



第9章 微机系统串行通信

1

串行通信基础

2

可编程串行异步通信
接口芯片8250

3

串行通信程序设计

9.1 串行通信基础

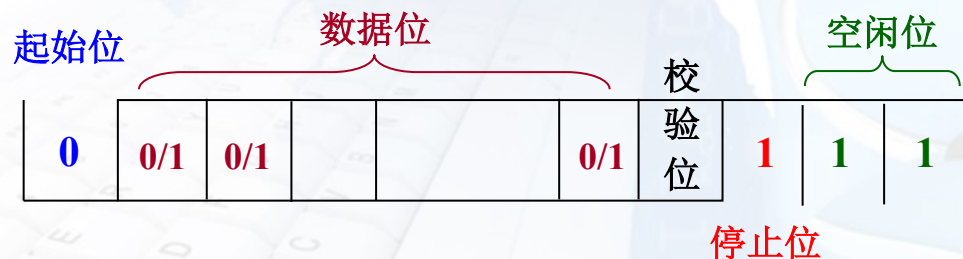


一、计算机通信方式

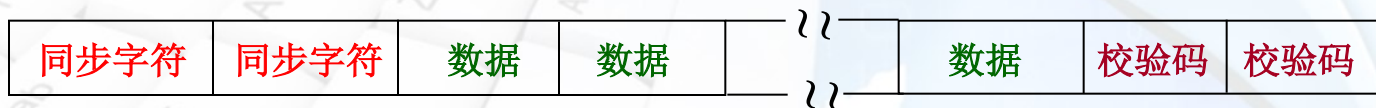
CPU与外部的信息交换称为通信。

通信 { 并行通信: 数据所有位同时被传输
串行通信: 数据被逐位顺序传输 { 串行异步通信
串行同步通信

1. 串行异步通信: 指一帧字符用起始位和停止位来完成收发同步。



2. 串行同步通信: 靠同步字符来完成收发双方同步, 要求严格的时钟同步。



数据传送速度: 并行通信 > 串行同步通信 > 串行异步通信

二、串行数据传输方式

1) 单工方式: 只允许数据按照一个固定的方向传送。



2) 半双工方式: 要求收发双方均具备接收和发送数据的能力, 由于只有一条信道, 数据不能同时在这两个方向上传送。



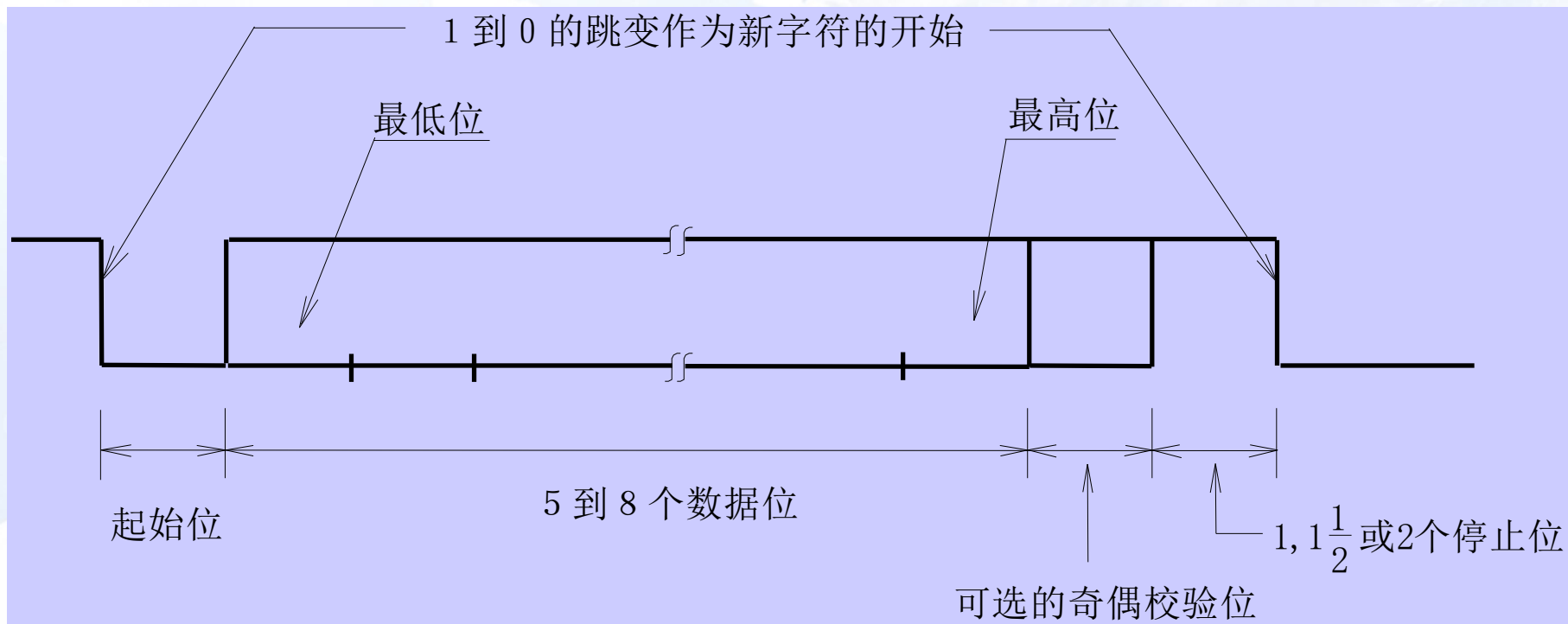
3) 全双工方式: 在全双工方式中, 收发双方可同时进行数据传送。



三、串行异步通信协议

1. 标准数据格式

图在P₃₁₉

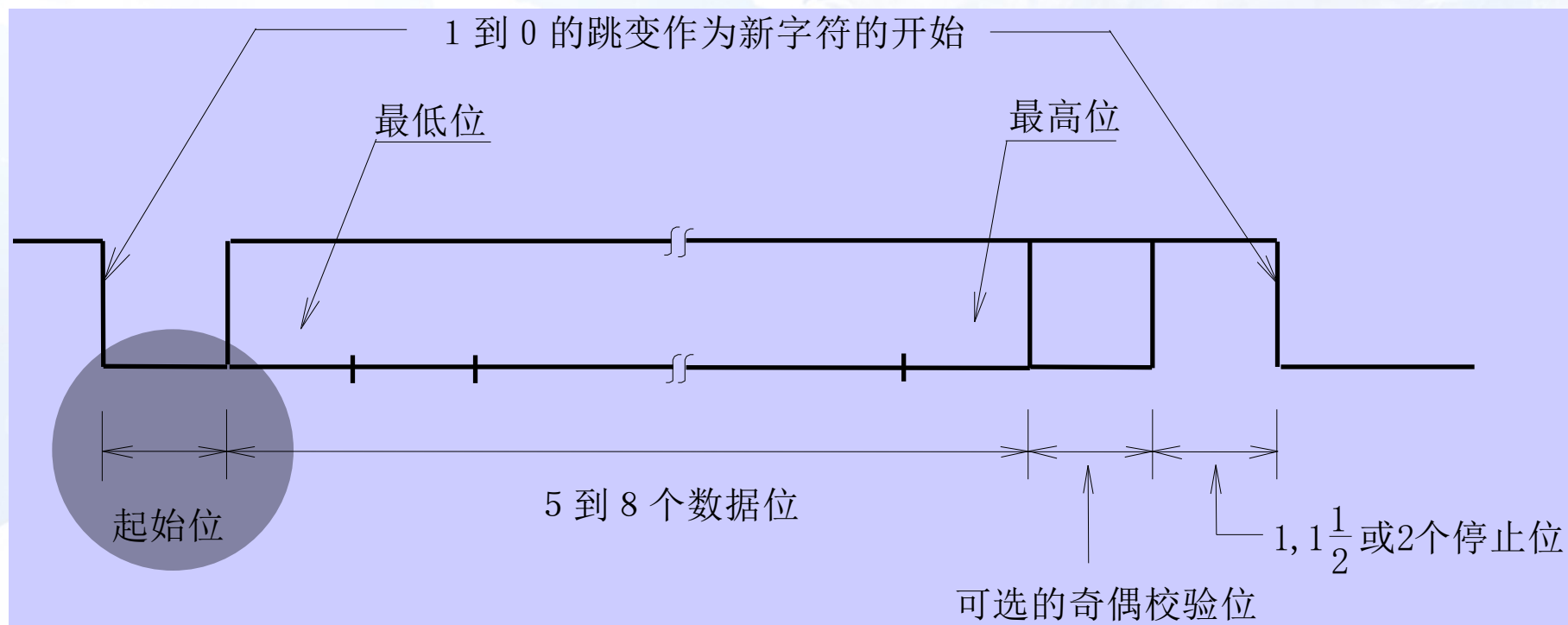


异步通信时，一帧字符以**起始位** ‘0’开始，紧跟着是**数据位**（**先发数据最低位，再是高位**），**奇偶校验位**，最后以**停止位**结束。

无数据传送时，通信线长时间逻辑 ‘1’，空闲位。

三、串行异步通信协议

