

# 串行通信技术学习要点

1. 了解串行通信的基本原理
2. 了解典型串行通信接口原理
3. 掌握PC机异步串行接口通信方法

2008-3-26

# 串行通信技术主要内容



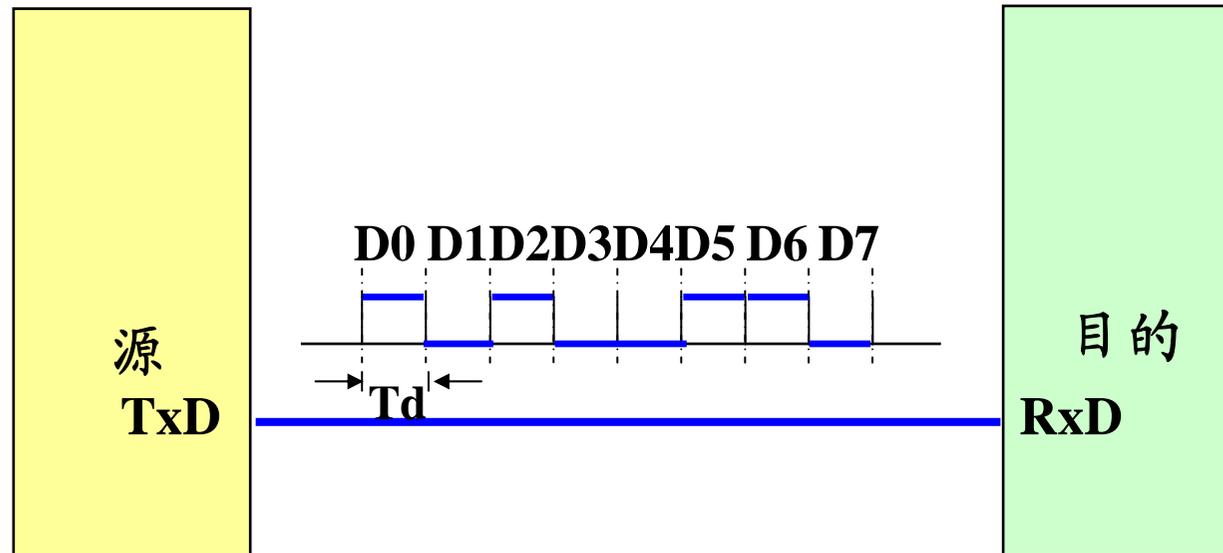
- 一、串行通信基础
- 二、串行通信接口简介
- 三、PC机的标准异步串行接口
- 四、利用标准串口进行数据通信

# 一、串行通信基础

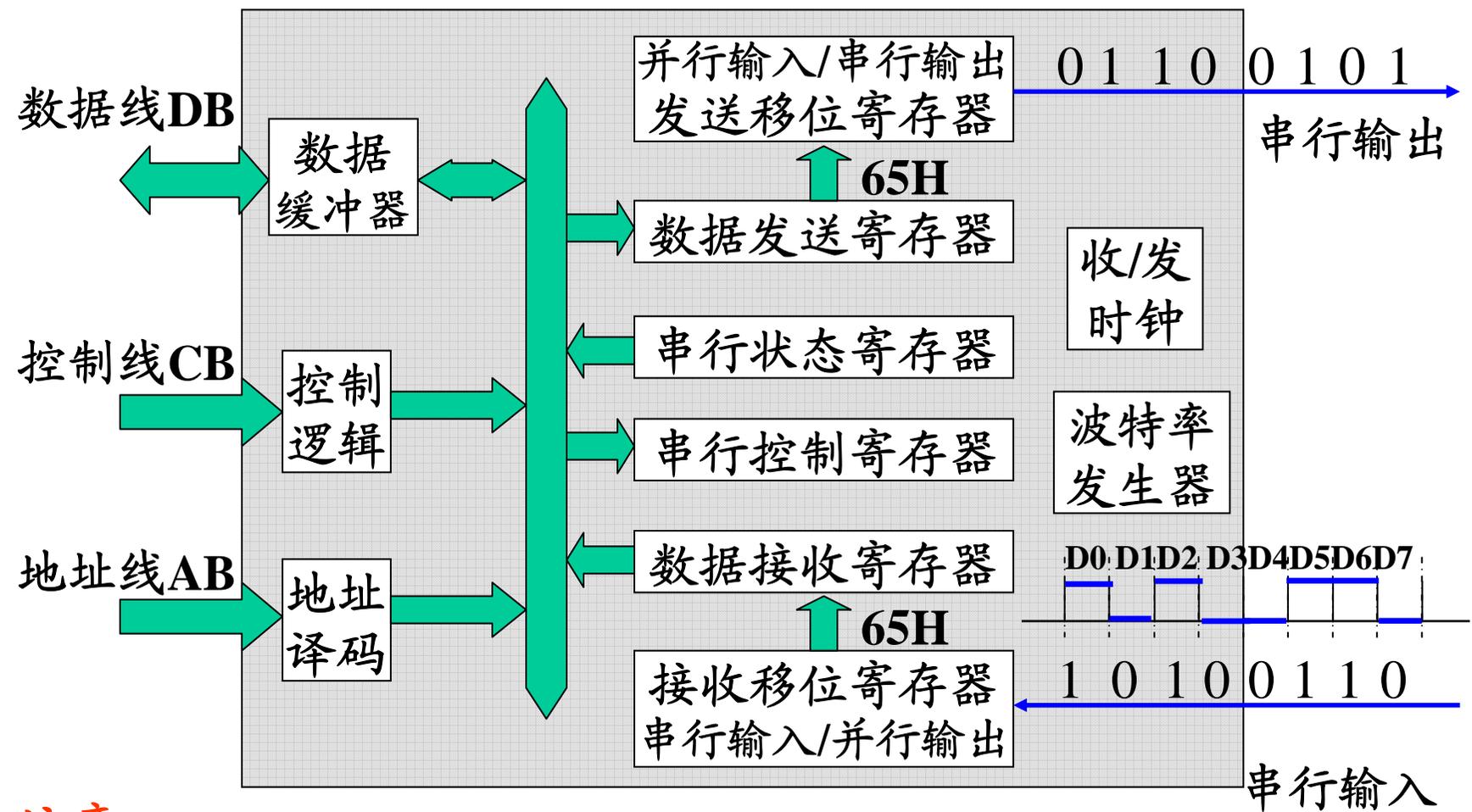
1. 串行通信的概念
2. 信息传输的检错和纠错
3. 串行数据传送方式
4. 波特率
5. 通信协议
6. 异步串行通信和同步串行通信
7. 接收/发送时钟和波特率因子

