

7.2 并行通信接口8255A

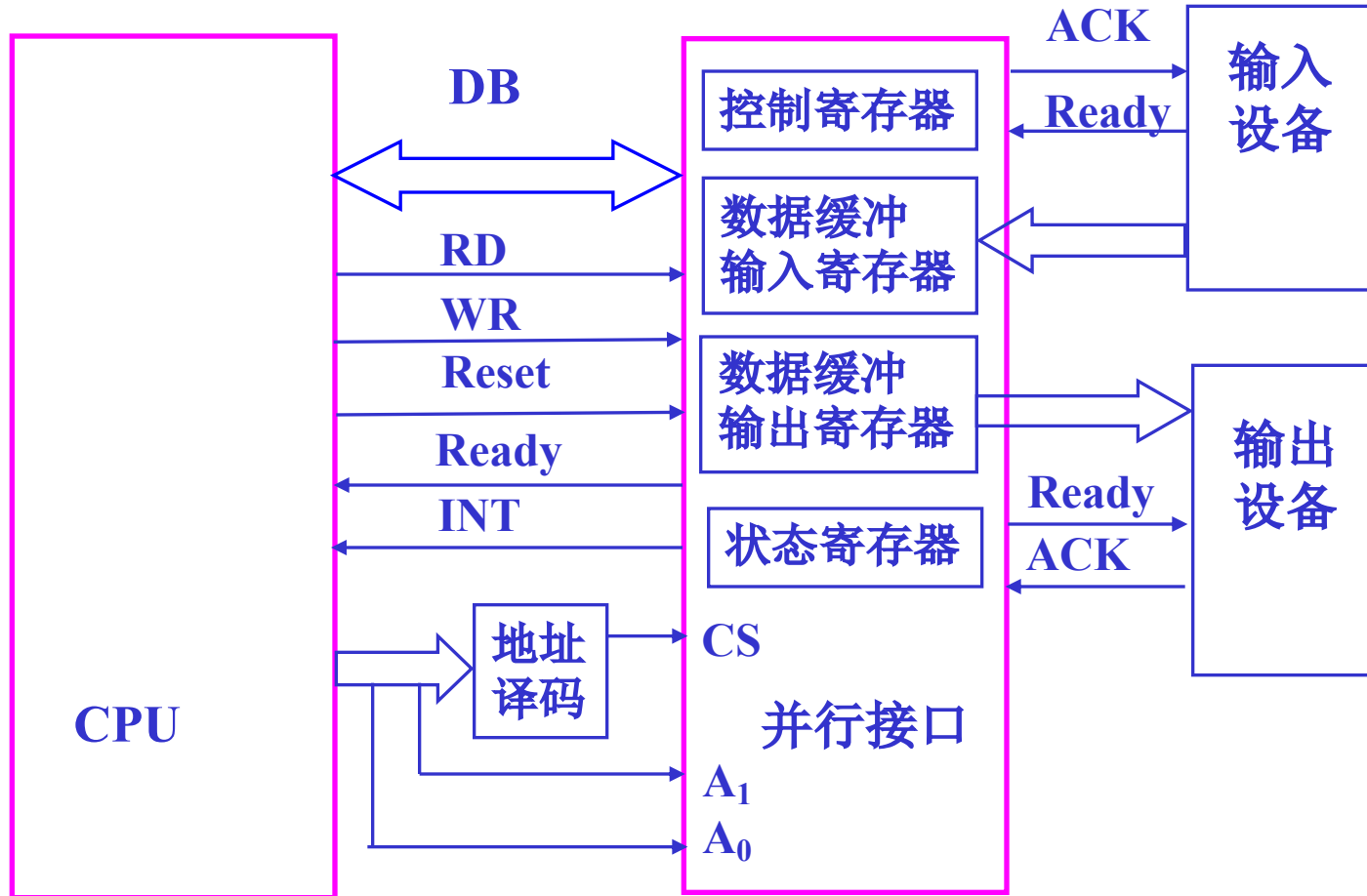
■ 并行通信

特点：每个字符的所有位是同时传送的，即：一个字符的每一位都有一个单独的通道。

并行通信通常用于短距离的高速传送。它的优点是传输速度快，然而对于远距离传送来讲，由于每一位都需要一个通道（即一根传输线）所以成本较高。



并行通信接口

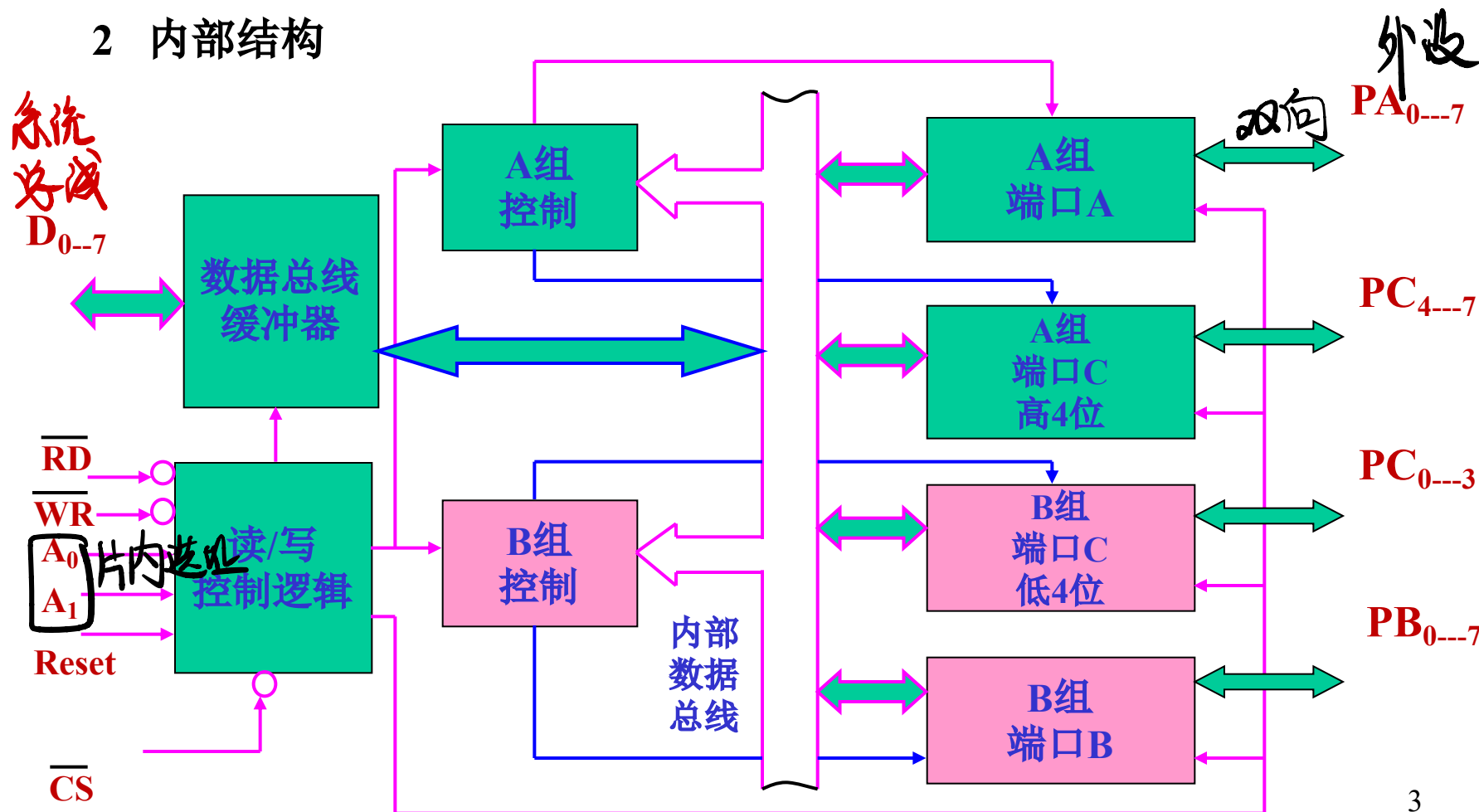


7.2.1 可编程并行通信接口8255A

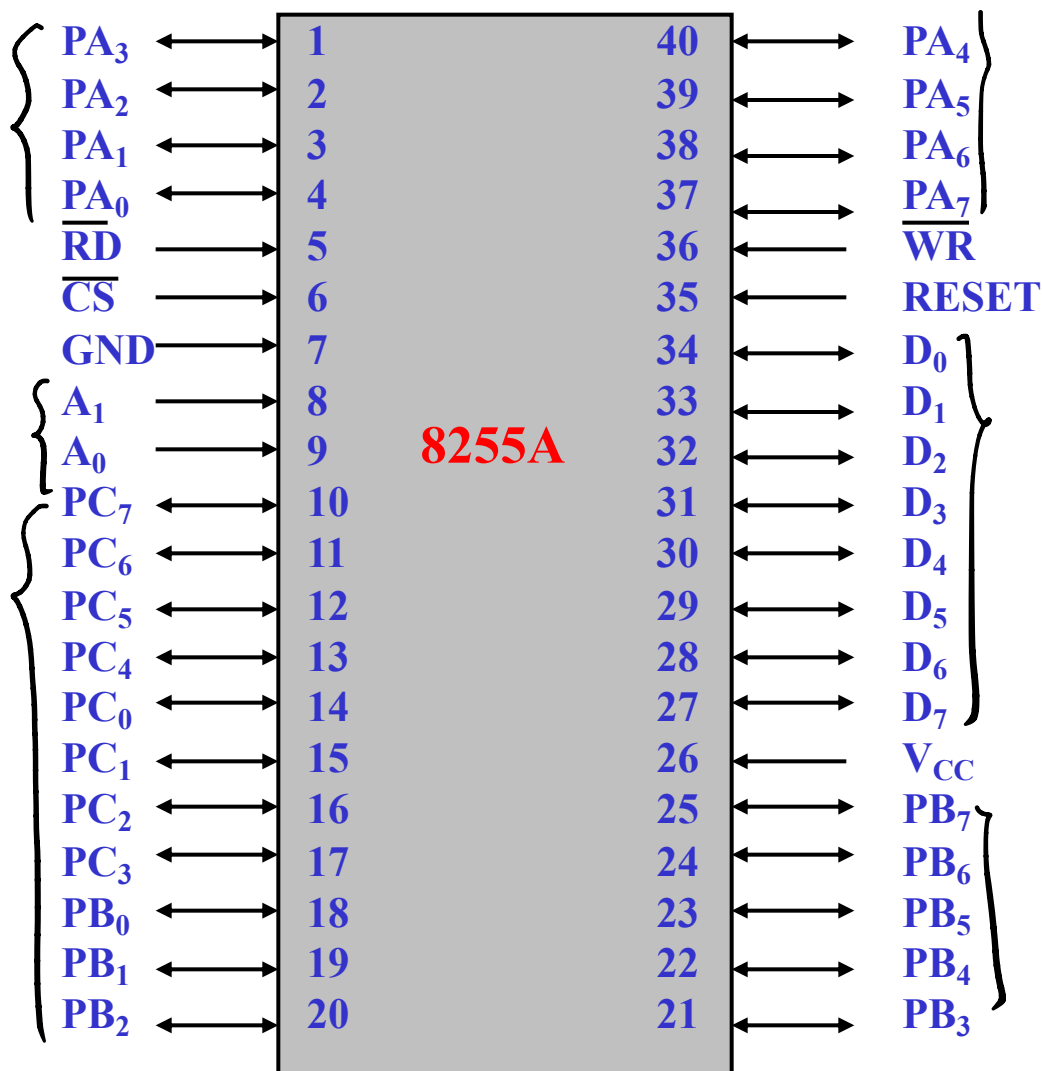
1 特点

- 1) 具有三个可编程为输入或输出的端口;
- 2) 端口C可被分成2组,每组4位;

2 内部结构



3 管脚结构



PA₀-PA₇: A端口
PB₀-PB₇: B端口
PC₀-PC₇: C端口
D₀-D₇: 数据总线
 \overline{RD} : 读有效
 \overline{WR} : 写有效
RESET: 复位
A₀、A₁: 片内寻址
 \overline{CS} : 片选
GND: 接地
V_{cc}: 工作电压

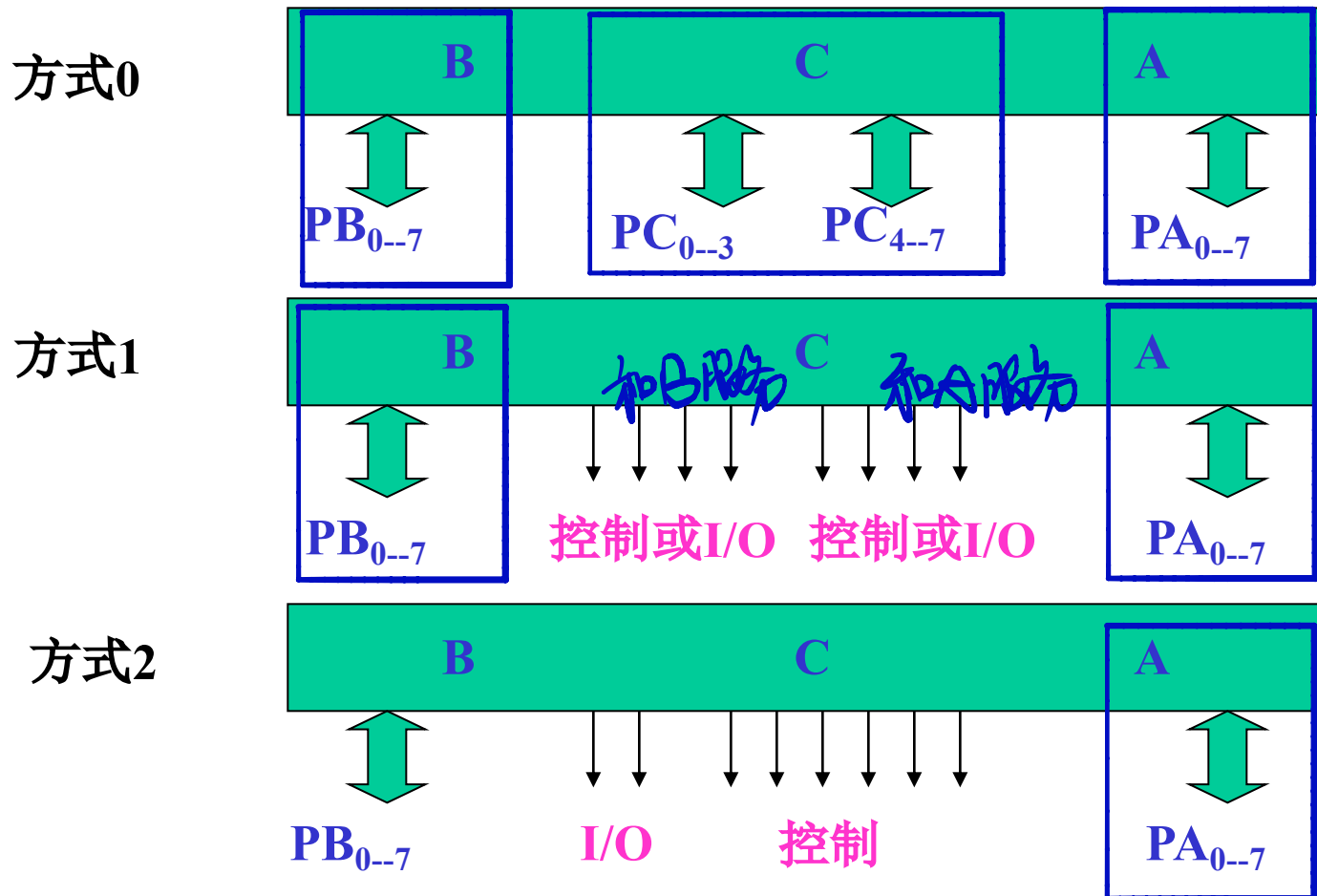
4 端口地址 重要

$\overline{\text{CS}}$	片内地址 A_1	A_0	$\overline{\text{RD}}$	$\overline{\text{WR}}$	读操作	内容
0	0	0	0	1	PA口->数据总线->CPU	数据
0	0	1	0	1	PB口->数据总线->CPU	数据
0	1	0	0	1	PC口->数据总线->CPU	数据
					写操作	数据或状态
0	0	0	1	0	PA口<-数据总线<-CPU	数据
0	0	1	1	0	PB口<-数据总线<-CPU	数据
0	1	0	1	0	PC口<-数据总线<-CPU	数据
0	1	1	1	0	控制寄存器<-数据总线 控制口	控制字

■ 8255A工作方式

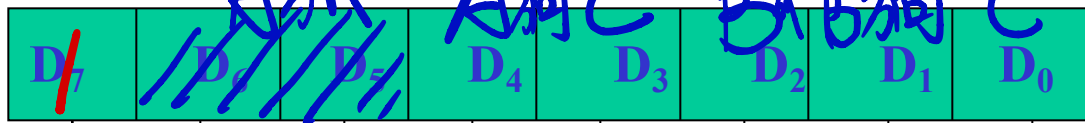
1 8255A的工作方式(3种)

- 1) 方式0-----基本输入/输出 (没有握手的输入/输出)
- 2) 方式1-----选通输入/输出 (具有握手的单向输入/输出) - 问答
- 3) 方式2-----双向输入/输出 (具有握手的双向输入/输出)



2 8255A的方式选择控制字 1字节(8位)

