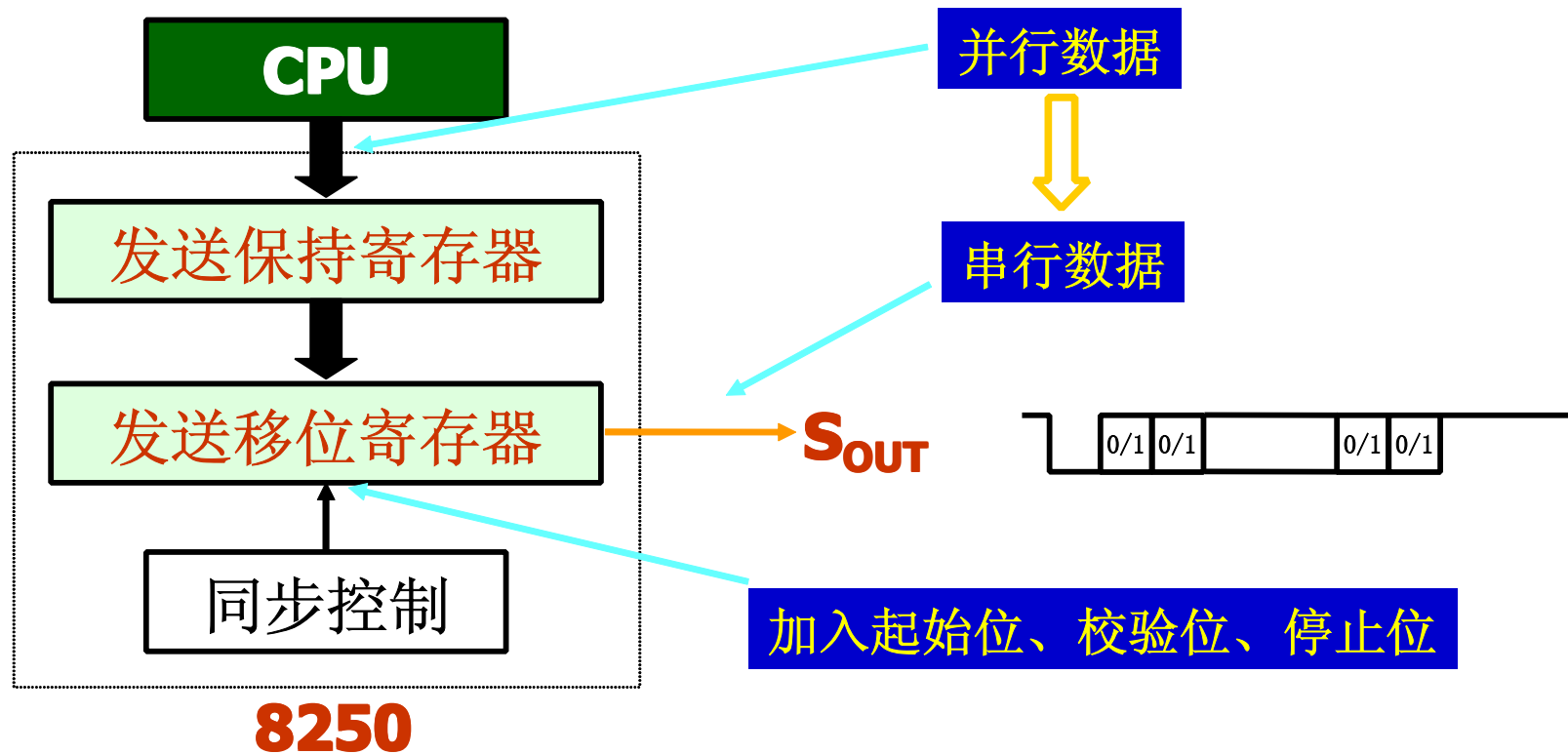


三、可编程串行接口 Ins 8250

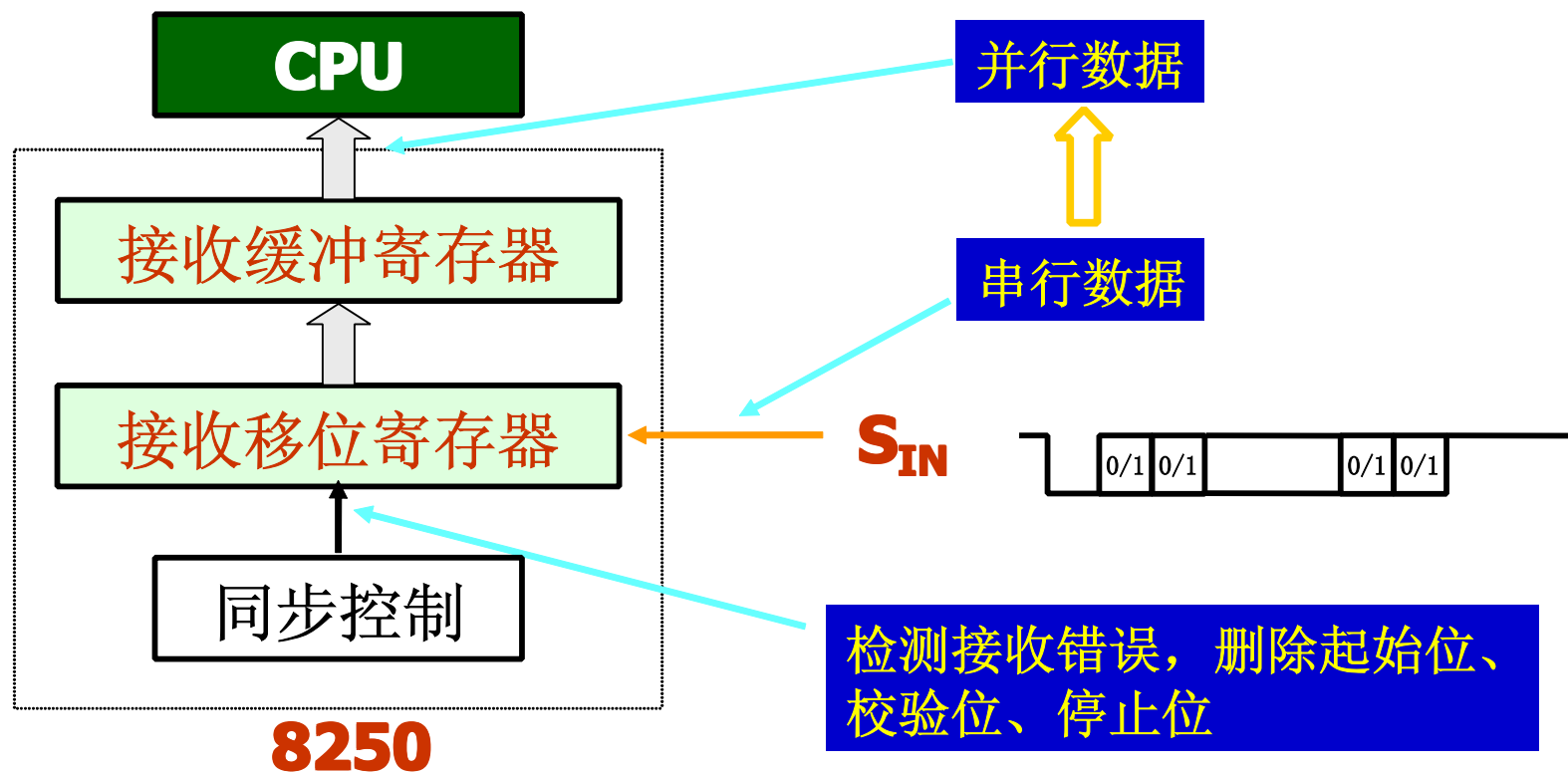
- 串行传输，需要并-----串和串----并转换，
并按照传输协议发送和接收每个字符（或数据块）
- **8250**实现了起止式串行异步通信协议，支持全双工通信：
 - 通信字符可选**5、6、7、8**位数据位
 - 停止位可选**1、1.5、2**位
 - 可选择奇校验、偶校验、不校验或校验位强制为“**1**”/“**0**”
 - 具有奇偶校验错、帧错和溢出等错错误检测电路
- **8250**支持的数据传输速率为**50~9600bps**
- **8250**内部有**10**个可寻址的**8**位寄存器，分为**3**类：数据类，控制类，状态类。