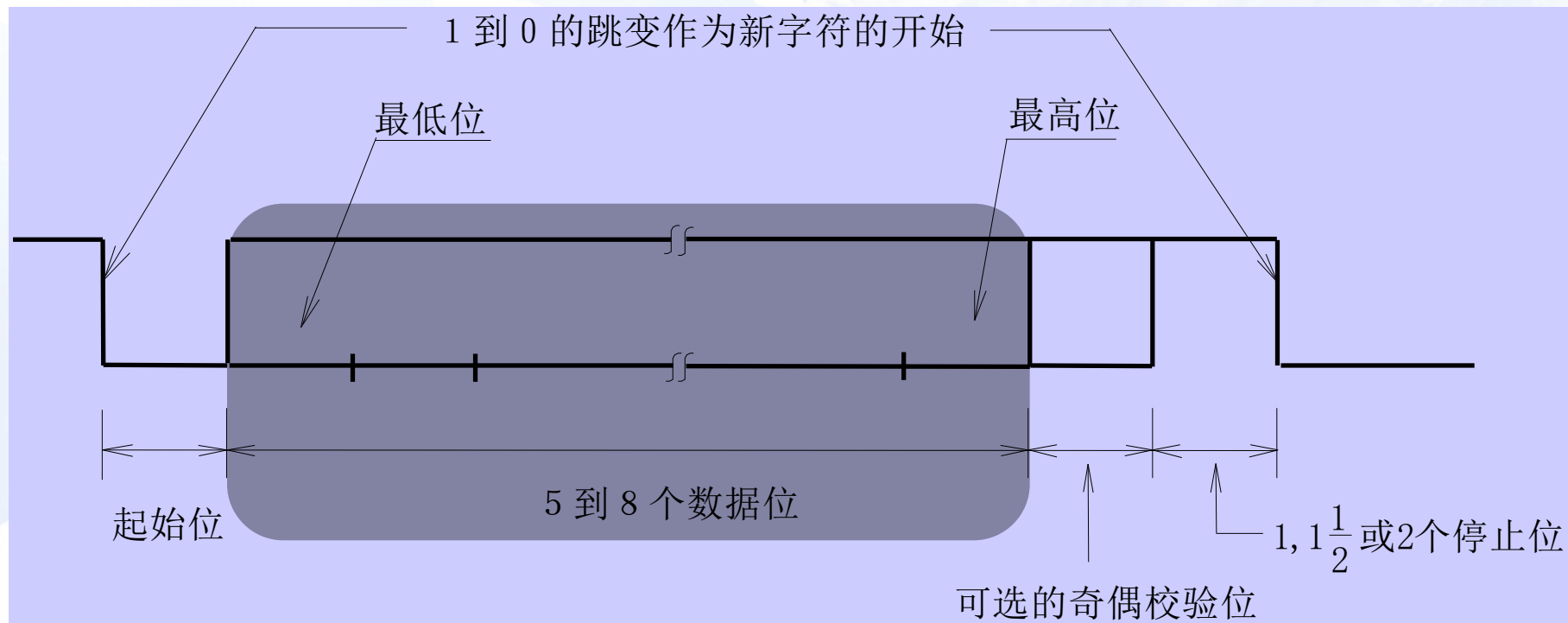
1. 标准数据格式



一帧数据包括四部分：1) **起始位**：1位**逻辑‘0’**；

三、串行异步通信协议

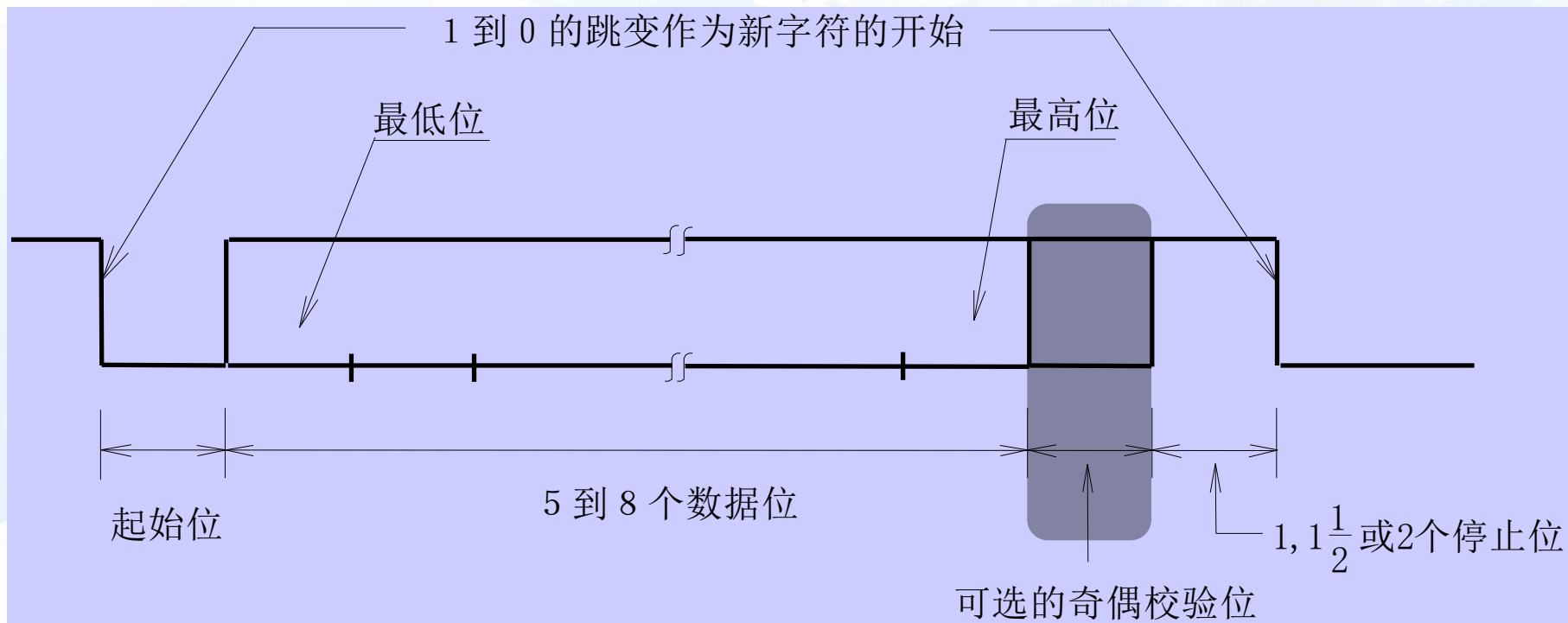
1. 标准数据格式



2) **数据位:** 5~8位**0、1**代码;

三、串行异步通信协议

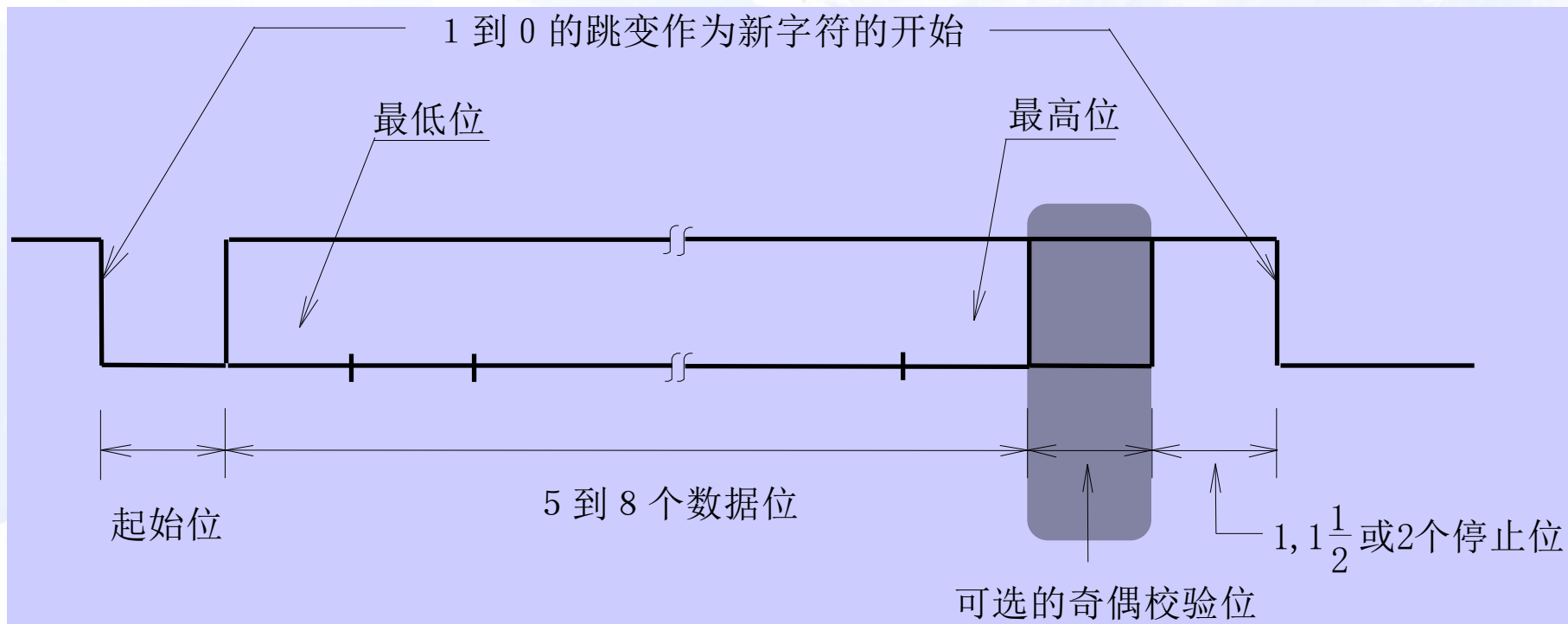
1. 标准数据格式



3) 奇偶校验位: 0~1位 ‘0’或 ‘1’;

三、串行异步通信协议

1. 标准数据格式



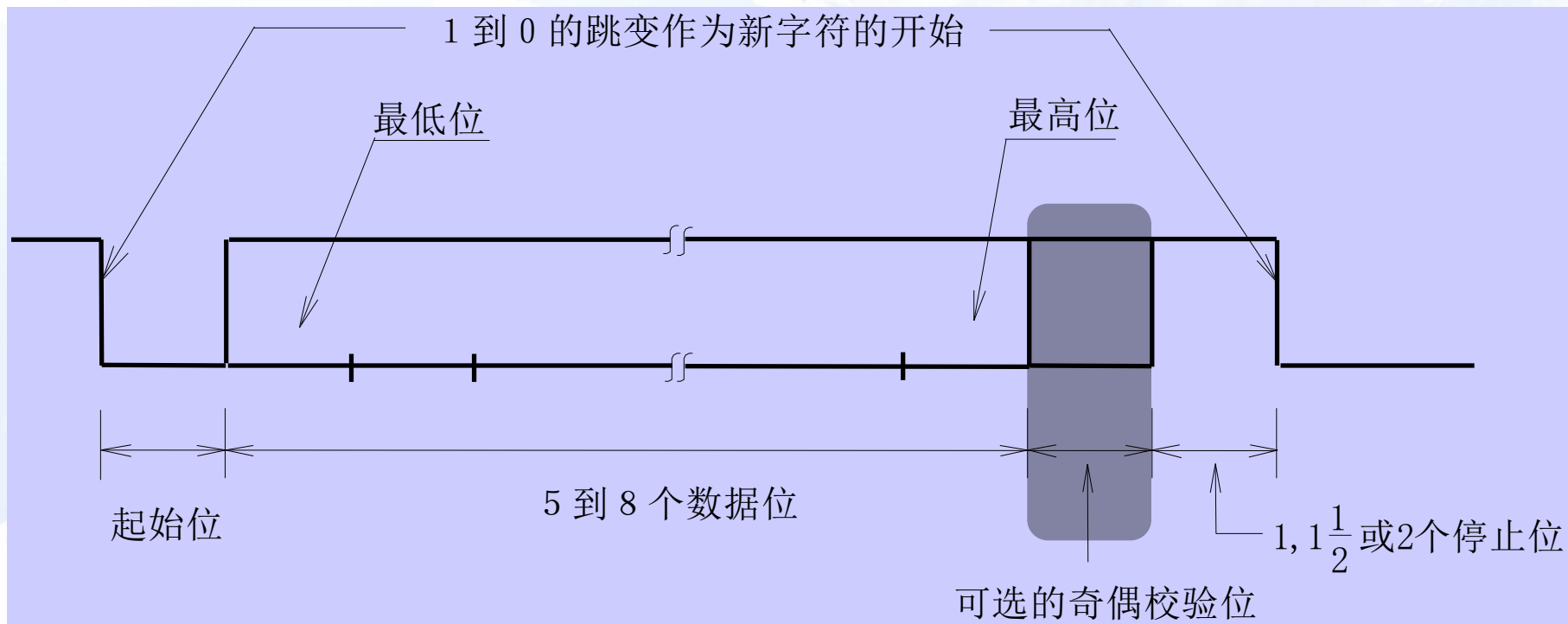
奇偶校验位 (可有可无)

奇校验: 数据位与校验位中 '1' 的个数为奇数。

偶校验: 数据位与校验位中 '1' 的个数为偶数。

三、串行异步通信协议

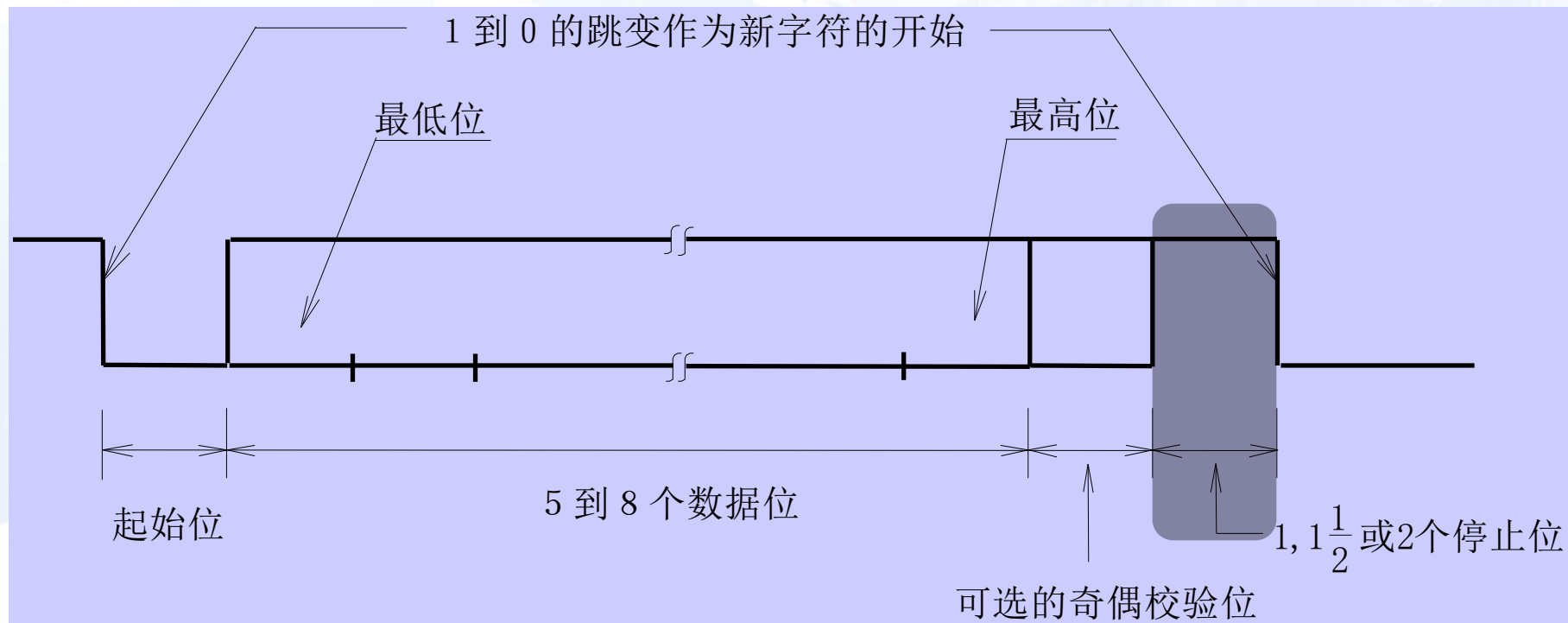
1. 标准数据格式



通信双方约定采用一致的奇偶校验方式，由**发送方设置校验位**，由**接收方负责校验**。

三、串行异步通信协议

1. 标准数据格式



4) **停止位:** 1、1.5、2位**逻辑‘1’**。

9.1 三、串行异步通信协议



例：传送一个字符“E”，7个数据位，奇校验，2个停止位。

E的ASCII码为：45H，7个数据位，即1000101B



例：已知异步串行通信的完整一帧信息为0110010101，其中包括1位停止位，7位数据位和1位校验位，此时传送数据的是（___）₁₆，该数据帧采用的校验方式是_____。

