

第13章 串行通信接口

主讲：许万茹





北京交通大学

BEIJING JIAOTONG UNIVERSITY



本章教学内容

1

串行通信的基本概念及协议

2

串行通信接口标准及接口电路

3

RS-232C标准的串行通信接口设计

4

RS-485标准的串行通信接口设计

5

基于UART的串行通信接口



13.1 串行通信的基本概念

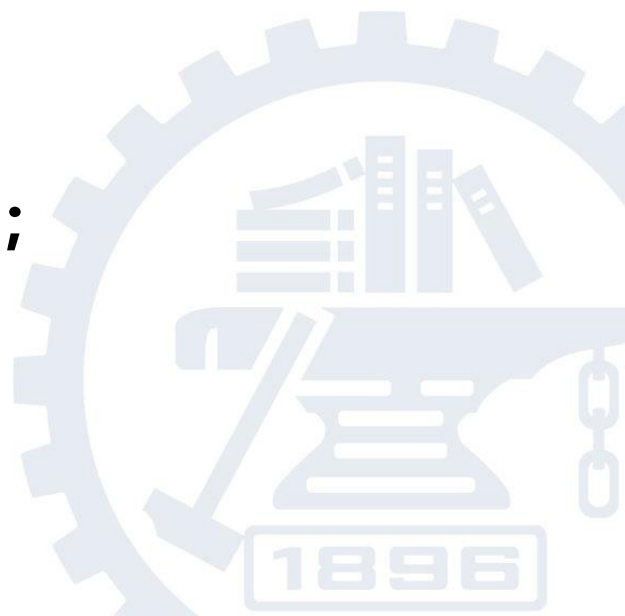
13.1.1 串行通信的基本特点

- ① 1根线上按位传输，该线既作数据线又作联络线
- ② 数据传输的格式有固定要求，分为异步和同步
- ③ 需要进行逻辑关系和逻辑电平的转换
- ④ 要求双方约定传输的波特率、控制传输速率
- ⑤ 串行通信易受干扰，需要进行差错检测与控制
- ⑥ 适用于远、近距离传送（需要外加MODEM）



串行通信接口电路需要解决的问题：

- ① 怎么才能知道数据传输的开始和结束；
- ② 怎么判断接收到的数据的正确性；
- ③ 如何进行传输速度控制；
- ④ 如何进行电平和逻辑的转换；
- ⑤ 如何进行通信双方的外部连接；
- ⑥ 如何进行串/并转换。

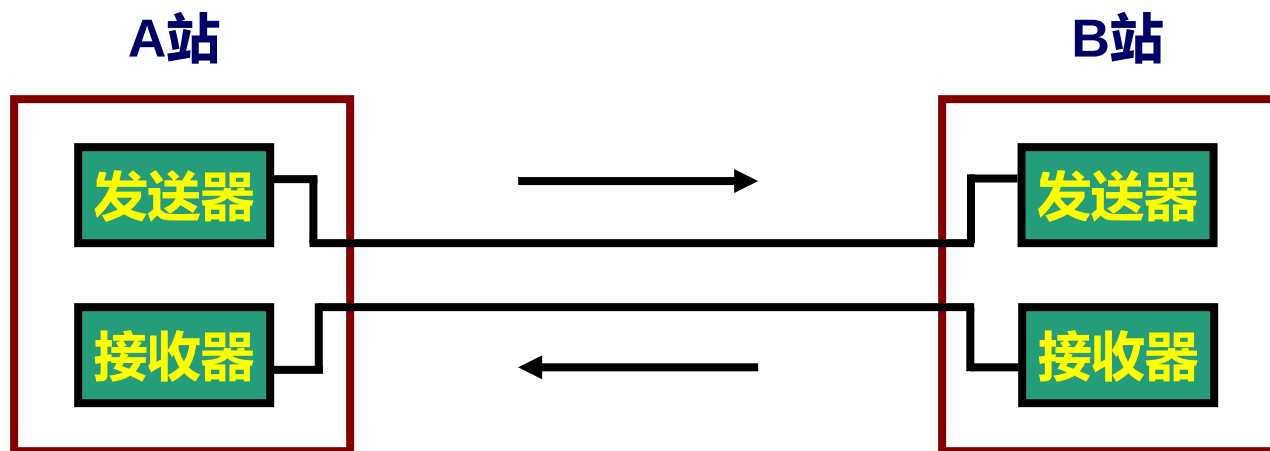




13.1.2 串行通信的工作方式（制式）

1. 全双工（Full Duplex）

全双工是通信双方**同时进行发送和接收操作**。数据的发送和接收分别由**两根**可以在两个不同的站点**同时**发送和接收的传输线进行传送，通信双方都能在同一时刻进行发送和接收操作，这种传送方式称为**全双工制**。全双工方式在通信过程中，**无须进行接收/发送方向的切换**，因此，**没有切换操作所产生的时间延迟**，有利于远程实时监测与控制。



全双工方式示意图

特点：① 每一端都有发送器和接收器

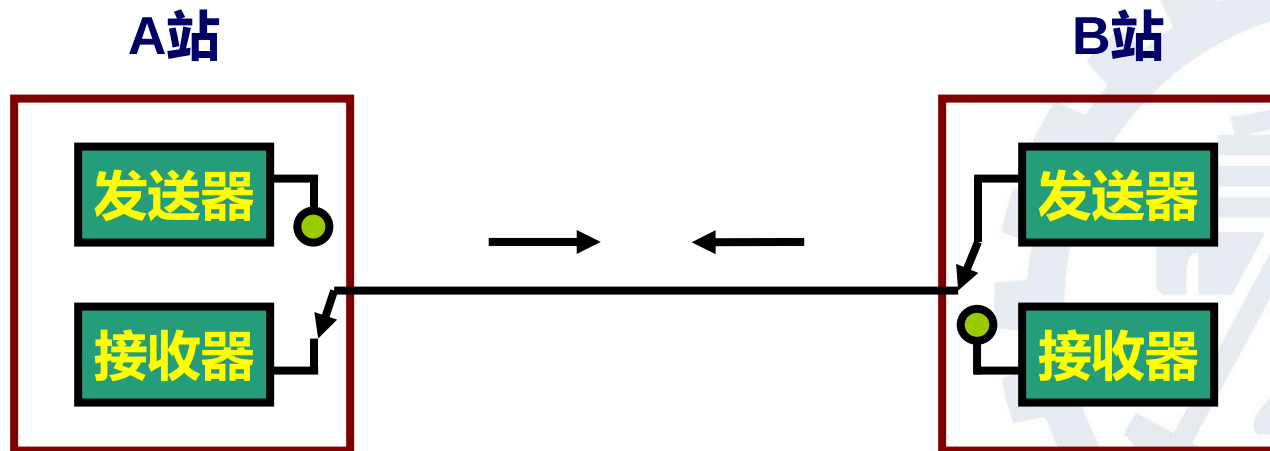
② 有二条传送线

应用：交互式应用、电话、远程监测控制



2. 半双工 (Half Duplex)

半双工只设置1根传输线，是通信双方分时进行发送和接收操作，即双方都可发可收，但不能在同一时刻发送和接收。在半双工通信过程中，需要进行接收/发送方向的切换，会有延时产生。可用于对讲机、单向传送设备等。



半双工方式示意图



3.单工 (Simplex)

使用一个单向传输介质，一方只能接收，另一方只能发送，如下图所示。



特点：发送方可以以广播方式传输数据到多个接收方

应用：电视台，广播站，卫星定位等。



13.1.3 串行通信中的差错检测

在串行数据传输过程中，由干扰引起误码是难免的，这直接影响通信系统的可靠性，对通信中的检/纠错能力是衡量一个通信系统的重要内容。

一、误码率控制

误码率：数据**传输后**发生错误的位数与总传输位数之比，一般要求在 10^{-6} 数量级

二、检纠错编码方法的使用

检错：发现传输中的错误，称为检错。

纠错：发现并消除传输中的错误，称为纠错。

如**奇偶校验检错**、**循环冗余码（CRC）检错**



三、错误状态的分析与处理

错误状态一般存放在接口电路的状态寄存器中供CPU分析和处理。常见的错误有：

- ① **奇偶校验错**：接收到的数据中1的个数与奇偶校验位不符合。接收方可要求发送方重发
- ② **溢出错**：接收方没来得及处理收到的数据，发送方已经送来下一个数据，造成数据丢失
- ③ **帧格式错**：接收到的数据与约定的格式不符
- ④ **查询方式的通信中有超时错**，一般由于接口硬件电路速度跟不上产生

四、错误检测只在**接收方**通过**软件方法**进行



奇校验：就是我想让这串数据是奇数个1。如果原数据是偶数个1，那么在最后一位补上1，这串数据就变成奇数个1。

偶校验也是同理，我想让这个数据是偶数个1，根据原数据1的个数加或加1。

位/数字	0	1	2	3	4	5	6	7	8	9	偶校验
C ₁	0	1	0	1	0	1	0	1	0	1	1
C ₂	0	0	1	1	0	0	1	1	0	0	0
C ₃	0	0	0	0	1	1	1	1	0	0	0
C ₄	0	0	0	0	0	0	0	0	1	1	0
C ₅	1	1	1	1	1	1	1	1	1	1	1
C ₆	1	1	1	1	1	1	1	1	1	1	1
C ₇	0	0	0	0	0	0	0	0	0	0	0



13.1.4 串行通信的同步方式

一、串行通信中的同步问题

接收方如何判断数据传输的开始？

同步问题包括字符同步和位同步：

- 1、字符同步：约定特定字符作为数据的开始
- 2、位同步：接收器通过时钟信号接收每一位数据



二、串行通信的基本方式

1、异步串行通信

以**字符**为单位传送。即：**传送同一字符的每一位时是同步的**，位与位之间有严格而精确的定时，而**字符与字符之间是异步的**，即发送端发出的每个字符在通信线上出现的时间是任意的。异步通信**不要求在收双方之间使用同一根时钟线**。

2、同步串行通信

以**数据块**（字符块）为单位传送。字符内**部位与位之间传送是同步的**，**字符与字符之间传送也是同步的**。收/发两端必须使用同一时钟来控制数据块内部位与位之间的定时，因此同步通信要求双方之间**必须设置一根时钟线**。



异步通信方式与同步通信方式的区别：

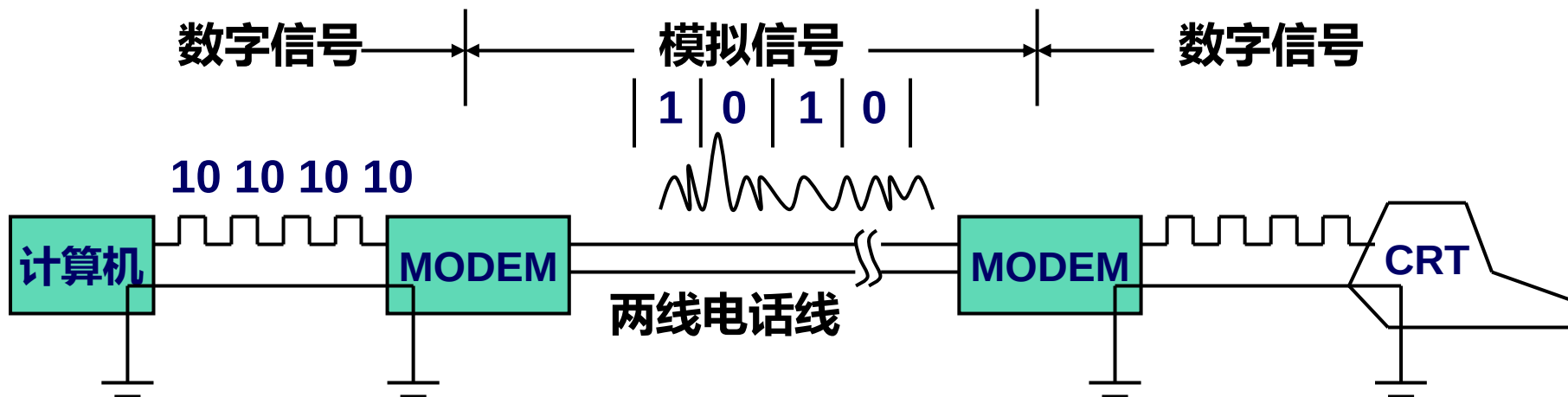
- 异步通信方式的传输速率低，同步通信方式的传输速率高。
- 异步传输的传输设备简单，易于实现，同步传输的传输设备复杂，技术要求高。
- 异步串行通信一般用在数据传输时间不能确定、发送数据不连续、数据量较少和数据传输速率较低の場合；而同步串行通信则用在要求快速、连续传输大批量数据の場合。



13.1.5 信号的调制和解调

串行通信是数字通信，包含了从低频到高频的谐波成分，因此**要求传输线的频带很宽**。在远距离通信时，为了降低成本，通信线路利用普通电话线，而这种电话线的频带宽度有限。如果用**数字信号直接传输**，**高次谐波的衰减就会很厉害**，从而使传输的信号产生严重的畸变和失真；若**用模拟信号传输时**，则失真较小。

因此，需要在**通信双方采用调制解调器(MODEM)**进行数字信号与模拟信号转换。



1. **发送采用调制器 (Modulator) 把数字信号转换为模拟信号，送到通信线路上。**
2. **接收器再通过解调器 (Demodulator) 把模拟信号转换为数字信号。**



