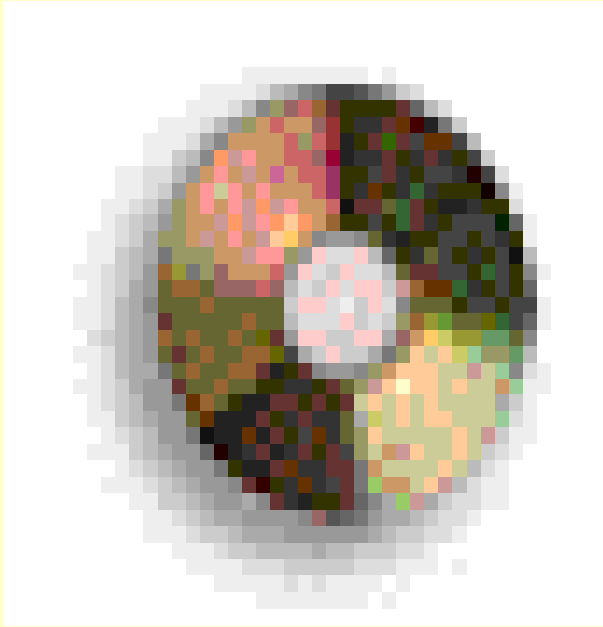


第十章 串行通信与串行接口



一、概述

二、串行通信的基本概念

三、串行接口

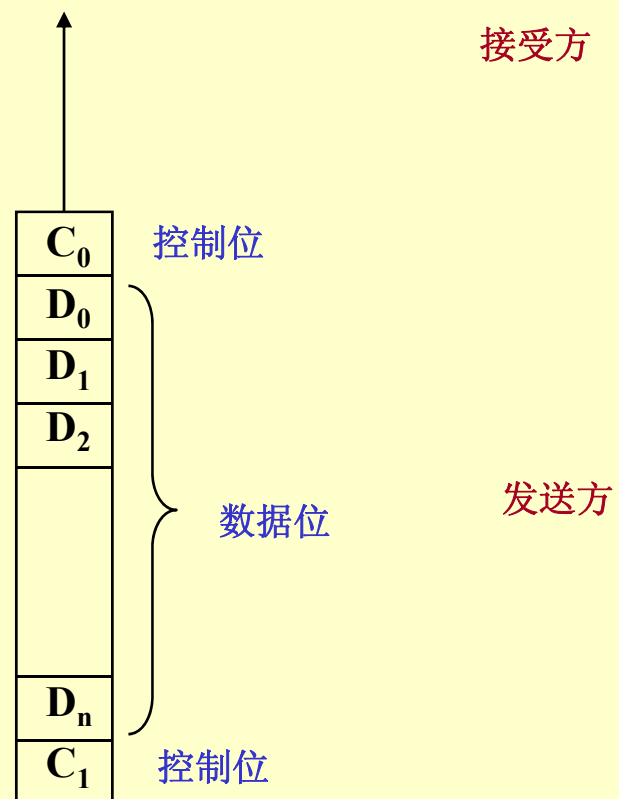
四、串行传送的实现

五、可编程异步通信接口芯片**8250**

六、**IBM PC/XT** 异步通信适配器电路

一、概述

1. 串行通信



特点：速度较慢，但造价低。

2. 传送编码

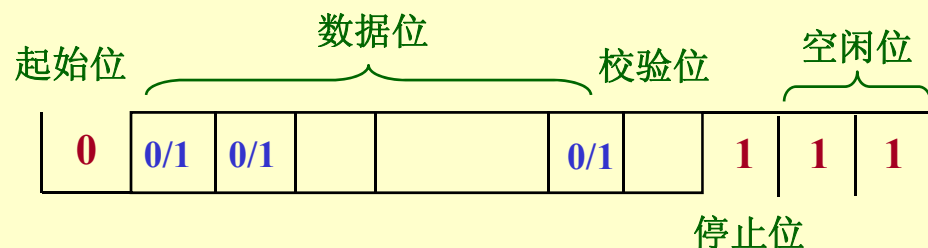
在计算机中，数字和字符都是以一定的编码表示，编码的种类很多，常用的有：

- ① 扩展的BCD交换码 EBCDIC —— 这是一种 8 位编码，较常用在同步通信中。
- ② 美国标准信息交换码 ASC II。

3. 两种通信方式

- ① 异步通信 ASYNC (Asynchronous Data Communication)