用于指定A、B、C三个端口的工作方式



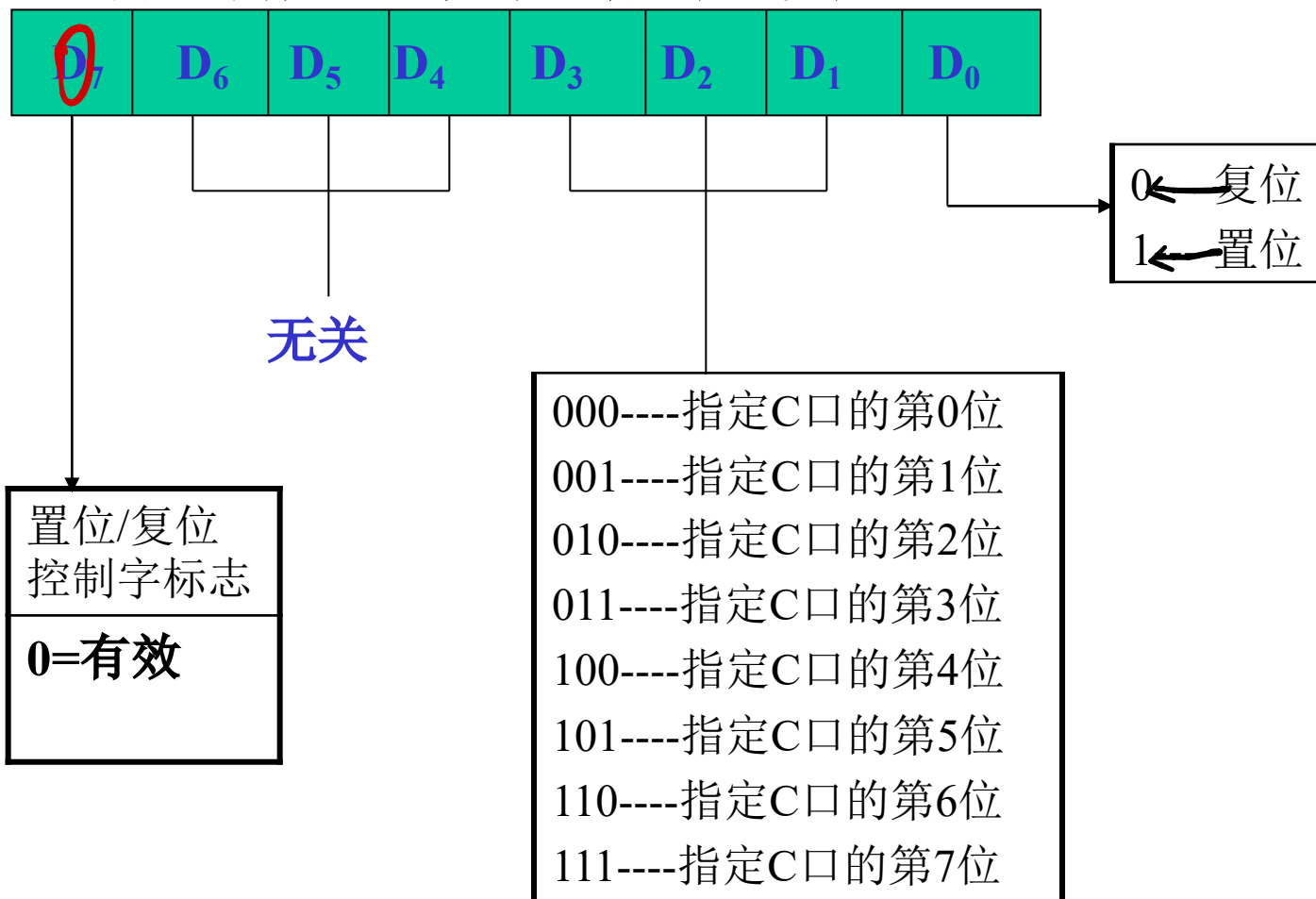
置方式标志
1=有效

A组	
端口C上半部分	
0----	输出
1----	输入
端口A	
0----	输出
1----	输入
方式选择	
00-----	方式0
01-----	方式1
1X ----	方式2

B组	
端口C下半部分	
0----	输出
1----	输入
端口B	
0----	输出
1----	输入
方式选择	
0----	方式0
1----	方式1

3 8255A的置位/复位控制字

只用于对端口C的每一位进行置位或复位



设8255A的控制口地址为FF83H，数据口A、B、C的地址分别为80H、81H、82H。请说明下列程序的功能。

MOV DX, 0FF83H *15b*

MOV AL, 91H ←

OUT DX, AL

DEC DX

IN AL, DX

10010001

A组----方式0

A口输入，C口上半部分输出

B组----方式0

B口输出，C口下半部分输入

从C口读入数据

8255两个控制字

MOV DX, 0FF83H

MOV AL, 0FH

OUT DX, AL

MOV AL, 06H

OUT DX, AL

00001111
C7

PC₇置1

00000110
C3

PC₃置0

2 8255A的程序设计

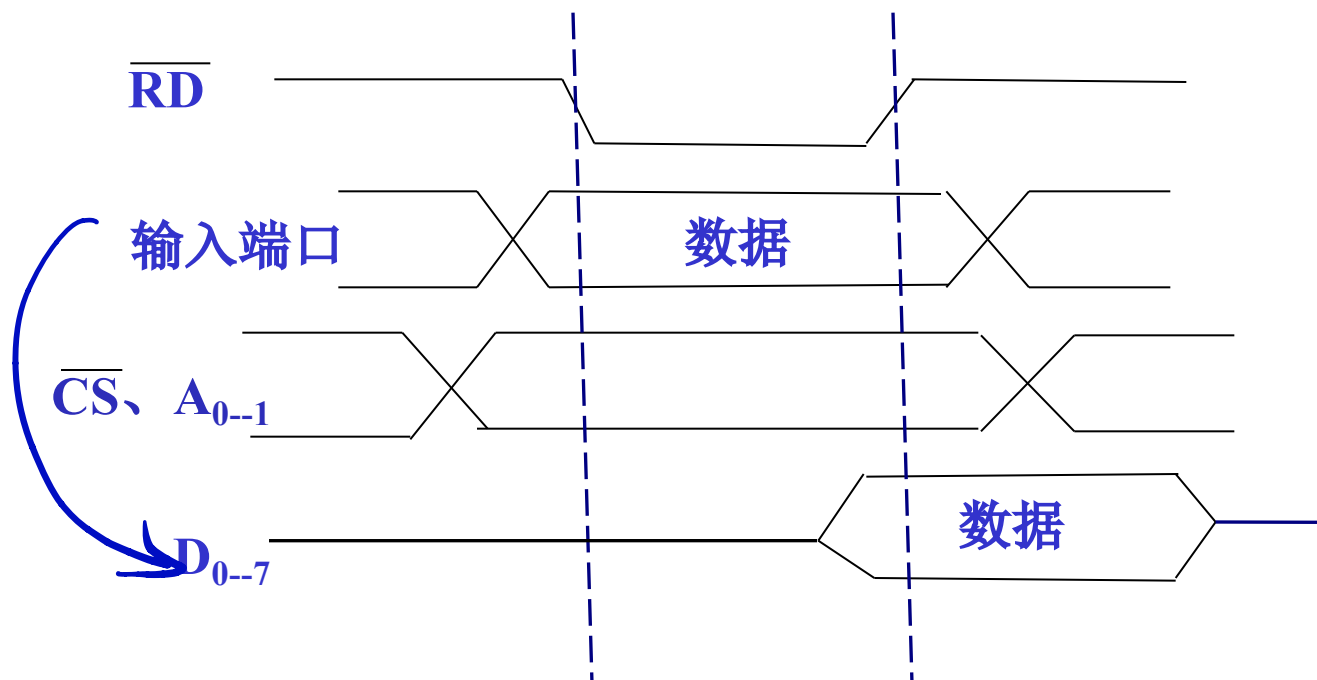
1) 方式0-----基本输入/输出（没有握手的输入/输出）

特点：

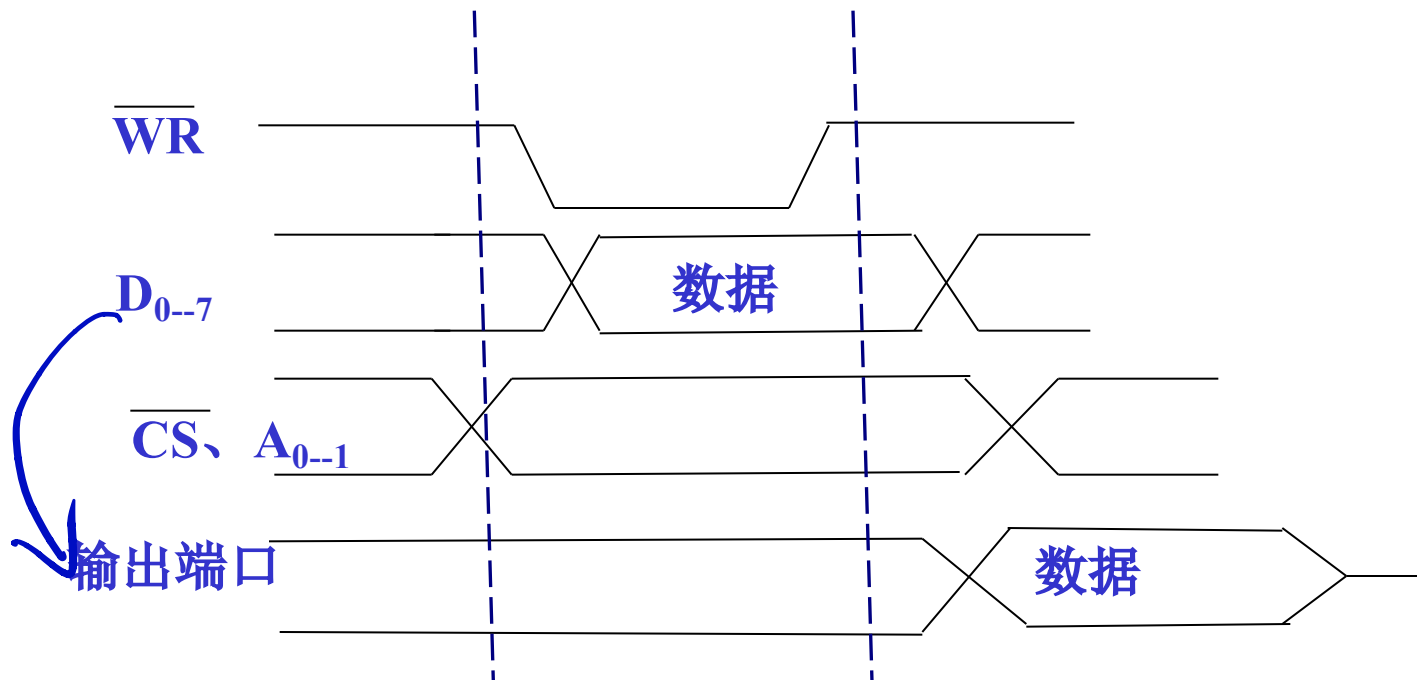
- ◆ 每一个端口都可以作为输入或者输出端口；
- ◆ 端口C可以分成二个（ PC_{0-3} 和 PC_{4-7} ）相互独立的部分。