13.2 串行通信中的传输速率控制

13.2.1 数据传输速率控制的实现方法

- 串行通信时，**要求双方的传输速率严格一致**，并在**传输开始之前**，**要预先设定**，否则，会发生错误。
- 在数字通信中，**传输速率的控制是通过波特率时钟发生器和设置波特率因子或波特率除数来实现的。**



13.3.3 波特率、发送/接收时钟

一、波特率

每秒钟传送的二进制位数。

用波特率表示：即1波特Baud=1b/s(位/秒，也可写为bps)

常用的标准波特率：110、150、300、600、1200等。

二、发送和接收时钟

① **移位脉冲**：执行数据的发送和接收

发送时，发送端在**发送时钟脉TxC**的作用下，将发送寄存器的数据按位串行移位输出，送到通信线上；

接收时，接收端在**接收时钟脉冲RxC**的作用下，对来自通信线上的串行数据，按位串行移入接收寄存器

② **进行位同步**



三、波特率因子

发/收时钟的频率一般是波特率的整数倍。这个倍数称为波特率因子或波特率系数Factor，发送/接收1位数据所需要的时钟脉冲个数，单位是个/位。

发/收时钟脉冲频率与波特率之间的关系是：

$$T_x C = \text{Baud} \times \text{Factor}$$

例如：要求传送速率为1200 Baud，则

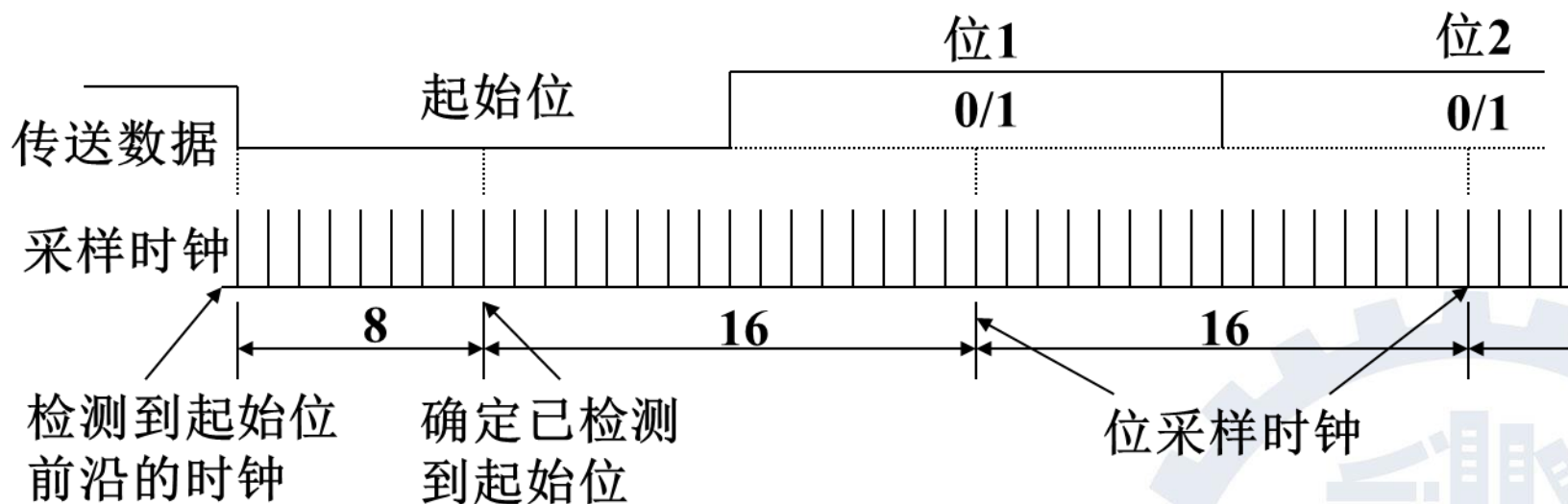
当Factor=1时，发/收时钟=1200*1=1.2kHz；

当Factor=16时，发/收时钟=1200*16=19.2kHz；

当Factor=64时，发/收时钟=1200*64=76.8kHz；



在异步通信中，为提高抗干扰性，接收器采用比波特率更高频率的时钟来提高定位采样的分辨率。



16倍波特率时钟的作用



四、波特率时钟发生器

不包含波特率时钟发生器的串行接口芯片需要自行设计
波特率时钟发生器

五、波特率时钟的使用

例1：甲乙两机进行串行通信。Factor=16个/位。第一次采用 $T_xC1=38400\text{Hz}$ 通信，第二次采用 $T_xC2 = 19200\text{Hz}$ 通信。这两次通信的波特率各为多少？

$$\text{Baud1} = T_xC1 \div \text{Factor} = 38400 \div 16 = 2400 \text{ b/s}$$

$$\text{Baud2} = T_xC2 \div \text{Factor} = 19200 \div 16 = 1200 \text{ b/s}$$



北京交通大学

BEIJING JIAOTONG UNIVERSITY



例2： 甲乙两机进行串行通信。甲机的 $T_xC=38400\text{Hz}$ ， $\text{Factor1}=16$ 个/位。乙机的 $\text{Factor2}=64$ 个/位。双方的波特率要保持一致，乙机的 R_xC 应为多少？

$$\text{Baud} = T_xC \div \text{Factor1} = 38400 \div 16 = 2400 \text{ b/s}$$

$$R_xC = \text{Baud} \times \text{Factor2} = 2400 \times 64 = 153600 \text{ Hz}$$

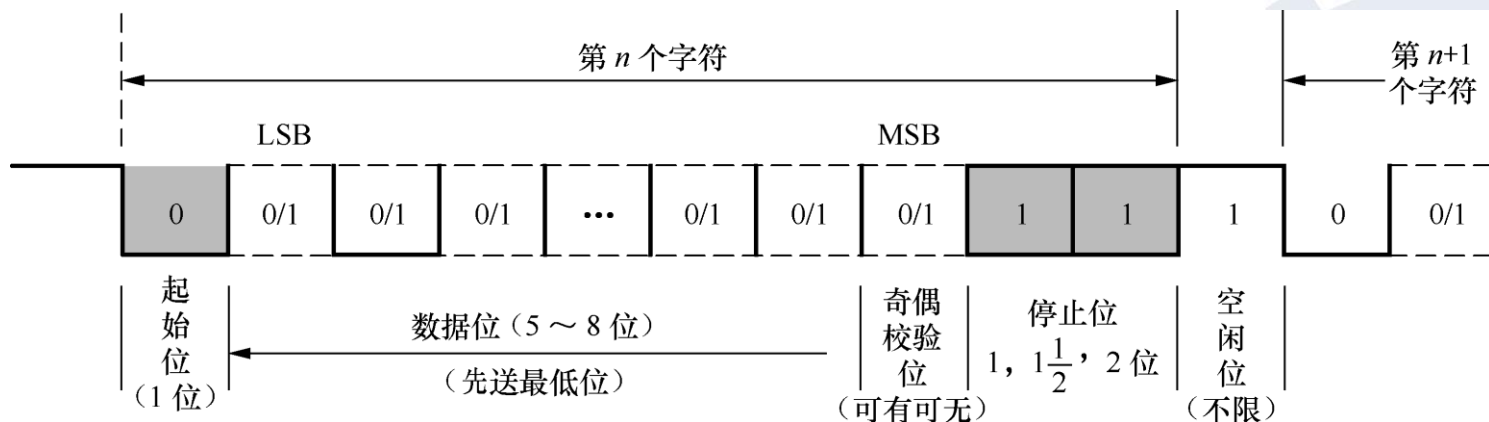




13.3 串行通信中的数据格式

13.3.1 起止式异步通信数据格式

异步通信是以字符为单位进行传输的。**起止式**是在**每个字符**的**前面加起始位**，**后面加停止位**，**中间可以加奇偶校验位**，**空闲位**，形成一个完整的字符帧格式。



起止式异步通信数据格式 (汽车)



每个字符帧由4个部分组成：

- **1位起始位**，**低电平**，逻辑值0。
- **5~8位数据位**，紧跟在起始位后，是要传输的有效信息。规定从低位至高位依次传输。
- **1位校验位**（也可以没有校验位）。
- **1位，或1位半，或2位停止位**，**高电平**（逻辑值1）。

停止位后面是不定长度的**空闲位**。

停止位和空闲位都规定为高电平，这样就保证起始位开始处一定有一个下跳变。

起/止位的作用：起始位和停止位是作为**联络信号**而附加进来的，当起始位由**高电平**变为**低电平**的**下跳变**时，告诉接收方数据的传输开始。它的到来，表示下面接着是数据位来了，要接收方准备开始接收。而**停止位**标志一个字符的结束，它的出现，表示一个字符传输结束。这样，就为通信双方提供了**何时开始收发、何时结束的标志**。



13.3.2 面向字符的同步通信格式

1. 功能:

同步通信一次传送若干个字符组成的数据块。并规定了10个特殊字符作为这个数据块的开头标志、结束标志以及整个传输过程的控制信息。

2. 数据格式（一帧）

源地址、目的地址、路由指示



同步字符
表示一帧
的开始

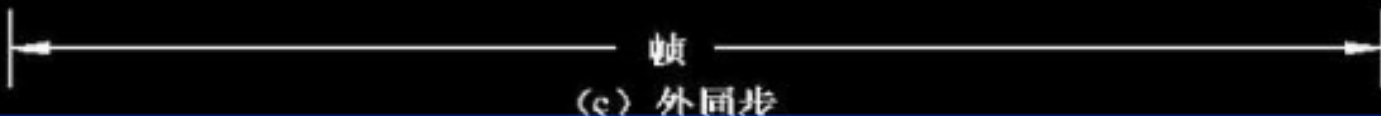
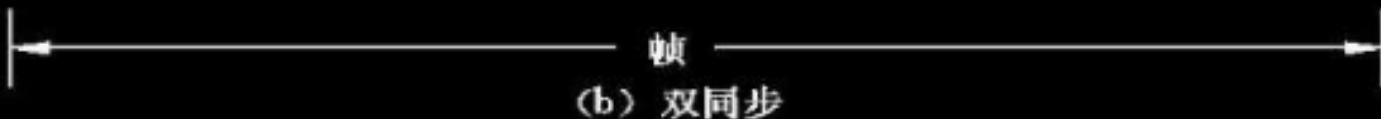
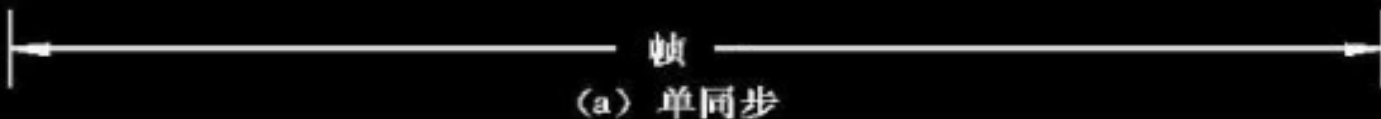
序始字符
表示标题
的开始

文始字符
表示正文
开始

组终/文
终字符



可采用单同步、双同步及外同步3种数据格式。



(火车)



单同步——指在传送数据之前先传送一个同步字符“**SYNC**”，

双同步——先传送两个同步字符“**SYNC**”。接收端检测到该同步字符后开始接收数据。

外同步——通信的数据格式中没有同步字符，而是用一条专用控制线来传送同步字符，使接收方及发送端实现同步。当每一帧信息结束时均用两个字节的循环控制码**CRC**为结束。



