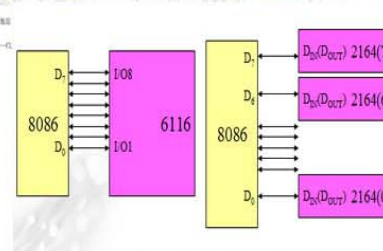
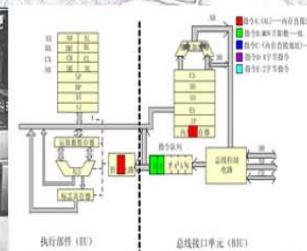
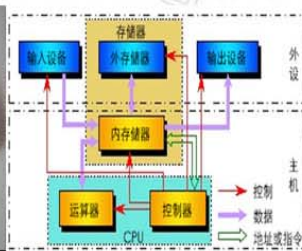




# 微型计算机原理及其应用

## 第十章 串行接口及可编程接口芯片8251A





# 第十章 串行接口及可编程接口芯片8251A

## ◆ 数据传输（通信）



并行传输



串行传输





## 第十章 串行接口及可编程接口芯片8251A



### 10.1 串行通信概述

### 10.2 可编程串行通信接口芯片8251A

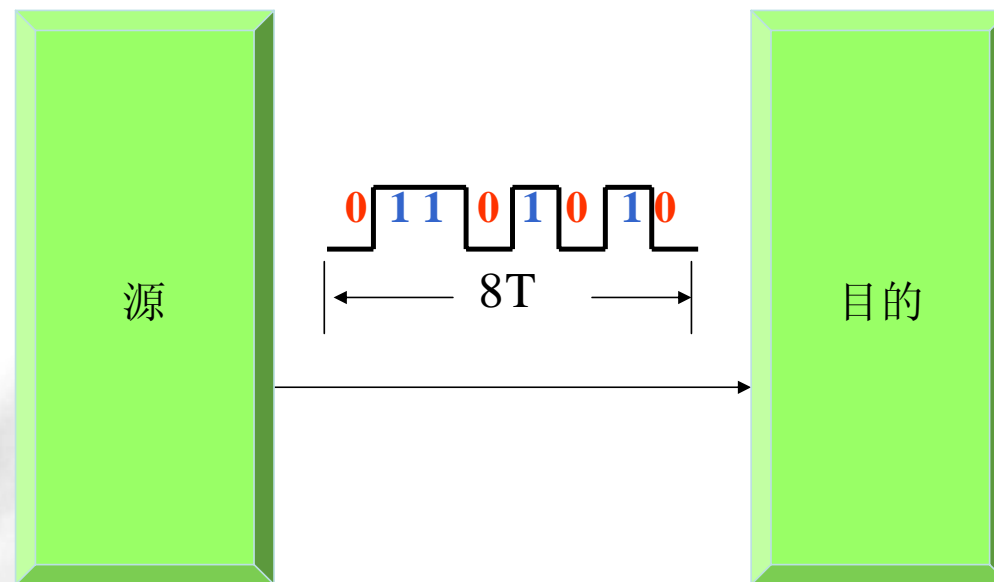
### 10.3 RS-232C串行口



## 10.1 串行通信概述

### 概念

- 数据在单条1位宽的传输线上，一位一位地按顺序传送。
  - ✧ 在串行传送中，1字节的数据要通过一条传输线分8次由低位到高位按顺序一位一位地传送。



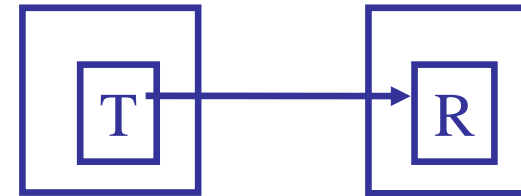


## 10.1 串行通信概述

### ◆ 串行通信的传送方向

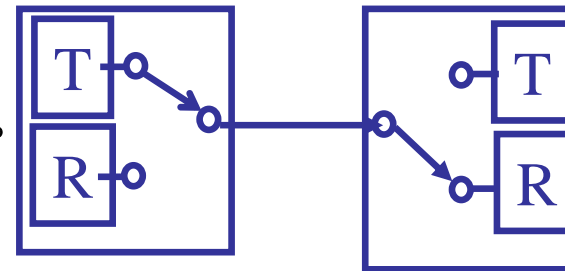
#### ➤ 单工

- ◇ 发送与接收之间只有一根传输线。
- ◇ 只能单向传送。



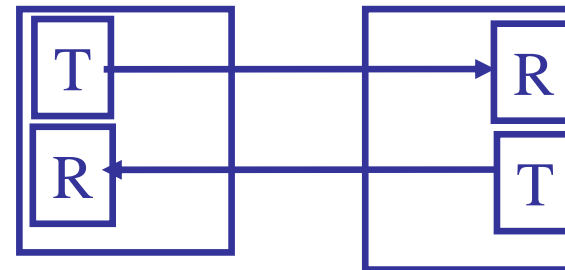
#### ➤ 半双工

- ◇ 发送与接收之间只有一根传输线。
- ◇ 但能分时交替进行双向传送。



#### ➤ 全双工

- ◇ 发送与接收之间有两条传输线。
- ◇ 能双向同时传送。





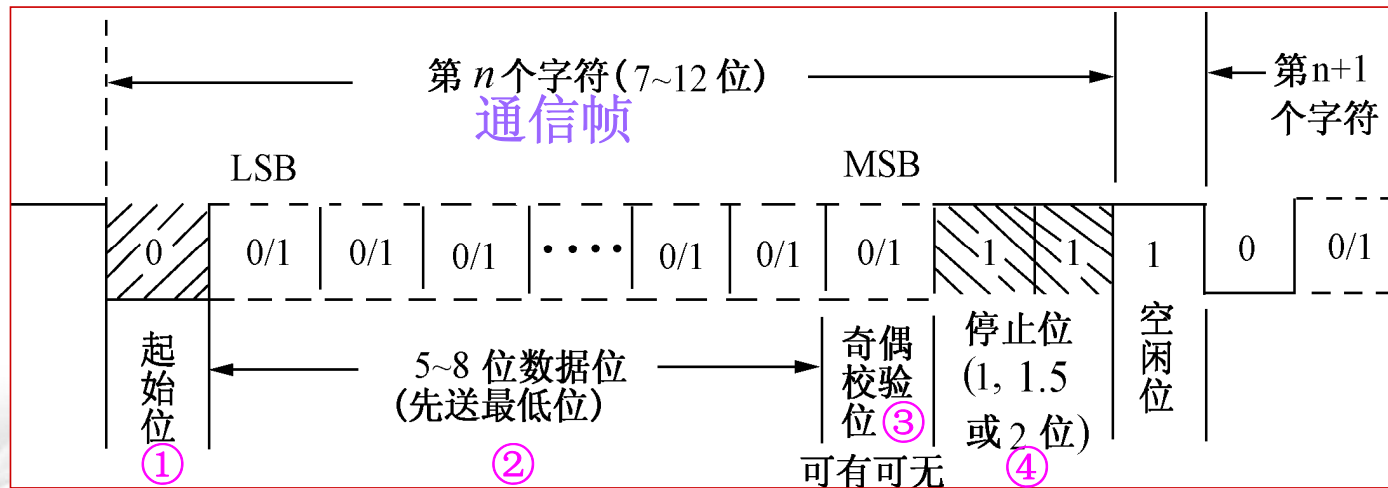
# 10.1 串行通信概述

## 串行通信的两种基本工作方式

传送效率低

### 异步通信方式 (ASYNC)

- 以**一个字符**为传输单位。通信中两个字符间的时间间隔是**不固定**的，但在同一个字符中的两个相邻位代码间的时间间隔是**固定**的。
- 异步通信的每个字符由四部分组成。



★ 两个相邻字符间的间隔叫**空闲位** (MARK状态, 高电平), 任意个。

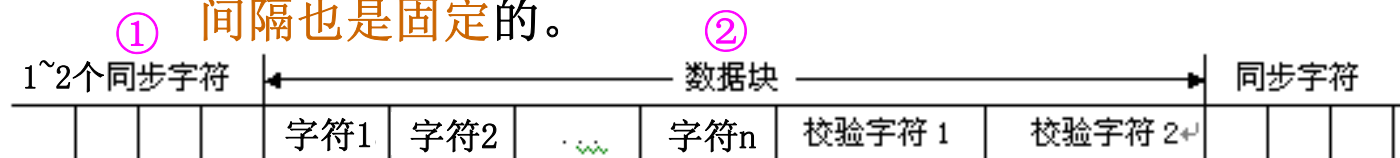


## 10.1 串行通信概述

### ◆ 串行通信的两种基本工作方式（续）

#### ➤ 同步通信方式（SYNC）

- ◇ 以数据块（一组字符）为单位。通信中每个字符间的时间间隔是相等的，而且每个字符中各相邻位代码间的时间间隔也是固定的。