1. 串行数据的发送



双缓冲寄存器结构，保证数据的连续发送

2. 串行数据的接收



双缓冲寄存器结构,保证数据的连续接收

3. 接收错误的处理

奇偶错误**PE** (**Parity Error**)

- 若接收到的字符的“1”的个数不符合奇偶校验要求

帧错误**FE** (**Frame Error**)

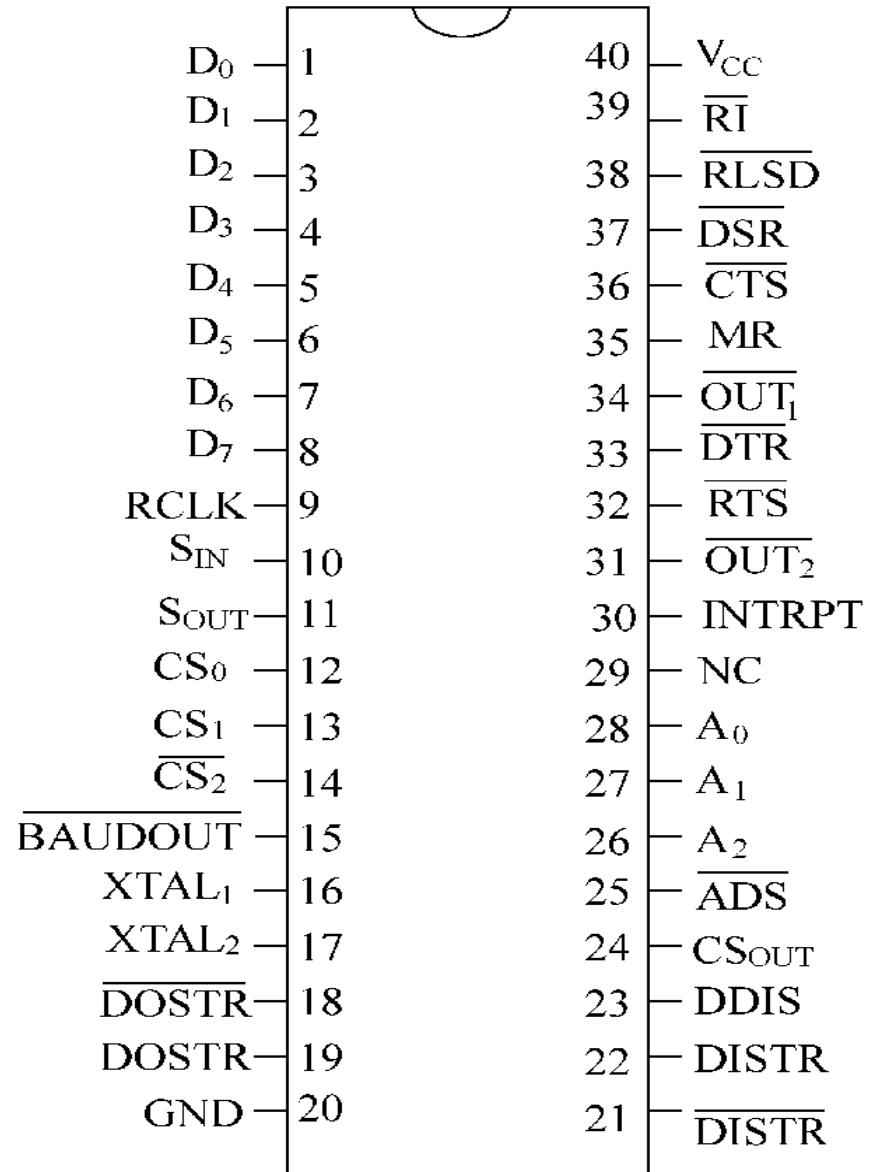
- 若接收到的字符格式不符合规定（如缺少停止位）

溢出错误**OE** (**Overrun Error**)

- 若接收移位寄存器接收到一个数据，并送至输入缓冲器时，**CPU**还未取走前一个数据，就会出现数据溢出
- 若接收缓冲器的级数多，则溢出错误的几率就少

4、8250的引脚

- 连接**CPU**的部分
- 连接外设的部分
- 注意：**8250**不是**Intel**公司的产品，所以该芯片引脚名称与前面学习的**8253**、**8255**等**Intel**产品有所不同，但是引脚功能却是类似的



处理器接口引脚（1）

数据线**D₀-D₇**：在**CPU**与**8250**之间交换信息

地址线**A₀-A₂**：寻址**8250**内部寄存器

片选线：包括

- 3个片选输入信号**CS₀**、**CS₁**、 **$\overline{CS_2}$**
- 1个片选输出信号**CS_{OUT}**。

当3个片选输入都有效时，才选中**8250**芯片，同时**CS_{OUT}**输出高电平有效（作为选中此片的指示）。

地址选通信号 **\overline{ADS}** ：当该信号低有效时，锁存上述地址线和片选线的输入状态，保证读写期间的地址稳定

处理器接口引脚（2）

读控制线

- 数据输入选通**DISTR**（高有效）和 **$\overline{\text{DISTR}}$** （低有效）
中一个信号有效，**CPU**从**8250**内部寄存器读出数据
- 相当于**I/O**读信号

写控制线

- 数据输出选通**DOSTR**（高有效）和 **$\overline{\text{DOSTR}}$** （低有效）
中一个有效，**CPU**就将数据写入**8250**内部寄存器
- 相当于**I/O**写信号

