

第六章 串行通信及接口芯片8251A

6.1 串行接口和串行通信

6.2 可编程串行通信接口8251A

6.1 串行接口和串行通信

6.1.1 串行通信涉及的几个问题

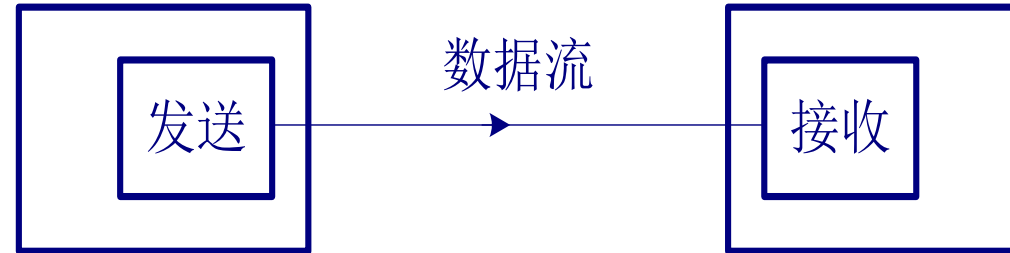
6.1.2 串行接口

6.1.1 串行通信涉及的几个问题

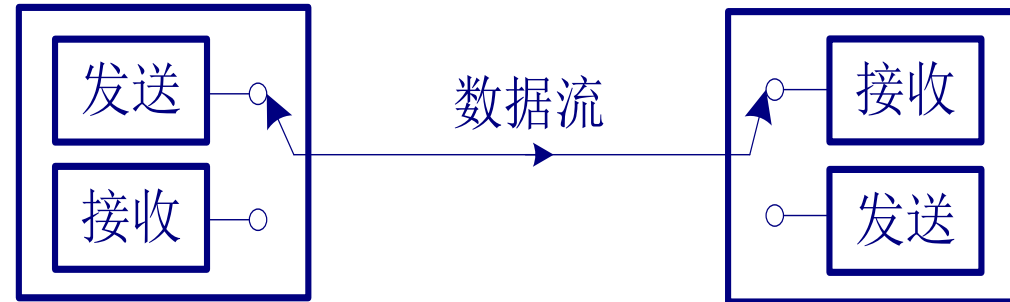
- ▶ 数据传送的方向
- ▶ 串行通信数据格式
- ▶ 数据的传输速率

1. 数据传送的方向

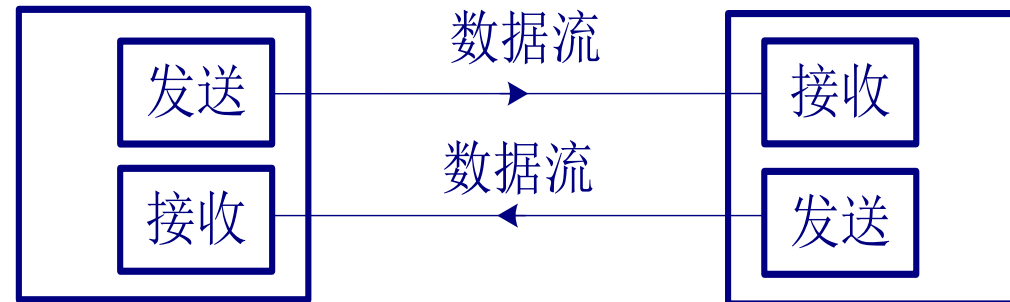
(1) 单工(Simplex): 通信是单向的。



(2) 半双工(Half duplex): 半双工指A能发信号给B, B也能发信号给A, 但这两个过程不能同时进行,A或B发送完后要切换到接受状态。典型的例子就是对讲机。



(3) 全双工(Duplex): 在A给B发信号的同时, B也可以给A发信号。典型的例子就是打电话。



2. 串行通信数据格式

在数据通信中，传输的对象是一系列的 0 和 1，这些 0、1 在不同的位置有不同的含义，这些含义都要事先约定好。

在通信中，两种最基本的串行通信方式：异步串行通信、同步串行通信。



不使用共同的时钟
和同步信号

使用同步信号

书上P199页

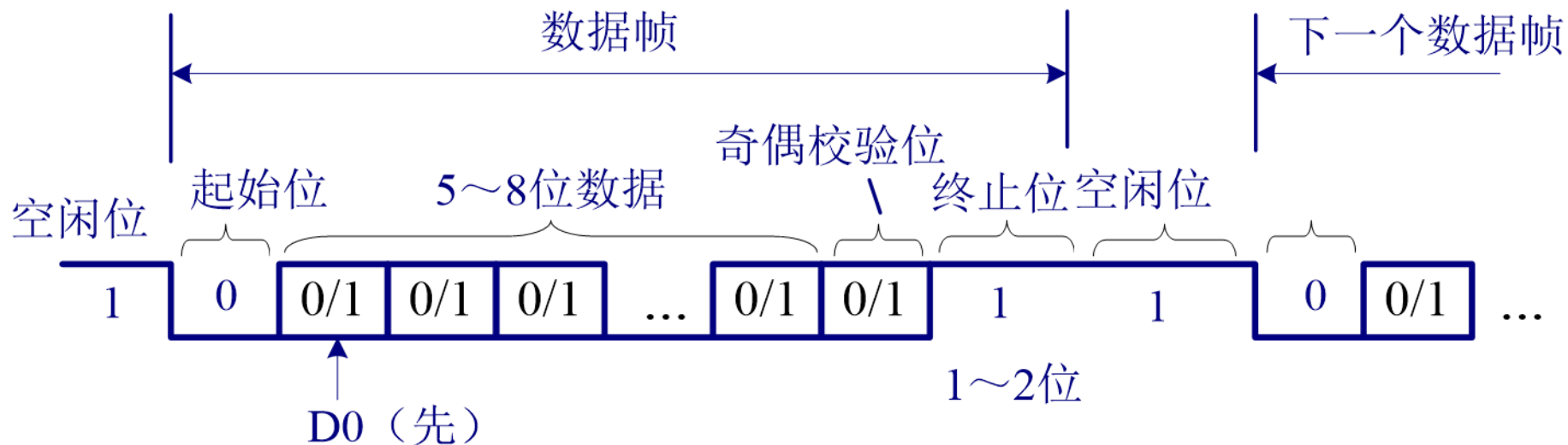
2. 串行通信数据格式——异步方式

所谓异步通信，是指数据传送以**数据帧**为单位，数据帧的传送是完全异步的，位与位之间的传送基本上是同步的。

异步串行通信的特点可以概括为：

- ①以**数据帧**为单位传送信息。
- ②相邻两数据帧间的**间隔**是任意长。
- ③异步方式特点简单的说就是：**字符间异步，字符内部各位同步。**

异步串行方式的数据格式



异步通讯的数据帧格式

异步串行通信的数据格式，每帧信息由4个部分组成：

- ①1位起始位，规定为低电平0；
- ②5~8位数据位，即要传送的有效信息；
- ③1位奇偶校验位；
- ④1~2位停止位，规定为高电平1。

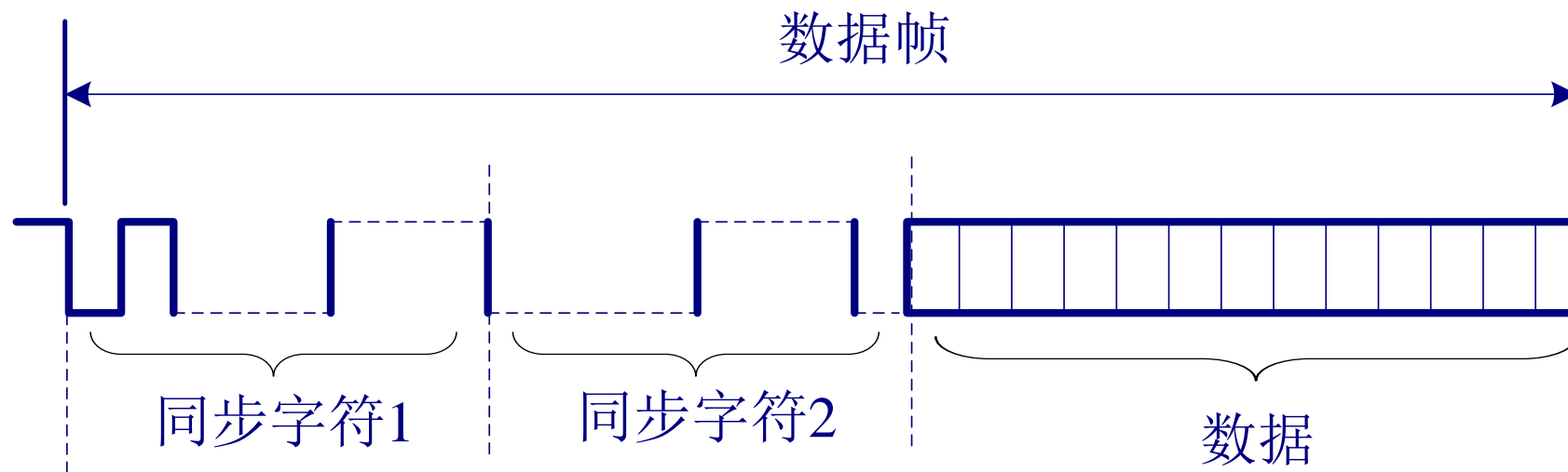
串行通信数据格式——同步方式

所谓同步通信，是指数据传送是以数据块（一组字符）为单位，字符与字符之间、字符内部的位与位之间都同步。

同步串行通信的特点可以概括为：

- ①以**数据块**为单位传送信息。
- ②在一个**数据块内**，字符与字符间**无间隔**。
- ③因为一次传输的数据块中包含的数据较多，所以接收时钟与发送时钟**严格同步**，通常要有**同步时钟**。

