

# 《微型计算机原理与接口技术》

## 第5版

### 第9章

## 串行通信和可编程 接口芯片8251A



## §9.2 可编程串行通信 接口芯片8251A



- ◆ 8251A是通用同步/异步数据收发器（USART），是常用的可编程通信接口器件，用于全双工通信并具有同步或异步工作方式。
- ◆ Intel8251A串行接口的基本任务。
  - ◆ （1）进行串、并转换。
  - ◆ （2）实现串行数据格式化。
  - ◆ （3）可靠性检验。
  - ◆ （4）实施接口与DCE之间联络控制



- ◆ 8251A的基本性能:
- ◆ 可用于同步和异步传输: 同步波特率是0~64K位/秒, 异步波特率是0~19.2K位/秒。
- ◆ 同步传送: 数据5~8位字符, 可使用内部/外部同步检验, 可自动插入同步字符。
- ◆ 异步方式: 数据5~8位, 时钟速率为波特率的1/16/64倍。即波特率系数为1、16和64,
- ◆ 可产生停止位1、1.5或2位, 能检查假启动位, 可自动产生、检测和处理中止符等。
- ◆ 全双工、双缓冲接收器和发送器
- ◆ 差错校验: 具有奇偶、溢出、帧错等检测电路。



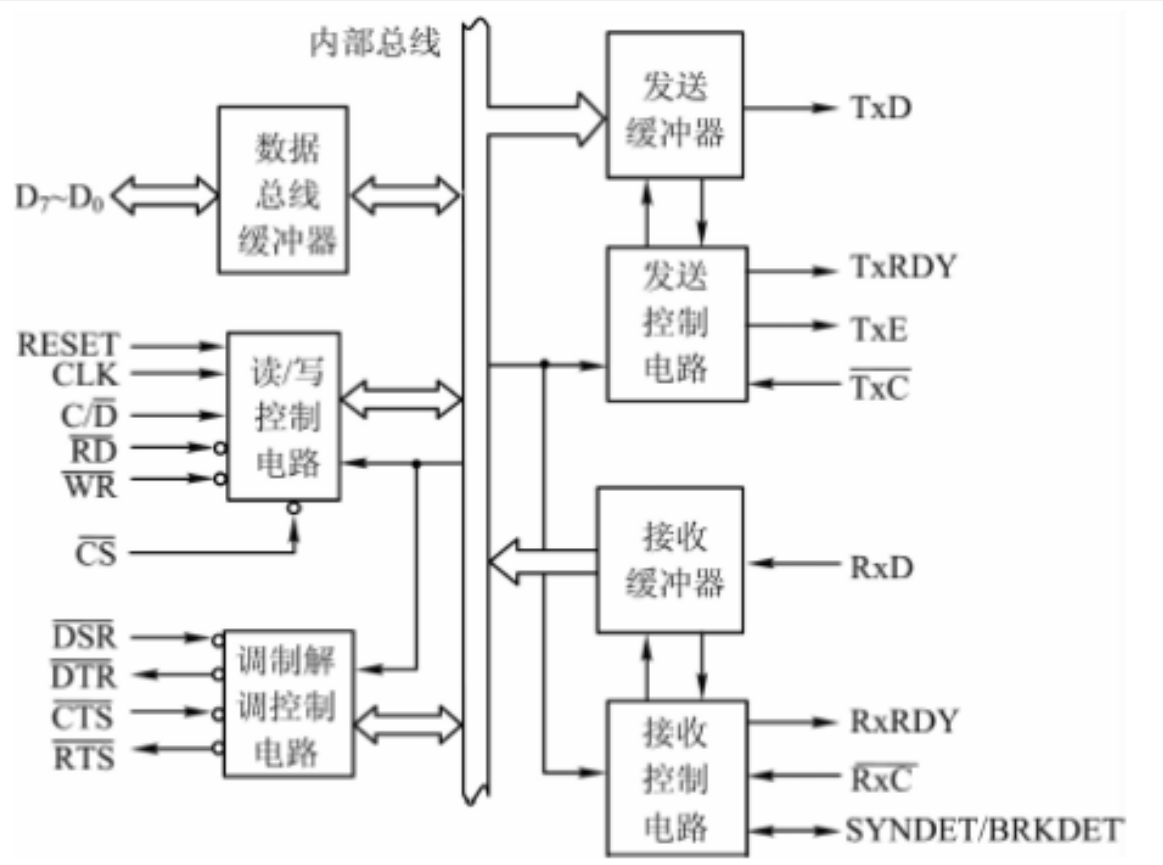