# 1. 串行通信的概念

指数据一位接着一位在一根线路上传输，  
在传输过程中，每一位数据都占据一个固定的时间长度。



# 串行通信接口



**注意:**

“串行”是外设与接口电路之间的数据传送方式，  
在CPU与接口之间仍是并行工作方式。

## 2. 信息传输的检错和纠错

- 串行数据在传输过程中，由于干扰可能引起信息的出错

如何发现传输中的错误，叫**检错**

发现错误后，如何消除错误，叫**纠错**

- 最简单的检错方法是奇偶校验，

即在传送字符的各位之外，再传送1位奇/偶校验位。

可采用奇校验或偶校验。

**奇校验**：使所有传送的数位(含校验位)中1的个数为奇数

**偶校验**：使所有传送的数位(含校验位)中1的个数为偶数

- 奇偶校验能够检测出1位误码，但是不能纠错。

### 3. 串行数据传送方式

按照数据流的方向，  
分成三中基本的传送方式：

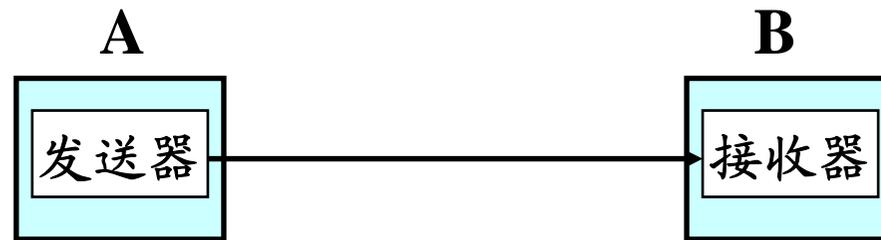
单工方式

全双工方式

半双工方式

## 单工方式

只允许数据按照一个固定的方向传送。

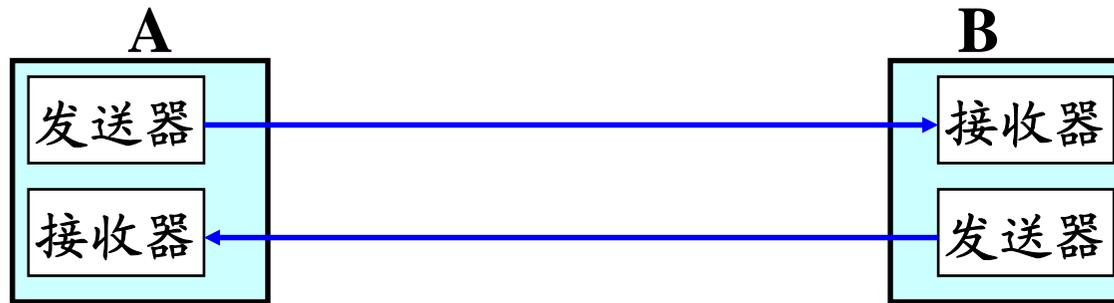


A方只能发送，称**发送器**

B方只能接收，称**接收器**

## 全双工方式

允许通讯双方同时进行发送和接收操作

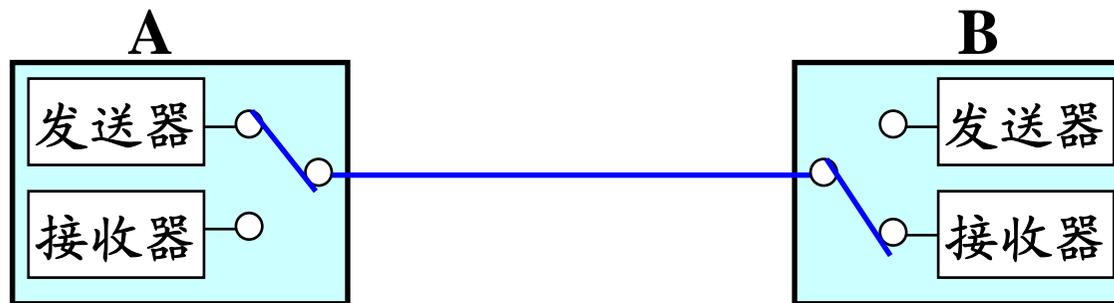


相当于把两个方向相反的单工方式组合在一起，  
需要两条传输线。

全双工方式主要应用于实时性较强的交互式应用中，  
如计算机之间的通信等。

## 半双工方式

数据能从A方传送到B方，也能从B方传送到A方，  
但是不能同时在两个方向上传送，  
每次只能由一方发送，另一方接收。



通信双方通过软件控制的电子开关，进行方向的切换，  
轮流地进行发送和接收。

一些简单的外部设备如键盘与主机的通信采用半双工方式。

## 4. 波特率 (Baudrate)

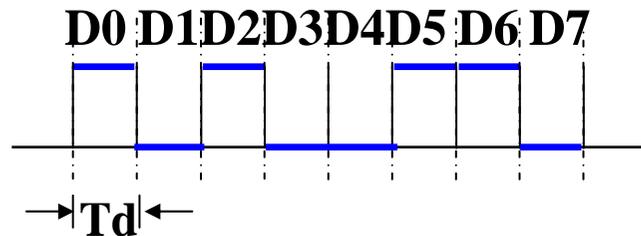
并行通信中，传输速率是以每秒多少字节(B/s)表示。

串行通信中，衡量数据传输速率的单位是波特率，

即每秒传送的二进制数据的位数，以位/秒(bps)表示。

有时也用“位周期” $T_d$ 表示传输速率，波特率是位周期的倒数。

$$\text{波特率} = 1/T_d$$



标准的波特率系列有：

110、300、600、1200、1800、2400、4800、9600和19200等。

## 5. 通信协议

要想保证通信成功，通信双方必须有一系列的约定，如：作为发送方，必须知道什么时候可以发送信息，对方是否收到，收到的内容有没有错，要不要重发、怎么通知对方结束等作为接收方，必须知道对方是否发送信息、发的是什么、收到的信息是否有错、如果有错怎么通知对方、怎么判断结束等。

通信协议就是通信双方为了保证通信正确，事先对数据传送控制规定的必须共同遵守的一种约定，包括对数据格式、同步方式、传送速率、传送步骤、纠错方式以及控制字符定义等问题做出的统一规定。

## 6. 串行通信基本方式

根据对数据流的分界、定时和同步方式的不同，  
串行通信的基本方式可以分为两种类型：

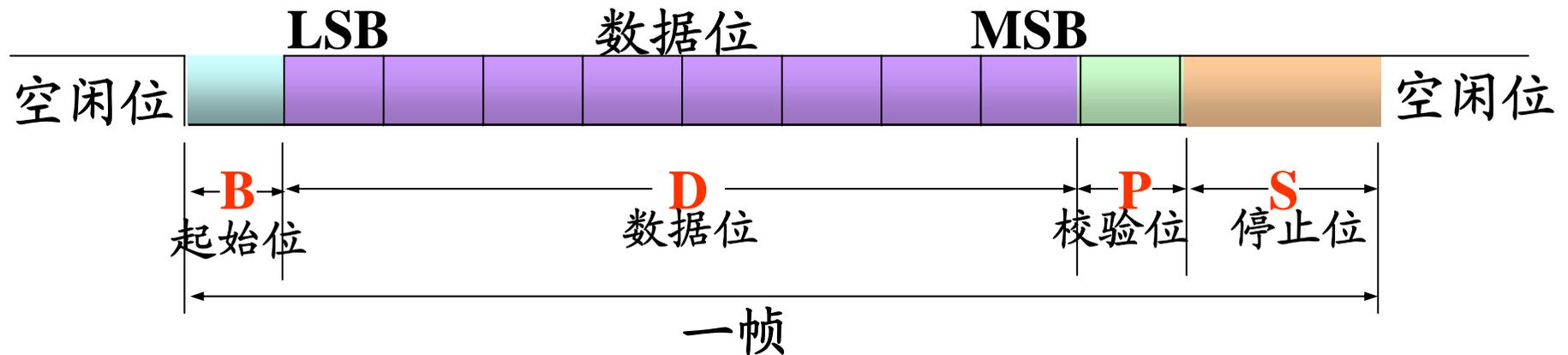
异步串行通信

同步串行通信

# 异步串行通信

数据是一帧(Frame)一帧传送，  
 每一帧包含起始位、数据位、校验位、停止位，  
 帧与帧之间可有任意个空闲位。

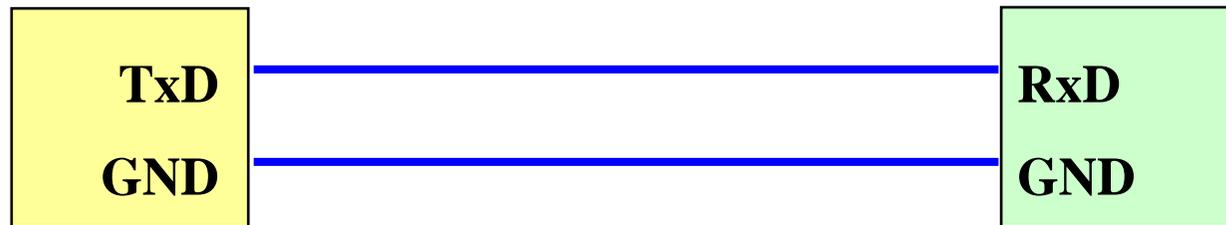
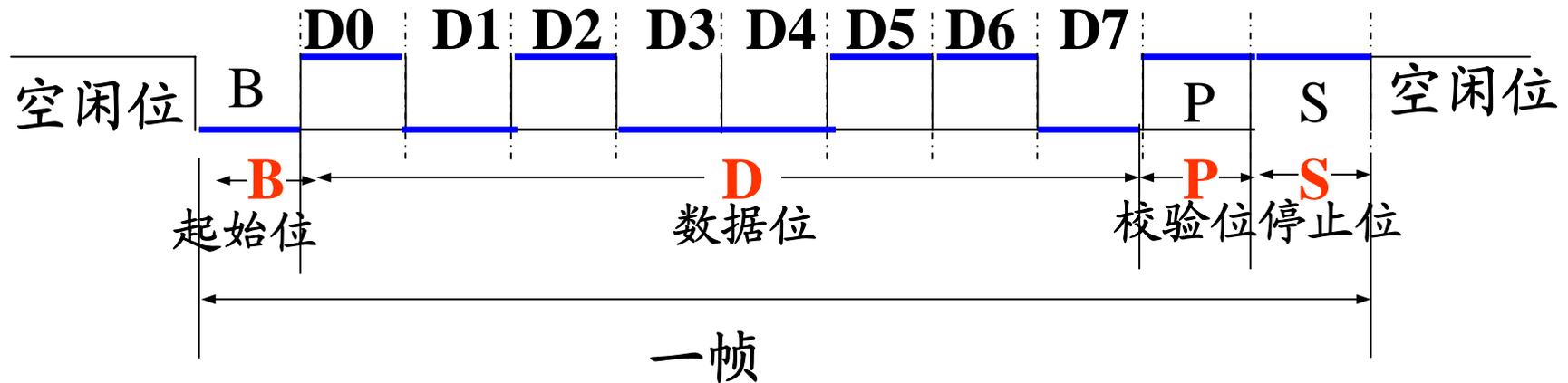
异步串行数据格式为：



起始位B	逻辑0	1位
数据位D	逻辑0或1	5位~8位
校验位P	逻辑0或1	1位或无
停止位S	逻辑1	1位或2位
空闲位	逻辑1	任意数量

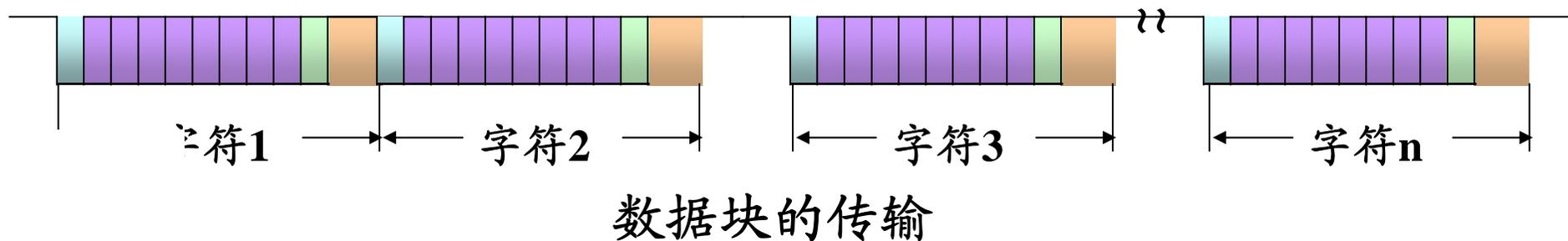
例

传送8位数据65H(01100101B)，奇校验，1个停止位，  
信号线上的波形为：



## 异步串行通信特点

- 异步串行通信以帧为信息单位传送，1帧包含1个字符
- 在数据格式中设置起始位和停止位来协调“同步”
- 异步串行通信的“异步”，指字符与字符之间通信没有严格的定时要求，每个字符作为一个独立的单位，可以随机出现在数据流中，即每个字符在数据流中出现的时间是任意的
- 位与位之间有严格的定时  
一旦传送开始，收/发双方则按预先约定的传输速率，在时钟的作用下，传送字符中的每1位
- 异步串行通信适合于发送数据不连续、传送数据量较少，或对传输率要求不高的场合。