(1) 方式0的输入输出时序

(A) 输入时序

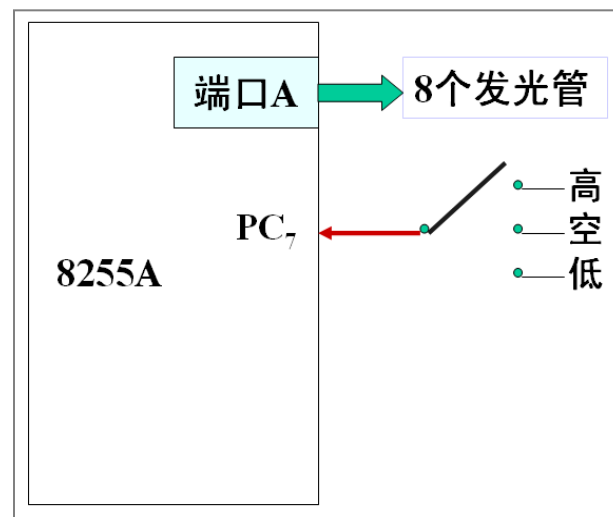


(B) 输出时序



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 80H    ; 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, OFFH     ; 设置全亮数据位
      OUT 60H, AL       ; 写A口, 点亮全部管
      JMP NEXT
```

2) 方式1-----选通输入/输出（具有握手的单向输入/输出）

特点：

- ◆三端口被分成两组（A组和B组）；
- ◆每一组分别包含有8位的数据通道和4位的控制/数据通道；
- ◆8位的数据通道既可以作为输入，也可以作为输出，且带有锁存；
- ◆若只有一个端口工作在方式1，余下的13位可工作在方式0；
- ◆若二个端口都工作在方式1，端口C还留下二位，它们可以由程序指定作为输入或输出，也具有置位和复位功能。

(A) 方式1的输入

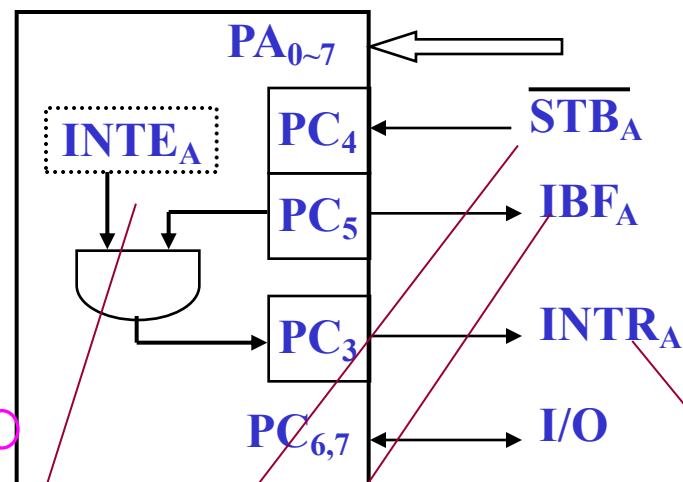
1	0	1	1	1/0	X	X	X
---	---	---	---	-----	---	---	---

方式1

A口为输入

PC_{6,7} (0-输出)

\overline{RD}



端口A方式1引脚配置

1	X	X	X	X	1	1	X
---	---	---	---	---	---	---	---

B口为输入

方式1

STB是外设数据输入选通信号。低电

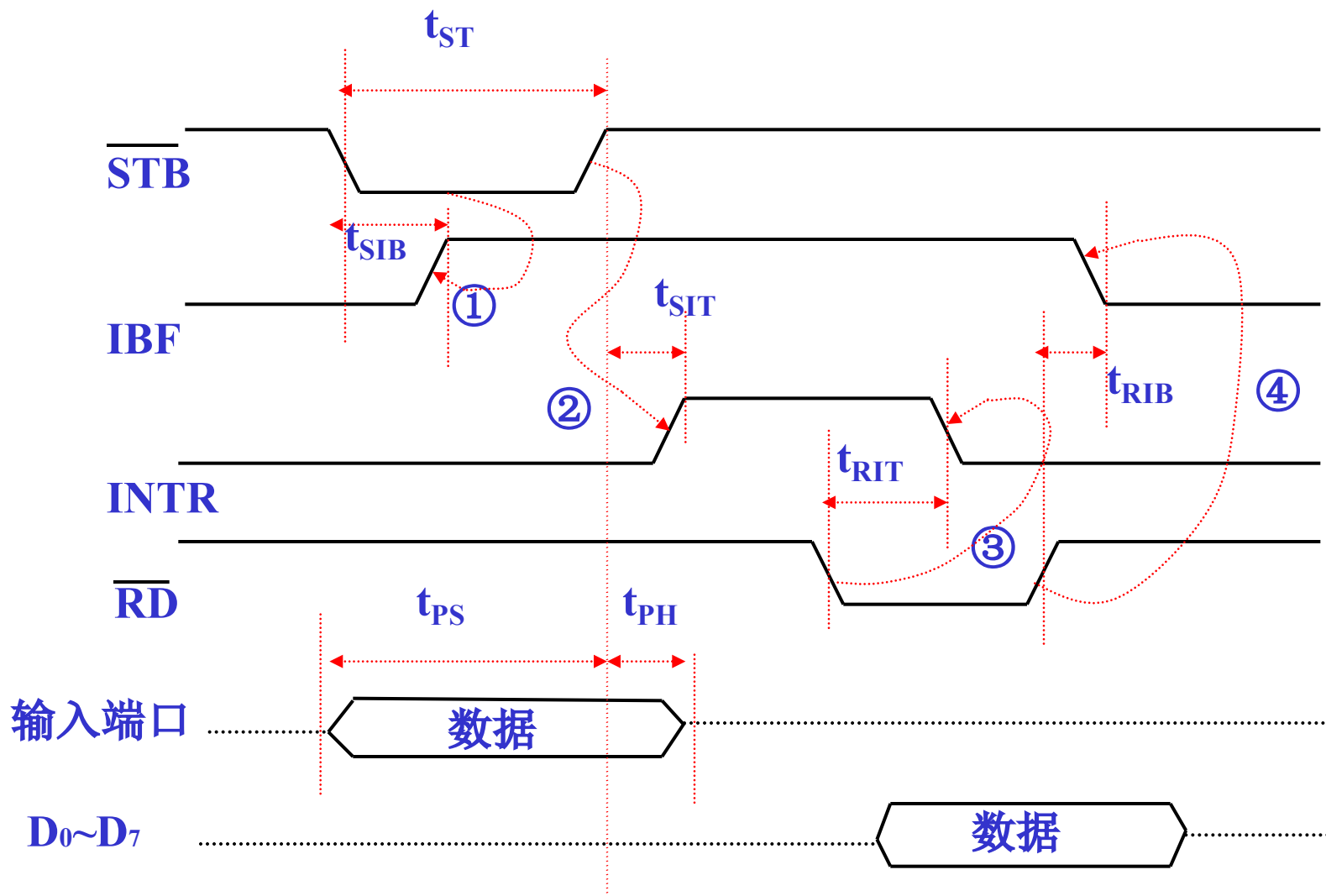
IBF 是端口数据输入缓冲器“满”的状态信号，高电平有效。A口的IBF信号由PC₅输出。有效时，表示端口“就绪”

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PC₄的置位/复位实现控制

★PC₄置1，A口处于中断允许状态，又当IBF=1时，INTR便有效，向CPU申请中断

★PC₄复位为0，则A口被置于中断屏蔽状态

(B) 方式1的输入时序



(C) 方式1的输出

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

方式1

PC_{4,5} (0-输出)

A口为输出

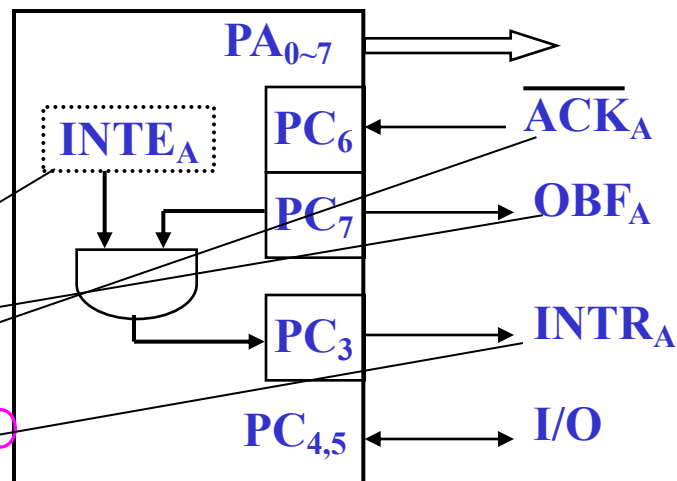
\overline{WR}

▲ OBF 输出缓冲器“满”信号，低电平有效。CPU数据到达端口后有效，通知外

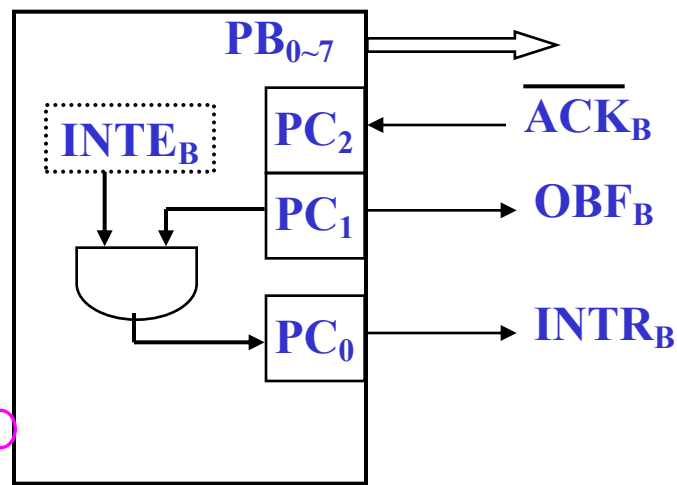
▲ INTR为中断请求信号，高电平有效。当外设取走数据后，端口已“空”，发出中断请求

▲ 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PC₆和PC₂的置位/复位实现

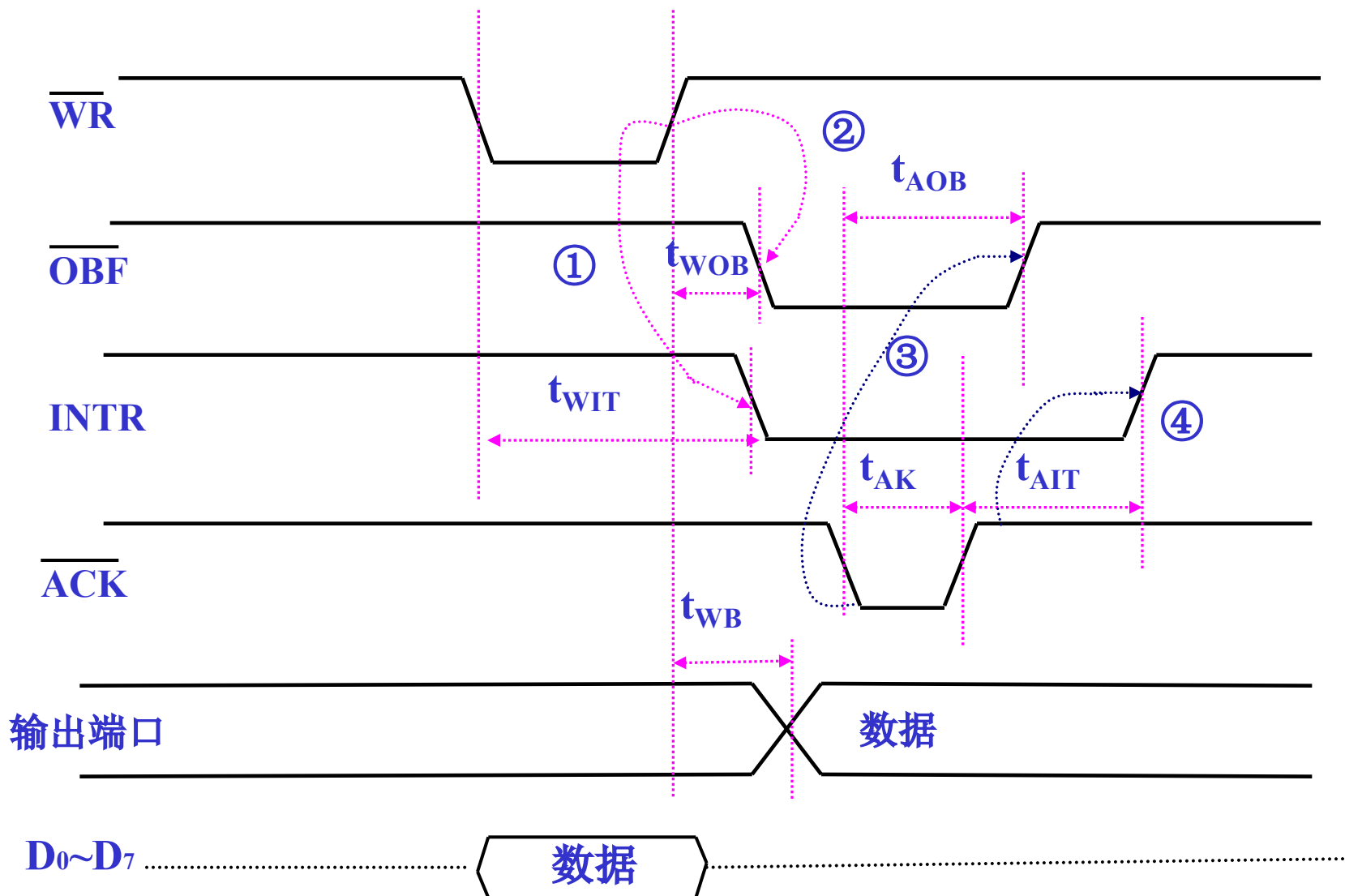


端口A方式1引脚配置



端口B方式1引脚配置

(D) 方式1的输出时序



3) 方式2-----双向输入/输出（具有握手的双向输入/输出）

特点:

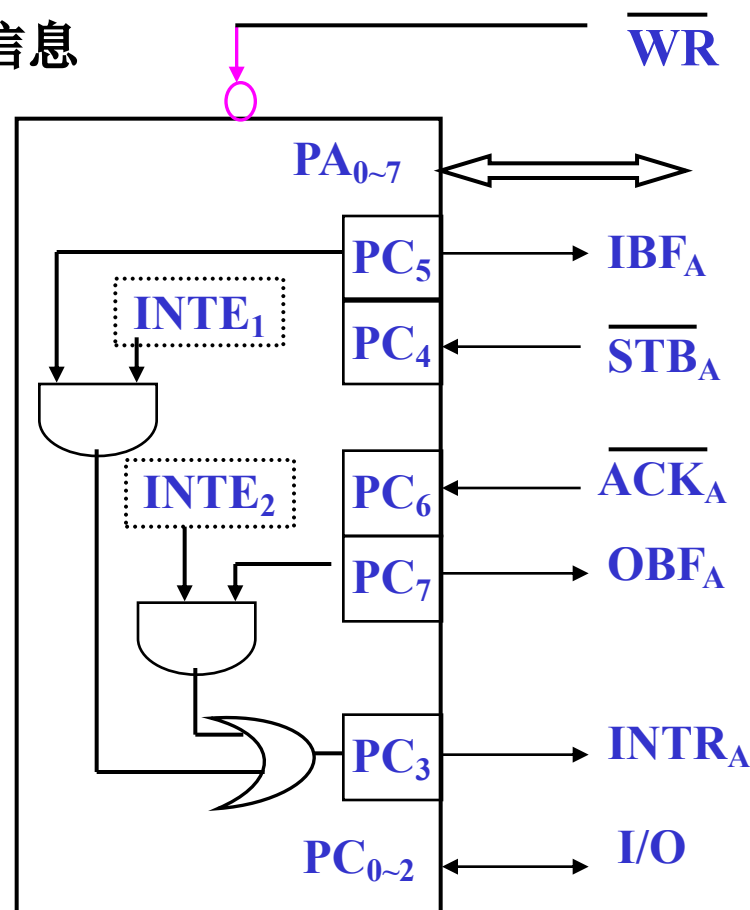
- ◆本方式只用于端口A;
- ◆具有一个8位的双向总线端口A和一个5位的控制端口C;
- ◆C口的5位作为端口A 的控制和状态信息
- ◆输入/输出都带有锁存功能。

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

PC_{0~2}
(0-输出)

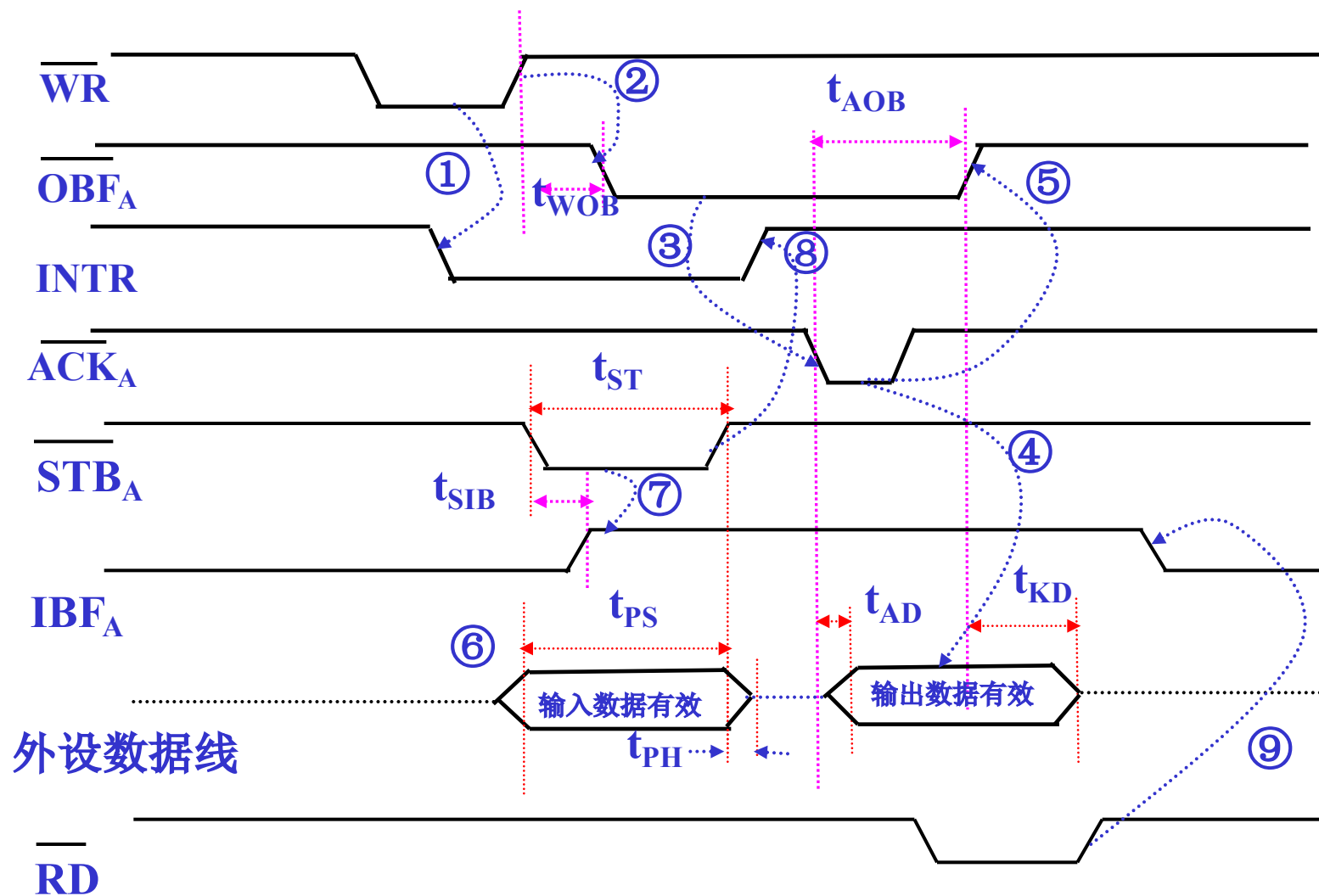
B口
(0--输出
1—输入)

B组方式
(0—方式0, 1—方式1)



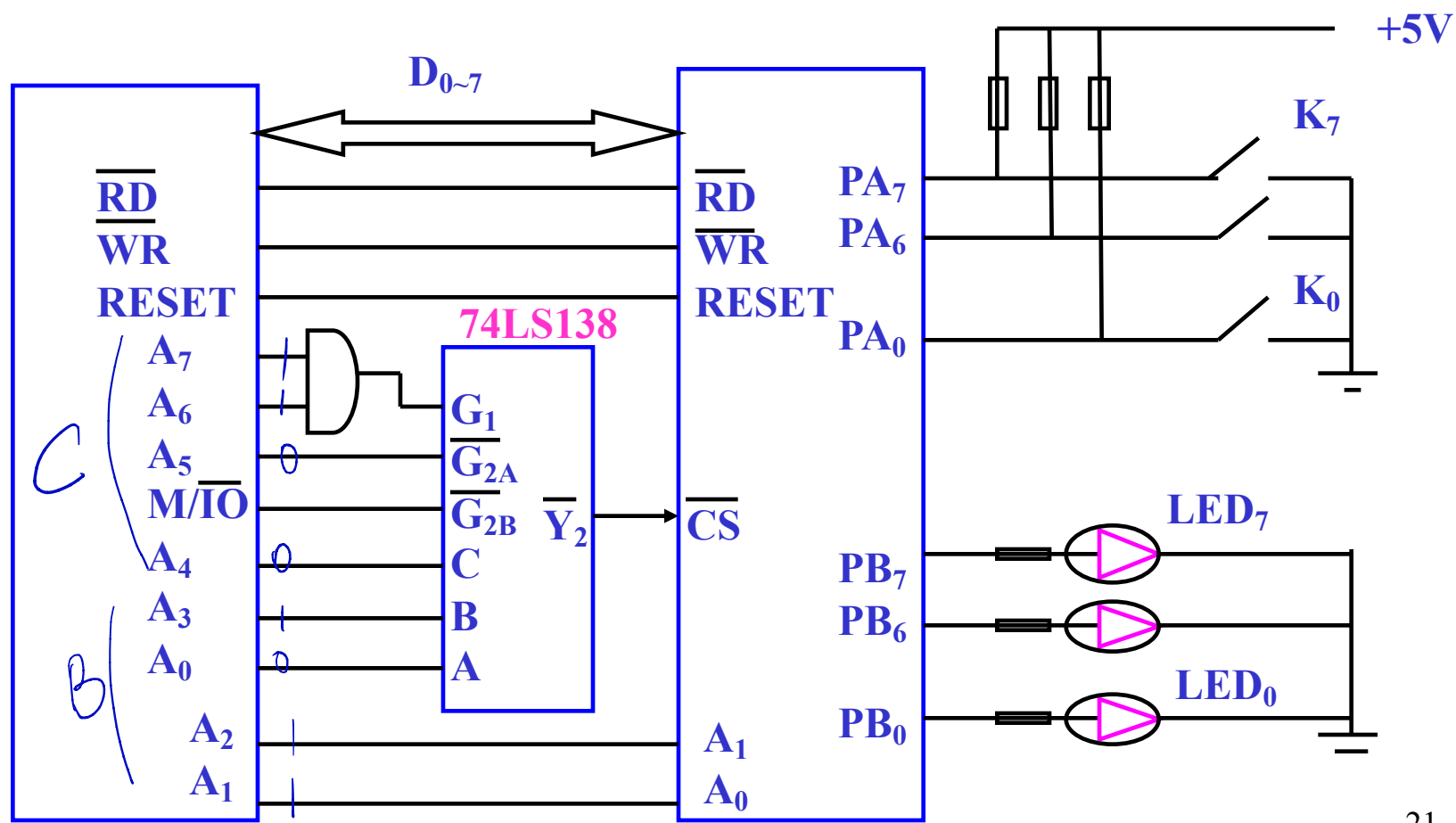
端口A方式2引脚配置

(1) 方式2 的输出时序



8255A应用举例

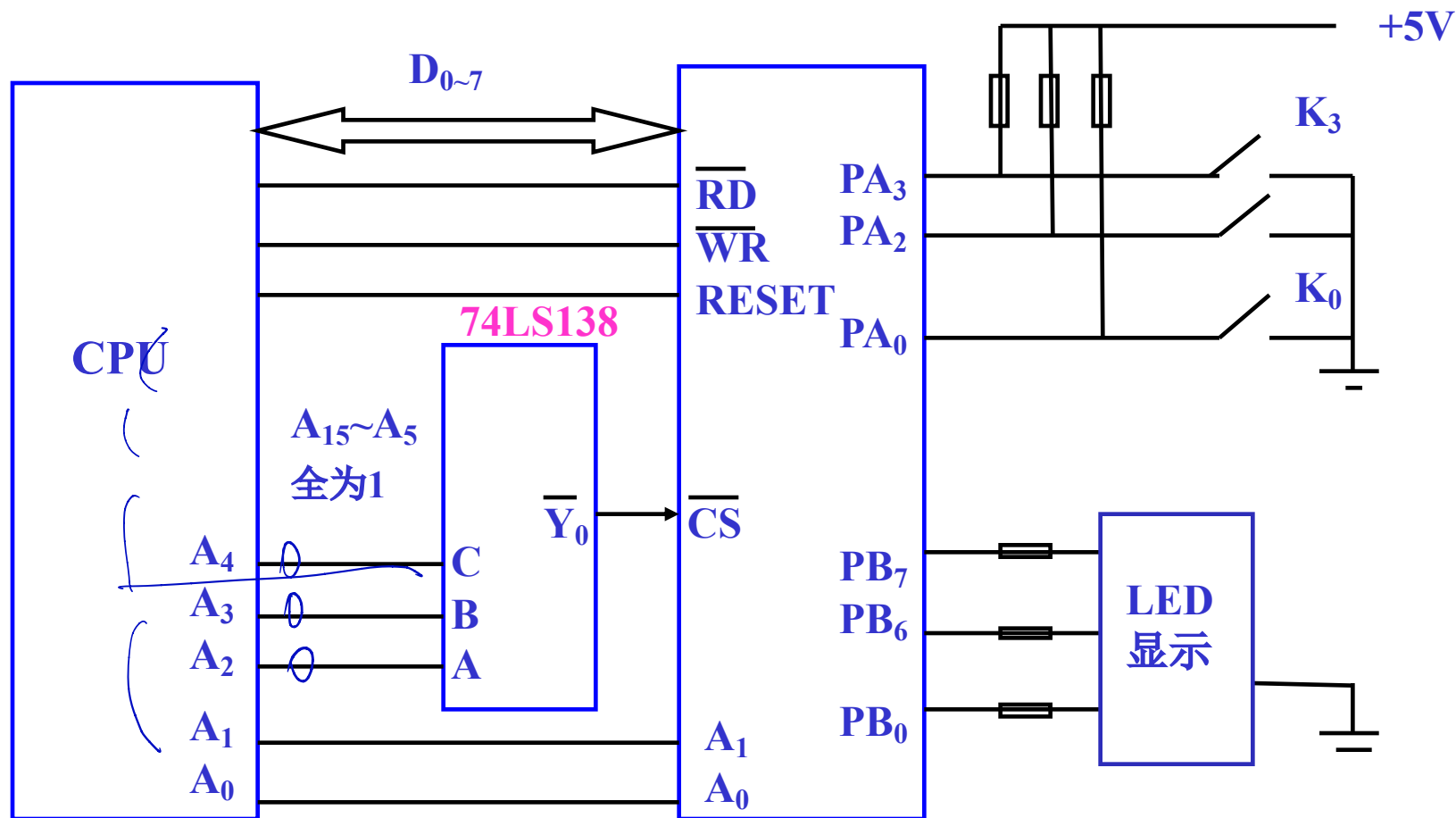
例1：某8086微机系统中有一片8255A，其端口A的PA₇₋₀接8个开关K₇₋₀，端口B的PB₇₋₀接8个发光二极管LED₇₋₀。A口、B口均工作于方式0，硬件连接如下图所示，要求实现编程，将开关状态K₇₋₀送入LED₇₋₀循环显示。



解： 1) 8255A的4个端口地址分别为0C8H、0CAH、0CCH、0CEH，对应于A口、B口、C口和控制口。
2) 8255A的控制字为90H。

	MOV AL, 90H	； 8255A的控制字
	OUT 0CEH, AL	； 控制字写入控制器
DISPLAY:	IN AL, 0C8H	； 从A口读开关状态
	OUT 0CAH, AL	； 将开关状态送B口上显示
	MOV CX, 200H	； 显示延时常数
DELAY:	DEC CX	
	JNZ DELAY	
	JMP DISPLAY	

例2：8255A端口A的PA₃₋₀接4个开关K₃₋₀，端口B的PB₇₋₀接共阴极LED显示器。A口、B口均工作于方式0，硬件连接如下图所示，要求实现编程，将开关状态K₃₋₀对应的16进制数字送入LED₇₋₀显示0~F。



题意：1) 8255A的4个端口地址分别为0FFE0H~ 0FFE3H，对应于A口、B口、C口和控制口。

2) 8255A的控制字为90H。

