



第八章 串行输入/输出接口

- 概述
- 串行通信的实现
- 串行通信的基本术语
- 可编程串行通讯接口芯片8251A简介
- 串行通信RS-232C
- USB总线简介





8.1 概述

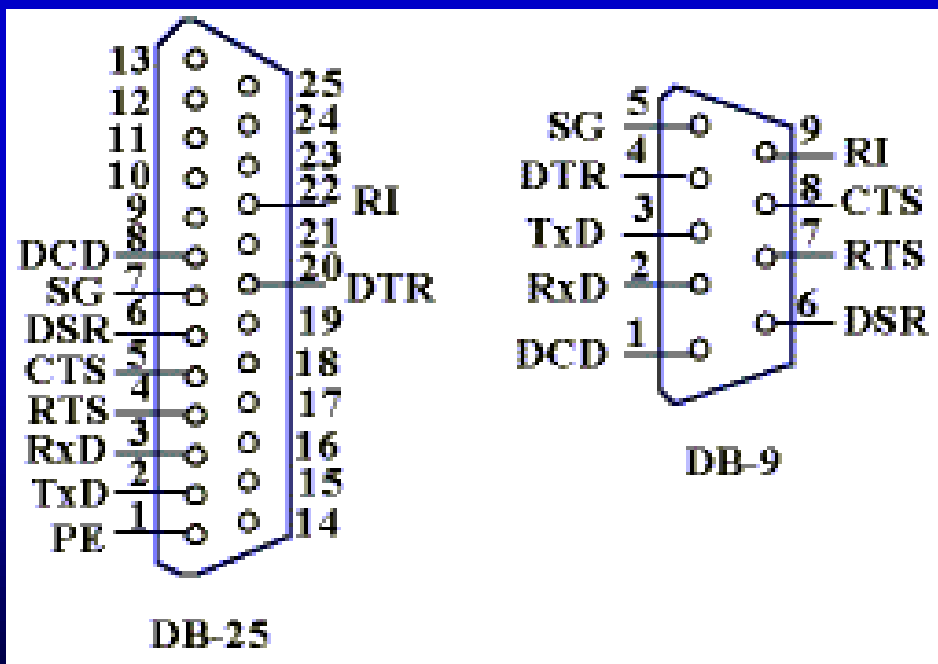
一、串行数据

计算机与外部信息交换方式有两种，一种是并行通信，另一种是串行通信。并行通信时，数据各位同时传送。而串行通信时，数据和控制信息是一位接一位串行地传送下去。这样，虽然速度会慢一些，但传送距离比并行通信长，硬件电路也相应简单些。





现在的台式PC机一般至少有两个RS-232串行口COM1和COM2，通常COM1使用的是9针D形连接器，而COM2使用的是老式的DB25针连接器。



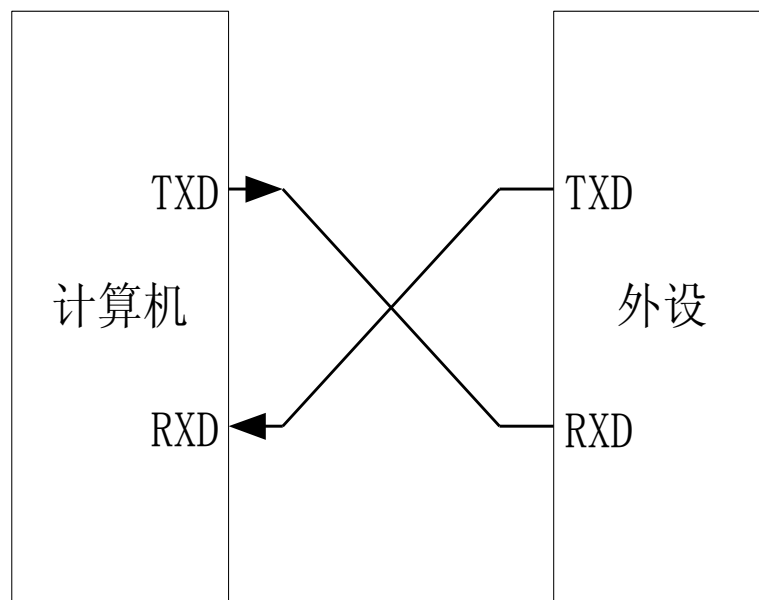


二、 串行通信的实现

1、 并行通信和串行通信的特点：

串行通信是逐位传送数据，所以一个多位数据需要多次传送。比如一个8位的字节型数据至少需要8次传送。它有如下特点：

- ◆ 速度较慢；
- ◆ 引线少；
- ◆ 距离长。

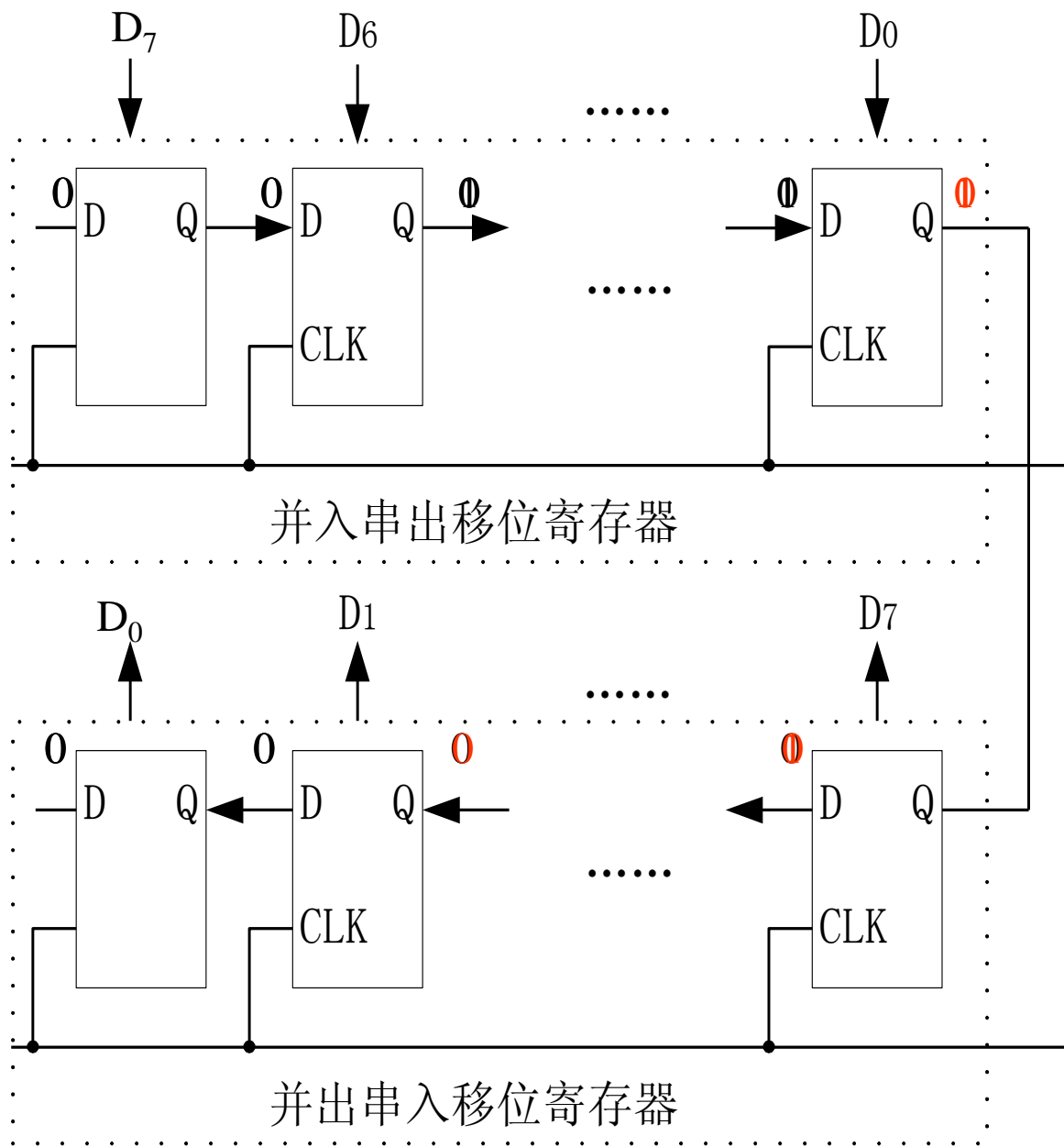




2、串行通信的硬件条件

在硬件上，串行通信系统的核心部件是移位寄存器，其中在发端要有一个并入串出移位寄存器，在收端要有一个串入并出移位寄存器。

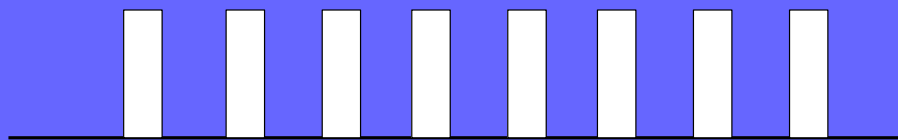
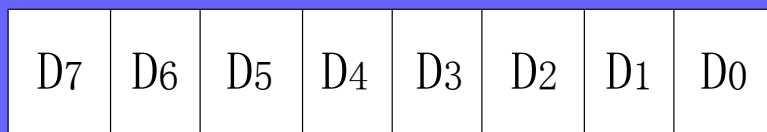






3、串行通信的同步

为了可靠的串行通信，同步信号起到至关重要的作用。在理想情况下，同步信号应该在数据信号线上出现有效数据期间的中心点有效，因为这时数据线上的数据是最稳定的。



同步脉冲



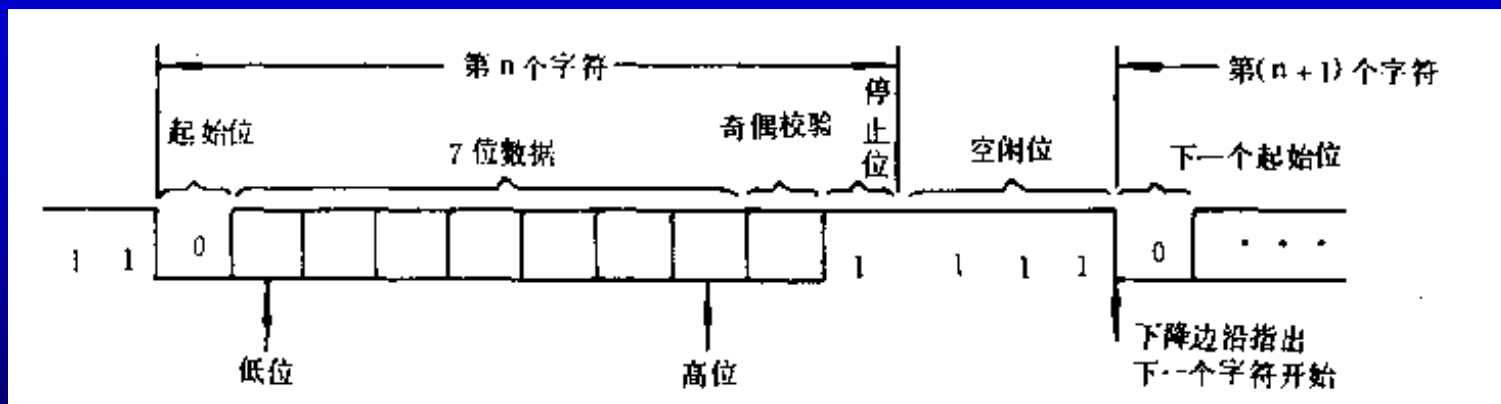


但在典型的串行通信系统中，**是不存在同步信号线的！** 如何实现串行通信的中数据传送的同步呢？系统从软件和硬件两方面采取了如下三个措施：





- (1) 设置波特率波特率是指单位时间内传送的位数，单位是比特/秒。
- (2) 设置数据的传送格式，一般地串行通信在传送数据时，并不是单纯地传送数据位信息。为了使数据传送可靠，还设定了其他一些辅助位。