它用一个起始位表示字符的开始，用停止位表示字符的结束。



以这样的方式表示字符，则字符可以一个接着一个传送。

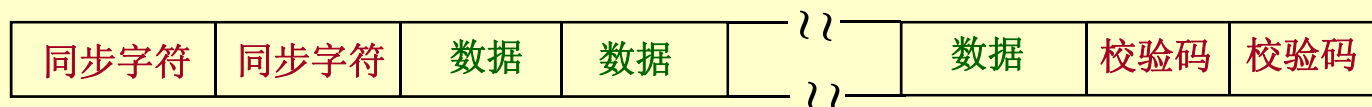
在这种方式中，CPU 与外设之间必须有两项规定：

- 传送字符格式；
- 数据传送速率——波特率（每秒钟传送的字符个数）。



② 同步通信

传送时，将数据连成块，在数据块开始处用同步字符指示。



同步传送的速度高于异步，但它要求用时钟来实现发送端和接收端之间的同步，故硬件复杂。

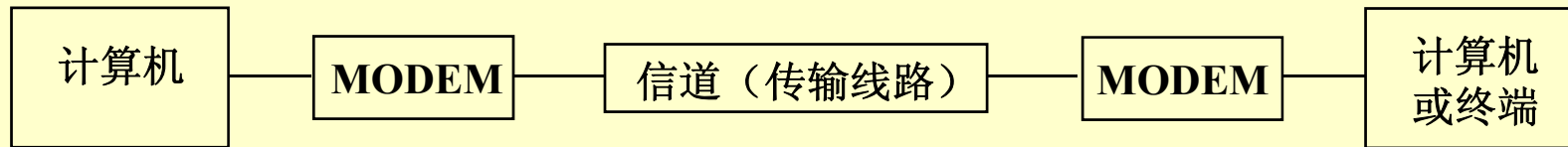
同步方式常用于：

计算机到计算机之间的通信；

计算机到**CRT**/外设之间的通信。

二、串行通信的基本概念

串行数据通信系统模型



对一个通信系统来讲，必须要考虑的问题有以下几个方面：

- (1) 信道的带宽和数据的传输速率；
- (2) 信号的调制与解调；
- (3) 串行通信的信号格式； 等。

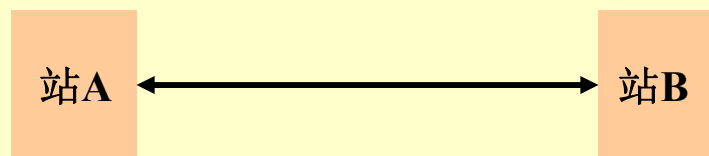
(一) 数据的传送方向

通常串行通信，数据在两个站之间是双向传送的，A站可作为发送站，B站可作为接收站，反之亦可。

通常传送可分：

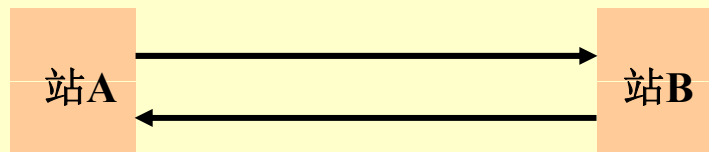
1. 半双工 (Half Duplex)

每次只有一个站发送，即只能由 A 发送到 B，或有 B 发送到 A，不能A和B同时发送。



2. 完全双工 (Full Duplex)