- ◇ 收、发两端需用同一个时钟源作为时钟信号。
- ◇ 同步通信传输效率高，适合于快速、大量数据的传送。



## 10.1 串行通信概述



### ◆ 小结

#### ➤ 同步通信方式

- ◇ 字符与字符之间、字符内部的位与位之间都是同步。
- ◇ 传送速度最高可达**500**千波特。
- ◇ 需要配置结构复杂的同步时钟电路。
- ◇ 常用于计算机之间的通信，或主机与外设间的通信。

#### ➤ 异步通信方式

- ◇ 字符与字符间的传送是完全异步的，位与位之间的传送基本上是同步的（借助起始位和停止位同步）。
- ◇ 传送速度一般在**50~19200**波特之间。
- ◇ 常用于计算机和**CRT**终端、字符打印机之间的通信。





## 10.1 串行通信概述



### ◆ 串行传送速率

#### ➤ 波特率 (Baud Rate)

◇ **定义:** 单位时间 (每秒) 内传送二进制数据的位数。它是衡量串行数据传送速度的重要参数。

◇ **单位:** 波特 (位/秒)

◇ **计算方法:** 波特率 = 数据传送速率 (字符/秒) × 字符位数

- **【举例】** 一个串行字符由1个起始位、7个数据位、1个奇偶校验位和1个停止位组成，每秒传送120个字符，则数据传送的波特率为：

$120 \text{ 字符/秒} \times 10 = 1200 \text{ 位/秒} = 1200 \text{ 波特}$  (即每位占用0.833毫秒)

◇ **异步串行** 传送常用的波特率：110、300、600、1200、2400、4800、9600、19200、28800、36400、57600波特。

◇ **同步传送** 的波特率高于异步传送方式，最高能到上百千波特。



## 10.1 串行通信概述

### ◆ 串行通信接口芯片

- 通用同步异步收发器UASRT——8251A
- 通用异步收发器UART

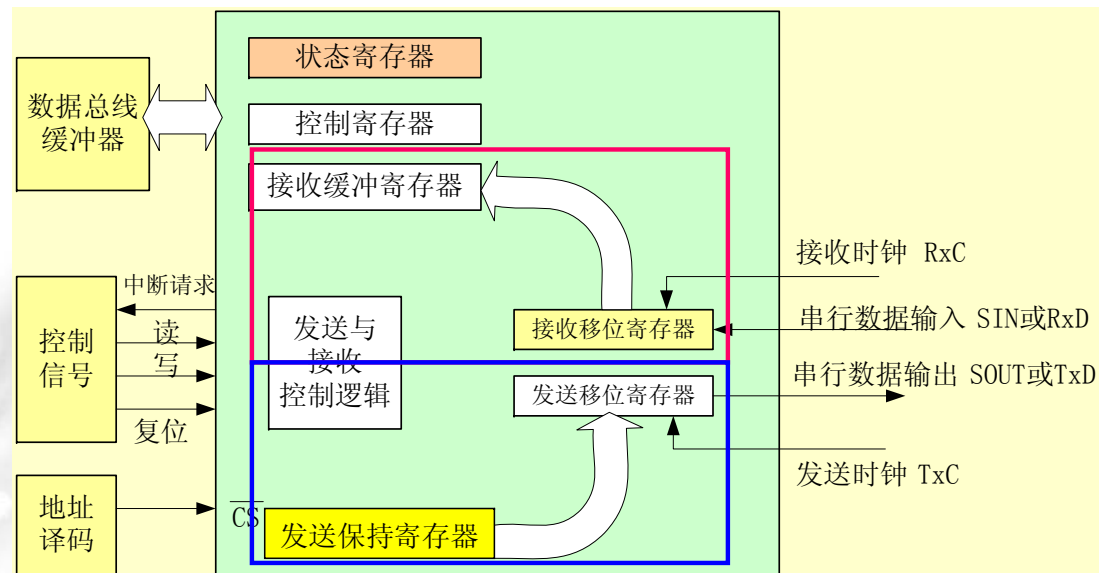


## 10.1 串行通信概述

### 通用异步收发器UART

#### 组成

- ◇ **接收器**: 负责数据接收, 并把串行码转换为并行码。
- ◇ **发送器**: 负责把并行码转换为串行码, 并执行发送任务。
- ◇ **控制器**: 主要用来接收CPU的控制信号, 执行CPU所要求的操作, 并输出状态信息和控制信息。