13.4 串行通信接口标准

13.4.1 RS-232C接口标准

- RS-232C标准定义数据终端设备DTE和数据通信设备DCE之间的连接

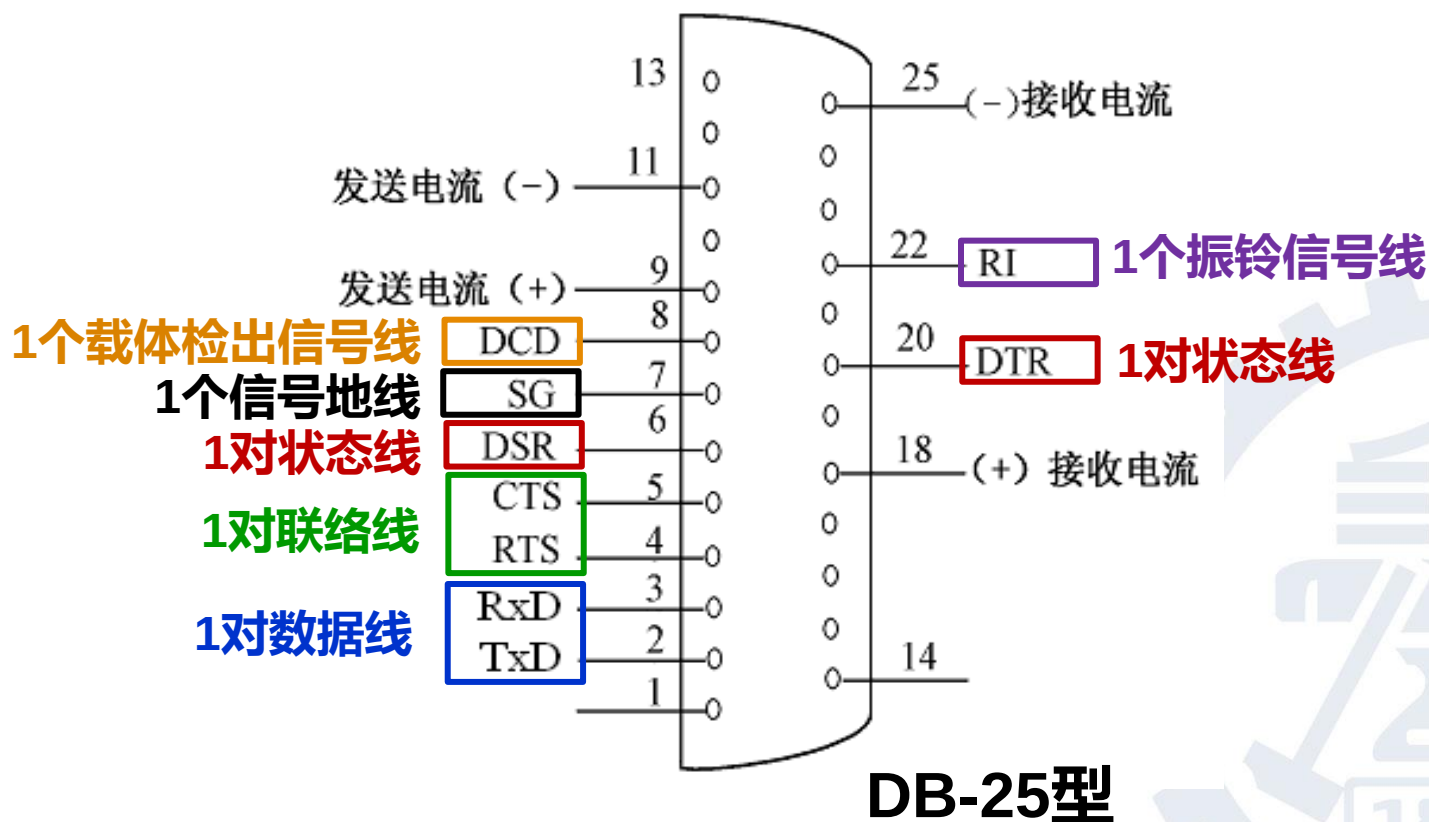


- RS-232C标准不是为计算机通信设计的，所以会出现与计算机不兼容的情况



1、RS-232C接口信号线的定义（功能特性）

共25线，分主信道和辅信道信号线，其中主信道9根线：



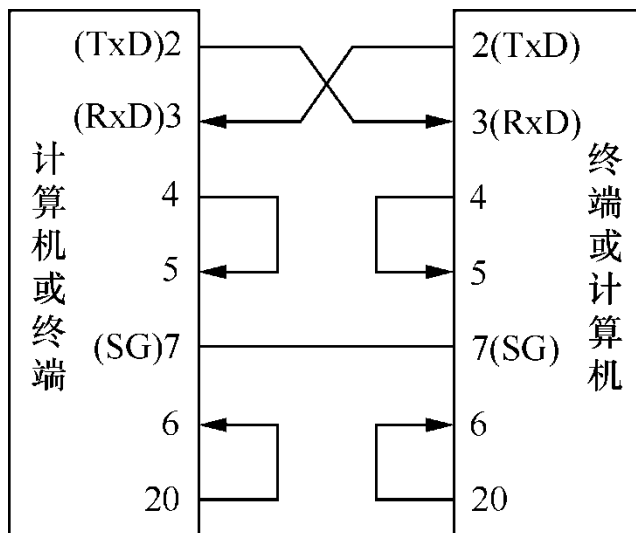


- ▲ TxD: 串行数据发送端, 输出。
- ▲ RxD: 串行数据接收端, 输入。
- ▲ DTR: 终端准备就绪信号
- ▲ DSR: 数据装置准备就绪信号
- ▲ RTS: 请求发送信号, 为高时表示终端要向Modem或其它通信设备发送数据。
- ▲ CTS: (允许) 清除发送信号, 是对RTS的响应。 当Modem或外设准备好接收来自终端的数据时, 该信号为高电平。
- ▲ DCD: 接收信号检测, 为高时表明Modem收到通信线路另一端Modem送来的正常载波信号。
- ▲ RI: 振铃提示, 当高时表明Modem收到交换台送来的振铃信号。



2、信号线的使用

- ① 使用MODEM，通过电话线路远距离通信
- ② 使用MODEM，通过专用线远距离通信
- ③ 近距离通信 (<15m，零MODEM方式，常用)



零MODEM方式的连接





3、电气特性

① RS-232C标准对信号的逻辑定义 (EIA逻辑)

逻辑 “1”

= -5V ~ -15V (驱动器输出端)

逻辑 “0”

= +5V ~ +15V (驱动器输出端)

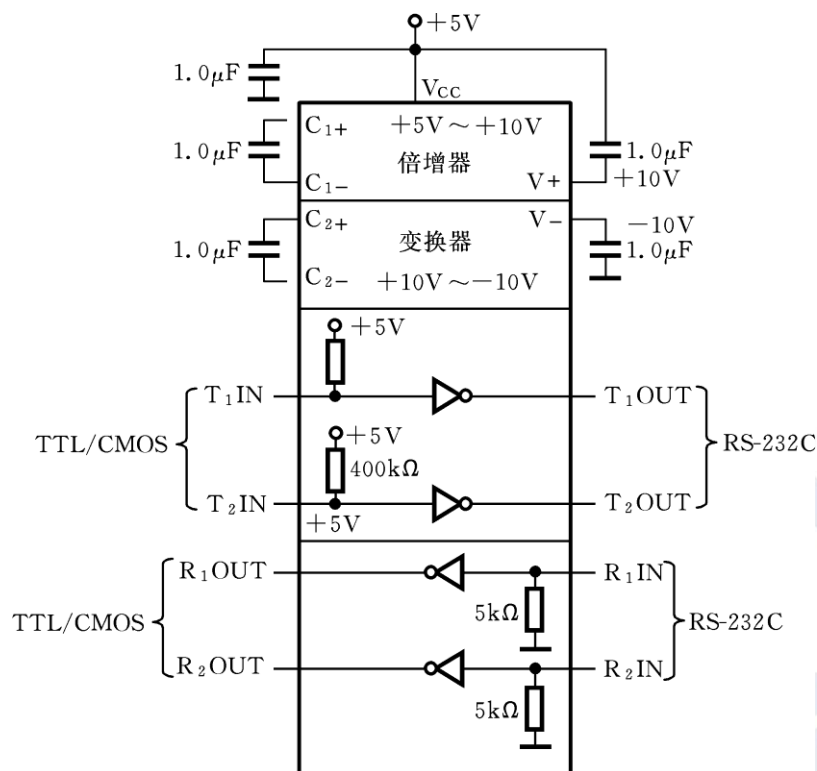
与TTL逻辑电平(“1”为+5V, “0”为0V)不一样

RS-232C采用的是负逻辑，并且逻辑电平幅值很高，摆幅很大。显然，RS-232C标准所采用的EIA与计算机终端所采用的TTL在逻辑电平和逻辑关系上并不兼容，故需要经过转换，才能与计算机或终端进行数据交换。



② EIA与TTL之间的转换

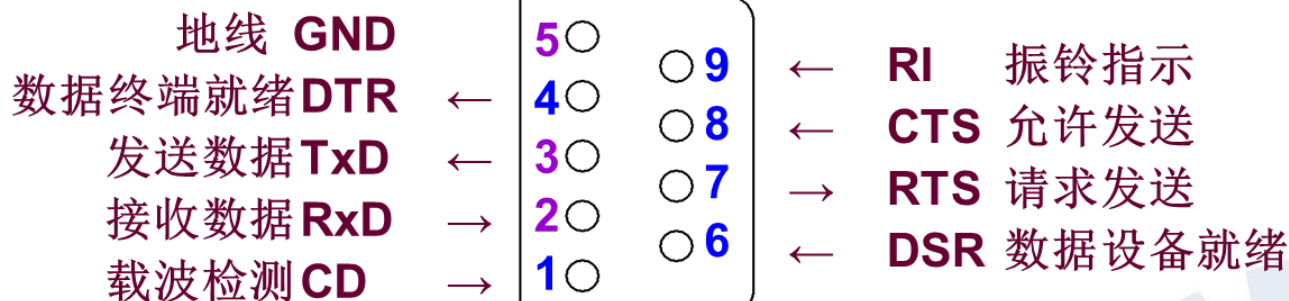
- 采用EIA电平比TTL电平具有更强的抗干扰性能，可用MAX232完成EIA与TTL间的电平转换，一个MAX232芯片可实现两对接收/发送数据线的转换。
- 通过MAX232可将TTL/CMOS电平（0 ~ 5V）转换成RS-232C的EIA电平（+10V ~ -10V）。





4、机械特性

① 连接器 (Connector)



DB-9型

② 通信电缆长度

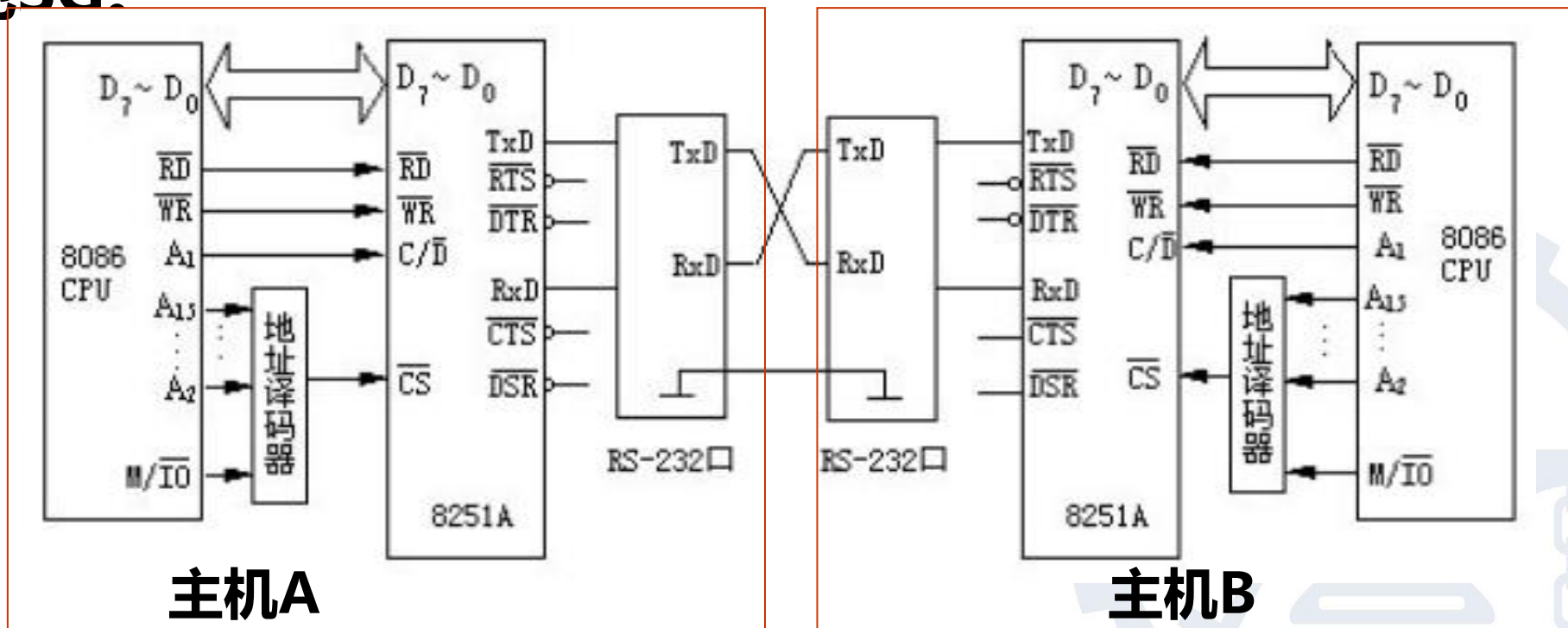
RS-232C标准规定当采用**双绞线屏蔽电缆**，传输速率为**20kb/s**时，在**零MODEM**下，两台计算机或终端的直接连接的最大物理距离为**15m**。



5、信号线的连接

① 近距离连接 (<15m, 零MODEM方式)