## 9.2.1 8251A的内部结构和外部引脚

## 9.2.2 8251A的编程

## 9.2.3 8251A应用举例



## 9.2.1 8251A的内部结构和外部引脚



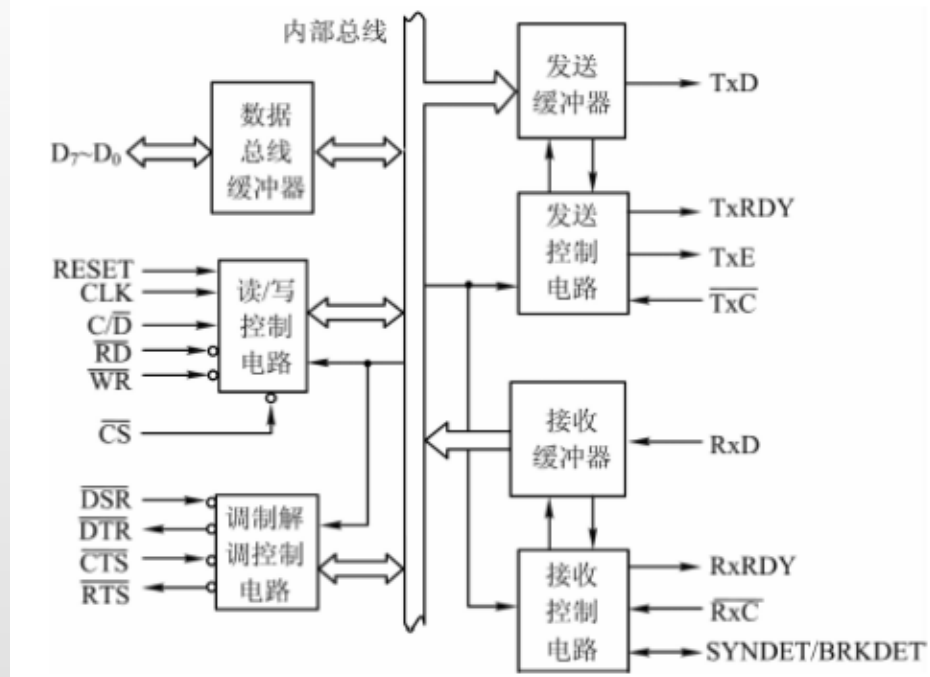
内部主要部件：

- 数据总线缓冲器
- 接收缓冲器和接收控制电路
- 发送缓冲器和发送控制电路
- 读写控制电路
- 调制解调控制电路

从图中也可看到各部件相应的引脚信号

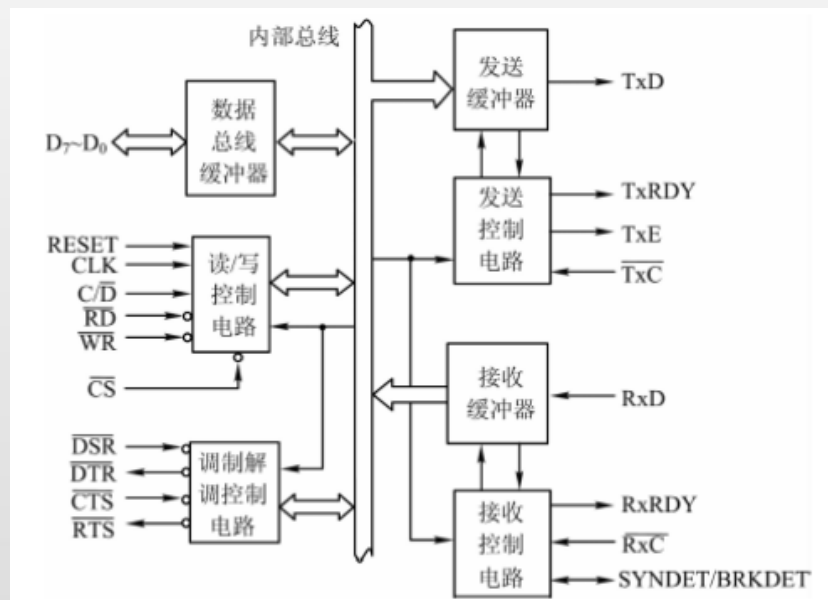
# 1. 数据总线缓冲器

- 它是8251A与系统数据总线间的接口，内部包含：
  - 状态缓冲器—存放8251A的状态信息；
  - 接收数据缓冲器—存放8251A接收的数据；
  - 发送数据/命令缓冲器—存放写入8251A的数据或命令（控制）字。
- $D_7 \sim D_0$  数据线
  - 与系统数据总线相连，用来传送在8251A和CPU间传送的数据信息、编程命令和状态信息。

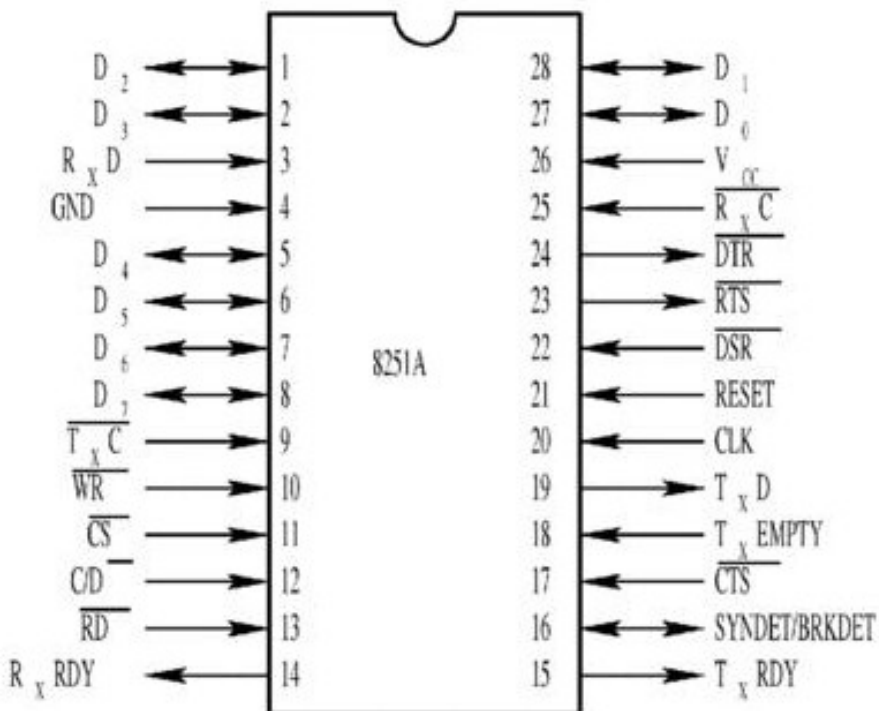


发送器：把待发并行数据转换成所要求的帧格式并加上校验位，在 $T_xC$ 作用下由 $T_xD$ 串行送出，发完一帧数据后 $T_xRDY=1$ ，通知CPU发下一帧数据。