## 10.1 串行通信概述

### 通用异步收发器UART（续）

#### ➤ 功能

##### ◇ 双向处理

- 接收外设输入的异步串行数据并转换为CPU所需并行数据。
- 将CPU的并行数据转换成串行数据并输出。

何谓  
奇/偶  
校验?

##### ◇ 奇偶校验——检测长距离传送中可能发生的错误。

- 发送时，检查要传送字符的“1”的个数，并在奇偶校验位上上添上“1”或“0”，使“1”的总数（包括校验位本身）为偶数或奇数。
- 接收时，检查字符的每一位以及奇偶校验位的“1”的个数，以确定是否发生传送错误。



## 10.1 串行通信概述

### ◆ 通用异步收发器UART（续）

#### ➤ 常设的错误标志

##### ◇ 奇偶错误

- 接收时对字符中“1”的个数进行统计，若与字符格式中设置的奇偶性不一致，则置位该标志，发奇偶错信息。

##### ◇ 帧错误

- 接收的字符格式不符合规定(如无停止位等)，则置位该标志，发帧错误信息。

##### ◇ 溢出(丢失)错误

- 当接收到第二个字符的停止位时，CPU还未取走前一个接收的字符，就会出现数据丢失现象，则置位该标志。



## 第十章 串行接口及可编程接口芯片8251A



**10.1 串行通信概述**

**10.2 可编程串行通信接口芯片8251A**

**10.3 RS-232C串行口**





## 10.2 可编程串行通信接口芯片8251A



- 1. 8251A的内部结构和引脚**
- 2. 8251A的初始化编程**
- 3. 8251A应用举例**



## 10.2.1 8251A的内部结构和引脚



### ◆ 基本性能

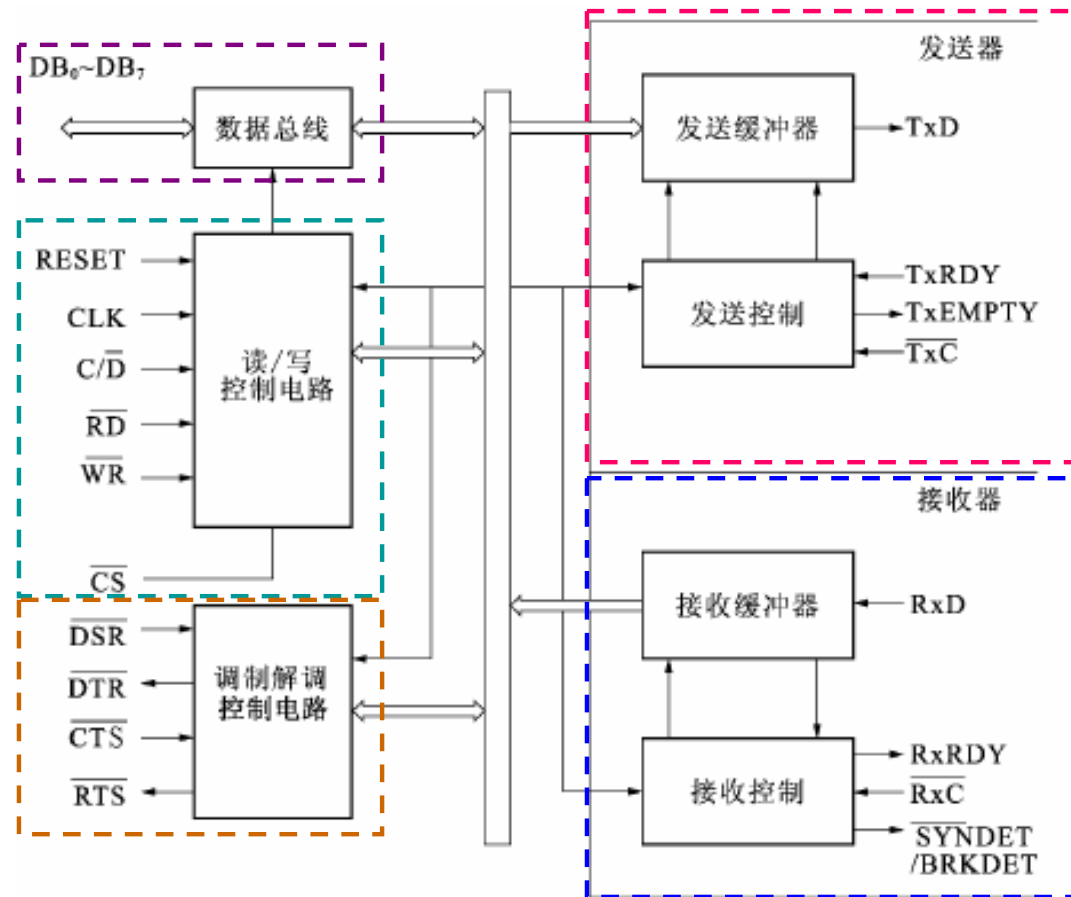
- 可用于同步或异步传送。
- 在同步方式中，可进行5~8位字符的操作，可用内部或外部同步，自动插入同步字符。
- 在异步方式中，可进行5~8位字符的操作，波特率系数可为1、16或64。
- 可产生1、1.5或2位的停止位，可检查假启动位；自动检测和处理终止字符。
- 波特率：0~19.2K（异步），0~64K（同步）。
- 全双工的工作方式：其内部提供具有双缓冲器的发送器和接收器。
- 出错检测，具有奇偶、溢出和帧错误等检测电路。