RS232口只连3条线：发送线 T_xD ，接收线 R_xD ，信号地SG。

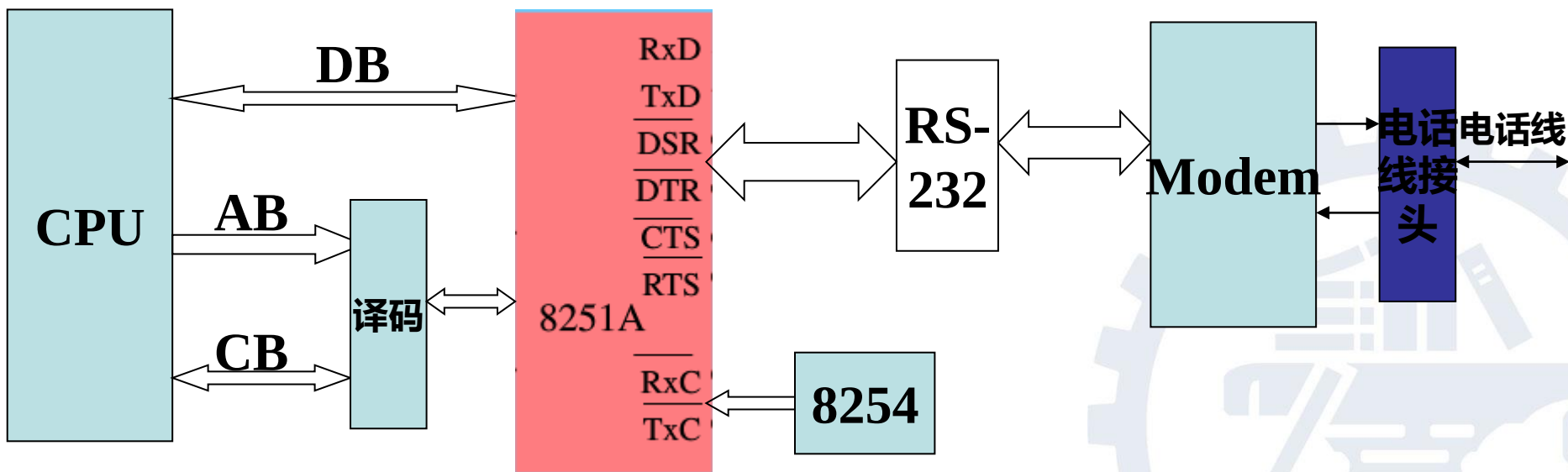




② 远距离连接 (>15m)

(1) 需用MODEM和专用电话线

(2) 2~9条信号线 (RS232在接口电路与MODEM之间, 除去前3根线之外, 还要用/DSR,/DTR等线, 供调制解调用)





8.3.2 RS-485标准

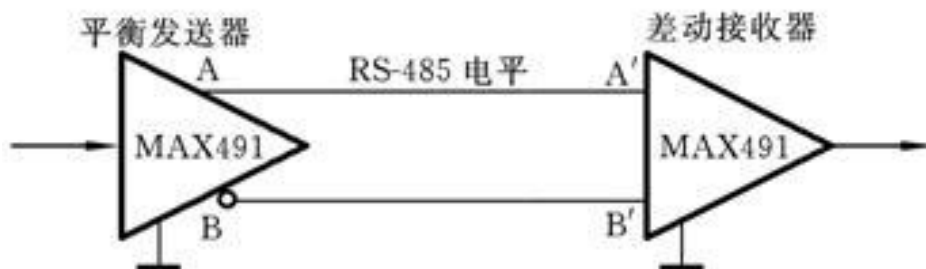
RS-485标准是在**RS-232C**的基础上，针对在**不使用MODEM**情况下，进行远距离串行通信而提出的，因此**只对RS-485标准的数据信号线和信号逻辑重新进行了定义**，不涉及其他控制信号线的定义，而通信过程中所传输的**数据帧格式沿用RS-232C的**。（**非微机标配**）



14.3.2 RS-485接口标准

1、特点:

(1) 采用平衡发送器和差动接收器，由于是双线传输，大大提高了抗共模干扰的能力。



两条传输线的电位差决定逻辑电平:

$AA' - BB' < -200\text{mV}$, 表示 “0”

$AA' - BB' > +200\text{mV}$, 表示 “1”

(2) 传输速率 10Mbps (<15m时)

100Kbps (<1200m时)

(3) 4芯水晶头，价格低廉，使用方便

(4) 通信距离：几百米，适用于考勤机、POS机等中短距离的通信，**非微机标配。**

- RS485在较远距离传输上，比无交换机的网线方式还是有很大优势的。如商业POS收银机、考勤机、智能大楼的联网、工业集散分布式系统等，
- 在通信终端中无需运行TCP/IP网络协议栈和OS，比起局域网方式，采用RS485连线方式的软件成本是很低的，相应所对应的硬件成本也随着下降。



13.5 串行通信接口电路

13.5.1 串口通信接口的基本任务

1. 实现串行数据格式化

因为来自计算机的数据是普通并行数据，所以**接口电路应具有使数据格式化的功能，实现不同串行通信方式下的数据格式化**。例如，对RS-232C标准，接口电路需自动生成异步通信的起止式数据帧格式。

2. 进行串/并转换

在**发送端**，把由计算机送来的**并行数据转换为串行数据**，然后再发送出去。在**接收端**，把从接收器接收的**串行数据转换为并行数据**后，再送至计算机或终端。



3. 进行错误检测

发送端，接口电路需对传输的字符数据**自动生成奇偶校验位或其他校验码**。**接收端**，接口电路需要对所接收的字符数据**进行奇偶校验或其他校验码的检测**，以确定所接收的数据中是否有错误信息和是什么性质的错误。

4. 提供符合RS-232标准定义的信号线

远距离通信使用MODEM时，接口电路需要**9根信号线**。对于**近距离零MODEM方式**，接口电路只需要**3根信号线**。

5. 进行TTL与EIA逻辑关系及逻辑电平的转换

计算机和终端均采用正逻辑和TTL电平。它们与EIA采用的负逻辑及高压电平**不兼容**，需在接口电路中**采用换门芯片（如MAX232）进行转换**。

6. 进行数据传输速率的控制

串行通要求双方的数据传输速率一致，这意味着接口电路**需设置波特率时钟发生器对数据传输速率进行控制**。



13.5.2 串口通信接口电路的组成

一般由**串行接口芯片**、**波特率时钟发生器**、**EIA与TTL转换器**和**地址译码电路**组成。

- **8251A**：支持RS-232C标准，不支持RS-485标准，是系统扩展串行通信接口电路，无BIOS/DOS支持，需设计硬件接口电路才能实现通信
- **8250/16550**：支持RS-232C标准，不支持RS-485标准，微机系统标配，是系统串行通信接口电路，BIOS/DOS提供支持，无需设计硬件电路
- 上述接口如需采用RS485通信标准，需设计RS232与RS485的转换电路



13.6 8251A USART

13.6.1 8251A的外部特性

面向 CPU	面向 RS232	状态及时钟
$D_{0..7}$	TxD	TxRDy
$\overline{WR}$	RxD	RxRDy
$\overline{RD}$	$\overline{RTS}$	SyNDET
RESET	$\overline{CTS}$	$\overline{TxC}$
$\overline{CS}$	$\overline{DTR}$	$\overline{RxC}$
$C/\overline{D}$	$\overline{DSR}$	TxEMPTY

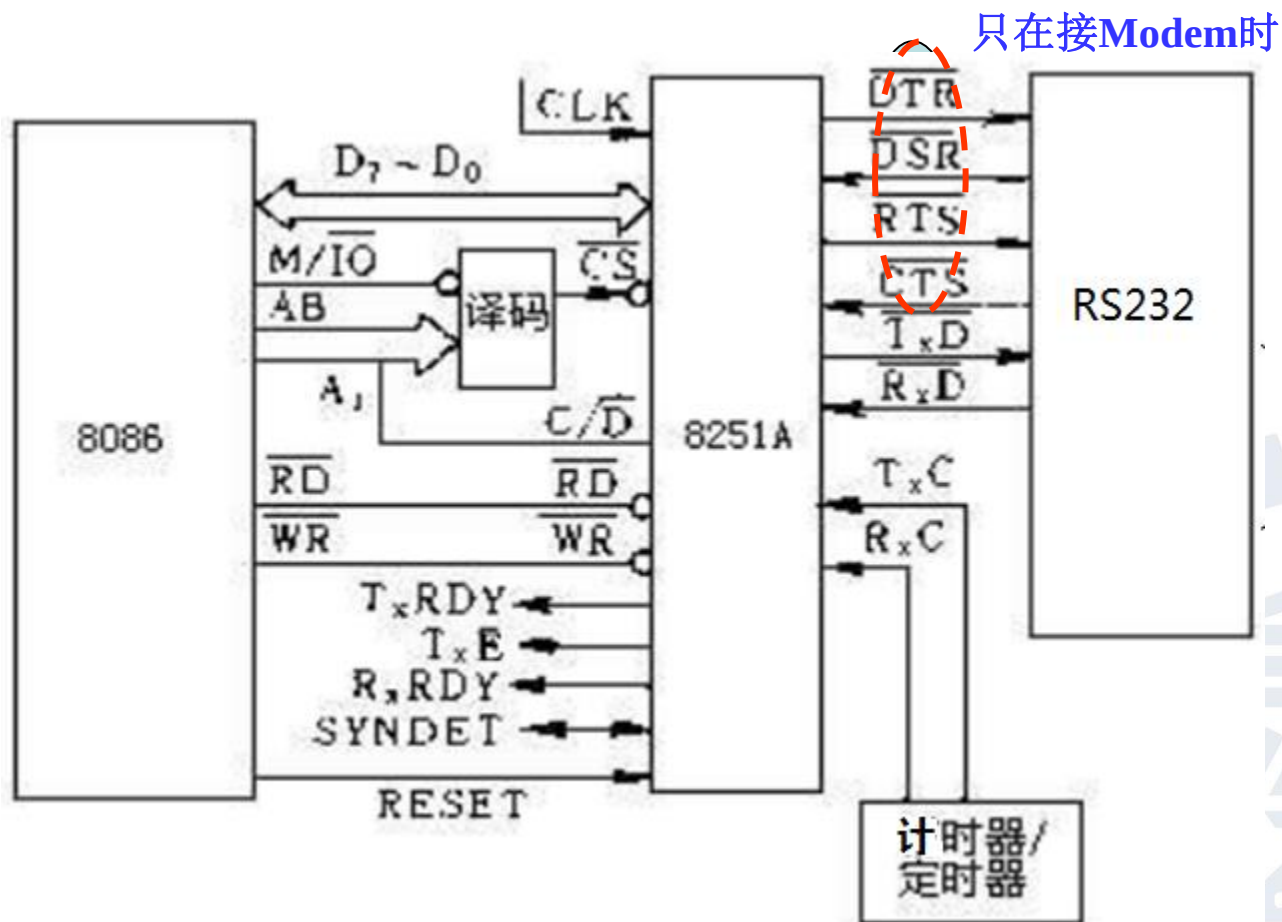
Intel 8251A是一款可编程的通用同步/异步收发器 (USART: Universal Synchronous Asynchronous Receiver and Transmitter), 是一个28引脚的双列直插芯片。

功能:
异步起止协议
同步面向字符协议

1	D_2	D_0	28
2	D_3	D_1	27
3	RxD	V_{CC}	26
4	GND	$\overline{RxC}$	25
5	D_4	$\overline{DTR}$	24
6	D_5	$\overline{RTS}$	23
7	D_6	$\overline{DSR}$	22
8	D_7	RESET	21
9	$\overline{TxC}$	CLK	20
10	$\overline{WR}$	TxD	19
11	$\overline{CS}$	TxEMPTY	18
12	$C/\overline{D}$	$\overline{CTS}$	17
13	$\overline{RD}$	SYNDET/BD	16
14	RxRDY	TxRDY	15

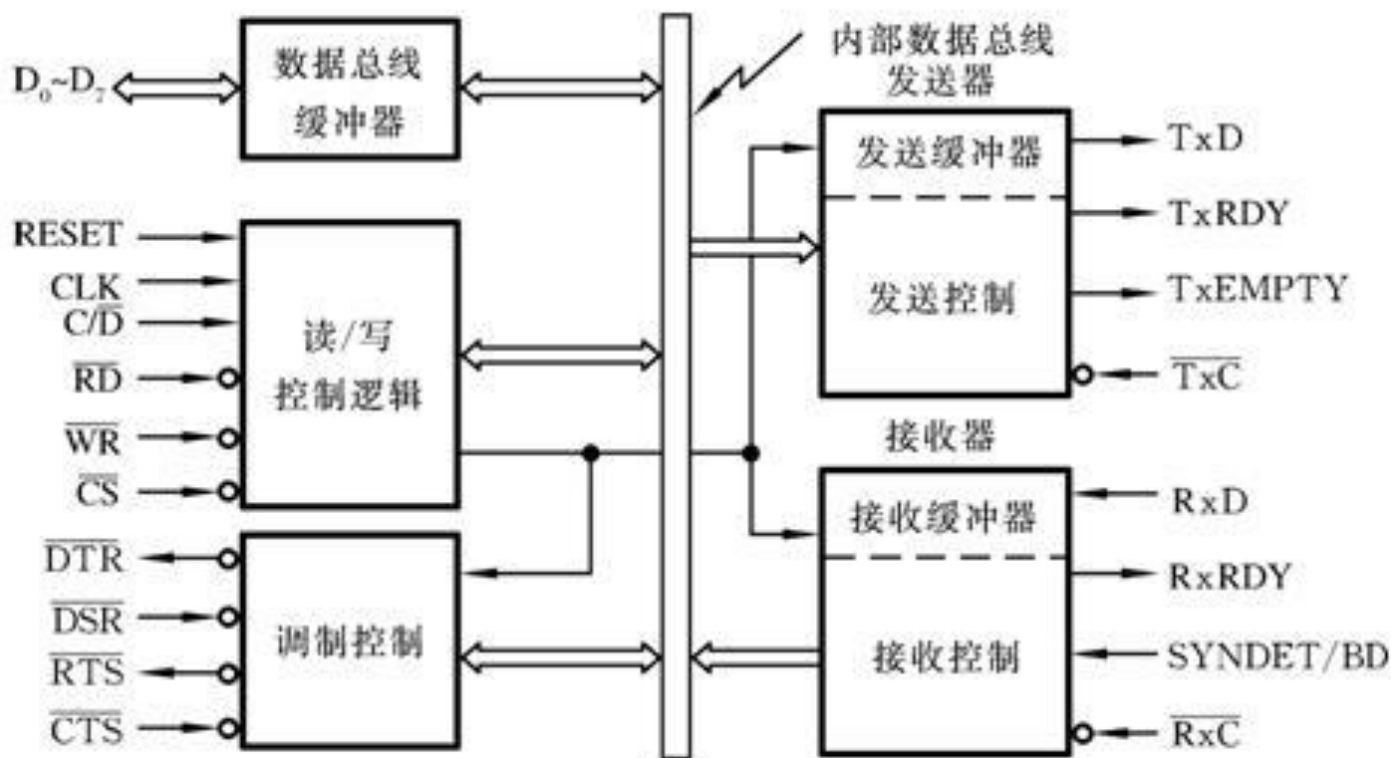


8251的面向CPU的管脚





13.5.2 8251A内部寄存器及编程命令



组成：接收器、发送器、调制控制、读/写控制、数据总线缓冲器。



发 送 器

首先把待发送的并行数据转换成为所需要的帧格式并加上校验位，然后在发送时钟 $\overline{T_xC}$ 的控制下，由引脚 T_xD 逐位串行发送出去。

- ① T_xRDY (Transmitter Ready) : **发送器准备好**，状态线，高电平有效，CPU从状态寄存器中读取该状态，确认有效后，即可向8251A写入待发送数据；
- ② T_xEMPTY (Transmitter Empty) : **发送器空**，状态线，高电平有效，表明8251A已经成功发送出了一个数据，CPU可从状态寄存器中读取该状态；
- ③ T_xD : **发送数据线**，输出串行数据；
- ④ $\overline{T_xC}$: **发送器输入时钟**，外部输入；