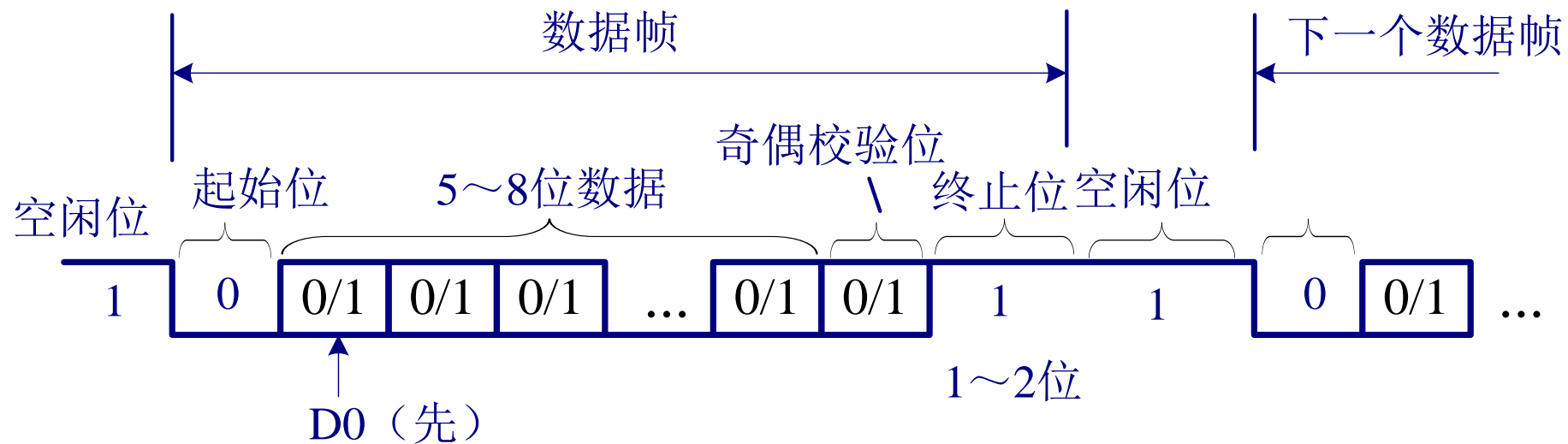
同步串行方式的数据格式



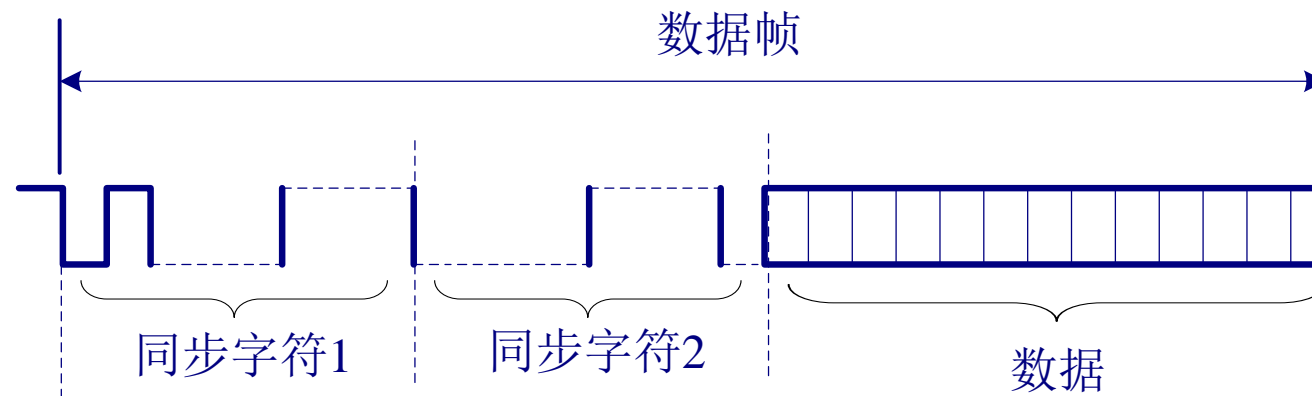
同步通讯的数据帧格式

同步串行通信的数据格式，每个数据块（信息帧）由2个部分组成：

- ① **1个或2个同步字符**作为一个数据块(信息帧)的起始标志；
- ② **n个连续传送的数据**



异步通讯的数据帧格式



同步通讯的数据帧格式

3. 传输速率

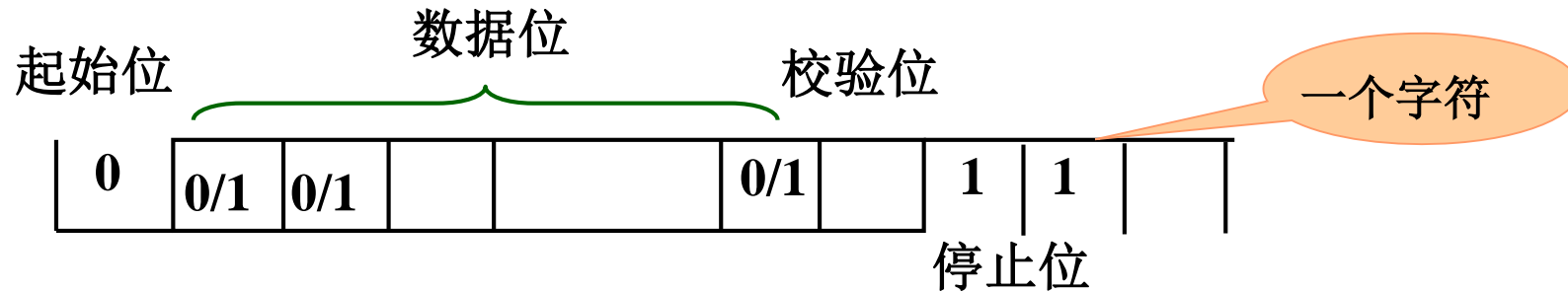
波特率： 单位时间传送的位数，单位bps(bit/s)。

波特率因子K： 每BIT占用的时钟周期数。

$K = \text{接收或发送时钟频率} / \text{比特率}$ ， 可取
1, 16, 32, 64

例1：一个异步串行发送器，发送具有8位数据位的字符，在系统中使用一个奇偶校验位和两个停止位。若每秒发送100个字符，则其波特率为多少？

格式



$$100 * (1 + 8 + 1 + 2) = 1200 \text{ bps}$$

例2：一个异步串行发送器，发送具有7位数据位的字符，传送波特率为1800，字符格式为：1个奇偶校验位，1个停止位，问，十秒钟内传送了多少个字符？

$$10 * 1800 / (1 + 7 + 1 + 1) = 1800$$

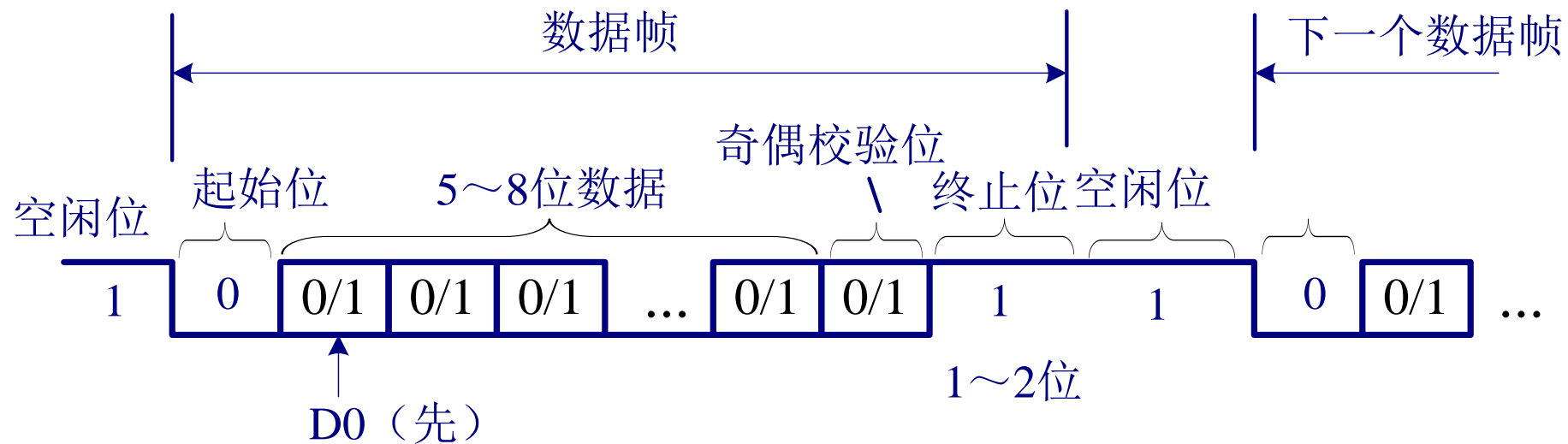
异步串行通信与同步串行通信的对比

异步串行通信	同步串行通信
双方使用各自的时钟	双方使用同一时钟
一帧以字符为单位(一个字符帧的长度取决于帧格式)	以数据块为单位(数据块长度可变)
传输效率低	效率高，速度快
应用于传输速率不高时，简单，应用较广	应用于大批量，高速率数据通信场合