接收器：在 $R_xC$ 作用下接收 $R_xD$ 的帧格式化串行数据并转换成并行，同时校验，如错误由放在状态寄存器中供CPU处理，无错误将并行数据放在数据总线缓冲器中，使 $R_xRDY=1$ ，通知CPU读数据







8251A

8251与外界信号可分为4组：

1. 与CPU的连接信号：

**D0-D7**：三态双向数据总线

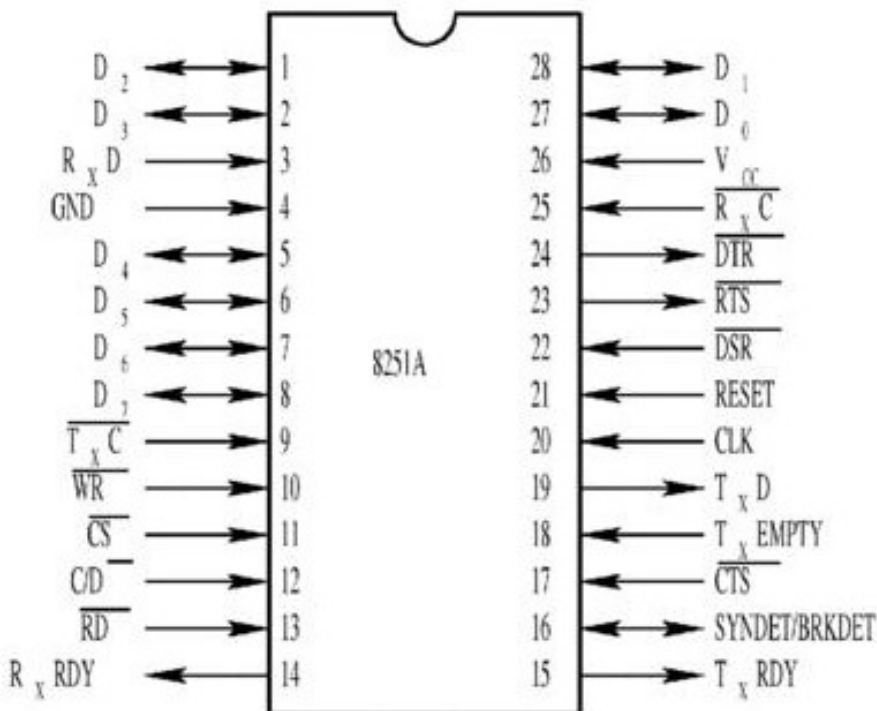
**RD#**，**WR#**：读写信号

**CS#**：片选信号

**RESET**：当出现6倍时钟的高电平时，芯片复位处于空闲状态等待命令

**C/D#**：控制/数据信号，  
等于1，CPU访问命令寄存器或状态寄存器；等于0：访问数据寄存器。





8251A

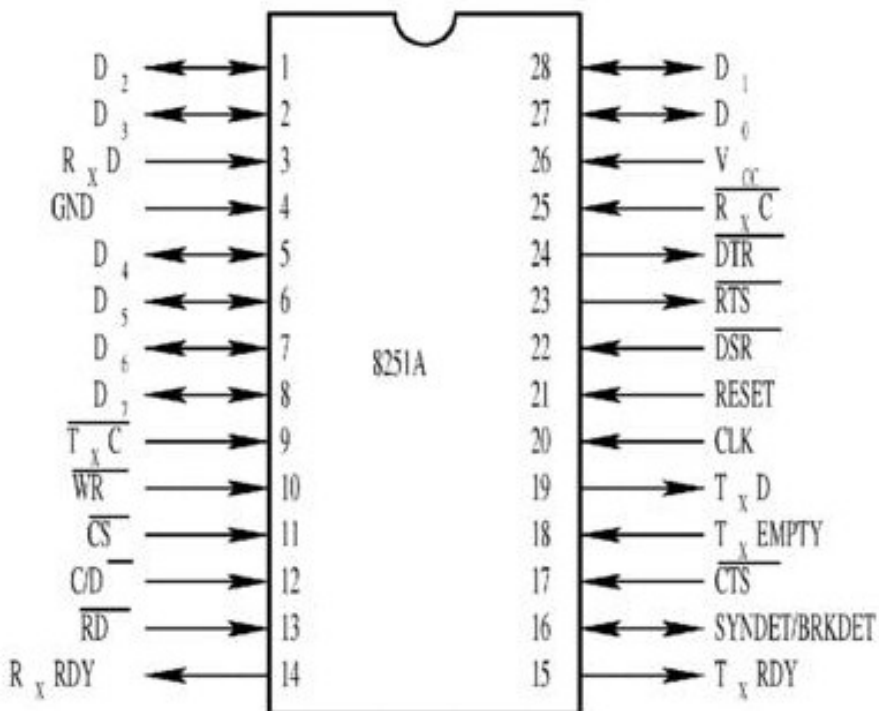
2. 状态信号或中断申请信号:

**TxRDY/RxRDY**: 高电平有效，发送/接收器准备好表示可以接收CPU送来的数据/已经将发送数据送往CPU，CPU发送/接收一个字符后，其自动复位。查询方式时，CPU通过状态寄存器D0/D1位检测该状态位。也可作为中断方式下的中断请求信号。