8250读写控制信号有两对，每对信号作用完全相同，只不过有效电平不同而已

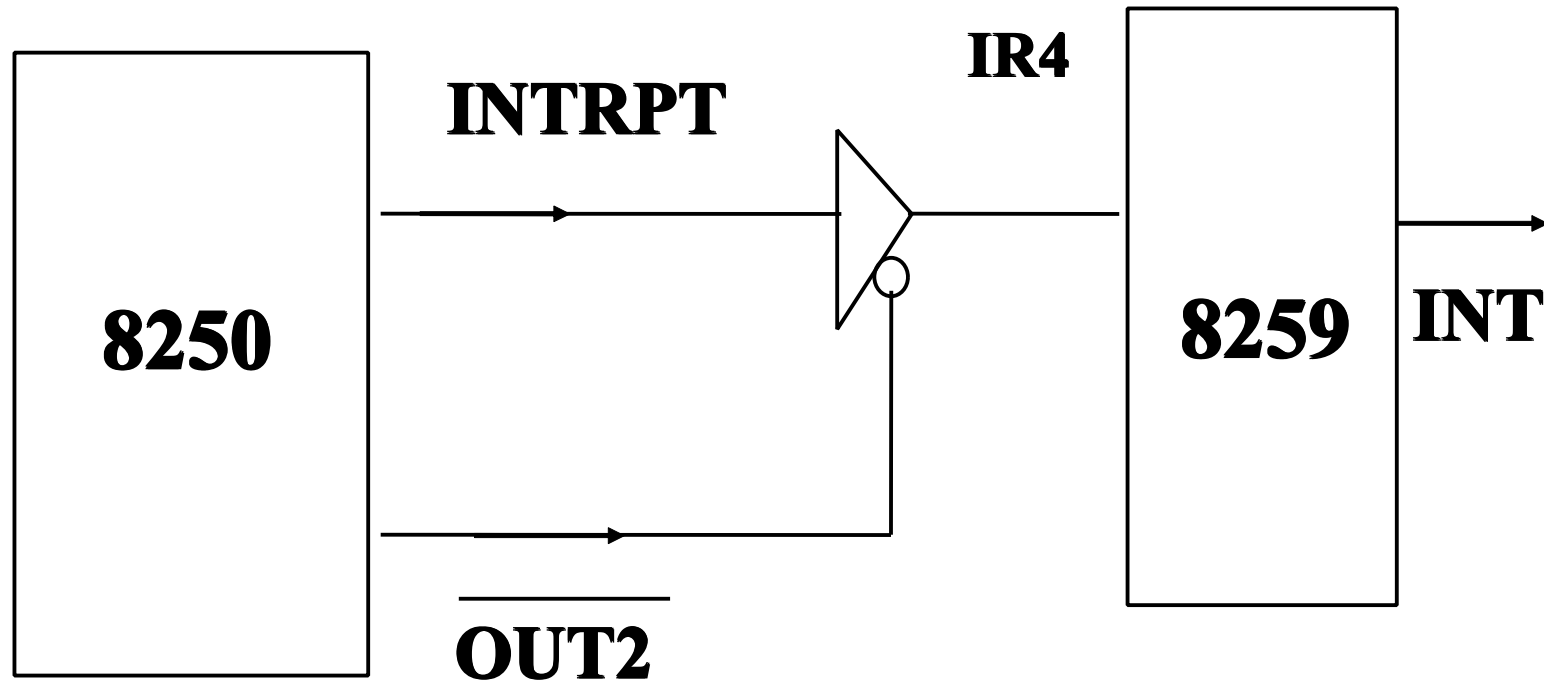
处理器接口引脚（3）

驱动器禁止信号**DDIS**：**CPU**从**8250**读取数据时，**DDIS**引脚输出低电平，用来禁止外部收发器对系统总线的驱动；其他时间，**DDIS**为高电平

主复位线**MR**：硬件复位信号

中断请求线**INTRPT**：**8250**有4级中断、共10个中断源，当任一个未被屏蔽的中断源有请求时，**INTRPT**均输出高电平向**CPU**请求中断

PC/XT 中 COM_i的中断请求信号



串行异步接口引脚



8250 9个信号的名称与**RS232-C**信号的名称稍有不同

时钟信号

时钟输入引脚**XTAL1**: **8250**的基准工作时钟

时钟输出引脚**XTAL2**: 基准时钟信号的输出端

波特率输出引脚**BAUDOUT**: 基准时钟经**8250**内部波特率发生器分频后产生发送时钟

接收时钟引脚**RCLK**: 接收外部提供的接收时钟信号。

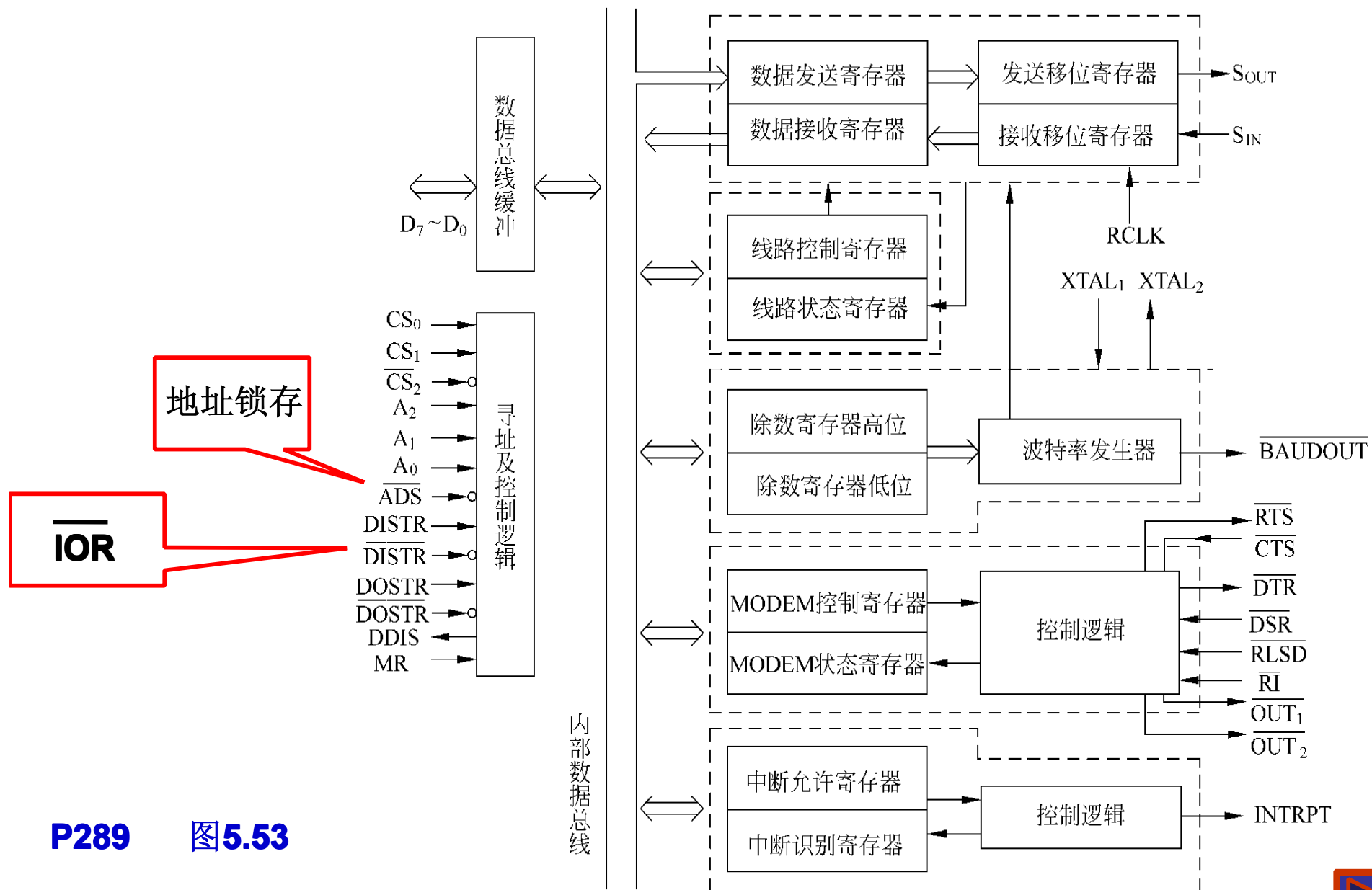
若采用发送时钟作为接收时钟，则只要将**RCLK**引脚和**BAUDOUT**引脚直接相连