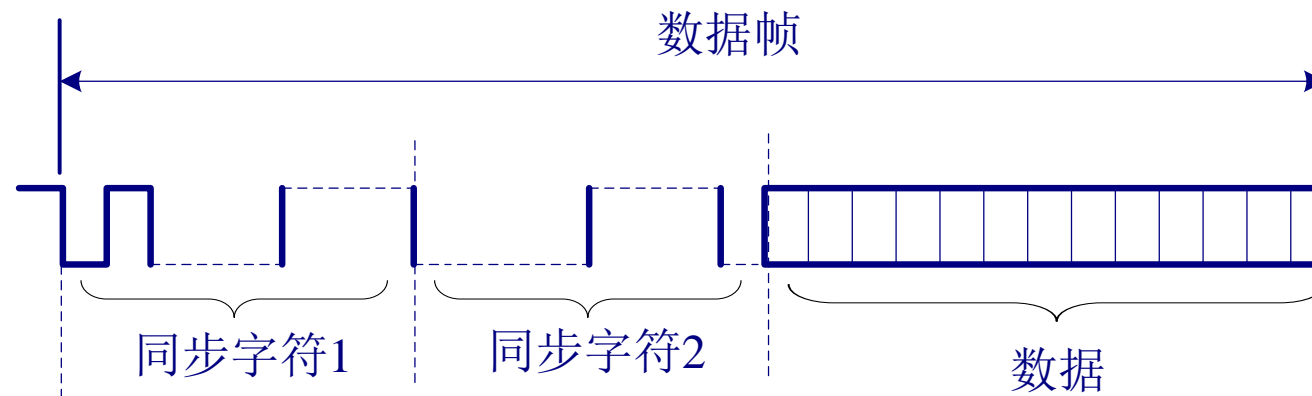
复习上节课

串行通信涉及的几个问题

- ▶ 数据传送的方向
- ▶ 串行通信数据格式
- ▶ 数据的传输速率



异步通讯的数据帧格式



同步通讯的数据帧格式

6.1.2 串行接口

串行接口部件4个主要寄存器：

控制寄存器

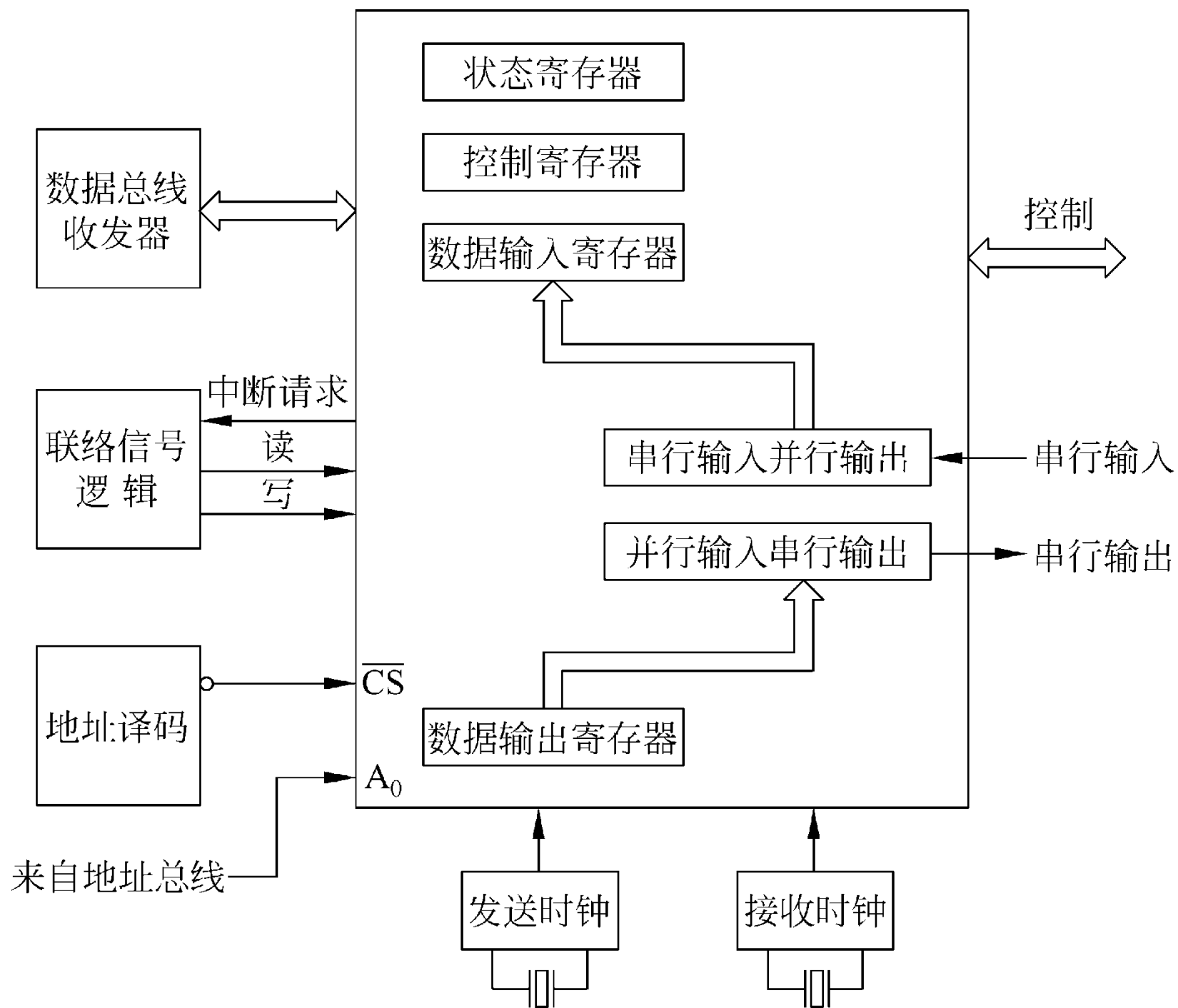
状态寄存器

数据输入寄存器

数据输出寄存器

注意：4个寄存器只用两个端口地址。

控制寄存器和数据输出寄存器是只写的，状态寄存器和数据输入寄存器是只读的，所以，用读信号和写信号区分这两组，再用1位来区分2个只读或2个只写寄存器。



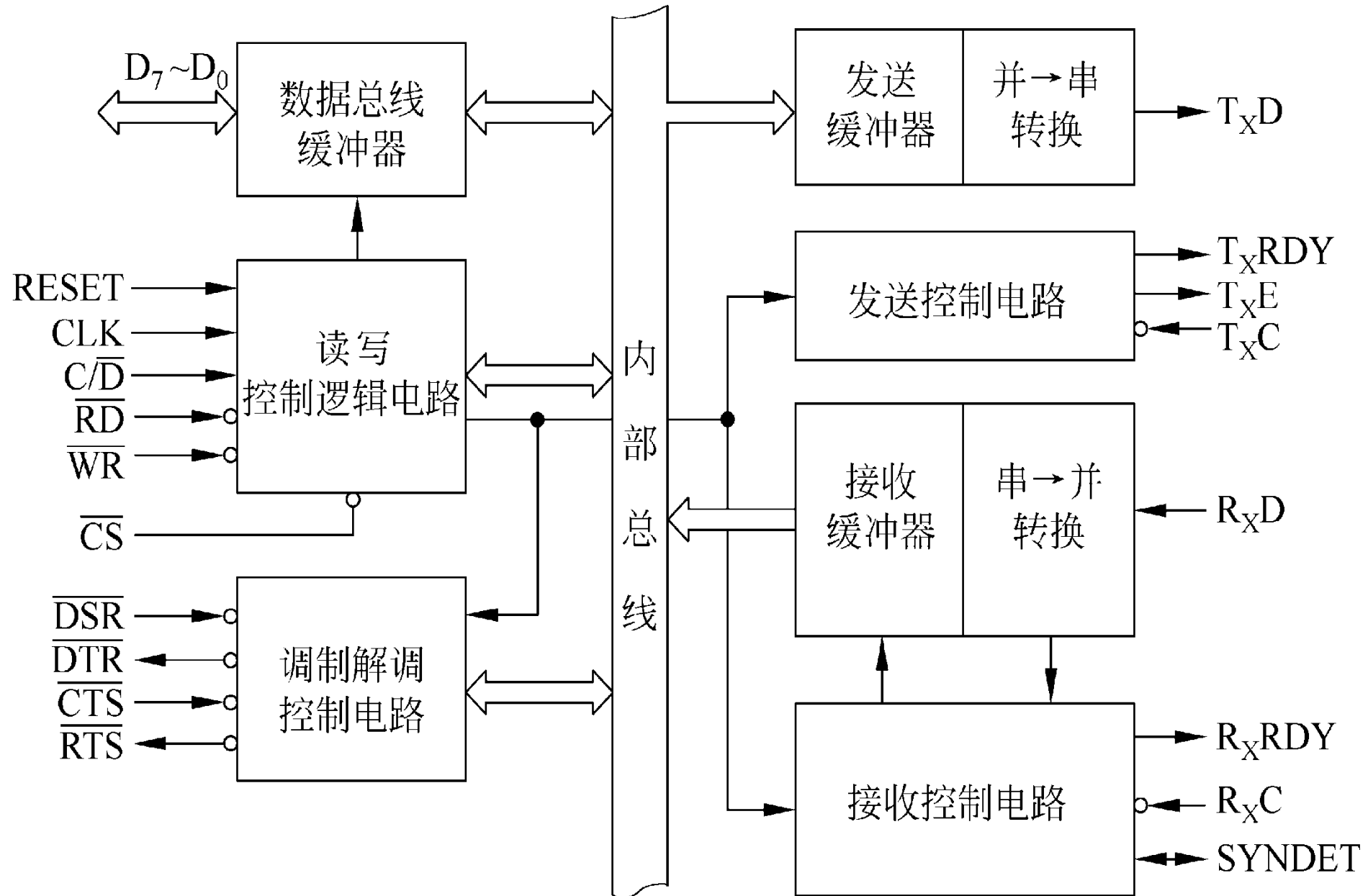
可编程串行接口的典型结构

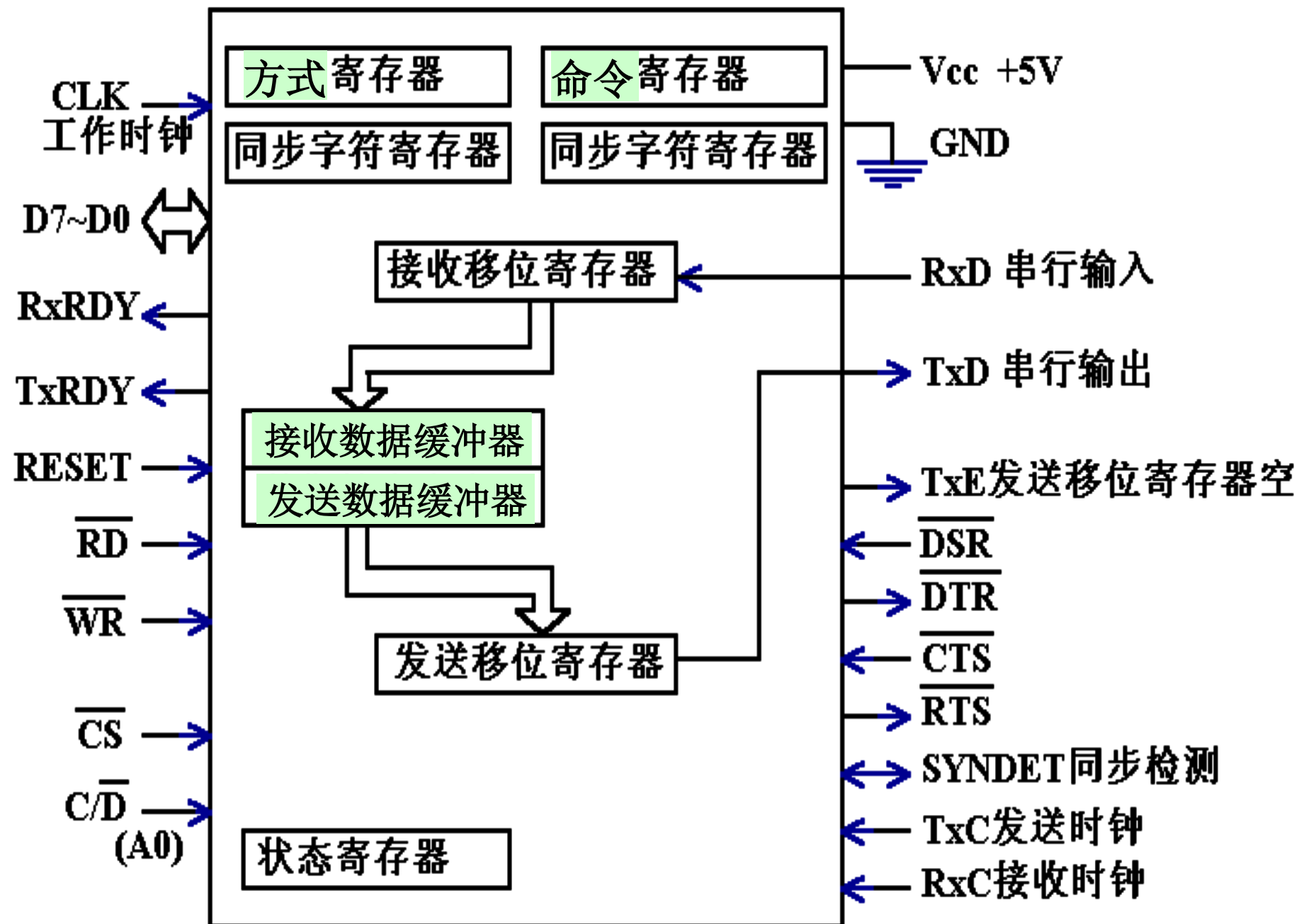
6.2 可编程串行通信接口8251A

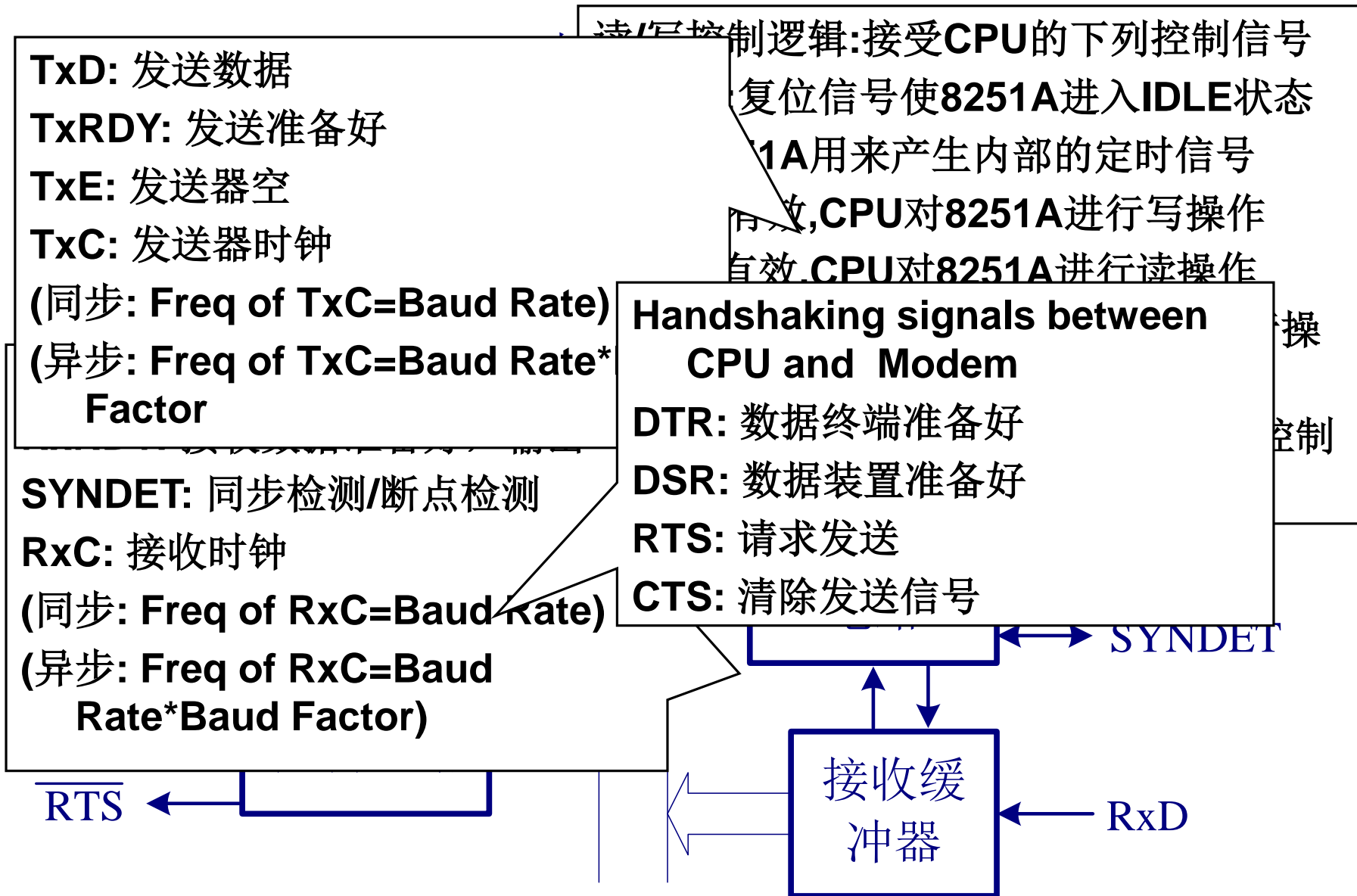
6.2.1 基本功能

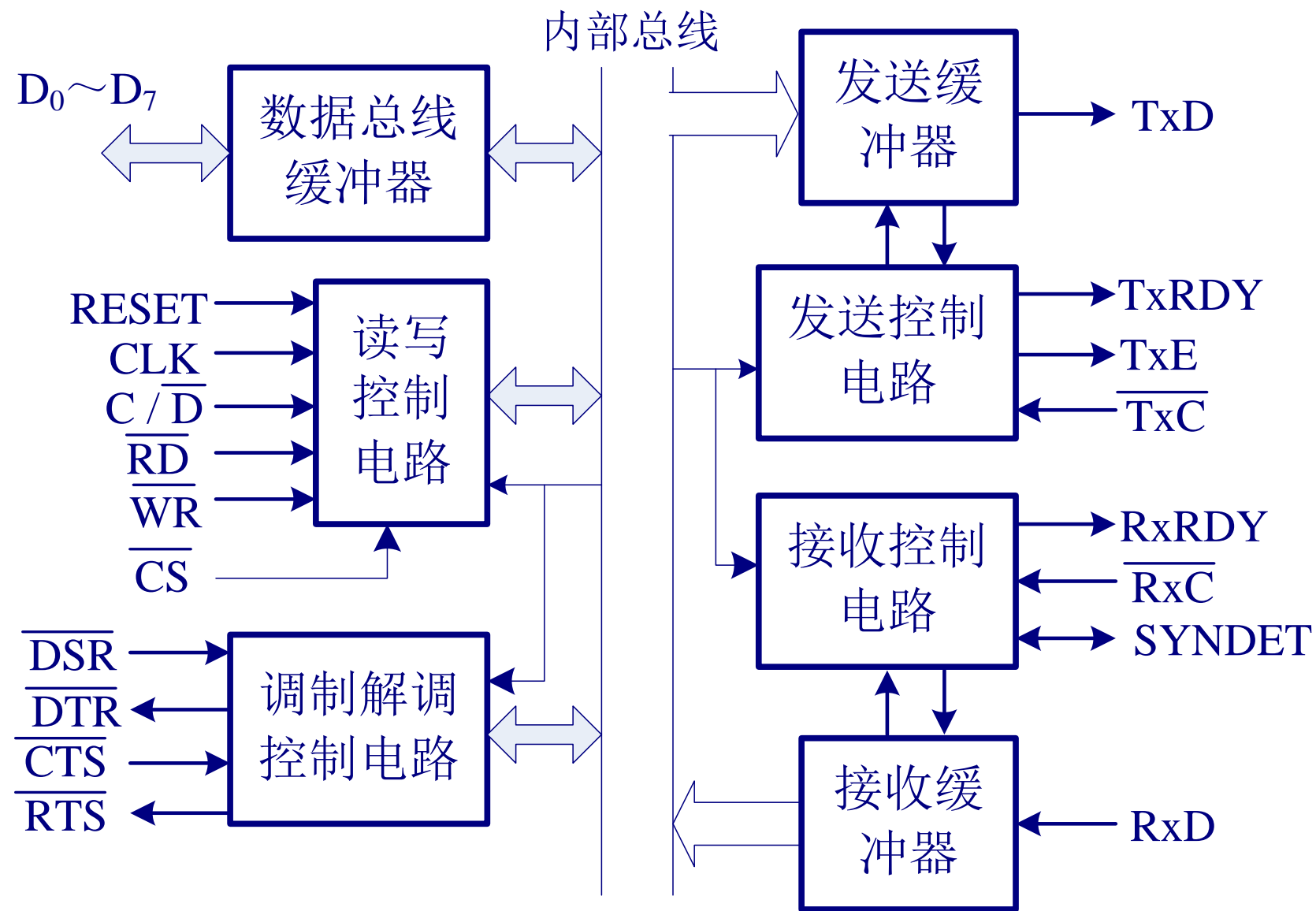
- (1) 可用于同步和异步通信方式(通信方式通过对方式字编程规定):
- (2)同步方式: 波特率0-64Kbps, 每字符为5, 6, 7, 8位, 可使用内部同步检测和外部同步检测, 能自动插入同步字符。
- (3)异步方式: 波特率 0-19.2Kbps, 每字符可为5, 6, 7, 8位, 自动增加起始位、停止位和校验位。时钟TxC, RxC速率为波特率的1, 16和64倍。
- (2) 完全双工, 双缓冲器接收器和发送器;