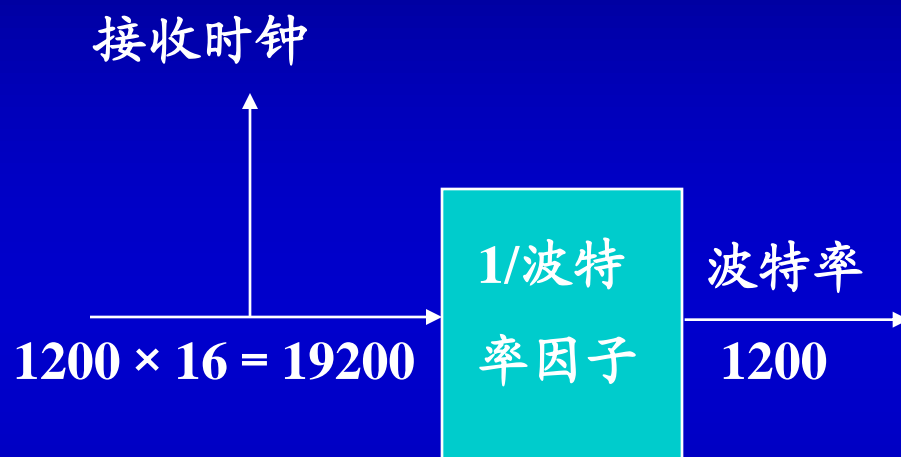


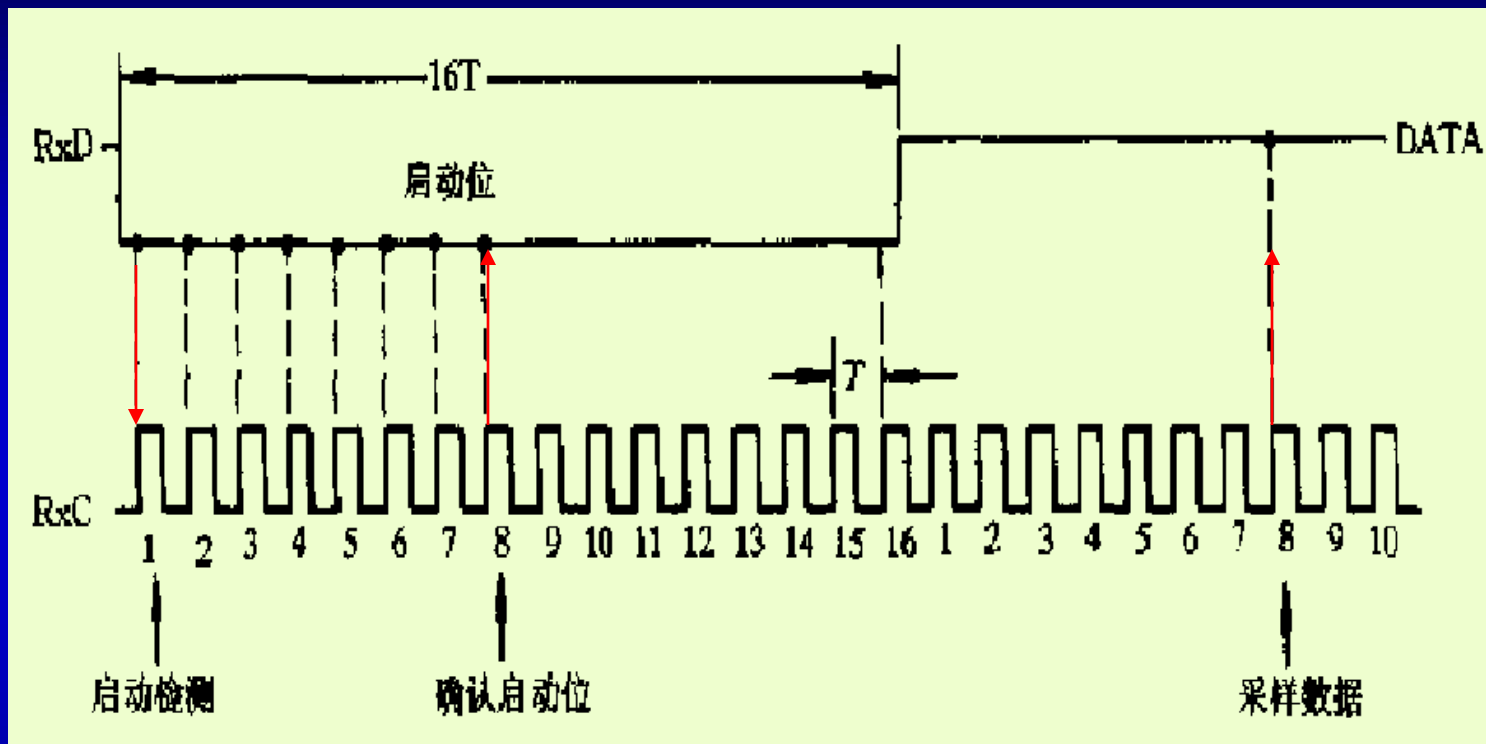


(3) 设置波特率因子

在异步通信方式进行通信时，发送端需要用时钟来决定每1位对应的时间长度，接收端也需要用一个时钟来测定每1位的时间长度，前一个时钟叫发送时钟，后一个时钟叫接收时钟。这两个时钟的频率可以是波特率的数倍，一般取16倍、32倍或64倍。这个倍数就称为波特率因子。









8.2 串行通信的基本技术

一、串行通信方式

1、异步通讯

异步通信是指以字符为单位传送数据，用起始位和停止位标识每个字符的开始和结束字符，两次传送时间间隔不固定。

2、同步通信

异步通讯为了可靠地传送数据，在每次传送数据的同时，附加了一些标志位。在大量数据传送时，为了提高速度，就去掉这些标志，这就是同步通讯。采用同步传送，在数据块开始处要用同步字符来指示，并在发送端和接收端之间要用时钟来实现同步。





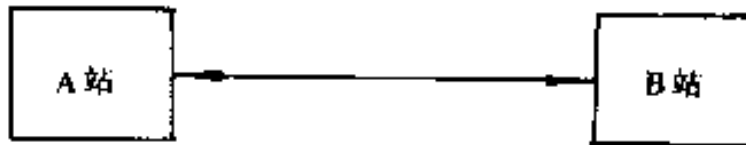
二、串行通讯中的数据传送方向

1、半双工

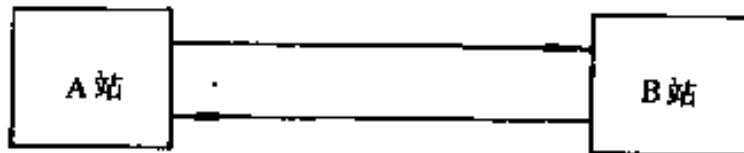
每次只能有一个站发送。

2、完全双工

两个站都能同时发送称完全双工。



(a) 半双工示意图



(b) 完全双工示意图

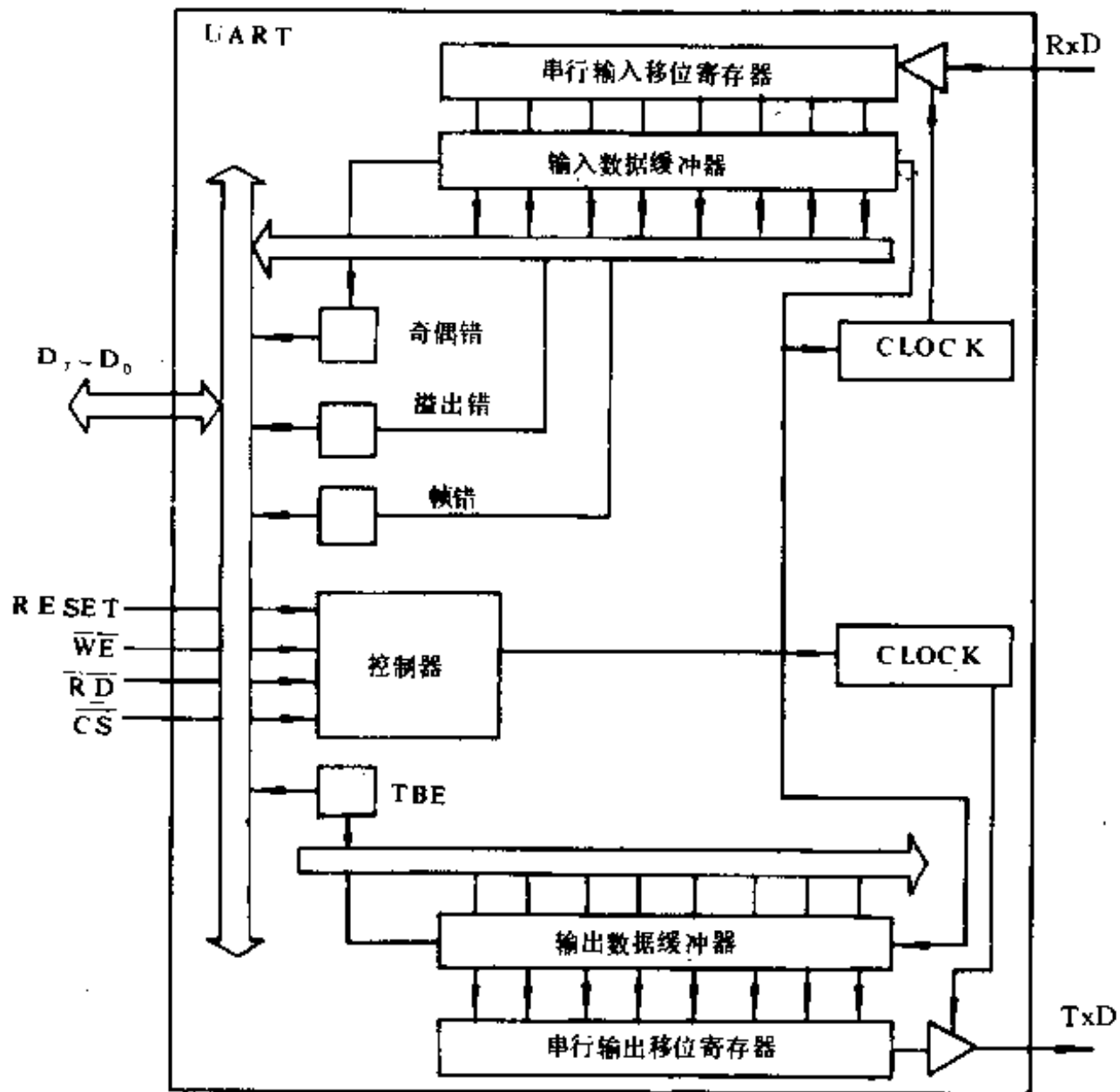




三、异步接收/发送器（UART）

串行接口的基本结构主要是异步接收/发送器（UART），它不仅包括并行数据和串行数据之间的相互转换，还有检测串行通信在传送过程中可能发生错误的逻辑部件。







a. 奇偶错误

如上所述，在接收时，UART检查接收到的每一个字符的“1”的个数这个标志，发出奇偶校验出错信息。

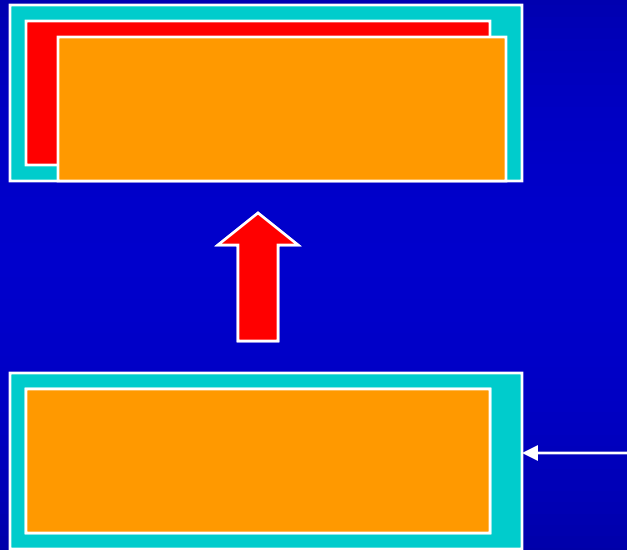
b. 帧错误

若接收到的字符格式不符合规定(如缺少停止位等)，则置位该标志，发出帧出错信息。

c. 丢失(溢出)错误

UART是一种双缓冲器结构。例如，在接收时，接收的数据先由移位寄存器移位，把串行数据变成并行数据，然后送到接收数据寄存器，由输入指令将数据送到CPU中。由于数据送到接收数据寄存器，所以即使CPU还没取走这个数据，UART也可以接收另一个新的字符。但若UART接收到第二个字符的停止位，且要把第二个字符传送到接收数据寄存器时，CPU还没取走上一个数据，于是第一个数据会被丢失。如果UART出现这种情况就置位丢失(溢出)标志，发出丢失出错信息。