接 收 器

在时钟 $\overline{R_xC}$ 作用下，接收 R_xD 引脚上的帧格式化串行数据，并把它转换成并行数据。同时进行校验，若发现错误，则在状态寄存器中保存错误状态，以便CPU查询和处理。

- ① R_xD : 接收数据线;
- ② R_xRDY : 接收器准备好, CPU可从状态寄存器中读取该状态;
- ③ $SYNDET$: 同步字符检出信号, 表示已收到对方发过来的同步字符, 用于同步串行通信。当 $SYNDET$ 有效时, 接收器可开始利用接收时钟采样 RxD , 串行接收数据。CPU可从状态寄存器中读取该状态;
; $SYNDET$ 异步方式下不用。
- ④ $\overline{R_xC}$: 接收器输入时钟。



数据总线缓冲器

三态双向，它含有数据缓冲器和命令缓冲器。是CPU与8251A交换数据、发布命令和读取状态的通道。

读/写控制逻辑

$\overline{CS}$	$C/\overline{D}$	$\overline{RD}$	$\overline{WR}$	功 能
0	0	0	1	CPU从8251A读数据
0	1	0	1	CUP从8251A读状态
0	0	1	0	CUP写数据到8251A
0	1	1	0	CPU写命令到8251A
1	X	X	X	总线浮空

$C/\overline{D}$ ：控制/数据信号输入端。 $C/\overline{D}=1$ ，数据总线传输的是控制信息和状态字； $C/\overline{D}=0$ ，传送的是数据信息。



调制解调器控制 (与MODEM的接口信号)

DTR: 终端准备就绪信号, 输出。为高时表明终端准备好接收来自Modem或外设的数据。

DSR: 数据装置准备就绪信号, 输入。为高时表明Modem或外设准备好发送数据。

RTS: 请求发送信号, 输出。为高时表示终端要向Modem或其它通信设备发送数据。

CTS: 清除发送信号, 输入。是对RTS的响应。当Modem或外设准备好接收来自终端的数据时, 该信号为高电平。

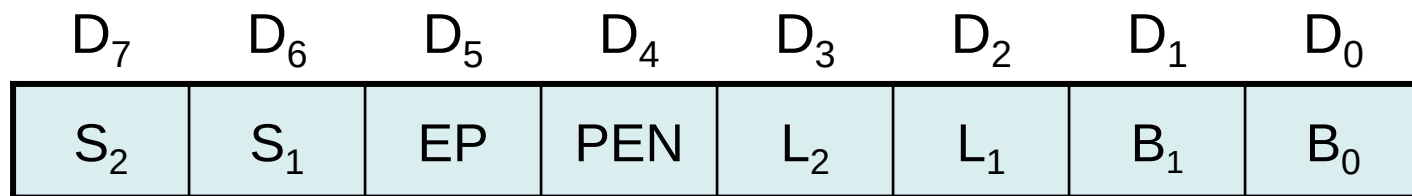


8251A内部有2个命令字和1个状态字：

- **方式命令**：指定8251A的**通信方式** (同步、异步)及其约定的数据帧格式。**初始化时使用**。
- **工作命令**：① 指定8251A**进行何种操作**(发送、接收、内部复位、检测同步字符等)；② **设置某种工作状态**，以便接收和发送数据。
- **状态字**：**报告**8251A**何时**可发送和接收数据，以及接收的数据**有无错误**。



1、方式命令：8251A复位后写到命令端口的第一个字节



异步方式：停止位

00: 无效
01: 1个停止位
10: 1.5个停止位
11: 2个停止位

同步方式：SYNC/内/外

00: 2个SYNC, 内同步
01: 2个SYNC, 外同步
10: 1个SYNC, 内同步
11: 1个SYNC, 外同步

1: 奇偶校验
0: 不用

1: 偶校验
0: 奇校验

波特率因子

00: 同步方式
01: 异步 (×1)
10: 异步 (×16)
11: 异步 (×64)

字符长度
00: 长度5位
01: 6位
10: 7位
11: 8位



北京交通大学

BEIJING JIAOTONG UNIVERSITY



例1：在**异步**通信中，采用**8位数据位**，**1位起始位**，**2位停止位**，**奇校验**，**波特率因数是16**，则方式字是：

11011110B=0DEH

MOV DX, 309H ; 8251A命令口

MOV AL, 0DEH ; 异步工作方式字

OUT DX, AL

例2：**同步**通信中，若帧数据格式为：**字符长度8位**，**双同步字符**，**内同步方式**，**奇校验**，则方式字是：

00011100B=1CH

MOV DX, 309H ; 8251A命令口

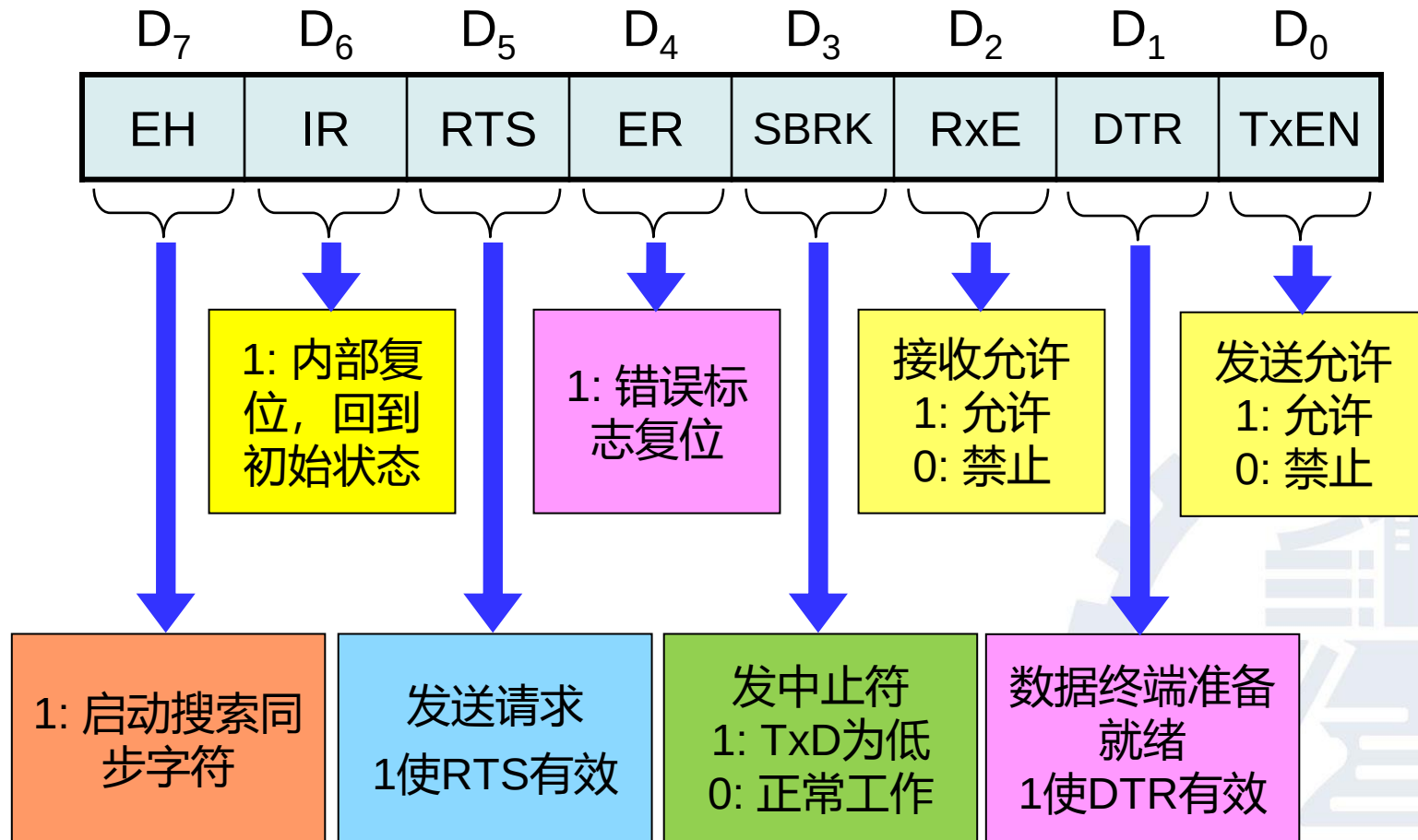
MOV AL, 1CH ; 同步工作方式字

OUT DX, AL

1896



2、工作命令：8251A复位后写到命令端口的第二个字节





EH位：让8251搜索同步字符，只在同步方式下用

IR位：当工作命令字中的IR=1，则进行内部复位，使得8251可以重新设置方式命令字；**如果无复位操作(包括reset复位)，则第一个方式命令字之后所有的命令字都是工作字！**

ER位置1时，状态寄存器中的3个错误标识位被清零；

SBRK位：发送断缺字符位。设为1时，TxD线一直发送“0”信号。正常通信过程中该位应当保持为“0”。

RTS位：请求发送位。CPU将其设为1时，8251A的RTS管脚有效，向Modem表示CPU已经做好发送数据的准备。

DTR位：数据接收端准备好位。CPU将其设为1时，8251A的DTR管脚有效，向Modem表示CPU已经做好接收数据的准备。



例3：使8251A内部复位，并且允许接收，又允许发送
则程序段为：

```
MOV DX, 309H      ; 8251命令口
MOV AL, 01000000B ; 置D6=1, 使内部复位
OUT DX, AL

.....

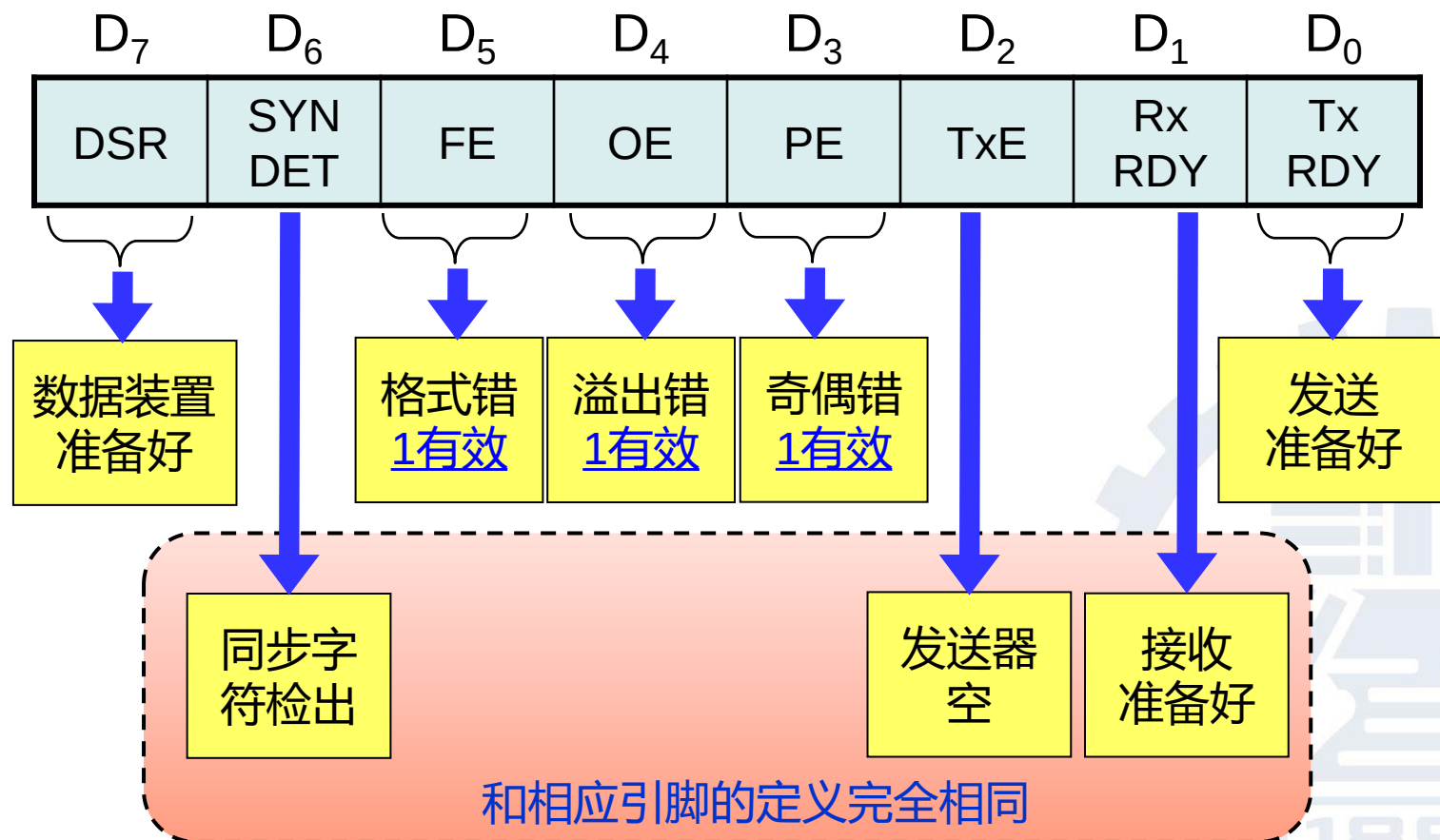
..... ;重新设置方式命令字;

.....

MOV AL, 00000101B ; 置D2=1, D0=1
                  ; 允许接收和发送
OUT DX, AL
```