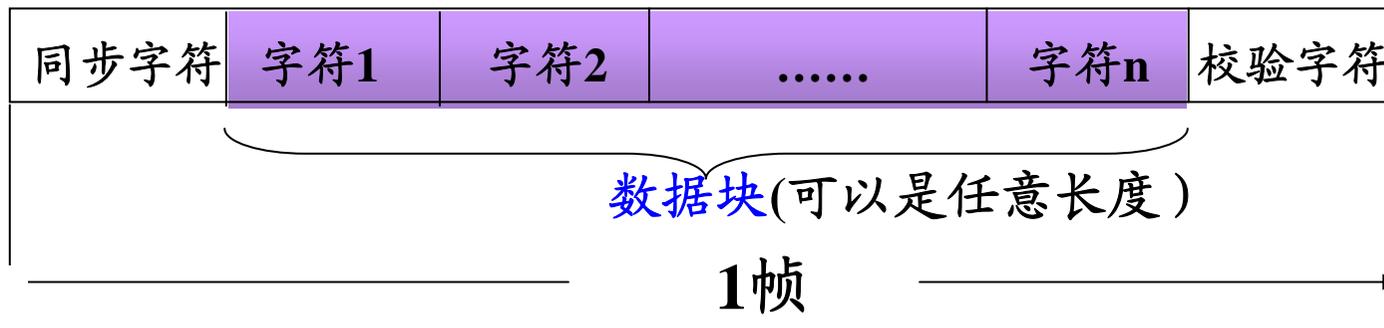


# 同步串行通信

同步串行通信是帧为单位传送，每帧包含1个数据块，每个数据块内可包含成百上千个字符，每个字符取相同的位数，字符之间可以是连续的，没有起始位和停止位。

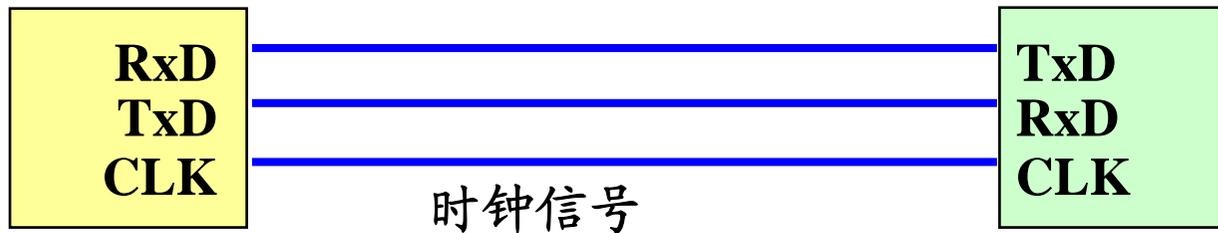
在数据块的前面一般设置有1个或2个同步字符，作为帧的边界和通知对方接收的标志，尾部是校验字符，用于校验数据传输的差错。

## 同步串行通信数据格式



## 同步串行通信特点

- 字符与字符之间和字符内部位与位之间都是同步传送，对时钟要求非常严格。
- 收/发两端必须使用同一时钟来控制数据块的传输，即使用同一时钟来触发双方移位寄存器的移位操作。
- 在近距离通信时可以在传输线上增加一根时钟信号线，在远距离通信时可以通过调制/解调器，在数据流中发送/提取同步信号，使发送和接收方得到完全相同的时钟信号。