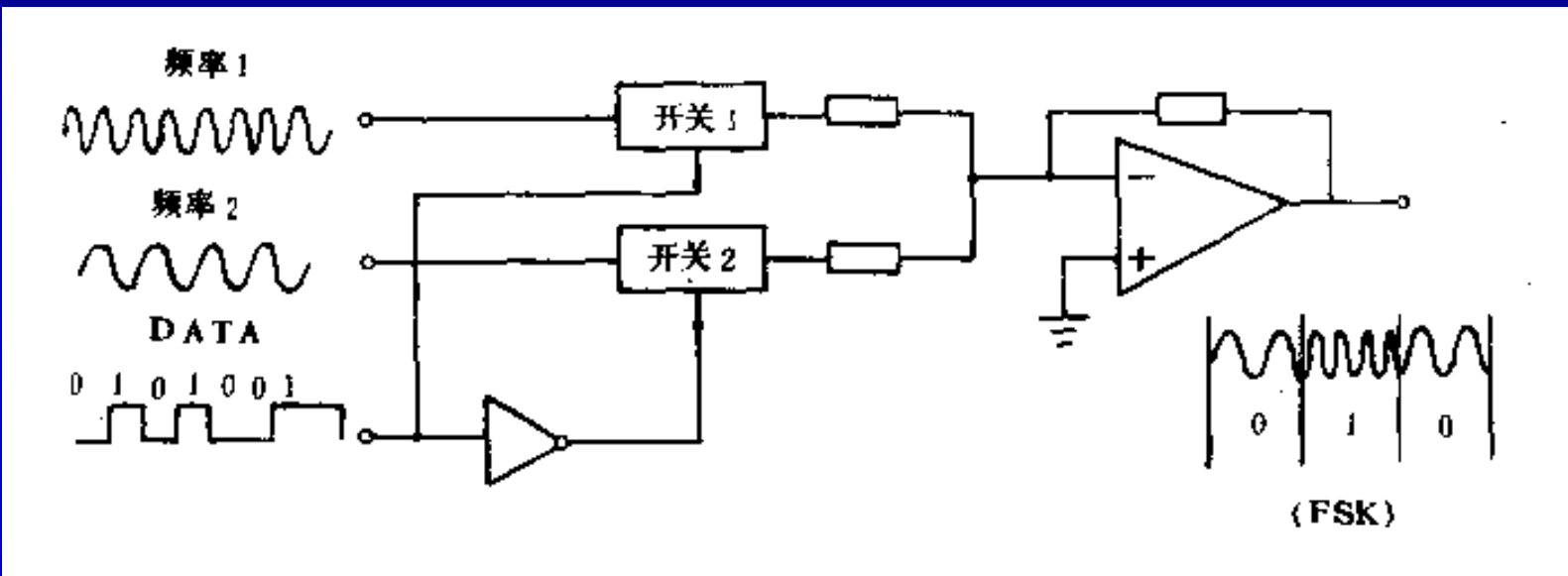




四、信号的调制和解调

由于模拟信号的传输比数字信号传输更为有效，因而可将数字信号调制成模拟信号进行传输，用解调器把接收的模拟信号再转换成数字信号。







8.3 可编程串行通讯接口芯片8251A简介

8251A芯片是INTEL公司生产的大规模集成电路芯片，是与INTEL系列CPU兼容的可编程的串行通讯接口。虽然8251A功能较强，但它需要外部时钟电路。因此采用8251A作为接口电路时需要比较复杂的外围电路。而目前流行的单片机如MCS51系列，CPU内部就集成了串行接口部件及定时器/计数器，几乎不需要外围辅助电路，使用起来非常简单，性能价格比很高，因此现在越来越多的数字化仪器仪表电路中不再采用8251A而是使用单片机作为串行通信接口了。





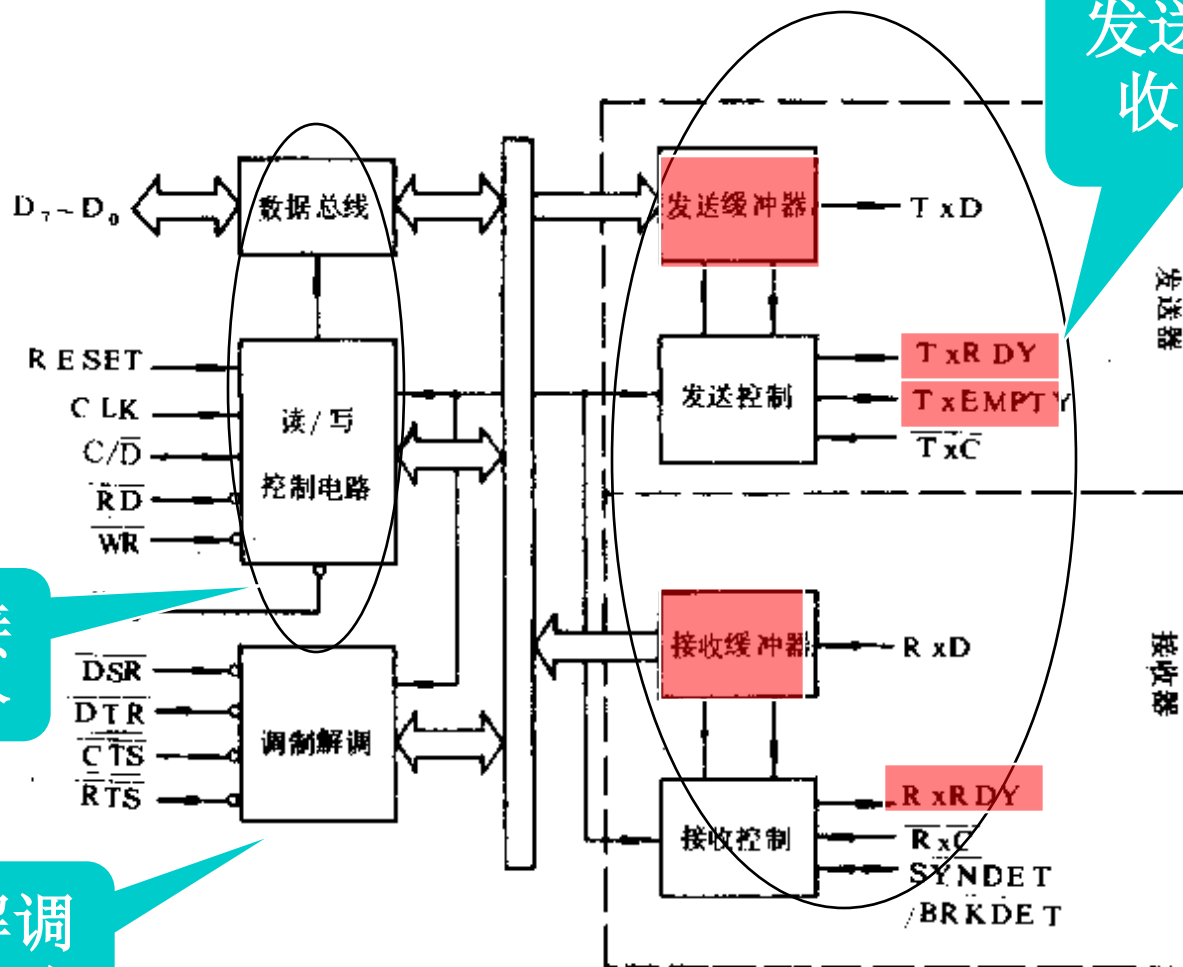
一、主要基本性能

1. 可用于同步和异步传送
2. 波特率, DC—19.2K(异步); DC—64K(同步)。
3. 完全双工, 双缓冲发送和接收
4. 与8080/8085CPU完全兼容





二、内部结构



发送和接收电路

总线接口部分

调制解调控制电路





1. 发送器

- a. 数据输出线 TxD
- b. 发送时钟 TxC
- c. 发送器准备好 TxRDY
- d. 发送器空信号 TxEMPTY

过程

TxRDY有效 → CPU写数据到8251 → 8251发数据
→ 发送完毕, TxEMPTY有效

2. 接收器

- a. 数据输入线 RxD
- b. 接收时钟 RxC
- c. 接收器准备好 RxRDY

过程

8251接数据 → RxRDY有效 → CPU读8251

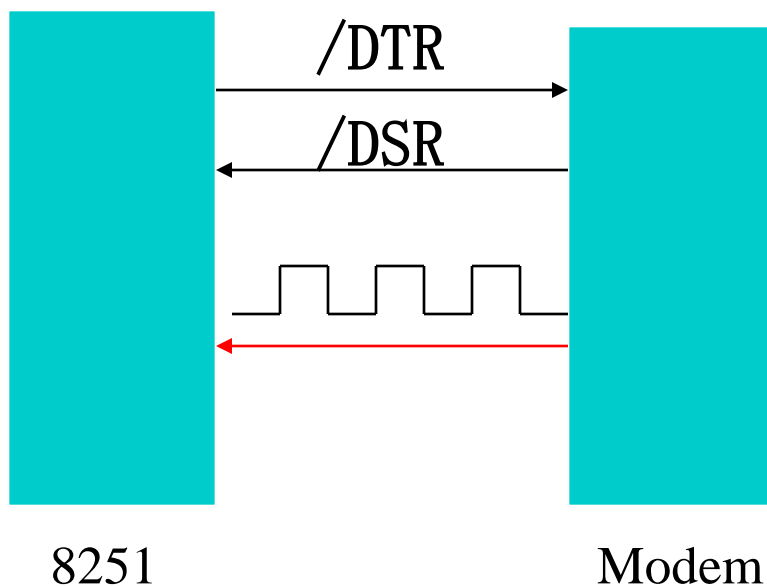




3. 调制解调控制电路

/DTR——数据终端准备好信号。是由8251A送往外设的，表示CPU当前已经准备就绪。

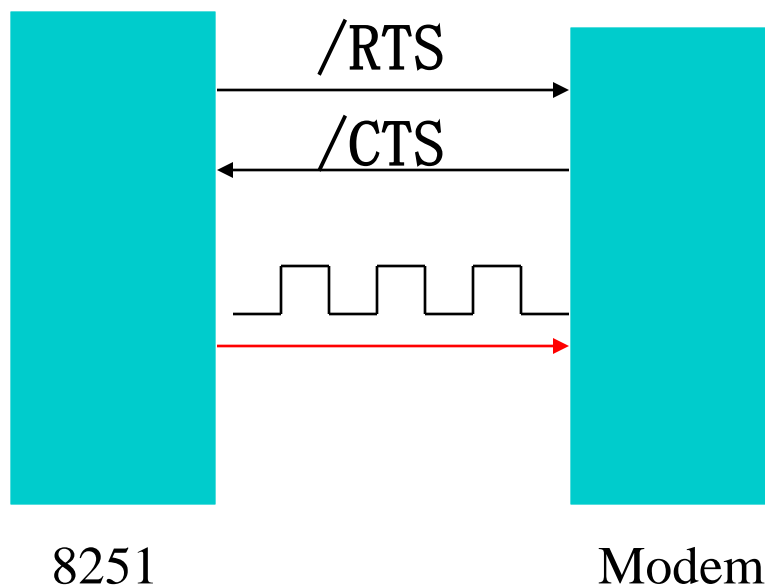
/DSR——数据设备准备好。是外设送往8251A的，表示当前外设已经准备好数据。