```
DATA    SEGMENT
LIST    DB      3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH,
              77H,7CH,39H,5EH,79H,71H
```

```
DATA    ENDS
```

```
CODE    SEGMENT
```

```
ASSUME CS:CODE,DS:DATA
```

```
START:  MOV     AX, DATA
```

```
        MOV     DS, AX
```

```
        MOV     AL, 90H
```

; 90H----10010000

```
        MOV     DX, 0FFE3H
```

```
        OUT     DX, AL
```

```
L0:     MOV     DX, 0FFE0H
```

; 从A口读入开关值

```
        IN      AL, DX
```

```
        AND     AL, 0FH
```

; 屏蔽A口高4位

```
        MOV     BX, OFFSET LIST
```

; 查表得到LED编码

```
        AND     AX, 00FFH
```

```
        ADD     BX, AX
```

```
        MOV     AL, [BX]
```

```
        MOV     DX, 0FFE1H
```

; 输出到B口

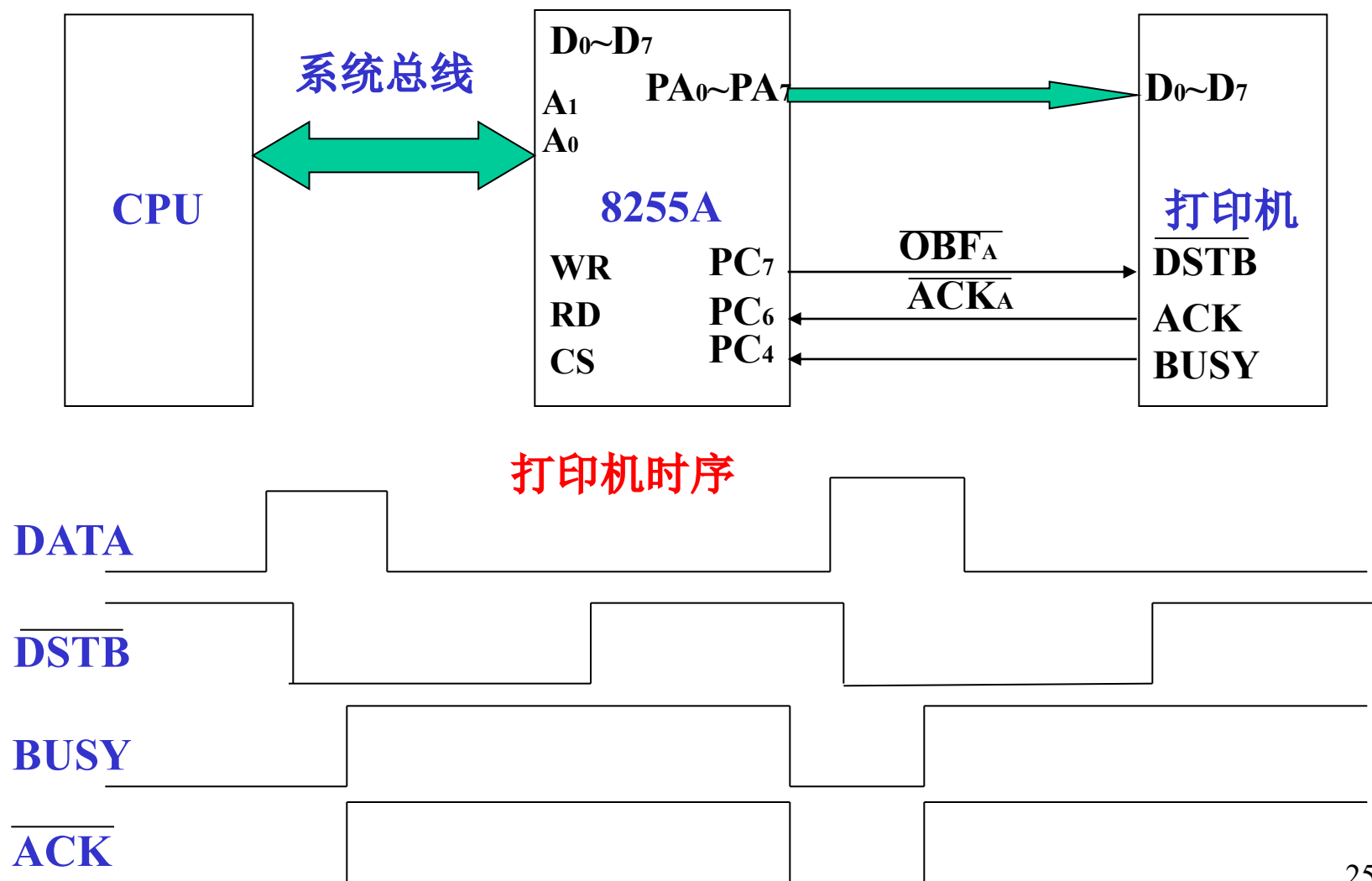
```
        OUT     DX, AL
```

```
        CALL    DAILLY
```

```
        JMP     L0
```

```
...
```


例3：用8255A作为连接打印机的接口,A口工作于方式1，查询方式将内存BUFFER中的100个字节送打印机输出。设8255A的控制口地址为43H，数据口A、B、C的地址分别为40H、41H、42H。请编写程序以查询方式实现打印的功能。



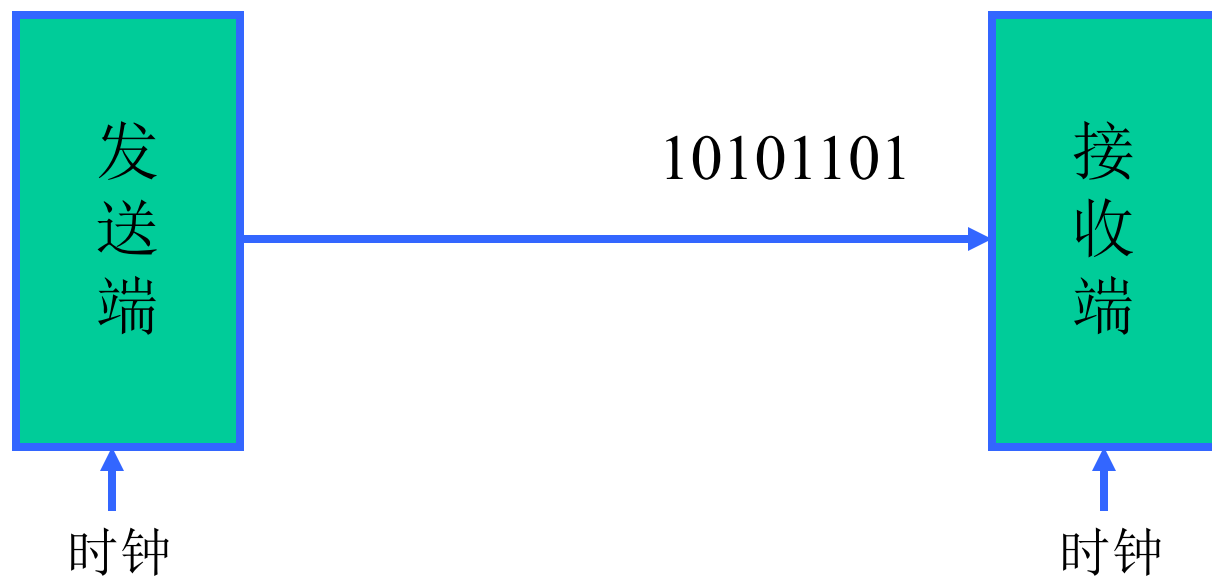
工作原理：当主机开始打印输出时，先测试打印机忙（BUSY）信号，如果BUSY=1，表示打印机正在打印，如果BUSY=0，则主机可以向打印机输出。