8251A的功能结构









6.2.3 8251A的对外信号

• 8251A和CPU之间的连接信号

- 片选信号
- 数据信号
- 读/写控制信号
- 收发联络信号

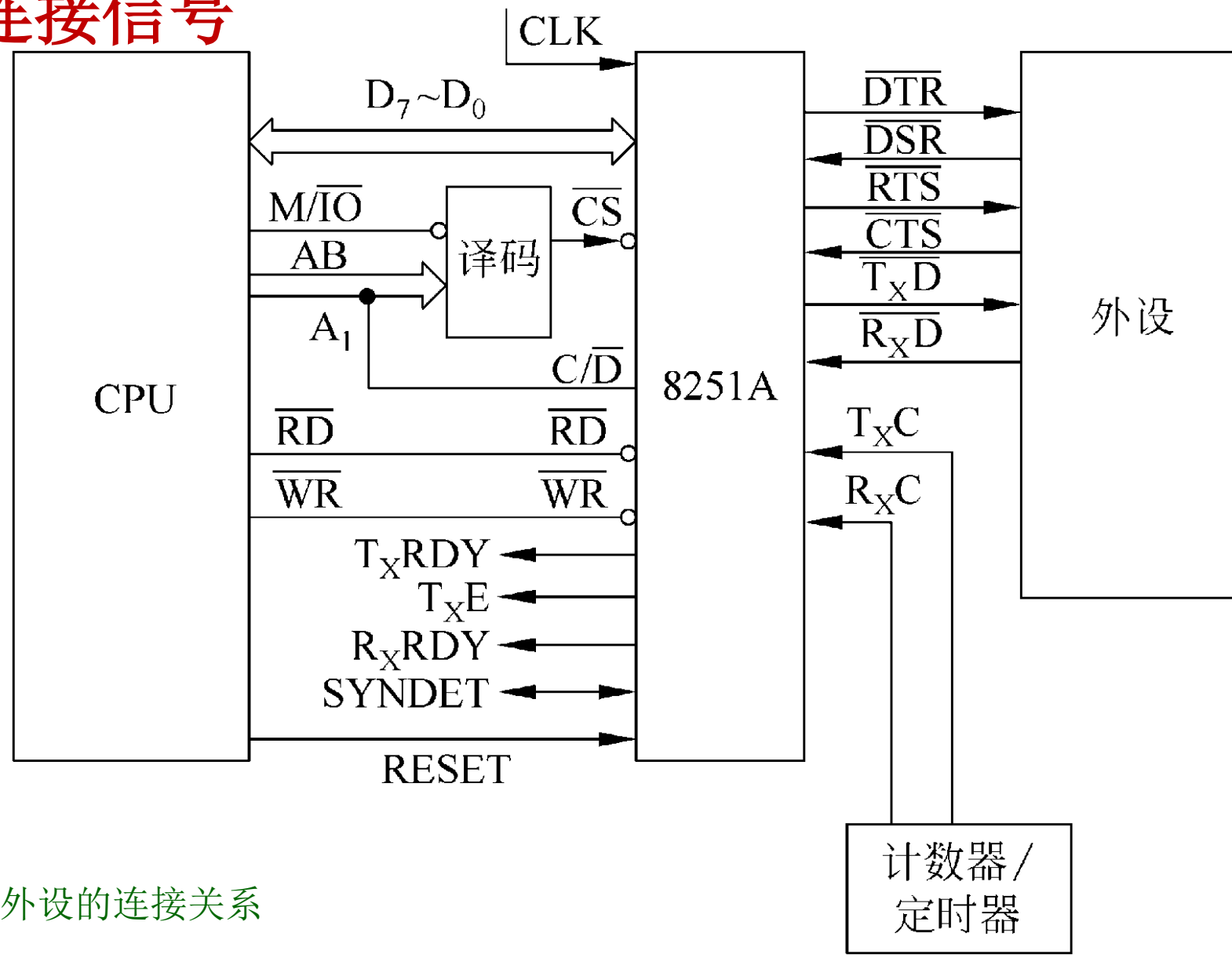


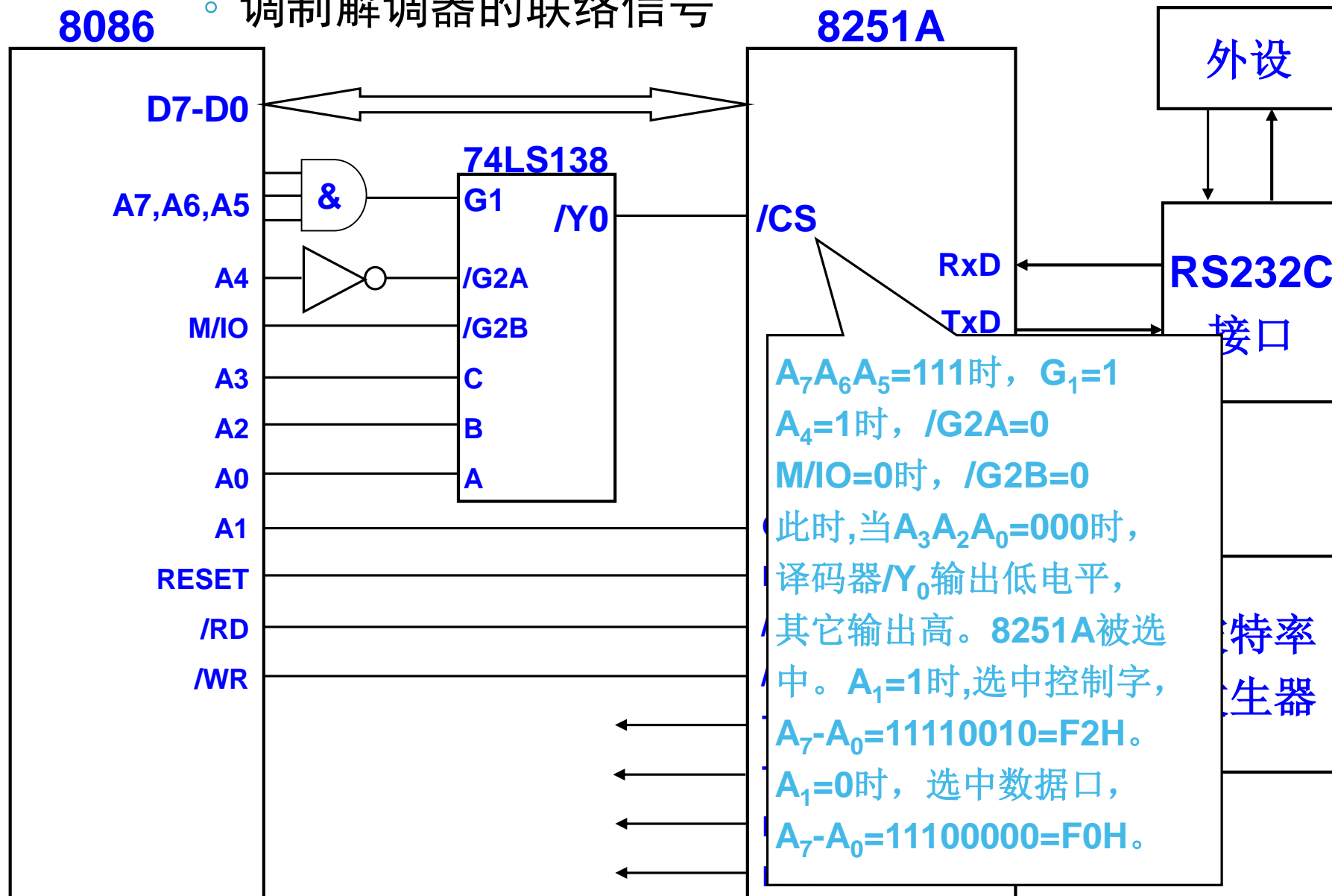
图6.6 8251A与CPU及外设的连接关系

8251A读写操作端口选择表

C/ \overline{D}	\overline{RD}	\overline{WR}	\overline{CS}	端口选择和操作
0	0	1	0	CPU从8251A接收数据寄存器读取数据
0	1	0	0	CPU向8251A发送数据缓冲器输出数据
1	0	1	0	CPU从8251A状态寄存器读状态
1	1	0	0	CPU向8251A写控制字（先方式字寄存器、后命令字寄存器）
X	X	X	1	数据总线悬空

▶ 8251A与外设之间的连接信号

- 数据信号
- 调制解调器的联络信号



6.2.4 8251A的编程

- 偶地址端口：数据输入/输出寄存器；
- 奇地址端口：状态寄存器、模式(方式)寄存器、控制寄存器和同步字符寄存器。
- 奇地址端口读出时对应状态寄存器；
- 奇地址端口写入时往模式寄存器、控制寄存器还是同步字符寄存器写入？