## 10.2.1 8251A的内部结构和引脚

### 内部结构

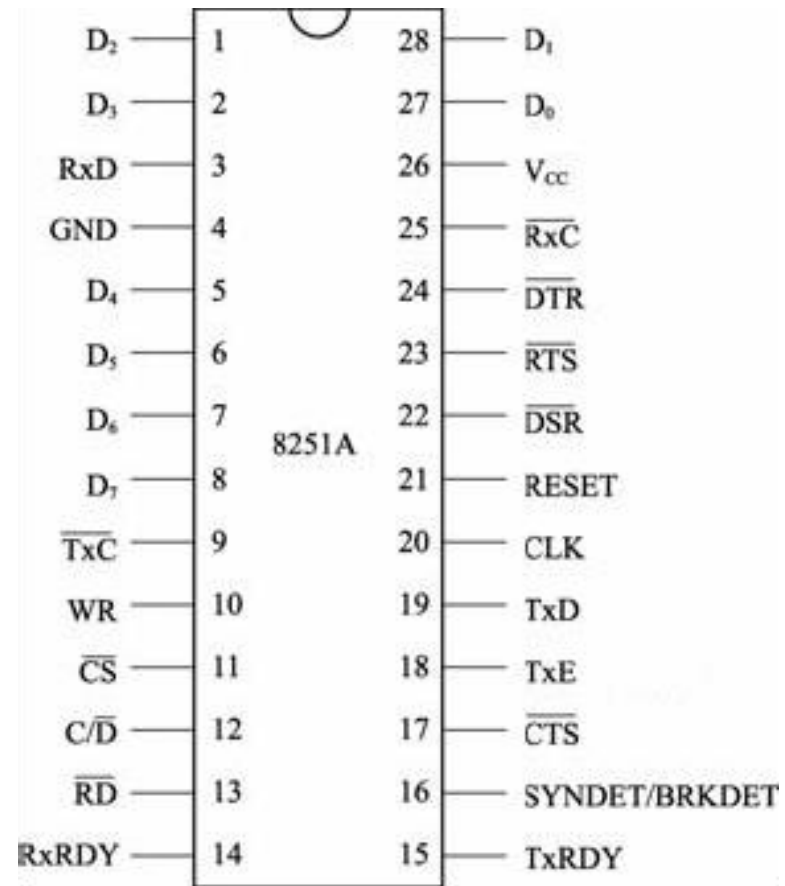




## 10.2.1 8251A的内部结构和引脚

### ❖ 芯片封装

- 双列直插
- 28根引脚





## 10.2.1 8251A的内部结构和引脚

### ❖ 数据总线缓冲器

- 作用：与系统总线间的接口。
  - ❖ 把接收到的数据送给CPU，或把CPU发来的数据送给外设。
  - ❖ 还可把状态寄存器中的状态信息读到CPU中。
  - ❖ 在8251A初始化时，可分别把方式字、控制字和同步字符送到方式寄存器、控制寄存器和同步字符寄存器中。
- 内部包含3个8位双向三态缓冲器：
  - ❖ 状态缓冲器：用IN指令从中读取状态信息。
  - ❖ 接收数据缓冲器：用IN指令从中读取数据信息。
  - ❖ 发送数据/命令缓冲器：存放CPU用OUT指令向8251A写入的数据或命令字。
- 对应引脚（8根）
  - ❖  $D_7 \sim D_0$



## 10.2.1 8251A的内部结构和引脚



### 读写控制电路

- **作用：**接收CPU的控制信号和命令字，用以决定8251A的工作方式，并向内部其它功能部件发出控制信号。
- **对应引脚<来自CPU的控制信号>（6根）**
  - ✧ **RESET**——复位信号，输入，高电平有效。
    - 使8251A进入空闲状态，等待初始化编程。
  - ✧ **CLK**——主时钟信号，输入。
    - 用来产生内部定时信号。
    - **同步方式**下，CLK的频率必须比 $\overline{TxC}$ 和 $\overline{RxC}$ 大30倍。
    - **异步方式**下，CLK的频率应比 $\overline{TxC}$ 和 $\overline{RxC}$ 大4.5倍。
  - ✧  **$\overline{RD}$** ——读信号，输入，低电平有效。
    - 表示CPU从8251A读出数据或状态信息。



## 10.2.1 8251A的内部结构和引脚

### 读写控制电路（续）

- ◇  $\overline{WR}$ ——写信号，输入，低电平有效。
  - 表示CPU把数据或控制字写入8251A。
- ◇  $\overline{CS}$ ——片选信号，输入，低电平有效。
  - 表示8251A被CPU选中，可以对它进行读写操作。
- ◇  $C/\overline{D}$ ——控制/数据信号，输入。
  - 高电平时，表示数据总线传输的是控制信号或状态字。
  - 低电平时，表示数据总线传输的是数据信息。

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	$C/\overline{D}$	执行的 操作
0	0	1	0	CPU由8251A输入数据
0	1	0	0	CPU向8251A输出数据
0	0	1	1	CPU读取8251A的状态
0	1	0	1	CPU向8251A写入控制命令



## 10.2.1 8251A的内部结构和引脚

### 接收器和接收控制电路

- **作用：**接收RxD引脚上输入的**串行数据**，并将它们转换为**并行数据**，存放在**接收数据缓冲寄存器**中。
- **工作原理**
  - ◇ **异步通信方式**
    - 在无字符传送时，RxD引脚上为高电平。
    - 当RxD引脚上出现低电平，则启动一个内部计数器。当计数到一个数据位宽度的一半时，重新采样RxD引脚，若仍为低电平，则确认为起始位，而非噪声信号。
    - 接收后续数据位，并送**移位寄存器**处理，经奇偶校验和去停止位后，得到并行数据，送**接收数据缓冲器**，同时发出RxRDY信号通知CPU。



## 10.2.1 8251A的内部结构和引脚

### 接收器和接收控制电路（续）

#### ◇同步通信方式

- 监视RxD引脚，以**一次一位**的方式将数据送至**接收寄存器**。
- 每接收一位，将接收寄存器与存放同步字符的寄存器比较。若不相等，则接收下一位后继续比较；若相等，则表示搜索到同步字符，置SYNDET=1，表示已达到同步。
- 若是双同步，必须要连续检测到两个同步字符后才认为已达到同步。
- 同步后，根据 $\overline{\text{RxC}}$ 引脚送入的同步时钟，逐个移位RxD引脚上的数据位，并按规定位数将其送至**接收数据缓冲器**，同时发出RxRDY信号通知CPU。