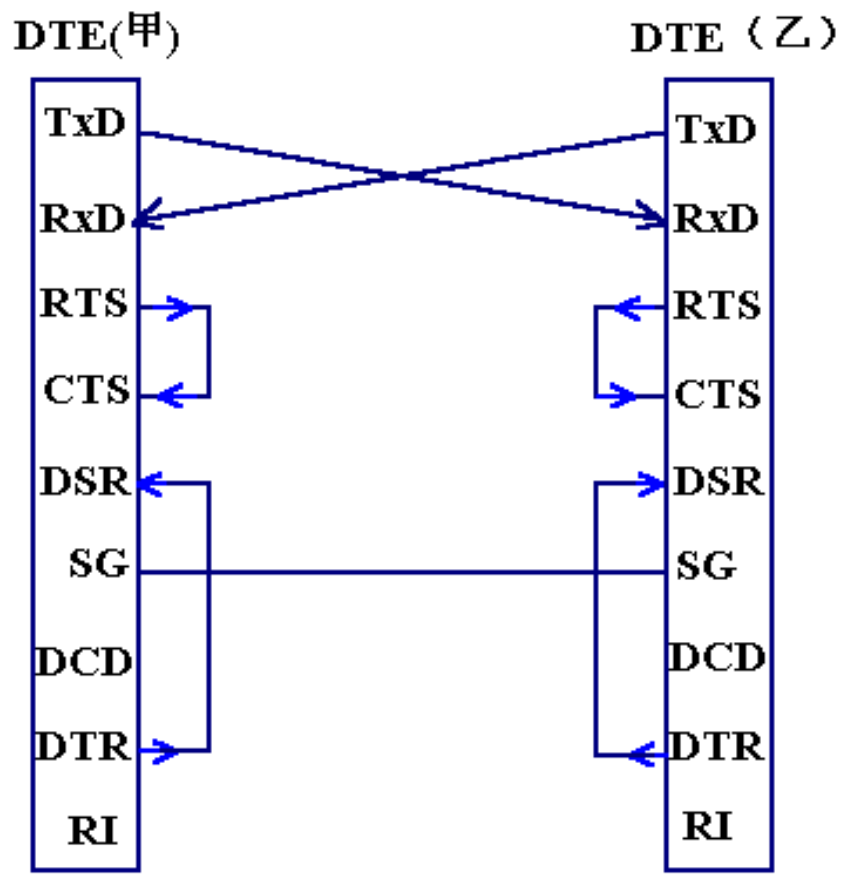




/RTS——请求发送信号。是8251A送往外设的，表示CPU已经准备好发送。

/CTS——清除请求发送信号。由外设送往8251A的，当为低电平时，允许8251A执行发送操作。



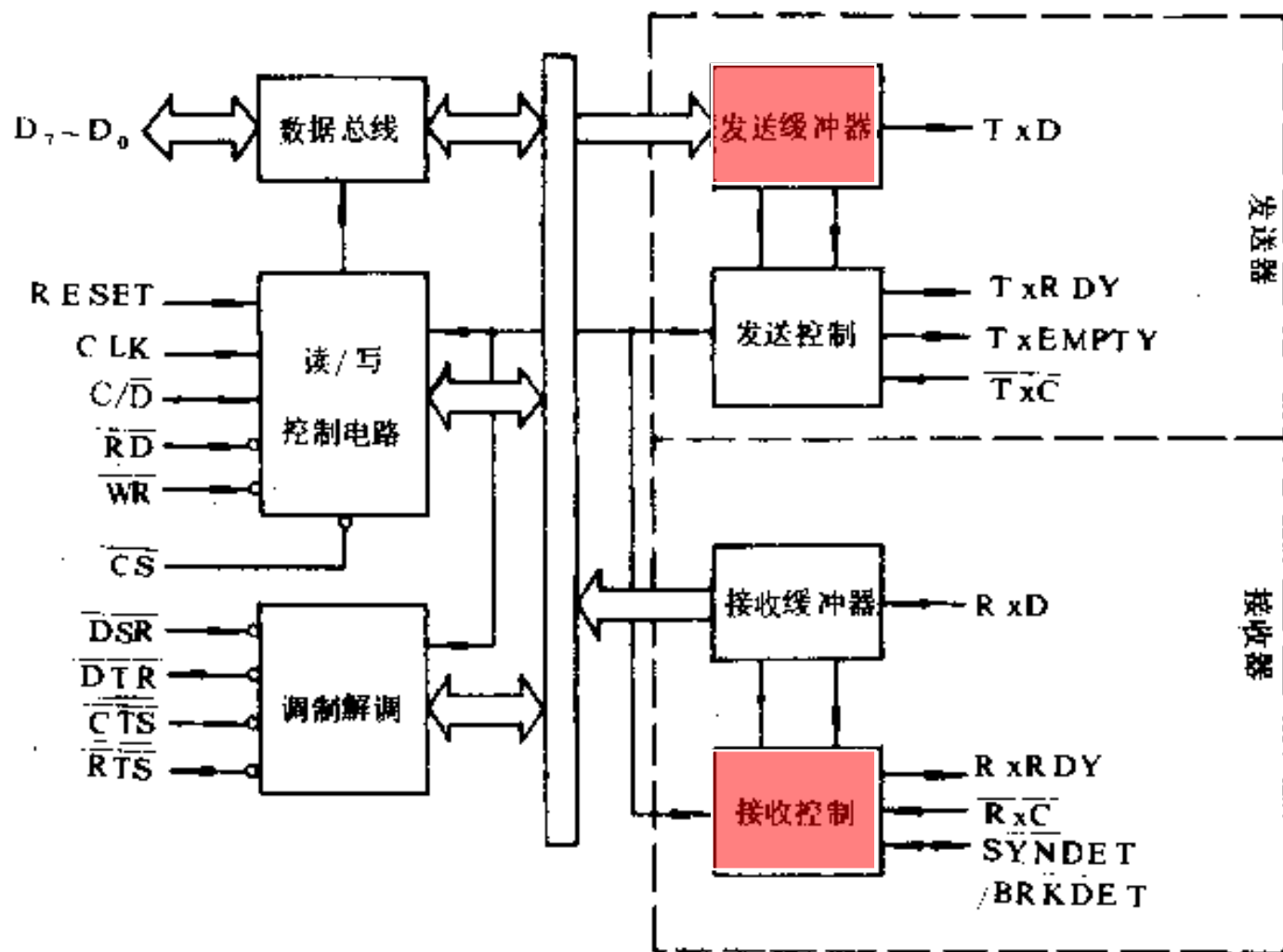




4. 读/写控制电路

\overline{CS}	C/\overline{D}	\overline{RD}	\overline{WR}	功 能
0	0	0	1	CPU 从 8251A 读数据
0	0	1	0	CPU 向 8251A 写数据
0	1	0	1	CPU 从 8251A 读状态
0	1	1	0	CPU 向 8251A 写命令
1	×	×	×	无操作