**TxE**: 发送器空，高电平有效。

**SYNDEF/BD**: 同步检出信号/间断检出信号





8251A

### 3. 时钟同步信号:

**TxC:** 发送器时钟

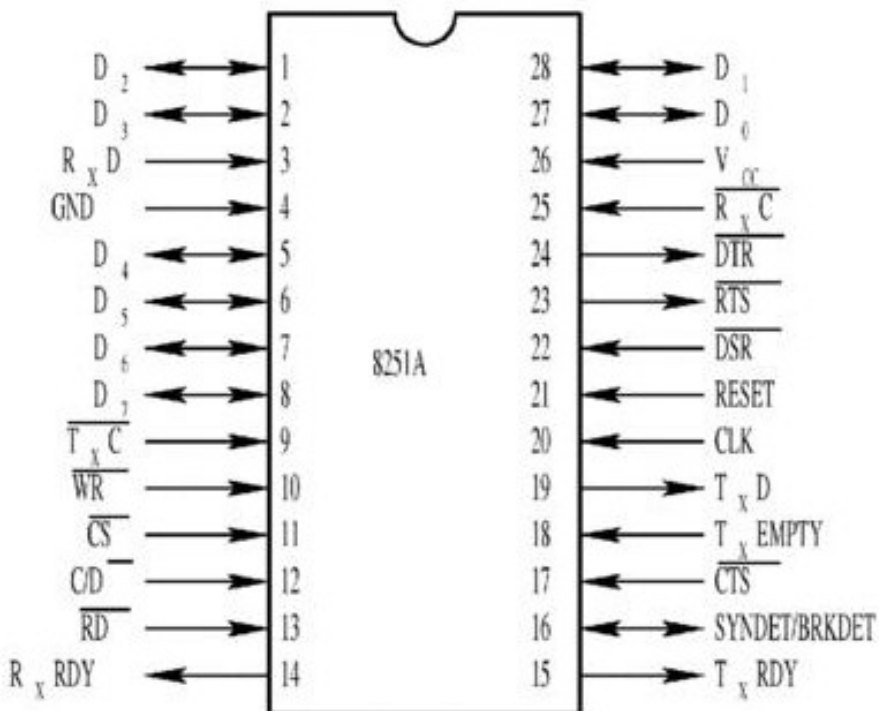
**RxC:** 接收器时钟

**CLK:** 工作时钟，由外部时钟源提供，为芯片内部电路提供定时，并非发送或接收数据的时钟。同步方式下CLK频率要大于TxC/RxC的30倍；异步方式下4-5倍。

### 4. 面向调制器的接口电路

8251提供了4个与MODEM相连的信号线其含义与RS232相同





8251A

## 5. 数据传送相关:

**DTR:** 数据终端准备好, 表示8251准备好、输出给外设。

**DSR:** 数据装置准备好, 输入。表示调制器准备好。

**RTS#:** 请求发送, 输出低电平有效, CPU通过编程输出低电平, 表示CPU已准备好发送。

**CTS#:** 清除传送 (允许发送) MODEM发给8251的对RTS的响应信号, 表示8251可发送数据。

**TxD:** 发送数据线。

**RxD:** 接收数据线。



## 4. 读/写控制电路

- CPU送到控制电路的信号:

- RESET: 高电平时复位, 等待初始化编程
- $\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{CS}$ : 读、写和片选信号
- CLK: 时钟信号, 产生8251A内部定时信号
- $C/\overline{D}$ : 控制口/数据口选择信号: 1-选择控制口, 0-选择数据口

$C/\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	操 作
0	0	1	0	CPU 从 8251A 读数据
0	1	0	0	CPU 向 8251A 写数据
1	0	1	0	CPU 读取 8251A 的状态字
1	1	0	0	CPU 向 8251A 写入控制字
×	1	1	0	数据总线浮空
×	×	×	1	数据总线浮空