2. 通信速率（传输速率） P321

数据信号速率，又称波特率，表示每秒传送的0、1代码个数（包括起始位、校验位、停止位），单位为“**波特**”。

接下来所涉及的串行数据通信均使用二进制信号传输，所说的通信速率实际是指**数据信号速率**。

异步通信传送中，收、发双方必须约定：

收发双方的帧数据格式必须一致；

收发双方的通信速率必须一致。

设串行异步通信一帧字符有7个数据位，奇校验，2个停止位，波特率为19200bps，则每秒能传输最大字符数是()。

☐ A 1477

☐ B 1600

☒ C 1745

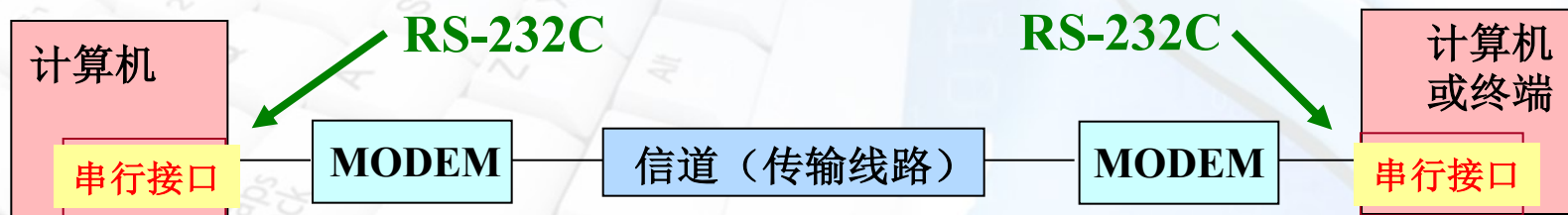
☐ D 1920

3. 串行通信接口标准——RS-232C标准

要进行串行通信，还要解决一个问题：计算机如何通过**串行接口电路**连接？

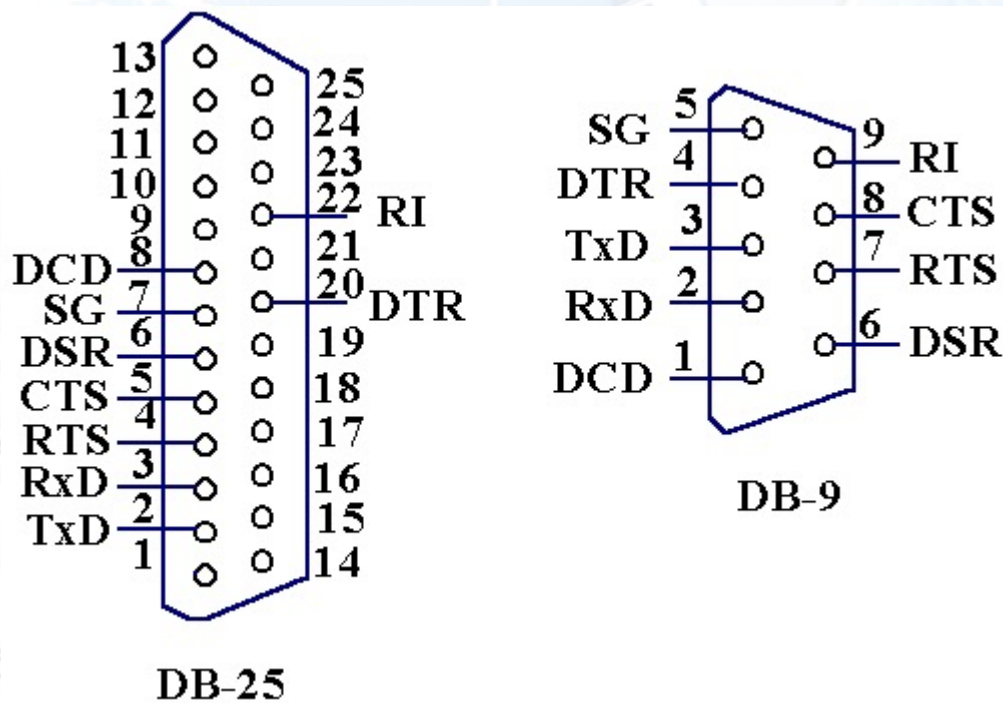
这个接口的**机械特性**、**电气特性**、**功能特性**都要遵循一定的规范，也就是要有一个标准。

目前计算机通信使用最普遍的是**RS-232C标准**。它对两方面作了规定：**信号的电平标准**和**控制信号的定义**。



1) 控制信号的定义（机械特性）

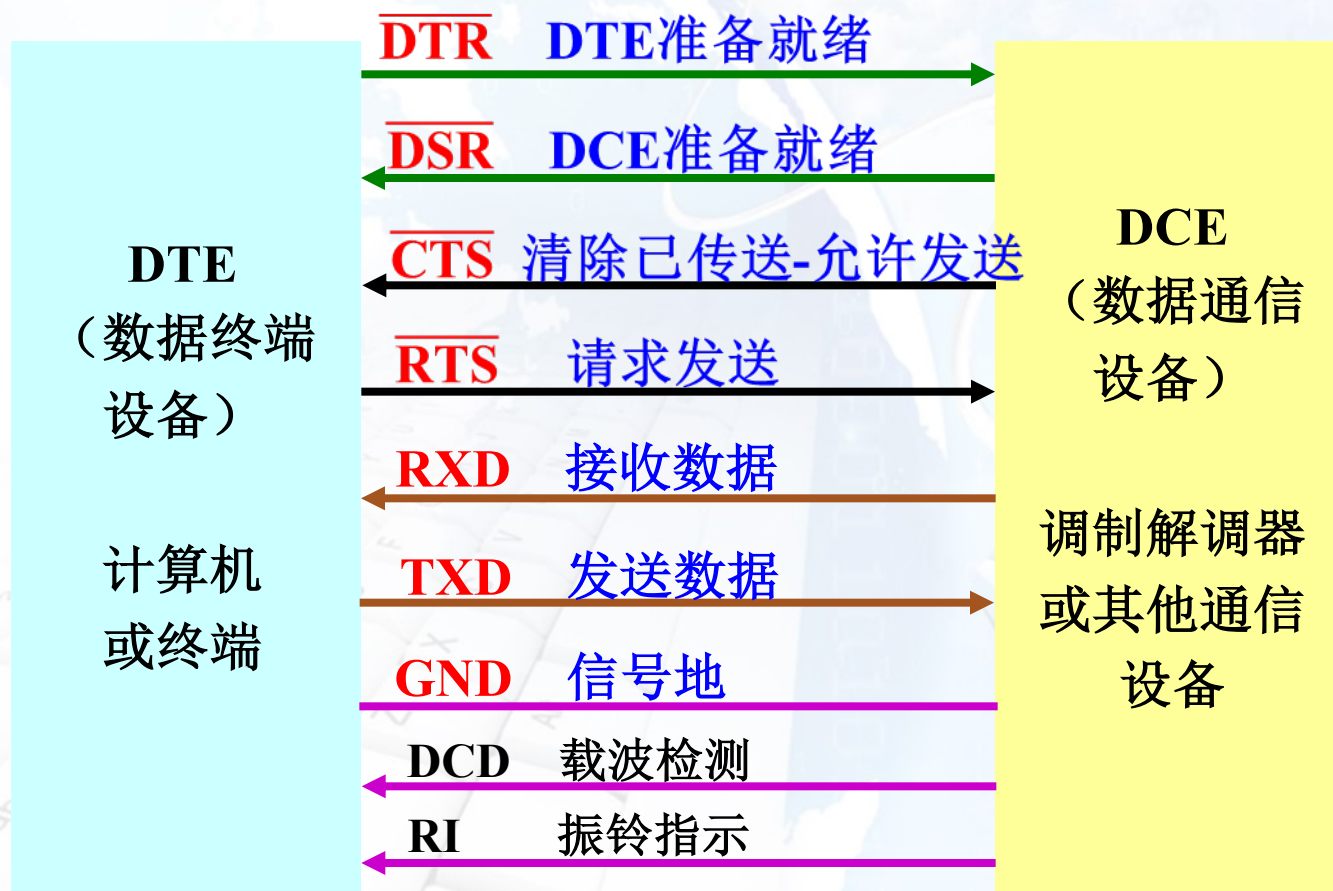
PC系列机有两个串行口：即COM1和COM2，使用9针和25针两种连接器，符合RS-232C接口标准。



9.1 三、串行异步通信协议



功能特性： RS-232C标准定义了25针连接器中的20条连接线，常用的信号线如下：



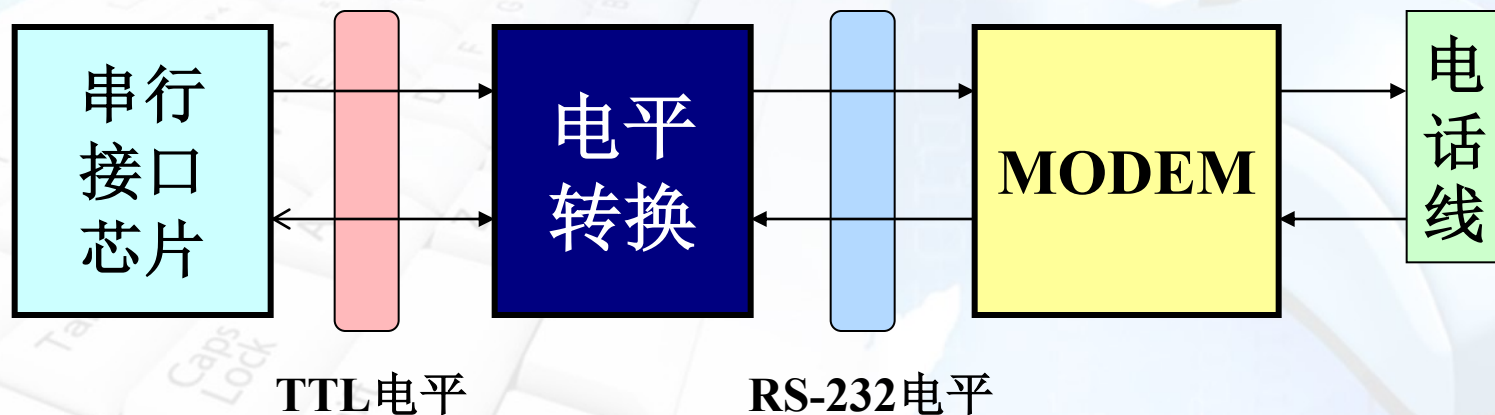
2) 信号电平标准（电气特性）

RS-232C采用**负逻辑**，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；

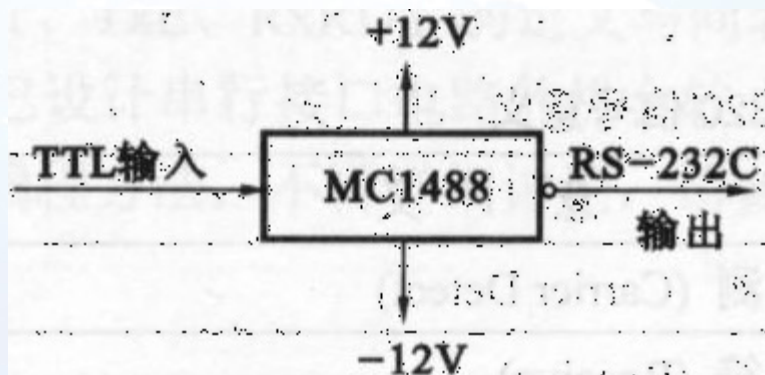
因此，使用RS-232C与微机接口时，需要将TTL电平（ $0 \sim 5V$ ）与RS-232C电平进行转换。



9.1 三、串行异步通信协议

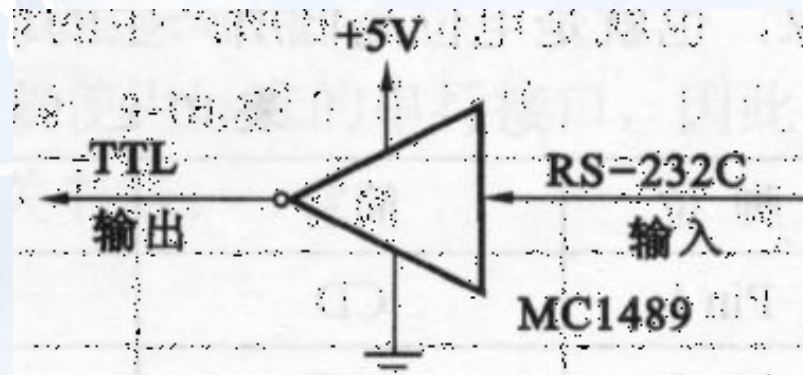


可用现成的转换芯片（如MC1488、MC1489等）转换：



MC1488:

TTL电平→RS232C电平
(用于发送方)




MC1489:

TTL电平←RS232C电平
(用于接收方)

9.2 可编程串行异步通信接口芯片8250



1. 能实现串行数据  并行数据的变换，实现全双工异步通信
2. 支持异步通信协议，数据格式、通信速率由初始化编程设定
3. 内部有MODEM控制器，可直接和MODEM相连
4. 内部有中断机制，CPU可用查询/中断方式与之交换信息