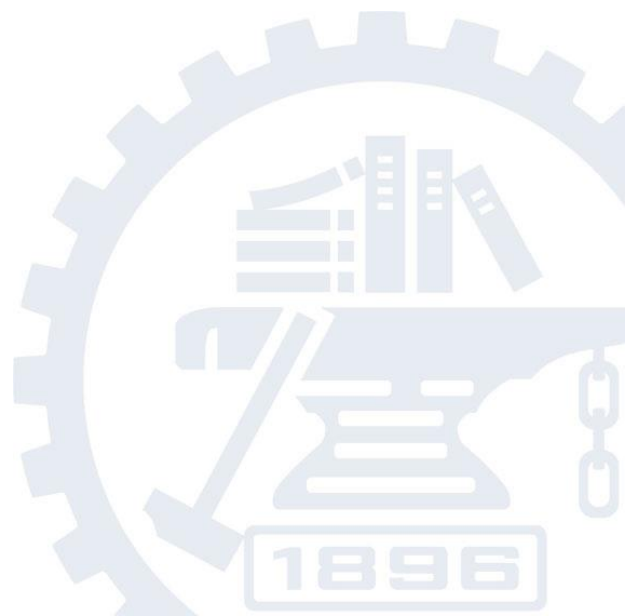
3、状态字：8251A复位后从状态端口读到的状态字。CPU通过读出状态字进行分析和判断以决定下一步的操作。





· 只有 $TxRDY=1$ 才能发送字符， $RxRDY=1$ 才能接收字符；

· 三个错误状态位：在编写接收程序时，要先对其进行检测，如果没有错误，再检查 $RxRDY$ 是否准备好。如果有错误，则先处理错误。





例4：若查询8251A**接收器是否准备好**，即查 $R_xRDY=1$?

则用下列程序段：

```
L: MOV DX, 309H    ; 状态口
    IN  AL, DX      ; 读状态字
    AND AL, 02H     ; 查 $D_1=1$ ? ( $R_xRDY=1$ ?)
    JZ  L           ; 未准备好，则等待
```

例5：检查**出错信息**，则用下列程序段

```
MOV DX, 309H ; 状态口
IN  AL, DX
TEST AL, 38H ; 检查 $D_5D_4D_3$ 三位 (FE, OE, PE)
JNZ ERROR    ; 若其中有一位为1，则出错
```



方式命令

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
S ₁	S ₀	EP	PEN	L ₁	L ₀	B ₁	B ₀
停止位		奇偶校验		字符长度		波特率因子	

(同步)	(异步)					(异步)	(同步)
×0=内同步	00=不用	×0=无校验		00=5位		00=不用	00=同步
×1=外同步	01=1位	01=奇校验		01=6位		01=×1	—
0×=双同步	10=1.5位	11=偶校验		10=7位		10=×16	—
1×=单同步	11=2位			11=8位		11=×64	—

工作命令

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
EH	IR	RTS	ER	SBRK	RxEN	DTR	TxEN
进入搜索方式	内部复位	发送请求	错误标志复位	发中止符	接收允许	数据终端准备好	发送允许

1=内部复位 0=不复位	1=错误标志复位 0=不复位	1=允许收 0=不允许收	1=允许发 0=不允许发
-----------------	-------------------	-----------------	-----------------

状态字

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
DSR	SYNDET	FE	OE	PE	TxE	RxRDY	TxRDY
数传机就绪	同步字符检出	格式错	溢出错	奇偶错	发送器空	接收准备好	发送准备好

1=格式错 0=无错	1=溢出错 0=无错	1=奇偶错 0=无错	1=接收就绪 0=未就绪	1=发送就绪 0=未就绪
---------------	---------------	---------------	-----------------	-----------------



4、8251A的方式字和命令字的使用

● 方式字、命令字和状态字间的关系

方式字：约定双方通信的方式、数据格式、传送速率等参数。

命令字：控制是发送数据，还是接收数据。

状态字：何时发/收、取决于状态字。

● 使用的顺序：

因方式字和命令字无特征位，且送到同一个端口，所以要**按一定的顺序写入**。

复位→方式字→命令字1→命令字2



- 就工作字和方式字两种字
- 约定：复位之后的第一个字是方式字！

若硬件reset不能确定是否被操作过，那么做法如下：

第1个字：可能是方式字、可能是工作字，不管怎样，随便写一个 $D6 \neq 1$ 的字（确保本操作不可能是一个复位工作字即可）

第2个字：如果第1个字不是复位工作字的话，那么作为第2个字，肯定不是方式字，必定是工作字！那么，让本字的 $D6=1$ ，则为复位工作字

第3个字：为复位后的第一个字，必定是方式字

第4个字：为方式字之后的工作字，可以开始工作了



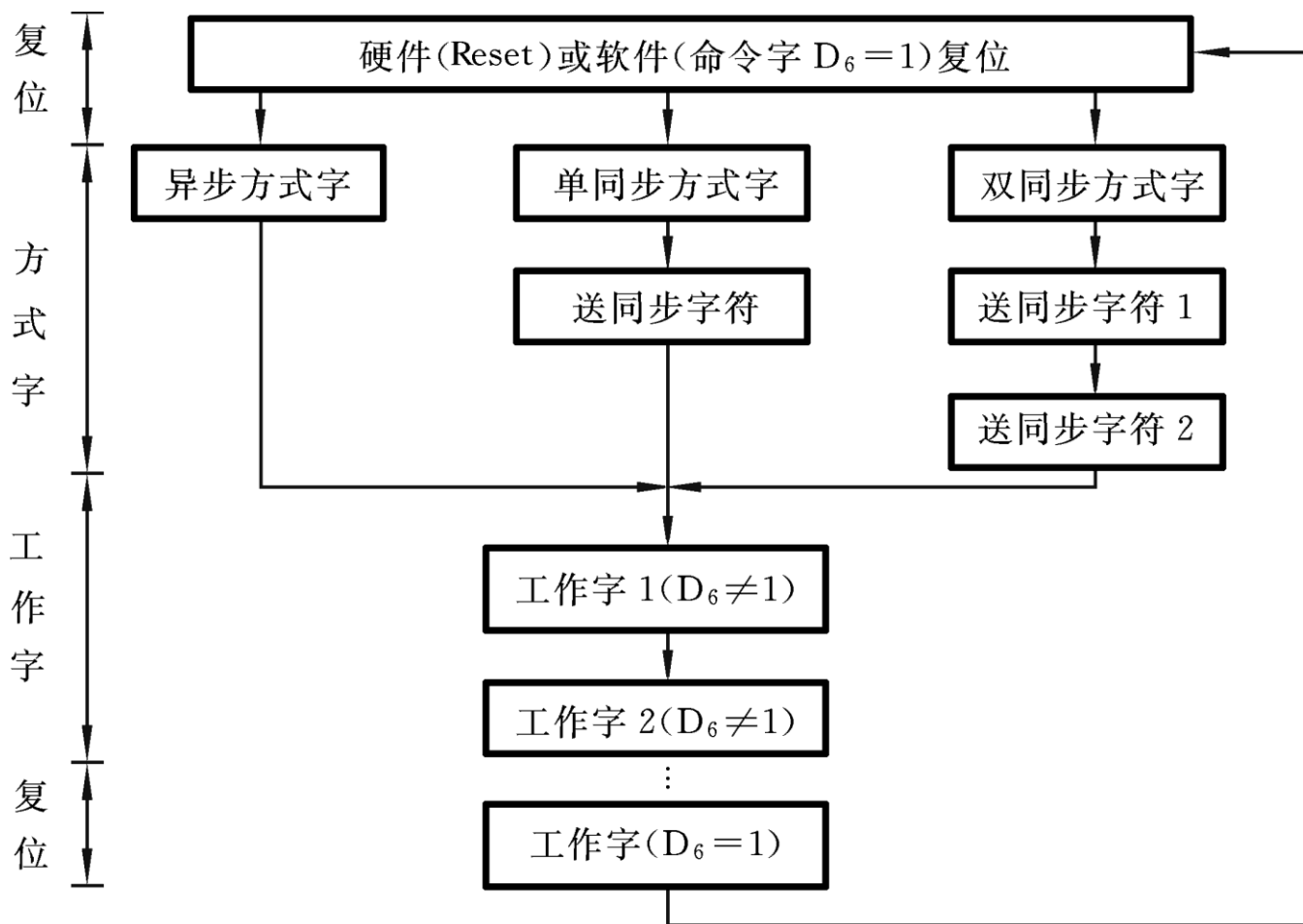


13.6.3 8251A的初始化内容与顺序

- 软件或硬件复位
- 设置方式字
- 如果设定在异步方式，则要写工作字，允许接收/发送；
- 如果设定在同步方式，那么在输出方式选择控制字后，应紧跟着输出一个同步字符或两个同步字符，然后再输出操作命令字，后面的操作与异步方式相同。
- 在数据传送过程中，可以使用操作命令字读取8251A的状态