**9.2.1 8251A的内部结构和外部引脚**

**9.2.2 8251A的编程**

**9.2.3 8251A应用举例**





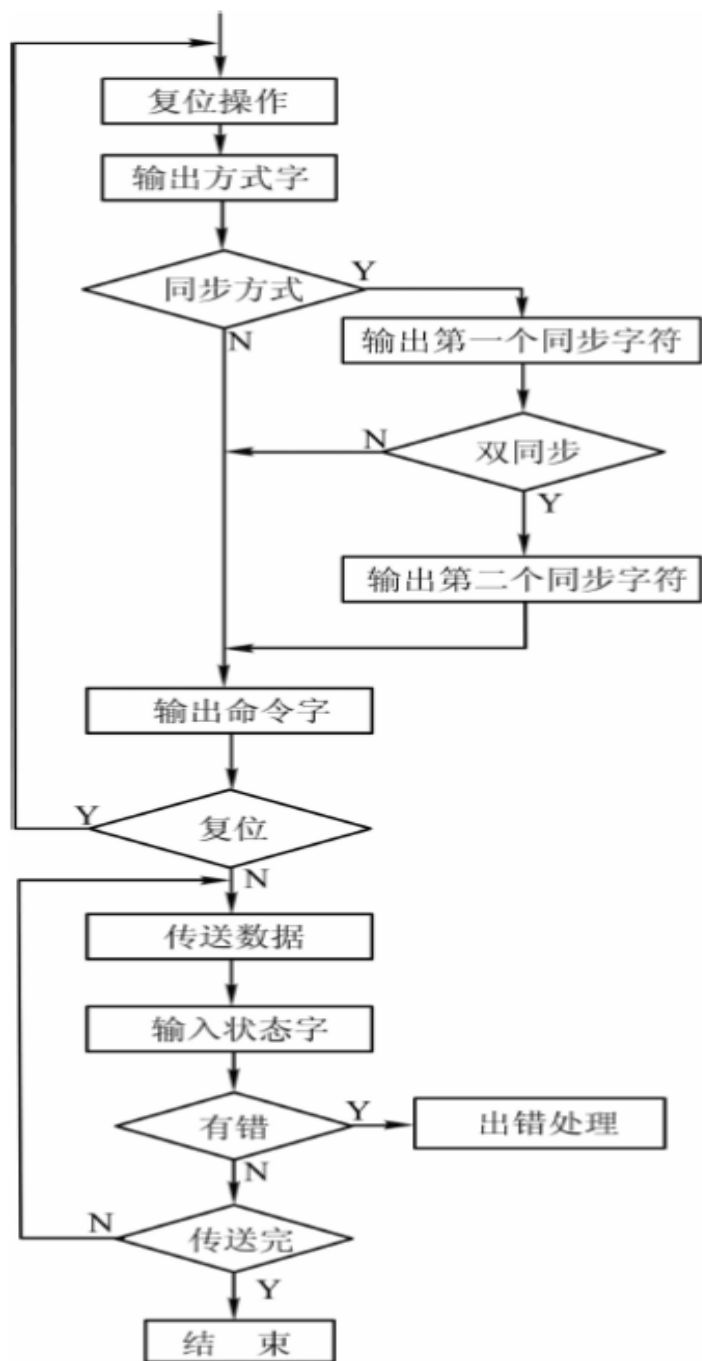
## 9.2.2 8251A的编程

- ◇ 使用时应对8251A进行初始化编程
- 为可靠复位，应先向控制口连续写入3个0，再写入1个复位字，然后写入方式字及命令字。
- 方式字用来确定工作方式，如规定同步/异步方式、传送的波特率、字符长度、奇偶校验等。
- 命令字控制它按规定的方式工作。如允许/禁止收发数据，启动搜索同步字符，迫使内部复位等。
- 对控制口进行一次写入操作后，要有写恢复时间，即延迟一点时间后才能再写入下个控制字。
- 用IN指令可随时读出状态字，了解工作状态。



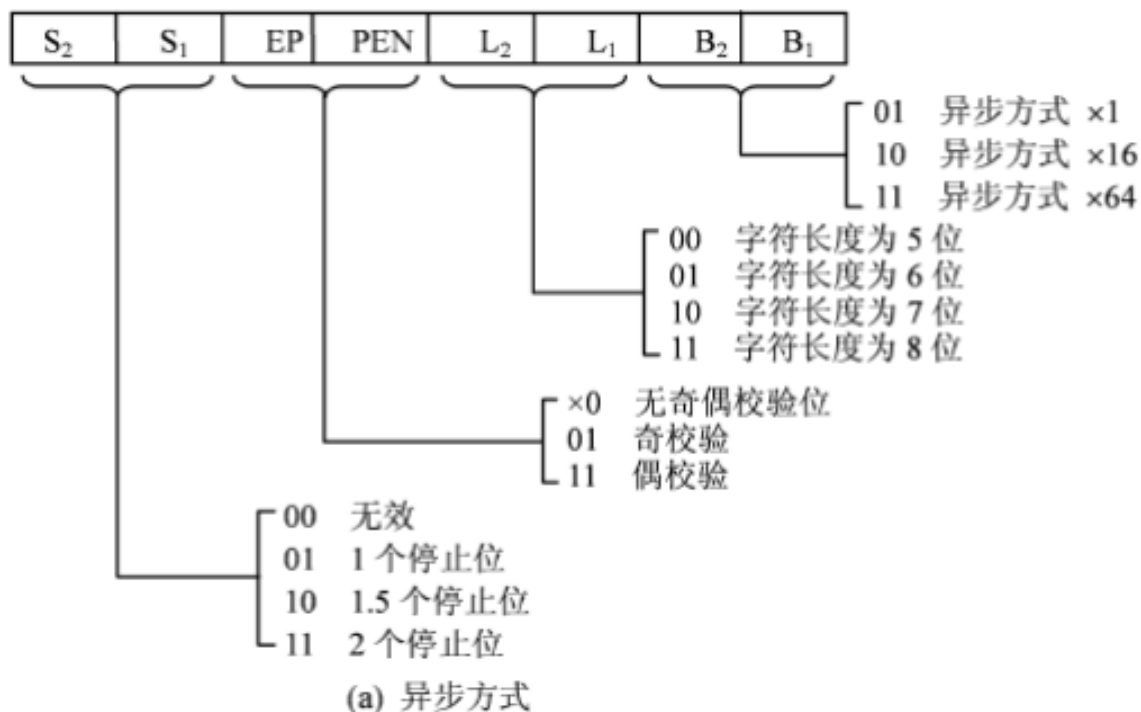
# 8251A的编程流程图

- 向控制口写入复位命令字。
- 写入方式字确定工作方式。
- 同步方式，方式字后需写入1或2个同步字符，随后写入命令字。
- 异步方式，写入方式字后，接着写入命令字。
- 若要改变数据传送方式，须写入复位命令字，再重新写入新的方式字和命令字。
- 工作中，可用IN指令读状态字，检查是否有错等。
- 沿出错，可不断传送数据





## 2. 方式字、命令字和状态字的格式



### 1) 方式字

异步方式下:

➤ B<sub>2</sub>B<sub>1</sub>确定波特率系数

收发时钟频率 =  
收发波特率  
× 波特率系数

例: 收发时钟频率9600  
波特率系数×16, 则  
收发波特率为:

$$9600/16=600$$

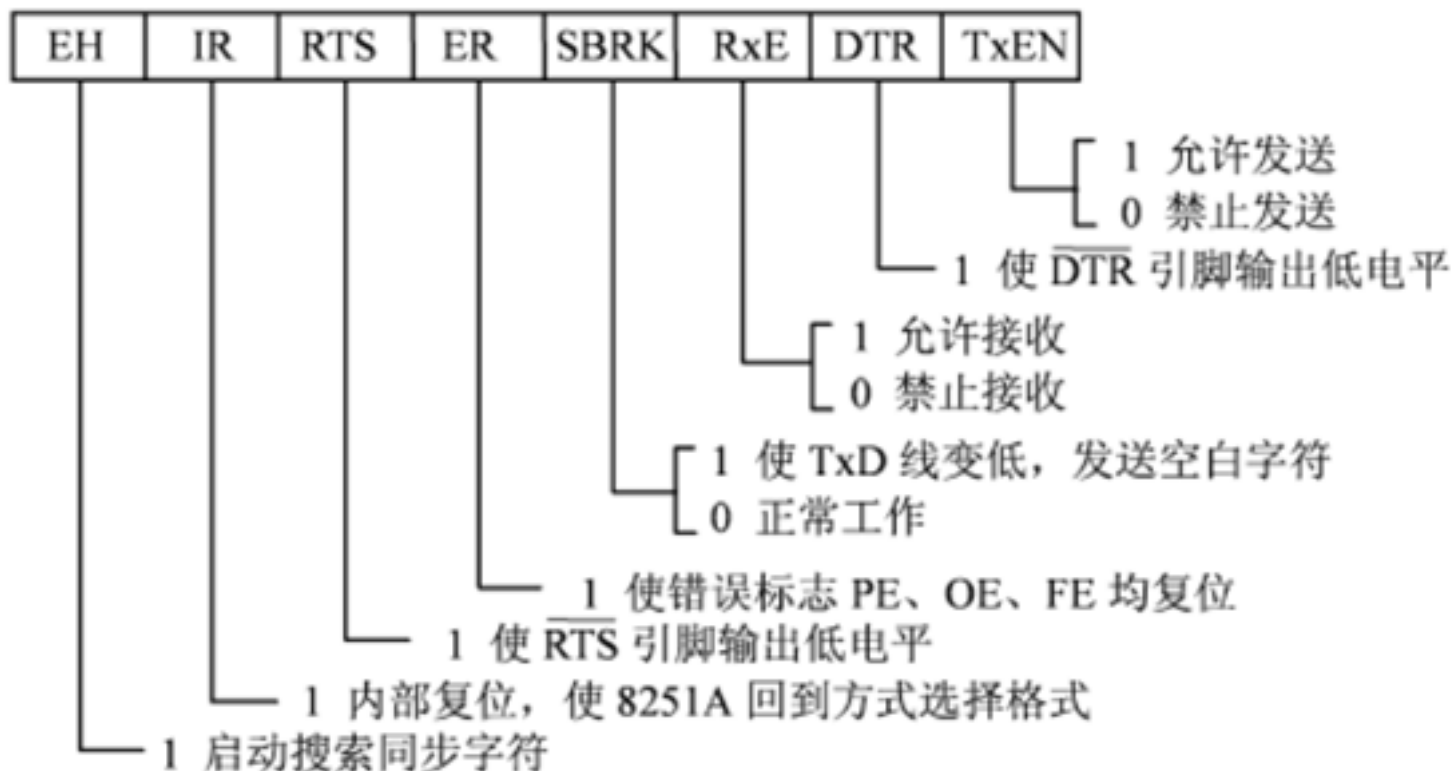
←其它见图



- ◆ 异步通信中，若帧数据格式主：字符长度8位，1位起始位，2位停止位，奇检验，波特率因子是16，期中308H8251的数据口，309H是命令状态口。
- ◆ 由其工作字是：
- ◆ 11011110=DEH
- ◆ MOV DX,309H
- ◆ MOV AL,0DEH
- ◆ OUT DX,AL



## 2) 命令字



各位的意义见图↑

- IR=1使8251A内部复位。例：复位字=0100 0000B=40H
- RxE =1，允许接收数据；TxEN =1，允许发送数据；
- ER=1，使各错误标志位清0；