## 10.2.1 8251A的内部结构和引脚

### 接收器和接收控制电路（续）

#### ➤ 对应引脚（4根）

✧ RxD (Receiver Data) —— 接收数据，输入。

- 外部串行数据从RxD引脚逐位移入接收移位寄存器，经串到并，变成并行数据后，送入接收数据缓冲器。

✧ RxDY (Receiver Ready) —— 接收数据准备好，输出，高电平有效。

- 表示已收到一个字符数据，可送往CPU。
- 当CPU从接收数据缓冲器读取该字符后，复位为低电平，待接收到一个新字符，又变为高电平。
- 该信号与CPU的工作方式可以是中断方式或查询方式（作为联络信号）。





## 10.2.1 8251A的内部结构和引脚

### 接收器和接收控制电路（续）

◇SYNDET/BRKDET (Sync Detect/Break Detect)——同步检测/断点检测，输入或输出。

- **同步方式**：用于同步检测。复位时，此引脚变为低电平。
  - **内同步方式**下，为输出信号。若检测到同步字符，则该引脚输出高电平，表示达到同步状态。
  - **外同步方式**下，为输入信号。该引脚由低变高时，使8251A在下一个接收时钟的上升沿开始接收字符。达到同步后，高电平可撤除。
- **异步方式**：用于断点检测，输出。
  - 当RxD引脚连续收到两个全0数位组成的字符时，该引脚输出高电平，表示当前线路上无数据可读。
  - 可作为状态位，供CPU读出。



## 10.2.1 8251A的内部结构和引脚

### 接收器和接收控制电路（续）

◇  $\overline{\text{RxC}}$  (Receiver Clock) —— 接收时钟，输入。

- 决定8251A接收数据的速率。
- 异步方式下，该引脚输入的时钟频率可以是波特率的1倍、16倍或64倍。
- 接收时钟通常应与对方的发送时钟相同。



## 10.2.1 8251A的内部结构和引脚



### ✦ 发送器和发送控制电路

➤ **作用：**接收CPU送来的**并行数据**，并加上由**控制字**规定的成帧信号，再转换为**串行数据**从TxD引脚发送出去。

### ➤ 工作原理

#### ✦ 异步通信方式

- 加上起始位，并根据控制字加上奇偶校验位和停止位。

#### ✦ 同步通信方式

- 先发送同步（SYN）字符，随后发送若干数据字符，且两个字符之间不允许有间隔。



## 10.2.1 8251A的内部结构和引脚



### ✦ 发送器和发送控制电路（续）

#### ➤ 对应引脚（4根）

✦ TxD (Transmitter Data) —— 发送数据，输出。

- 将CPU送来的并行数据，转换成串行数据后，逐位从TxD引脚发送给外设。

✦ TxRDY (Transmitter Ready) —— 发送数据准备好，输出，高电平有效。

- 在允许发送数据，且数据总线缓冲器中的发送数据/命令缓冲器为空时，表示已准备好从CPU接收一个数据。
- 该信号与CPU的工作方式可以是**中断方式或查询方式（作为联络信号）**。



## 10.2.1 8251A的内部结构和引脚



### ✦ 发送器和发送控制电路（续）

✦ TxE (Transmitter Empty) —— 发送缓冲器空，输出，高电平有效。

- 表示发送器中的并到串转换器空，即完成一次发送操作，缓冲器中已无数据向外部发送。
- 异步方式下，由TxD引脚向外部输出空闲位。
- 同步方式下，由TxD引脚向外部输出同步字符。

✦  $\overline{\text{TxC}}$  (Transmitter Clock) —— 发送器时钟，输入。

- 决定8251A的发送速率。
- 同步方式下，该引脚输入的时钟频率应等于发送数据的波特率。
- 异步方式下，由软件定义发送的时钟是波特率的1倍、16倍或64倍。



## 10.2.1 8251A的内部结构和引脚

### ◆ 调制解调控制电路

- 利用8251A与**调制解调器 (Modem)** 相连，经标准电话线可实现远距离数据传输。
  - ✧ 发送方通过调制解调器将输出的串行**数字信号变为模拟信号**，再发送出去。
  - ✧ 接收方必须将**模拟信号经过调制解调器变为数字信号**，才能由串行接口接收。
- 用于为**调制解调器**提供控制信号。



带串行接口的  
adsl modem



## 10.2.1 8251A的内部结构和引脚



### ✦ 调制解调控制电路（续）

#### ➤ 对应引脚<控制信号>（4根）

✧  $\overline{\text{DTR}}$  (Data Terminal Ready) —— 数据终端准备好，输出，低电平有效。

- 表示CPU准备好进行数据传送。
- 由控制命令字的D<sub>1</sub>位置1使之有效。

✧  $\overline{\text{DSR}}$  (Data set Ready) —— 数据设备准备好，输入，低电平有效。

- 对应状态寄存器D<sub>7</sub>位，CPU可通过读状态字获取数据设备的状态。



## 10.2.1 8251A的内部结构和引脚

### ✦ 调制解调控制电路（续）

✦  $\overline{\text{RTS}}$  (Request To Send) —— 请求传送数据，输出，低电平有效。

- 表示CPU已准备好发送数据。
- 由控制命令字的D<sub>5</sub>位置1使之有效。

✦  $\overline{\text{CTS}}$  (Clear To Send) —— 允许发送数据，输入，低电平有效。

- 是调制解调器或外设对 $\overline{\text{RTS}}$ 的响应信号
- 仅当 $\overline{\text{CTS}}$ 为低电平时，8251A才能执行发送操作。即是使引脚TxRDY=1的条件之一。