PC机有2个串行口：主串口**3FXH**、辅串口**2FXH**。

高档微机中，8250的功能被一些多功能芯片取代。

对于8250芯片，需要掌握三个知识要点：

8250的内部结构及引脚功能

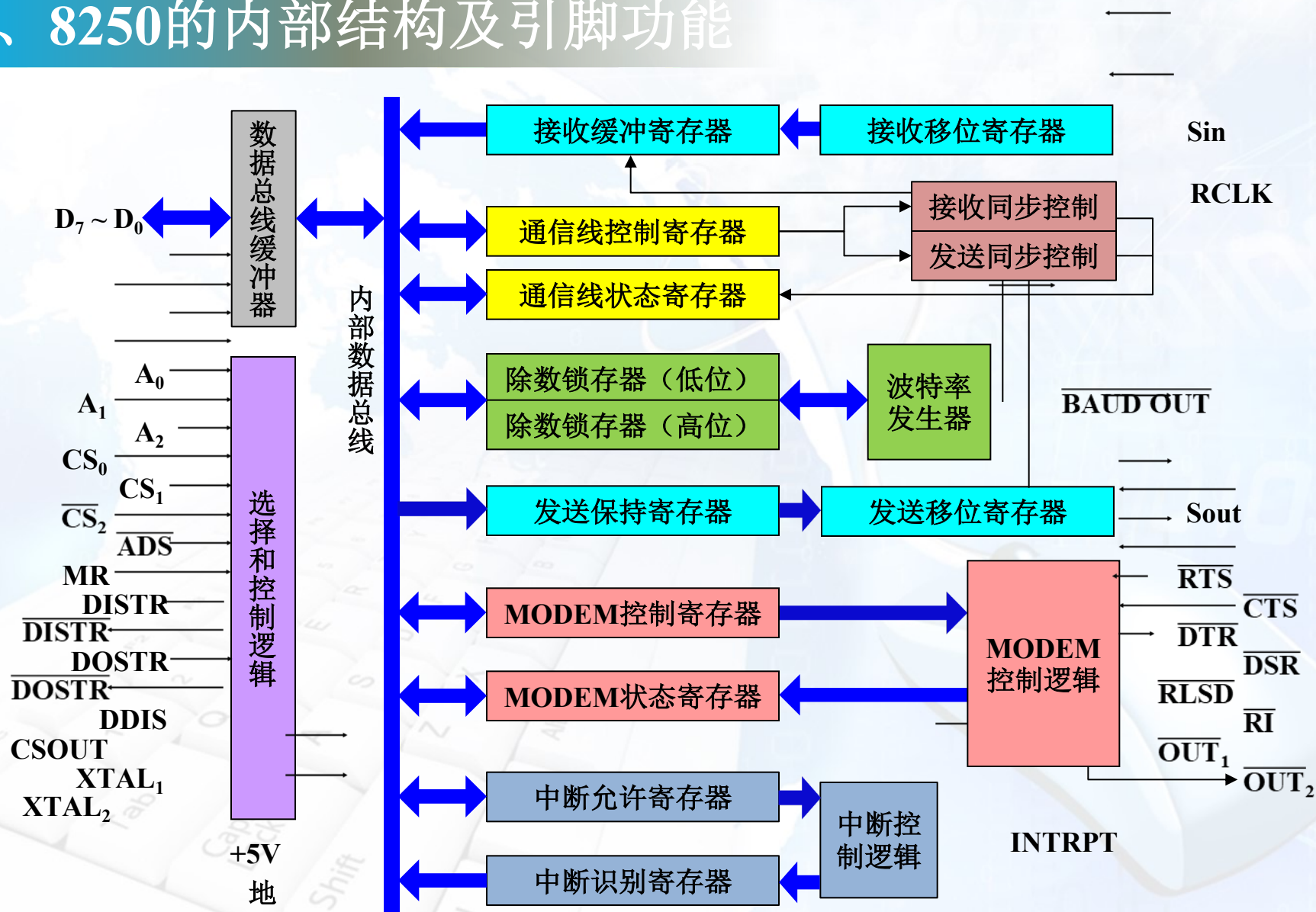
8250的内部寄存器

8250的初始化编程

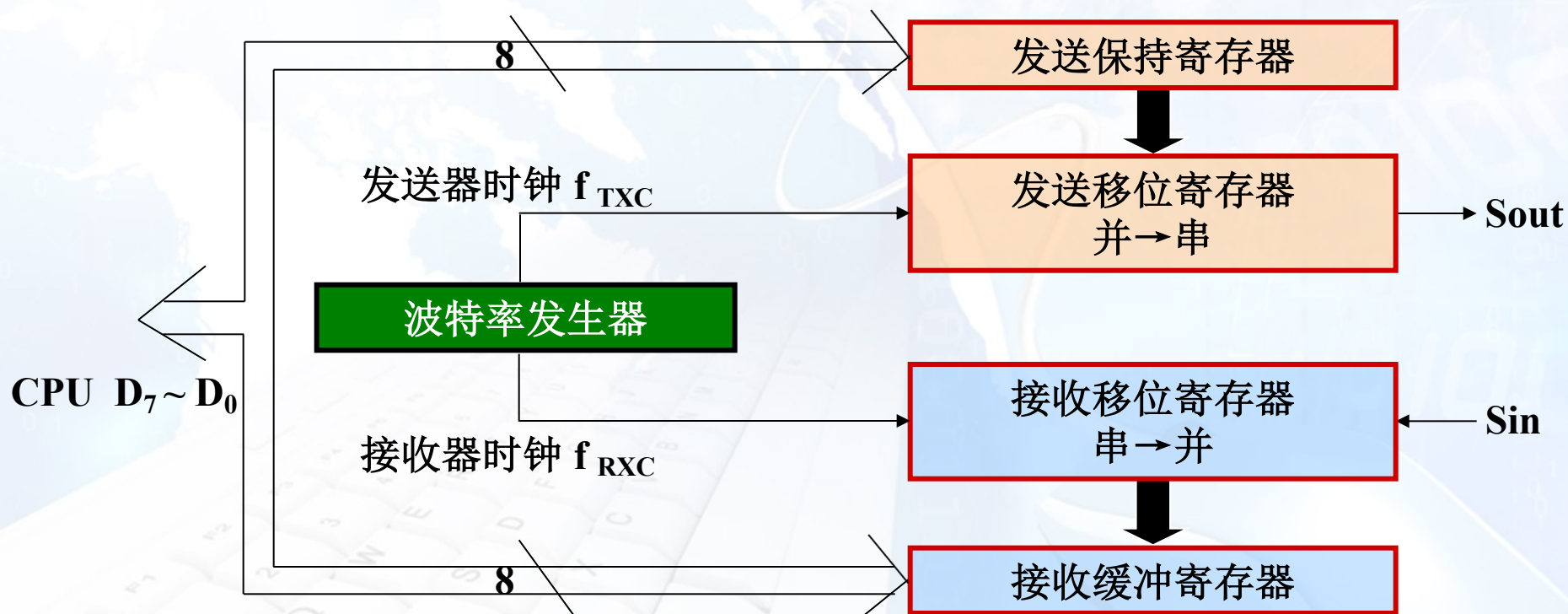
9.2 可编程串行异步通信接口芯片8250



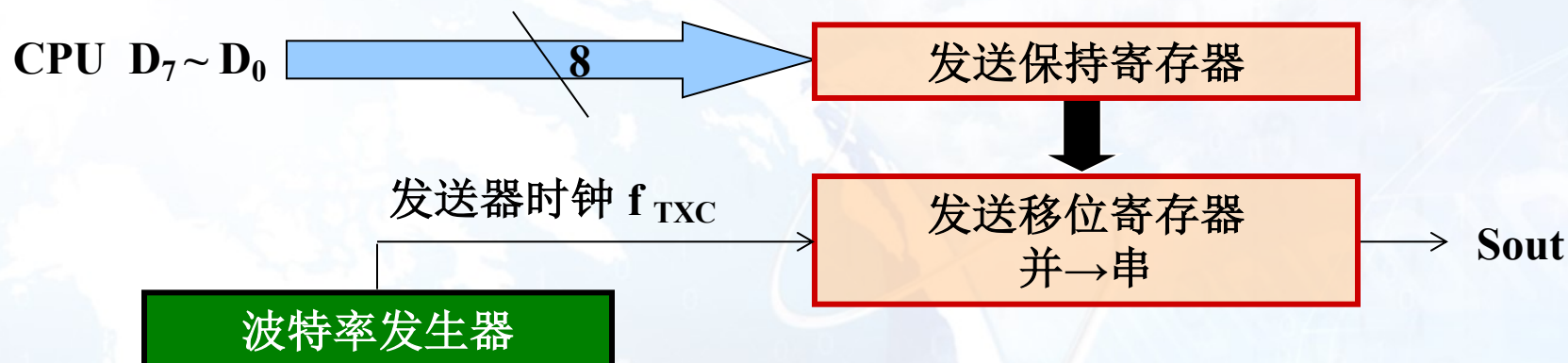
一、8250的内部结构及引脚功能



9.2 一、8250的内部结构及引脚功能



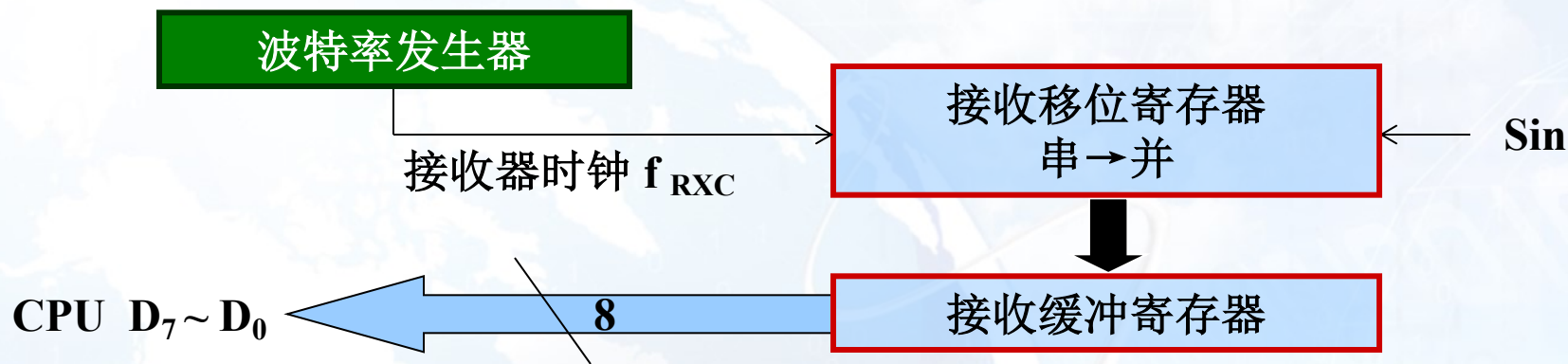
9.2 一、8250的内部结构及引脚功能



串行数据发送过程:

CPU执行OUT指令，将待发送的数据 → 发送保持寄存器暂存，当发送移位寄存器空闲时自动装入，后者在发送器时钟控制下将并行数据添加起始位、校验位、停止位，一位一位发出。

9.2 一、8250的内部结构及引脚功能



串行数据接收过程:

在接收器时钟控制下，接收移位寄存器一位一位地接收串行数据，自动的去掉起始位、校验位、停止位，并转换成并行数据 → 接收缓冲寄存器暂存，在接收过程中对一帧数据自动进行校验。

二、8250的内部寄存器

8250内部寄存器从使用角度来说有**10个**，分成3类：

第一类：2个数据寄存器	在系统机中的口地址	
发送保持寄存器	3F8H/2F8H （写入）	
接收缓冲寄存器	3F8H/2F8H （读出）	
第二类：5个命令字寄存器		
通信线控制寄存器	3FBH/2FBH	
2个除数寄存器	3F8H/2F8H （低位）	3F9H/2F9H （高位）
中断允许寄存器	3F9H/2F9H	
MODEM控制寄存器	3FCH/2FCH	
第三类：3个状态寄存器		
通信线状态寄存器	3FDH/2FDH	
中断识别寄存器	3FAH/2FAH	
MODEM 状态寄存器	3FEH/2FEH	

1. 发送保持寄存器 (3F8H/2F8H)

该寄存器保存CPU传送来的并行数据，并转移至发送移位寄存器。

注：只有在发送保持寄存器空闲时，CPU才能写入下一个数据。

2. 接收缓冲寄存器 (3F8H/2F8H)

接收移位寄存器将接收到的数据去掉起始位、校验位和停止位，转换成并行数据，转换后的并行数据存入接收缓冲寄存器，等待CPU接收。

注：只有当一帧数据收完后，CPU才能用IN指令读接收缓冲寄存器。

3. 通信线状态寄存器 (3FDH/2FDH)

