



7.3 异步串行通讯

7.3.1 串行通讯中基本概念

1. 计算机和外部通讯形式

串行：每次传送1bit 数据，一个字节数据至少需要传送8次，一般是使用于计算机系统间。

并行：一次能传送的数据位为多位，使用系统内芯片间。

2.异步和同步串行通讯

- 1) 同步串行通讯：数据开始传送前用同步字符串（1-2个字节）来指示，并由同一时钟实现发送和接收端同步。

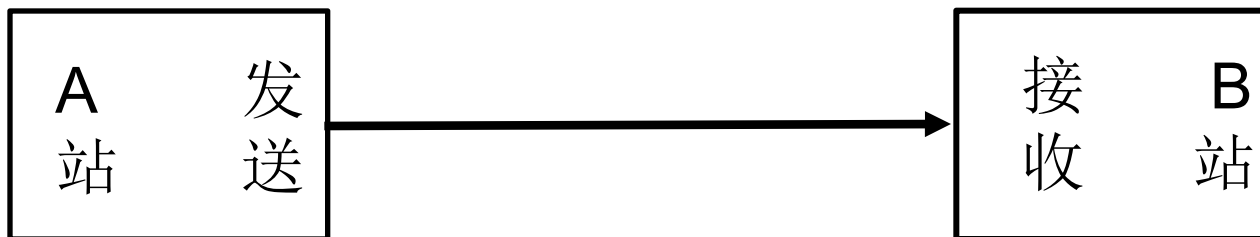
如 SPI, I²C

- 2) 异步串行通讯：传送字符（数据）的时间间隔是任意的，只需要在每个字符的前面和后面加上标志位就可以，发送和接收端的时钟可以是不同步。

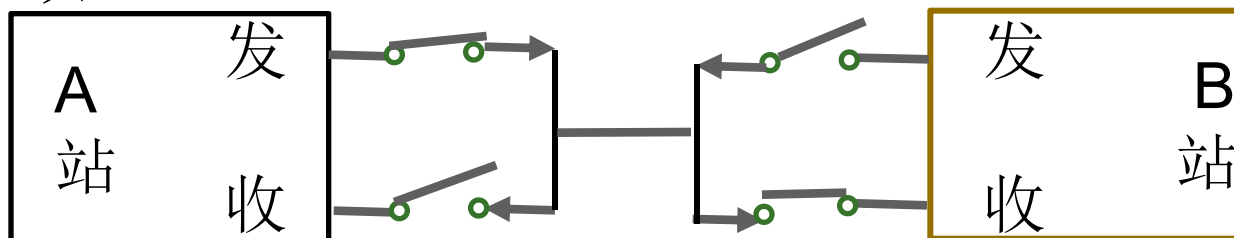
如 485, 232

3. 通讯的信道和方向（制式）

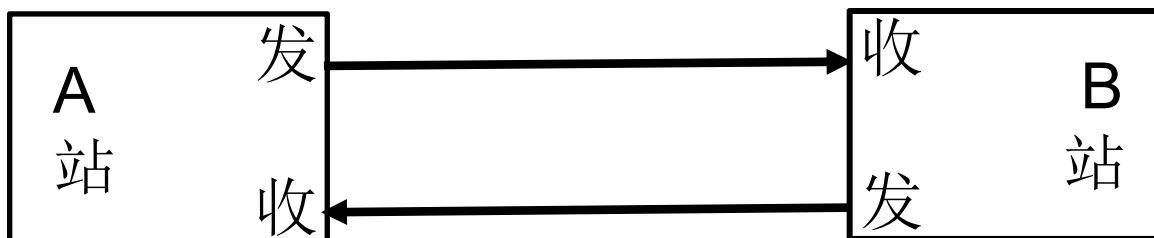
1) 单工



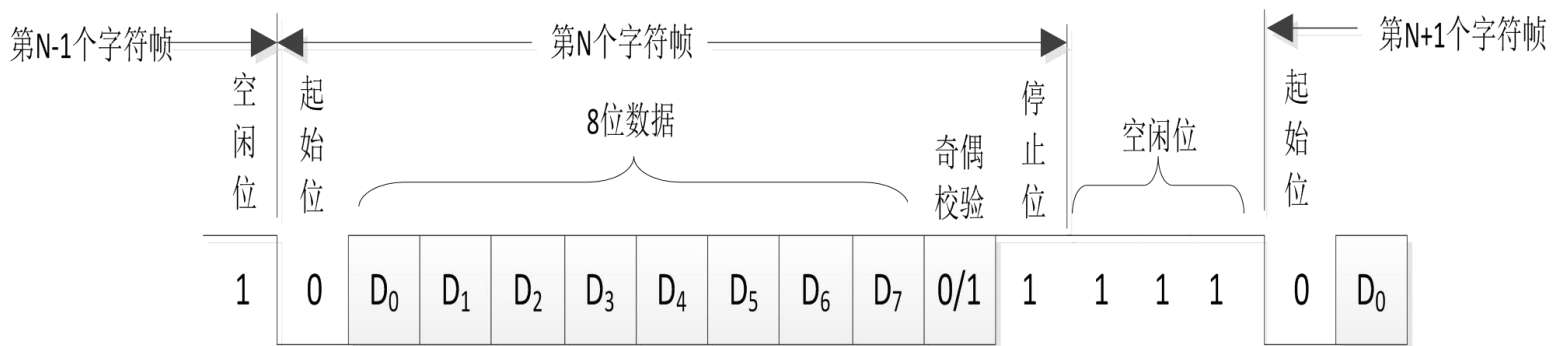
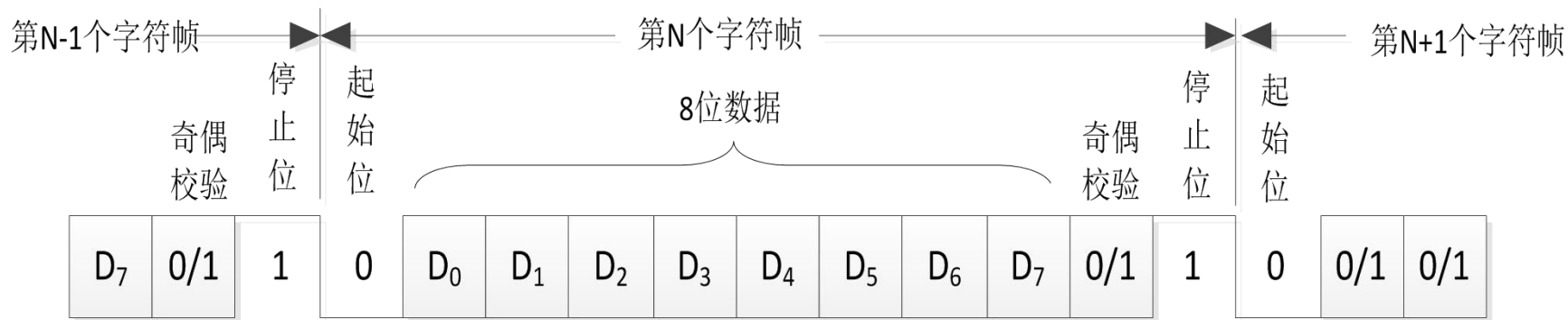
2) 半双工