## 10.2 可编程串行通信接口芯片8251A

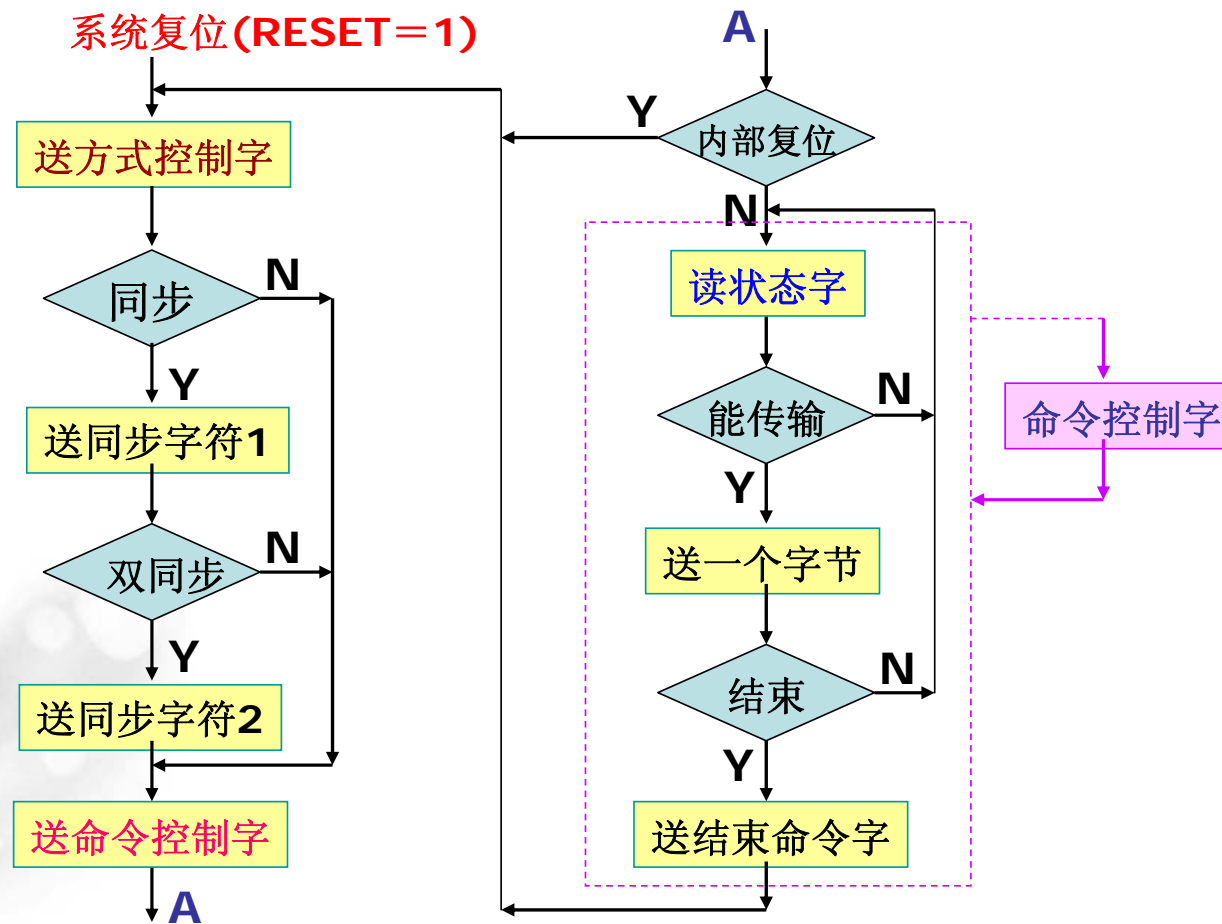


1. 8251A的内部结构和引脚
2. 8251A的初始化编程
3. 8251A应用举例



## 10.2.2 8251A的初始化编程

### 初始化流程





## 10.2.2 8251A的初始化编程



### ◆ 控制字与状态字

#### ➤ 方式控制字

- ◇ 用于定义8251A的一般工作特性。
- ◇ 紧接在复位操作后由CPU写入。

#### ➤ 命令控制字

- ◇ 用于指定芯片的实际操作。
- ◇ 写入方式控制字后，才能写入同步字符和命令控制字。
- ◇ 在工作过程中，可根据需要随时写入新的命令控制字。

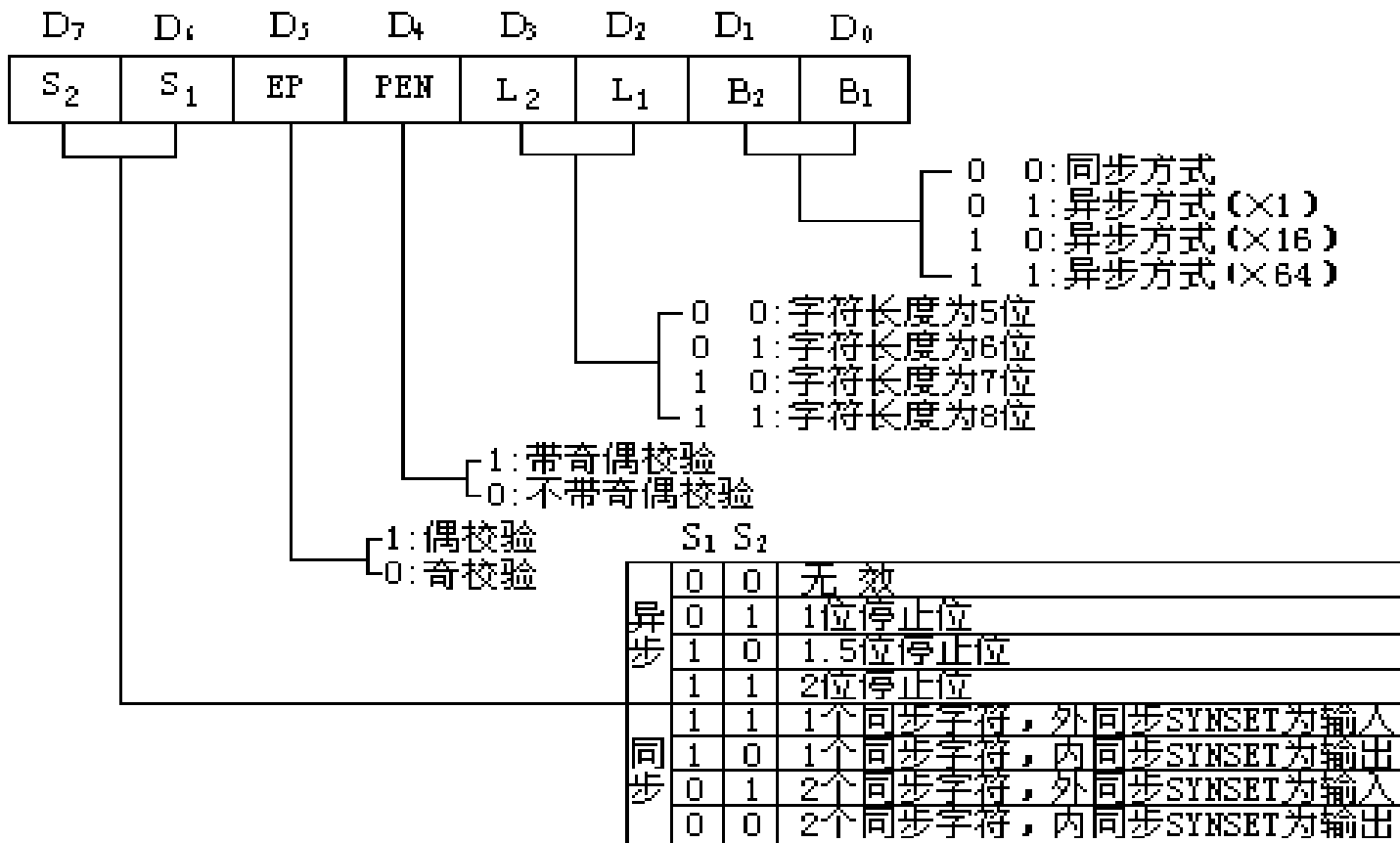
#### ➤ 状态字

- ◇ 给出8251A当前的运行状态。
- ◇ 除个别状态位外，其余状态位的定义与相应同名引脚有相同含义。



## 10.2.2 8251A的初始化编程

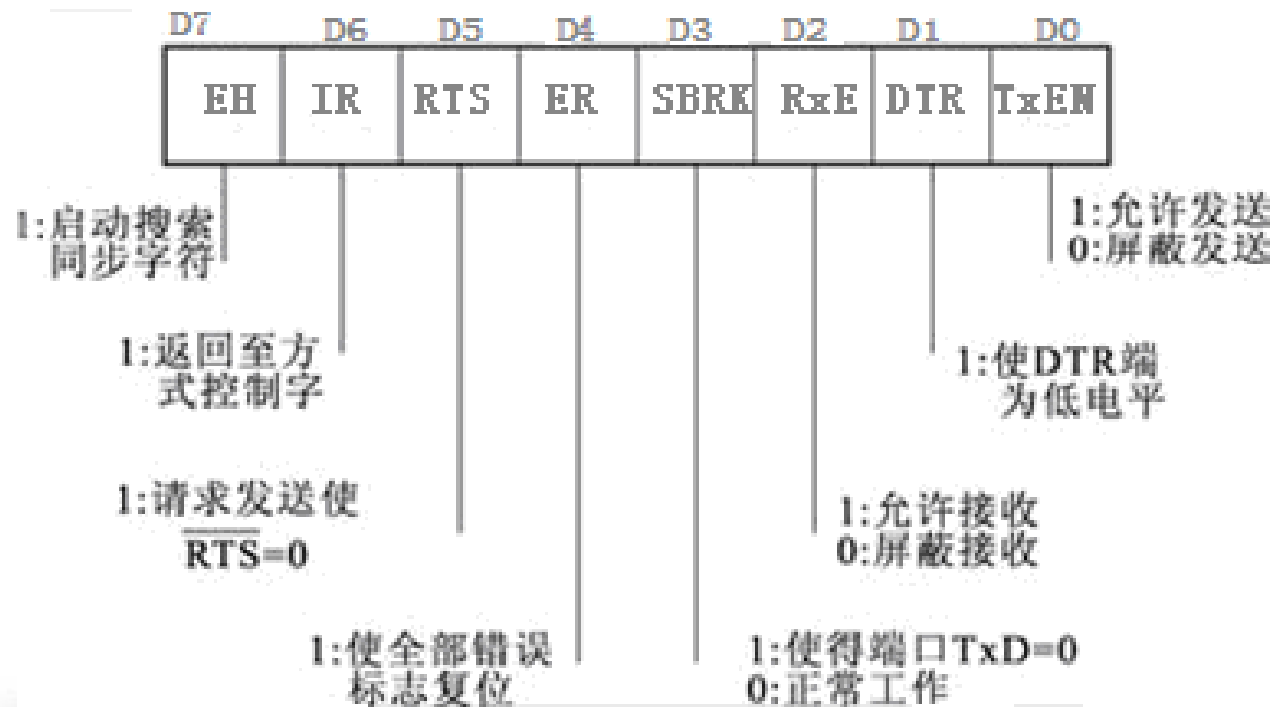
### ◆ 方式控制字





## 10.2.2 8251A的初始化编程

### ◆ 命令控制字

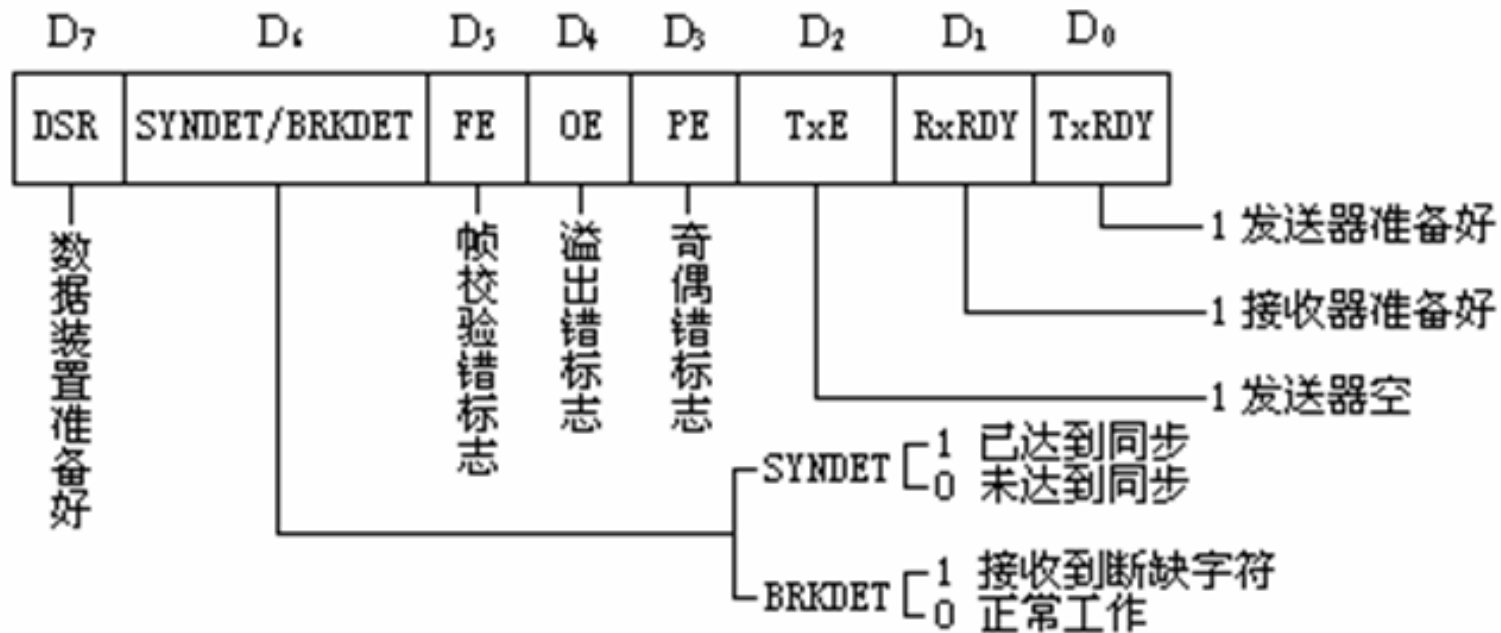


- ★说明:
1. 与命令控制字写入相同端口地址。
  2. 复位后写入的是方式控制字，此后写入的是命令控制字，且在对芯片复位以前，所写入的控制字都是命令控制字。



## 10.2.2 8251A的初始化编程

### 状态字



- ★说明：
1. 与命令控制字写入相同端口地址。
  2. 复位后写入的是方式控制字，此后写入的是命令控制字，且在芯片复位以前，所写入的控制字都是命令控制字。



## 10.2 可编程串行通信接口芯片8251A



1. 8251A的内部结构和引脚
2. 8251A的初始化编程
3. 8251A应用举例