该寄存器提供数据传输的状态信息，其各位含义如下：

D₀位：接收数据准备好（接收缓冲器满）标志位。

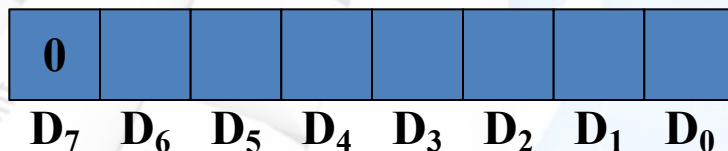
D₀=1，表示接收器已接收到一帧完整的数据，并已转换成并行数据，存入接收缓冲寄存器（CPU接收数据时一定要查询）。

D₁位：溢出错标志位。

D₁=1，表示接收缓冲器中的字符未取走，8250又接收到新输入的数据，造成前一数据被破坏。

D₂位：奇偶错标志位。

D₂=1，表示收到的数据有奇偶错。



D₃位：接收格式错标志位。

D₃=1，表示接收数据没有正确的停止位。

D₄位：线路间断标志位。

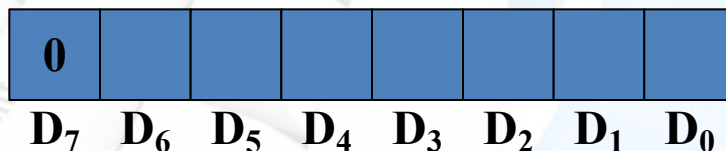
D₄=1，表示收到长时间“0”信号，即中止信号。

D₅位：发送保持寄存器空闲标志位。

D₅=1，表示数据已经从发送保持寄存器转移到发送移位寄存器，发送保持寄存器空闲，**CPU可以写入新数据**，当新数据送入发送保持寄存器后，**D₅置0**。

D₆位：发送移位寄存器空闲标志位。

D₆=1，表示一帧数据已**发送完毕**。当下一个数据由发送保持寄存器移入发送移位寄存器时，该位被置0。

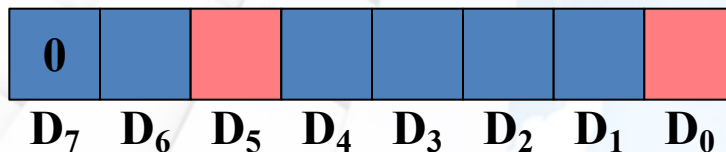


注:

D₀位（接收数据准备好）和**D₅**位（发送保持寄存器空）是串行接口最基本的标志位，它们决定了CPU能不能向8250进行读写操作！

只有当**D₀=1**时，CPU才能读数据；

只有当**D₅=1**或**D₆=1**时，CPU才能写数据。



程序段示例

EG: 利用主串口查询方式发送一个“A”

```
SCANT: MOV DX, 3FDH
        IN  AL, DX
        TEST AL, 20H ; 00100000B
        JZ  SCANT ; D5=1

        MOV DX, 3F8H
        MOV AL, 'A'
        OUT DX, AL
```

EG: 利用辅串口查询方式接收一个字符

```
SCANR: MOV DX, 2FDH
        IN  AL, DX
        TEST AL, 00011110B
        JNZ ERROR

        TEST AL, 01H ; 00000001B
        JZ  SCANR ; D0=1

        MOV DX, 2F8H
        IN  AL, DX
```

4. 中断允许寄存器 (3F9H/2F9H)

8250 有
4 级中断

- 接收线路状态中断（接收数据错 OE、PE、FE、BI 置1 产生中断）；
- 接收缓冲器满；
- 发送缓冲器空；
- MODEM 状态中断（MODEM应答联络信号状态变化产生中断）；

优先级

高



低

当上述4种类型的中断有一种或多种出现时，8250便输出
INTRPT信号。

说明：寄存器中的各位置1时，如相应的中断允许位也为1，则：

$D_0=1$ ，产生接收中断；

$D_5=1$ ，产生发送中断；

$D_1\sim D_4=1$ ，产生“字符错”中断。

执行下列操作后，寄存器中的相应位自动执行：

数据→发送保持寄存器

$D_5=0$

通信线状态寄存器内容→AL

$D_1\sim D_4=0$

接收缓冲寄存器内容→AL

$D_0=0$

发送保持寄存器内容→移位寄存器

$D_6=0$

9.2 二、8250的内部寄存器



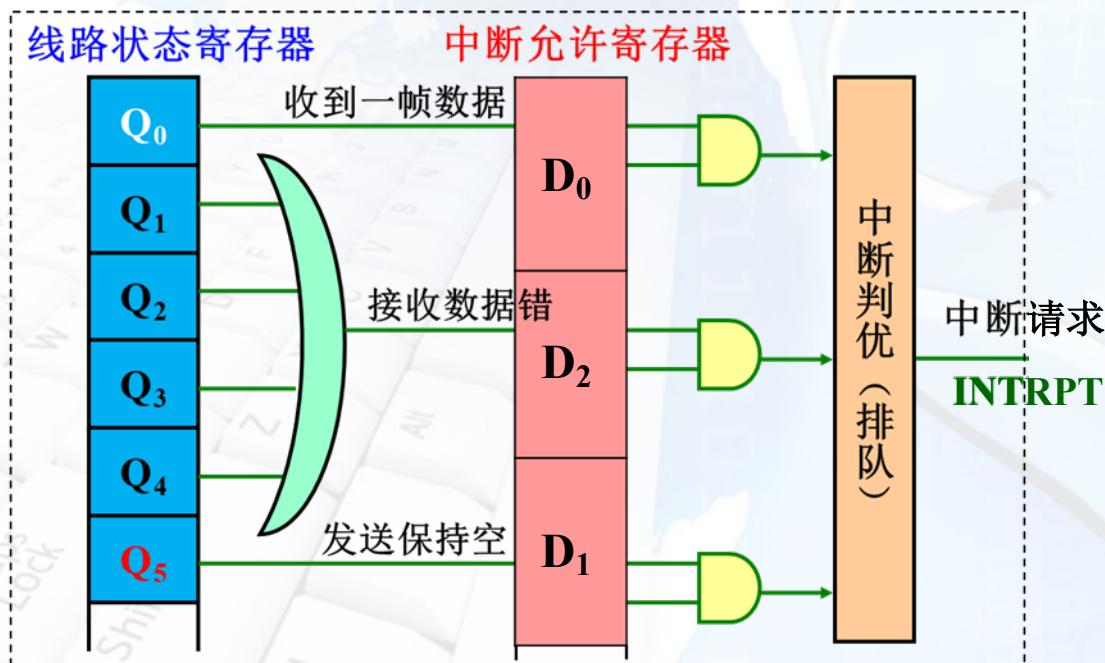
该寄存器的 $D_7 \sim D_4$ 位恒为0， $D_3 \sim D_0$ 位表示8250的4级中断是否被允许：

$D_0=1$ ，允许接收到一帧数据后，内部提出“接收中断请求”。

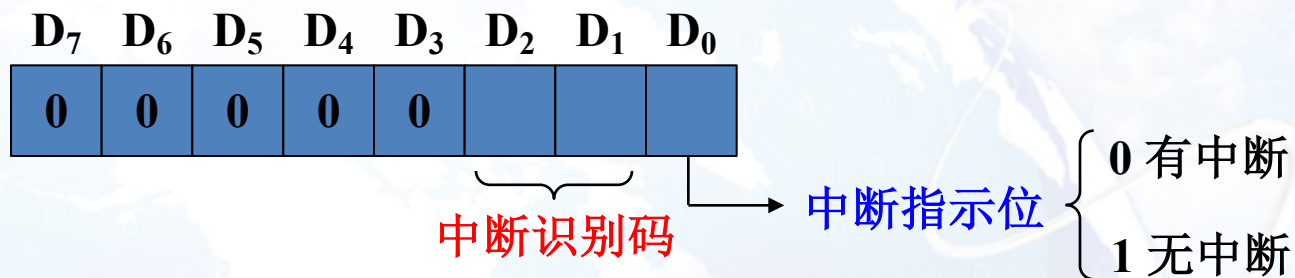
$D_1=1$ ，允许发送保持寄存器空时，内部提出“发送中断请求”。

$D_2=1$ ，允许接收出错时，内部提出“接收数据错中断请求”。

$D_3=1$ ，允许MODEM状态改变时，内部提出“MODEM中断请求”。



5. 中断识别寄存器 (3FAH/2FAH)



由于8250仅能向CPU发出一个总的中断请求信号，为了识别是哪一個中断源引起的中断，应当在转入中断服务程序之后，读取中断识别寄存器的内容进行判断，然后再转入相应的处理程序。

MOV	DX, 3FAH
IN	AL, DX
CMP	AL, 0
JZ	MODEM中断
CMP	AL, 2
JZ	发送中断
CMP	AL, 4
JZ	接收中断
CMP	AL, 6
JZ	字符错中断

6. MODEM控制寄存器 (3FCH/2FCH)

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	0	0					

D₇~D₅位恒为0。

D₀位=1，使引脚 $\overline{\text{DTR}} = 0$ ，从而使RS-232C引脚 $\overline{\text{DTR}}$ 为0。

D₁位=1，使引脚 $\overline{\text{RTS}} = 0$ ，从而使RS-232C引脚 $\overline{\text{RTS}}$ 为0。

D₂位=1，使引脚 $\overline{\text{OUT}}_1 = 0$ ，系统机上没有使用。

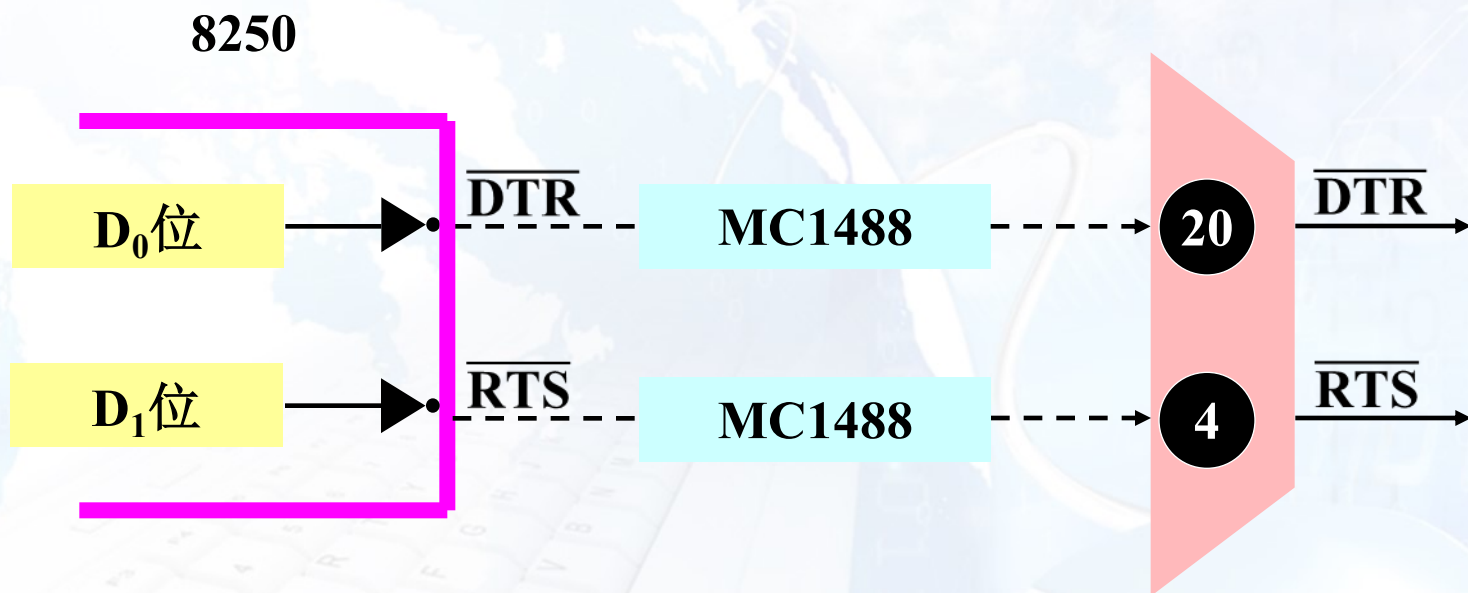
D₃位=1，使引脚 $\overline{\text{OUT}}_2 = 0$ ，8250能送出中断请求。

D₄位通常置0，设置8250工作在正常收/发方式；

若D₄位置1，则8250工作在内部自环方式，即发送移位寄存器的输出在芯片内部被回送到接收移位寄存器的输入。

利用这个特点，可以编写程序测试8250的工作是否正常，而不需任何附加装置。

说明 (1)