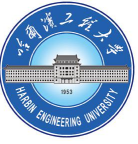


3) 全双工



4.异步串行通讯格式





5.波特率

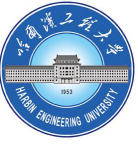
单位时间内传送数据位的速率，位/秒（bit/s）

如： 9600bps

11520bps

例1 计算机A和计算机B进行串行异步通讯，波特率为9600bps，数据帧格式11位：1位起始，8位数据位，1位校验位，1位停止位，问计算机A和计算机B传送10个字节数据至少需要多少时间t？

$$t = (1+8+1+1) * 10 / 9600$$

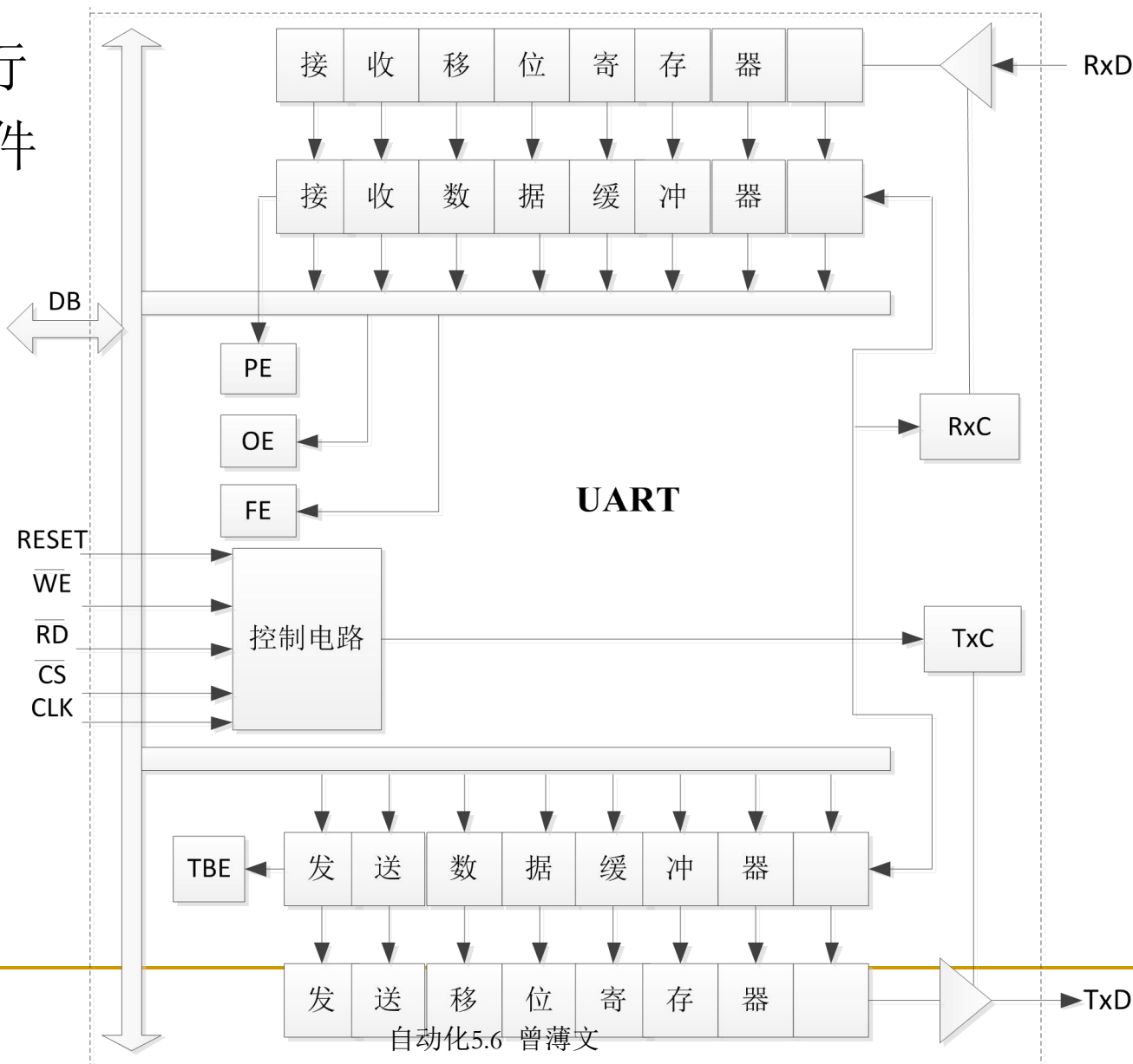


例2 计算机**A**和计算机**B**进行串行异步通讯，波特率为**9600bps**，数据帧格式**10**位：**1**位起始，**8**位数据位，**0**位校验位，**1**位停止位；当**A**方发送给**B****10**个字节数据，**B**要应答一个字节数据，问计算机**A**给计算机**B**传送**100**个字节数据至少需要多少时间**t**？