1步串行通信适用于要求快速、大量数据传送的场合。

## 7. 接收/发送时钟和波特率因子

### 接收/发送时钟

在串行通信中，无论发送或接收，都必须有时钟脉冲信号对传送的数据进行**同步**和**定位**控制，这就需要有接收/发送时钟。

**同步**，就是**确认通信什么时候开始**

**定位**，就是**确认每1个的数据位**

## 波特率因子

为了提高串行通信的抗干扰能力，  
往往用多个时钟调制1个二进制数据，  
调制1个二进制数据所用的收/发时钟个数称为波特率因子，  
用n表示波特率因子，则：

$$\text{收/发时钟频率} = \text{波特率} \times \text{波特率因子}n$$

**例如** 波特率因子为16，则16个时钟脉冲移位1次

当 $n=16$ 时，

接收器以数据波特率16倍的时钟对所接收的数据进行检测：

首先正确地检测到起始位，然后逐位确定各个数据位。

具体过程如下：

接收器在每个接收时钟的上升沿采样接收数据线，

当发现接收数据线出现低电平时，就认为是起始位的开始，

若在连续的8个时钟周期内检测到接收数据线仍保持低电平，

则确认它为起始位，而不是干扰信号。

通过这种方法，

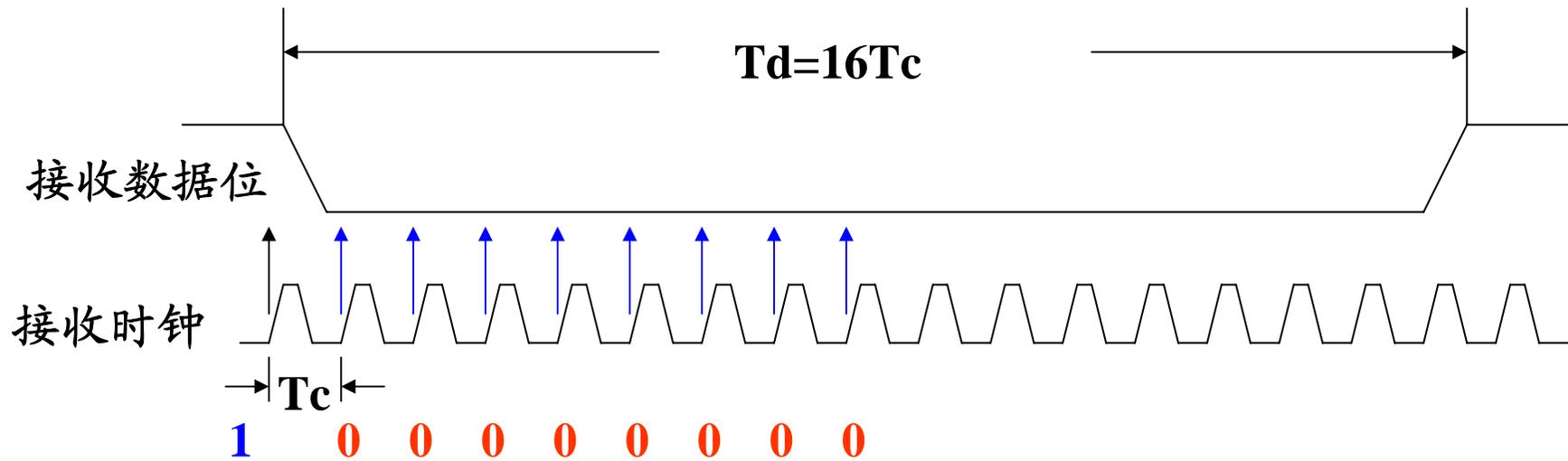
不仅能够排除接收线上的噪声干扰，识别假起始位，

而且能够相当精确地确定起始位的中间点，

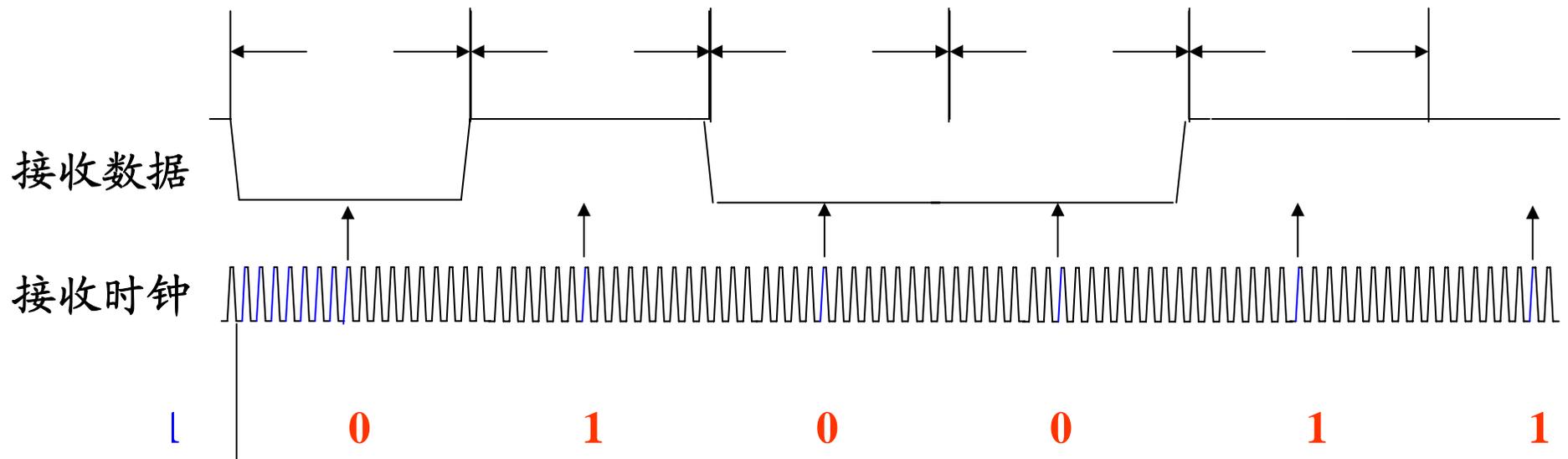
从而提供一个准确的时间基准，

从这个基准算起，每个16个 $T_c$ 采样一次数据线，作为输入数据。

### 起始位的确定



### 数据位的采样



# 串行通信技术主要内容

一、串行通信基础



二、串行通信接口简介

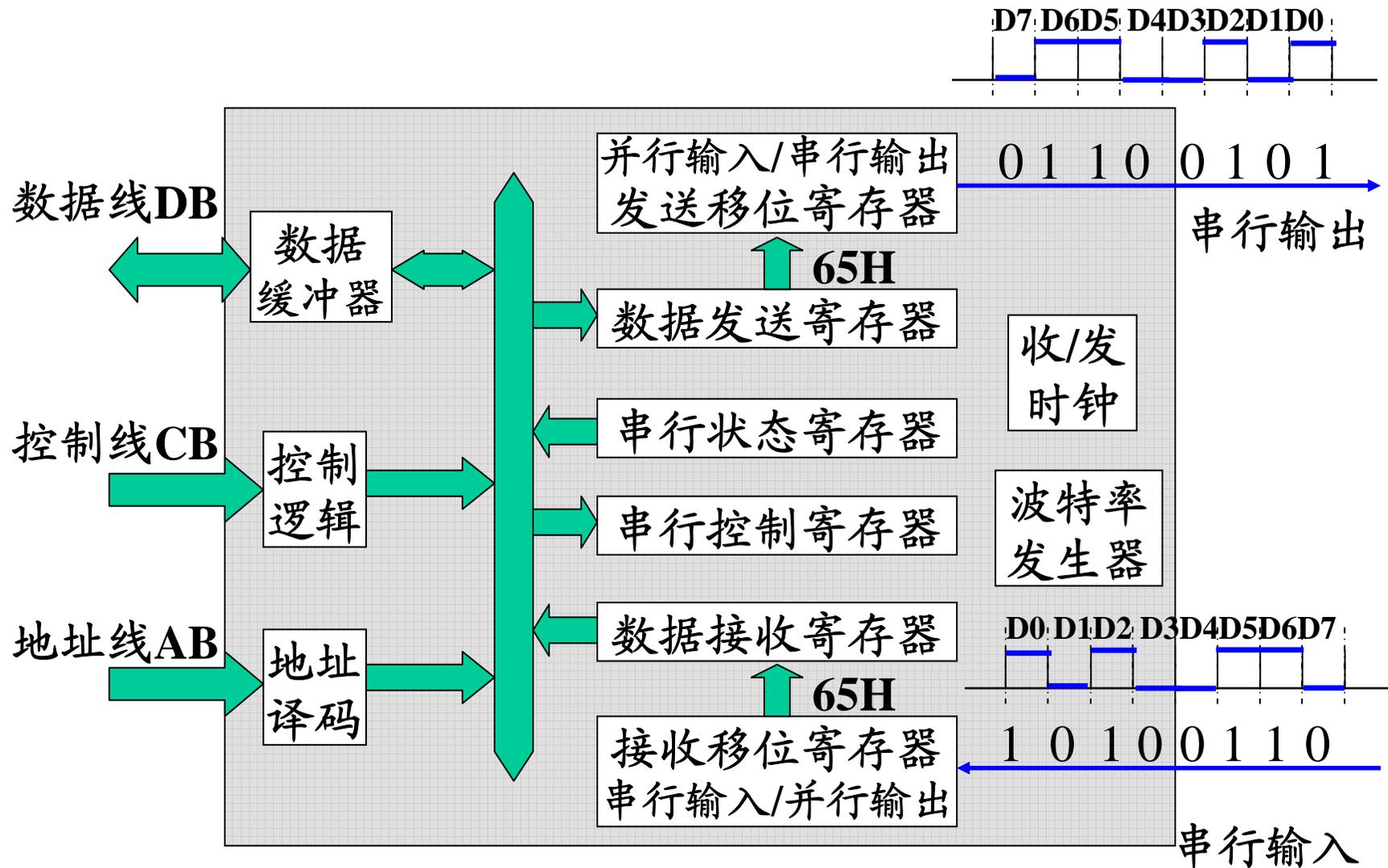
三、PC机的标准异步串行接口

四、利用标准串口进行数据通信

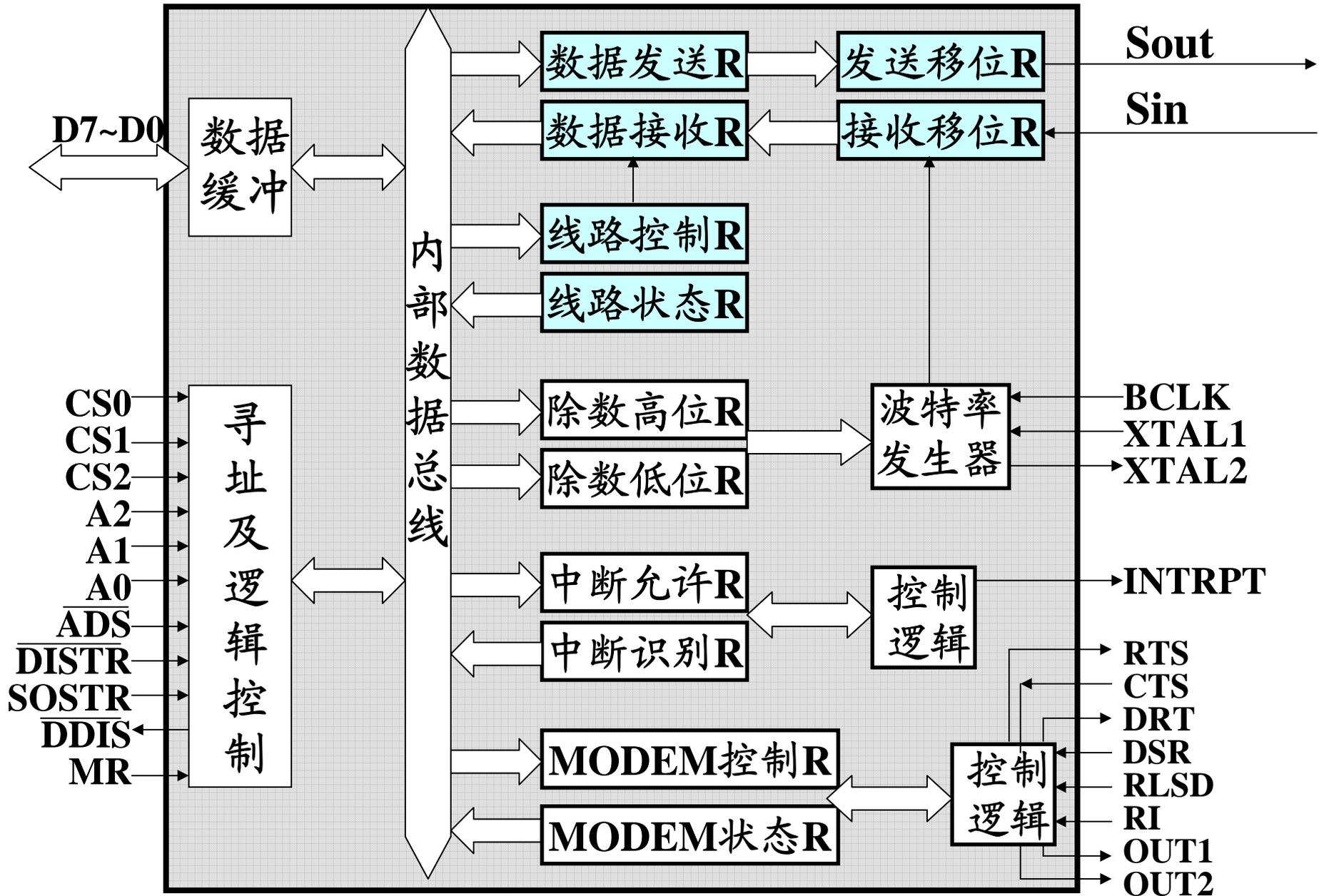
## 二、串行通信接口简介

1. 串行通信接口典型结构图
2. 串行通信有关寄存器
3. 串行发送数据过程
4. 串行接收数据过程

# 1. 串行通信接口典型结构图



# 例 异步通信接口INS8250结构图



## 2. 串行通信有关寄存器

- 发送数据寄存器
- 发送移位寄存器
- 接收移位寄存器
- 接收数据寄存器
- 串行控制寄存器
- 串行状态寄存器



## 发送数据寄存器

用于存放要发送的并行数据，  
如检测到发送移位寄存器为空，  
自动将保存的数据传送到发送移位寄存器。

## 发送移位寄存器(并行输入/串行输出)

用于将并行数据转换成串行数据，  
并通过串行输出管脚发送数据。  
CPU不能直接对发送移位寄存器存取操作。



### 接收移位寄存器(串行输入/并行输出)

用于接收从串行输入管脚传入的数据，并转换成并行数据。

当接收到1字符的串行数据，就自动传送给接收数据寄存器。

**CPU不能直接对接收移位寄存器存取操作。**

### 接收数据寄存器

用于存放由接收移位寄存器传送来的数据，

**CPU通过对接收数据寄存器进行读操作，获取传入的数据。**

## 串行控制寄存器

设置串行通信的数据格式，

包括波特率、停止位长度、数据位长度、奇偶校验方式的设置

异步串行数据格式为：



## 例如 某串行通信接口控制寄存器的含义

<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

➤ **D7~D5: 波特率设置**

<b>000 /110bps</b>	<b>001/150bps</b>	<b>010/300bps</b>	<b>011/600bps</b>
<b>100 /1200bps</b>	<b>101/2400bps</b>	<b>110/4800bps</b>	<b>111/9600bps</b>

➤ **D4~D3: 奇偶校验设置**

<b>00/无校验</b>	<b>01/奇校验</b>	<b>10/无校验</b>	<b>11/偶校验</b>
---------------	---------------	---------------	---------------

➤ **D2 : 停止位长度设置**

<b>0: 1位</b>	<b>1:2位</b>
--------------	-------------

➤ **D1~D0: 数据位长度设置**

<b>10: 7位</b>	<b>11:8位</b>
---------------	--------------

## 串行状态寄存器

存放串行通信的状态，

包括发送数据寄存器，接收数据寄存器，通信过程的状态等。

### 通信口状态参数含义

<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

- **D6 : 1=发送数据寄存器空**
- **D3 : 1=帧格式错误**
- **D2 : 1=奇偶校验错误**
- **D1 : 1=溢出错误**
- **D0 : 1=接收数据寄存器满**