## 10.2.3 8251A应用举例

### ✦ 异步方式初始化

- 【例】要求8251A工作于异步方式，波特率系数为16，具有7个数据位，1个停止位，有偶校验，控制口地址为3F2H，写恢复的延时程序为REVTIME。

#### ✦ 编程注意

- 为确保8251A可靠复位，应先向其控制端口连续写入3个全0，然后再向该端口送入一个使命令寄存器D<sub>6</sub>（IR）位为“1”的复位控制字（40H）。
- 复位之后，将方式控制字和命令控制字写入控制端口。
- 为确保控制字可靠写入，在两次写控制口操作之间必须至少有16个时钟周期间隔。





## 10.2.3 8251A应用举例



### ◆ 异步方式初始化（续）

初  
始  
化  
程  
序

```
MOV DX, 3F2H ; 控制口地址
MOV AL, 00H
OUT DX, AL ; 向控制口写入“0”
CALL REVTIME ; 延时，等待写操作完成
OUT DX, AL ; 向控制口写入第二个“0”
CALL REVTIME ; 延时
OUT DX, AL ; 向控制口写入第三个“0”
CALL REVTIME ; 延时
MOV AL, 40H ; 复位字
OUT DX, AL ; 写入复位字
REVTIME ; 延时
MOV AL, 01111010B ; 方式字
OUT DX, AL ; 写入方式字
CALL REVTIME ; 延时
MOV AL, 00010101B ; 写入命令字
OUT DX, AL ; 写入命令字
```



## 10.2.3 8251A应用举例

### ◆ 同步方式初始化

➤ 【例】假设8251A的控制口地址为3F2H，写恢复延时程序为REVTIME，如要求8251A工作于同步方式，采用双同步字符、奇校验、数据位为7位。

#### ◇ 编程注意

- 和异步方式一样，先向控制口写入3个0和一个软件复位命令字（40H）。
- 复位后，先写入方式控制字，然后向控制口写入同步字符。
  - 若方式字中规定为双同步方式，则必须对控制口写入第二个同步字符（同样的同步字符连续写两次）。
  - 常用ASCII码中的16H作为收发双方的同步字符。
- 同步字符写入后，再对控制口写入命令控制字，选通发送器和接收器，允许芯片对从RxD引脚上送来的数据位搜索同步字符。