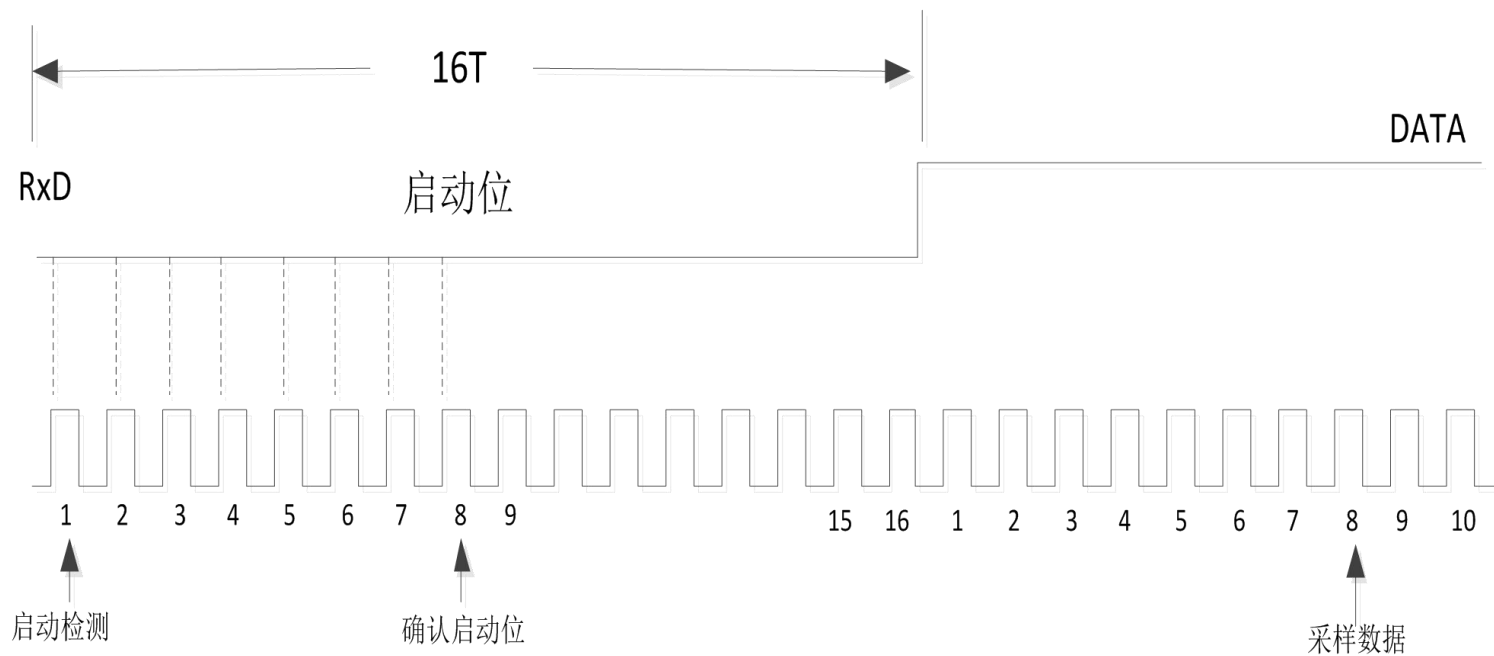
$$t = (1+8+1) * (100+10) * (1/9600)$$

7.3.2 串行通讯硬件实现原理

1.异步串行接收器硬件框图

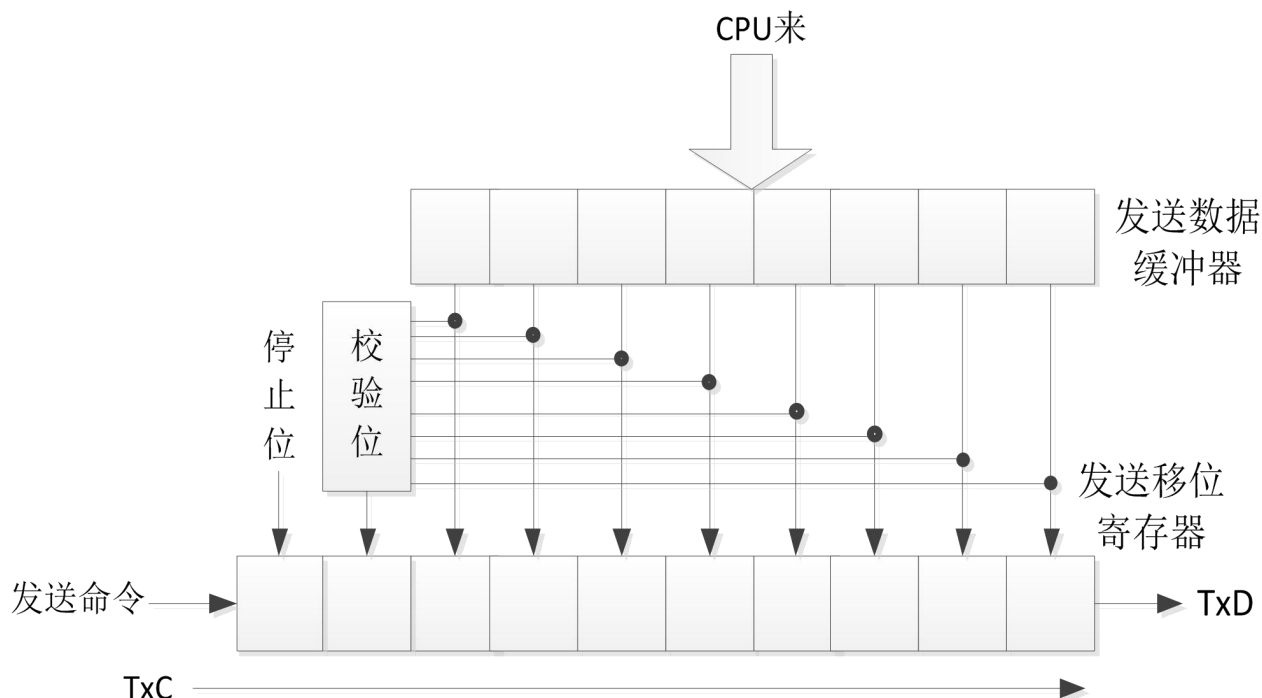


2.UART对RxD数据的采样

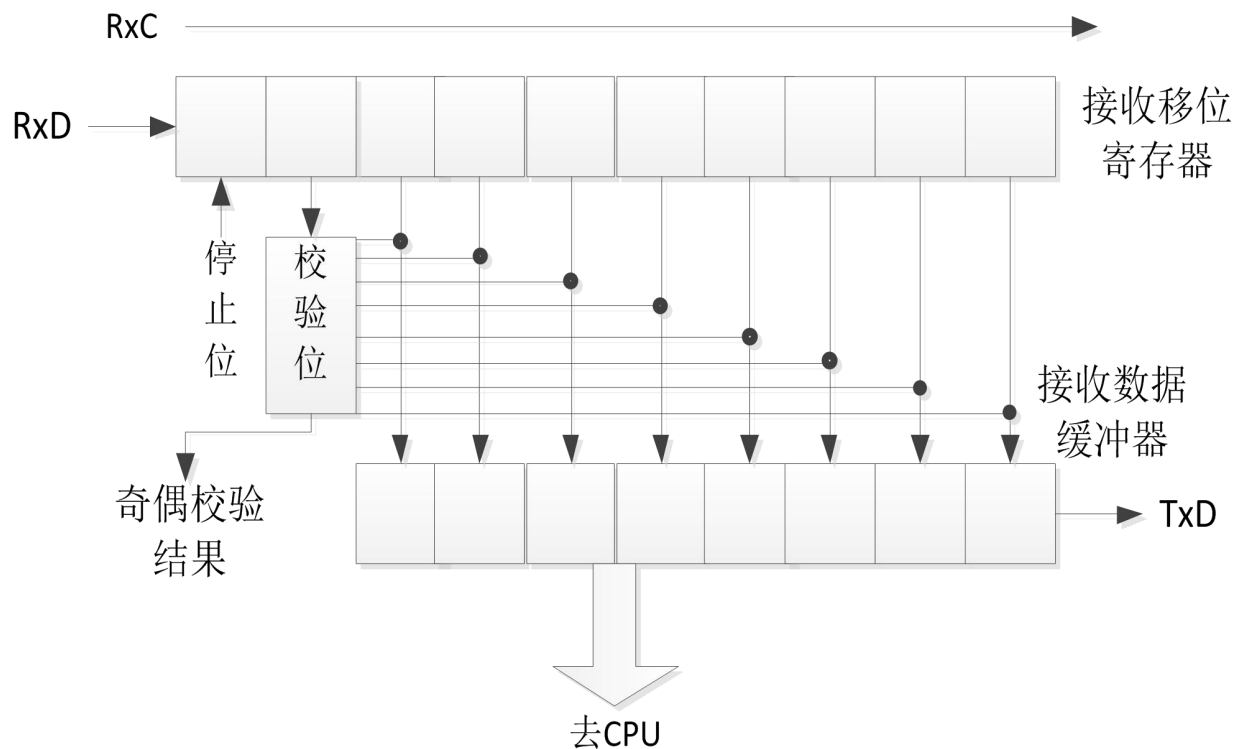


3.数据的错误校验

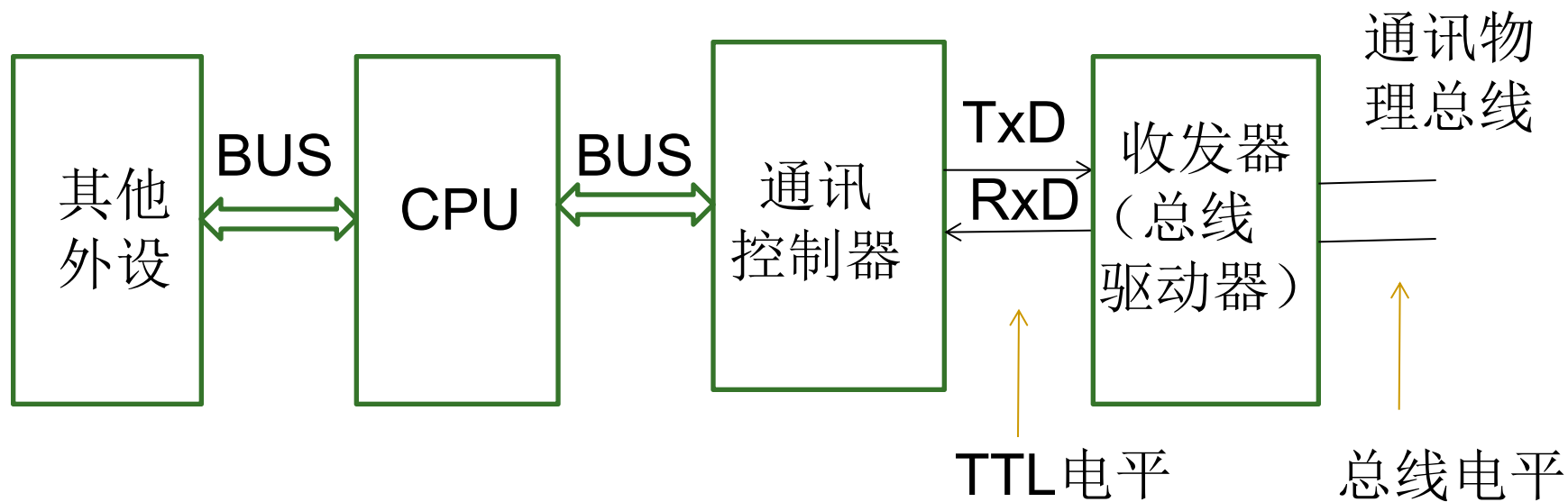