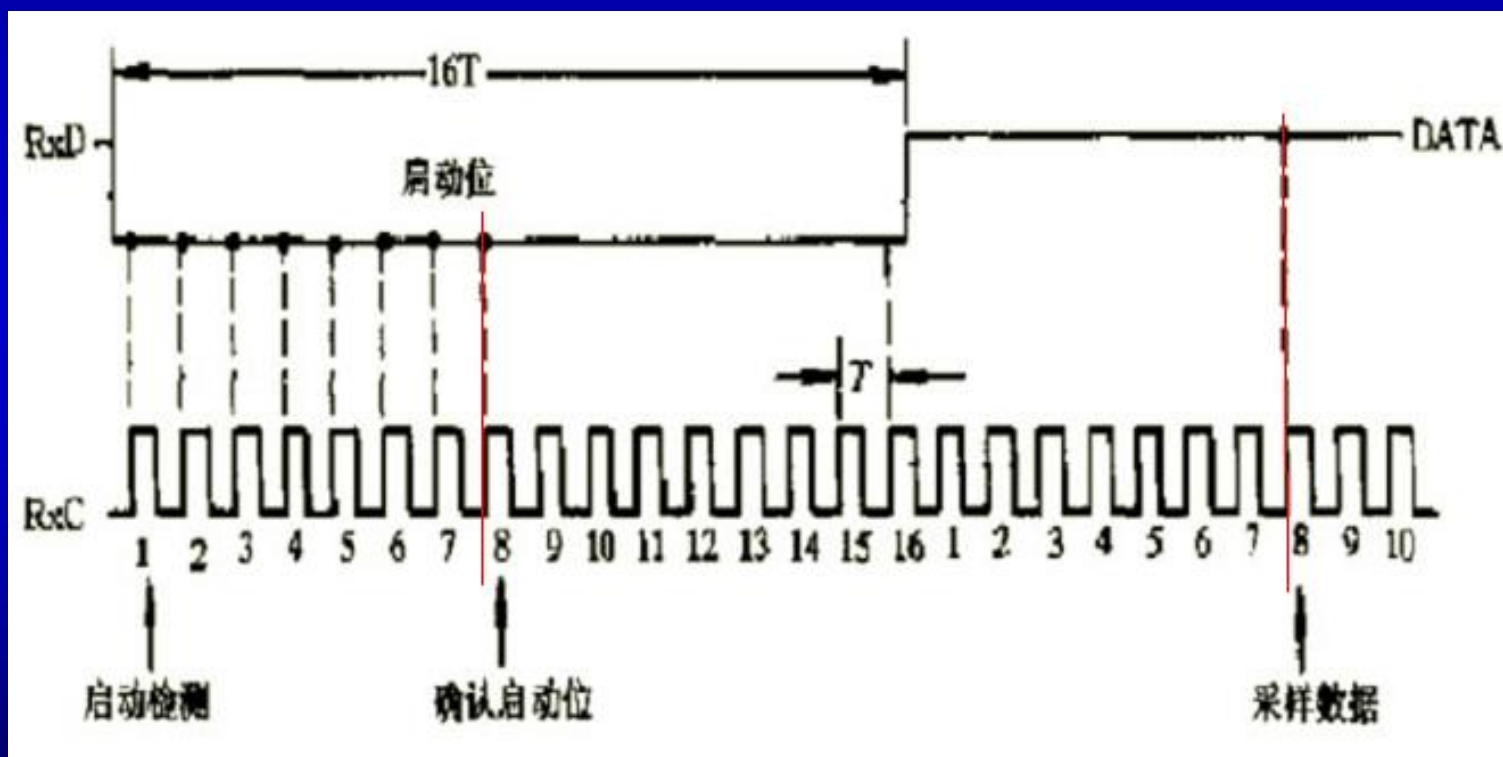






三、8251编程

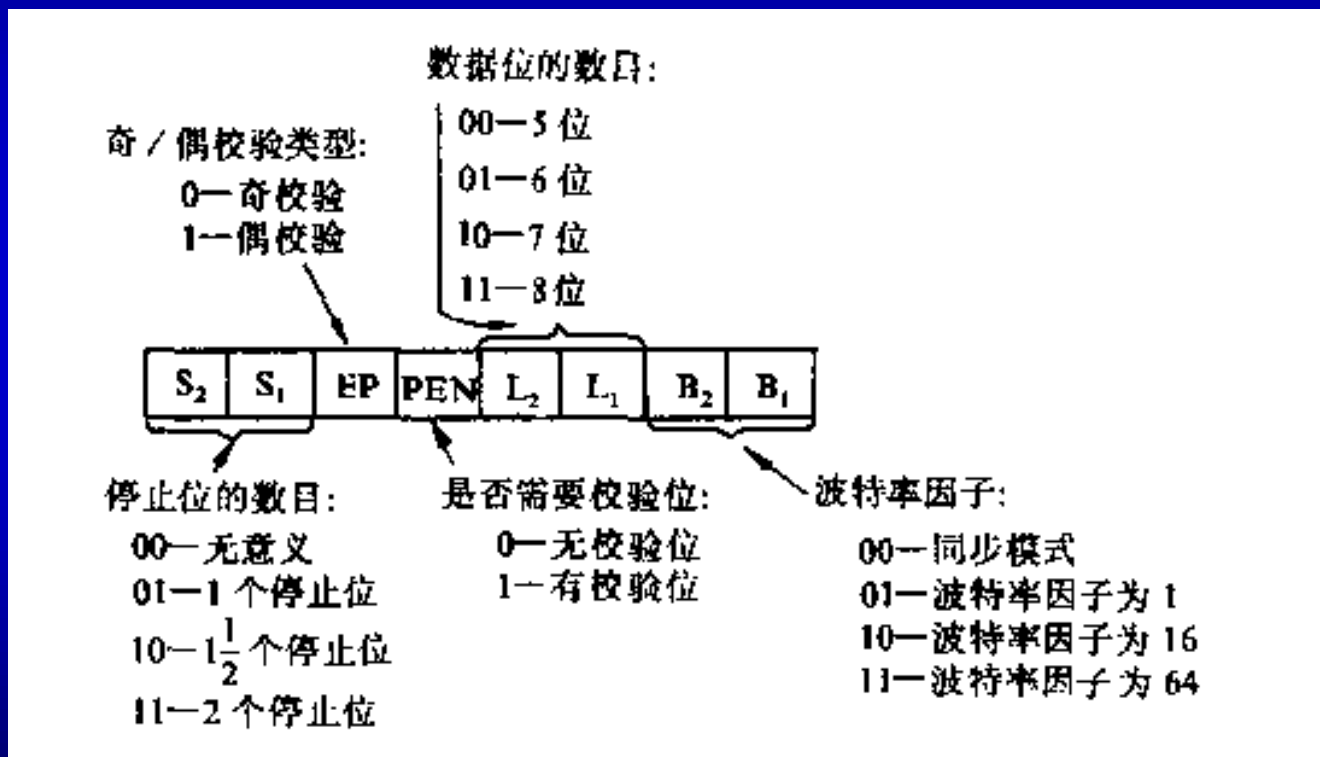
■ 1、方式选择控制字





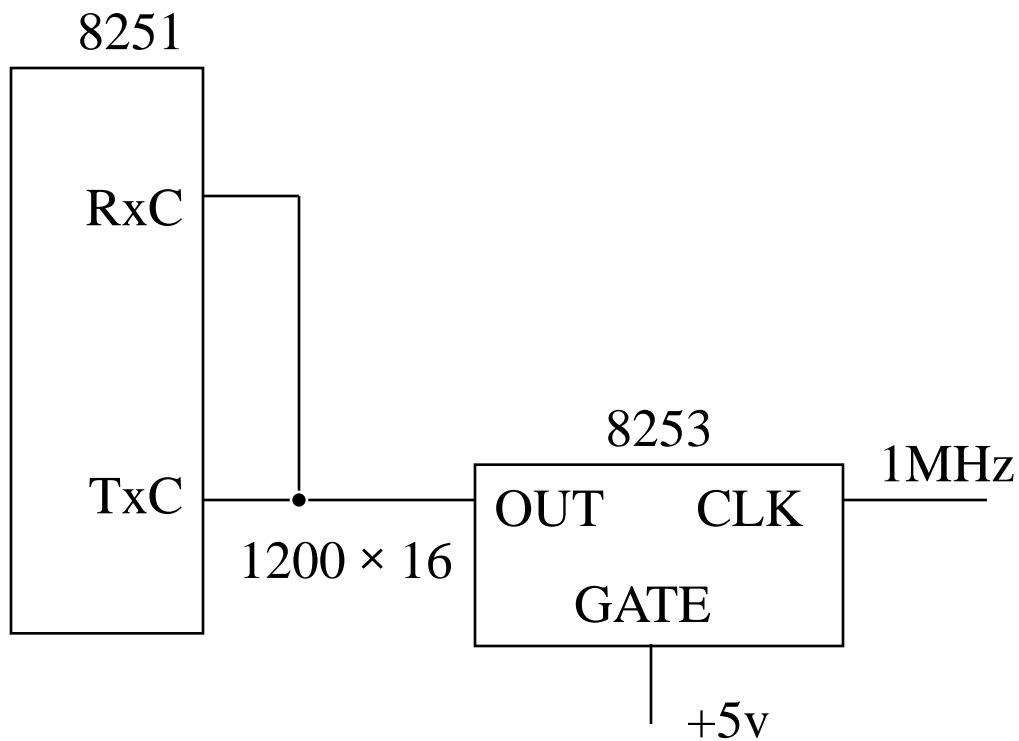
三、8251编程

■ 1、模式寄存器的格式（方式选择控制字）



$$\text{时钟频率} = \text{波特率因子} \times \text{波特率}$$



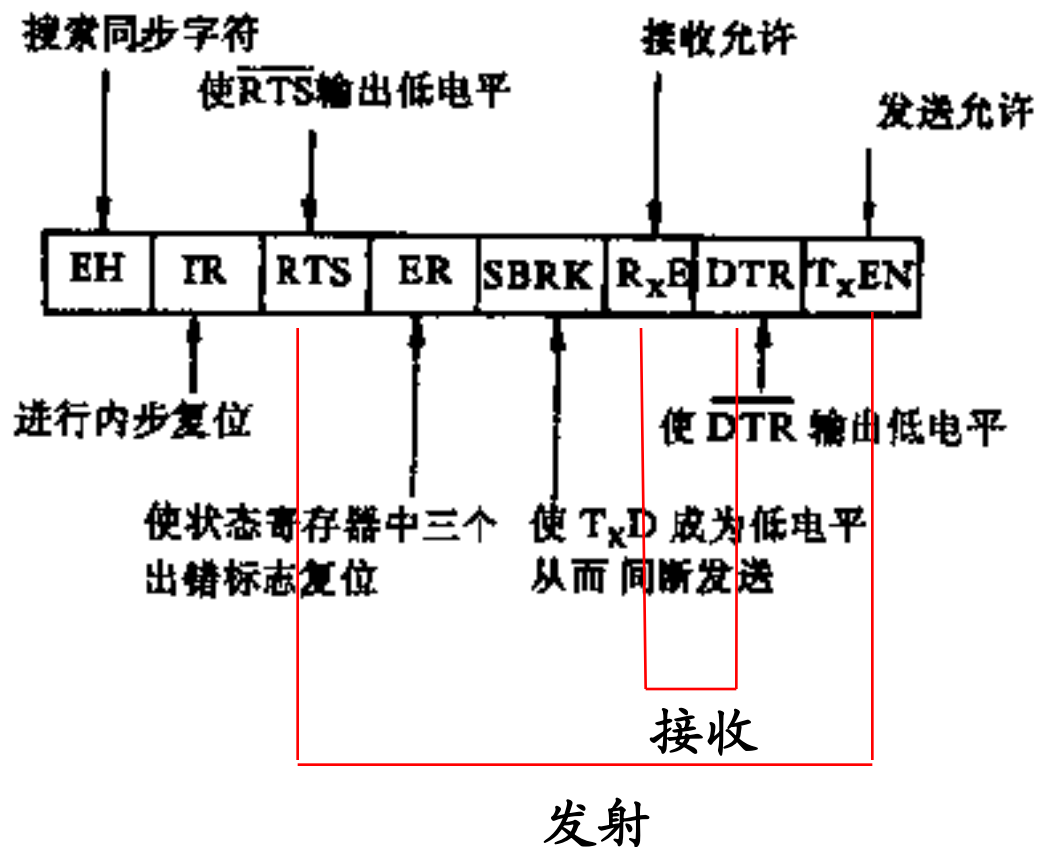


如果波特率为1200，波特率因子为16，问计数初值？



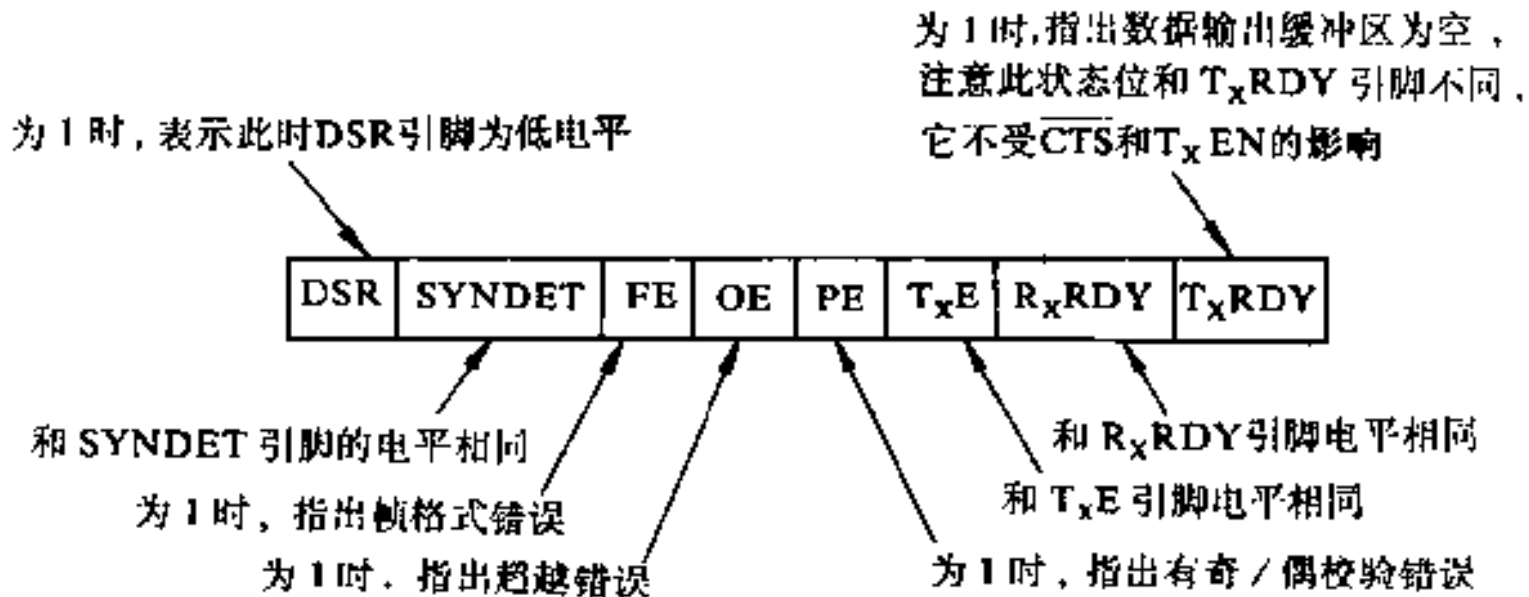


2. 控制寄存器的格式（操作命令控制字）





3. 状态寄存器的格式（状态控制字（读））

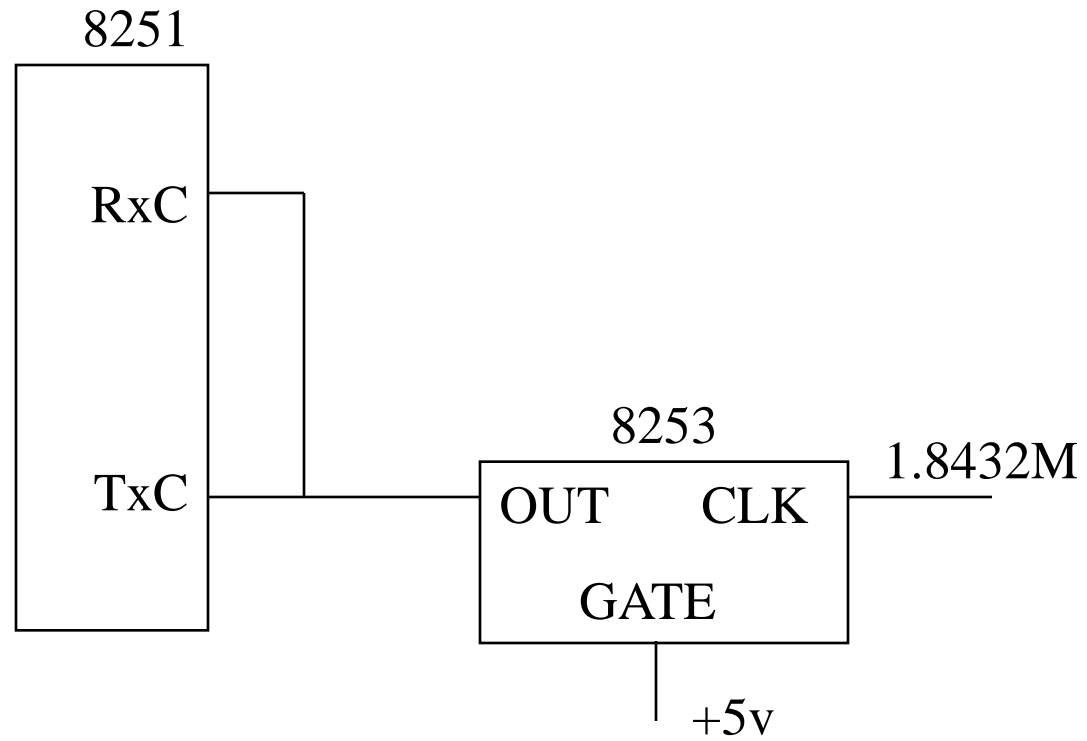




四、举例

- 例：编写8251异步模式下的接收和发送程序，设端口地址：90H，92H，波特率因子16，1起始位，1停止位，无奇偶校验，每字符8位。







解：

1、8253工作方式及计数初值

工作方式 mode3

8253输出频率：

$$f_{out} = 9600 * 16 = 153600 = 153.6K$$

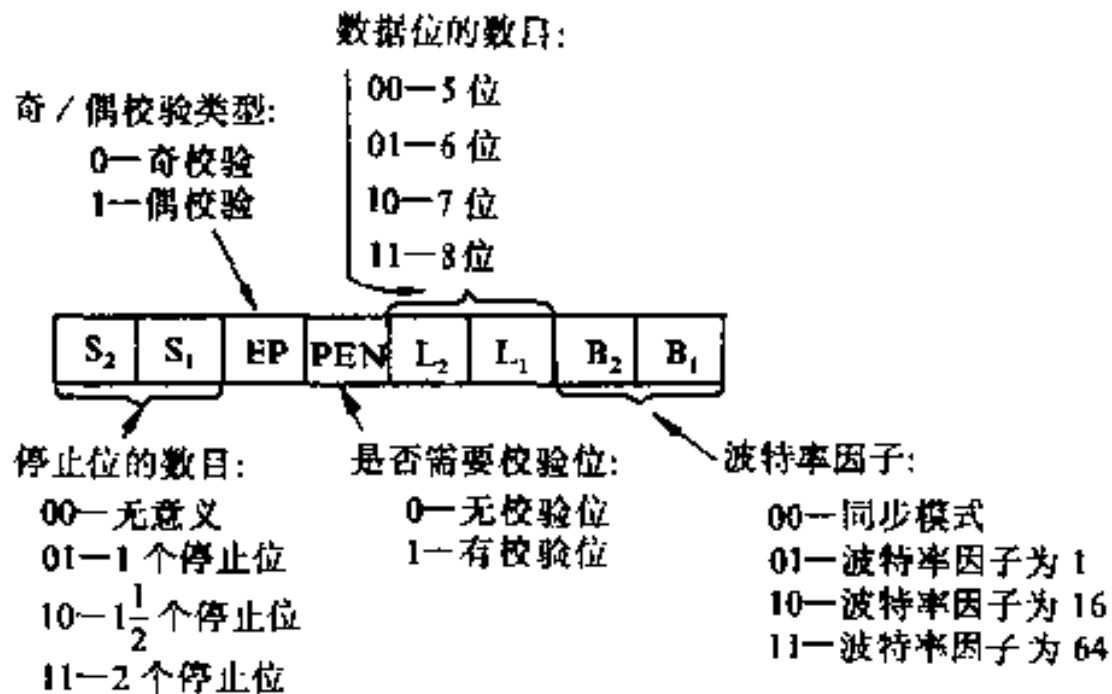
$$f_{clk} = 1M \quad N = 1000 / 153.6 = \sim 6.51 \text{ (不能用!)}$$