6.2.4 8251A的编程

- **8251A的初始化**

- **8251A初始化的约定：**

- ① 复位后，第一次写入的值作为模式字

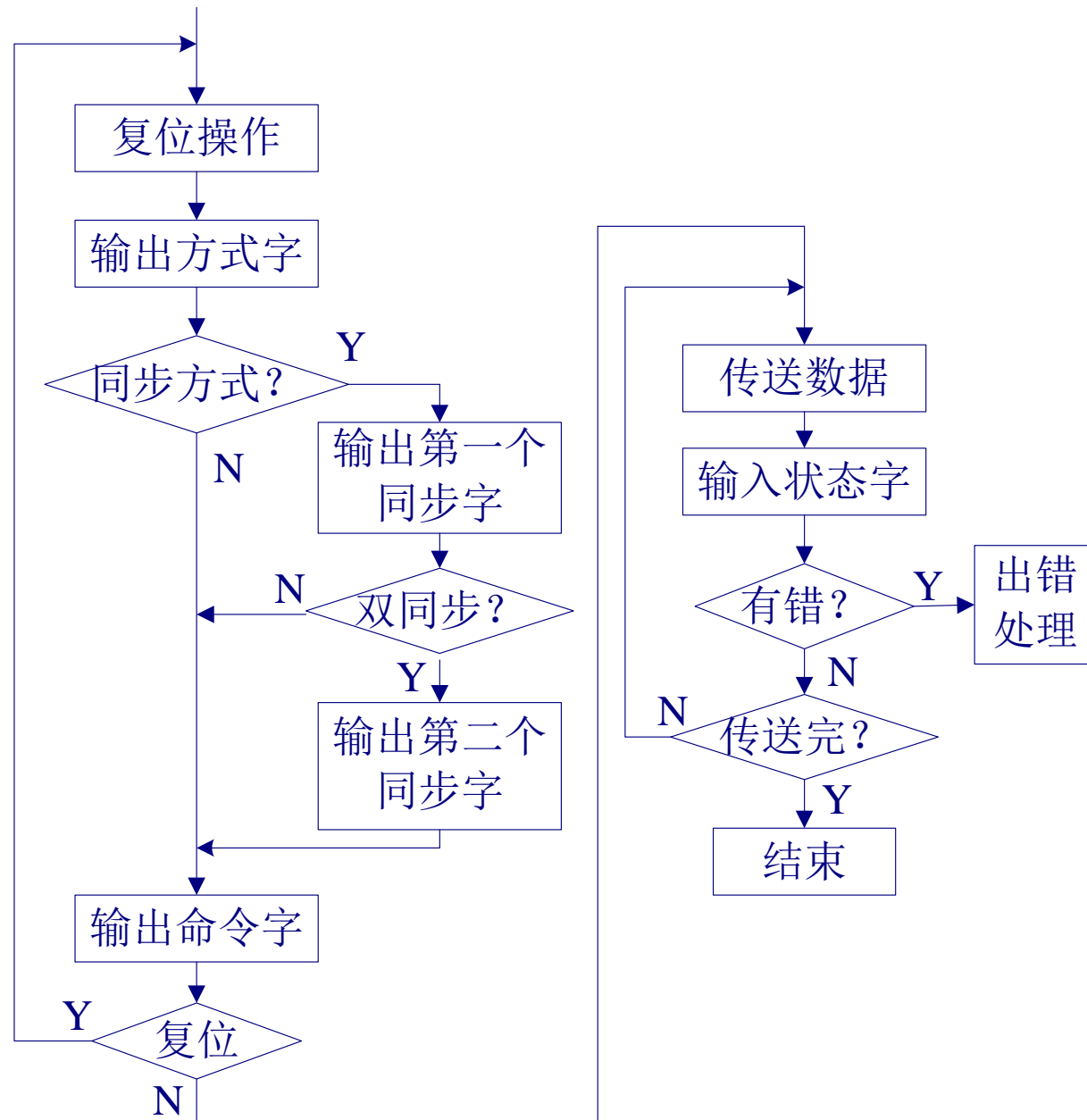
- ② 如为同步模式，则接着同步字符

- ③ 此后不管是同步模式还是异步模式，奇地址端口写入的值作为控制字，偶地址端口写入的值为数据

怎么写，往哪儿写，写什么？

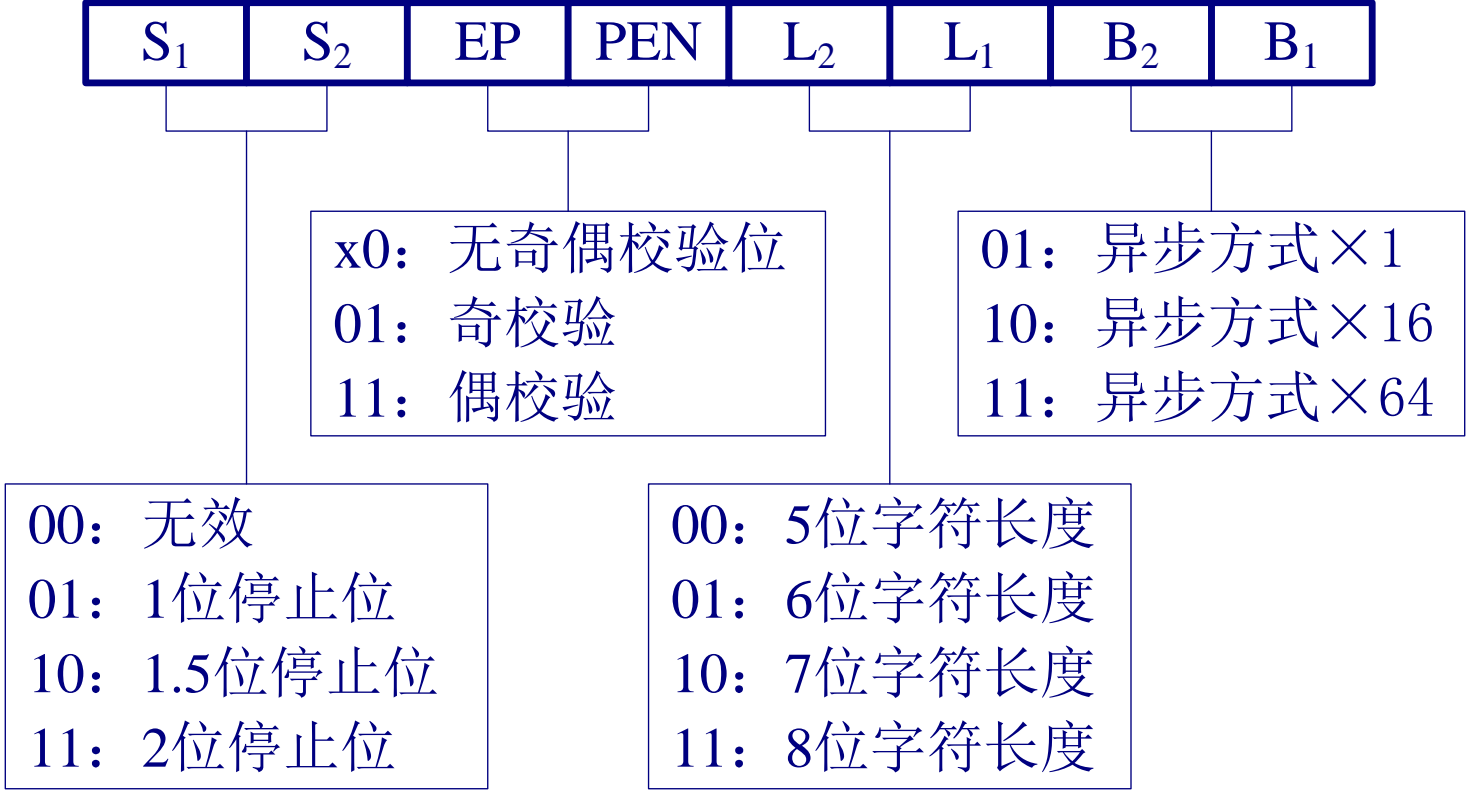
6.2.4 8251A的编程

编程流程

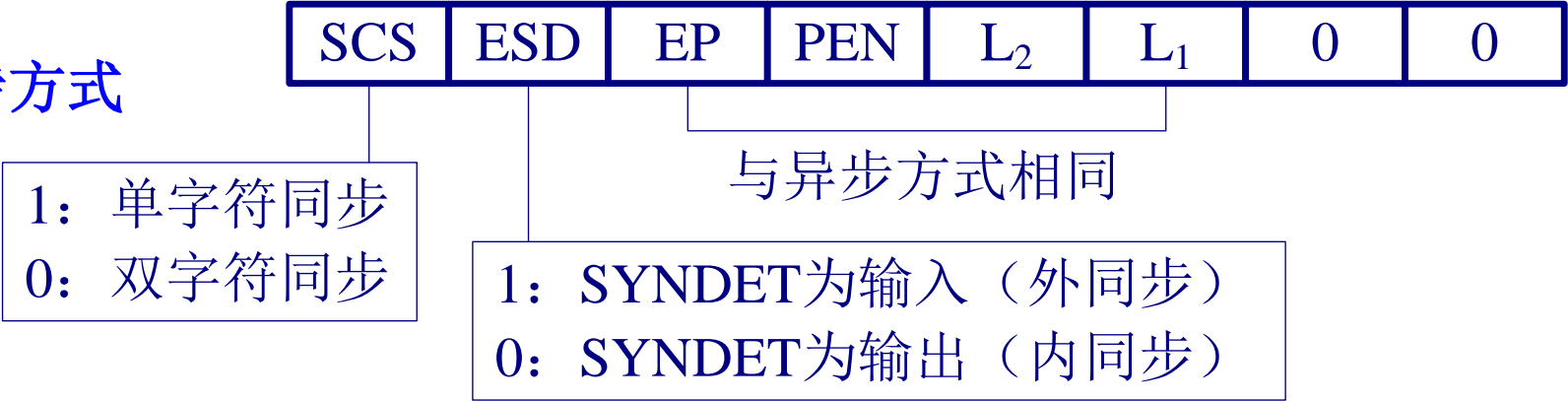


方式（模式）字

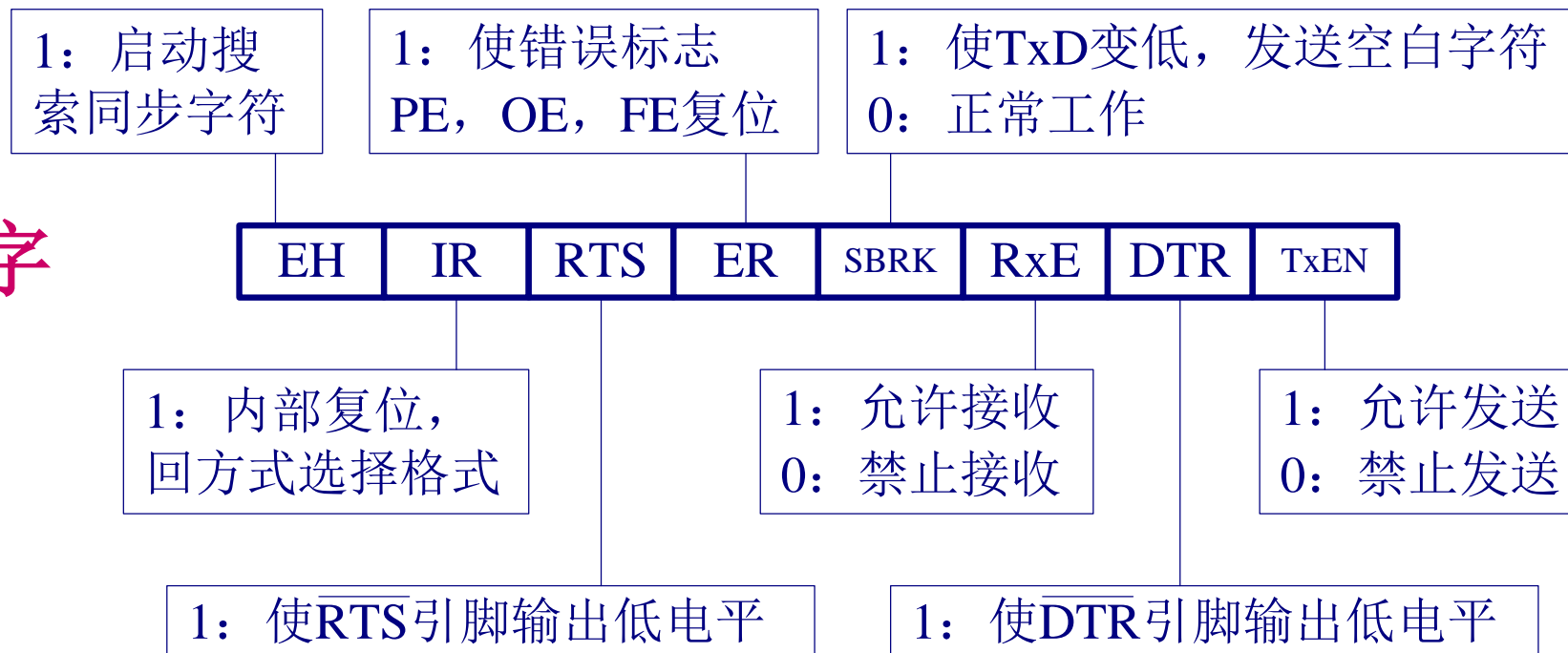
异步方式



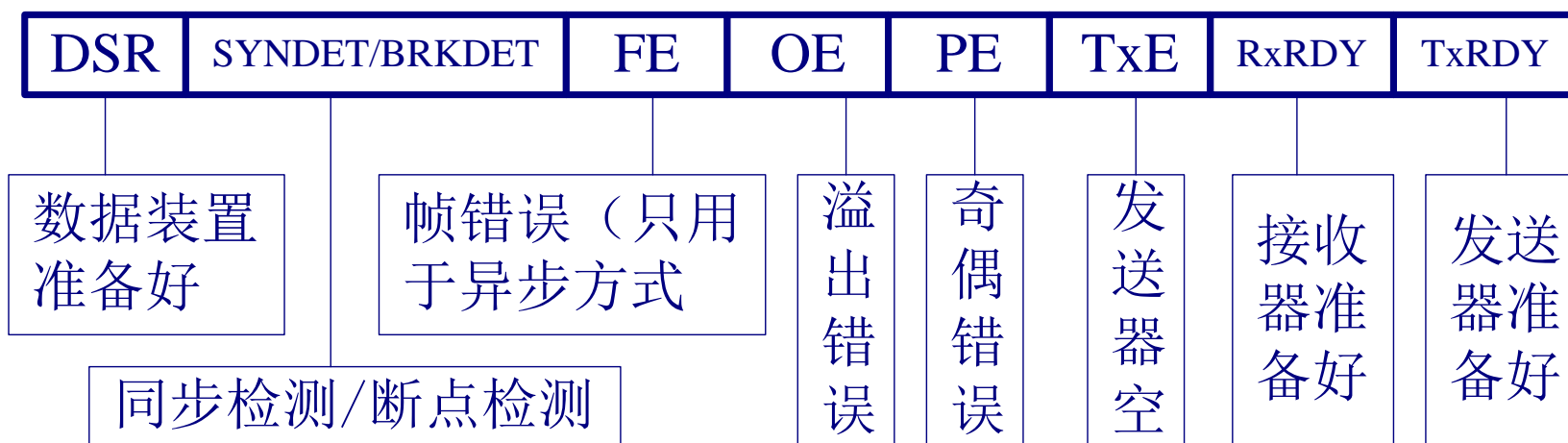
同步方式



命令字



状态字



6.2.5 8251A编程举例

▶ 8251A和调制解调器连接的例子

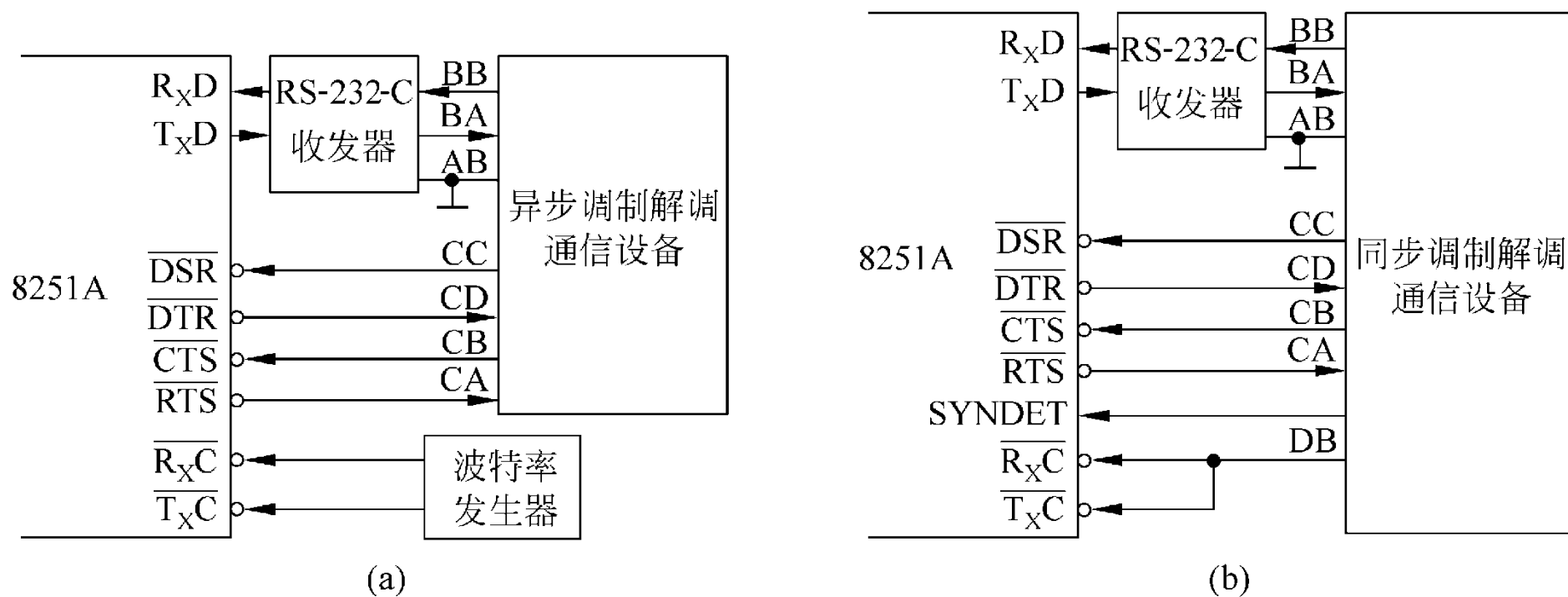


图6.11 8251A和调制解调器的连接

(a) 异步模式; (b) 同步模式

6.2.5 8251A编程举例

▶ 异步模式下的初始化程序举例

```
MOV     AL, 0FAH
OUT     42H, AL
MOV     AL, 37H
OUT     42H, AL
```