串行状态寄存器

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

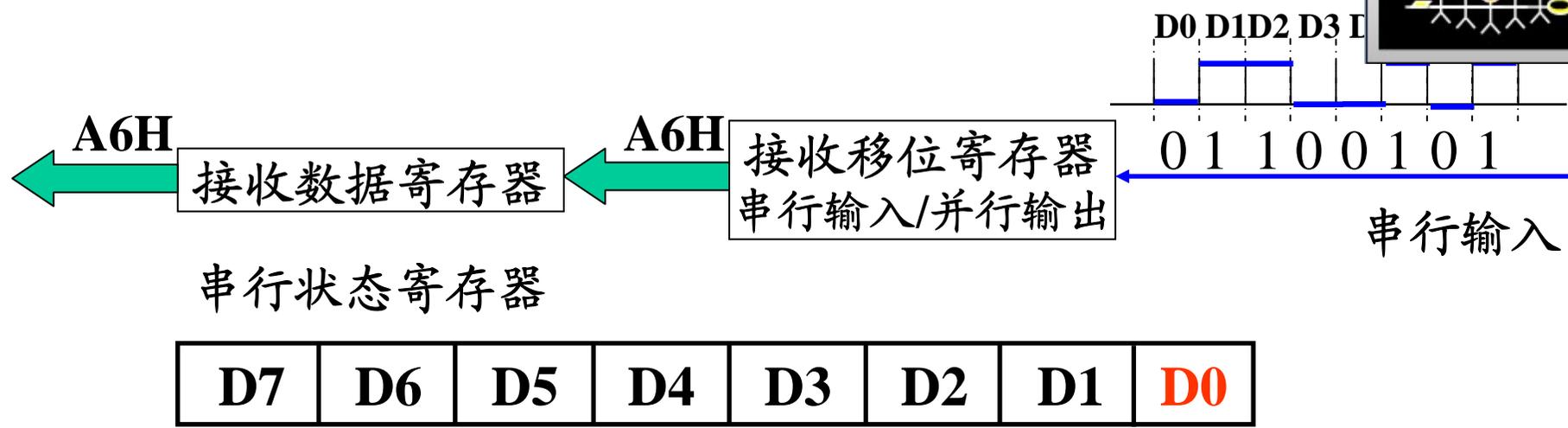
## 发送数据寄存器的状态

**满:**当发送数据寄存器的数据未传送到移位寄存器处于满状态;

**空:**当发送数据寄存器的数据传送到移位寄存器,

新的数据还未写入发送数据寄存器时, 未处于空状态

当发送数据寄存器为**空状态**时, 可以**发送新数据**。



## 接收数据寄存器的状态

**满:** 当接收移位寄存器从串行输入管脚接收一个字节的数据，并转换成并行数据自动存放到接收数据寄存器后，接收数据寄存器处于满状态。

**空:** 当接收数据寄存器的数据被取走，但新的数据未传入，数据接收寄存器处于空状态。

接收数据寄存器为**满状态**时，可以**接收新数据**。

## 接收/发送通信过程状态

### 奇偶校验错误

通信线上的噪音引起某些数据位的改变，  
使接收数据位的奇偶数与设定的奇偶校验方式不一致，  
从而产生奇偶校验错误。

**例** 传送方发送8位数据65H(01100101B)，奇校验，1个停止位，  
数据格式为 **01010011011** (含校验位在内1的个数5为奇数)

如果接收方按同样的通讯协议接收数据，  
假如收到的数据为 **01011011011**  
由于收到1的个数为6，为偶数，产生奇偶校验错误。

意最后1位“1”是停止位，不在检验数据之内)

## 串行状态寄存器



### 溢出错误

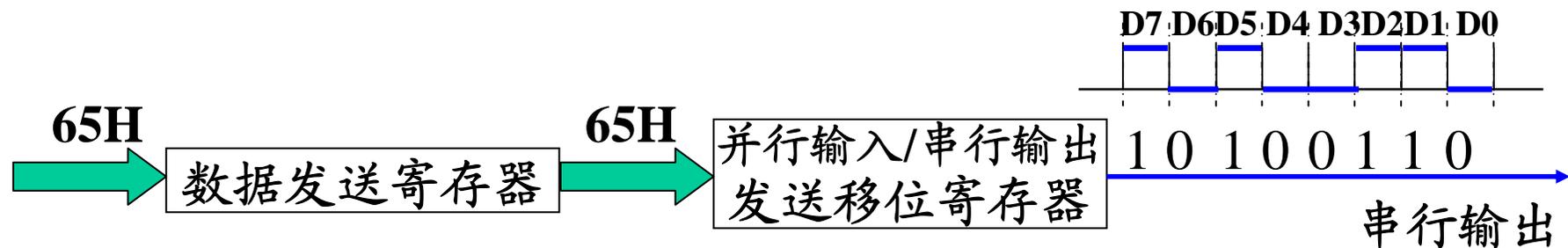
当上一个存在接收数据寄存器的数据还没有被CPU取走，又有字符要传送到接收数据寄存器时，产生接收数据寄存器溢出错误。

### 帧格式错误

当接收的数据没有停止位时，产生帧格式错误，这种错误可能是由于通信线上的噪音引起停止位的丢失，或是由于接收方和发送方通信协议初始化不匹配引起。

### 3. 串行发送数据过程

CPU将待发送的数据写入发送数据寄存器，  
 在发送移位寄存器空时，  
 会自动从数据寄存器传送到移位寄存器，  
 发送控制逻辑将自动按规定的数据格式构成信息帧，  
 然后逐位由串行输出管脚输出；  
 当发送数据寄存器空时，  
 将置相应的状态标志位为“1”，以通知CPU发下一个数据。



## 4. 串行接收数据过程

串行的位信息由串行输入管脚输入，

接收侧监测通信线路，如果检测出一个起始位，就进行内部同步，

开始在接收时钟控制下采样串行输入管脚上的数据，

移位寄存器将接收的串行数据，转换成并行数据，

当一个字符数据接收完，自动存放在接收数据寄存器中，

并置数据接收寄存器满标志位为“1”，通知CPU可以读取数据。



# 串行通信技术主要内容

一、串行通信基础

二、串行通信接口简介



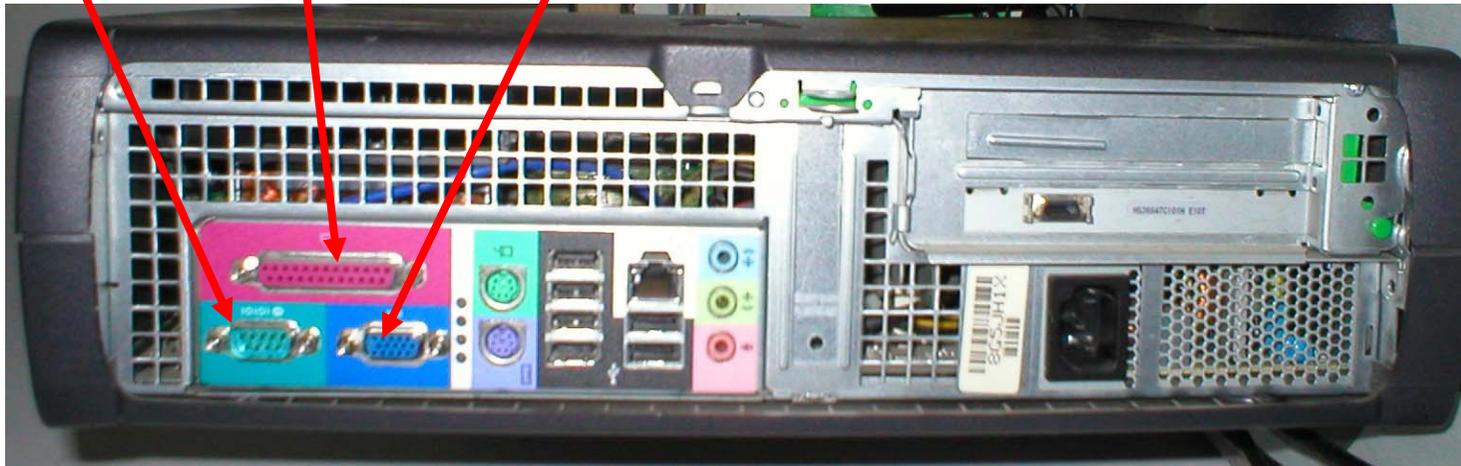
三、PC机的标准异步串行接口

四、利用标准串口进行数据通信

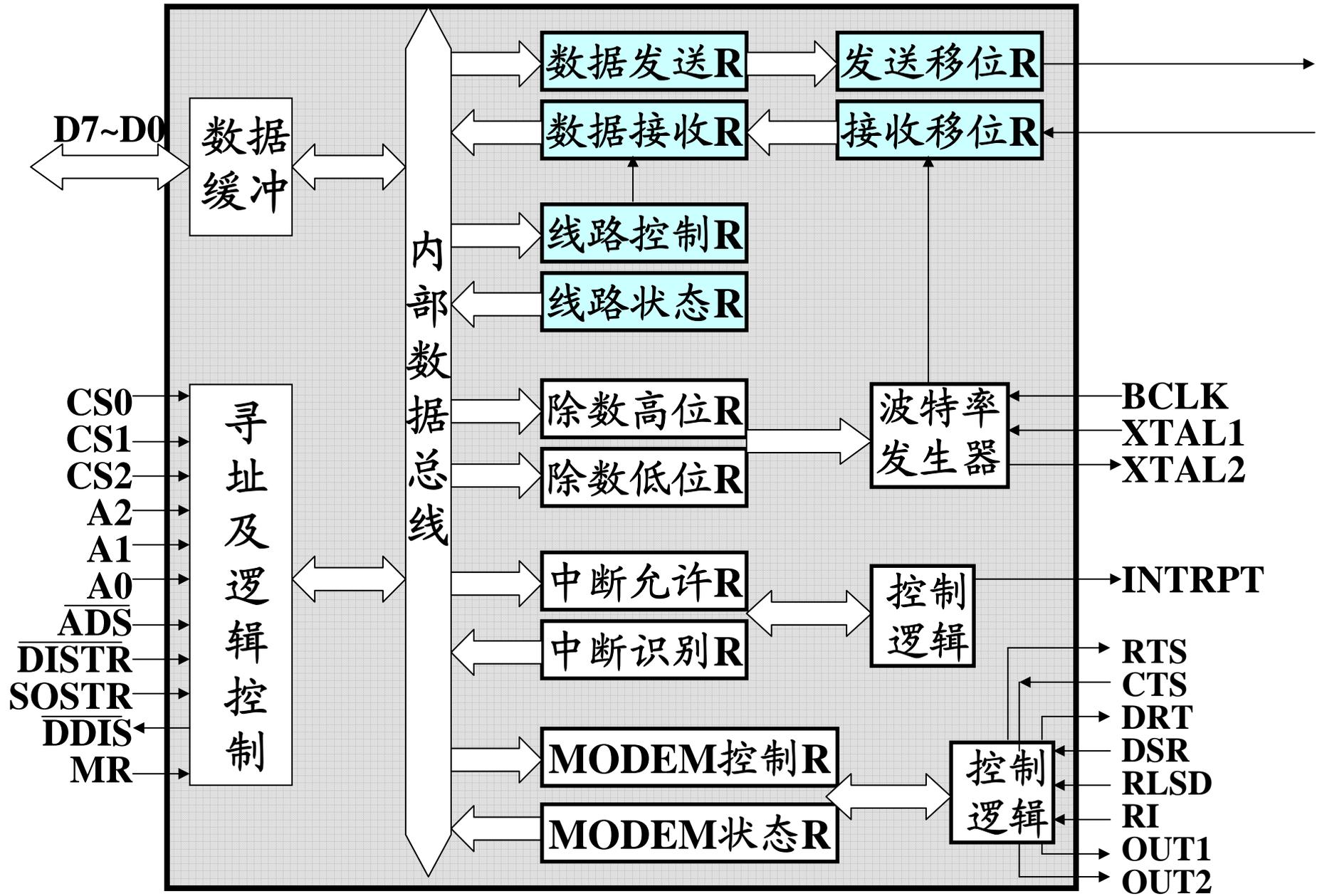
### 三、PC机的标准异步串行接口

1. PC机一般有1或2个异步串行通信接口，  
组装在主板上，称为COM1和COM2
2. 使用的串行通信接口芯片是INS8250，  
或与之兼容的串口芯片(如16550等)。  
8250是一种可编程的异步串行通信接口芯片，  
支持异步通信协议，可通过编程改变传送数据的波特率，  
提供与MODEM连接的联络信号，实现远程通信。
3. 两串口的结构相同，只是占用的系统资源不同，  
COM1: **3F8~3FFh**，中断IRQ4  
COM2: **2F8~2FFh**，中断IRQ3