$$f_{clk} = 1.8432M \quad N = 12$$





2、8251工作方式 方式选择控制字

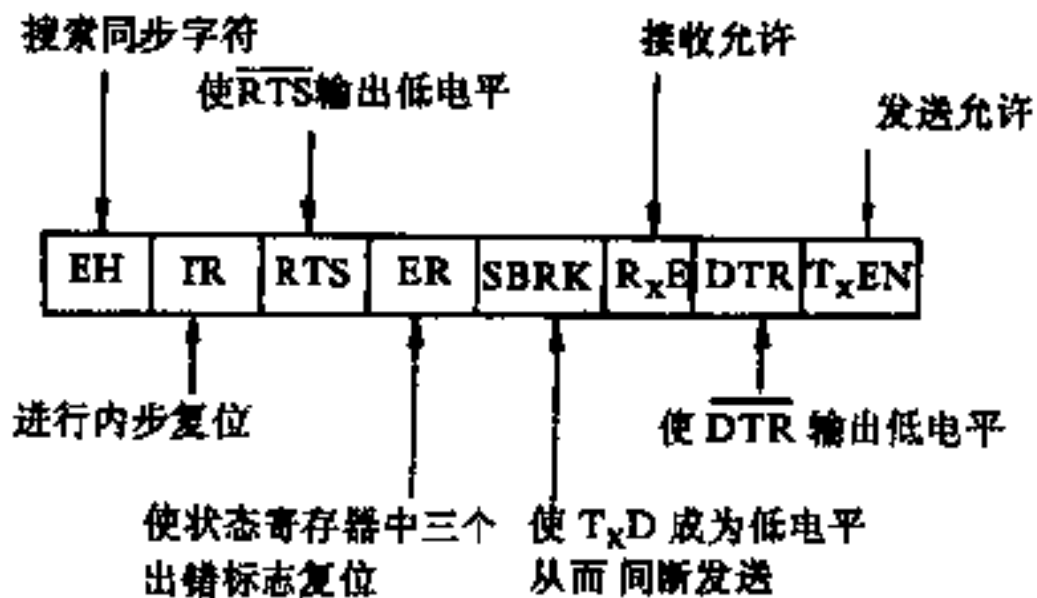


$$\begin{array}{|c|c|} \hline 0 & 1 \\ \hline \end{array} \begin{array}{|c|c|} \hline 0 & 0 \\ \hline \end{array} \begin{array}{|c|c|} \hline 1 & 1 \\ \hline \end{array} \begin{array}{|c|c|} \hline 1 & 0 \\ \hline \end{array} = 4EH$$





操作命令控制字



0 0 1 1 0 1 1 1 = 37H



3、程序

8251初始化设置:

- a、先对8251A软件复位，一般采用先送3个0，再送1个40H的方法，这也是8251A的编程约定，40H可以看成是使8251A执行复位操作的实际代码。
- b、对8251A进行工作方式及操作命令设置



.....

```
MOV AL, 00H ; 复位8251
OUT 92H, AL
CALL DELAY
OUT 92H, AL
CALL DELAY
OUT 92H, AL
CALL DELAY
MOV AL, 40H
OUT 92H, AL
CALL DELAY ; 等待8251复位
```

```
MOV AL, 4EH ; 置工作方式
OUT 92H, AL
MOV AL, 37H ; 操作命令
OUT 92H, AL
```

.....



DSR	SYNDET	FE	OE	PE	T _x E	R _x RDY	T _x RDY
-----	--------	----	----	----	------------------	--------------------	--------------------

发送:

```

SENDDAT      PROC                ; DL:要发送的数据
CHKTXR:      IN AL, 92H           ; 输入状态字
              AND AL, 01H        ; 查TXRDY
              JZ CHKTXR
              MOV AL, DL
              OUT 90H, AL        ; 发送
              RET
SENDDAT      ENDP

```

接收:

```

RECIDAT      PROC                ; AL:接收的数据
CHKRXD:      IN AL, 92H           ; 输入状态字
              AND AL 02H         ; 查RXRDY?
              JZ CHKRXD
              IN 90H, AL         ; 发送
              RET
RECIDAT      ENDP

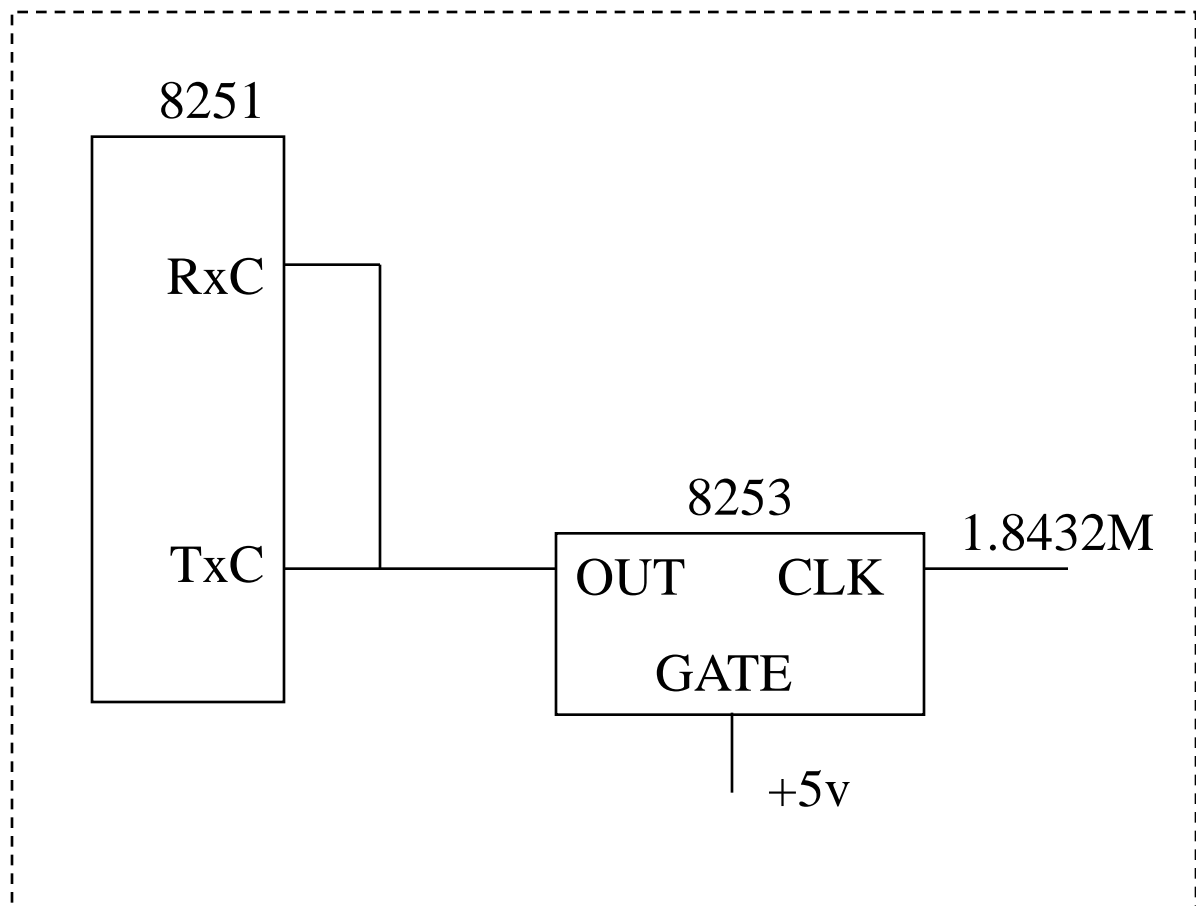
```

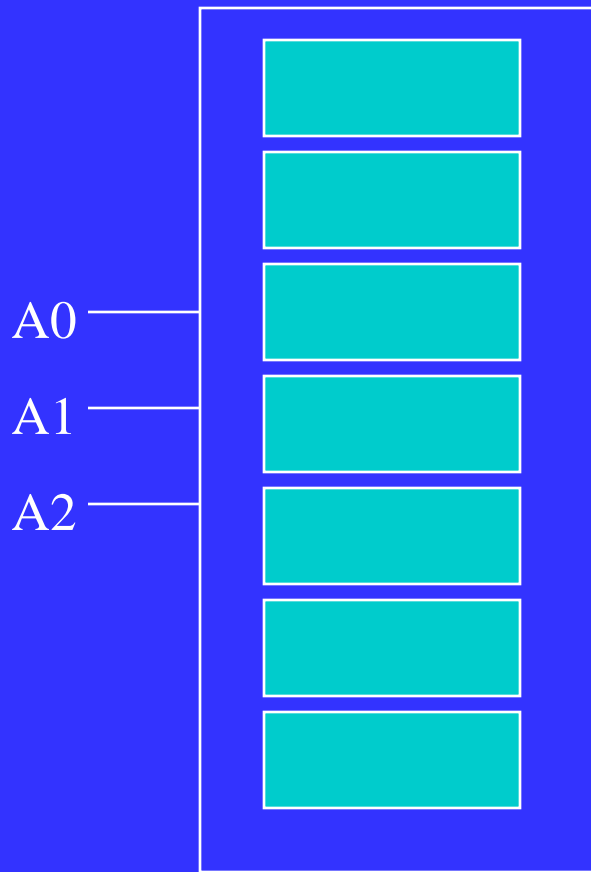




五、8250和16550

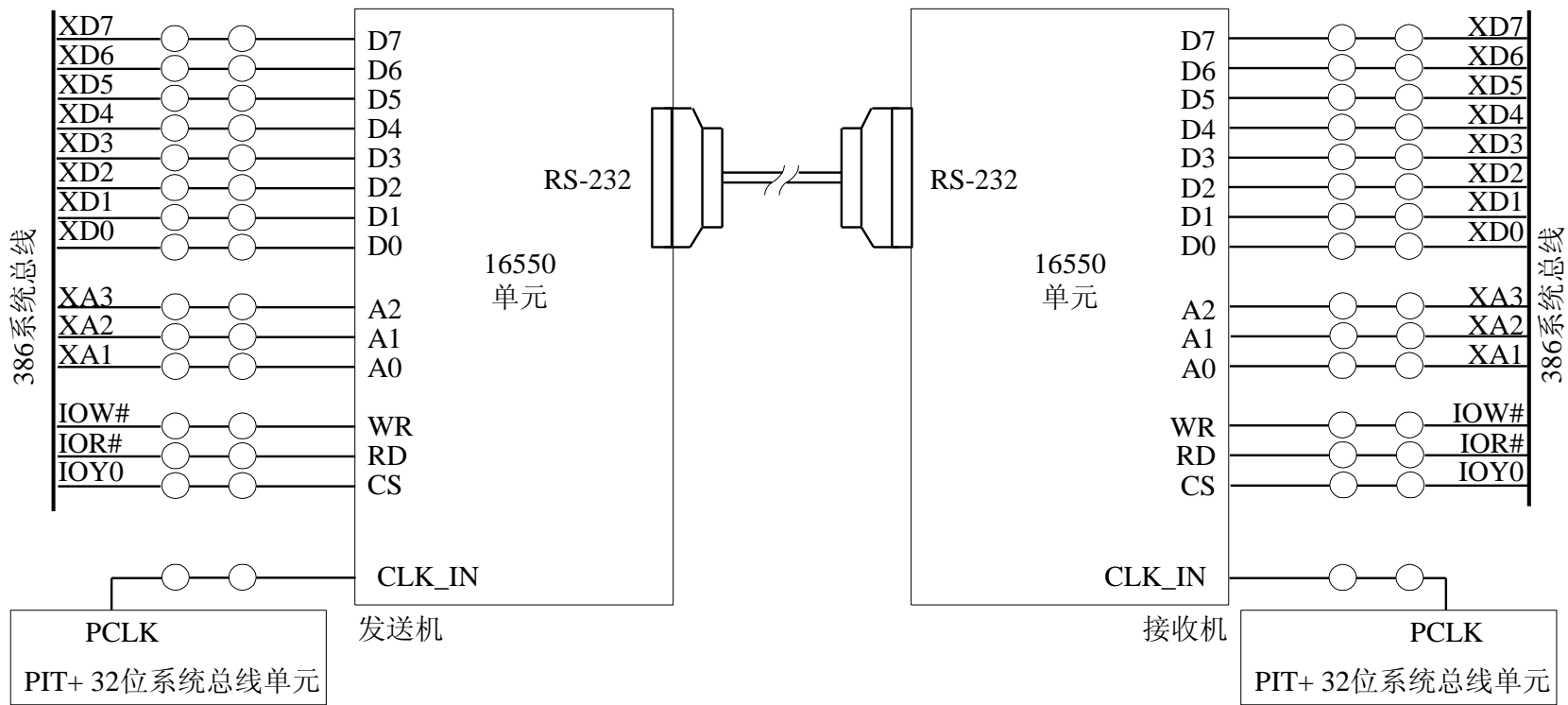
速度 8250: 50~9600 16550: 115200





000	数据口/波特率因子低字节
001	波特率因子高字节
011	线路控制寄存器
101	线路状态寄存器





;允许访问除数寄存器

MOV DX,MY16550_3

MOV AL,80H

OUT DX,AL

CALL DALLY

;波特率9600 ; 波特率 = 1.8432/(波特率因子 × 16)