```
...
MOV    AL, 0A8H                ; 0A8H----10101000
OUT     43H, AL                 A口方式1，PC4、PC6方式0输入
MOV     CX, 100                ; 传送字节数100
MOV     SI, OFFSET BUFFER      ; SI为内存首地址
L1:     IN      AL, 42H          ; 从C口读入状态
        TEST    AL, 10H
        JNZ     L1              ; BUSY=1则继续查询
        MOV     AL, [SI]
        OUT     40H, AL         ; A口输出数据
        INC     SI
        LOOP    L1
...
```

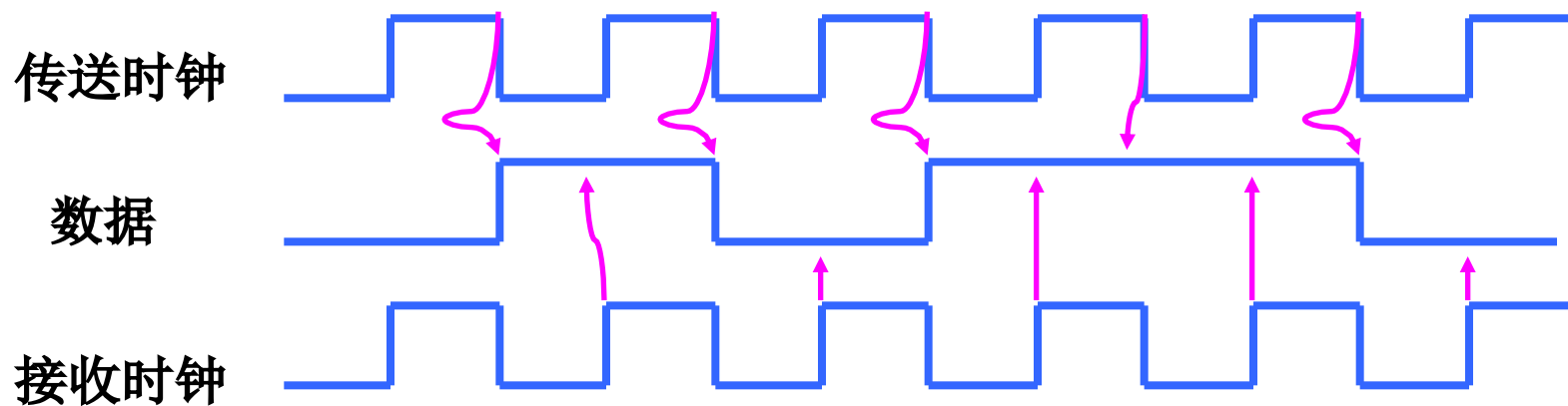
■ 串行通信

特点：每个字符的所有位是按位传送。

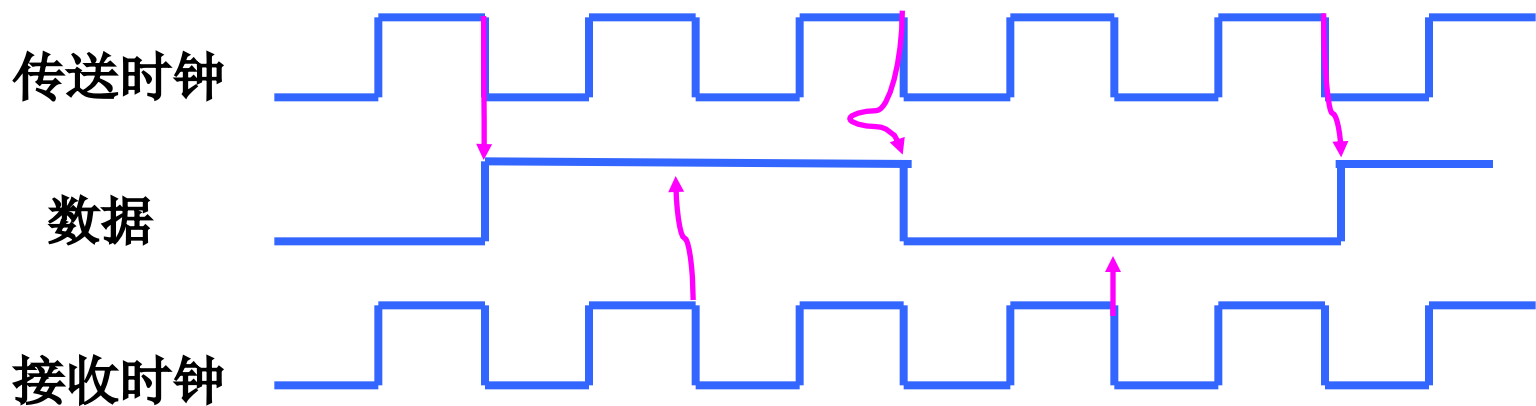
字符的每一位放在传输线上和采样传输线的状态，随着系统的不同而不同。



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。



每个时钟传送一个位数据



每二个时钟传送一位数据

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：

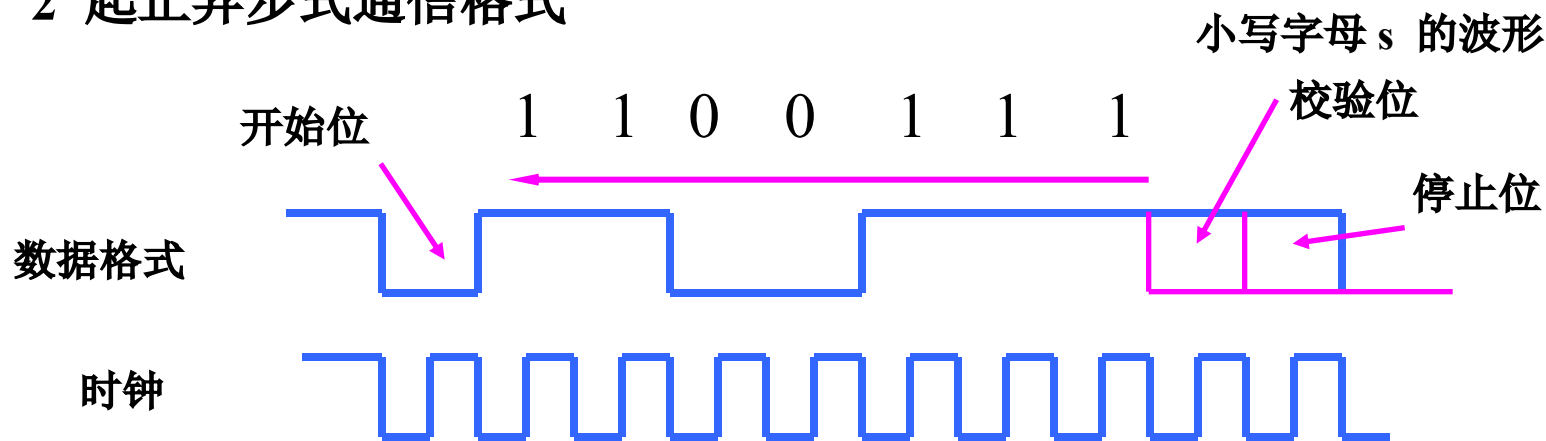
- 异步传送方式
- 同步传送方式

● 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

● 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。

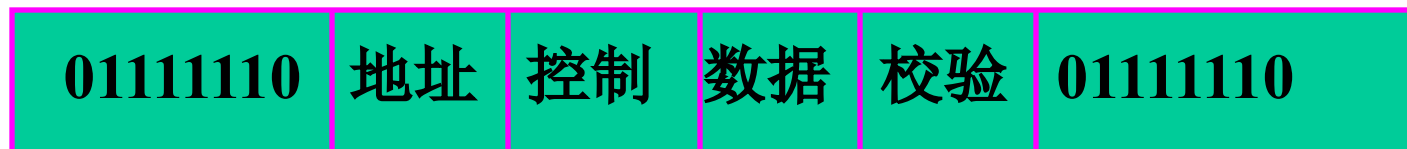
SYN	SYN	SOH	标题	STX	数据块	ETB/ETX	块校验
-----	-----	-----	----	-----	-----	---------	-----

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）



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，可通过编程设置。

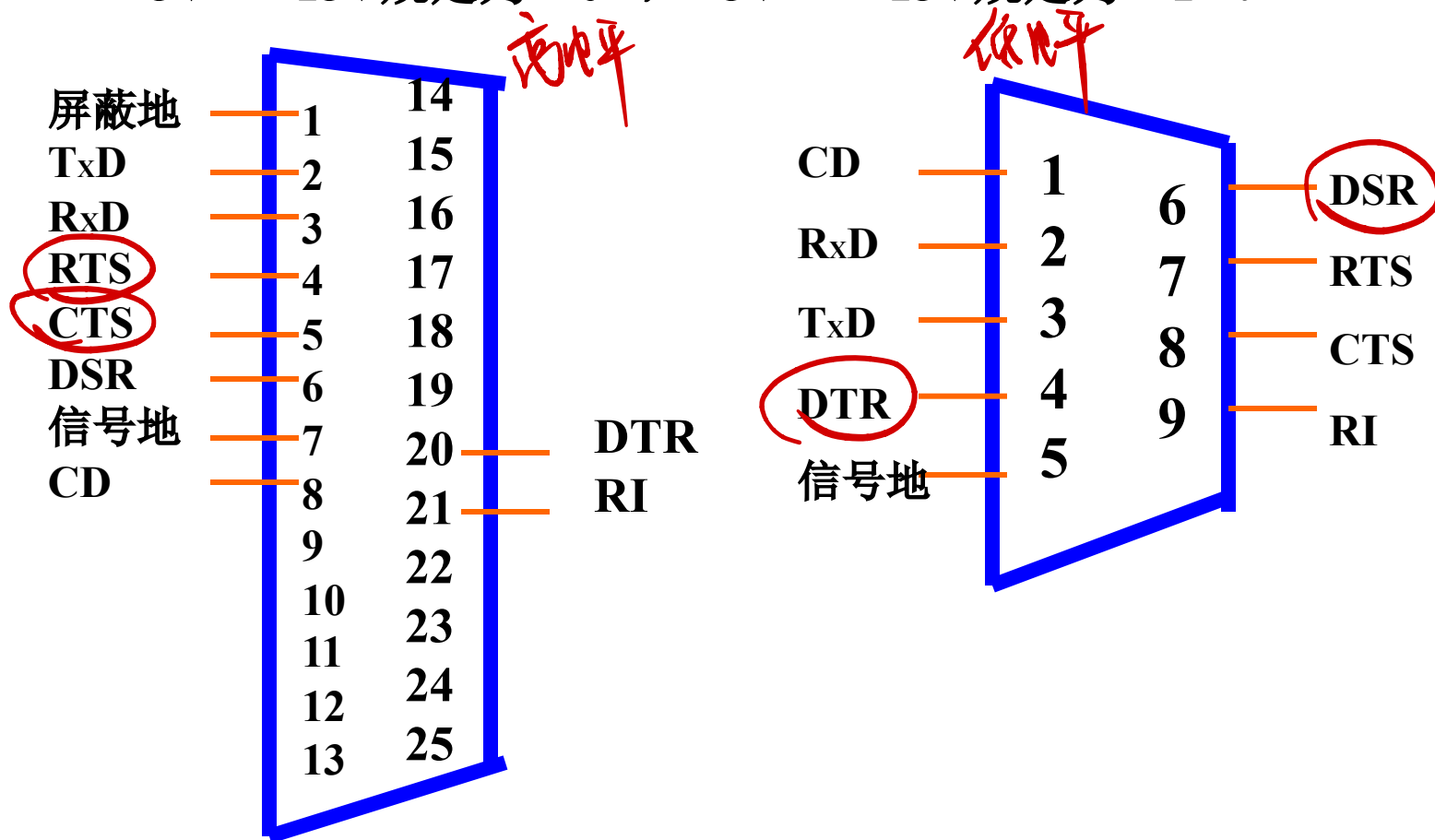
■ 串行通信的接口标准

● 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

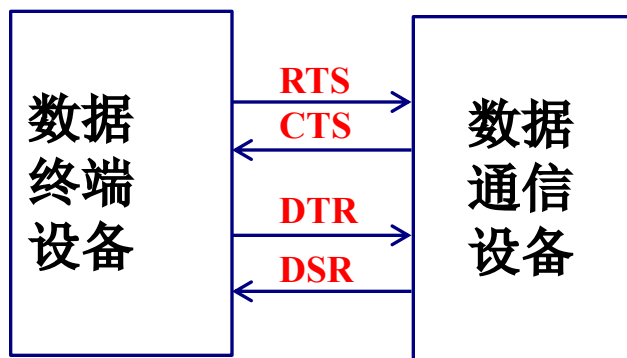
RxD：接收数据

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。

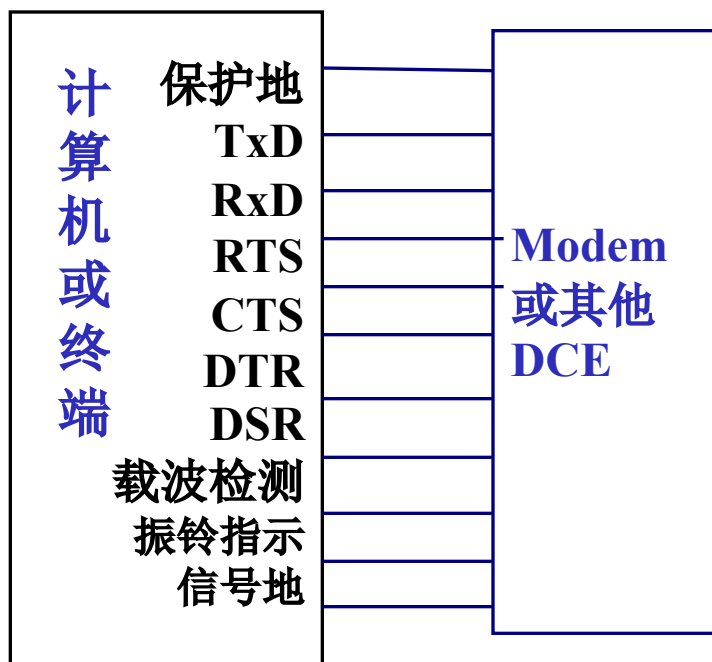
DSR：数据通信设备准备就绪。



● RS-232连接

1 使用Modem连接

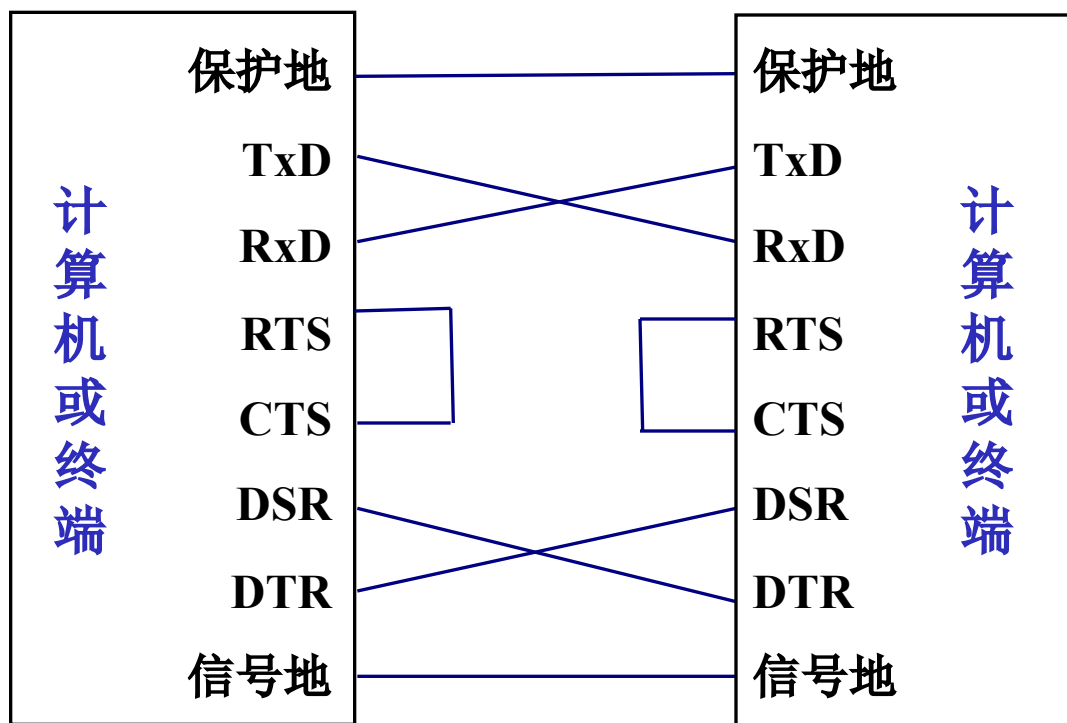
计算机通过Modem或其他数据通信设备（DCE）使用一条电话线通信时，一般只需要9个常用信号。



● RS-232连接

2 直接连接

计算机不通过Modem或其他数据通信设备（DCE），直接通过RS-232C接口连接，一般只需要5个常用信号。

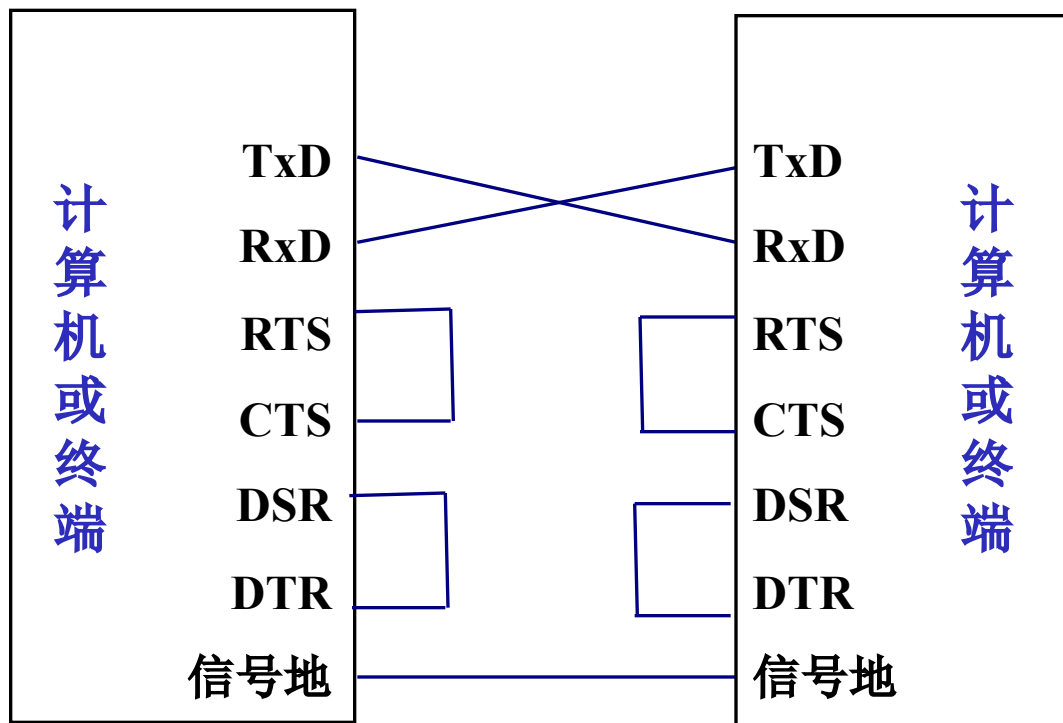