串口1      并口      串口2

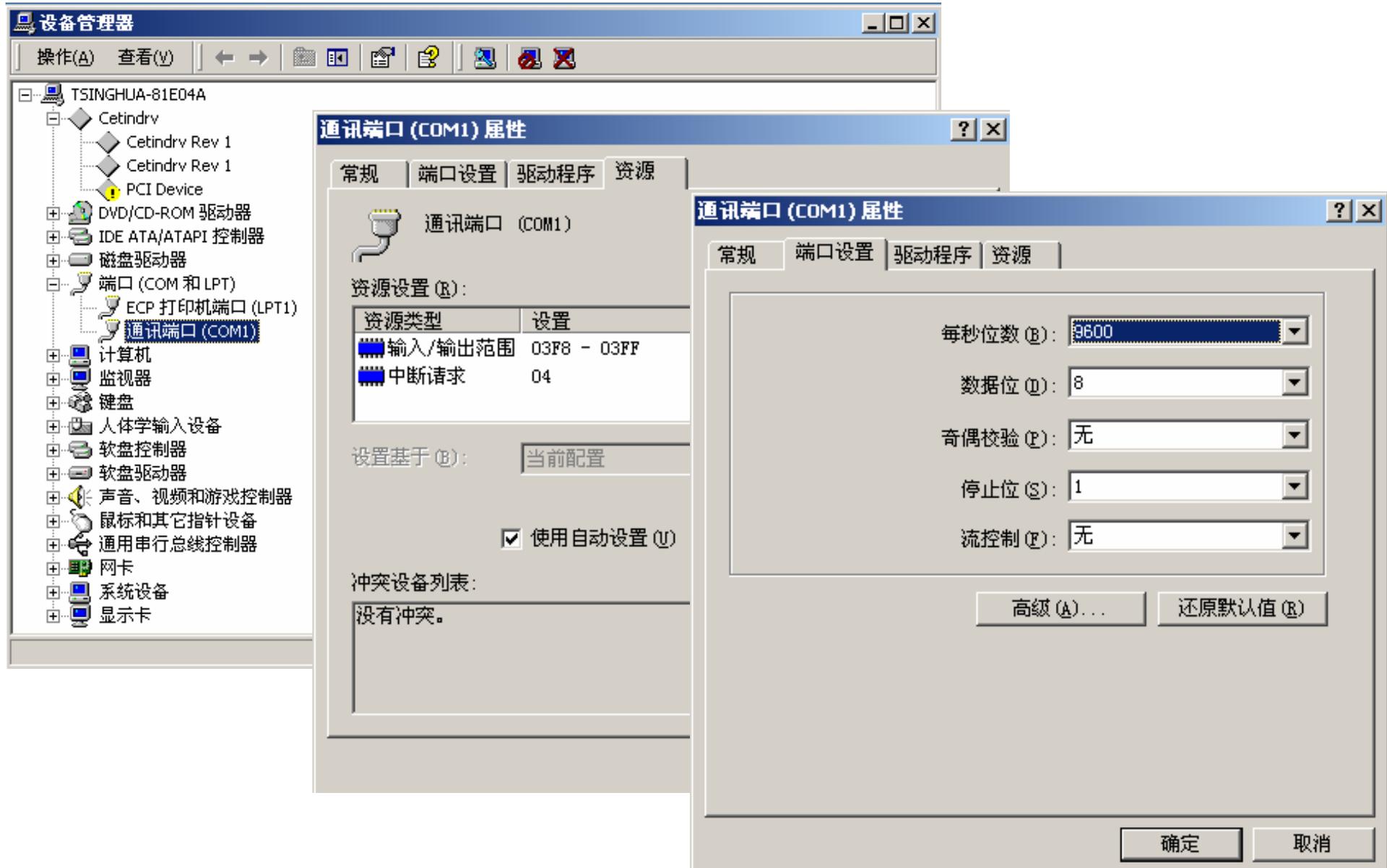


# 异步通信接口 INS8250 结构图



## 在Windows2000下查看串口硬件资源

### 我的电脑/属性/硬件/设备管理/端口/通信端口COM1



The image shows the Windows 2000 Device Manager window with the 'Ports (COM and LPT)' section expanded. The 'Communications Port (COM1)' device is selected. Two overlapping dialog boxes are shown, both titled 'Communications Port (COM1) Properties'.

The first dialog box shows the 'Resources' tab with the following table:

Resource Type	Setting
Input/Output Range	03F8 - 03FF
Interrupt Request	04

The second dialog box shows the 'Port Settings' tab with the following configuration:

- Baud Rate (B): 9600
- Data Bits (D): 8
- Parity (P): None
- Stop Bits (S): 1
- Flow Control (F): None

Buttons at the bottom of the dialog boxes include 'Advanced (A)...', 'Restore Defaults (R)', 'OK', and 'Cancel'.

## 4. 串行通信接口标准RS-232

- 在串行通信中，设备之间的连接要符合接口标准
- 计算机通信中使用最普遍的是：

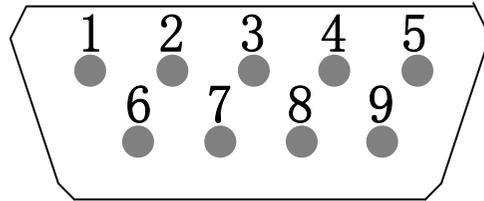
美国电子工业协会(EIA)推荐使用的RS-232C标准

**EIA**——**Electronic Industries Association**  
美国电子工业协会

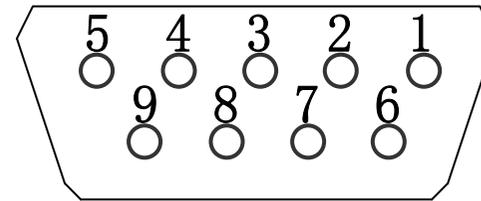
**RS**——**Recommend Standard** 推荐标准

**C**表示第三版本，之前有过**A**、**B**版

- PC机上的COM1、COM2连接器，符合RS-232C接口



针型DB-9M



孔型DB-9F

引脚号	符号	信号名称	方向
1	<b>DCD</b>	载波检测	输入
2	<b>RxD</b>	接收数据	输入
3	<b>TxD</b>	发送数据	输出
4	<b>DTR</b>	数据终端就绪	输出
5	<b>GND</b>	信号地	
6	<b>DSR</b>	数据设备就绪	输入
7	<b>RTS</b>	请求发送	输出
8	<b>CTS</b>	允许发送	输入
9	<b>RI</b>	振铃指示	输出

**RS-232C采用负逻辑，TTL采用正逻辑，**

**RS-232C信号电平与TTL不兼容**

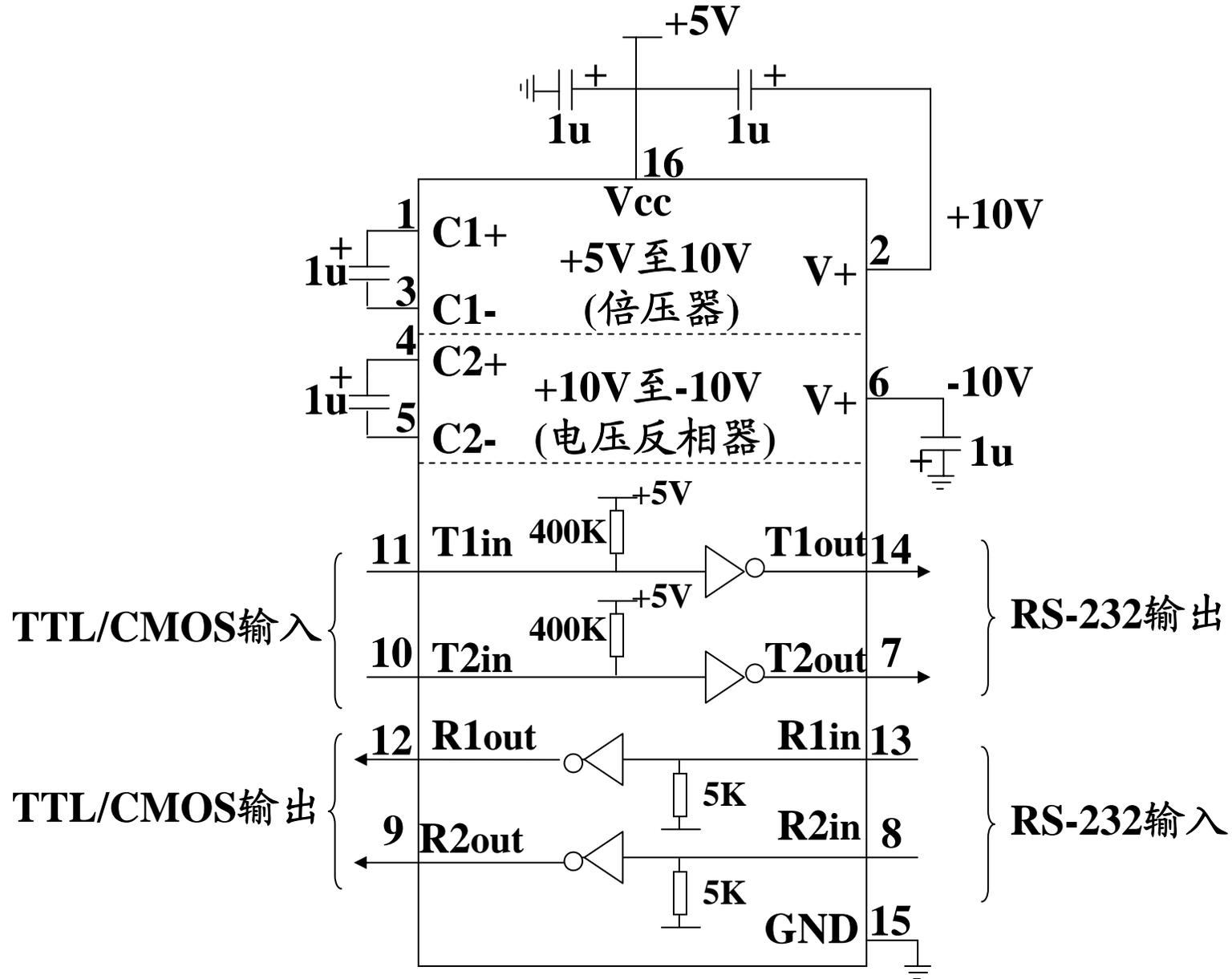
	RS-232C标准	TTL标准
逻辑0	+5 ~ +15V	0.3 ~ 1.0V
逻辑1	-5 ~ -15V	2.5 ~ 5.0V