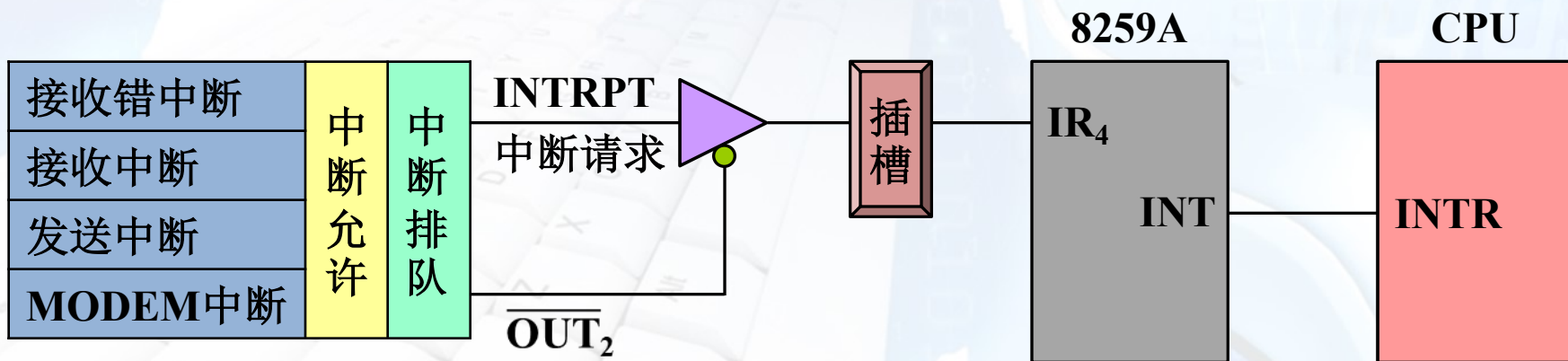


D_0 、 D_1 位直接控制RS232C的 \overline{DTR} 和 \overline{RTS} 引脚，**向外部表明8250是否做好了收/发的准备。**

说明（2） 引脚 $\overline{\text{OUT}}_1$ ， $\overline{\text{OUT}}_2$ 外接器件由硬件决定。

现状： $\overline{\text{OUT}}_1$ 空，在PC机中 $\overline{\text{OUT}}_1$ 引脚没有使用，因此初始化时 $\text{D}_2=0/1$ 皆可。

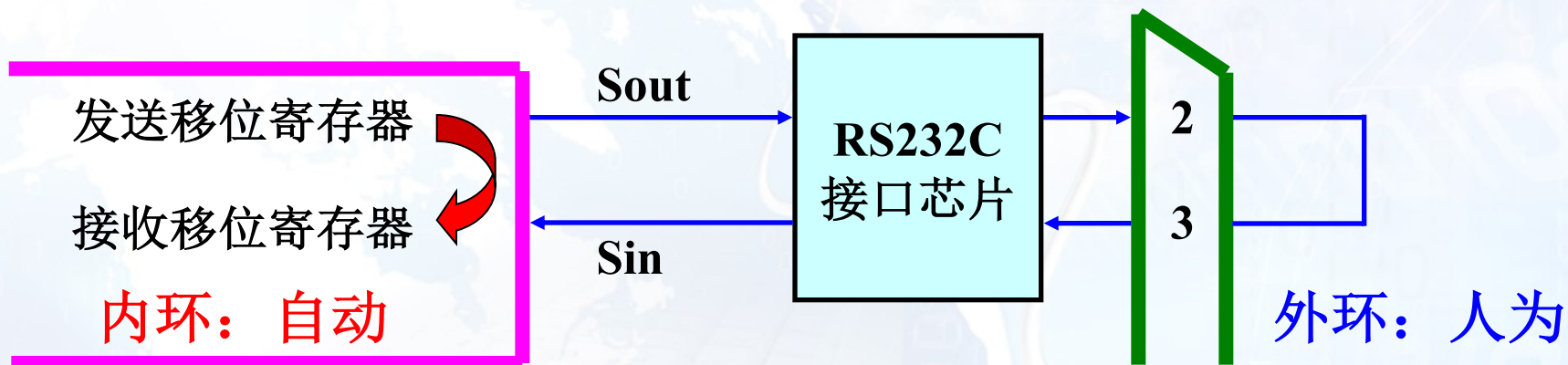
$\overline{\text{OUT}}_2$ 接线如下（以主串口为例）：



说明（3）与8250之间采用中断方式交换信息，应采取以下措施：

- 1) 中断允许寄存器相应位置1;
- 2) MODEM控制寄存器 $D_3=1$ ，即 $\overline{OUT}_2=0$ ，打通
8250的中断请求通道;
- 3) 8259A相应中断屏蔽位开放（主8259A IR_3 ， IR_4 ）；—— 8259A
- 4) CPU处于开中断（STI）。—— CPU

说明（4）8250的自发自收：

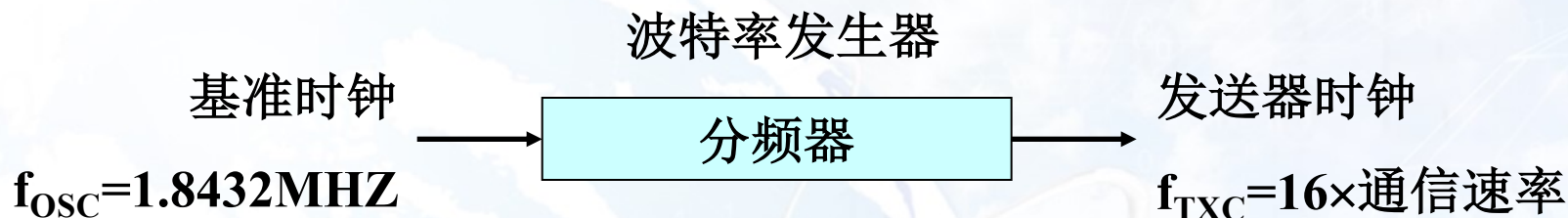


内环：自动将发送移位寄存器和接收移位寄存器短接。

外环：人为将RS-232C接口芯片端子2和端子3用线连接起来，属正常通信。

7. 除数寄存器（高8位3F9H/2F9H、低8位3F8H/2F8H）

功能：存放初始化时编程时写入的分频系数



8250内部的波特率发生器是软件控制的可编程的分频器。

$$\frac{f_{osc}}{\text{分频系数}} = 16 \times \text{通信速率}$$

∴

$$\text{分频系数（即除数）} = \frac{1.8432\text{MHZ}}{16 \times \text{通信速率}}$$

如果接收器时钟和发送器时钟相同，只需把8250芯片的引脚9（RCLK）和引脚15（BAUD OUT）相连即可。

16被称为：波特率因子，可选1、16、64。

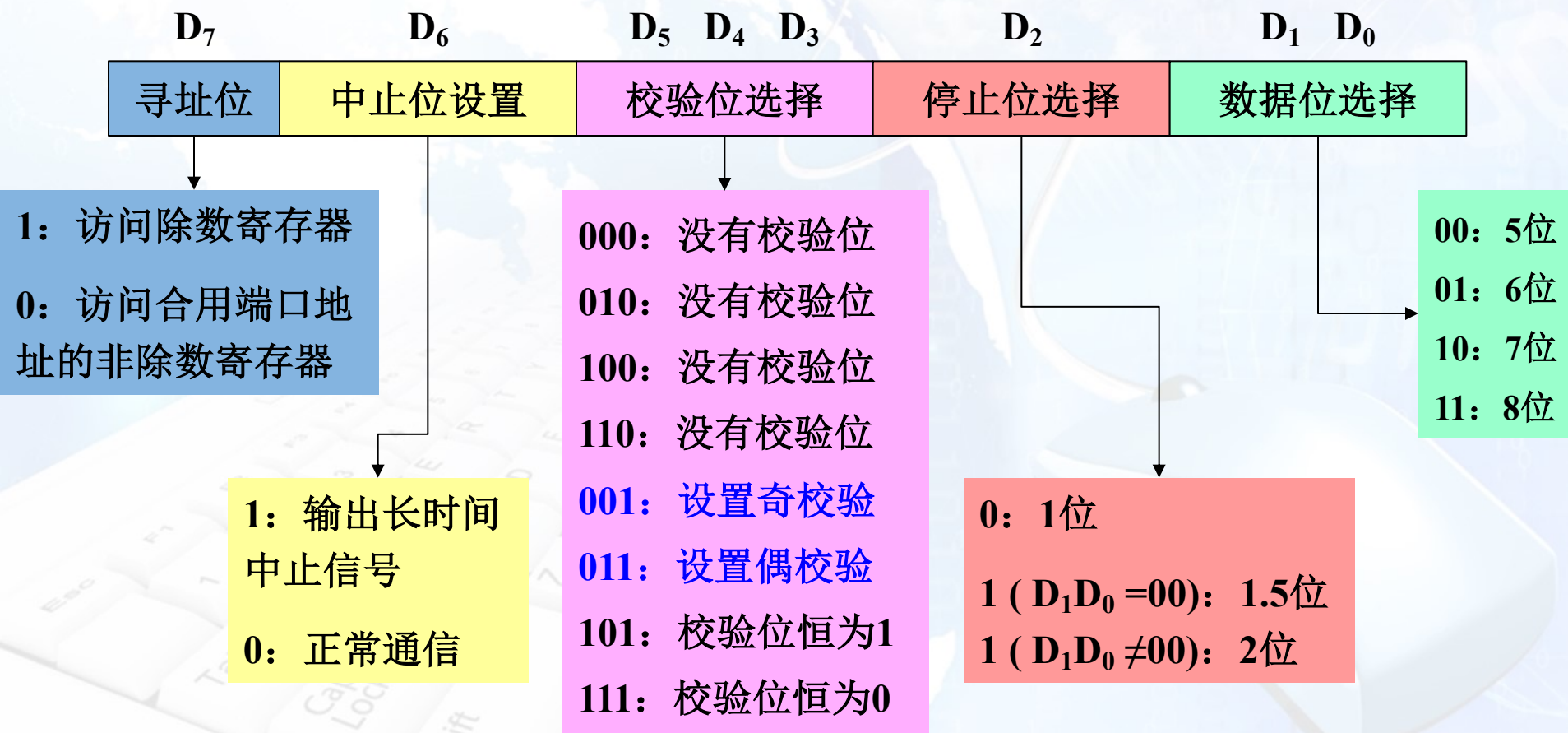
除数寄存器存放分频系数，由高8位和低8位两个寄存器组成，分两次写入。**8250** 波特率与分频系数（除数锁存器中的值）的关系：

波特率	除数高8位	除数低8位
50	09H	00H
75	06H	00H
110	04H	17H
150	03H	00H
300	01H	80H
600	00H	C0H
1200	00H	60H

波特率	除数高8位	除数低8位
1800	00H	40H
2000	00H	3AH
2400	00H	30H
3600	00H	20H
4800	00H	18H
7200	00H	10H
9600	00H	0CH

8. 通信线控制寄存器 (3FBH/2FBH)

该寄存器主要用于指定串行异步通信的数据格式:



问题：设置寻址位？

$D_7=1$ 表明后继写入合用端口的数据写入除数寄存器。

$D_7=0$ 表明后继写入合用端口的数据写入非除数寄存器。

寄存器	合用口地址
除数寄存器高8位 中断允许寄存器	3F9H/2F9H
除数寄存器低8位 发送保持寄存器 接收缓冲寄存器	3F8H/2F8H

MOV DX, 3F9H

OUT DX, AL

写入哪个寄存器？

EG: MOV DX, **3FBH**

MOV AL, **80H** ; **10000000B**

OUT DX, AL

MOV DX, **3F9H**

MOV AL, **N**

OUT DX, AL

∴ **N**写入除数寄存器高8位

EG: MOV DX, **3FBH**

MOV AL, **01111111B**

OUT DX, AL

MOV DX, **3F8H**

MOV AL, **N**

OUT DX, AL

∴ **N**写入发送保持寄存器

9. MODEM状态寄存器 (3FEH/2FEH)

该寄存器反映8250与通信设备（如MODEM）之间联络信号的当前状态以及这些信号的变化情况。

$D_7 \sim D_4$ 记录了4个输入引脚的状态电平：

$D_7=1$ 表示输入引脚 $\overline{RLSD}=0$ ，MODEM收到来自电话线的载波信号。

$D_6=1$ 表示输入引脚 $\overline{RI}=0$ ，MODEM收到振铃信号。

$D_5=1$ 表示输入引脚 $\overline{DSR}=0$ ，MODEM做好了发送准备，请8250准备接收。

$D_4=1$ 表示输入引脚 $\overline{CTS}=0$ ，MODEM做好了接收准备，8250可以发送数据。

$D_3 \sim D_0$ 记录了上一次读取该寄存器后，上述引脚是否发生过电平变化：

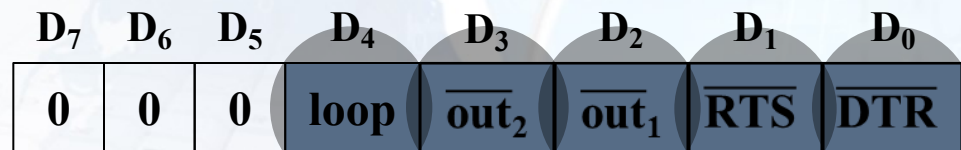
$D_3=1$ 表示输入引脚 \overline{RLSD} 有电平变化。 $D_2=1$ 表示输入引脚 \overline{RI} 有电平变化。