约定：复位之后的第一个字是方式字！





13.6.4 基于8251A的串行通信接口设计

例1:采用RS-232C标准的双机串行通信接口设计

1. 要求：在甲乙二台PC之间基于8251A进行串行通信，甲发送乙接收，要求把甲机上开发应用程序（数据长度为2DH）传送到乙机中去。

采用：① 起止式异步方式；

② 字符长度为8位；

③ 2位停止位；

④ 波特率因子为64，无校验；

⑤ 波特率为4800bps；

⑥ CPU与8251A之间用**查询方式**交换数据

端口地址：309H命令/状态口，308H为数据口

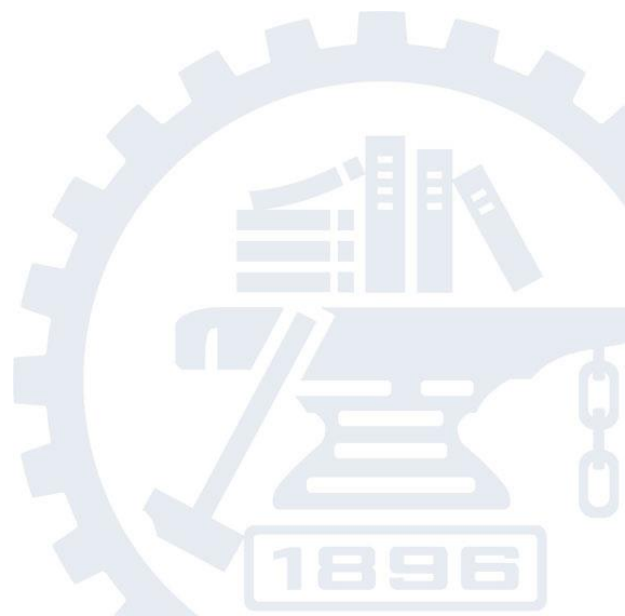


2. 分析

由于是近距离传输，不用MODEM，直接互连。由于采用查询I/O方式，故收/发程序中只需检查发/收准备好的状态是否置位，即可收发1个字节。

发送时查 T_xRDY 是否为1

接收时查 R_xRDY 是否为1





3. 设计

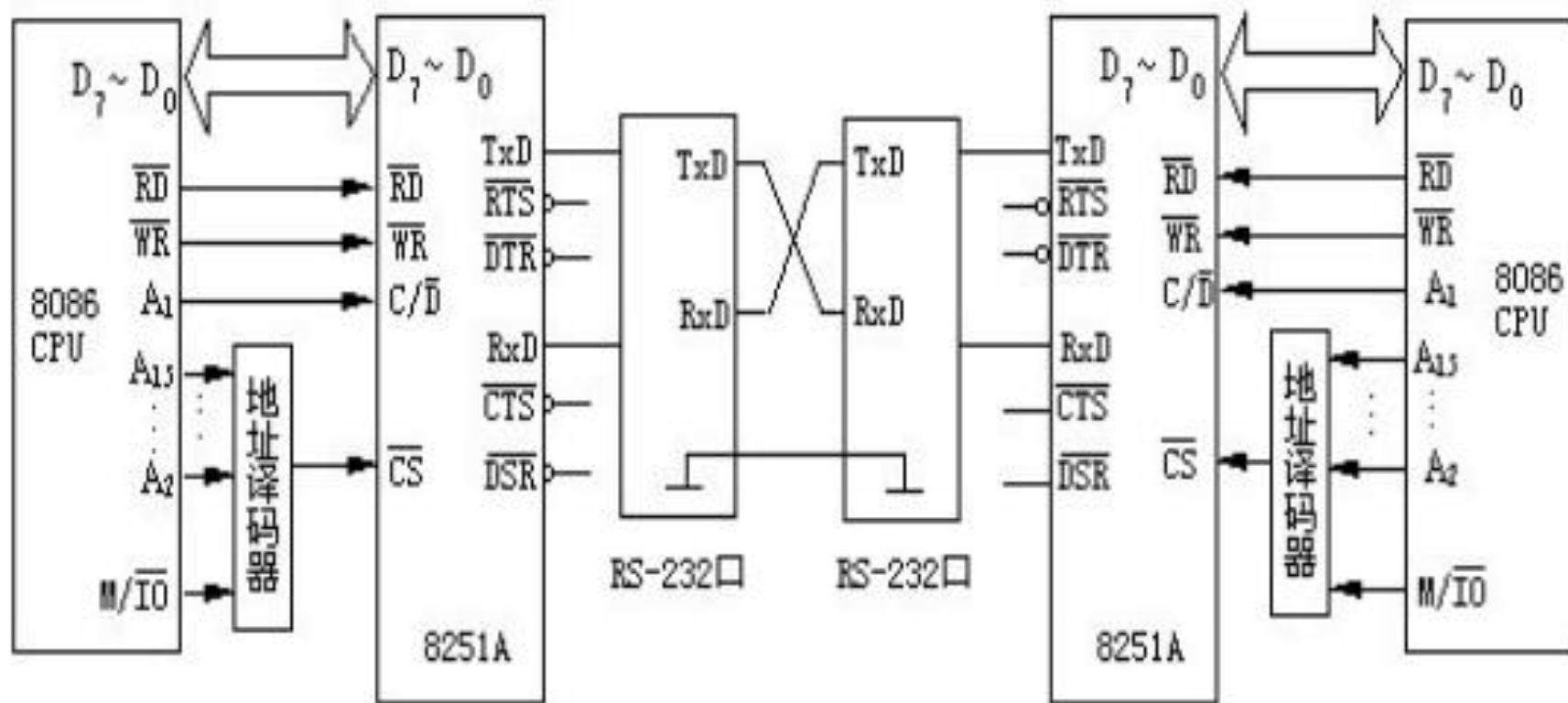
1) 硬件连接

根据以上分析把二台PC机都当作数据终端设备DTE，它们之间**只需 T_xD 、 R_xD 和SG**三根线连接就能通信
收/发时钟为 $64 \times 4800 = 307200\text{hz}$ ，直接接到甲机的发送时钟端和乙机的接收时钟端。

主要电路：

- ① 8251A串行口 ② TTL/EIA变换器
- ③ 波特率发生器 ④ 地址译码电路

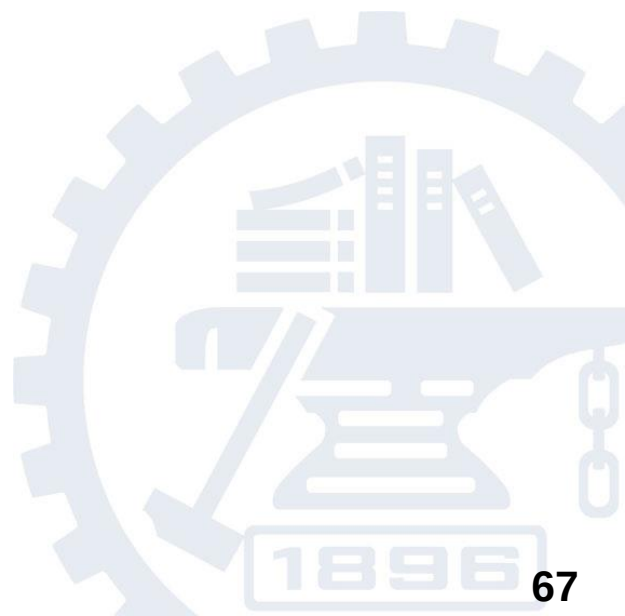




RS232标准的双机串行通信接口电路框图



2) 软件编程：接收和发送程序分开编写，每个程序段包括初始化、状态查询和输入输出部分。





① 甲机发送程序

DATA SEGMENT

BUF_T DB 45 DUP(?)

DATA ENDS

START: MOV DX, 309H ; 8251A命令端口

MOV AL, 00H ; 为提高可靠性（因为不确定reset是否被按下过），向命令口发送任意（ $D_6! = 1$ ）的数据，这样就能够确保后续再次向该端口写入的命令一定就是工作字，从而可以进行复位操作；如果一上来就写一个字，难以确认其是方式字还是工作字，更别说要复位了

OUT DX, AL

MOV AL, 40H ; 内部复位（ $D_6=1$ ），工作字

OUT DX, AL

NOP

MOV AL, 0CFH ; 方式命令字, 11001111B

OUT DX, AL

- ① 起止式异步方式，2位停止位
- ② 字符长度为8位
- ③ 无校验
- ④ 波特率因子为64



D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
EH	IR	RTS	ER	SBRK	RxEN	DTR	TxEN
进入搜索方式	内部复位	发送请求	错误标志复位	发中止符	接收允许	数据终端准备好	发送允许

1=内部复位
0=不复位

1=错误标志复位
0=不复位

1=允许收
0=不允许收

1=允许发
0=不允许发

MOV AL, 11H; 工作命令字

OUT DX, AL

MOV CX, 2DH ; 传送字节数

MOV SI, OFFSET BUT_T ; 发送区首址

L1: MOV DX, 309H ; 状态端口

IN AL, DX ; 查状态位D₀ (T_xRDY) =1?

AND AL, 01H

JZ L1 ; 发送未准备好, 则等待

MOV DX, 308H ; 数据口

MOV AL, [SI] ; 准备好, 则从发送区取一个字节发送



```
OUT DX, AL
INC SI          ; 内存地址加1
DEC CX          ; 字节数减1
JNZ L1          ; 未发送完, 继续
MOV AX, 4C00H   ; 已送完, 回DOS
INT 21H
CSEG ENDS
END START
```



D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
S ₁	S ₀	EP	PEN	L ₁	L ₀	B ₁	B ₀
停止位		奇偶校验		字符长度		波特率因子	

(同步) (异步)

×0=内同步 00=不用

×1=外同步 01=1位

0×=双同步 10=1.5位

1×=单同步 11=2位

×0=无校验

01=奇校验

11=偶校验

00=5位

01=6位

10=7位

11=8位

(异步) (同步)

00=不用 00=同步

01=×1 —

10=×16 —

11=×64 —

① 起止式异步方式，2位停止位

② 字符长度为8位

③ 无校验

④ 波特率因子为64

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
EH	IR	RTS	ER	SBRK	RxEN	DTR	TxEN
进入搜索方式	内部复位	发送请求	错误标志复位	发中止符	接收允许	数据终端准备好	发送允许

1=内部复位

0=不复位

1=错误标志复位

0=不复位

1=允许收

0=不允许收

1=允许发

0=不允许发

BEGIN: MOV DX, 309H ; 命令口

MOV AL, 00H ; 空操作，向命令口送任意数

OUT DX, AL

MOV AL, 40H ; 内部复位 (使D₆=1)

OUT DX, AL

NOP

MOV AL, 0CFH ; 方式命令字, 11001111B

OUT DX, AL

MOV AL, 00010100B; 工作命令字

OUT DX, AL



D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
DSR	SYNDET	FE	OE	PE	TxE	RxRDY	TxRDY
数传机就绪	同步字符检出	格式错	溢出错	奇偶错	发送器空	接收准备好	发送准备好
		1=格式错 0=无错	1=溢出错 0=无错	1=奇偶错 0=无错		1=接收就绪 0=未就绪	1=发送就绪 0=未就绪

MOV CX, 2DH ; 接收字节数

MOV DI, OFFSET BUF_R ; 接收区首址

L2: MOV DX, 309H ; 状态口

IN AL, DX ; 先查接收是否有错

AND AL, 38H ; 同时查3种错误

JNZ ERR ; 有错误, 到ERR

AND AL, 02H ; 查状态D₁ (R_xRDY) =1?

JZ L2 ; 没接收到数据, 等待





```
MOV DX, 308H      ; 数据口
IN  AL, DX        ; 接收一个字节
MOV [DI], AL      ; 存一个字节到接收区
INC  DI           ; 内存地址加1
LOOP L2
JMP  STOP         ; 没接收完, 继续
ERR: (略)
STOP: MOV AX, 4C00H ; 已送完, 回DOS
      INT 21H
CSEG ENDS
      END BEGIN
```





例2：全双工串行通信接口设计

(1) 要求

- ① 甲乙两台微机之间，基于8251A进行**全双工异步**串行通信，双方在各自的键盘上按键向对方**发送字符的同时又可接收**对方发来的**字符**，**采用查询方式**。
- ② 通信数据格式为：1位停止位，7位数据位，无校验，波特率因子为16。
- ③ 发送的字符和接收的字符均**在屏幕上显示**。
- ④ 波特率可选（2400b/s、4800b/s、9600b/s等）。
- ⑤ 按下ESC键，程序退出。



(2) 分析

- ① 由于要求波特率可选，故**需要设置波特率时钟发生器**（使用82C54）。
- ② **82C54应该输出方波**，所以必定工作在**3方式**，此时的门控信号GATE要由外部控制（82C55）。
- ③ 该通信接口电路涉及到8251A、MAX232等。

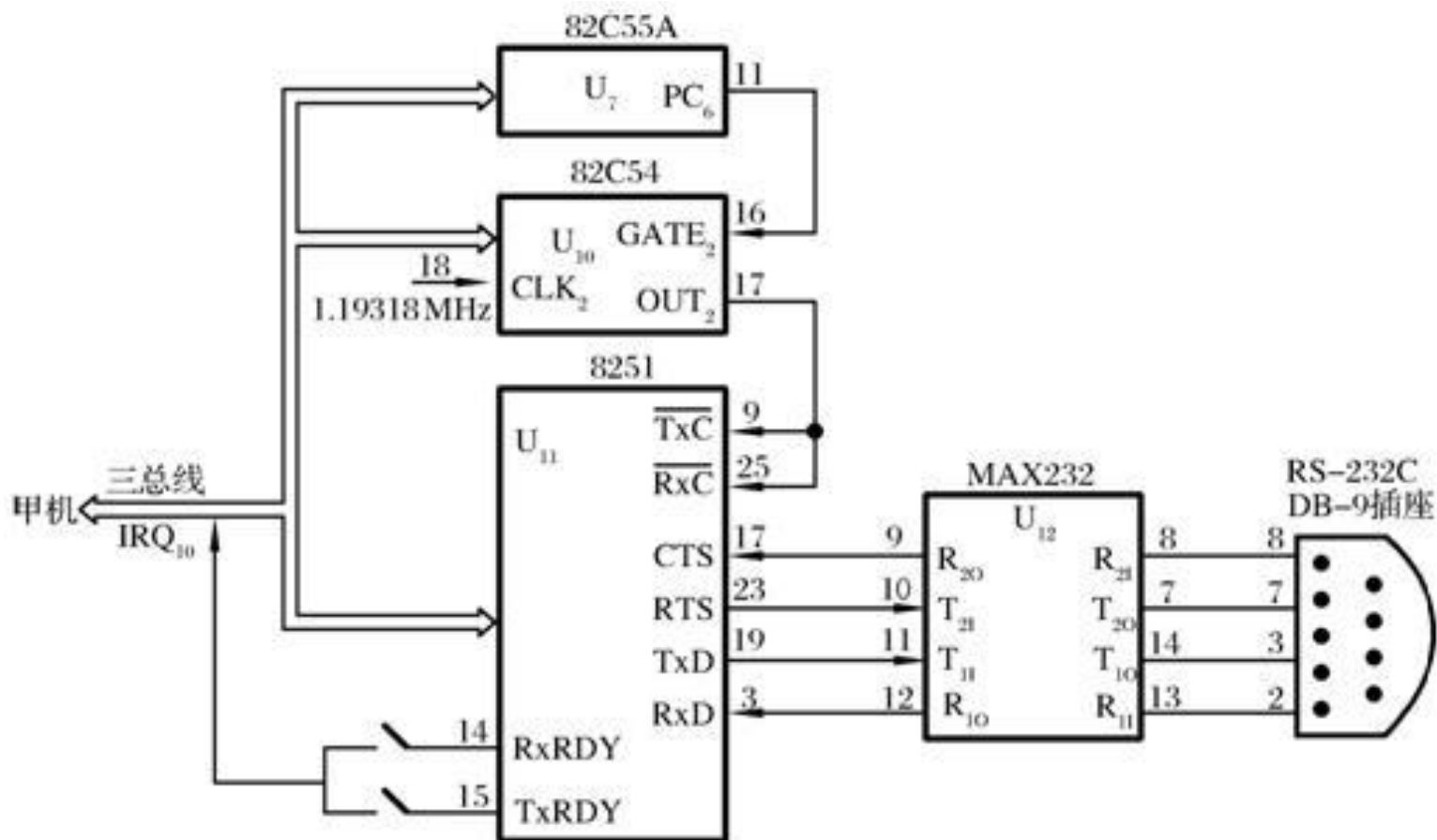
(3) 设计

① 硬件设计

串行通信接口电路以8251A为主，还要加上82C54作波特率时钟发生器，82C55控制GATE₂，MAX232作电平转换。



硬件电路





② 软件设计

通信程序由主程序和子程序组成。

子程序的工作：

8251A和82C55A的初始化；

82C54A初始化和计数初值的装入。

主程序的工作：

波特率的选择；

接收和发送字符；

在屏幕上显示；

查有无ESC键按下，以便结束程序。





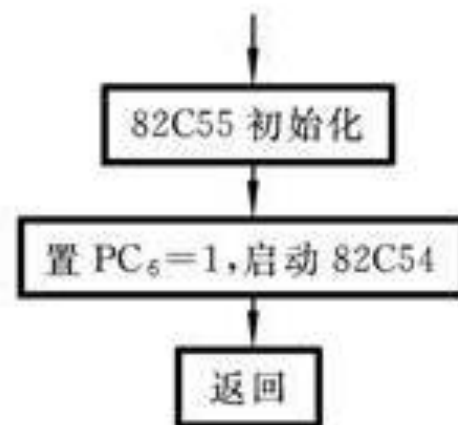
子程序流程图:



(a)



(b)

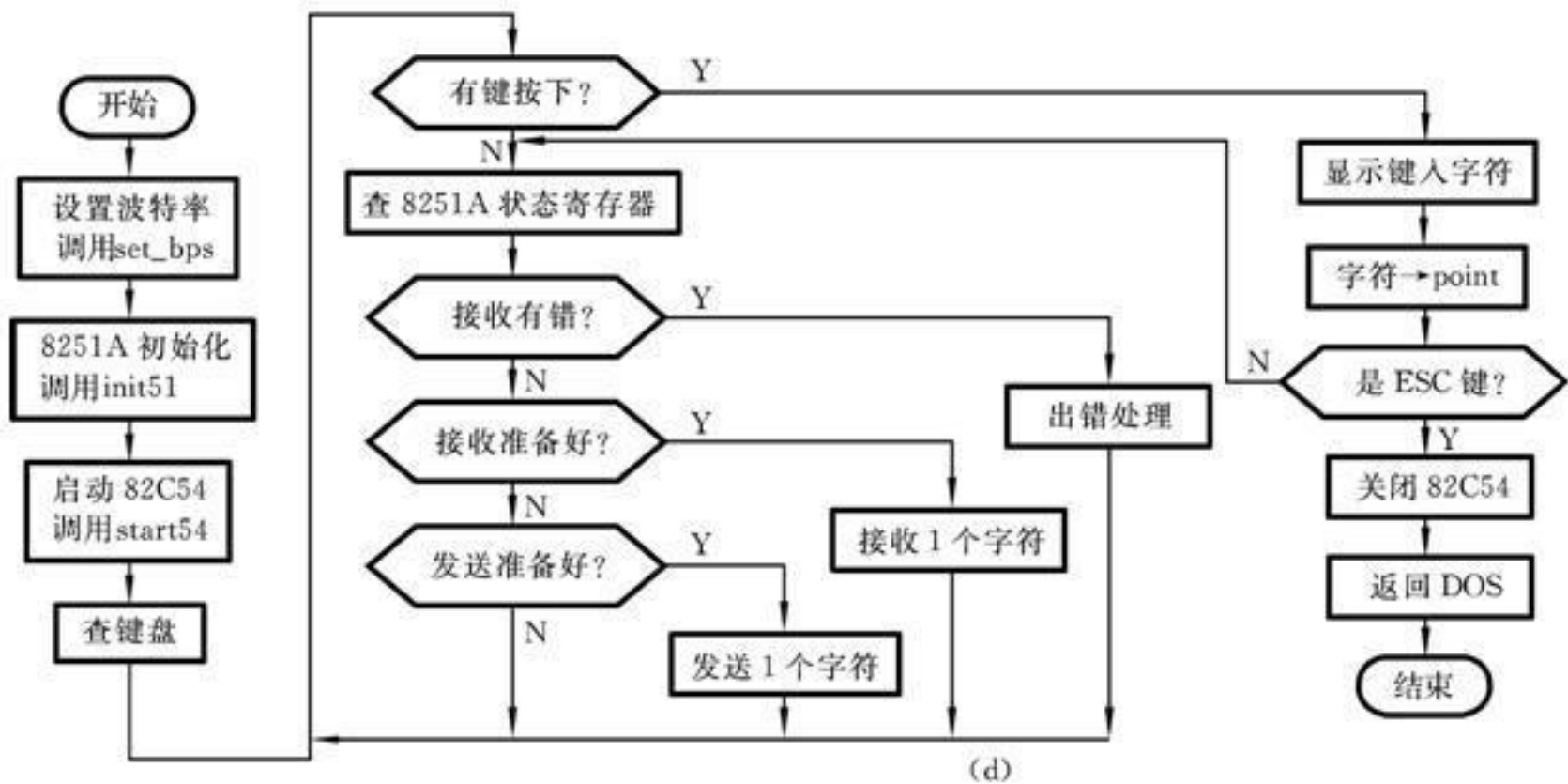


(c)





主程序流程图:





8251A初始化子程序

INIT51 PROC NEAR

MOV DX, CTR51 ; CTR51是8251A命令口

XOR AX, AX

OUT DX, AL ; 空操作, 向命令口送任意数

MOV AL, 40H ; 内部复位 (使D₆=1)

OUT DX, AL

MOV AL, 01001010B ; 方式命令字 (异步, 1位停止位,
; 字符长度为7位, 无校验, 波特
; 率因子为16个/位)

OUT DX, AL

MOV AL, 00010101B ; 工作命令字 (ER=R_xEN=T_xEN=1)

OUT DX, AL

RET





8251A初始化子程序

INIT51 PROC NEAR

MOV DX, CTR51

XOR AX, AX

OUT DX, AL ; 空

MOV AL, 40H

OUT DX, AL

MOV AL, **01001010B** ; 方式命令字 (**异步**, **1位停止位**,
; **字符长度为7位**, **无校验**, **波特**
; **率因子为16个/位**)

OUT DX, AL

MOV AL, **00010101B** ; 工作命令字 (ER=RxEN=TxEN=1)

OUT DX, AL

RET

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
S ₁	S ₀	EP	PEN	L ₁	L ₀	B ₁	B ₀
停止位		奇偶校验		字符长度		波特率因子	

(同步) (异步)

×0=内同步 00=不用 ×0=无校验

×1=外同步 01=1位 01=奇校验

0×=双同步 10=1.5位 11=偶校验

1×=单同步 11=2位

00=5位

01=6位

10=7位

11=8位

(异步) (同步)

00=不用 00=同步

01=×1 —

10=×16 —

11=×64 —

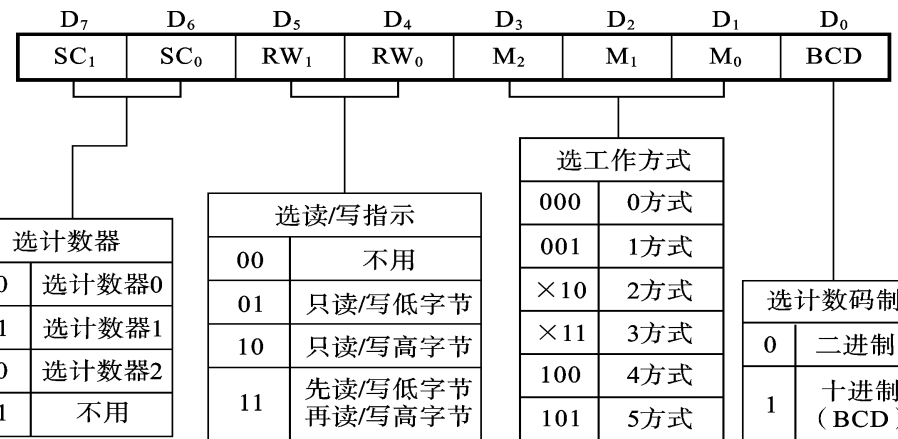
D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
EH	IR	RTS	ER	SBRK	RxEN	DTR	TxEN
进入搜索方式	内部复位	发送请求	错误标志复位	发中止符	接收允许	数据终端准备好	发送允许

1=内部复位
0=不复位

1=错误标志复位
0=不复位

1=允许收
0=不允许收

1=允许发
0=不允许发



82C54A初始化子程序

调用该子程序时，时间常数已经放入BX中

```
INIT54 PROC NEAR
```

```
    MOV DX, TIMCTR    ; TIMCTR是82C54命令口
```

```
    MOV AL, 10110110B; 定时器2, 先低后高,
```

```
                    ; 3方式, 二进制
```

```
    OUT DX, AL ; 对82C54初始化
```

```
    MOV DX, TIMER2    ; TIMER2是定时器2端口
```

```
    MOV AX, BX
```

```
    OUT DX, AL        ; 先低 (时间常数)
```

```
    MOV AL, AH
```

```
    OUT DX, AL        ; 后高 (时间常数)
```

```
    RET
```



D ₇ = 1	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
特征位		A 组方式		A 端口	PC ₄ ~ PC ₇	B 组方式	B 端口
		00 = 0 方式		0 = 出	0 = 出	0 = 0 方式	0 = 出
		01 = 1 方式		1 = 入	1 = 入	1 = 1 方式	1 = 入
		10 = 2 方式					
		11 = 不用					
							PC ₀ ~ PC ₃

82C55初始化和启动82C54子程序

INIT55 PROC NEAR

```
MOV DX, CTR55      ; CTR55是82C55命令口
MOV AL, 10000000B; 方式0, 各口均设为输出
OUT DX, AL ; 对82C55初始化
MOV AL, 00001101B
OUT DX, AL          ; 置PC6=1, 打开GATE2
                    ; 使定时器2输出方波
RET
```

D ₇ =0	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
特征位		未使用（写 000 或 111）		指定输出的引脚			输出电平
				000=PC ₀			1= 输出 1
				001=PC ₁			0= 输出 0
				⋮			
				111=PC ₇			



④ 主程序的相关部分

```
...  
CALL INIT54          ; 82C54A初始化程序  
CALL INIT51          ; 8251A初始化程序  
CALL INIT55          ; 82C55A初始化程序  
...  
LOP:  MOV AH, 0BH      ; 有键按下吗?  
      INT 21H          ; 有键按下AL=0FFH, 无键AL=0H  
      CMP AL, 0  
      JE  CHECK        ; 无, 查8251的状态寄存器  
      MOV AH, 01H      ; 有, 从键盘上读入, 并显示  
      INT 21H  
      MOV POINT, AL    ; 保存键值, 以备发送  
      CMP AL, 1BH      ; 是ESC键吗?  
      JE  OVER         ; 是就返回DOS
```





D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
DSR	SYNDET	FE	OE	PE	TxE	RxRDY	TxRDY
数传机就绪	同步字符检出	格式错	溢出错	奇偶错	发送器空	接收准备好	发送准备好
		1=格式错 0=无错	1=溢出错 0=无错	1=奇偶错 0=无错		1=接收就绪 0=未就绪	1=发送就绪 0=未就绪



CHECK: **MOV DX, CTR51** ; 查8251的状态寄存器
IN AL, DX
TEST AL, 38H ; 数据接收出现错误了吗?
JNZ ERROR ; 有错, 进行出错处理
TEST AL, 02H ; 没有错, 已接收到数据吗?
JNZ RECEIVE ; 转入接收字符程序
TEST AL, 01H ; 发送字符准备好了吗?
JNZ TRANSMIT ; 转入发送字符程序
JMP LOP ; 转LOP去查键盘按键

TRANSMIT:

MOV DX, DATA51 ; 82C51的数据口
MOV AL, POINT ; 取键入值
OUT DX, AL ; 发送该键入字符
JMP LOP





RECEIVE:

MOV DX, DATA51 ; 82C51的数据口

IN AL, DX ; 接收1个数据

MOV DL, AL

MOV AH, 02H ; 显示接收的数据

INT 21H

JMP LOP ; 接收完1个字符后, 继续循环

ERROR: ... ; 略

JMP LOP

OVER: MOV AH, 4CH ; 接收到ESC键后结束程序

INT 21H



小结：8251A后面接什么驱动器，就完成什么样的通信线路的传送。

例：仅接RS232转换，则完成零MODEM的近距离(<15m)传送。

例：仅接RS485转换，则完成稍远距离(几十米、几百米)点到点直通传送。

例：接RS232转换的同时又接MODEM，则完成专线或共用电话网的远距离（大于千米）传送。





串行通信接口16550自学内容:

1、16550的外部引脚特性

理解管脚与状态寄存器的关系

2、16550的内部寄存器及端口地址

- 16550的内部寄存器的端口地址公用问题
- 16550内部寄存器的格式

3、16550的编程

- 16550的初始化
- 16550的查询方式数据传输
- 16550的中断方式数据传输