1) 发送端奇偶添加电路



2) 接收端奇偶校验电路



4. 串行通讯接口



5.串行通讯总线的形式

异步串行通讯电气接口

1) 232总线

2) 485总线

422总线

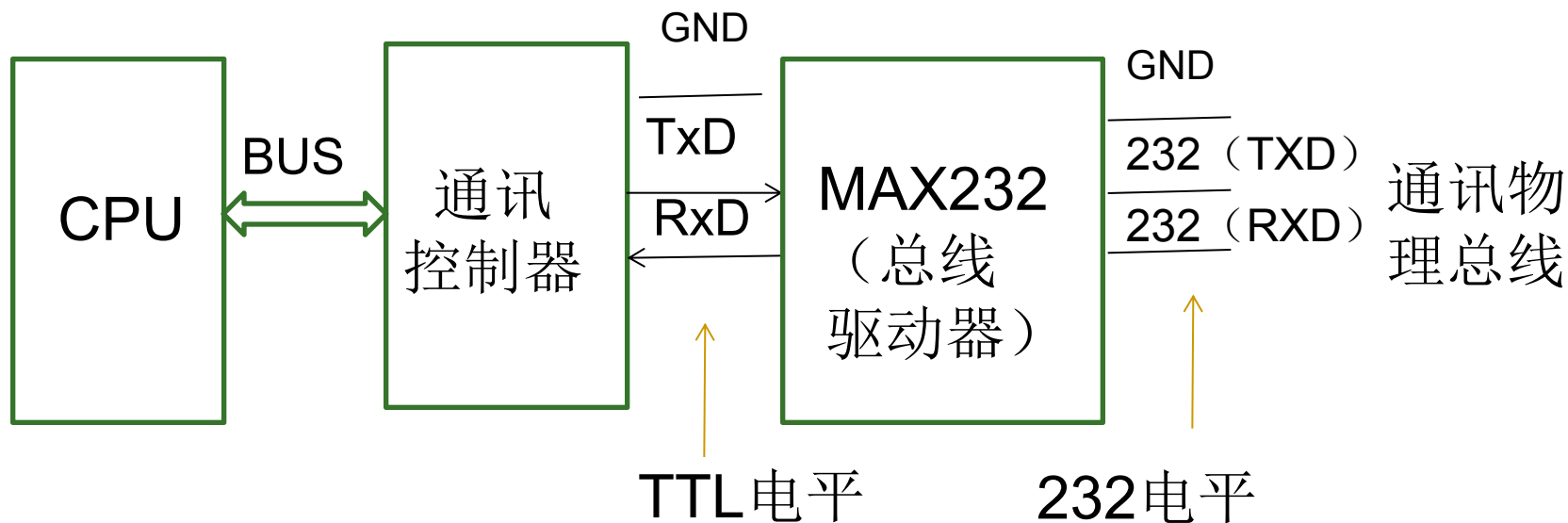
以上二者可共用通讯控制器，通用软件，只是物理层的接口（电气）不一样。

其他串行通讯总线

3) CAN总线

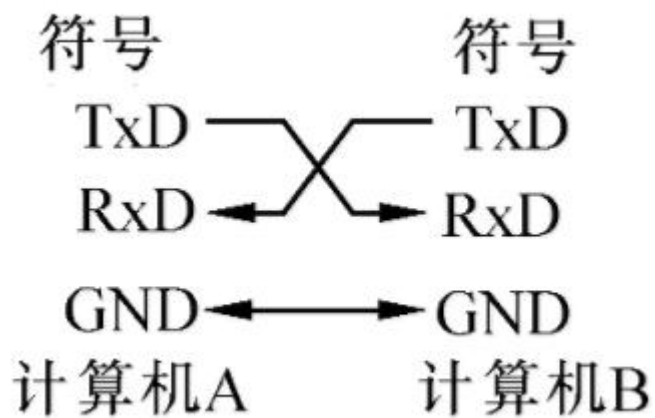
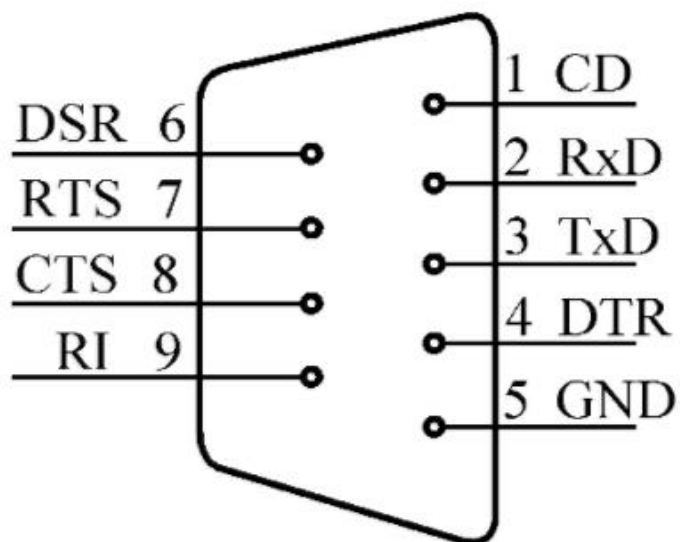
4) 以太网

