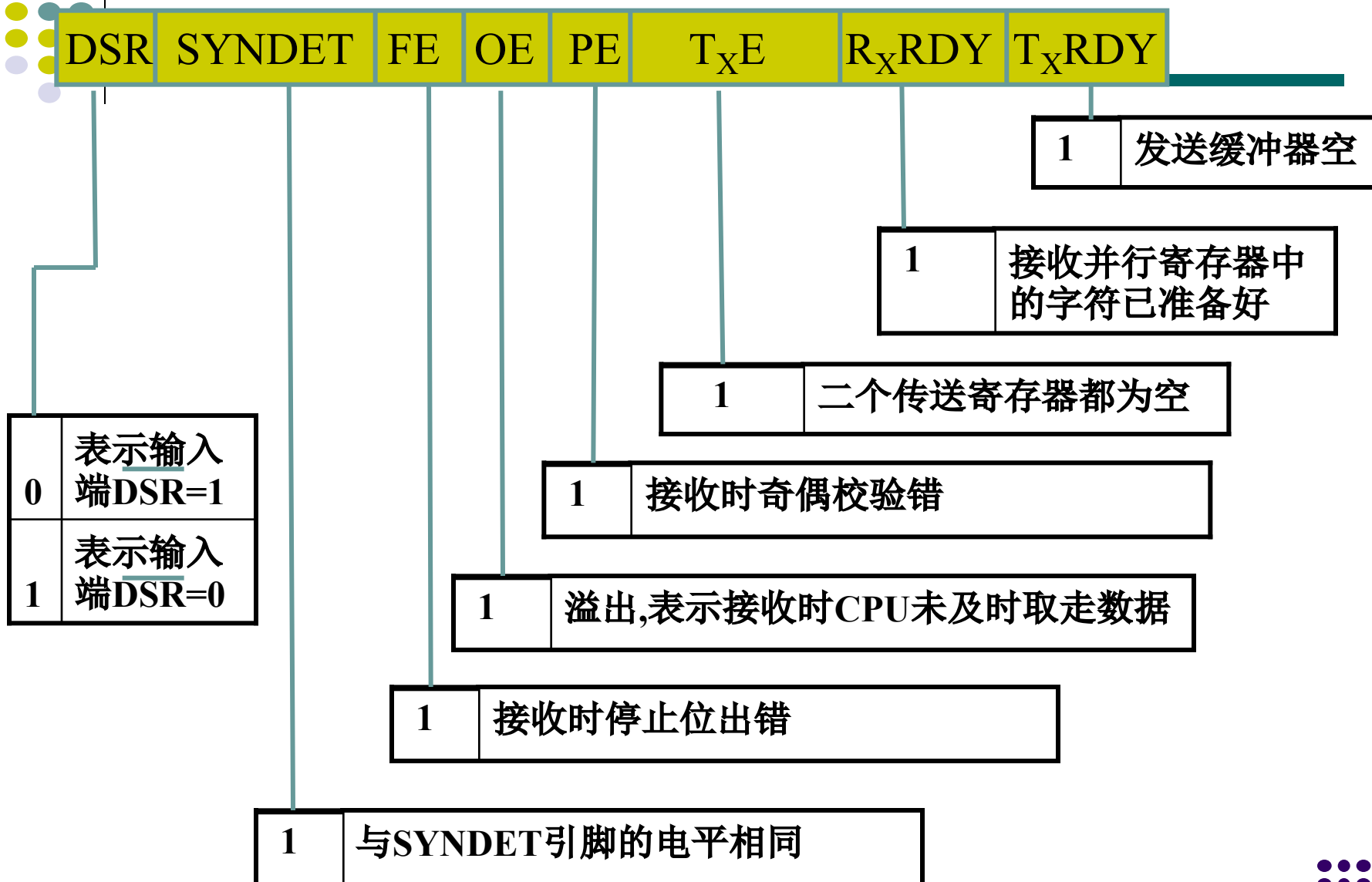
1	使状态寄存器的 PE,OE,FE 复位
0	不影响 PE,OE,FE

0	不允许接收
1	允许接收

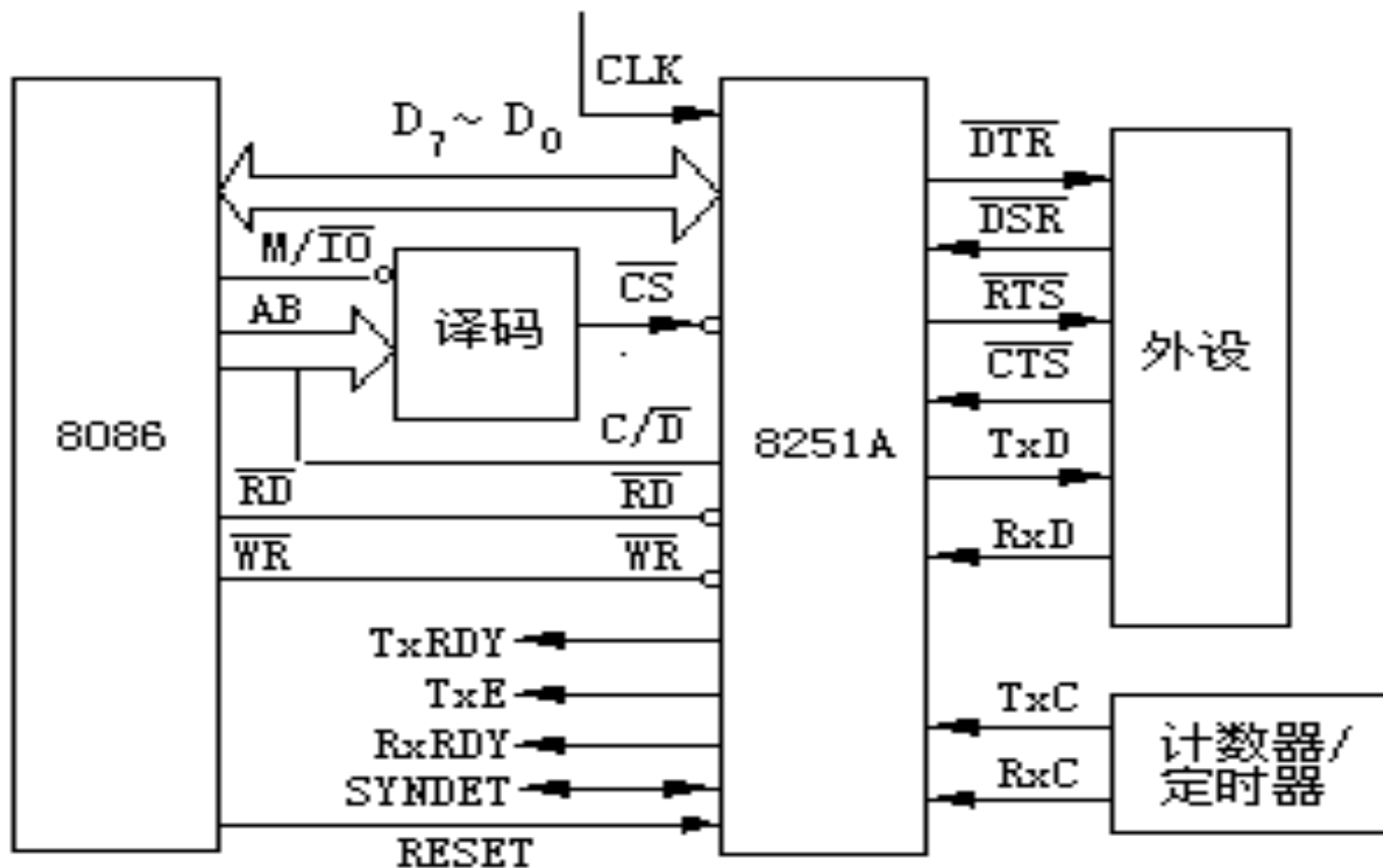
0	下一个不为模式字
1	下一个为模式字

0	使输出端 <u>RTS</u> = 1
1	使输出端 <u>RTS</u> = 0

8251的状态字



8251A与CPU的连接





8251A异步模式下的初始化程序

设8251A工作在异步模式，波特率系数(因子)为16，7个数据位/字符，偶校验，2个停止位，发送、接收允许，设端口地址为42H和44H。完成初始化程序。

程序段如下：

MOV AL, 0FAH

OUT 42H, AL

MOV AL, 37H

OUT 42H, AL

填充

； 设置模式字，异步模式，波特率因子16，7个数据位，偶校验，2个停止位 **11111010**

； 设置控制字，使发送、接收允许，清出错标志 **00110111**



8251A同步模式下的初始化程序

设端口地址为42H，采用内同步方式，2个同步字符（设同步字符为16H），偶校验，7位数据位/字符。

MOV AL, 38H	； 设置模式字，同步模式，用2个同步字符
OUT 42H, AL	7个数据位，偶校验 00111000
MOV AL, 16H	； 送同步字符16H
OUT 42H, AL	
OUT 42H, AL	
MOV AL, 97H	； 设置控制字，使接收器启动
OUT 42H, AL	10010111

8251编程举例

例1: 要求异步方式下, 输入80个字符, 波特率因子为16, 7位数据, 2位停止位, 偶校验。设数据端口地址为50H, 控制端口地址为52H。

```
MOV     AL, 0FAH
OUT     52H, AL           ; 写入模式选择字 11111010
MOV     AL, 37H
OUT     52H, AL           ; 控制字写入奇地址端口 00110111
MOV     DI, 0             ; 变址寄存器置“0”
MOV     CX, 80            ; 送入计数初值80个字符
BEGIN:  IN     AL, 52H
        TEST    AL, 02H    ; 测试状态字第2位RxRDY
        JZ      BEGIN
        IN      AL, 50H    ; RxRDY有效, 从偶地址口输入数据
        MOV     BX, OFFSET Buffer
        MOV     [BX+DI], AL
        INC     DI
        IN      AL, 52H    ; 再读状态字
        TEST    AL, 38H    ; 测试有无三种错误 00111000
        JNZ     ERROR
        LOOP    BEGIN     ; 不够80个字符, 转BEGIN
        JMP     EXIT      ; 输入80个字符, 则转结束
ERROR:
        CALL    ERR_OUT
EXIT:
```