输出线

$\overline{\text{OUT}}_1$ 和 $\overline{\text{OUT}}_2$:

- 两个可由用户定义用途的输出信号
- 由调制解调器控制寄存器的**D₂**和**D₃**位控制其输出
- 使用时，一般低电平有效，复位时恢复为高

(一) 可编程串行接口 Ins 8250 结构



P289

图 5.53



8250的寄存器

- **8250**内部有**9**种可访问的寄存器，其中，除数寄存器是**16**位的，占用两个地址连续的**8**位端口
- 内部寄存器通过引脚**A₀-A₂**来进行寻址；
- 利用**通信线路控制寄存器**的最高位，即除数寄存器访问位**DLAB**，来区别共用两个端口地址的不同寄存器
 - **DLAB=1**，拟访问除数寄存器的高**8**位及低**8**位
 - **DLAB=0**，拟访问其他控制或状态寄存器

(二) 寄存器及寻址 P289 表5. 10

$A_9A_8A_7A_6$	$A_5A_4A_3A_2A_1A_0$	DLAB (标志位)	寄存器
	000	0	写发送寄存器/读接受寄存器
	000	1	除数寄存器低字节
	001	1	除数寄存器高字节
	001	0	中断允许
	010		中断识别
	011		线路控制 D₇ 为 DLAB
	100		MODEM 控制
	101		线路状态
	110		MODEM 状态
1111111		COM1	3F8—3FFH
1011111		COM2	2F8—2FFH

(二) 寄存器及寻址 P289 表5. 10

注意:

$A_2A_1A_0 = 011$ 时 访问线路控制寄存器
其 D_7 位称为 **DLAB** 除数标志位

当 **DLAB** = 1 时

$A_2A_1A_0 = 000$ 时 访问除数寄存器低字节

$A_2A_1A_0 = 001$ 时 访问除数寄存器高字节

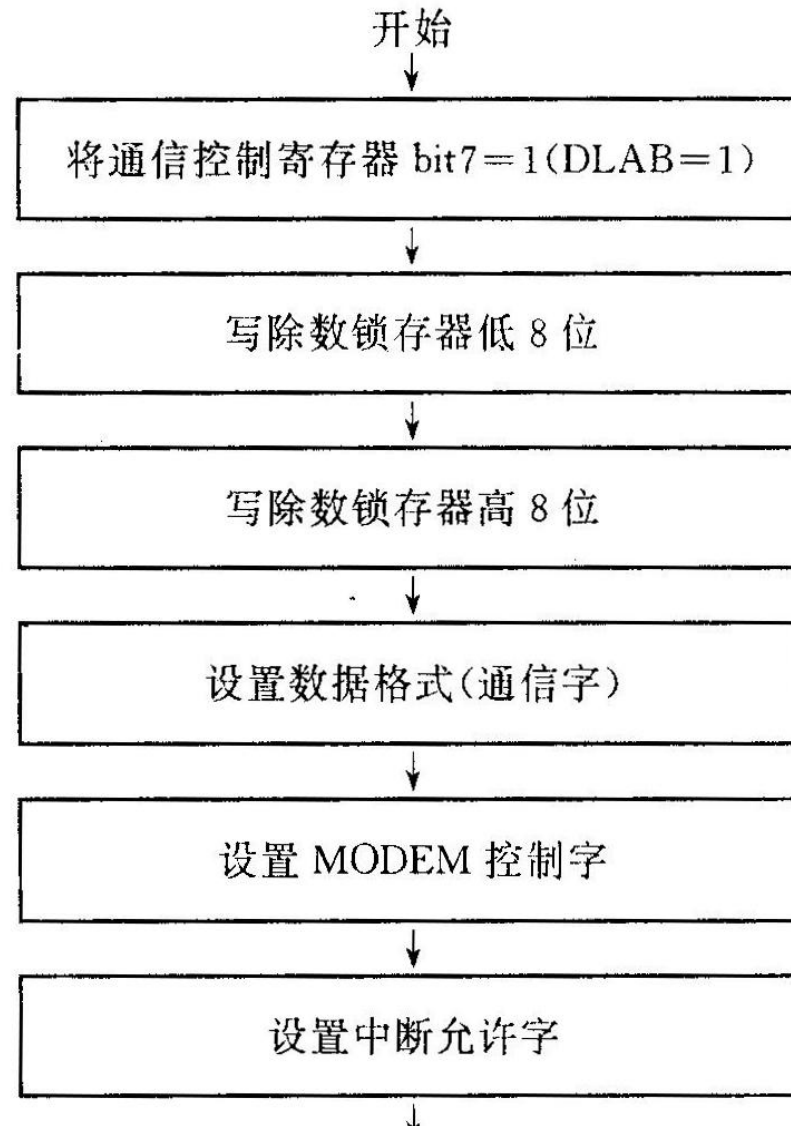
IMB PC/XT 系统的两个异步串行通信口
COM1 地址 3F8—3FFH, COM2 地址 2F8—2FFH

8250内部寄存器的地址

DLAB	A2 A1 A0	寄存器操作	com1	com2
0	0 0 0	读接收缓冲器/写发送保持寄存器	3F8+0	2F8+0
0	0 0 1	中断允许寄存器	3F8+1	2F8+1
0	0 1 0	中断识别寄存器（只读）	3F8+2	2F8+2
*	0 1 1	通信线路控制寄存器	3F8+3	2F8+3
*	1 0 0	调制解调器控制寄存器	3F8+4	2F8+4
*	1 0 1	通信线路状态寄存器	3F8+5	2F8+5
*	1 1 0	调制解调器状态寄存器	3F8+6	2F8+6
*	1 1 1	不用	3F8+7	2F8+7
1	0 0 0	除数寄存器（低8位）	3F8+0	2F8+0
1	0 0 1	除数寄存器（高8位）	3F8+1	2F8+1