■ 232总线

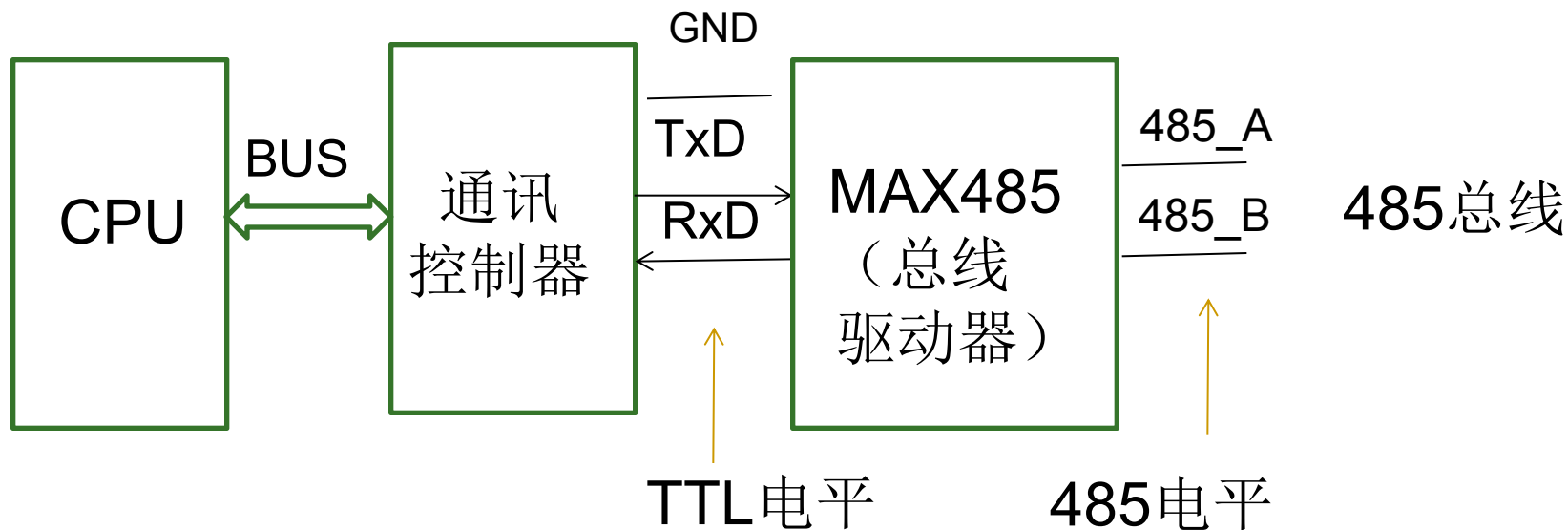


232总线 -5~-15V 表示逻辑 ‘1’
 +5~+15V 表示逻辑 ‘0’