6.2.5 8251A编程举例

▶ 异步模式下的初始化程序举例

```
MOV    AL, 0FAH
OUT    42H, AL
MOV    AL, 37H
OUT    42H, AL
```

6.2.5 8251A编程举例

▶ 同步模式下的初始化程序举例

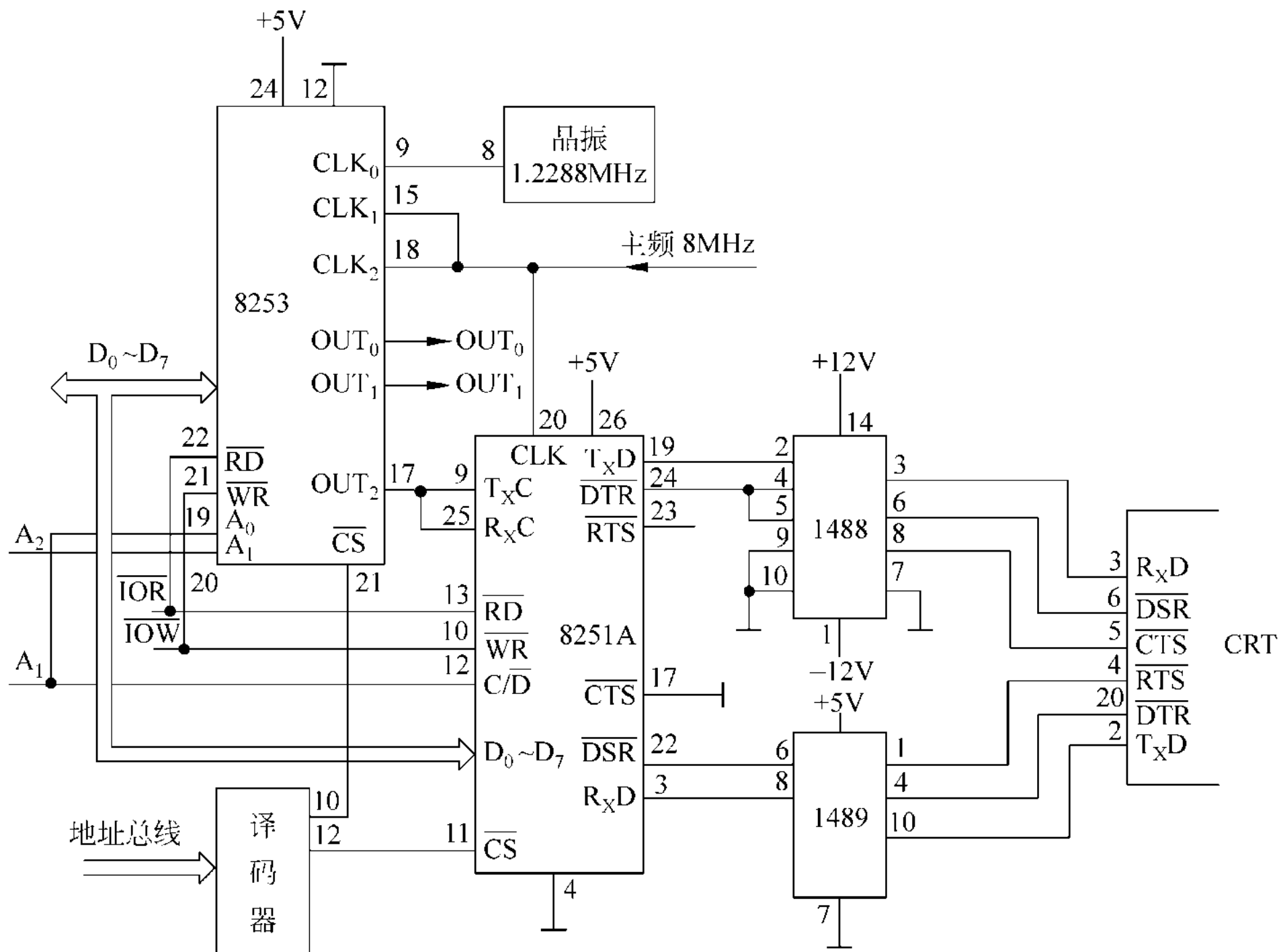
```
MOV      AL, 38H
OUT      42H, AL
MOV      AL, 16H
OUT      42H, AL
OUT      42H, AL
MOV      AL, 97H
OUT      42H, AL
```

利用状态字进行编程的举例

```
MOV    AL,0FAH
OUT    42H,AL
MOV    AL,35H
OUT    42H,AL
MOV    DI,0
MOV    CX,80
BEGIN: IN  AL,42H
TEST   AL,02H
JZ     BEGIN
IN     AL,40H
MOV    DX,OFFSET BUFFER
MOV    [DX+DI],AL
INC    DI
IN     AL,42H
TEST   AL,38H
JNZ    ERROR
LOOP   BEGIN
JMP    EXIT
ERROR: CALL  ERR_OUT
EXIT:  ...-
```

6.2.6 8251

图6.12 用8251A作为
CRT接口的实际例子



6.2.6 8251A的使用实例

```
INIT:  XOR    AX, AX
        MOV    CX, 0003
        MOV    DX, 00DAH
OUT1:   CALL    KKK
        LOOP   OUT1
        MOV    AL, 40H
        CALL    KKK
        MOV    AL, 4EH
        CALL    KKK
        MOV    AL, 27H
        CALL    KKK

        .....
KKK:    OUT     DX, AL
        PUSH    CX
        MOV     CX, 0002
ABC:    LOOP    ABC
        POP     CX
        RET
```

6.2.6 8251A的使用实例

- ▶ 往CRT输出一个字符的例子

CHAROUT: MOV DX, 0DAH

STATE: IN AL, DX

TEST AL, 01

JZ STATE

MOV DX, 0D8H

POP AX

OUT DX, AL

8251编程示例

例：编写8251异步模式下的接收和发送程序，完成256个字符的发送和接收，设端口地址：208H，209H，波特率因子16，1起始位，1停止位，无奇偶校验，每字符8位。

发送程序

```
LEA DI, Buf1
MOV DX, 209H
MOV AL, 00H      ;复位
OUT DX, AL
CALL DELAY
MOV AL, 00H      ;复位
OUT DX, AL
CALL DELAY
MOV AL, 00H      ;复位
OUT DX, AL
CALL DELAY
MOV AL, 40H      ;复位命令
OUT DX, AL
```

```
MOV AL, 01001110B ;模式选择字
OUT DX, AL
MOV AL, 00110111B ;控制字
OUT DX, AL
MOV CX, 256 ;发送256字节
NEXT: MOV DX, 209H ;状态字寄存器209H
IN AL, DX ;状态字
AND AL, 01H ;TxRDY?
JZ NEXT
MOV AL, [DI]
MOV DX, 208H ;数据寄存器208H
OUT DX, AL ;发送
INC DI
LOOP NEXT
```


接收程序：接收256字节，放在buf2中

Data segment

buf2 DB 256 dup(?)

Data ends

⋮

MOV DX,209H

MOV AL,00H ;复位

OUT DX, AL

CALL DELAY

MOV AL,00H ;复位

OUT DX, AL

CALL DELAY

MOV AL,00H ;复位

OUT DX, AL

CALL DELAY

MOV AL,40H ;复位

OUT DX, AL

MOV AL, 01001110B ;方式字

OUT DX, AL

MOV AL, 00110111B ;命令字

OUT DX, AL

MOV CX, 256 ;接收256字节

MOV SI, 0

NEXT: MOV DX, 209H

IN AL, DX ;状态字

AND AL, 02H ;RXRDY?