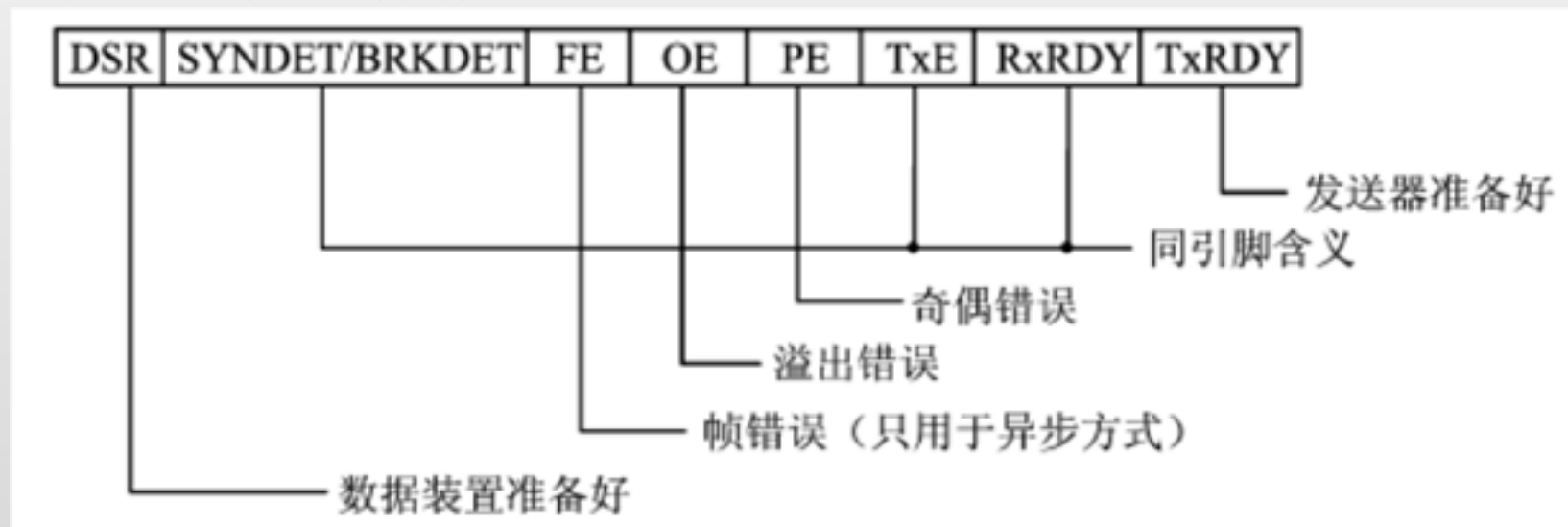


- ◆ 使8251内部复位，并且放接收，又允许发送。
- ◆ 程序段为：
- ◆ `MOV DX,309H` ; 8251命令口
- ◆ `MOV AL,01000000B`; 置D6=1，使内部复位
- ◆ `OUT DX,AL`
- ◆ `MOV AL,00000101B`; 置D0=1，D2=1，允许接收和发送
- ◆ `OUT DX,AL`
- ◆ 发送命令之前，一定要进行内部复位，使8251处于等待命令方式。



### 3) 状态字

- 作用：8251A执行命令进行数据传送后的状态字存放在状态寄存器中，CPU通过读出状态字进行分析和判断，以决定下一步的操作。



- TxRDY=1 发送数据缓冲器空，准备好接收数据。
- PE=1 奇偶校验错。
- OE=1 溢出错。
- FE=1 帧错误，仅用于异步方式。
- DSR=1，MODEM已准备好数据。
- 其余位的意义与引脚功能相同。



- ◇ 若查询8251A接收器是否准备好
- ◇ 程序段：
- ◇ L:MOV DX,309H
  - ◇ IN AL,DX
  - ◇ AND AL,02H
  - ◇ JZ L
  - ◇ MOV DX,308H
  - ◇ IN AL,DX



- ◇ 检查出错信息
- ◇ 程序段
- ◇ MOV DX,309H
- ◇ IN AL,DX
- ◇ TEST AL,38H; 检查 D5D4D3三位 (PE、OE、PE)
- ◇ JNZ ERROR



- ◆ 8251A的命令字与状态字
- ◆ 方式选择字：约定双方通信的方式，数据格式，传送速率等参数。
- ◆ 命令字：规定是发送数据，还是接收数据。
- ◆ 状态字：何时发/收，取决于状态字。
- ◆ 使用的顺序：
- ◆ 复位---->方式字---->命令字1---->命令字2.....





### 3. 8251A初始化编程举例

#### 1) 异步方式初始化程序

##### 例9.1

若要求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。假设REVTIME是写恢复时间的延时宏指令，类似于一个子程序。试编写对8251A进行初始化的程序。

程序如下：

```
MOV    DX, 3F2H
```

； 控制口

```
MOV    AL, 00H
```

```
OUT    DX, AL
```

； 向控制口写入“0”

```
REVTIME
```

； 延时，等待写操作完成



OUT	DX, AL	; 向控制口写入第2个“0”
REVTIME		; 延时
OUT	DX, AL	; 向控制口写入第3个“0”
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: 波特率系数16, 7个数据位, 1个停止位, 偶校验		
OUT	DX, AL	; 写入方式字
REVTIME		; 延时
MOV	AL, 00010101B	
; 命令字: 允许接收发送数据, 清错误标志		
OUT	DX, AL	; 写入命令字



## 2) 同步方式初始化程序

### 例9.2

设8251A的控制口地址为3F2H，写恢复延时程序仍用宏指令REVTIME，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，试编写8251A写入复位字以后的初始化程序。



...