一个发，一个接
近距离

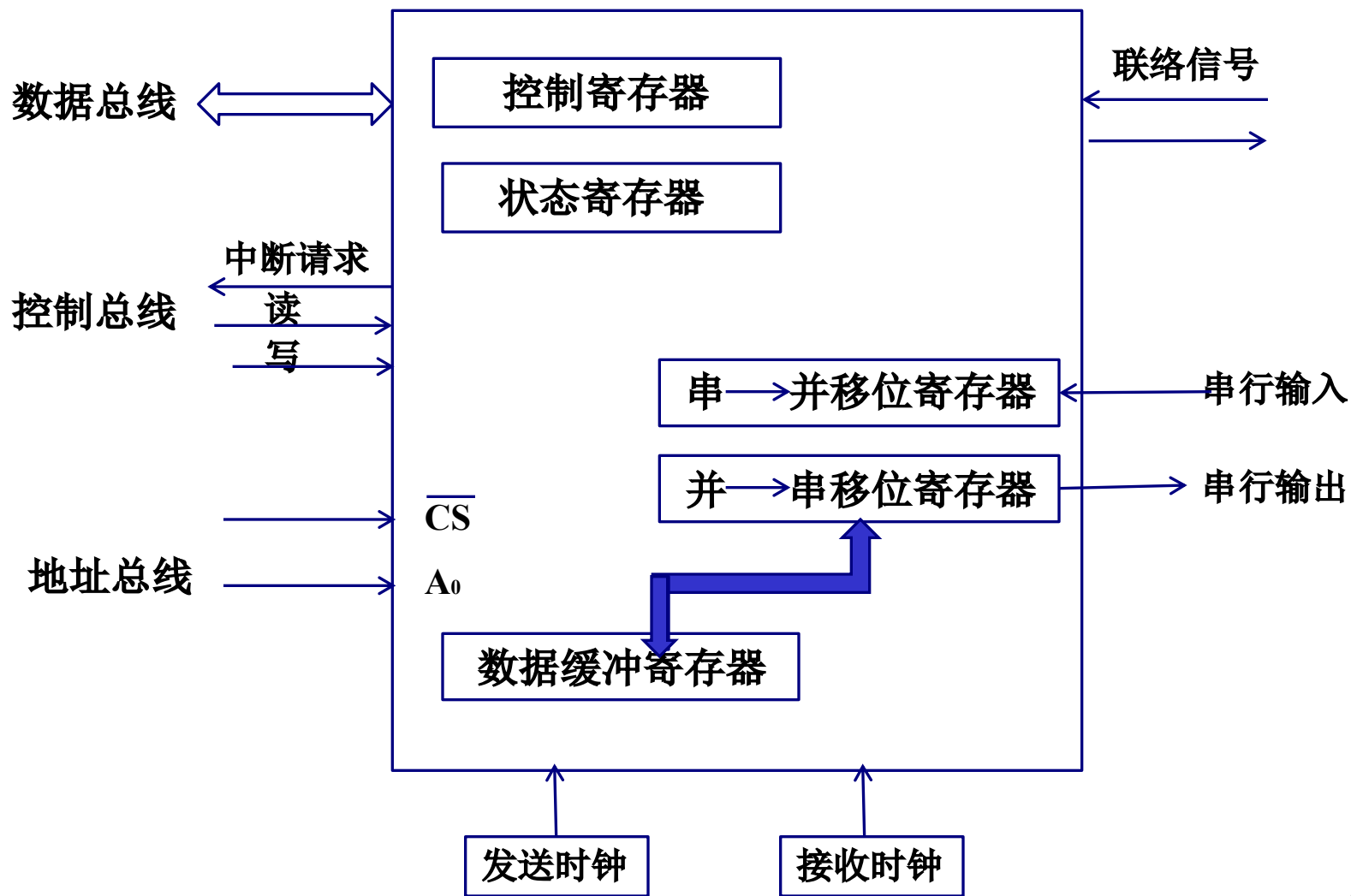
● RS-232连接

3 三线连接法

最简单的连接，只用3根线。



■ 可编程串行接口的典型结构



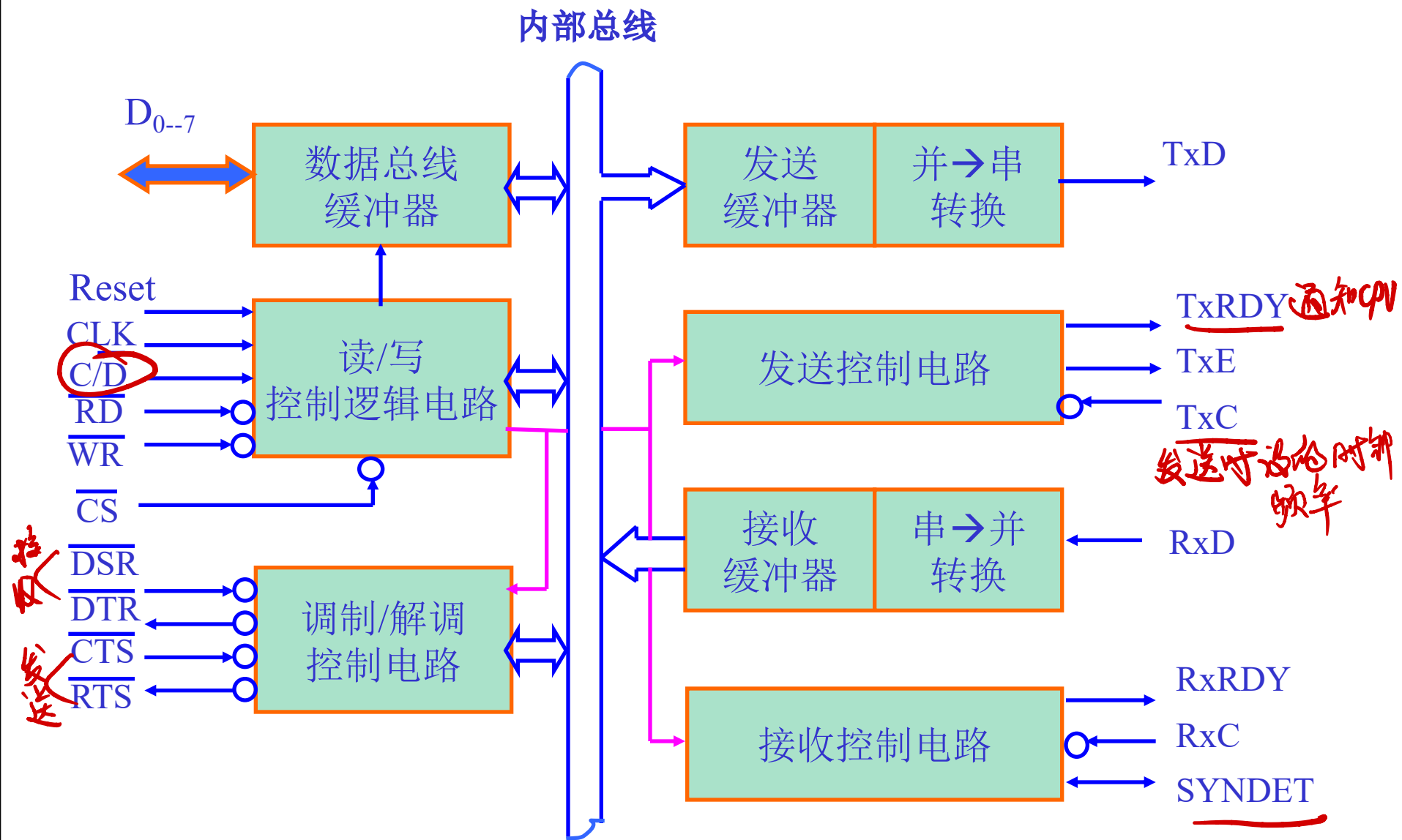
■ 可编程串行通信接口8251A

1 特点

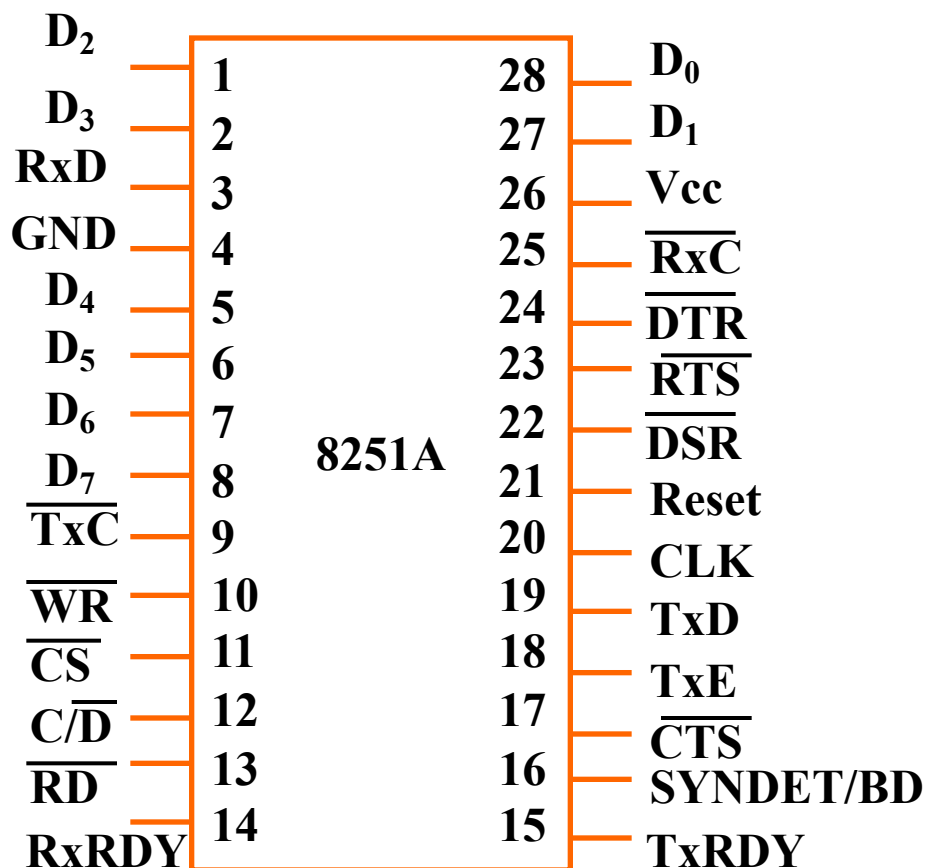
- 1) 可工作在全双工方式的串行通信;
- 2) 可用于同步和异步传送;
- 3) 同步传送: 5—8个比特代表字符, 内部或外部字符同步, 可自动插入一个或二个同步字符;
- 4) 异步传送: 5—8个比特代表字符, 波特率为时钟的1/1、1/16或1/64;
- 5) 具有奇偶、溢出和帧错误等检测电路;
- 6) 可产生1、1.5或2个停止位, 可检测假启动位, 自动检测和处理停止位。

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

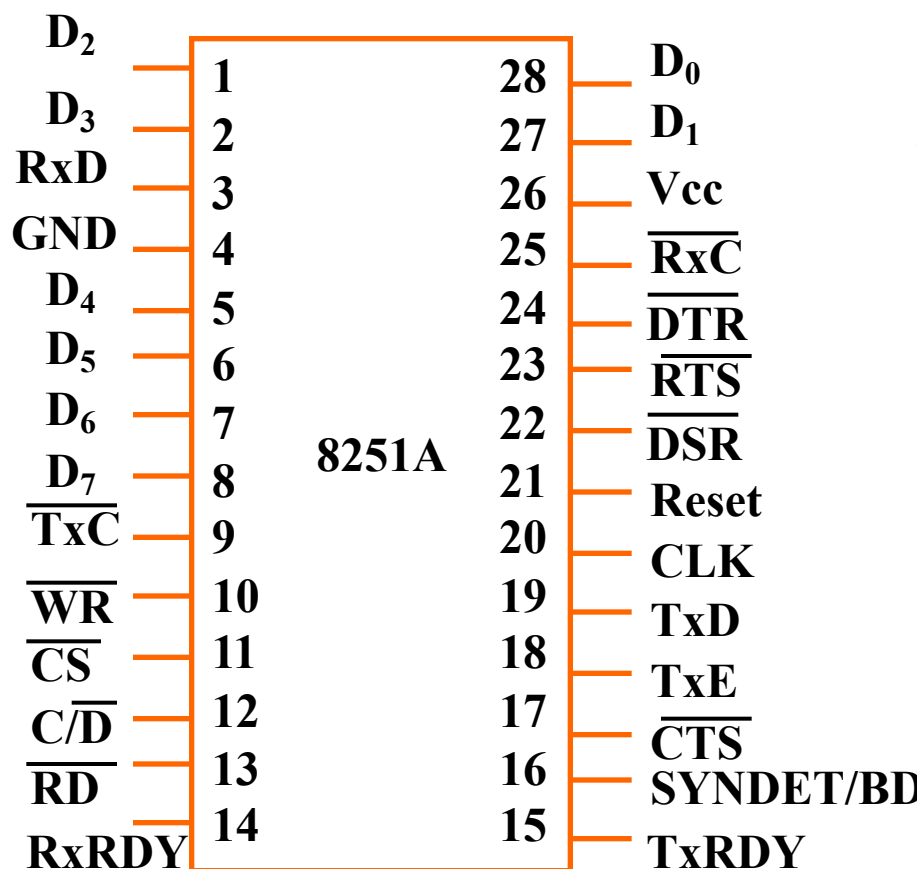
片选信号；

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；

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

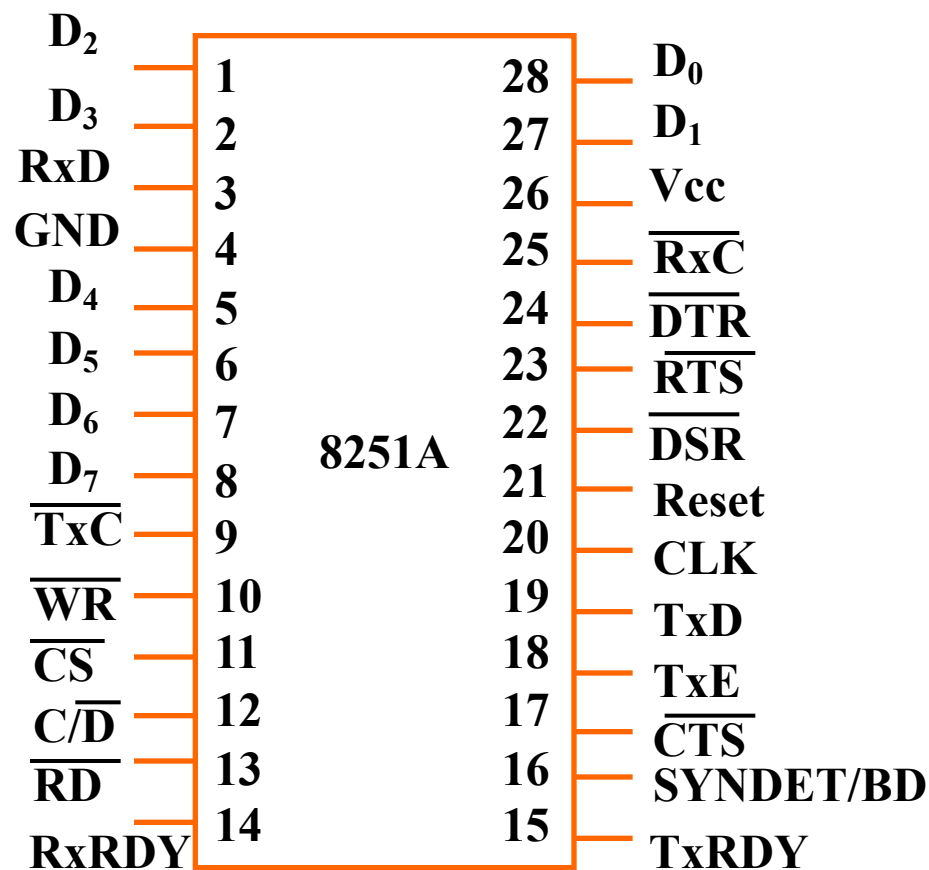
接收准备好，用于通知CPU，8251已经接收了一个字符，CPU可以取走，8251每收到一个字符就使R_xRDY=1，CPU取走后又变为0；

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) T_xE

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T_xE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) T_xD

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

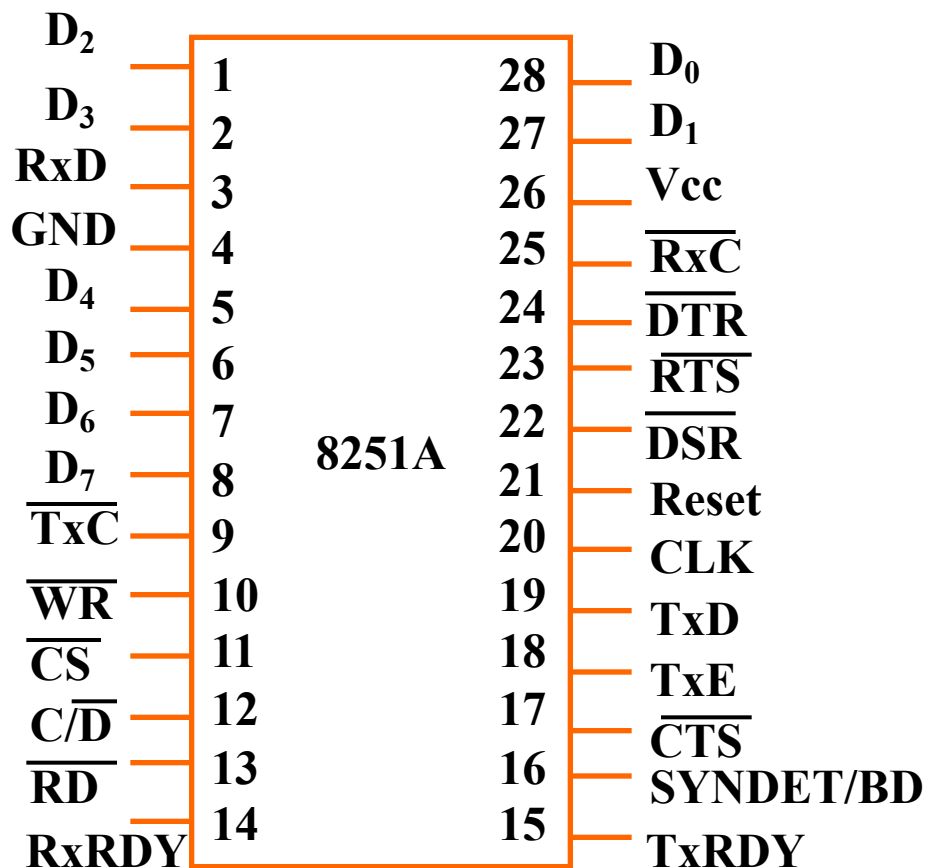
系统时钟；

14) RESET

复位信号；

15) DSR

MODEM来的信号，表示MODEM已准备好；



8251A引脚图

16) $\overline{\text{RTS}}$

8251 输出给MODEM，用于通知MODEM，8251准备传送数据；

17) $\overline{\text{DTR}}$

8251 输出给MODEM，用于通知MODEM，8251已准备好；

18) RxC

接收时钟；

19) V_{cc}

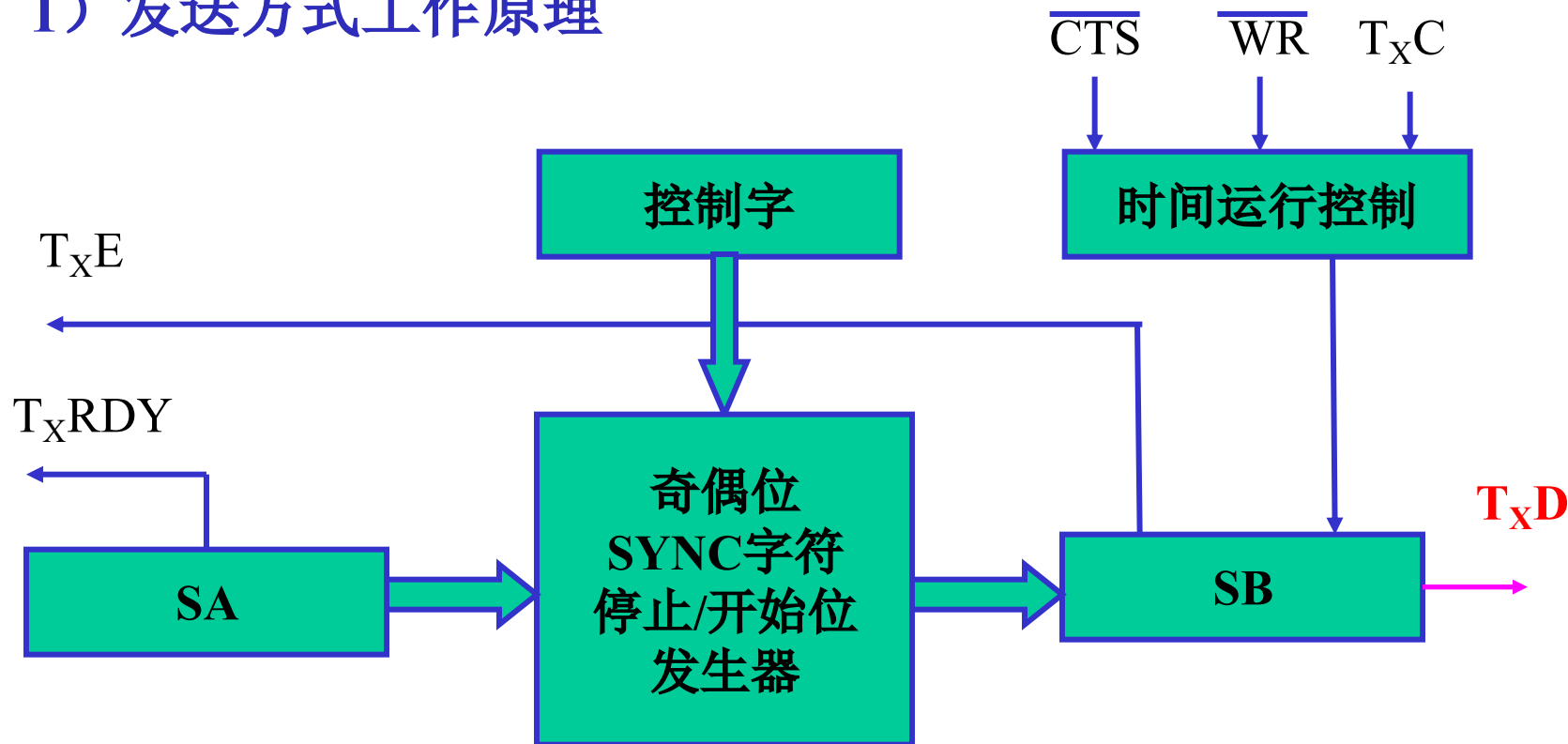
电源；

20) GND

信号地。

■ 8251的基本工作原理

1) 发送方式工作原理



- 控制字中的 $\text{T}_{\text{x}}\text{EN}=1$ 时，传送方式开始工作
- 此时 $\overline{\text{CTS}}=0$ ，CPU把要传送的数据放在寄存器SA中，然后送到SB中
- $\text{T}_{\text{x}}\text{C}$ 的控制下向外发出 $\text{T}_{\text{x}}\text{D}$
- 如果SB中没有数据，则 $\text{T}_{\text{x}}\text{D}=1$ ，即处于连续传送停止位的状态

■ 8251与发送有关的信号

TxD: 发送数据，CPU送入的并行数据转换成串行格式后，从TxD输出。