(三) 编程——分初始化及工作两部分

初始化顺序



(三) 编程——分初始化及工作两部分

1. 初始化步骤:

置 **DLAB=1** (线路控制寄存器 $D_7 = 1$;

COM1—3FBH)

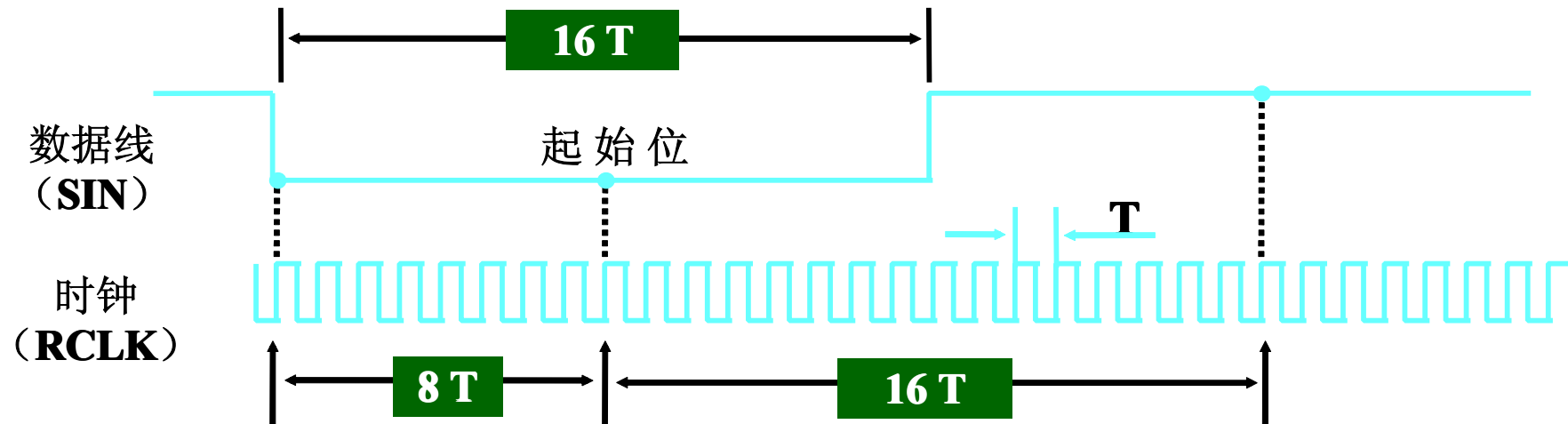
写除数 (分高 (COM1—3F9)、

低 (COM1—3F8) 字节写两次)

写线路控制字 (**DLAB=0**,

其余位控制数据格式)

除数寄存器



除数寄存器保存设定的分频系数
分频系数 = 基准时钟频率 ÷ (16 × 比特率)

写除数寄存器 高/低 字节

波特率与除数的关系 (P292 表5.11)

波特率	高字节	低字节
50	09	00
.....		
1800	00	40
2000	00	3A
.....		

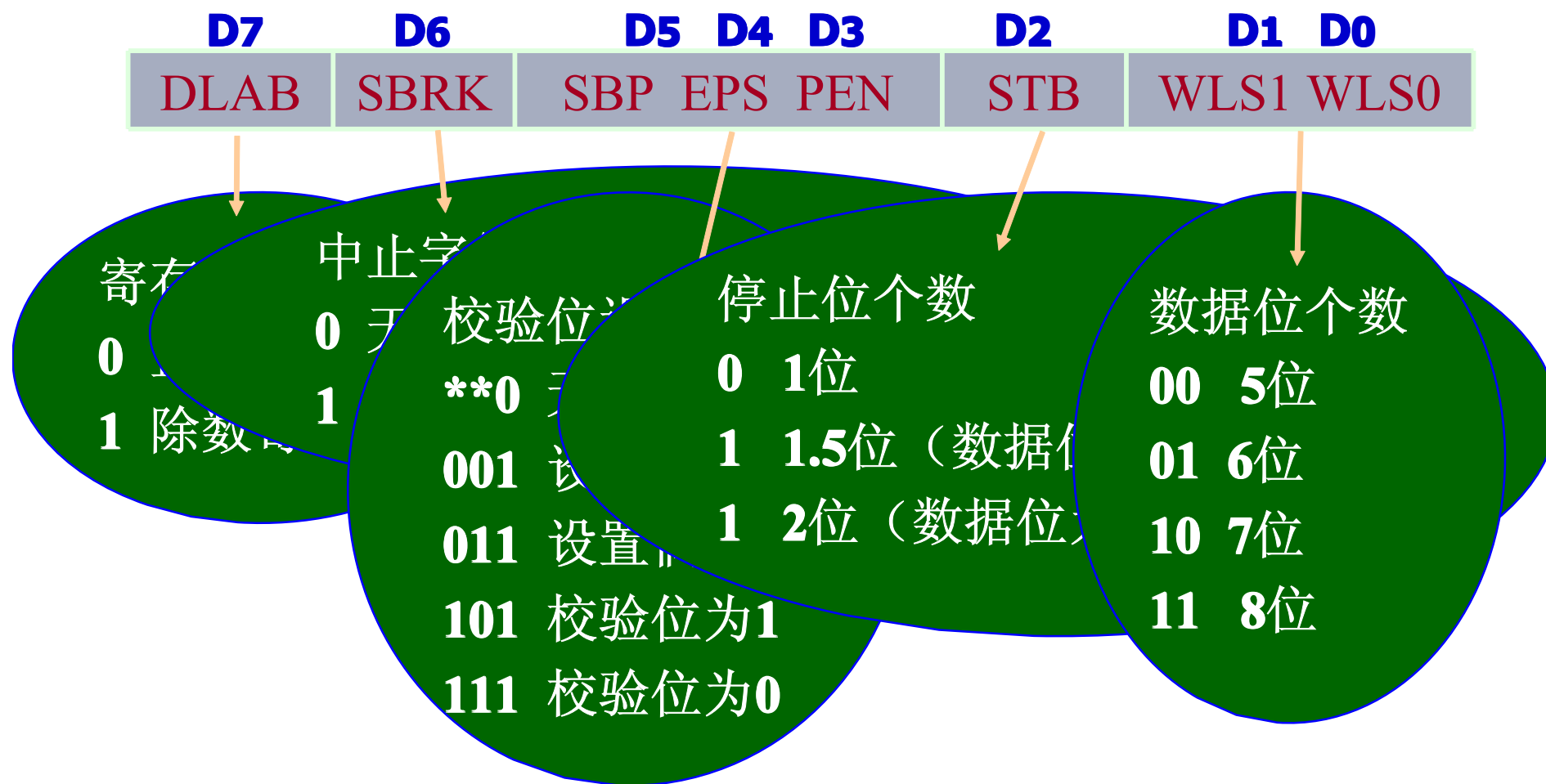
时钟频率除以除数寄存器数得数据发送器的工作频率，再除以16即得波特率

1. 写线路控制字 (**DLAB=0**, 其余位控制数据格式, **P291图5.55**)