MOV DX,MY16550_0

;波特率低8位

MOV AL,0CH

OUT DX,AL

CALL DALLY

MOV DX,MY16550_1

;波特率高8位

MOV AL,00H

OUT DX,AL

CALL DALLY





; 工作方式选择

MOV DX,MY16550_3

MOV AL,1BH

;偶校验,1位停止位,8位数据

OUT DX,AL

CALL DALLY

; modem设置

MOV DX,MY16550_4

;MCR

MOV AL,03H

;RTS, DTR有效

OUT DX,AL

CALL DALLY

; 中断设置

MOV DX,MY16550_1

; 中断允许寄存器

MOV AL,00H

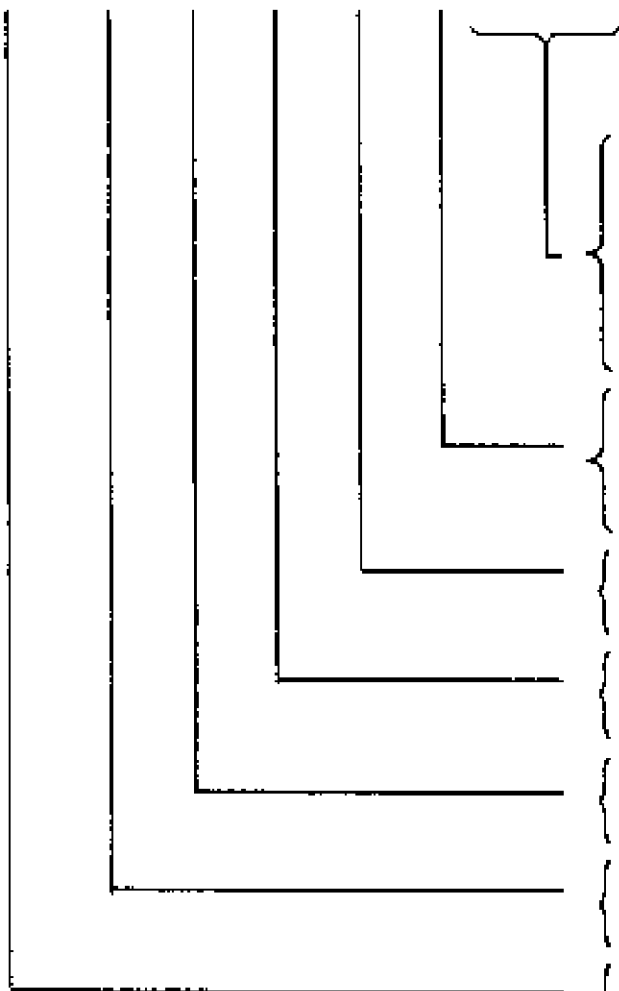
;中断屏蔽

OUT DX,AL

CALL DALLY

0 0 0 1 1 0 1 1

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



字长

- = 00 5 位
- = 01 6 位
- = 10 7 位
- = 11 8 位

停止位

- = 0 1 位
- = 1 1.5 位(数据位 5 位)
- 2 位(数据 6、7、8 位)

- = 0 无奇偶校验
- = 1 允许奇偶校验
- = 1 偶校验(且 D₃ = 1)
- = 0 奇校验(且 D₃ = 1)

- = 0 无附加奇偶校验位
- = 1 有附加奇偶校验位

- = 0 正常
- = 1 发空缺位

- = 0 允许访问接收、发送数据寄存器或中断允许寄存器。
- = 1 允许访问波特率因子寄存器



;发送

TR: MOV DX,MY16550_5 ;发送保持寄存器检查LSR
IN AL,DX
AND AL,20H
JZ TR
MOV DX,MY16550_0
MOV AL,TXD_data ;发送数据
OUT DX,AL
RET

;接收

REV: MOV DX,MY16550_5 ;查询接收
IN AL,DX
AND AL,01H
JZ REV
MOV DX,MY16550_0 ;接收数据寄存器
IN AL,DX
RET





8.4 串行通信接口RS-232C

一、RS-232C电器特性及接口信号

1. 电气特性

在数据线TXD和RXD上:

逻辑1 = $-3V \sim -15V$

逻辑0 = $+3V \sim +15V$

在控制线和状态线RTS、CTS、DSR、DTR和DCD上:

信号有效 = $+3V \sim +15V$

信号无效 = $-3V \sim -15V$





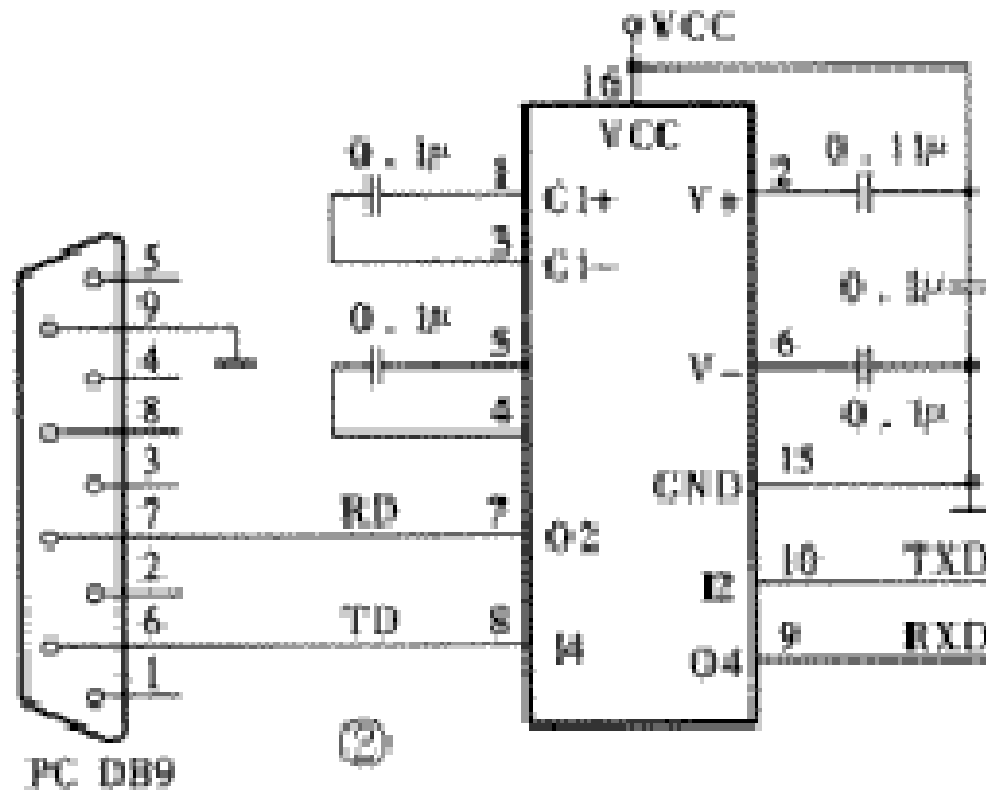
2. RS-232C与TTL转换

RS-232C是用正负电压来表示逻辑状态，与TTL以高低电平表示逻辑状态的规定不同。因此，为了能够使计算机接口与终端的TTL器件连接，必须在RS-232C与TTL电路之间进行电平和逻辑关系变换。





MAX232





3. RS-232C的接口信号

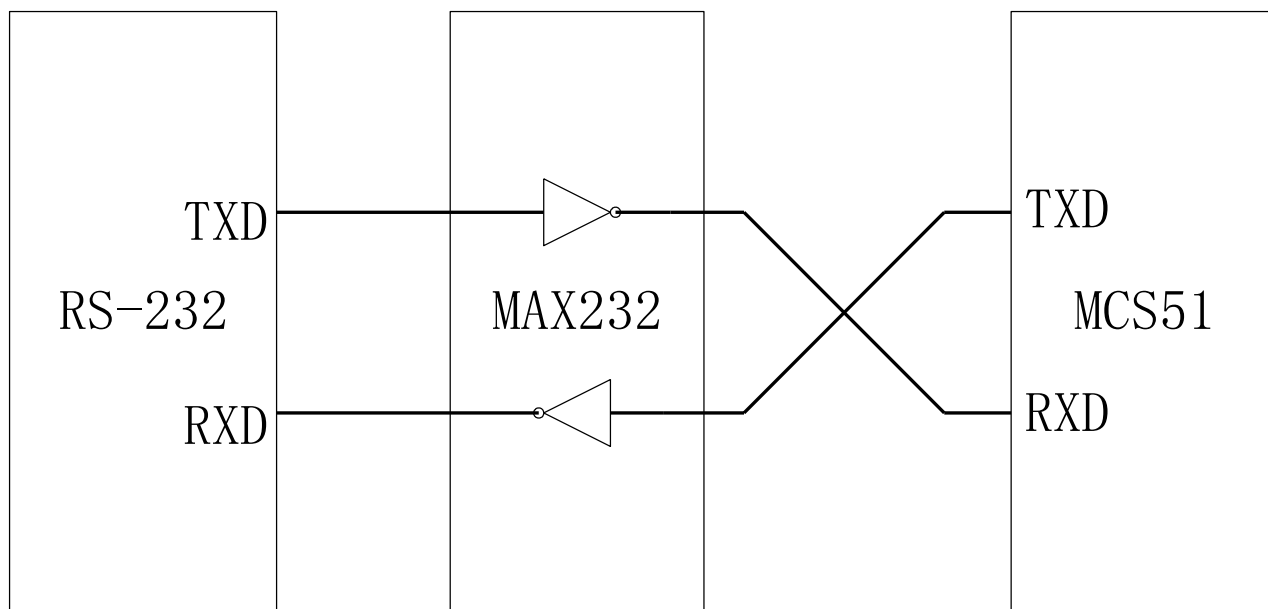
可以用电缆线直接连接标准RS232端口，但通信距离较近($<12\text{m}$)。若 距离较远，可附加调制解调器(MODEM)。RS—232C规标准接口有25条线，最常用的信号线有9根。





二、RS-232C应用举例

1. RS-232C串口通信接线方法（三线制）





2. BIOS串行通信口功能

IBM PC及其兼容机提供了比较灵活的关于串行口的BIOS中断调用方法，即通过INT 14H调用ROM BIOS串行通信口例行程序。该例行程序包括将串行口初始化为指定的字节结构和传输速率，检查控制器的状态，读写字符等功能





3. 举例

设通信双方有一台计算机为PC机，使用COM1端口。在串行通信中，必须首先设定通信双方所使用字符串的数据结构，才能进行软件编程。这里我们设数据在接收和发送的字符串中，序号为0的字节为数据长度，其后的字节为所接收的数据。





接收数据子程序:

```
receive  proc far
```

```
rec_check:
```

```
    mov ah,3
```

```
    mov dx,0
```

```
    int 14h
```

```
    test ah,20H
```

```
    jz rec_check
```

```
    mov ah,2
```

```
    mov dx,0
```

```
    int 14h
```

```
    ret
```

```
receive  endp
```

; 这是接收数据子程序,

; 出口AL, AH

; 读通讯口状态字

; 测试数据准备好位

; 数据未准备好, 再读状态字

; 读通信端口数据





发送数据子程序:

```
send          proc far          ;这是发送数据子程序,
                                     ;输入参数:AL,输出参数:AL,AH

                                     push ax
send_check:   mov ah,3          ; 读通讯口状态字
                                     mov dx,0
                                     int 14h
                                     test ah,20h          ; 测试“发送保存寄存器空”位
                                     jz send_check          ; 发送保存寄存器满,再读状态字
                                     pop ax
                                     mov ah,2          ; 发送数据
                                     mov dx,0
                                     int 14h
                                     ret
send          endp
```



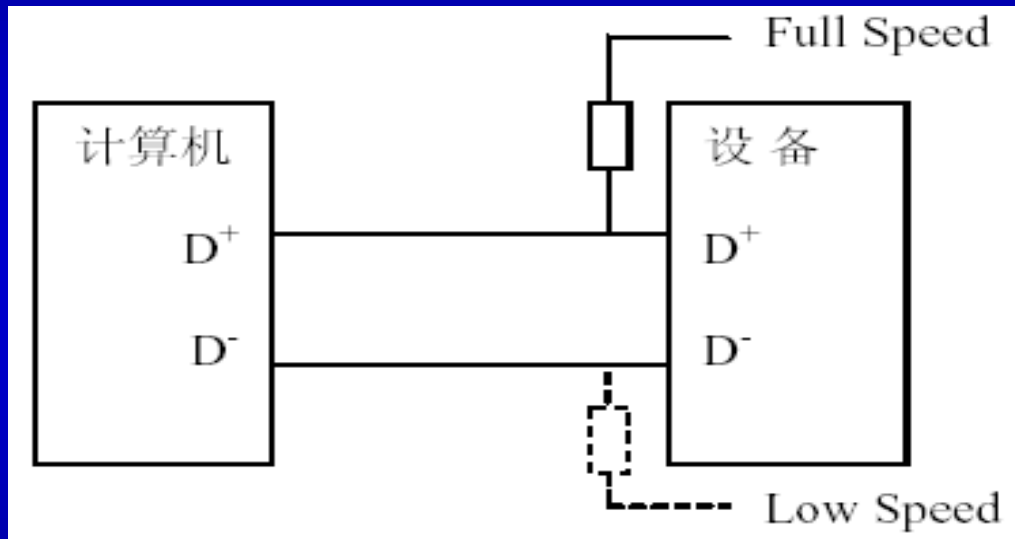


8. 5 USB总线简介

一、USB的定义

USB是Universal Serial Bus的缩写，中文意思就是“通用串行总线”，是连接有USB接口的计算机外围设备到计算机的一种计算机外部总线结构







二、 USB的开发背景

制定了USB的规范:

1. 连接PC 与电话。
2. Plug-and-Play （即插即用）。
3. 端口扩展。USB提供双向低成本低速到中速(USB 2.0 可达480Mb/s)的通用外设总线适用于连接各种各样的外设并且易于扩展。





三、 USB的变革

至目前为止，USB共经历以下几次重大变革：

0.7版本 于1994年11月11日发布。是USB的最早版本。

1.0版本 于1995年11月13日制定发布了，规定USB具有2种传输速度 Low-speed 1.5Mb/s, Full-speed 12Mb/s。

1.1版本 于1998年9月23日制定了1.1规格版本，速度不变。

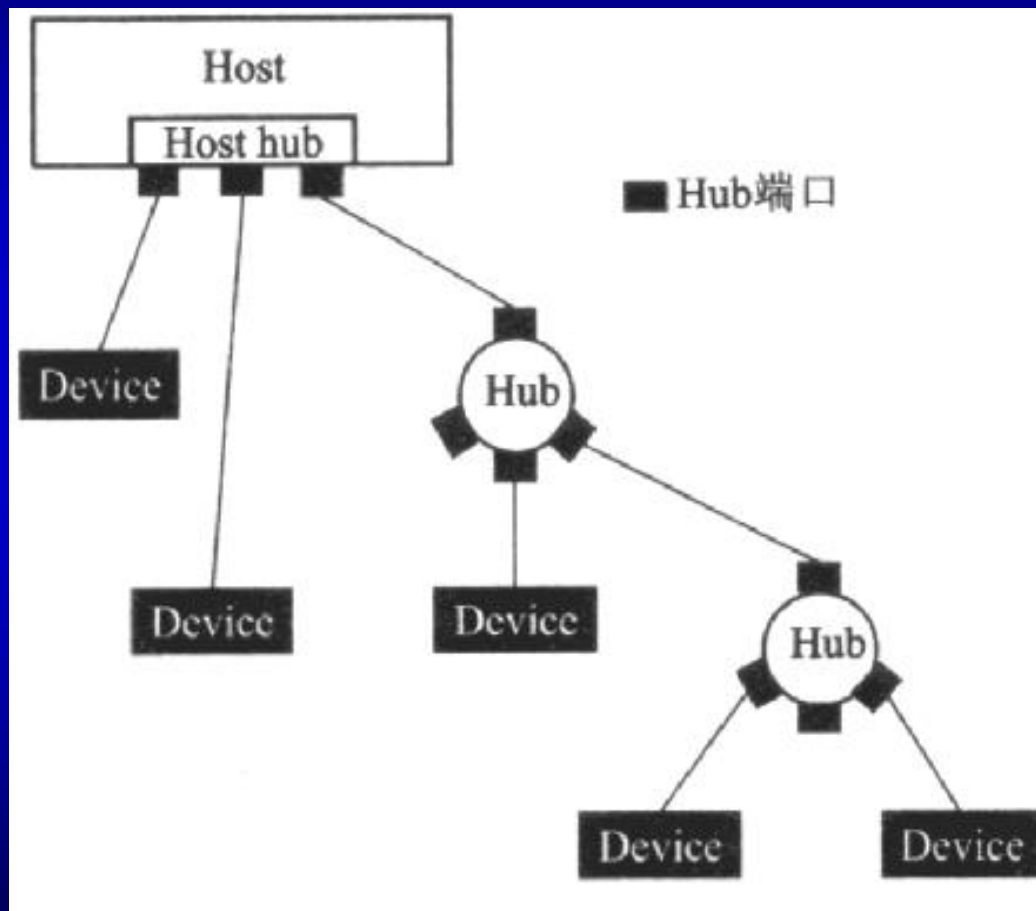
2.0 (草案)版本 于1999年10月5日发布。制定了High-speed的概念，规格。

2.0版本 于2000年4月27日发布。





四. USB系统拓扑结构





五. USB的特点

1. USB为所有的USB外设提供了单一的、易于操作的标准的连接类型。
2. USB整个的系统只有一个端口和一个中；
3. USB支持热插拔 (hot plug)；
4. USB支持PNP；
5. USB在设备供电方面提供了灵活性；
6. USB提供全速12Mbps的速率和低速1.5Mbps的速率来适应各种不同类型的外设；USB2.0还支持480Mbps 的高速传输速率；





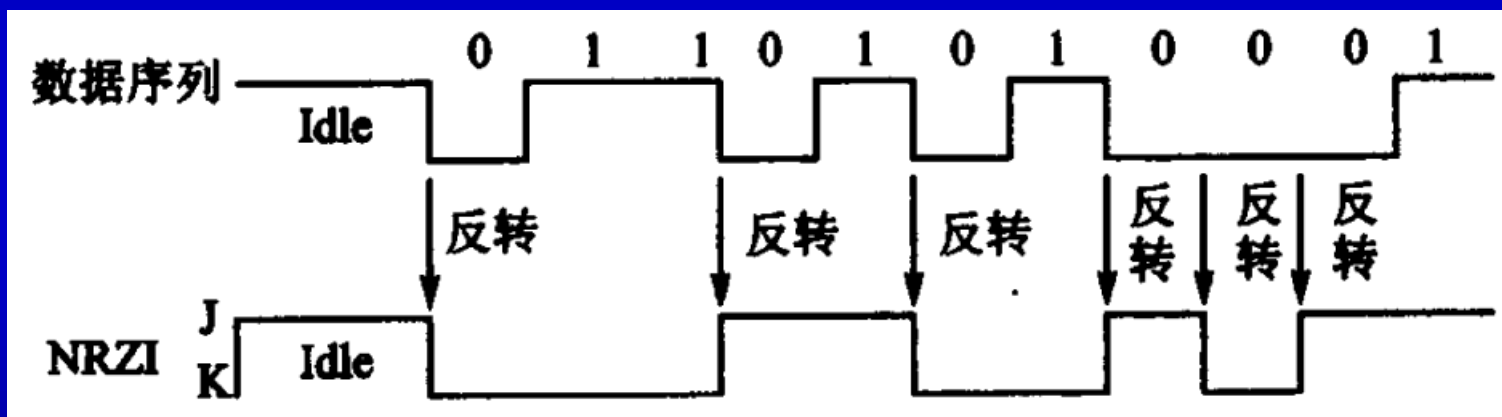
7. USB2.0的High-speed模式支持音频和视频设备，可以保证其固定带宽；
8. 为了适应各种不同类型外围设备的要求USB提供了四种不同的数据传输类型：控制传输，Bulk数据传输，中断数据传输，同步数据传输。同步数据传输可为音频和视频等实时设备的实时数据传输提供固定带宽。
9. USB 的端口具有很灵活的扩展性。一个USB 端口串接上一个USB Hub 就可以扩展为多个USB端口。规范中说，USB可以扩展到127个外设端口。





六、 USB工作原理

1、NRZI编码





2、分组传送

USB技术和IP技术一样，都是基于分组传送方式的。对USB总线来讲，当要传送一个文件时，程序按照USB协议，首先把数据分成若干块，然后在每块数据前面添上同步信号、包标识，后面添上CRC校验，就形成了USB封包。一个文件可能可能有多个封包。全速USB总线把1毫秒作为一个时间帧，总线在一个帧内依次传送不同文件的封包。因此从宏观上，就仿佛总线同时对不同的USB外设进行数据传送。

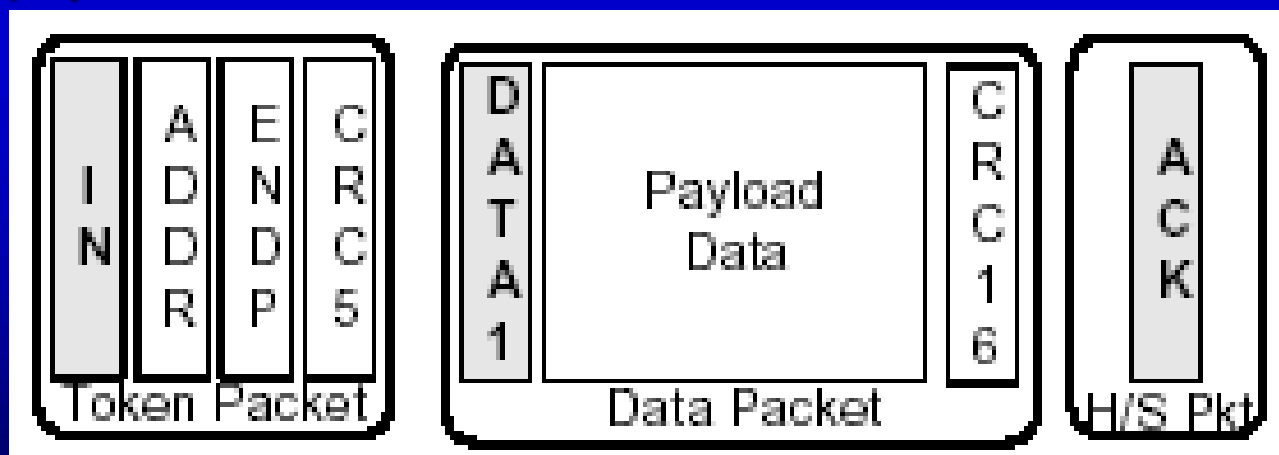




3、USB封包

根据信息包所实现的功能可分为3种类型：
令牌包、数据包和握手包。

令牌包定义了数据传送的类型。比如图
所示。





七、USB传输方式

1、USB的四种传输方式

(1) 同步(isochronous)方式传输 同步传输提供了确定的带宽和间隔时间, 以固定的传输速率, 连续不断地在主机与USB 设备之间传输数据, 在传送数据发生错误时, USB并不处理这些错误, 而是继续传送新的数据, 它被用于时间严格并具有较强的容错性的流数据传输或者用于要求恒定的数据传输率的即时应用中。





(2) 中断传输方式(Interrupt) 该方式传送的数据量很小,但这些数据需要及时处理,以达到实时效果。此方式主要用于定时查询设备是否有中断数据要传输。

(3) 控制传输方式(Control) 控制传输是双向传输,数据量通常较小,用来处理主机到USB设备的数据传输。包括设备控制指令、设备状态查询及确认命令。

(4) 批(Bulk)传输方式 主要应用在数据大量传输和接受数据上,同时又没有带宽和间隔时间要求的情况下,要求保证传输正确无误的数据。





2、USB设备类型

设备类型(device class)	设备举例	类型常量(Class constant)
音频(audio)	扬声器	USB_DEVICE_CLASS_AUDIO
通信	MODEM	USB_DEVICE_CLASS_COMMUNICATIONS
HID	键盘、鼠标	USB_DEVICE_CLASS_HUMAN_INTERFACE
图像	照相机、扫描仪	USB_DEVICE_CLASS_IMAGE
显示	监视器	USB_DEVICE_CLASS_MONITOR
物理回应设备	动力回馈式游戏操纵杆	USB_DEVICE_CLASS_PHYSICAL_INTERFACE
电源	不间断电源供应	USB_DEVICE_CLASS_POWER
打印机		USB_DEVICE_CLASS_PRINTER
Bulk 存储器	硬盘	USB_DEVICE_CLASS_STORAGE
HUB		USB_DEVICE_CLASS_HUB





八、USB设备列举

在USB规范中有一个非常重要的“动作”或“过程”。这个动作将会让PC机知道何种USB设备刚接上以及其所含的各种信息。这样，PC机就可以与这个USB设备开始进行数据传输的工作了。这个动作称之为设备列举(enumeration)。





1、USB描述符

在设备列举过程中，USB设备向主机发送了一个重要的信息，使主机知道了这个设备是“谁”，并启动了该设备的驱动程序。这个信息就是USB描述符





2、设备列举

设备列举包含两方面的功能，一是主机搜集USB设备的信息，也就是USB描述符。另一个是主机根据USB描述符对设备进行配置，比如USB设备地址设置，USB设备各端点数据传输方式设置。