■ PC上 232总线

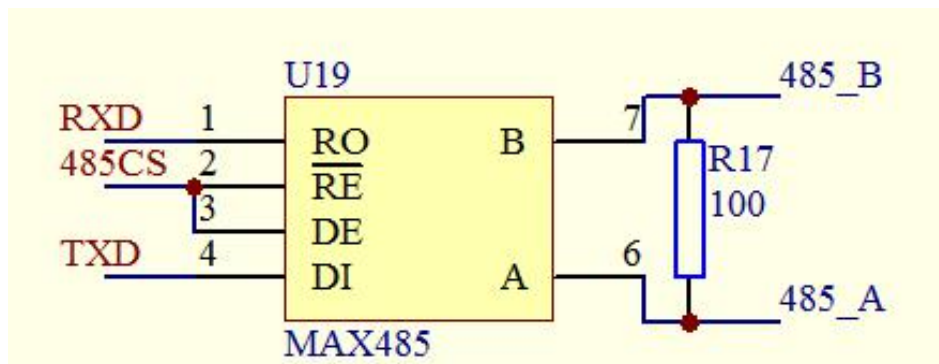


■ 485总线

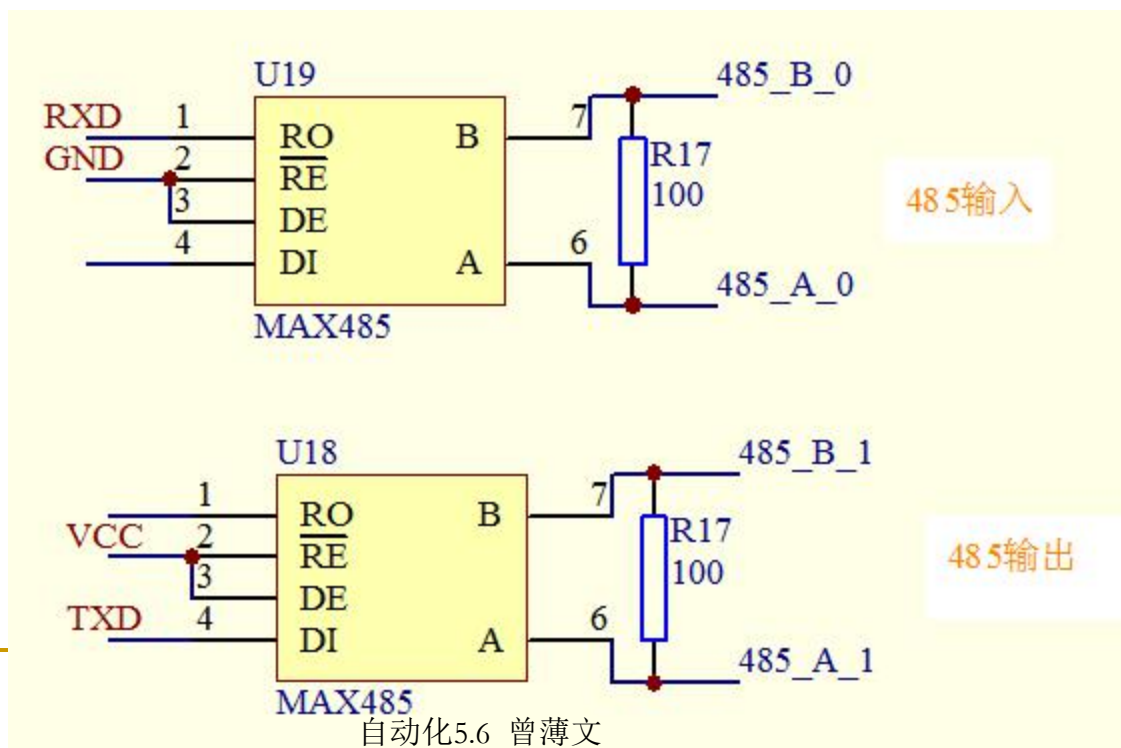


485为差分电平，485_A为'+'端，485_B为 '-' 端，较232抗共模干扰能力强通讯距离更远

■ 485总线（半双工）

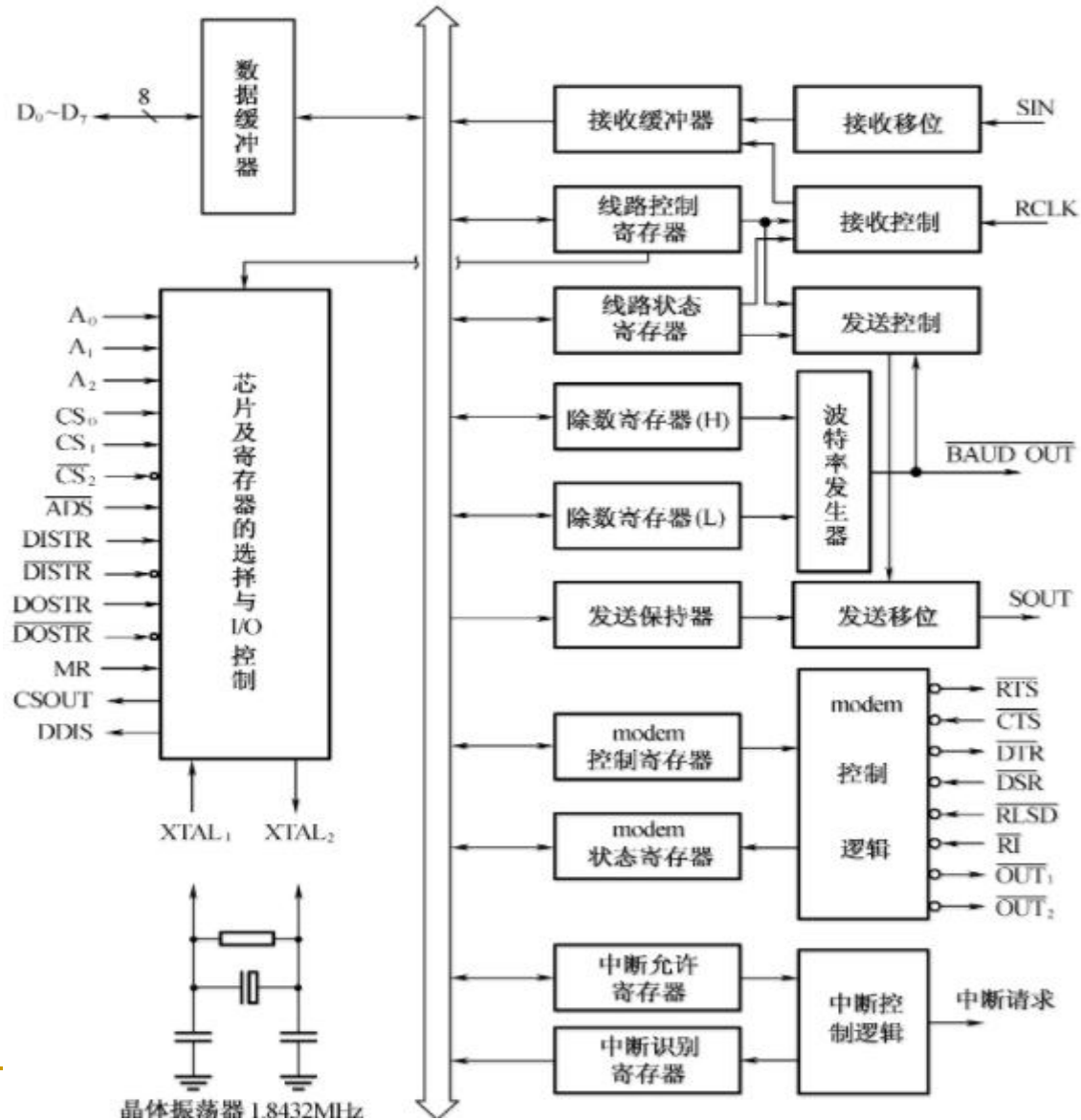


■ 422总线（全双工）

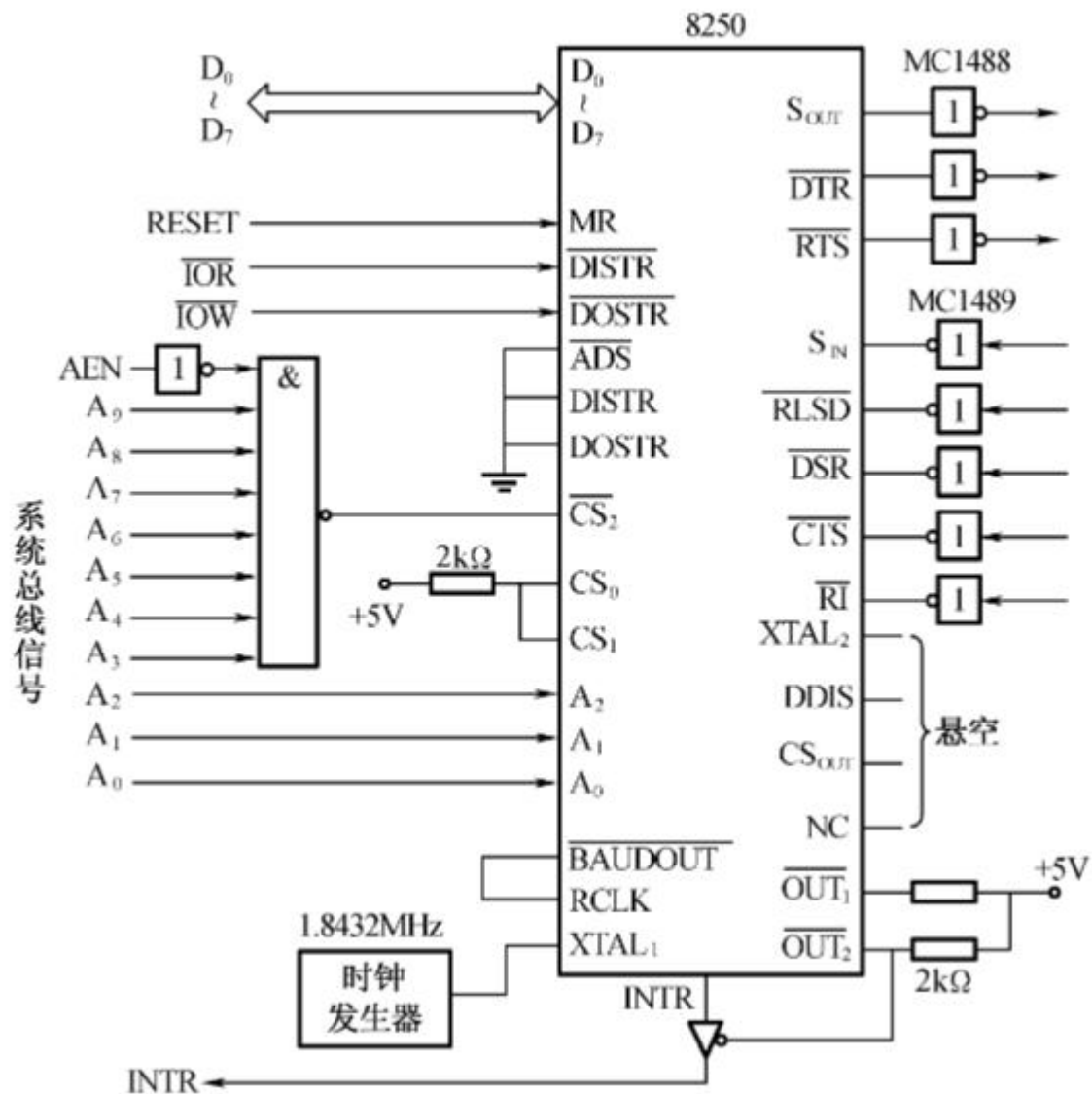


7.3.3 串行通讯接口芯片8250

1.内部结构

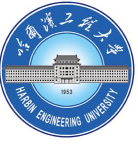


2. 外部管脚连接



3.8250主要寄存器

A2	A1	A0	DLAB	地址	寄存器
0	0	0	0	3F8H	发送保持寄存器THR（写） 接收缓冲寄存器RBR（读）
0	0	0	1	3F8H	除数锁存寄存器（低8位）DLL
0	0	1	1	3F9H	除数锁存寄存器（高8位）DLH
0	0	1	0	3F9H	中断允许寄存器IER
0	1	0	×	3FAH	中断识别寄存器IIR
0	1	1	×	3FBH	通信控制寄存器LCR
1	0	0	×	3FCH	MODEM控制寄存器MCR
1	0	1	×	3FDH	通信线路状态寄存器LSR
1	1	0	×	3FEH	MODEM状态寄存器MSR
1	1	1	×	3FFH	无效