(地址: **COM1—3FBH**)

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
除数标志	中止位	奇偶校验选择			停止位长	数据位长	
0 正常 1 访问除数	0 正常 1 发中止符				0 1位	00	5位
					1 1.5位 或2位	01	6位
		000 无校验				10	7位
		001 奇校验				11	8位
		011 偶校验					
		101 附加位为1					
		111 附加位为0					

线路控制寄存器LCR



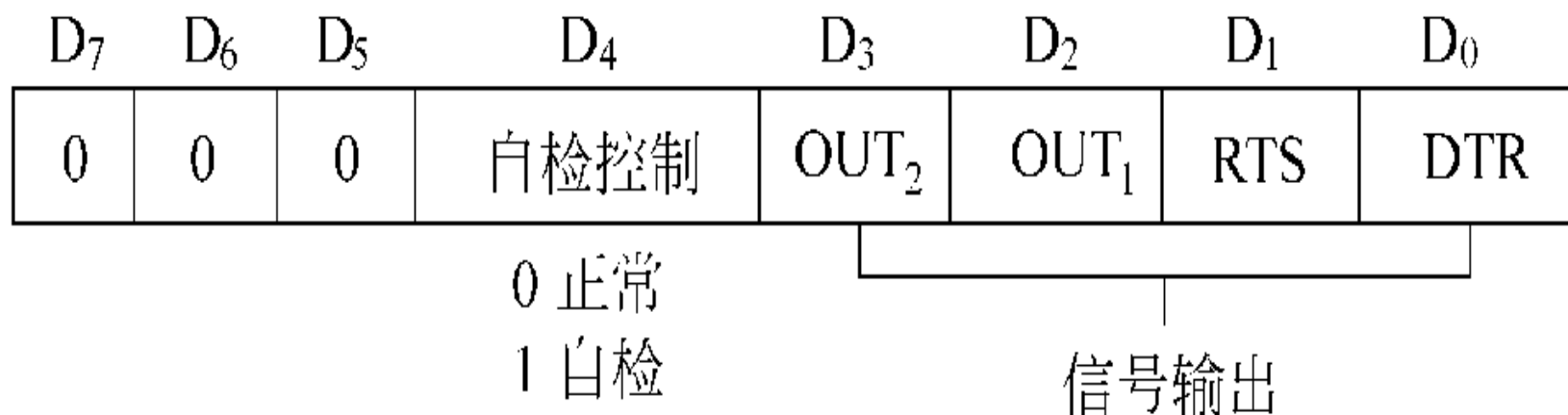
指定串行异步通信的字符格式

2. 写MODEM控制寄存器

（确定联络信号， P292图5. 57）

若用中断须置 $OUT_2=1$

（地址： **COM1—3FCH**）



调制解调器控制寄存器MCR

D7 – D5	D4	D3	D2	D1	D0
0 0 0	LOOP	OUT2	OUT1	RTS	DTR

为1
使8250为
循环工作方式

否则为正常工
作方式

为1使 $\overline{\text{OUT2}}$ 引脚为低，否则为高

为1使 $\overline{\text{OUT1}}$ 引脚为低，否则为高

为1使 $\overline{\text{RTS}}$ 引脚为低，否则为高

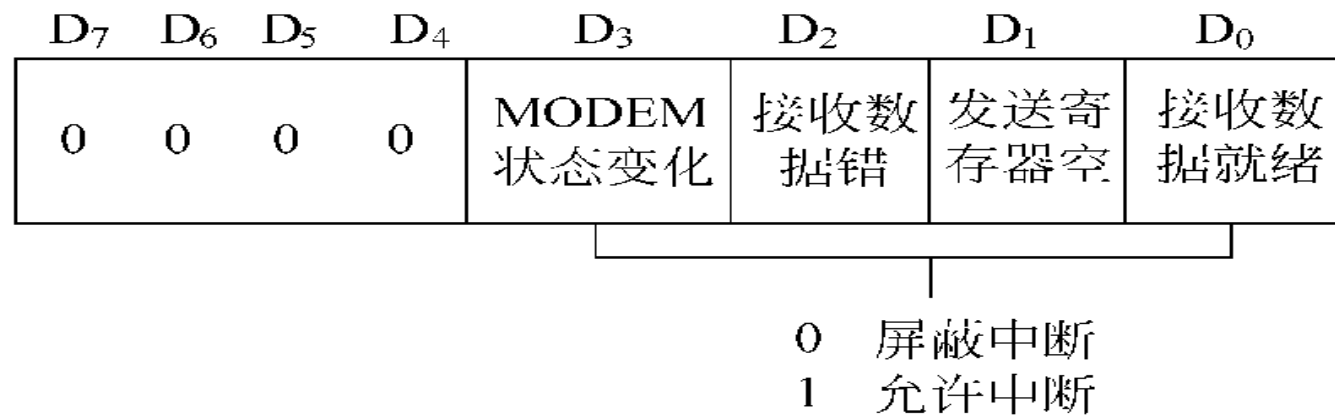
为1使 $\overline{\text{DTR}}$ 引脚为低，否则为高

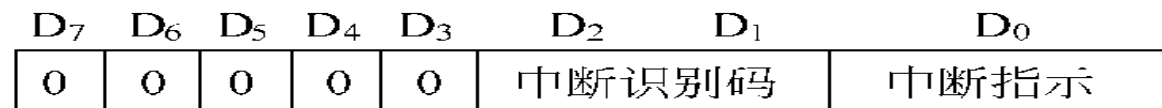
设置8250与数据通信设备之间联络应答的输出信号

3. 写中断允许寄存器

(中断逻辑, P293图5.59)

(地址: COM1—3F9H)





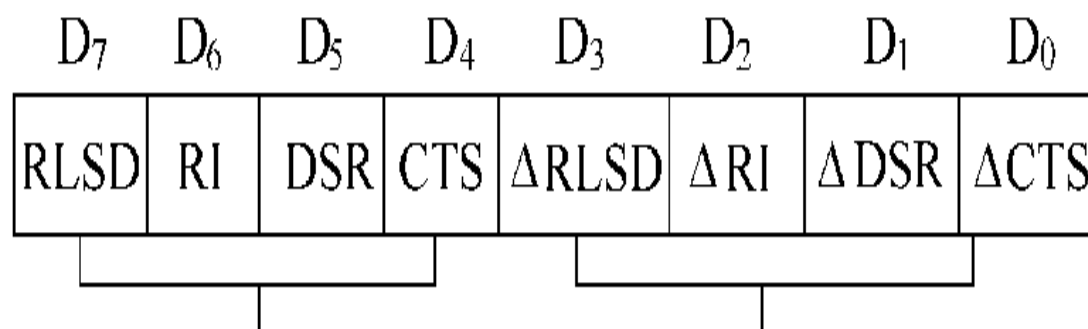
0 有中断
1 无中断

0	0	MODEM 状态变化 (最低级)
0	1	发送寄存器空
1	0	接收数据就绪
1	1	接收数据错 (最高级)