； 先向控制口写入3个0，再送复位字40H

MOV DX, 3F2H ; 控制口

MOV AL, 00011000B

； 方式字: 双同步, 内同步, 奇校验, 7个数据位

OUT DX, AL ; 送方式字

REVTIME ; 延时

MOV AL, 16H ; 16H为同步字符

OUT DX, AL ; 送入第一个同步字符

REVTIME

OUT DX, AL ; 送入第二个同步字符

REVTIME

MOV AL, 10010101B

； 命令字: 启动搜索同步字符, 错误标志复位, 允许收发

OUT DX, AL



**9.2.1 8251A的内部结构和外部引脚**

**9.2.2 8251A的编程**

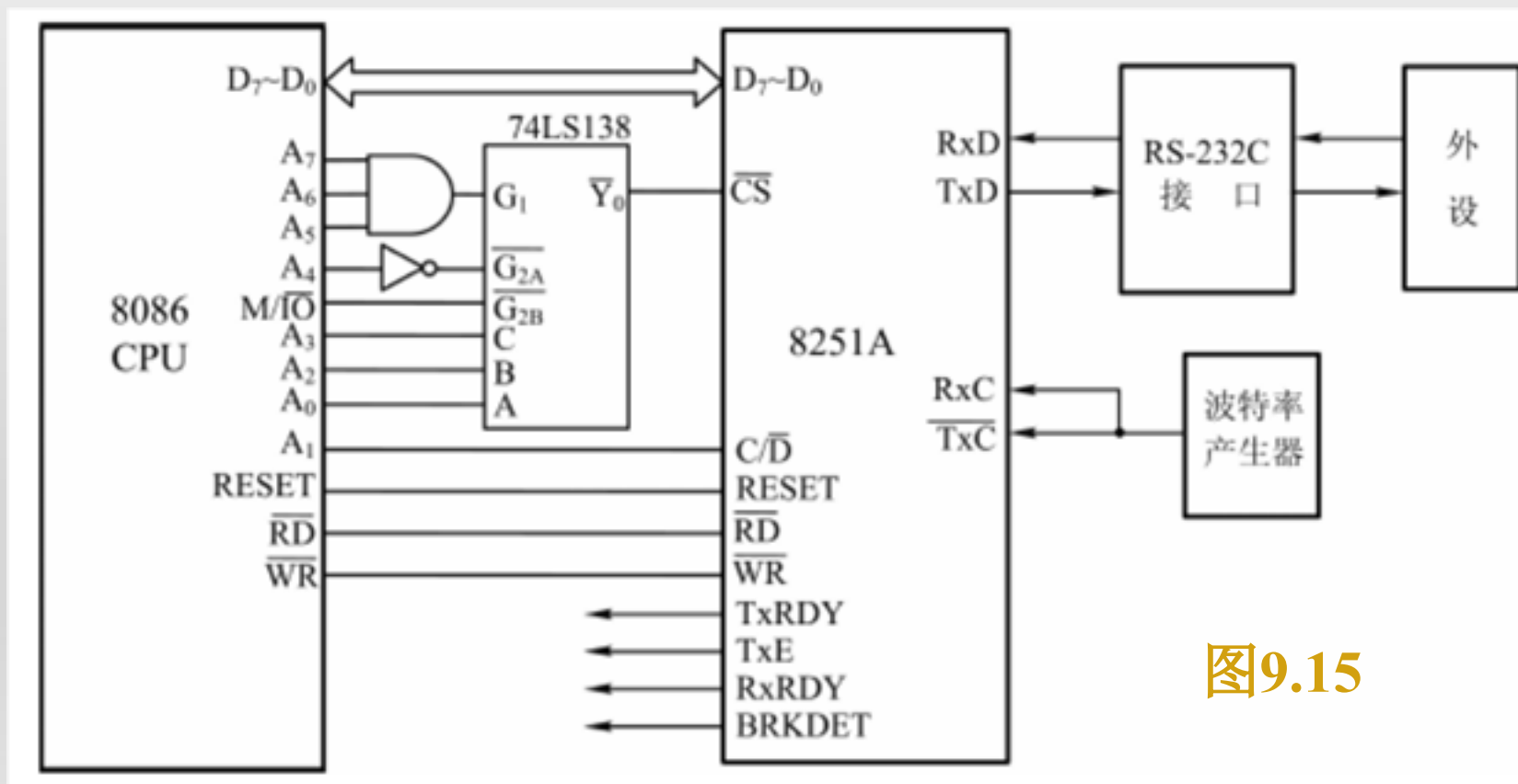
**9.2.3 8251A应用举例**



## 9.2.3 8251A应用举例

### 1. 8251A与CPU及外设的连接

- 用8251A构成的串行口，可与CRT显示器、鼠标等串口外设相连，工作于异步方式，采用RS-232C串行接口标准通信，不用MODEM信号。连线示意如图9.15：



# 1) 8251A与CPU之间的连线

- $\overline{RD}$   $\overline{WR}$ 、CLK和RESET信号,可直接与CPU的相应引脚相连。
- 数据线 $D_7 \sim D_0$ 与CPU的低8位数据总线 $D_7 \sim D_0$ 相连。
- $C/\overline{D}$ 与系统地址总线 $A_1$ 相连。
- 地址总线的其余位参与I/O译码,形成片选信号,与8251A的 $\overline{CS}$ 相连。
- $TxRDY$ 、 $TxE$ 、 $RxRDY$ 和BRKDET都是输出信号,是CPU与8251A间的收发联络信号。
- 查询方式下,它们被用作状态信号;
- 中断方式下,  $TxRDY$ 和 $RxRDY$ 可作为向CPU请求发送或接收数据的中断请求信号。



## 2) 8251A与外设之间的连线

- RxD接收外设送来的串行数据，TxD向外设发送串行数据，都是TTL电平。
- 采用RS-232C串行接口通信时，需用专门的电平变换电路如MAX 233实现电平转换：
  - 发送数据时，将TTL电平的TxD信号，转换为RS-232C电平后发送出去；
  - 接收数据时，把接收来的RS-232C电平的RxD信号，转换为8251A能接受的TTL电平。
- 发送时钟输入端  $\overline{\text{TxC}}$  和接收时钟输入端  $\overline{\text{RxC}}$  可以连在一起，由波特率产生器提供相同的时钟脉冲信号。





### 3) 端口地址译码电路

- 与8位CPU连接，不用考虑奇偶地址问题
  - $\overline{C/\overline{D}}$  接地址线  $A_0$ :  $A_0=0$ 选中数据口,  $A_0=1$ 选控制口。  
比如: 数据口为F0H, 控制口为F1H。
- 与16位CPU连接，要考虑奇偶地址问题
  - 若使用低8位数据总线，端口地址总是偶地址， $A_0$ 参与I/O地址译码， $A_0=0$ 才选中8251A。应将 $A_1$ 与  $\overline{C/\overline{D}}$ 端相连， $A_1=1$ 选中控制口， $A_1=0$ 选中数据口。
  - 若使用高8位数据总线，则端口地址要求总是奇地址。

对于图9.15电路，使用的是低8位数据总线:

- $A_7A_6A_5A_4=1111$ , 即  $G_1=1$ ,  $\overline{G_{2A}}=0$
- I/O操作时  $\overline{M/\overline{IO}}=0$ , 即  $\overline{G_{2B}}=0$
- $A_1=1$ 选控制口,  $A_1=0$ 选数据口
- $A_3A_2A_0$ 译码形成端口地址: F0H数据口, F2H控制口