$D_1=1$ 表示输入引脚 \overline{DSR} 有电平变化。 $D_0=1$ 表示输入引脚 \overline{CTS} 有电平变化。

三、8250的初始化编程

1. 直接对8250端口进行初始化编程步骤:

- ① 确定波特率——设置除数寄存器;
- ② 确定数据格式——设置通信线路控制寄存器;
- ③ 若使用中断方式——需设置中断允许寄存器的相应位 (置“1”);
- ④ 设置MODEM控制寄存器:



D₄=0: 正常通信
D₄=1: 内环自检

PC机未用

D₀=1: 该
信号有效

D₃=1: 允许8250中断信号通过系统总线送至8259A中断控制器
D₃=0: 不允许送出中断信号

D₁=1: 该
信号有效

通常, 该寄存器的值设置为**03H**——使8250输出 $\overline{\text{RTS}}$ 、 $\overline{\text{DTR}}$ 两个MODEM控制信号, 即使系统中不用这两个信号, 这样的设置也不会带来问题。

9.2 三、8250的初始化编程



例：编写子程序，对PC系列机主串口进行初始化，要求：

- ① 通信速率=1200波特，一帧数据包括：8个数据位，1个停止位，无校验。
- ② 查询方式，完成内环自检。

分析：

1) 通信速率 = 1200

分频系数 = $1.8432\text{M} / (16 \times 1200) = 0060\text{H}$ ，（也可查表得

到）

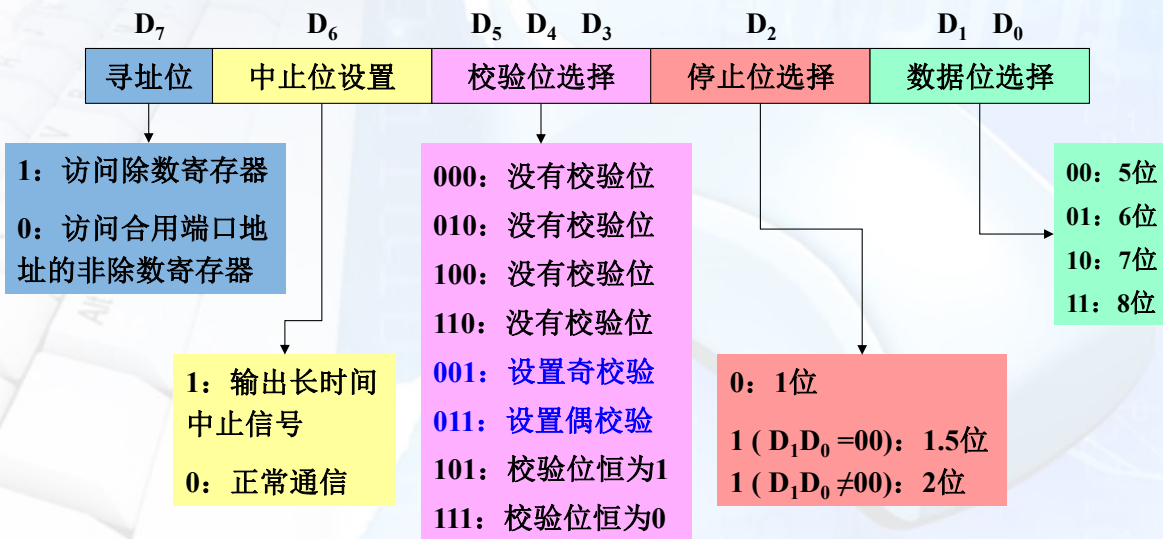
2) 帧数据结构控制字：

00000011B = 03H

3) 中断允许控制字：0

4) MODEM控制字：

00010000B = 10H



9.2 三、8250的初始化编程



I8250 PROC

MOV DX, 3FBH
MOV AL, 80H
OUT DX, AL

} 寻址位=1

MOV DX, 3F9H
MOV AL, 0
OUT DX, AL
MOV DX, 3F8H
MOV AL, 60H
OUT DX, AL

} 设置分
频系数
=0060H

MOV DX, 3FBH
MOV AL, 03H
OUT DX, AL

} 定义一帧
数据格式

MOV DX, 3F9H
MOV AL, 0
OUT DX, AL

} 设置中断允
许寄存器

MOV DX, 3FCH
MOV AL, 10H
OUT DX, AL

} 设置
MODEM控
制寄存器

RET

I8250 ENDP

9.2 三、8250的初始化编程



例：要求以9600bps进行异步串行通信，每个字符 7 位数据位， 2位停止位，奇校验，允许所有中断。

假设 端口地址的高位为： 0 0 1 1, 1 1 1 1, 1 $A_2 A_1 A_0$

MOV DX, 3FBH ; 设置除数寄存器

MOV AL, 80H (分频系数), 通

OUT DX, AL 讯线控制寄存器最

MOV DX, 3F8H 高位置“1”

MOV AL, 0CH

OUT DX, AL ; 除数低8位

MOV DX, 3F9H

MOV AL, 0

OUT DX, AL ; 除数高8位

MOV DX, 3FBH ; 设置数据格式:

MOV AL, 00001110B 7 位数据位, 2位

OUT DX, AL 停止位, 奇校验

MOV DX, 3F9H ; 设置中断允许

MOV AL, 0FH 寄存器: 允许所

OUT DX, AL 有中断

MOV DX, 3FCH ; 设置MODEM

MOV AL, 0BH 控制器: 使

OUT DX, AL $\overline{OUT_2}$ 、 \overline{RTS} 、 \overline{DTR} 有效

2. BIOS通信软件

BIOS通过INT 14H向用户提供了4个中断子程序，分别完成：串口初始化编程、发送数据、接收数据、测试通信线状态。

功能	入口参数	出口参数
1) AH=0: 初始化串口	AL = 初始化参数 DX = 串口号 (0/1)	AH = 通信线 路状态 AL = MODEM 状态

0号子功能的执行流程：

- ① 截取AL₇~AL₅查表，取出相应波特率除数→除数寄存器。
- ② 截取AL₄~AL₀→通信线控制寄存器。
- ③ 0→中断允许寄存器。

AL中需设置的参数为：

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
波特率：			奇偶校验：		代码长：		
000=110			00=无		00=5位		
001=150			01=奇校验		01=6位		
010=300			10=无		10=7位		
011=600			11=偶校验		11=8位		
100=1200							
101=2400							
110=4800							
111=9600							
					停止位：		
					0=1位		
					1=1.5位（代码长5位）		
					=2位（代码长6-8位）		

- ④ 取通信线状态寄存器内容→AH。
- ⑤ 取MODEM状态寄存器内容→AL。
- ⑥ 执行IRET返回。

注

调用注意事项:

调用INT 14H的0号子功能初始化串行口，通信波特率只有8种选择，奇偶校验也只有3种选择。

从执行流程可以看出，使用0号功能初始化之后，8250的**内部中断将被禁止**，**只能采用查询方式**发送和接收数据。

如果在0号子功能初始化之后，再对中断允许寄存器和MODEM控制寄存器写入相应的命令字，仍然能使其工作在中断方式。

功能

2) **AH=1**: 发送数据

3) **AH=2**: 接收数据

4) **AH=3**: 测试通信线状态

入口参数

AL = 待发送数据

DX = 串口号 (0/1)

DX = 串口号 (0/1)

DX = 串口号 (0/1)

出口参数

AH₇ = 1: 表示发送失败

AH₇ = 0: 表示发送成功

AH₇: 同上

AH₄~AH₁: 接收数据的错误标志

AL = 接收到的数据

AH = 通信线路状态

AL = MODEM状态

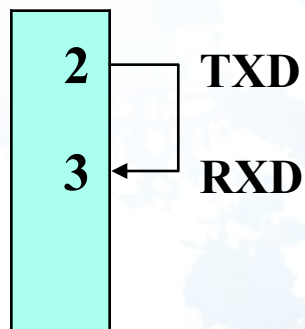
一、串行通信的外部环境

使用**BIOS**通信软件，在发送和接收之前都要使用联络线与对端联络，只有联络畅通，才允许发送或接收数据。

因此，当使用**BIOS**通信程序发送和接收数据时，必须具备相应的外部环境，所谓外部环境，就是**RS-232C**引脚的连接方式。

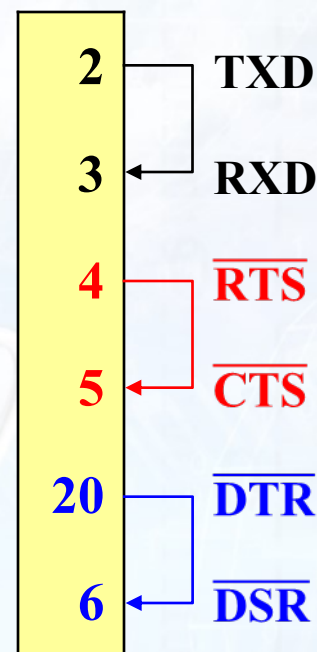
RS-232C怎样连线，与串口的通信方式（单工、半双工 or 全双工）有关，与编程时使用的编程手段（对端口直接操作 or 调用**BIOS 1#、2#**）有关。

1. 外环自发自收



无联络线

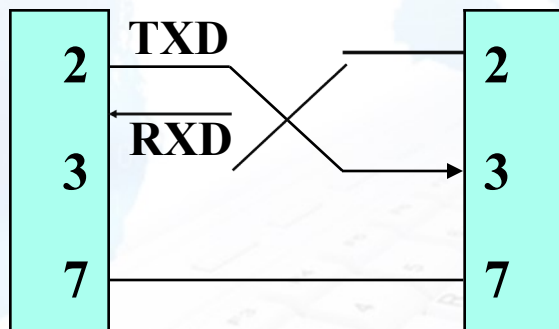
适用：对端口直接操作编程。



有联络线

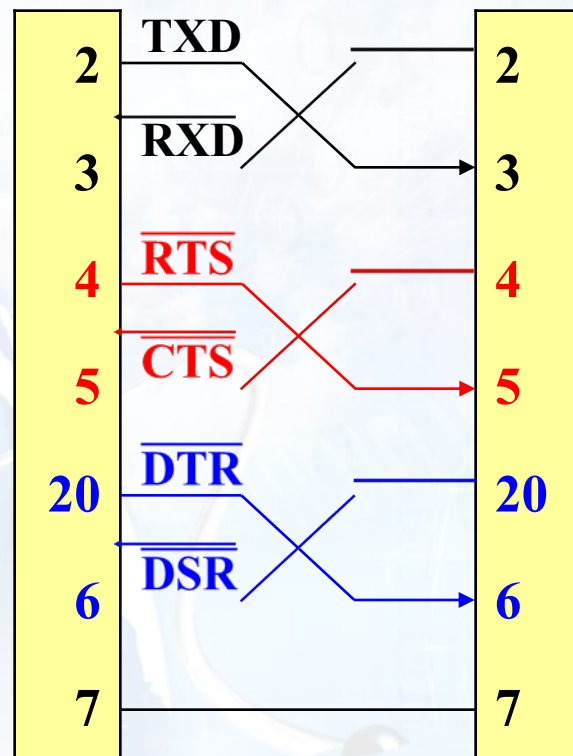
适用：对端口直接操作编程；
调用BIOS通信软件；
调用INT 21H的4#、3#
(发/收)。

2. 短距离（无MODEM） 点-点全双工通信



无联络线：3线方式

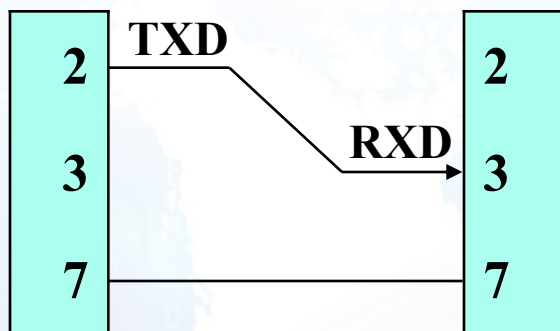
适用：仅能对端口直接操作编程。



有联络线：7线方式

适用：对端口直接操作；
调用BIOS通信软件；
调用INT 21H的4#、3#（发/收）——只能用主串口。

3. 短距离单工通信



无联络线：2线方式

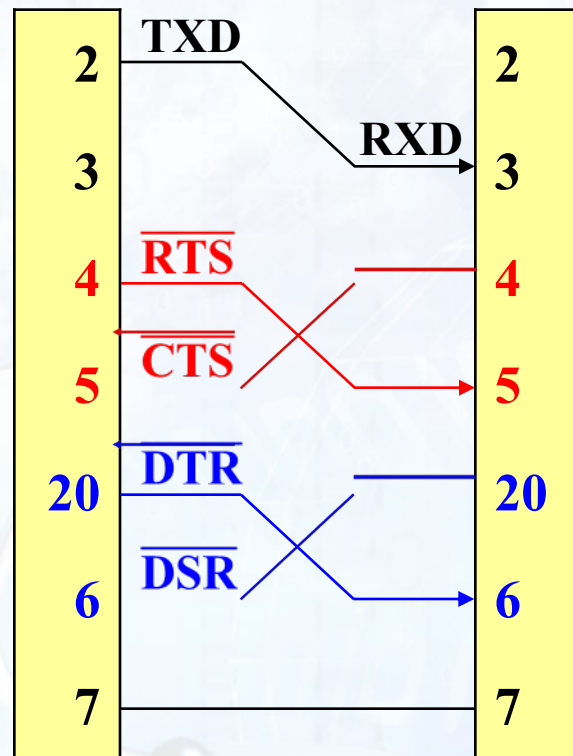
适用：仅能对端口直接操作编程。

有联络线：
6线方式

适用：

对端口直接操作；
调用BIOS通信软件；
调用INT 21H的4#、3#（发/收）——只能用主串口。

注意：调用BIOS通信软件实现单工通信时，在初始化阶段应使发方的 $\overline{\text{DTR}} = 0$ ，收方的 $\overline{\text{RTS}} = \overline{\text{DTR}} = 0$ 。