TxRDY: 发送数据准备好，TxRDY=1表示发送数据缓冲器已准备好从CPU接收一个字符数据，当收到CPU送来的字符后就变为低电平。CPU可采用中断或查询方式输出一个数据。

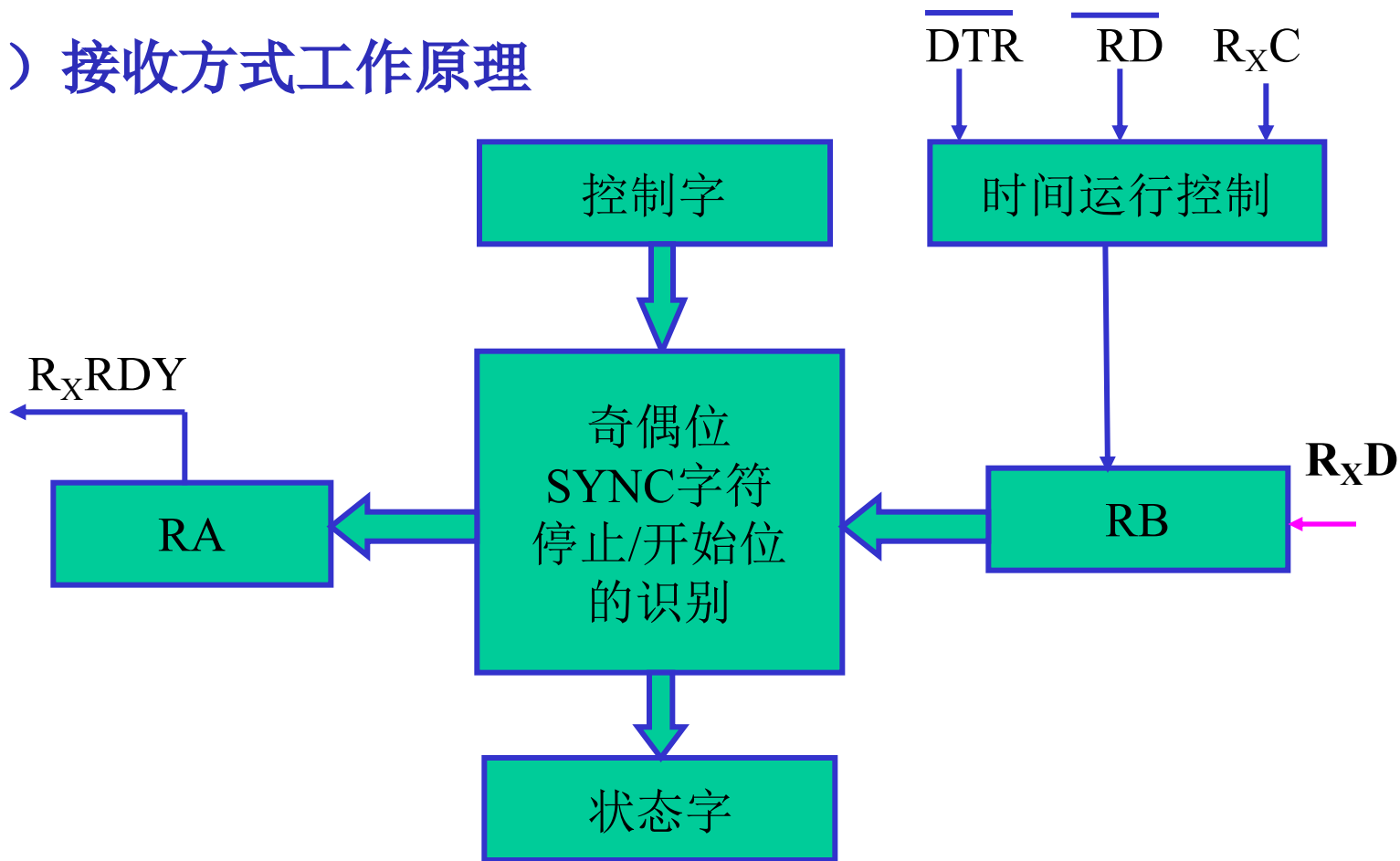
TxE: 发送器空，TxE=1表示发送移位寄存器中已无数据可以向外部发送。此时，由TxD输出空闲位（同步字符）。当8251从CPU接收到一个数据后，TxE=“0”。

TxC: 发送时钟，波特率和它的关系可由软件编程为：

同步
1/1

异步
1/1
1/16
1/64

2) 接收方式工作原理



- 控制字中的 $R_xEN=1$ 时，接收方式开始工作；
- 8251在 R_xC 的上升沿，且在一个位的中间采样 R_xD ；
- 接收端把RB中的数据去掉奇偶位、开始/停止位后移入RA中；
- $R_xRDY=1$ 用于通知CPU,寄存器RA中已有一个字符，当CPU取走这字符后 R_xRDY 又变为低电平；

■ 8251与接收有关的信号

RxD: 接收数据，从RxD端 串行输入数据输入

RxRDY: 接收数据准备好，表示接收数据缓冲器已收到一个字符数据，CPU可采用中断或查询方式读取数据。

SYNDET: 同步检测

◆**内同步方式:** 8251检测到同步字符， SYNDET输出为高电平，表明8251A已处于同步状态。CPU执行一次读操作后， SYNDET自动复位；

◆**外同步方式:** SYNDET引脚输入一个正跳变脉冲，8251在下一个接收时钟的下降沿开始接收字符，一旦达到同步， SYNDET引脚的高电平即可撤除。

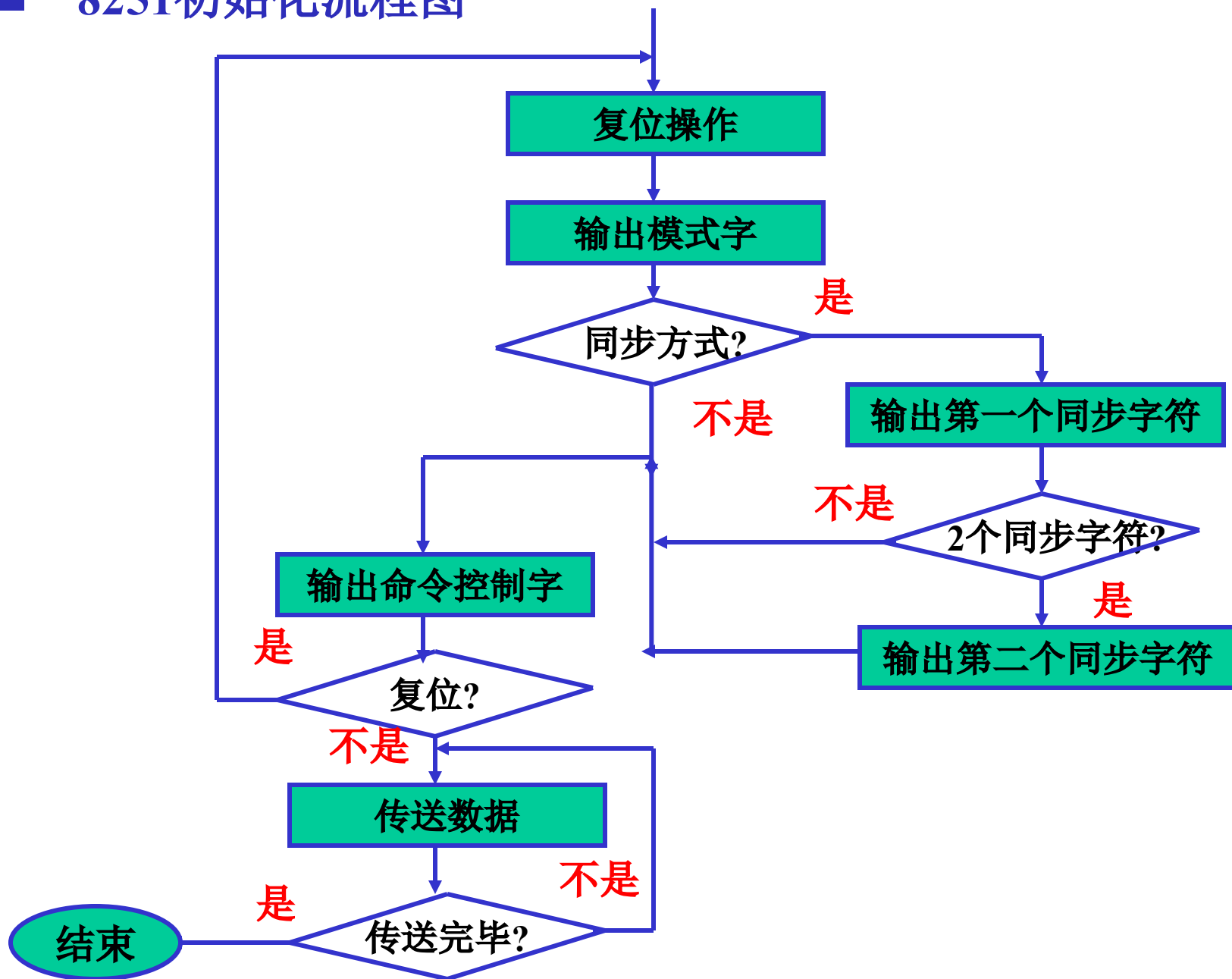
RxC: 接收时钟

8251A可以检查三种错误:

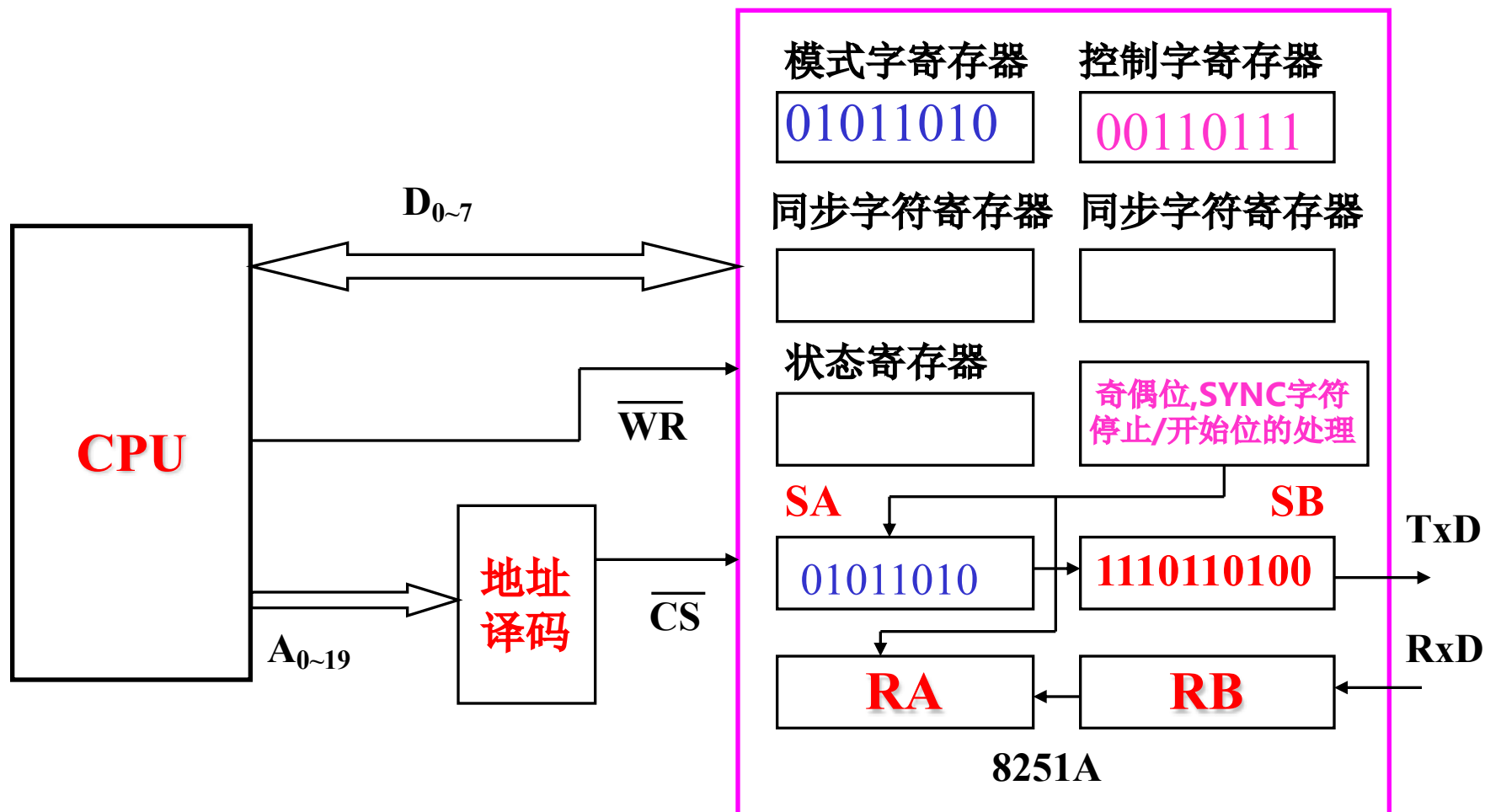
- **帧格式错:** 在异步通信中, 如果没有发现停止位, 则出现顺序错;
- **覆盖错:** 当CPU取走RA中的字符前, 若又有一个新的字符送入RA, 则出现超时错, 此时RA中的老字符被替代;
- **奇偶错:** 奇偶校验错误

CPU可以通过对状态寄存器的相应位进行检测来发现这些错误, 在出现错误时并不影响接收端的工作。

■ 8251初始化流程图



■ 8251的初始化过程图示



■ 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的模式字

S ₂ /SCS	S ₁ /ESD	EP	PEN	L ₂	L ₁	B ₂	B ₁
---------------------	---------------------	----	-----	----------------	----------------	----------------	----------------

异步方式

		停止位个数
0	0	不确定
0	1	1 位
1	0	1.5 位
1	1	2 位

		工作方式	波特率因子
0	0	同步	1
0	1	异步	1
1	0	异步	16
1	1	异步	64

同步方式

	SYNC字符个数
0	2
1	1

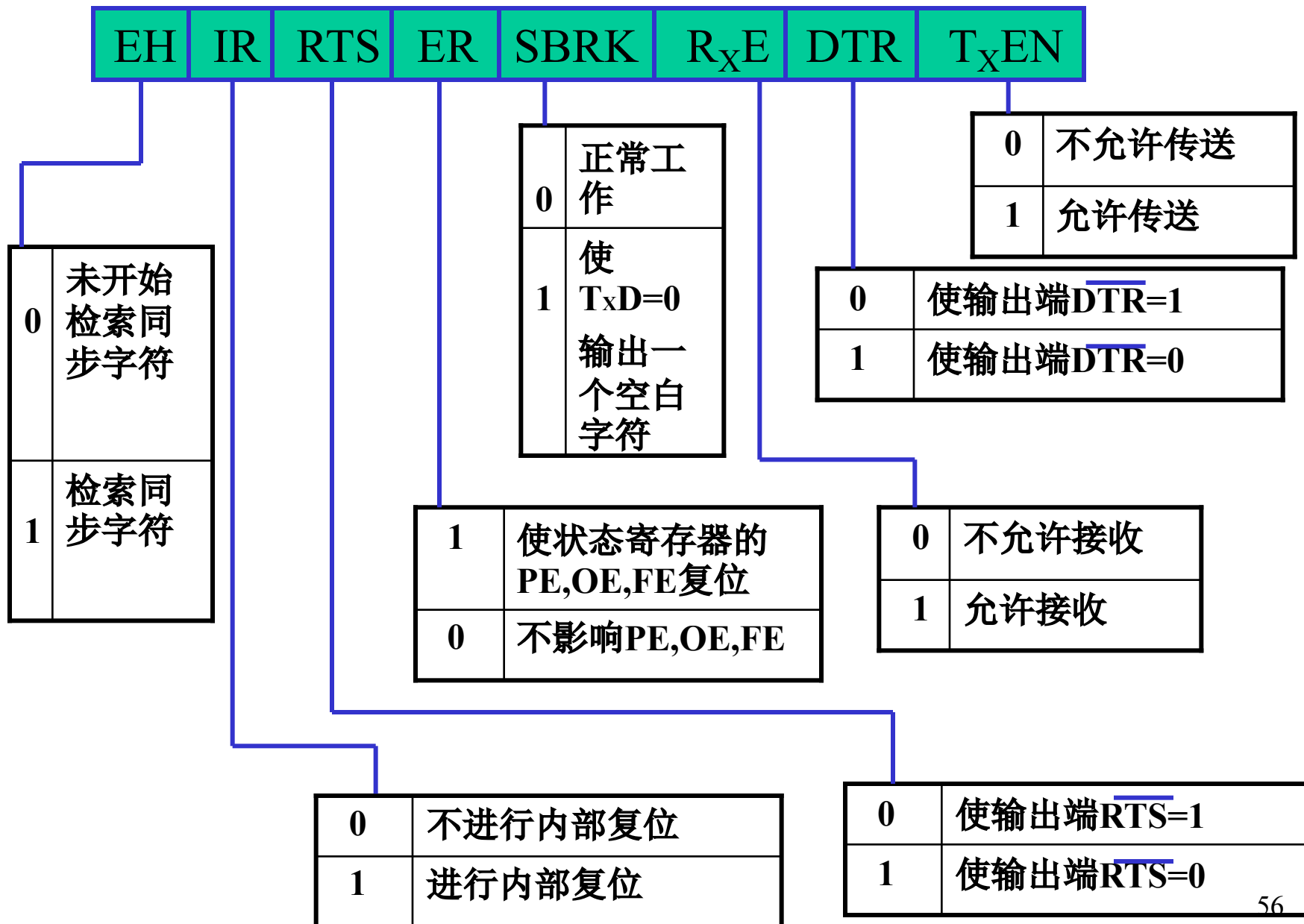
	奇偶方式
0	奇
1	偶

	奇偶校验位
0	无
1	有

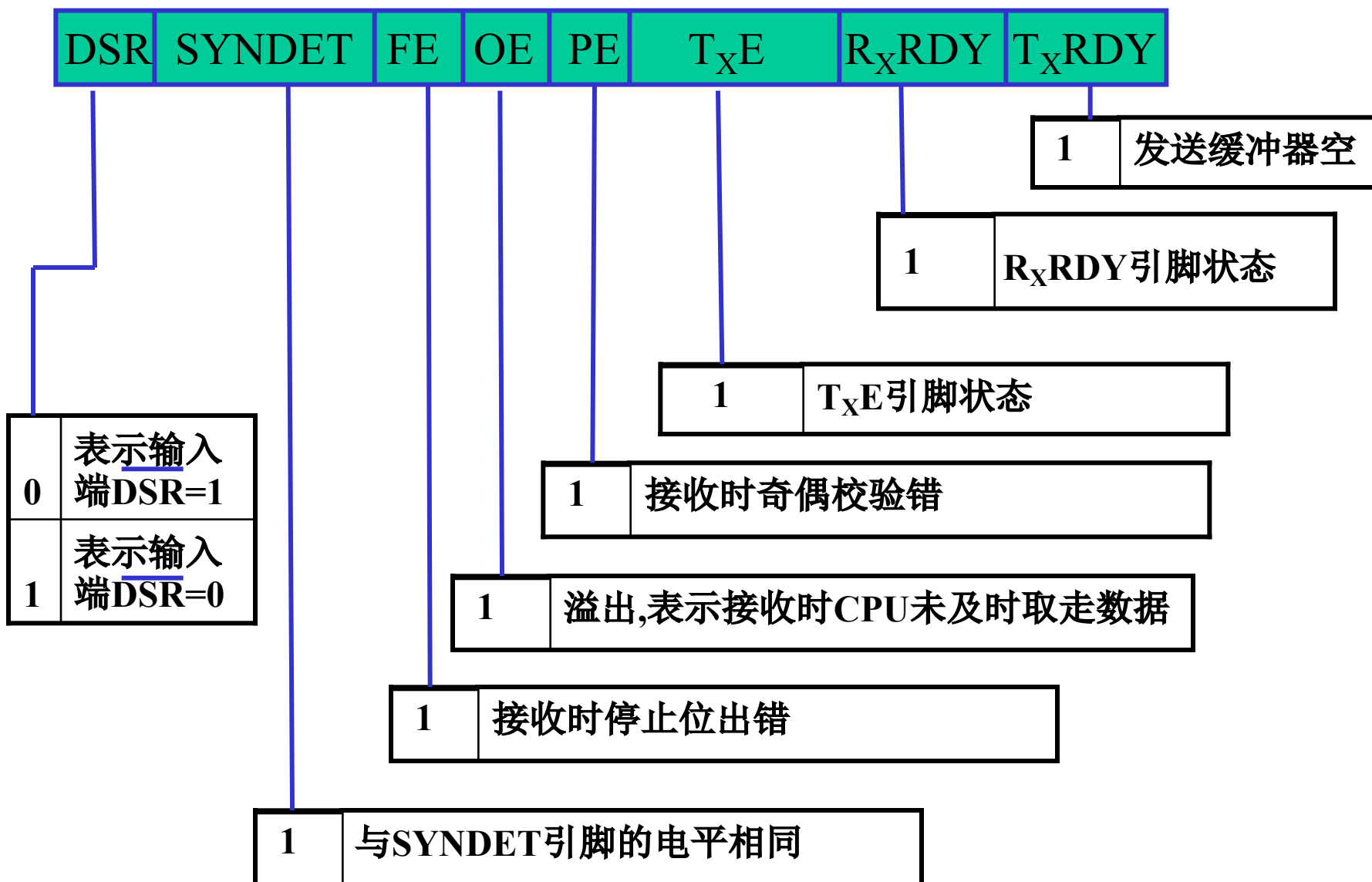
		字符长度
0	0	5 位
0	1	6位
1	0	7位
1	1	8位

	字符同步
0	内部
1	外部

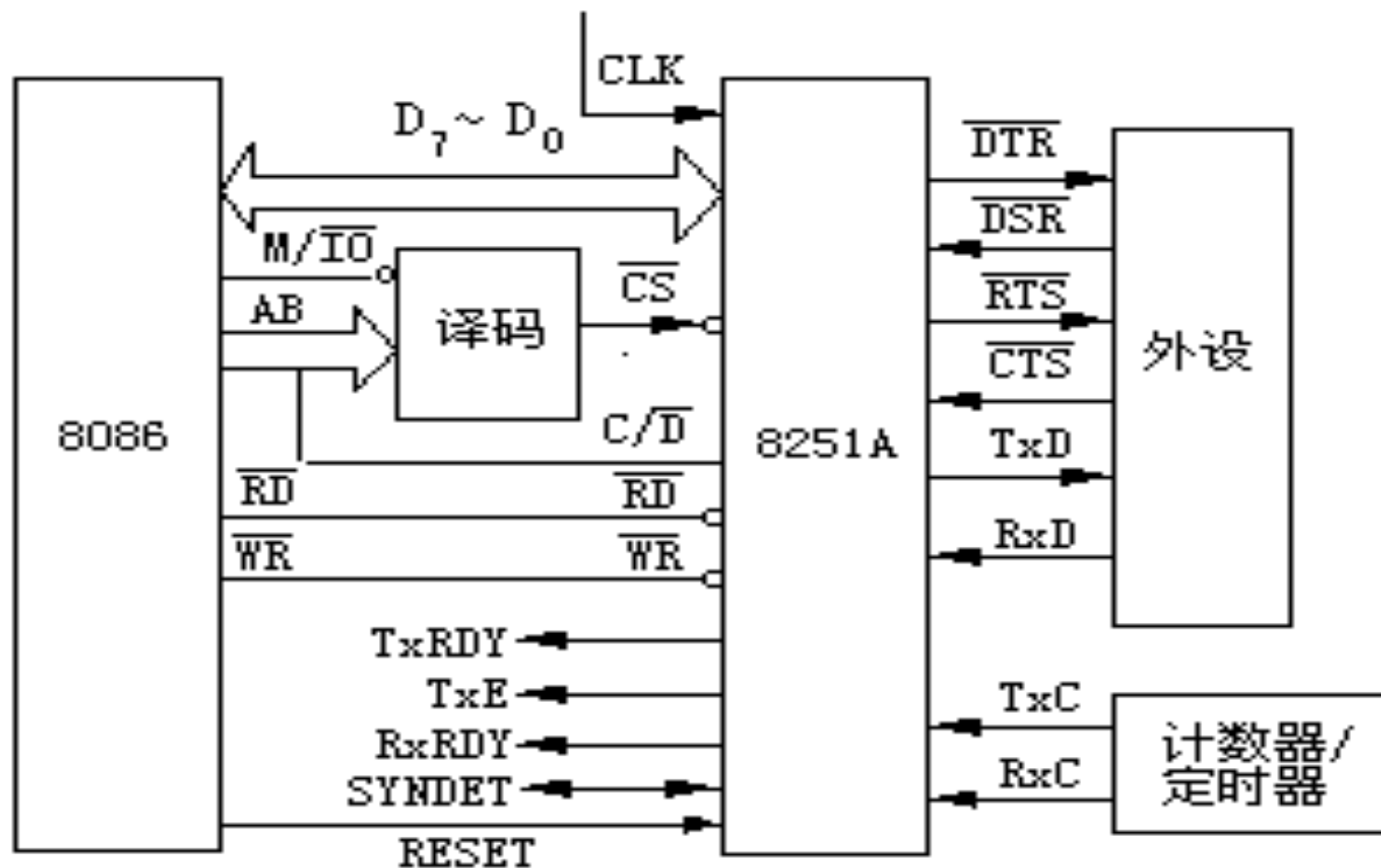
■ 8251的控制字



■ 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
```