4. 中断识别寄存器

P294图 5.60

(地址: **COM1—3FAH**)



信号输入

信号输入变化

5. MODEM状态寄存器

P293图 5.56

(地址: **COM1—3FEH**)

4级中断（4个优先级、10个源）

1. 接收线路状态中断

- 奇偶错
- 溢出错
- 帧错
- 收到中止字符

2. 接收器数据准备好中断

3. 发送保持寄存器空中断

4. 调制解调器状态中断

- 清除发送状态改变
- 数据终端准备好状态改变
- 振铃接通变成断开
- 接收线路信号检测状态改变

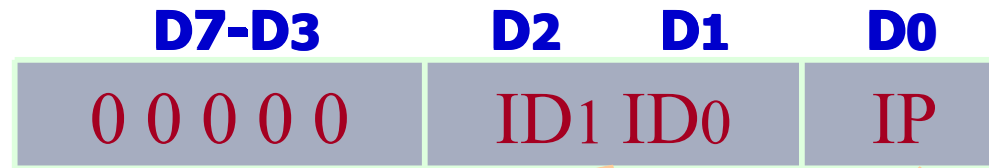
优先级高

优先级低

6. 中断允许寄存器IER

- **8250**设计有**4**级中断和**2**个中断寄存器
 - ◆ **4**级中断指优先权的等级为**4**级，它是按照串行通信过程中事件的紧迫程度安排的、是固定的
 - ◆ 用户可利用中断允许或禁止进行控制，中断允许寄存器的低**4**位控制**8250**这**4**级中断是否被允许
 - 某位为**1**，则对应的中断被允许
 - 否则，被禁止

7. 中断识别寄存器IIR



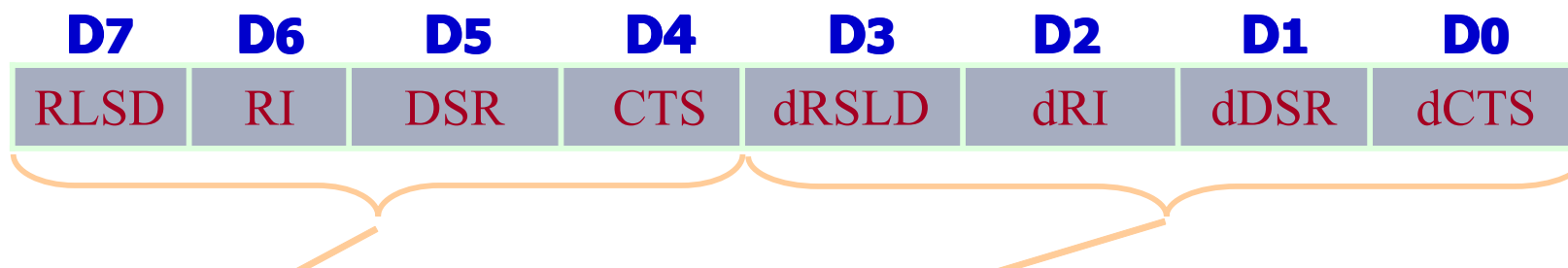
标识哪一级有中断

ID1 ID0	优先权	中断类型
1 1	1	接收线路状态中断
1 0	2	接收数据准备好中断
0 1	3	发送保持寄存器空中断
0 0	4	调制解调器状态中断

0 有中断
1 无中断

保存正在请求中断的优先权最高的中断级别编码

8. 调制解调器状态寄存器MSR



低4位中某位为**1**，则说明从上次**CPU**读取该状态字后，相应输入信号已发生改变，从高变低或反之

MSR反映**4**个控制输入信号的当前状态及其变化
MSR低**4**位中任一位置**1**，均将产生调制解调器状态中断，
当**CPU**读取该寄存器或复位后，低**4**位被清零

初始化编程例

例（P296）

```
MOV DX, 3FBH      ; COM1 线路控制寄存器  
MOV AL, 80H       ; DLAB = 1  
OUT DX, AL  
MOV DX, 3F9H      ; 除数高字节  
MOV AL, 0  
OUT DX, AL
```


MOV DX, 3F8H ; 除数低字节
MOV AL, 30H
OUT DX, AL ; 0030H—波特率2400

MOV DX, 3FBH ; 线路控制寄存器
MOV AL, 1AH ; 00011010B
; 数据位长7，停止位长1，偶校验
OUT DX, AL

MOV DX, 3FCH ; MODEM 控制寄存器

.....

2. 工作编程（查询方式通信）

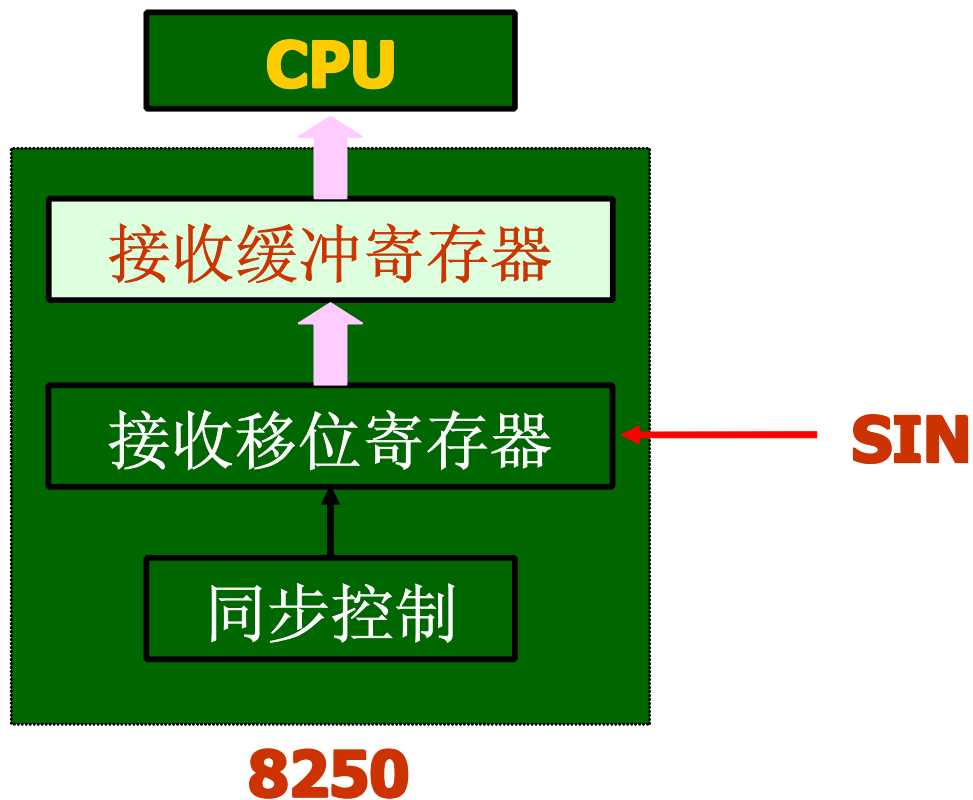
数据的发送与读取：

```
MOV DX, 3F8H    ; COM1 发送/接受R
OUT DX, AL       ; 发送数据 snt-R
IN  AL, DX       ; 读取数据 rec-R
```