## 10.2.3 8251A应用举例

### ◆ 同步方式初始化（续）

初  
始  
化  
程  
序

```
..... ; 先写入3个0，再送复位字40H
MOV DX, 3F2H ; 控制口地址
MOV AL, 00011000B ; 方式字
OUT DX, AL ; 送方式字
CALL REVTIME ; 延时
MOV AL, 16H
OUT DX, AL ; 送入第一个同步字符
CALL REVTIME
OUT DX, AL ; 送入第二个同步字符
CALL REVTIME
MOV AL, 10010101B ; 命令字
OUT DX, AL
```



## 第十章 串行接口及可编程接口芯片8251A



**10.1 串行通信概述**

**10.2 可编程串行通信接口芯片8251A**

**10.3 RS-232C串行口**





## 10.3 RS-232C串行口



- 1. RS-232C电器特性及接口信号**
- 2. RS-232C与TTL电平的转换**



## 10.3.1 RS-232C 电器特性及接口信号

### ◆ 同步方式初始化

➤ 【例】假设8251A的控制口地址为3F2H，写恢复延时程序为REVTIME，如要求8251A工作于同步方式，采用双同步字符、奇校验、数据位为7位。

#### ◇ 编程注意

- 和异步方式一样，先向控制口写入3个0和一个软件复位命令字（40H）。
- 复位后，先写入方式控制字，然后向控制口写入同步字符。
  - 若方式字中规定为双同步方式，则必须对控制口写入第二个同步字符（同样的同步字符连续写两次）。
  - 常用ASCII码中的16H作为收发双方的同步字符。
- 同步字符写入后，再对控制口写入命令控制字，选通发送器和接收器，允许芯片对从RxD引脚上送来的数据位搜索同步字符。



## 本章要点

- ✦ 弄清8255A三种工作方式的特点及三个端口分别可以工作在何种工作方式下。
- ✦ 掌握8255A两种控制字（方式选择控制字和C口置位/复位控制字）的分析方法。
- ✦ 熟悉将控制字写入控制端口的初始化编程。