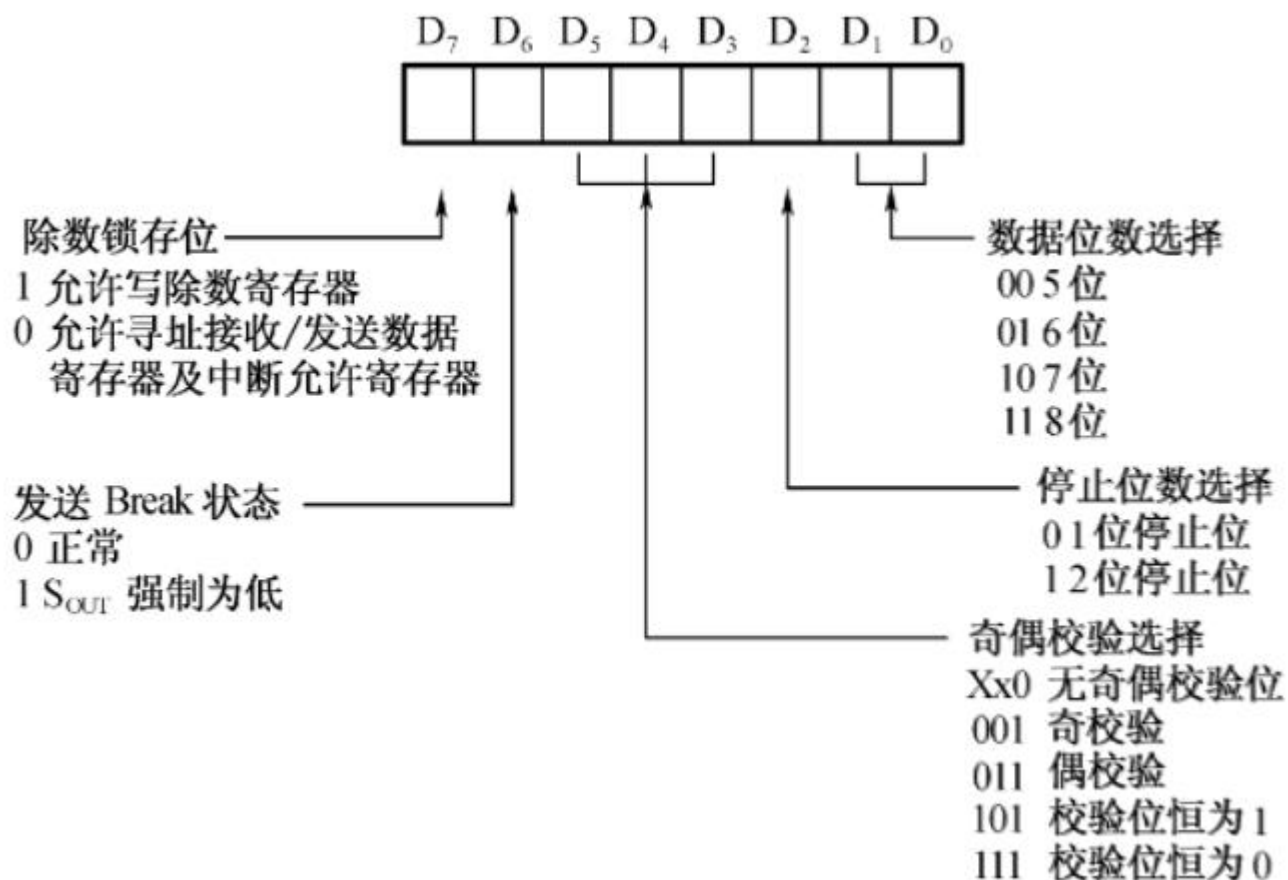
4.寄存器功能介绍

- 波特率设置（除数锁存寄存器3F8H ,3F9H）

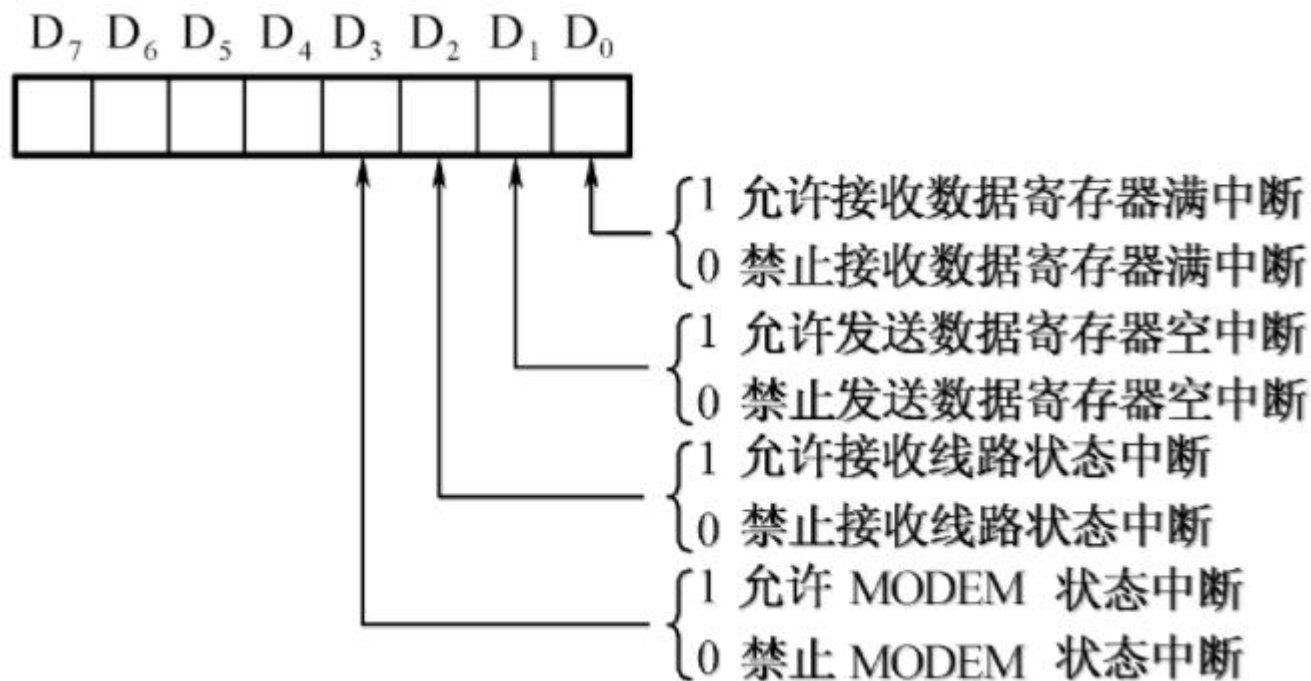
$$\text{除数}(N)=f/(F*16)$$

； f 为时钟， F 为要设置的波特率

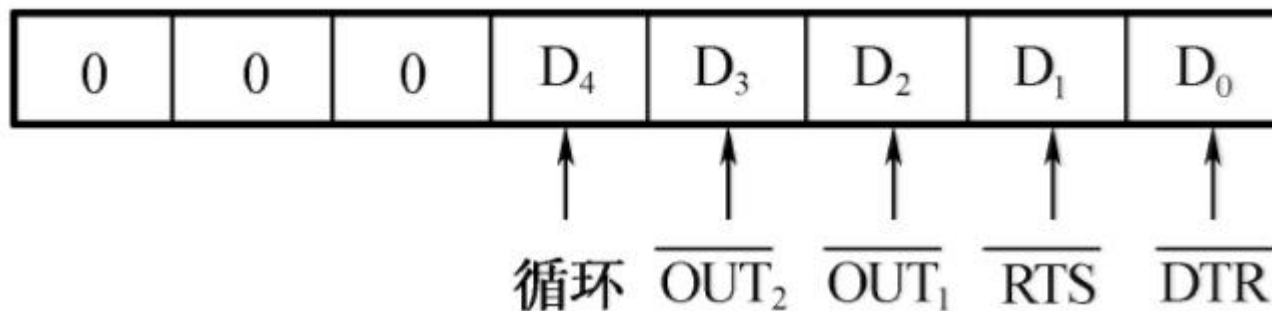
■ 通讯格式设置（通讯控制寄存器3FBH）



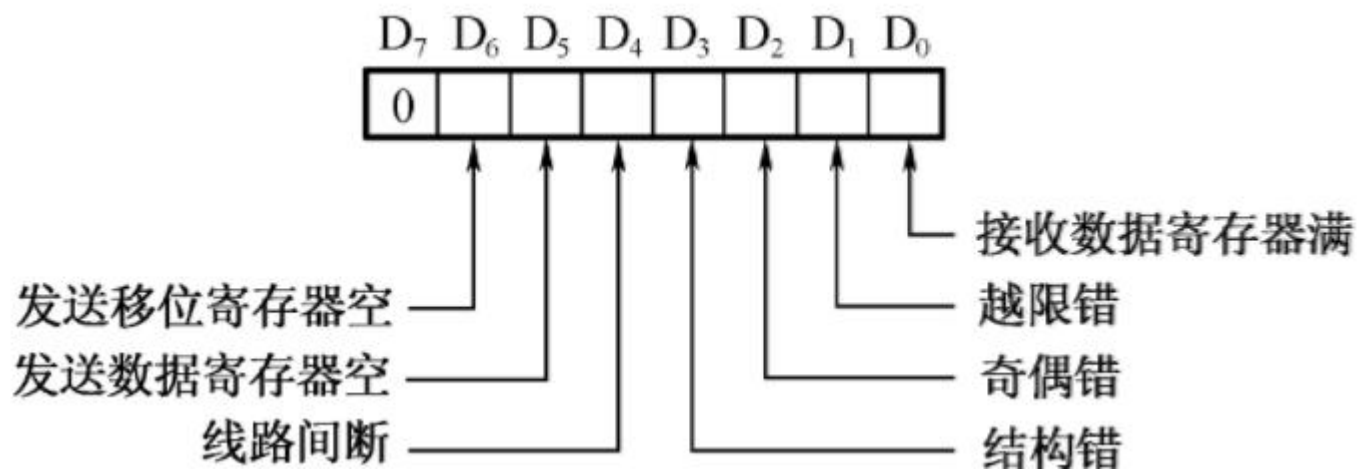
■ 中断允许寄存器设置(3F9H)



■ MORDEM控制寄存器设置(3FCH)



■ 通讯状态寄存器(3FDH)

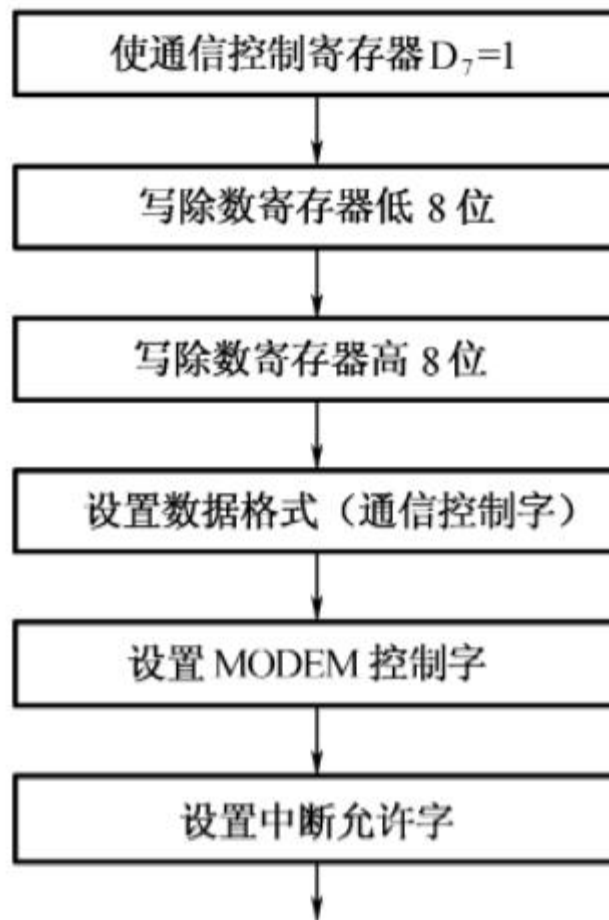


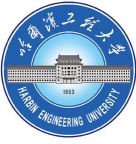
通讯状态寄存器不需要初始化，在工作过程中读取获得8250的工作状态，用于收发数据。

5.收发数程序设计

1) 初始化程序

初始化流程如图





- 设置波特率（波特率9600 ,波特率除数因子N=12，输入时钟1.8432MHz）

```
MOV DX, 3FBH
```

```
MOV AL, 80H ;设置波特率
```

```
OUT DX, AL
```

```
MOV DX, 3F8H
```

```
MOV AL, 12
```

```
OUT DX, AL
```

```
INC DX ;3F9H
```

```
MOV AL, 0
```

```
OUT DX, AL
```

- 设置串行通讯数据格式，数据位为8位，1位停止位，奇校验

```
MOV DX, 3FBH
```

```
MOV AL, 0BH
```

```
OUT DX, AL
```

设置中断:

a) MORDEM控制器 OUT2设置

无中断(OUT2=1):

```
MOV AL,3
```

```
MOV DX,3FCH
```

```
OUT DX,AL
```

有中断 (OUT2=0) :