线路状态的信息采集：

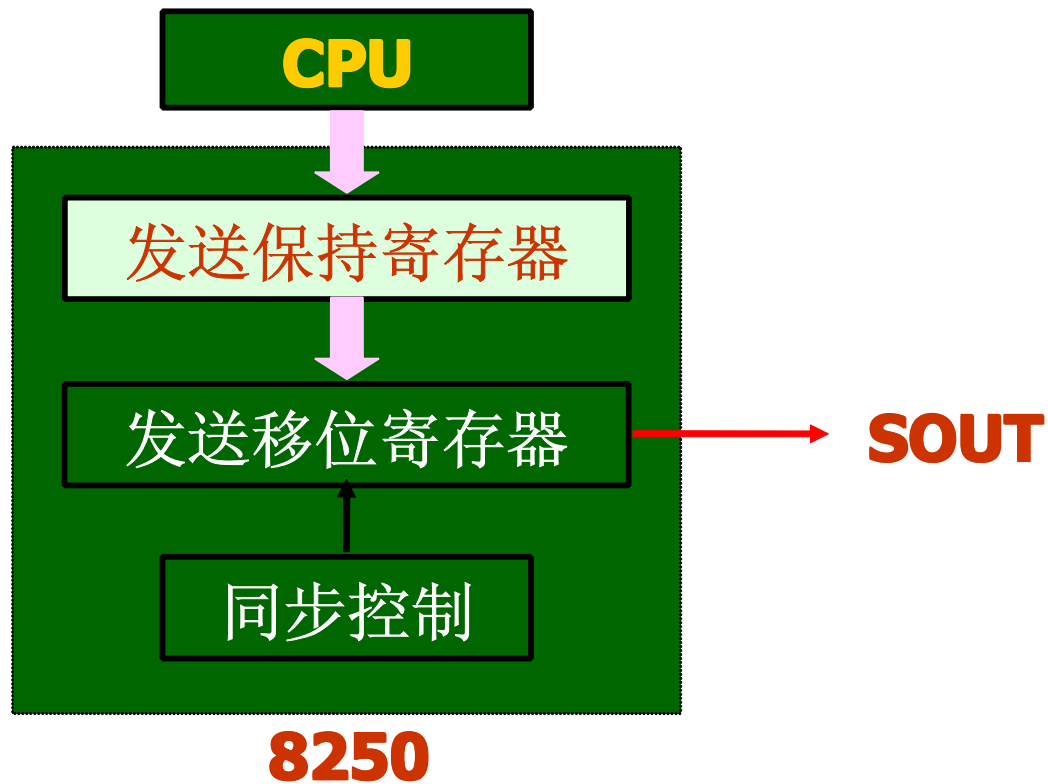
```
MOV DX, 3FDH ; COM1 线路状态
IN  AL, DX
```

9. 接收缓冲寄存器RBR



存放串行接收后转换成并行的数据

10 .保持寄存器THR



包含将要串行发送的并行数据

数据读取条件：接受数据就绪

数据有错则进行数据有错处理

数据发送条件：发送寄存器空

以上信息从线路寄存器

（**COM1**, 由**3FDH**端口读取）得到

11. 线路状态寄存器 P291图5. 56

(地址: **COM1—3FDH**)

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	发送移位 寄存器空	发送寄 存器空	中止符 检 测	帧格 式错	奇偶 错	溢出 错	接收数 据就绪

接受移位寄存器收全时 **D₀ = 1**

出错时, **D₁—D₄**的相应位为**1**

通信线路状态寄存器LSR



为1，表示接收数据缓冲器收到一个数据，既接收数据准备好；当CPU读走数据后，该位为0

提供串行异步通信的当前状态供CPU读取和处理

例 P297 程序

```
WAIT_FOR: mov    dx, 3fdH      ; 线路状态R
           in     al, dx        ;
           test   al, 1eH      ; P291图5.56
           jne    ERROR        ; 非零转移
           test   al, 1        ; 检查是否收到数据
           jnz    RECEIVE      ; 转接收（非零转移）
```



```
test    al, 20H    ; 检查可否发送数据
jz      WAIT_FOR   ; 重新检查
```

.....

```
mov     dx, 3f8H    ; 发送寄存器
mov     al, cl      ; 字符
out     dx, al
jmp     WAIT_FOR
```

```
RECEIVE: mov  dx, 3f8H    ; 接收寄存器
         in   al, dx
         .....
```

3. 应用举例——P296-297 例5.2

工作内容：从键盘读入字符、经8250发送、
由8250自行接收、再在显示器显示

。 。 。

读键盘：

```
MOV AH, 1    ; 看是否被按键，不等待
INT 16H
```

。 。 。

读按键：MOV AH, 0 ; 读下按键内容 参见P395BIOS调用
INT 16H

。 。 。

发送：

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

。 。 。

接收：

```
MOV DX, 3F8H
IN  AL, DX
```

。 。 。

显示：

```
MOV AH, 0EH
INT 10H
```

中断方式举例

以查询方式发送数据，以中断方式接收数据，

INISIR: MOV DX, 3FBH

MOV AL, 80H

OUT DX, AL ; 置DLAB=1

MOV DX, 3F8H

MOV AL, 0CH

OUT DX, AL

MOV DX, 3F9H

MOV AL, 0 ; 置除数为000CH,

; 规定波特率为9600波特

OUT DX, AL

MOV DX, 3FBH
MOV AL, 0AH ; 1 位停止位, 7位数据位, 奇校验
OUT DX, AL ; 初始化通信控制寄存器

MOV DX, 3FCH
MOV AL, 0BH ; **00001011B,**
; **使OUT2 , DTR和RTS 有效**
OUT DX, AL ; 初始化**MODEM** 寄存器

MOV DX, 3F9H
MOV AL, 01H ; 允许接收数据寄存器满产生中断
OUT DX, AL ; 初始化中断允许寄存器
STI ; **CPU** 开中断

中断服务程序段

RCVE PROC FAR

PUSH AX

PUSH BX

PUSH DX

PUSH DS

MOV DX, 3FDH

IN AL, DX

MOV AH, AL ; 保存接收状态

MOV DX, 3F8H

IN AL, DX ; 读入接收到的数据

AND AL, 7FH ; 7位数据位

TEST AH, 1EH ; 检查有无错误产生

JZ SAVEDATA

MOV AL, '?' ; 出错的数据用问号替代

SAVEDATA:

MOV DX, SEG BUFFER

MOV DS, DX

MOV BX, OFFSET BUFFER

MOV [BX], AL ; 存数据

MOV AL, 20H ; 将EOI命令发给中断控制器8259

OUT 20H, AL ; 中断控制器端口地址

POP DS

POP DX

POP BX

POP AX

STI

IRET

RECVE ENDP