JZ NEXT

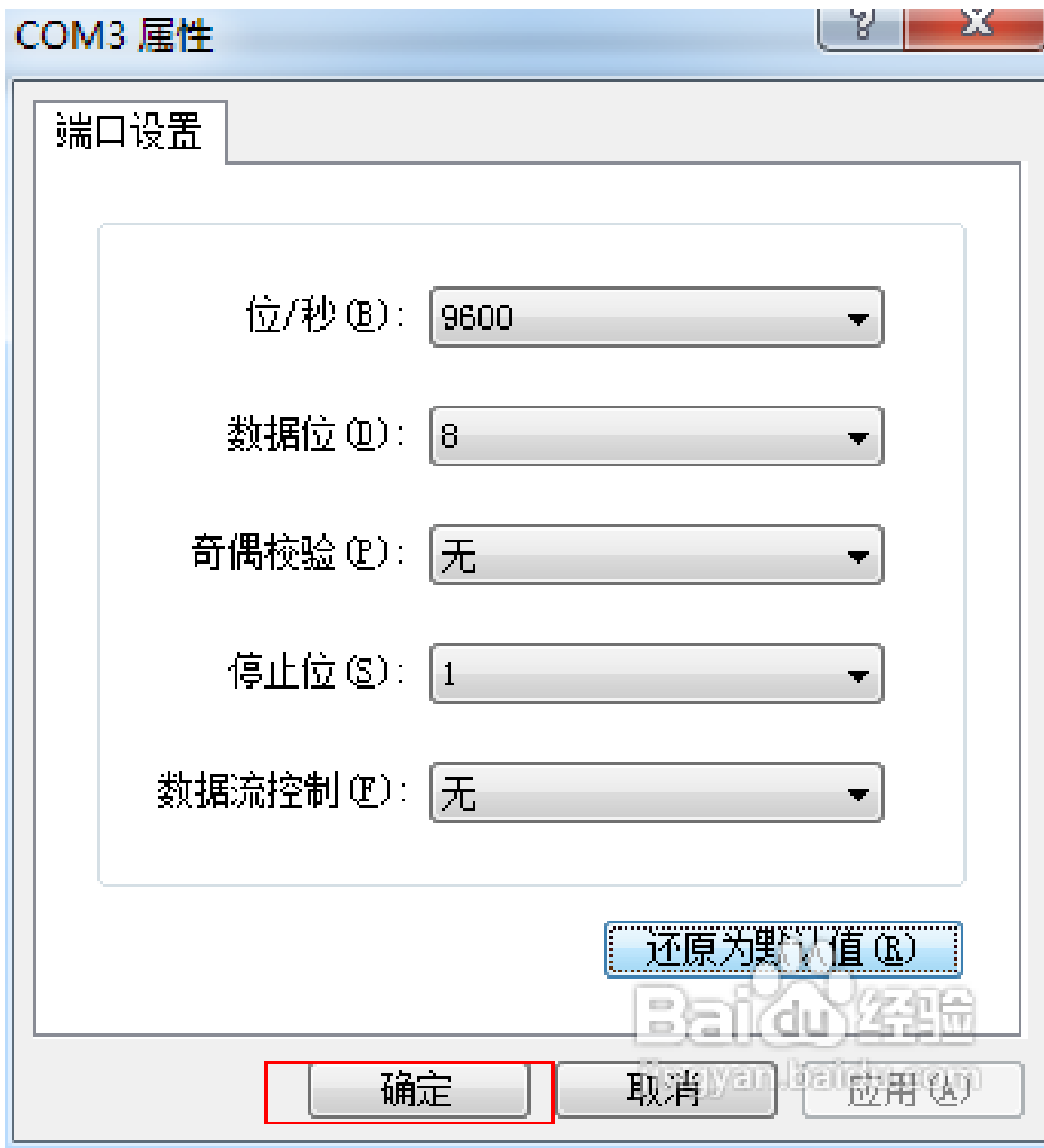
MOV DX, 208H

IN AL, DX ;接收1字符

MOV buf2[SI], AL

INC SI

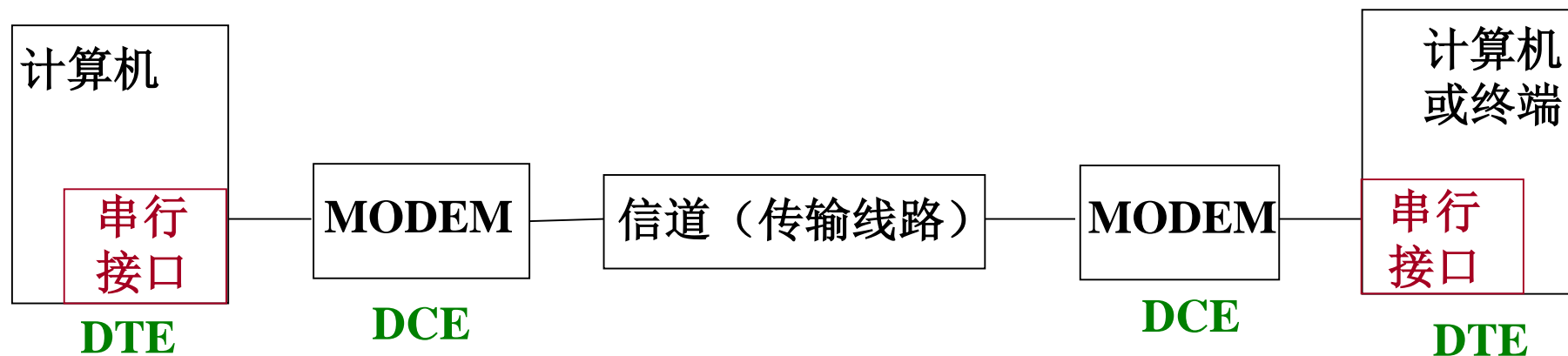
LOOP NEXT



串行接口标准

RS-232C串行接口标准

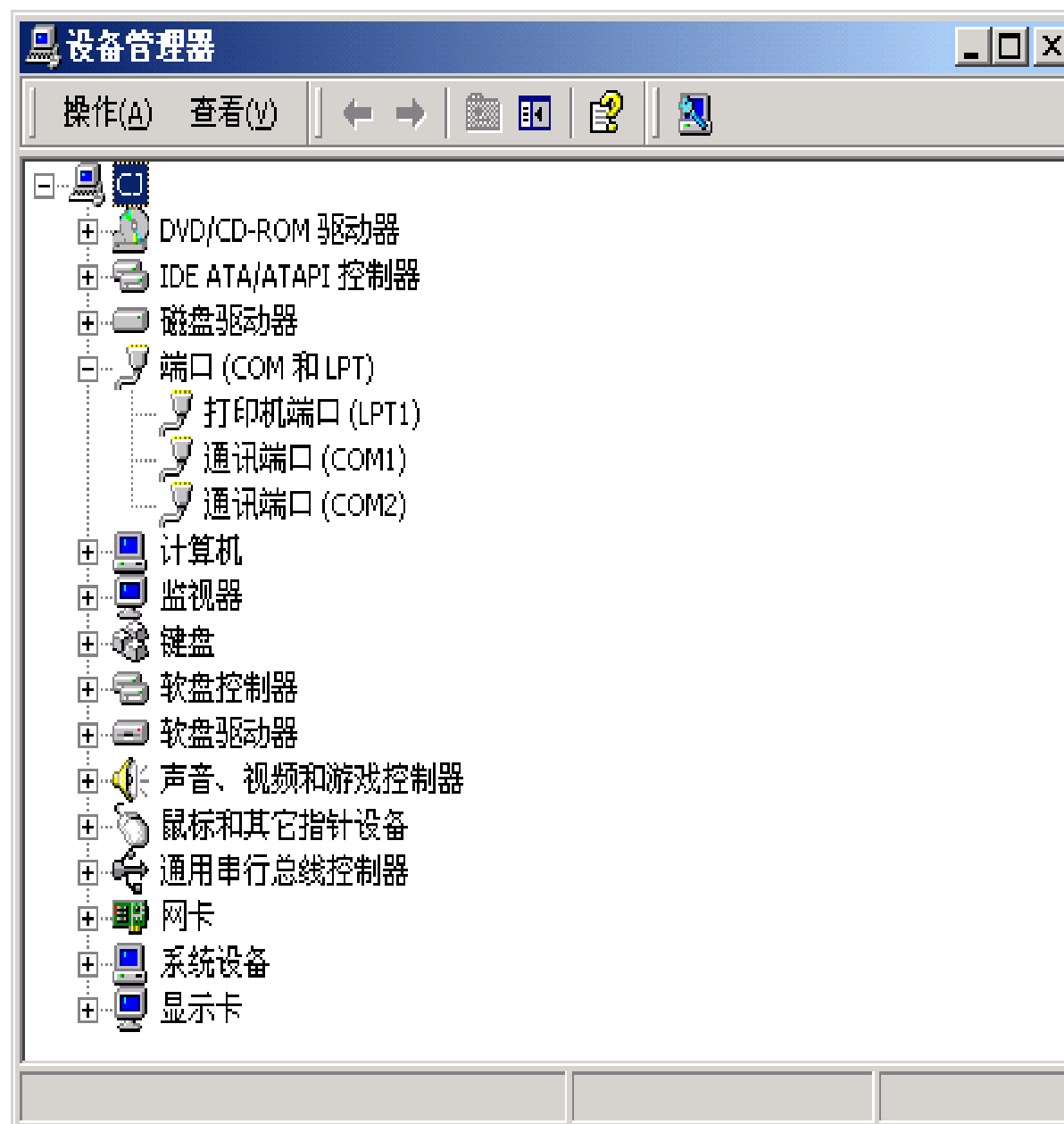
串行通讯系统



为了使来自不同厂家的计算机、外部设备以及数据通信设备都能正确连接，这个接口的机械特性、电气特性、功能特性都要遵循一定的规范，也就是要有一个标准。

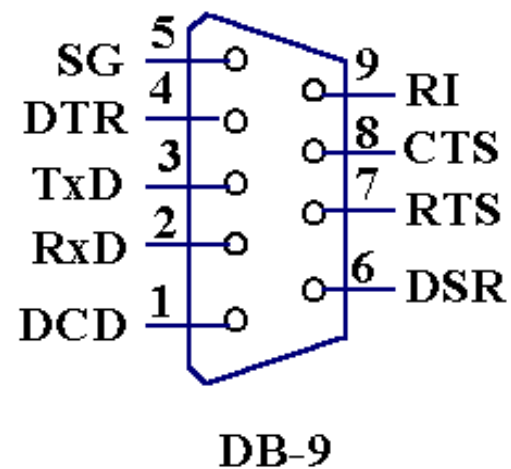
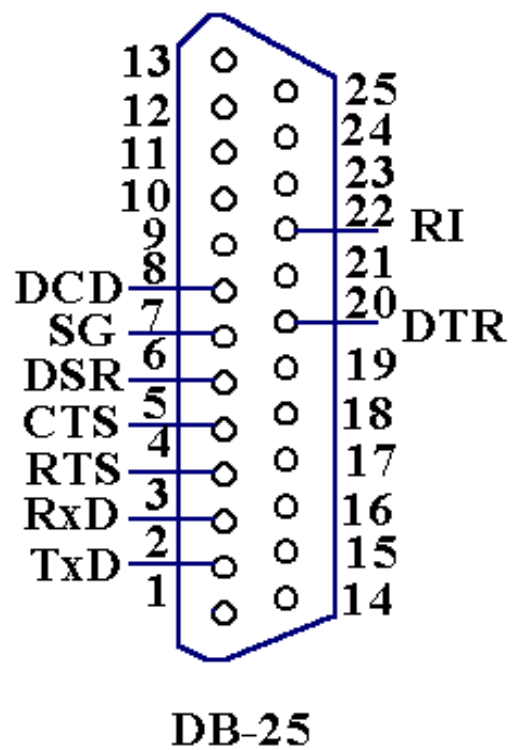
RS-232C

(Recommended Standard推荐标准) 是美国电子工业协会 (EIA) 推荐的标准串行接口，是应用于数据通信设备和数据终端设备之间的标准接口。通常有两个信道 COM1、COM2。



接口机械特性

目前COM1、COM2 均使用9针连接器。



接口功能特性

RS-232C的引脚

9针引脚	25针引脚	名称	25针引脚	名称
	1	保护地	12	次信号载波检测
3	2	发送数据TxD	13	次信号清除发送
2	3	接收数据RxD	14	次信号发送数据
7	4	请求发送RTS	16	次信号接收数据
8	5	清除发送CTS	19	次信号请求发送
6	6	数据装置准备好DSR	21	信号质量检测
5	7	信号地GND	23	数据速率检测
1	8	载波检测CD	24	终端发生器时钟
4	20	数据终端准备好DTR	9、10	保留
9	22	振铃提示RI	11	未定义
	15	发送时钟TxC	18	未定义
	17	接收时钟RxC	25	未定义



DCD 数据载波检出 (DCE→DTE) 当本地DCE收到对方的DCE设备送来的载波信号时，使DCD有效，通知DTE准备接收，并且由DCE将接收到的载波信号解调为数字信号，经Rx D线送给DTE。