一般串行接口芯片如8250、8251、

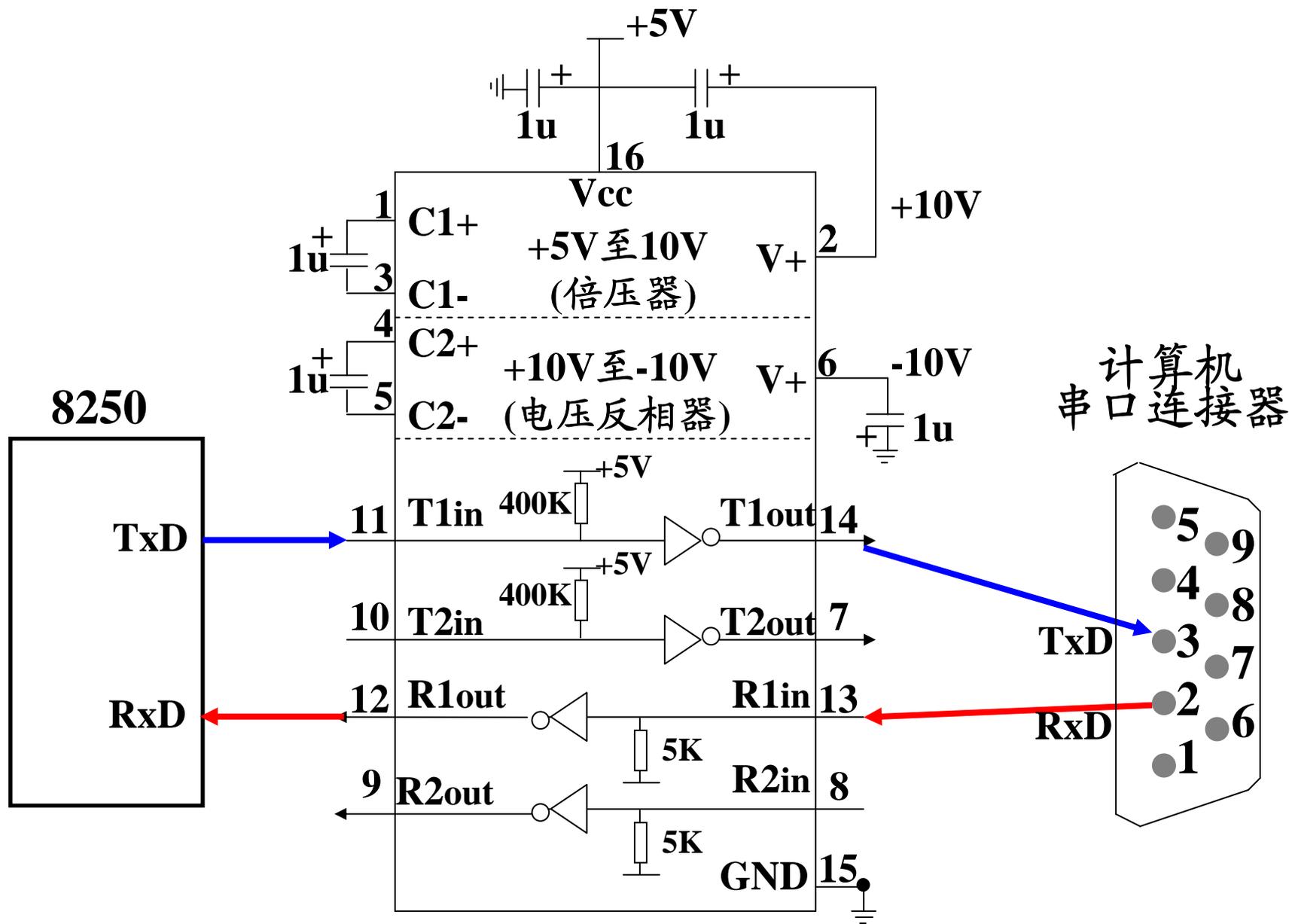
单片机内串行接口均使用TTL电平，

**使用电平转换电路才能与RS-232C连接器连接。**

# MAX232电平转换器

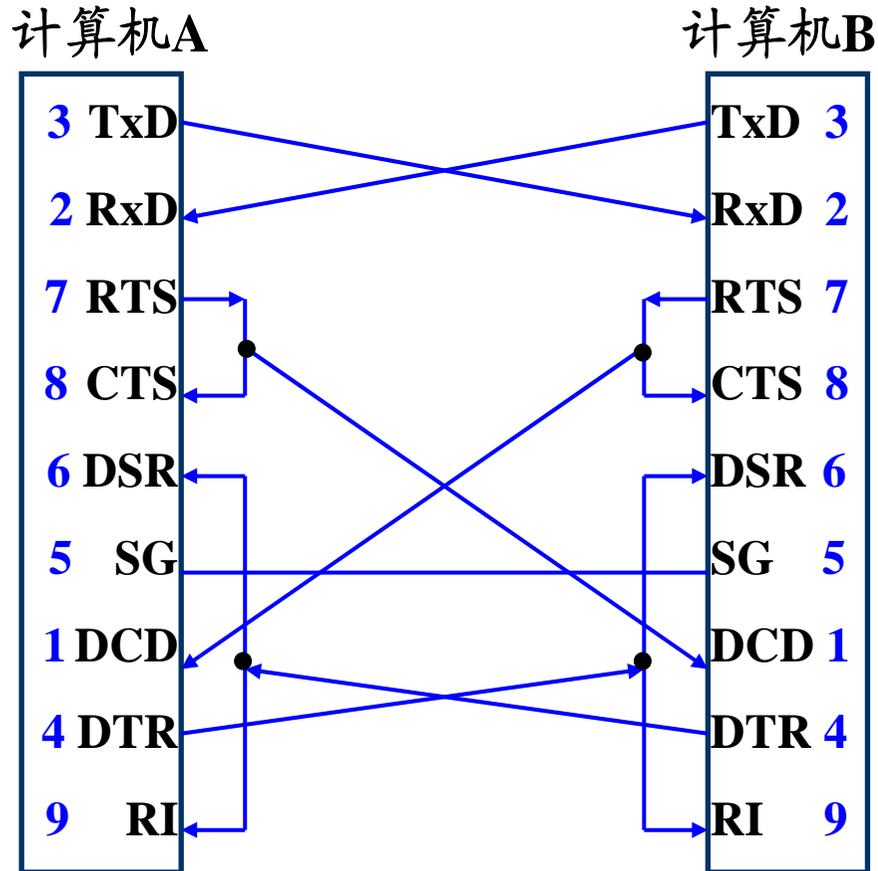


# 利用MAX232电平转换器进行电平转换

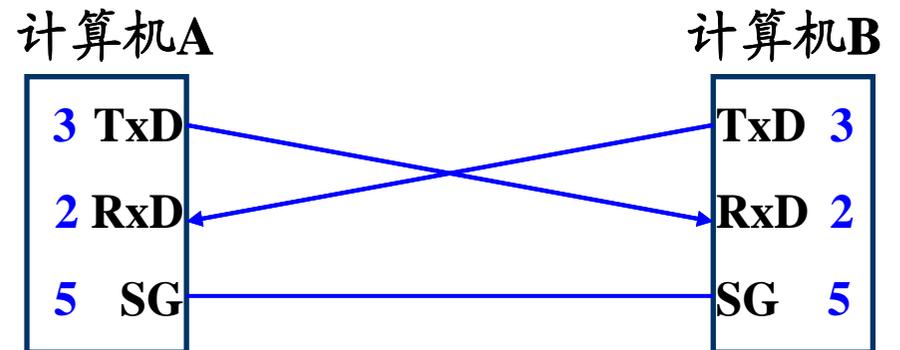


- 两台PC机或设备进行近距离通信时，可直接将串口的信号线对接。
- 当进行远距离通信时，要是用调制解调器(MODEM)连接到电话线上。  
因为RS-232标准串行接口输出的是电压信号，不能直接接到电话线上，调制解调器把代表逻辑1和逻辑0的电压信号转换成能在电话线上除数的不同频率的信号。  
电话线另一端的调制解调器把不同频率的信号转换成接口1电压信号。