； 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**

S ₂ /SCS	S ₁ /ESD	EP	PEN	L ₂	L ₁	B ₂	B ₁
---------------------	---------------------	----	-----	----------------	----------------	----------------	----------------

； 设置控制字，使发送、接收允许，清出错
标志 **00110111**

EH	IR	RTS	ER	SBRK	R _X E	DTR	T _X EN
----	----	-----	----	------	------------------	-----	-------------------

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
```

```
OUT 42H, AL
```

```
MOV AL, 16H
```

```
OUT 42H, AL
```

```
OUT 42H, AL
```

```
MOV AL, 97H
```

```
OUT 42H, AL
```

； 设置模式字，同步模式，用2个同步字符
7个数据位，偶校验

00111000

； 送同步字符16H

； 设置控制字，使接收器启动

10010111

S ₂ /SCS	S ₁ /ESD	EP	PEN	L ₂	L ₁	B ₂	B ₁
---------------------	---------------------	----	-----	----------------	----------------	----------------	----------------

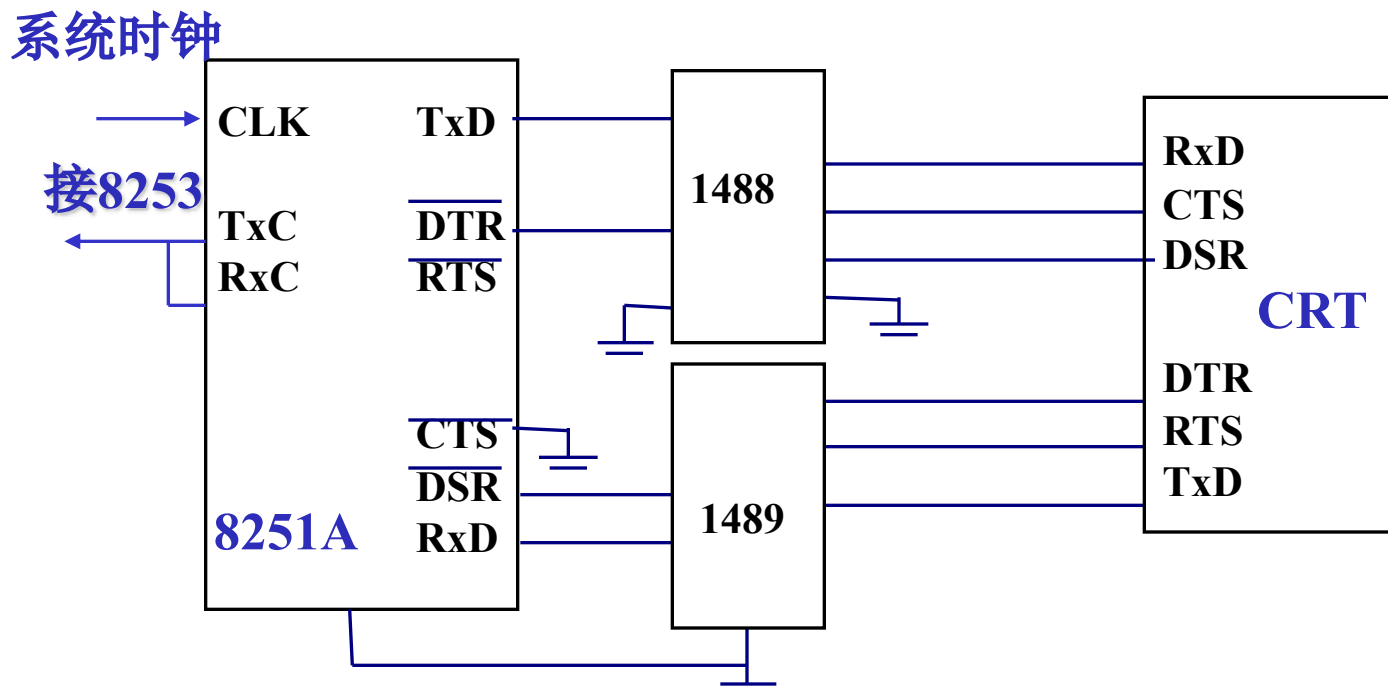
EH	IR	RTS	ER	SBRK	R _x E	DTR	T _x EN
----	----	-----	----	------	------------------	-----	-------------------

■ 8251编程举例

例1：要求异步方式下，通过查询8251A状态字，从外设输入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, 35H	
OUT	52H, AL	; 控制字写入奇地址端口 00110101
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R _x RDY
JZ	BEGIN	
IN	AL, 50H	; R _x RDY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:		

例2：微型机系统中，8251A作为CRT显示器和键盘的串行通信接口。



8251A实现与CRT的串行通信

8251复位

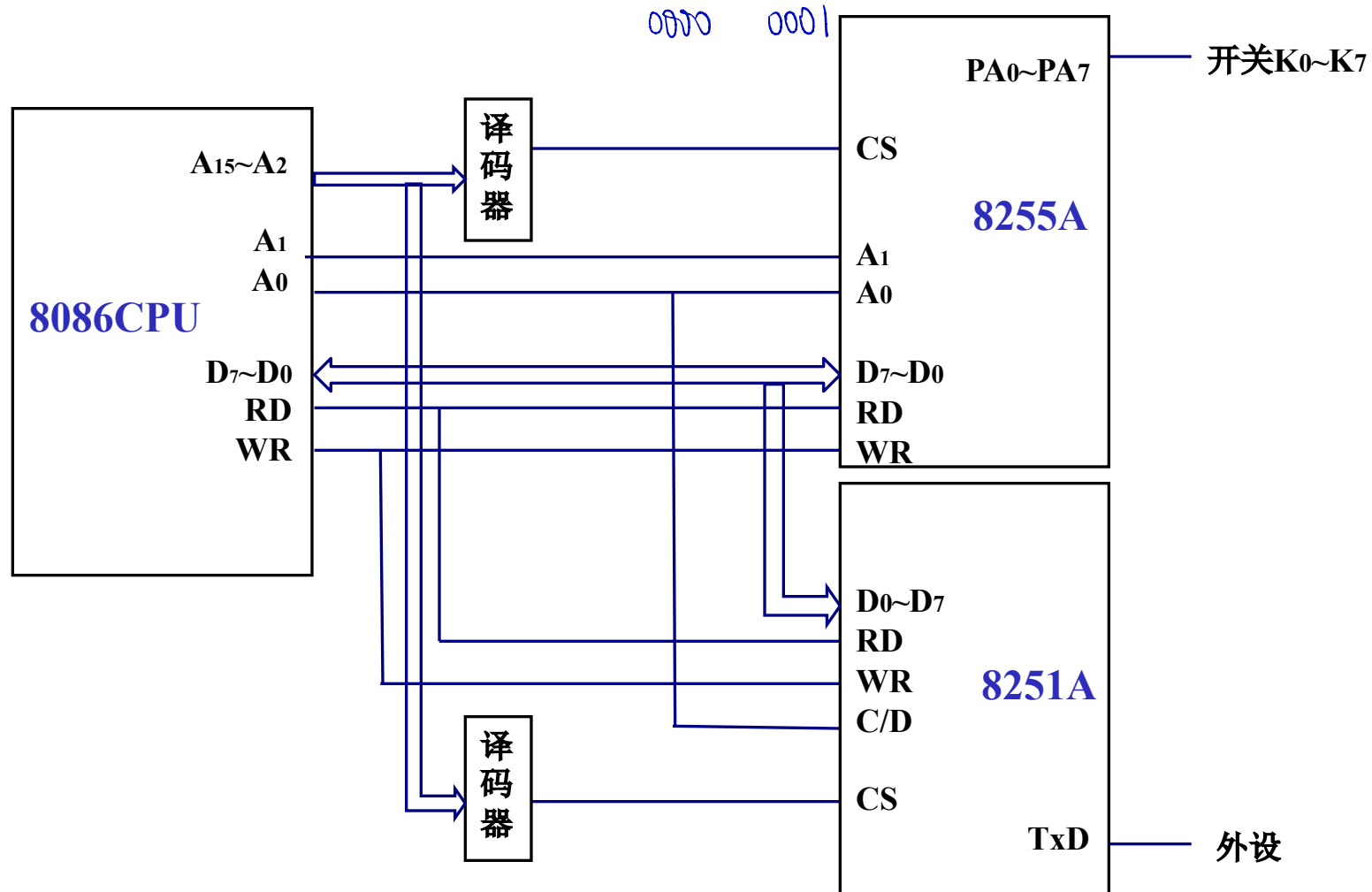
电源上电，8251自动进入复位状态，为了确保在送模式字和命令字之前8251正确复位，可向8251的控制口连续写入3个“0”，然后再写入一个复位控制字（40H），用软件使8251可靠复位。

注：对8251的控制口进行一次写入操作，需有写恢复时间。（写恢复时间一般为16个时钟周期）

```
DELAY1:  MOV  CX, 02H  
D0:      LOOP D0
```

INIT:	XOR	AX, AX	
	MOV	CX, 0003	
	MOV	DX, 00DAH	
OUT1:	CALL	KKK	; 向8251A的控制口送入3个0
	LOOP	OUT1	
	MOV	AL, 40H	; 向8251A的控制口送入1个40H, 复位
	CALL	KKK	
	MOV	AL, 4EH	; 设置模式字, 4EH 0100 1110
	CALL	KKK	
	MOV	AL, 27H	; 设置命令字, 27H <u>0010 0111</u>
	CALL	KKK	
	...		
KKK:	OUT	DX, AL	
	PUSH	CX	; 输出子程序
	MOV	CX, 0002	
ABC:	LOOP	ABC	; 等待输出动作完成
	POP	CX	
	RET		; 恢复CX内容, 并返回
CHAROUT:	MOV	DX, 0DAH	; 往CRT输出一个字符程序
STATE:	IN	AL, DX	
	TEST	AL, 01H	; 测试状态位TxRDY是否为1
	JZ	STATE	
	MOV	DX, 0D8H	
	POP	AX	; 需输出的字符已事先保存在堆栈, 现弹出到AX
	OUT	DX, AL	

例3：设计一个应用系统，要求8255A的端口A输入8个开关信息，并通过8251A以串行方式将开关信息发送出去。8251A采用异步查询方式，波特率因子为16，数据长度为8位，2位停止位，奇校验。已知8255的端口地址为100H~103H，8251A的端口地址为50H~51H。



8255A初始化:

```
MOV    AL, 90H
MOV    DX, 103H
OUT    DX, AL
```

8251A初始化:

```
MOV    AL, 0DEH
OUT    51H, AL
MOV    AL, 31H
OUT    51H, AL
LOP:   MOV    DX, 100H
        IN     AL, DXH
        MOV    BL, AL
L1:    IN     AL, 51H
        TEST   AL, 01H
        JZ     L1
        MOV    AL, BL
        OUT    50H, AL

        JMP    LOP
ENDD:   HLT
```

1001000 A口工作方式0, 输入

11011110 异步方式、波特率因子=16、
8个数据位, 奇校验、2个停止位

00110001 设置控制字发送允许,
出错标志复位

```
; 如果按 'Q' 键退出
; MOV    DL, 0FFH
; MOV    AH, 06H
; INT    21H
; CMP    AL, 'Q'
; JZ     ENDD
```

作业:

P286: 5, 9, 14, 18