RI 振铃信号 (DCE→DTE) 当DCE收到交换机送来的振铃呼叫信号时，使该信号有效，通知DTE已被呼叫。

接口电气特性

标准规定：逻辑“1”信号，电平在 $-3V \sim -15V$ 之间；

逻辑“0”信号，电平在 $+3V \sim +15V$ 之间；

因此，使用RS-232C与微机接口时，需要将TTL电平（ $0 \sim 5V$ ）与RS-232C电平进行转换。

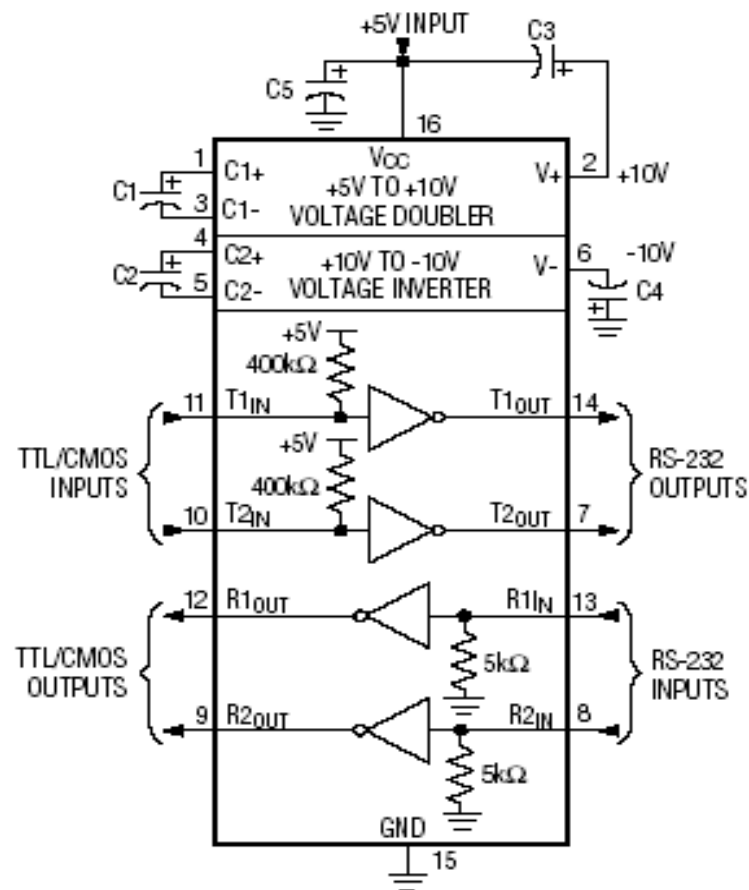
MAXIM公司

MAX232/MAX233

是现在常用的TTL↔

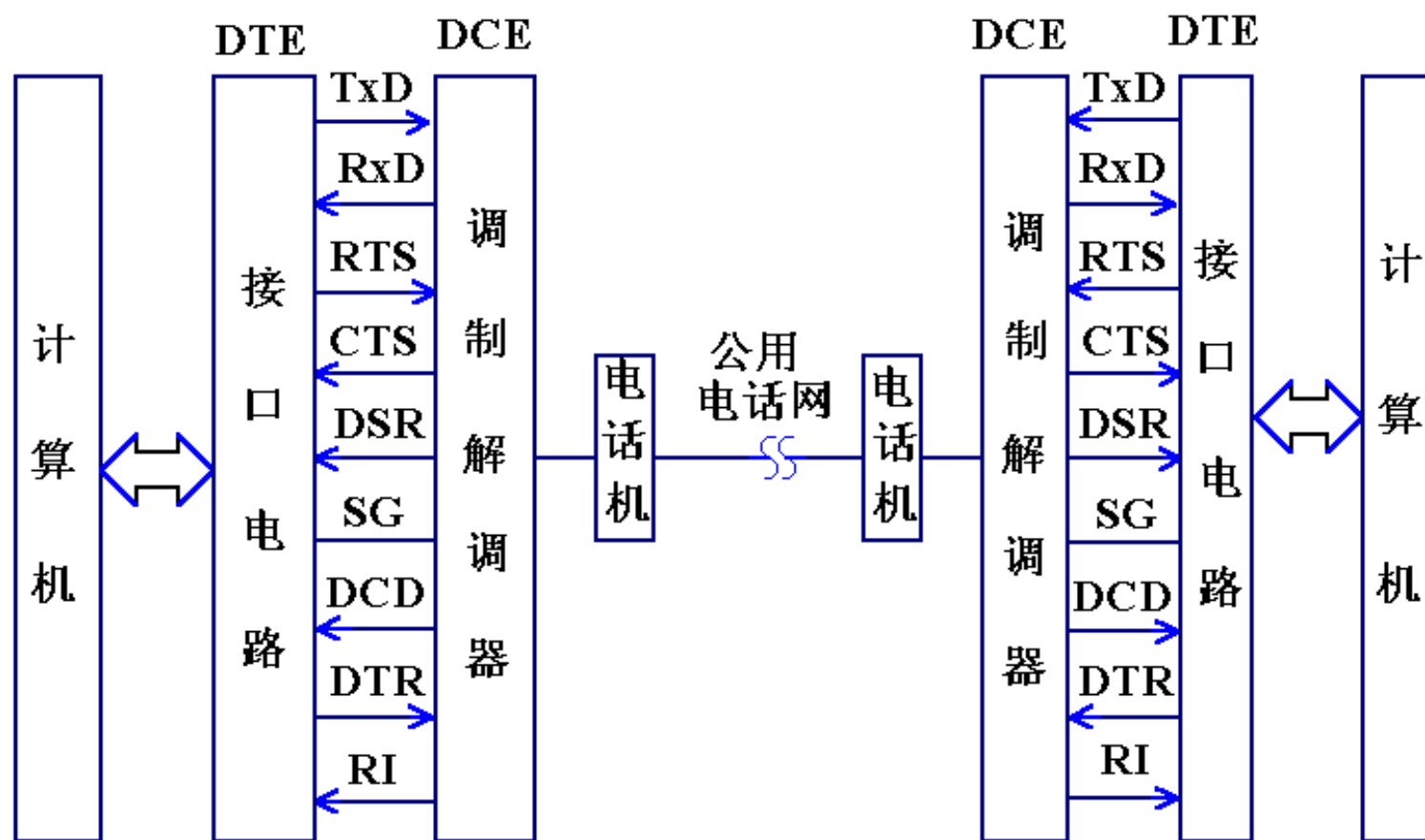
RS232C电平转换器

，其最大的优点是只需单+5V电源。

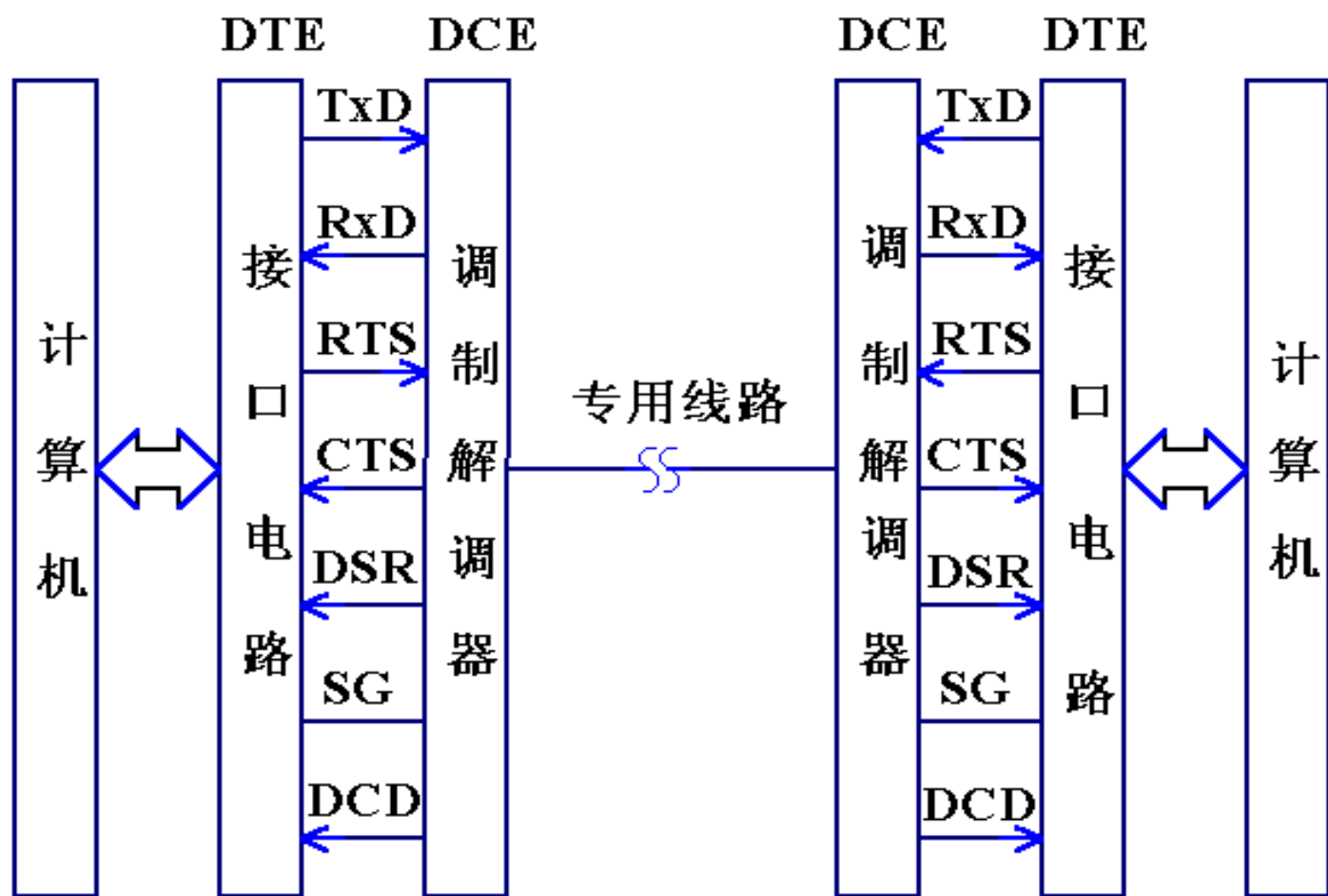


系统连接

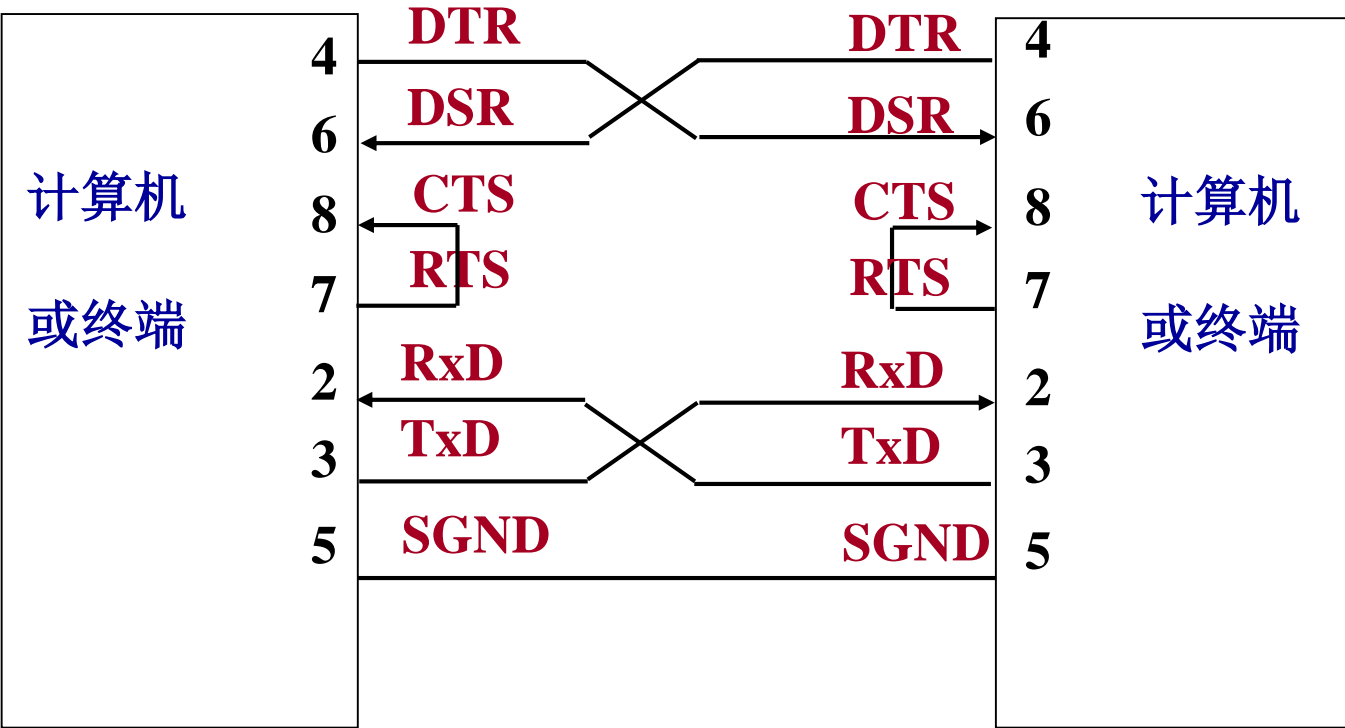
(1) 采用Modem(DCE)和电话网通信时的信号连接



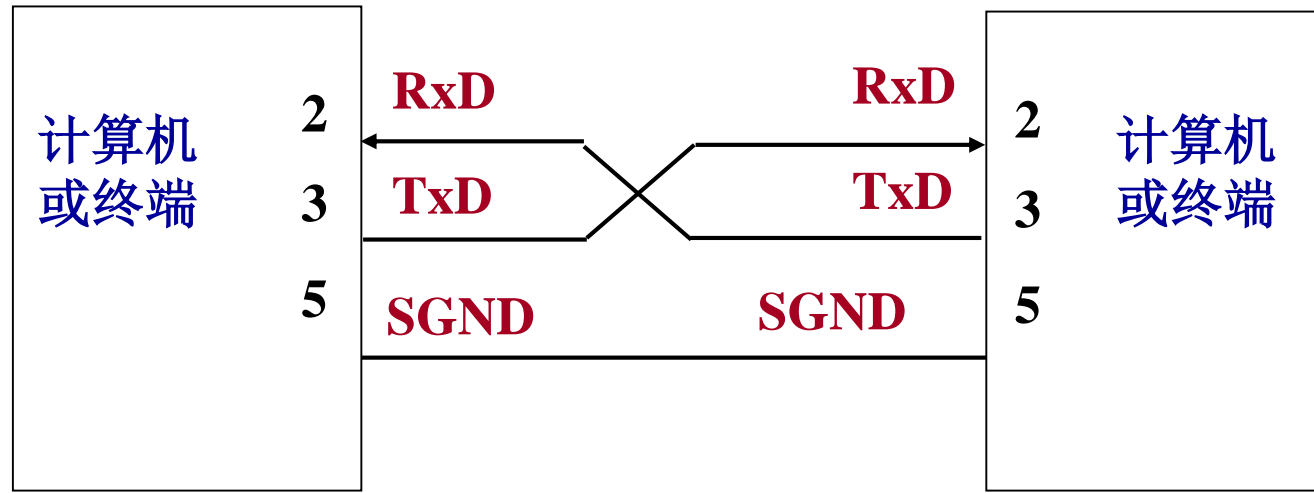
(2) 采用专用线通讯时的信号连接



(3) 无Modem的标准连接

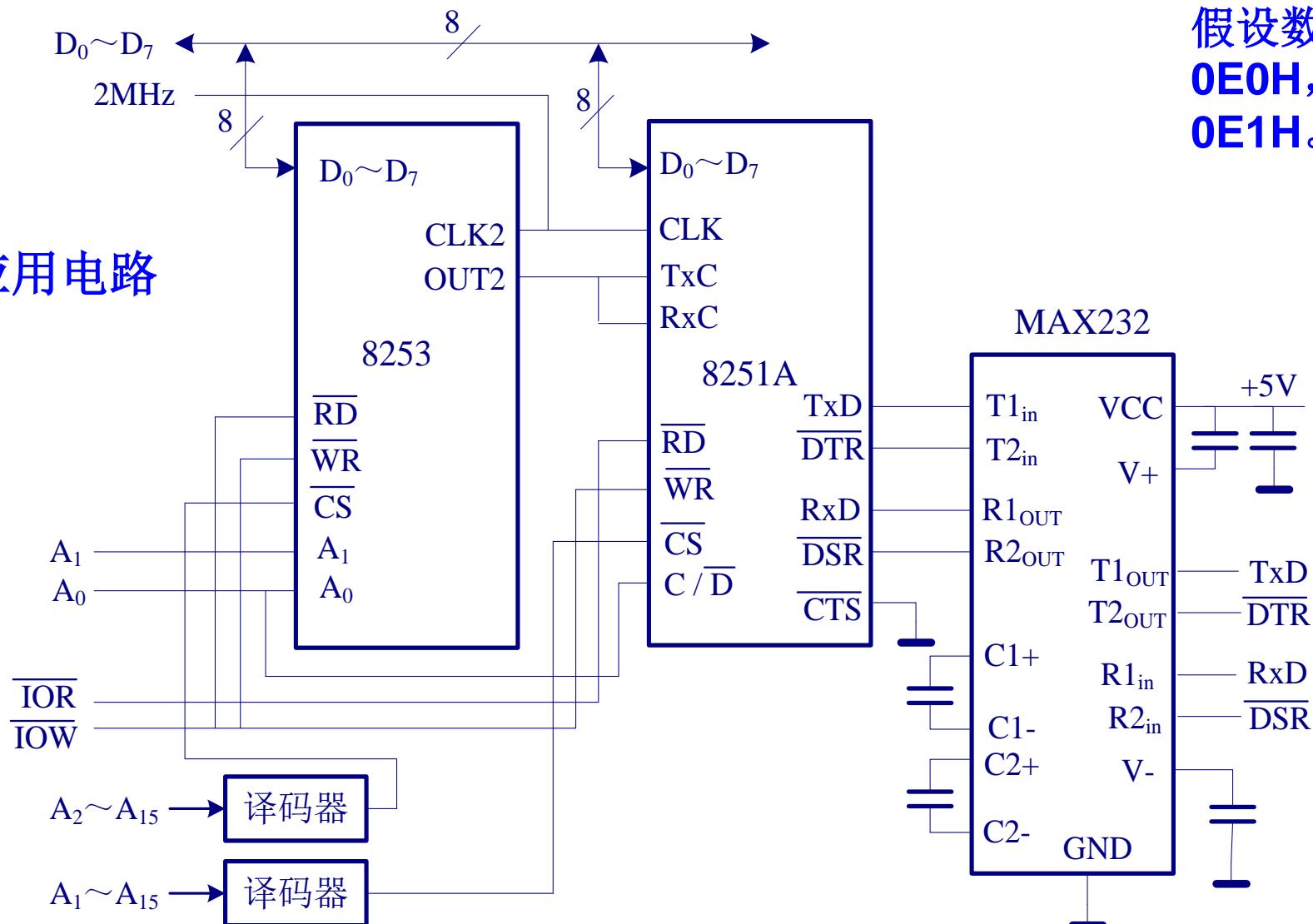


(4) 无Modem的最简单的方式:



要求：8251A的波特率为
2400，波特率系数为16，
假设数据端口地址为
0E0H，控制端口地址为
0E1H。

应用电路



初始化程序:

```
MOV AL,00H      ;复位
OUT 0E1H, AL
CALL DELAY
OUT 0E1H, AL    ;复位
CALL DELAY
OUT 0E1H, AL    ;复位
CALL DELAY
MOV AL,40H      ;复位
OUT 0E1H, AL    ;复位
CALL DELAY
MOV AL, 01001110B ;方式字
OUT 0E1H,AL
MOV AL, 00100111B
; 命令字, 启动发送器和接收器
OUT 0E1H,AL
```

设发送数据已放入AH中, 数据输出程序如下:

```
WAIT:  IN AL, 0E1H      ;状态字
        TEST AL, 01H     ;RXRDY?
        JZ WAIT
        MOV AL, AH
        OUT 0E0H, AL
```