同时两个站都能发送。



(二) 信道的带宽和传输速率

1. 信道——信号传输的通道，包括传输媒体及有关的中间通信设备。

有线信道的传输媒体有架空线、同轴电缆、光导纤维等，考虑到传输时信号产生衰减，每隔一定距离要设置中继器等放大设备。

2. 信道的带宽

信道是电信号通过的路径，信道传输信号的频率范围称为信道的带宽（**Bandwidth**）。

信道的带宽由——

传输媒体；	} 共同决定
有关的附加设备；	
信号的频率；	

带宽的衡量——通常用一个信道的截止频率来衡量。

截止频率：当一些频率信号通过时,对某个频率信号产生较大衰减的频率。

3. 传输速率

指一个信道每秒钟传送的二进制的位数。

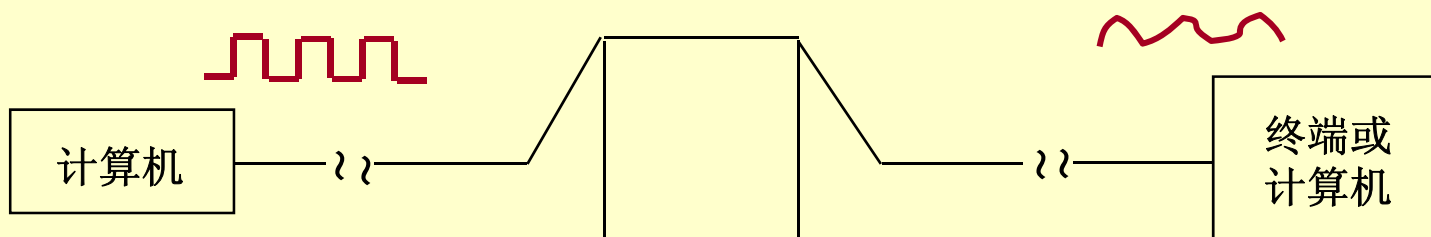
（**bite per sedond—— bps**）—— 波特率

信道的最大传输速率受信道的带宽的限制。异步通信传输速率为 **50 ~ 9600 bps**。

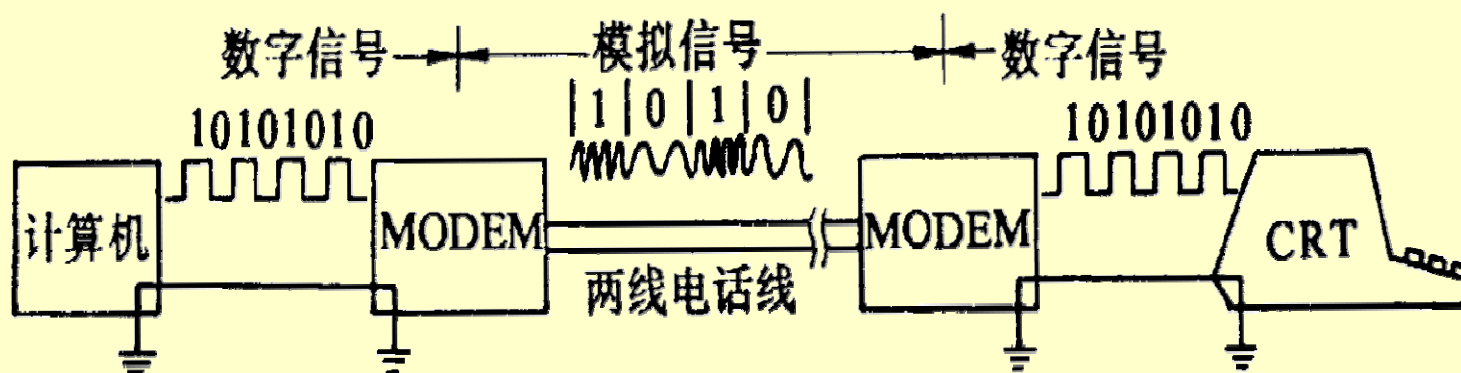


(三) 信号的调制与解调

数据通信传输的是数字信号，要求传送线的频带很宽，而我们在长距离传送时，有时是利用电话线传送的，而电话线的带宽为 $300 \sim 3000 \text{ Hz}$ ，因此若直接传输数字信号，信号就要发生畸变。



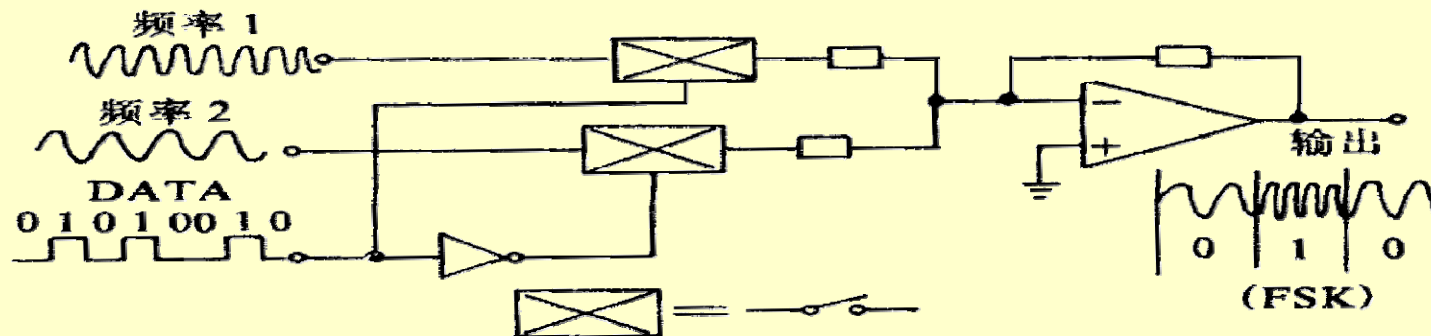
因此，需用调制器将数字信号转换成模拟信号 — 传输 — 再用解调器将其转换成数字信号。