二、串行通信程序设计举例

设计考虑：

- ① 题型：单端自发自收 or 点一点通信（全双工、单工）；
- ② CPU与串口交换信息的方式：查询 or 中断？
- ③ 编程手段：对端口直接编程 or 调用BIOS通信软件；
- ④ 根据题目要求组织相应的外部环境。

说明：

1) 对端口直接操作发送和接收数据，程序中查询联络线，可以按“有联络线”方式接线；程序中不查询联络线，可以按“无联络线”方式接线。

2) BIOS通信软件是一个全双工的通信软件，发送和接收之前都要使用联络线与对端“握手”，只有联络畅通，才能发送或接收数据。

3) 当8250设置为内环自检方式的时候，只能采用查询方式，而且只能采用对端口直接操作的编程手段，完成数据的发送和接收。

例1: A、B两机利用主串口，查询方式，进行单工通信，A机发送电文“HELLO”至B机。

试为A机编写发送程序，要求：波特率=2400，奇校验，停止位1位，数据位7位，采用查询方式。

分析：

1) 通信速率=2400，分频系数

= $1.8432\text{M} / (16 \times 2400) = 0030\text{H}$

2) 帧数据结构命令字: 00001010B

3) 中断允许命令字 = 0

4) MODEM控制字 $00000011\text{B} = 03\text{H}$

```
DATA    SEGMENT USE16
BUF      DB      'HELLO'
LENS     EQU      $-BUF
DATA     ENDS
CODE     SEGMENT USE16
ASSUME   CS:CODE , DS:DATA
BEG:     MOV      AX,  DATA
         MOV      DS,  AX
```


CALL I8250 

LEA BX, BUF

MOV CX, LENS

SCAN: MOV DX, 3FDH

IN AL, DX

TEST AL, 20H

JZ SCAN

MOV DX, 3F8H

MOV AL, [BX]

OUT DX, AL

INC BX

LOOP SCAN

NEXT: MOV DX, 3FDH

IN AL, DX

TEST AL, 40H

JZ NEXT

MOV AH, 4CH

INT 21H

I8250 PROC

MOV DX, 3FBH

MOV AL, 80H

OUT DX, AL

MOV DX, 3F9H

MOV AL, 0

OUT DX, AL

MOV DX, 3F8H

MOV AL, 30H

OUT DX, AL

MOV DX, 3FBH

MOV AL, 00001010B

OUT DX, AL

MOV DX, 3F9H

MOV AL, 0

OUT DX, AL

MOV DX, 3FCH

MOV AL, 03H

OUT DX, AL

RET ◀

I8250 ENDP

CODE ENDS

END BEG

例2：要求甲乙两台微机之间通过**RS-232C**接口进行短距离的串行通信。

甲机作为发送端从主串口将一串字符逐个发送，乙机作为接收端在辅串口进行字符接收并显示在屏幕上，其中甲机发送字符串时以 ‘**ETX**’ 字符（**ASCII码：03H**）作为输入结束标志字符。

通信双方约定波特率为**2400**波特，数据位**7**位，停止位**1**位，奇校验，发送和接收均采用查询方式。

[设计思路]

1) 甲乙两台微机之间进行短距离的单工通信，因此采用直接访问8250端口寄存器的方式进行编程，程序运行前将甲机的主串口与乙机的辅串口进行点到点单工通信连接，不使用联络线。连线图如下所示。



2) 一串字符经逐个发送，发送字符和接收字符均采用查询方式。发送前，先读取通信线状态寄存器，查询发送保持寄存器是否空；接收前先读取通信线状态寄存器，查询一帧数据是否收完。

3) 通信速率2400波特，分频系数为0030H，一帧字符有7个数据位，1个停止位，奇校验，数据帧格式字为0AH。

9.3 二、串行通信程序设计举例



[程序清单1] 甲机发送程序

.586

DATA SEGMENT USE16

BUF DB 'Hello', 03H

LENS EQU \$-BUF

DATA ENDS

CODE SEGMENT USE16

ASSUME CS:CODE, DS:DATA

BEG: MOV AX, DATA

MOV DS, AX

CALL I8250  ; 主串口初始化

LEA BX, BUF

MOV CX, LENS

SCAN: MOV DX, 3FDH

IN AL, DX

TEST AL, 20H

; 发送保存寄存器是否为空?

JZ SCAN

MOV DX, 3F8H

MOV AL, [BX] ; 取字符

OUT DX, AL

; 送主串口数据寄存器

INC BX

LOOP SCAN

MOV AH, 4CH

INT 21H

乙机接
收程序

I8250 PROC

MOV DX, 3FBH

MOV AL, 80H

OUT DX, AL

MOV DX, 3F9H

MOV AL, 0

OUT DX, AL

MOV DX, 3F8H

MOV AL, 30H

OUT DX, AL

MOV DX, 3FBH

MOV AL, 0AH

OUT DX, AL

MOV DX, 3F9H

MOV AL, 0

OUT DX, AL

MOV DX, 3FCH

MOV AL, 0

OUT DX, AL

RET 

I8250 ENDP

CODE ENDS

END BEG

[程序清单2] 乙机接收程序

.586

DATA SEGMENT USE16

MESG DB 'The received characters are', '\$'

DATA ENDS

CODE SEGMENT USE16

ASSUME CS:CODE, DS:DATA

BEG: MOV AX, DATA

MOV DS, AX

CALL I8250  ; 辅串口初始化

MOV AH, 9

MOV DX, OFFSET MESG

INT 21H

SCAN: MOV DX, 2FDH

IN AL, DX

TEST AL, 01H

; 一帧数据收完否?

JZ SCAN

MOV DX, 2F8H

IN AL, DX

AND AL, 01111111B

CMP AL, 03H

; 是结束标志字符?

JE EXIT

MOV DL, AL

MOV AH, 02H

INT 21H

JMP SCAN

EXIT: MOV AH, 4CH

INT 21H

I8250 PROC

MOV DX, 2FBH

MOV AL, 80H

OUT DX, AL

MOV DX, 2F9H

MOV AL, 0

OUT DX, AL

MOV DX, 2F8H

MOV AL, 30H

OUT DX, AL

MOV DX, 2FBH

MOV AL, 0AH

OUT DX, AL

MOV DX, 2F9H

MOV AL, 0

OUT DX, AL

MOV DX, 2FCH

MOV AL, 0

OUT DX, AL

RET 

I8250 ENDP

CODE ENDS

END BEG

例3：对主串口进行外环自动测试，将下列测试电文10行，经主串口发出，通过外环短路线接收，显示在屏幕上，测试电文如下：
THE QUICK BROWN FOX JUMPS OVER LAZY DOG

[设计思路]

- 1) 在一条报路上，长时间的循环发送这条电文，接收方统计在一定时间内的差错率即可知道该报路的通信质量。
- 2) 电文必须逐个字符发送，为了简化程序设计，发送字符和接收字符均采用**查询方式**。发送前，先读取通信线状态寄存器，查询发送保持寄存器是否为空；接收前先读取通信线状态寄存器，查询一帧数据是否接受完毕。
- 3) 本例采用两种方法编程：
 - ① **直接访问8250端口寄存器**，程序运行前RS232连接器按图9.10(a)接线，没有使用联络线；
 - ② **调用BIOS通信软件**，程序运行前需按图9.10(b)接线，准备好自发自收的外部环境。

[程序清单] ① 直接访问8250端口寄存器**DATA SEGMENT****TEXT DB 'THE QUICK BROWN FOX JUMPS OVER LAZY DOG'**
DB 0DH, 0AH**LLL EQU \$-TEXT****ERROR DB 'COM1 BAD !', 0DH, 0AH, '\$'****DATA ENDS****CODE SEGMENT****ASSUME CS:CODE, DS:DATA****BEG: MOV AX, DATA****MOV DS, AX****CALL I8250**

;主串口初始化

MOV CH, 10

;10行送CH

AGAIN: MOV CL, LLL

;电文长度送CL

MOV BX, OFFSET TEXT

TSCAN: MOV DX, 3FDH

IN AL, DX

TEST AL, 20H

JZ TSCAN

MOV AL, [BX]

;发送保持寄存器空？

;否

;取字符

SEND: MOV DX, 3F8H

OUT DX, AL

;送主串口数据寄存器

MOV SI, 0

RSCAN: MOV DX, 3FDH

IN AL, DX

TEST AL, 01H

;一帧数据收完否？

JNZ REVEICE

;收完转

DEC SI

JNZ RSCAN

;延时

JMP DISPERR

;超时，转出错处理

REVEICE: MOV DX, 3F8H

IN AL, DX

;读数据寄存器

AND AL, 7FH

DISP: MOV AH, 2

MOV DL, AL

INT 21H

;屏幕显示

INC BX

DEC CL

;计数

JNZ TSCAN

DEC CH

;行计数

JNZ AGAIN

JMP RETURN

DISPERR: MOV AH, 9

MOV DX, OFFSET ERROR

INT 21H

;显示出错信息

RETURN: MOV AH, 4CH

INT 21H

;返回 DOS

;-----

I8250 PROC

MOV DX, 3FBH

MOV AL, 80H

OUT DX, AL

MOV DX, 3F9H

MOV AL, 0

OUT DX, AL

MOV DX, 3F8H

MOV AL, 60H

OUT DX, AL

MOV DX, 3FBH

MOV AL, 03H

OUT DX, AL

MOV DX, 3F9H

MOV AL, 0

OUT DX, AL

MOV DX, 3FCH

MOV AL, 0

OUT DX, AL

RET

I8250 ENDP

CODE ENDS

END BEG

[程序清单] ② 调用BIOS通信软件

```
DATA    SEGMENT
TEXT    DB    'THE QUICK BROWN FOX JUMPS OVER LAZY DOG'
        DB    0DH,0AH
LLL     EQU    $-TEXT
ERROR   DB    'COM1 BAD !',0DH,0AH,'$'
DATA    ENDS
CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA
BEG:    MOV    AX, DATA
        MOV    DS, AX
        CALL   I8250
        MOV    CH, 10
AGAIN:  MOV    CL, LLL
        MOV    BX, OFFSET TEXT
```

;主串口初始化

;10行送CH

;电文长度送CL

```
SEND:    MOV    AL,    [BX]    ;取数据
          MOV    AH,    1
          MOV    DX,    0
          INT    14H          ;发送数据
          TEST   AH,    80H    ;发送成功否？
          JNZ    DISPERR      ;失败,转出错处理

RECEIVE: MOV    AH,    2
          MOV    DX,    0
          INT    14H          ;接收一个数据
          TEST   AH,    80H    ;接收成功否？
          JNZ    DISPERR      ;失败,转出错处理
          AND    AL,    7FH
```

AH=1: 发送数据

入口参数:

AL = 待发送数据

DX = 串口号 (0/1)

出口参数:

AH₇ = 1: 表示发送失败

AH₇ = 0: 表示发送成功

AH=2: 接收数据

入口参数:

DX = 串口号 (0/1)

出口参数:

AH₇ = 1: 表示发送失败

AH₇ = 0: 表示发送成功

AL = 接收到的数据

DISP: MOV AH, 2

MOV DL, AL

INT 21H

;送屏幕显示

INC BX

DEC CL

;计数

JNZ SEND

DEC CH

;行计数

JNZ AGAIN

JMP RETURN

DISPERR: MOV AH, 9

MOV DX, OFFSET ERROR

INT 21H

;显示出错信息

RETURN: MOV AH, 4CH

INT 21H

;返回 DOS

AH=0: 初始化串口

入口参数:

AL = 初始化参数

DX = 串口号 (0/1)

AL中需设置的参数为:

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
波特率： 000=110 001=150 010=300 011=600 100=1200 101=2400 110=4800 111=9600			奇偶校验： 00=无 01=奇校验 10=无 11=偶校验		↓	代码长： 00=5位 01=6位 10=7位 11=8位	
				停止位： 0=1位 1=1.5位（代码长5位） =2位（代码长6-8位）			

```

I8250  PROC
        MOV     AX,    0083H
        MOV     DX,    0
        INT     14H
        RET
    
```

;波特率1200

;无校验, 8位数据

```

I8250  ENDP
CODE   ENDS
END     BEG
    
```

出口参数

AH = 通信线路状态

AL = MODEM状态