```
MOV AL,0BH
```

```
MOV DX,3FCH
```

```
OUT DX,AL
```

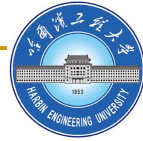
b) 设置中断允许寄存器

接收中断:

```
MOV AL,01 ;
```

```
MOV DX,3F9H
```

```
OUT DX,AL
```



2) 收发数据程序

(1) 查询发送数据

MOV DX, 3FDH ; 状态寄存器

WAITSEND:

IN AL , DX

TEST AL , 20H ; 状态寄存器D5是否为1

JZ WAITSEND ; 为0没有数, 继续查询等待

MOV DX, 3F8H ; 发送数据保持寄存器

MOV AL, DAT_SEND ; DATA_SEND待发数据

MOV DX, AL ; 启动发送

; 通过状态寄存器的D5判断是否发送结束

(2) 查询接收数据

```
MOV DX, 3FDH
```

```
WAIT_RECEIVE:
```

```
IN AL , DX
```

```
TEST AL , 1EH ; 状态寄存器D1-D4是否为1
```

```
JNZ ERR_DEAL ; 跳转错误处理
```

```
TEST AL,01H ; 状态寄存D0表示是否有新数
```

```
JZ WAIT_RECEIVE
```

```
MOV DX,3F8H ;
```

```
IN AL, DX
```

(3) 中断接收数据

在主程序首先建立数据接收程序的中断向量表；
开对应中断。中断服务程序示意如下所示：

SCI_REV:

。 。 。 ; 压栈保护

MOV DX, 3FDH

IN AL, DX

TEST AL, 1EH ; 状态寄存器D1-D4是否为1

JNZ ERR_DEAL ; 跳转错误处理

MOV DX, 3F8H ; 接收缓冲寄存器器

IN AL, DX

JMP OUTT

ERR_DEAL: . . .

. . .

;

错误处理程序

OUTT:

. . .

;

发EOI

. . .

;

出栈

IRET