调制：用一个信号控制另一个信号的某个参数随之变化的过程，叫调制。

这两个信号分别叫调制信号和被调信号，被调信号只作为传输过程中信息的载体，也叫载波。

调制方法：只介绍一种常用的方法——数字键控调制方法的 频移键控法（FSK）。
（Frequency Shift Keying）



两个不同频率的模拟信号，分别由电子开关控制，在运算放大器的输入端输入；
电子开关由需传输的数据信号控制：

当信号为“1”时，上面的电子开关导通，送出一串频率较高的模拟信号；

当信号为“0”时，下面的电子开关导通，送出一串频率较低的模拟信号；

于是，在运算放大器的输出端，得到了调制后的信号。

解调：将已经调制的信号恢复成原来的数字信号的过程。

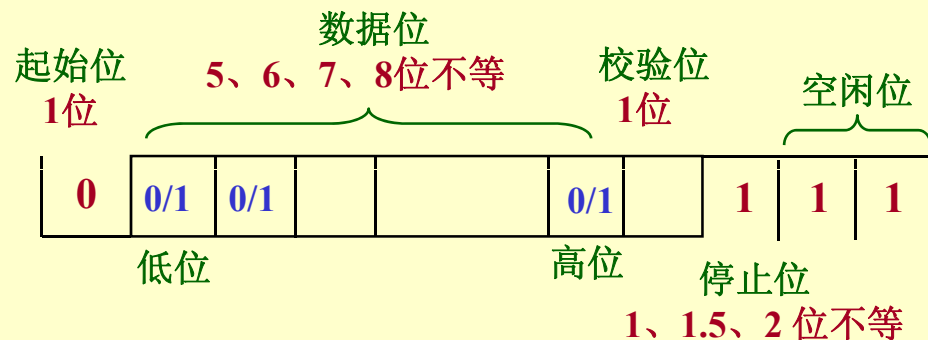
(三) 串行通信的信号格式

在数据通信中，传输的对象是一系列的 0，1，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。

在串行通信中，有两种最基本的通信方式 { 异步通信方式
同步通信方式 } 都有各自的约定。

1. 异步串行通信

在发送一个字符时，包含一个起始位，若干数据位，一个奇偶校验位，停止位。

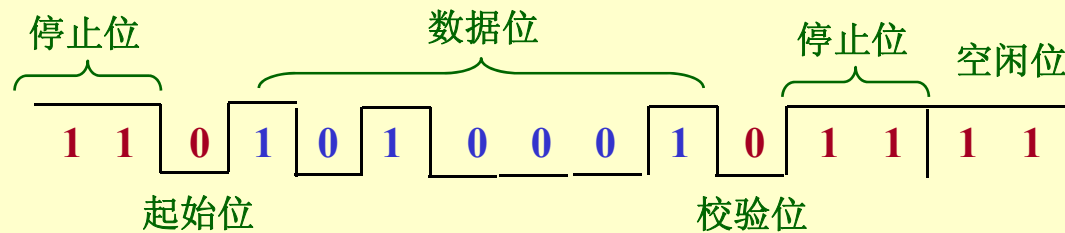


一个帧结构

- 起始位：用来通知接收器已经开始字符传送。接收器收到这个信号后，开始装配一个字符。
目的——是接收器和发送器能够同步工作。
- 终止位：保证下一个字符的起始位在通信线路上能够同步工作。
- 奇偶校验位：检验数据的正确性。