## 无Modem的完整连接

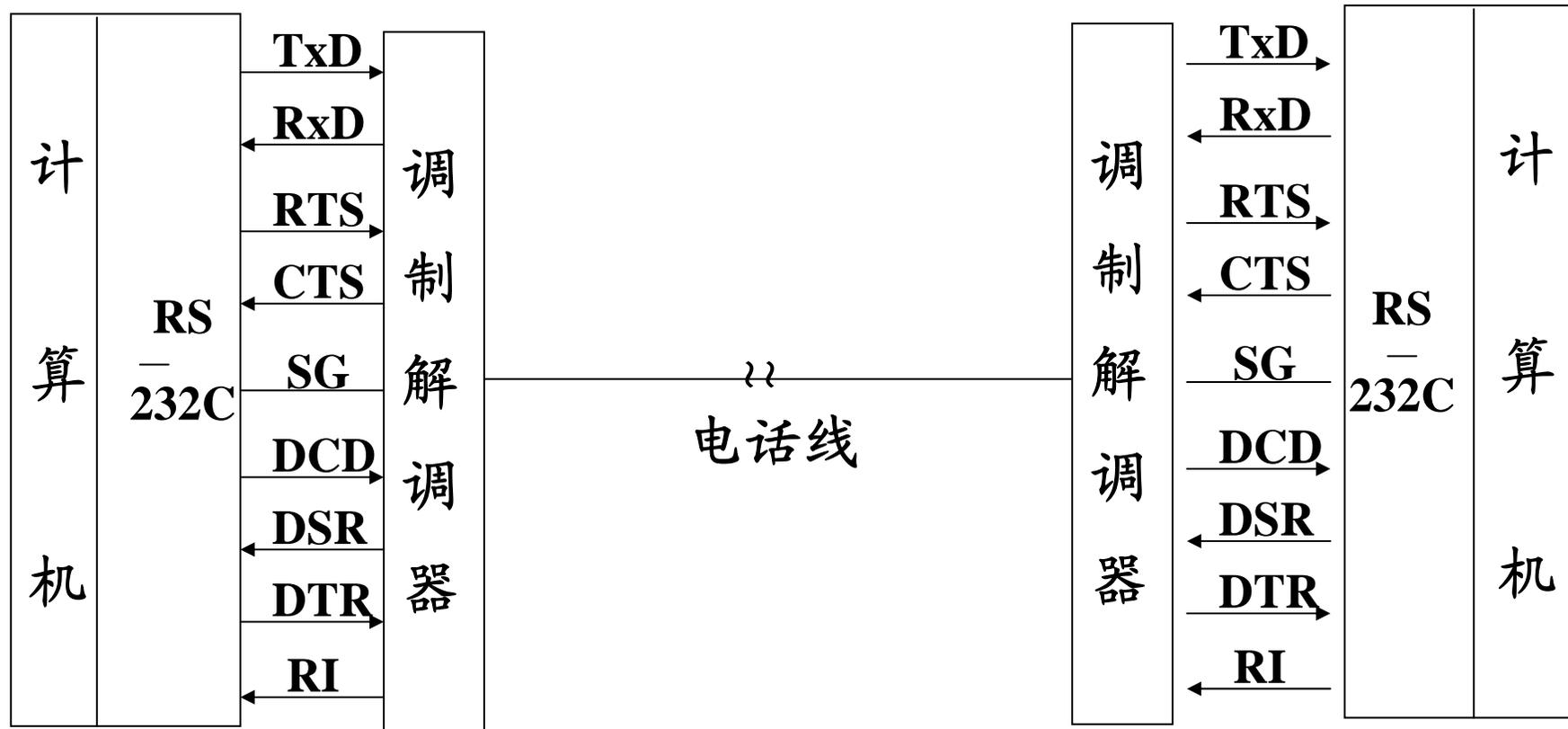


## 无Modem的最简单连接



孔-孔的串行通信线

## 采用Modem和电话网通信时的信号连接



# 串行通信技术主要内容

一、串行通信基础

二、串行通信接口简介

三、PC机的标准异步串行接口



四、利用标准串口进行数据通信

## 四、利用标准串口进行数据通信

- 1、串口通信编程方法
- 2、有关串行通信的**BIOS**功能调用
- 3、查询方式发送、接收数据流程

# 1、串口通信编程方法

## 编程方法1: 采用对端口直接操作

使用该法, 需要了解8250的编程结构和编程方法  
通过对8250的端口(3F8~3FFH)读写,  
实现对串口通信的控制

## 编程方法2: 采用BIOS提供的功能调用实现

使用该法, 不需了解8250的编程结构和编程方法,  
利用BIOS提供的功能调用, 实现对串口通信的控制

## 2、有关串行通信的BIOS功能调用

- 初始化串行通信口
- 读取通信状态
- 向串行通信口发送字符
- 从串行通信口接收字符

## ● 初始化串行通信口

**功能号**            **AH = 00H**

**类型号**            **14H**

**入口参数**        **AL = 初始化参数**

**DX = 通信口号    0: COM1    1: COM2**

**实现功能**        **将初始化参数设置到指定的通信口中**

## 初始化参数含义

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

➤ **D7~D5: 波特率设置**

000 /110bps    001/150bps    010/300bps    011/600bps  
100 /1200bps    101/2400bps    110/4800bps    111/9600bps

➤ **D4~D3: 奇偶校验设置**

00/无校验    01/奇校验    10/无校验    11/偶校验

➤ **D2 : 停止位长度设置**

0: 1位    1:2位

➤ **D1~D0: 数据位长度设置**

10: 7位    11:8位

## ● 读取通信口状态

功能号      **AH = 03H**

类型号      **14H**

入口参数    **DX = 通信口号**

**0: COM1    1: COM2**

返回参数    **AL = 通信口状态**

实现功能    **读取指定通信口的状态存放到AL中**

## 通信口状态参数含义

<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

- **D6** : 1=发送数据寄存器空
- **D3** : 1=帧格式错误
- **D2** : 1=奇偶校验错误
- **D1** : 1=溢出错误
- **D0** : 1=接收数据寄存器满

## ● 向串行通信口发送字符

**功能号**      **AH = 01H**

**类型号**      **14H**

**入口参数**    **AL = 欲发送的字符**

**DX = 通信口号    0: COM1    1: COM2**

**实现功能**    **将字符通过指定的通信口发出**

## ● 从串行通信口接收字符

**功能号**      **AH = 02H**

**类型号**      **14H**

**入口参数**    **DX = 通信口号    0: COM1    1: COM2**

**返回参数**    **AL = 接收到的字符**

**实现功能**    **从指定的通信口读取字符**

### 3、查询方式发送、接收数据流程

- 在传送方式上，

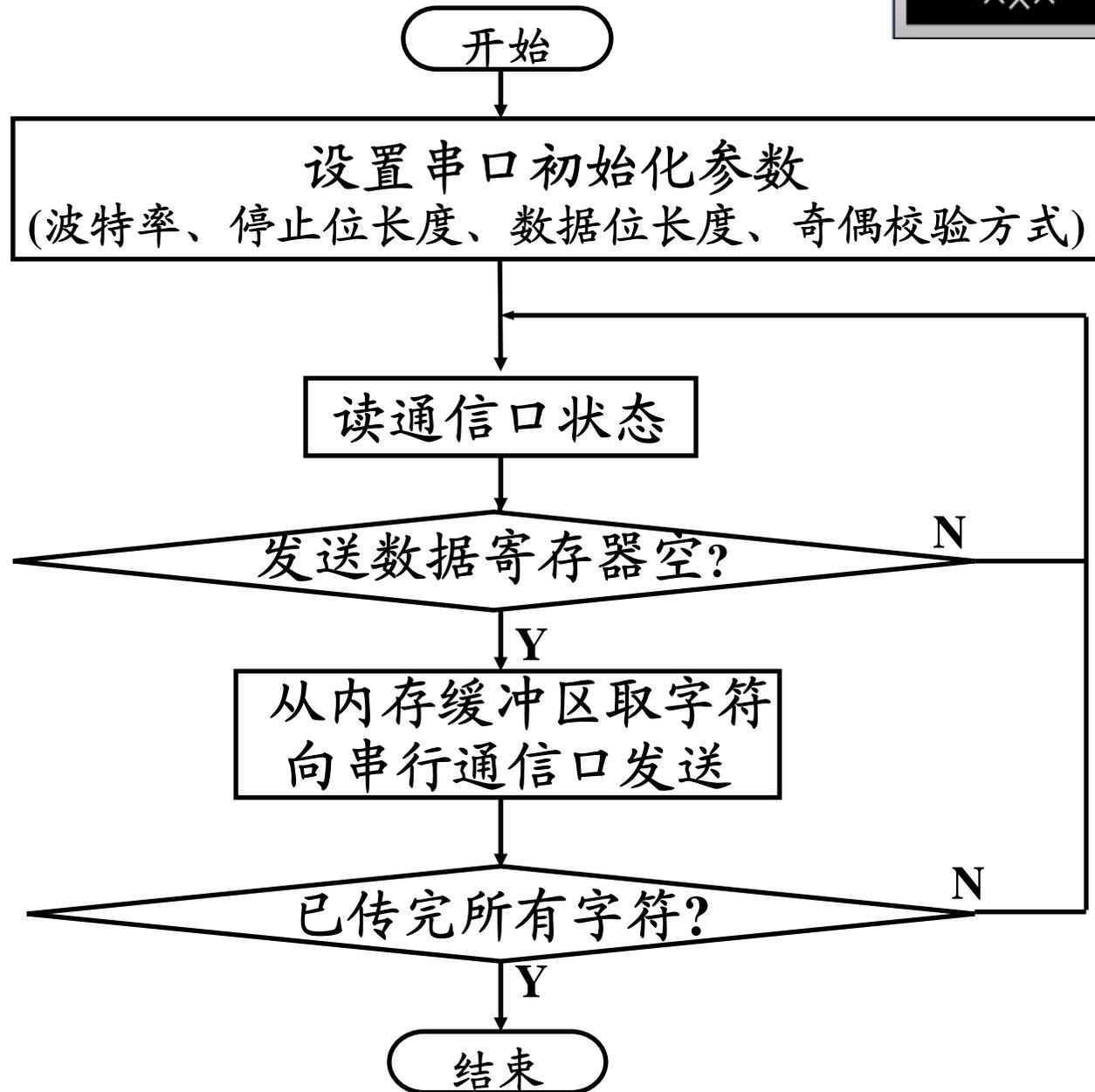
CPU与串口之间可以采用**查询方式**或**中断方式**进行通信

- 由于BIOS没有提供对有关串行接口内有关中断的信息的操作，

使用BIOS调用，只能用**查询方式**；

若用**中断方式**，需使用直接对**8250**芯片操作的方法

## 串口查询方式 发送数据流程



## 串口查询方式 接收数据流程