例如： 传送一个字符 “E”，（ASC II 码为：1000101B = 45H）



从上面分析可以看出，异步通信方式是按字符传输的，从一个字符到下一个字符之间，没有固定的时间间隔，故，称之为异步传输方式。

它的传输，靠起始位的起始信号，使发、收双方在一个字符的传输时间内保持同步，以保证正确接收。

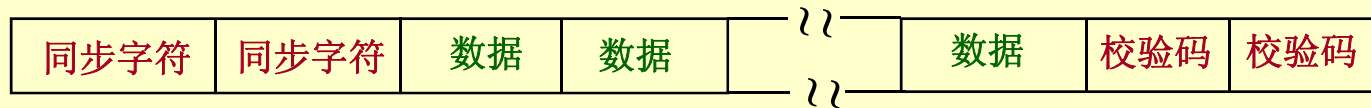


2. 同步串行通信

在异步通信中，每一个字符要用起始位和停止位标志字符传输的开始与结束，占用了时间。

在同步传输中，去掉这些控制位，把字符顺序地连接起来，组成一个数据块，这样的数据块称为一个纪录。

在纪录的开始加同步字符，在纪录的末尾加出错校验字符，形成帧。



同步字符的格式和个数根据需要而定。

在同步方式中，接收器接收数据时，首先搜索同步字符，在得到同步字符后，才开始装配数据。

（五）串行通信的校验方法

串行通信主要适用于远距离通信，因而噪声和干扰较大，为了保证高效而无差错地传送数据，对传送的数据进行校验就成了串行通信中必不可少的重要环节。

常用的校验方法有：奇偶校验、循环冗余校验（CRC）。

1. 奇偶校验

软件奇偶校验
硬件奇偶校验

这种校验方法主要用于对一个字符的传送过程进行校验。

发送时：在每一个字符的最高位后都附加一个奇偶校验位；

接收时：检查所接收的字符连同奇偶校验位，若“1”的个数不符合规定，则传输错，由CPU进行处理。

① 软件奇偶校验 若字符为7位，用偶校验，则产生奇偶校验位的程序如下：

发送：	MOV AL, DATA	;取要发送的数据
	AND AL, AL	
	JPE TRANS	;若为偶数则传送
	OR AL, 10000000B	;否，最高位置“1”
	TRANS: OUT 口地址, AL	;传送



接收：（进行奇偶校验）

IN AL, 口地址	; 输入接收的数据
AND AL, AL	
JPO ERROR	; 若为奇数，则转至出错处理
MOV DATA, AL	; 若为偶数，则存入内存

① 硬件奇偶校验

- 用专门的奇偶发生器/校验器，对 7 位或 8 位字符进行奇偶校验。
这种器件一般是中规模的集成电路。
- 在实际应用中常采用的可编程串行通信接口芯片，也包含奇偶校验和产生电路。

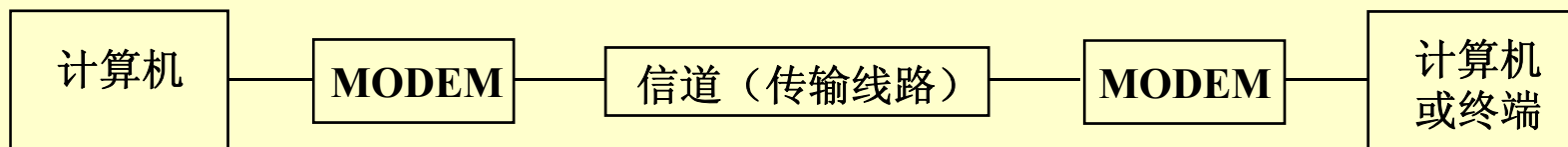
2. 循环冗余校验（CRC）

它对一个数据块进行校验，主要用于同步方式。



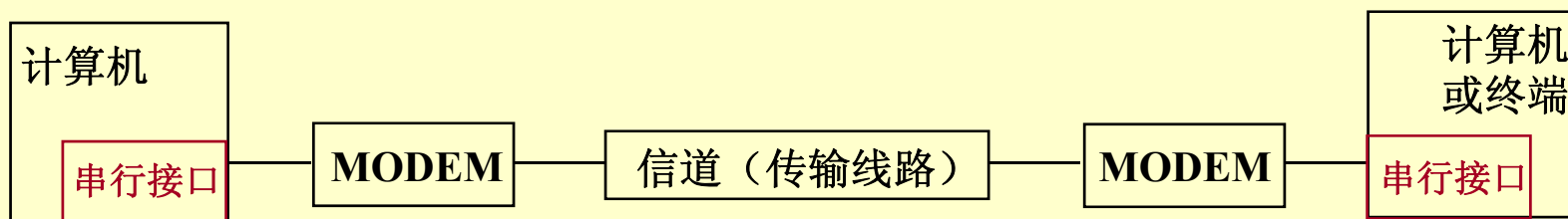
三、串行接口

串行通信系统模型为：



要进行串行通信，还要解决一个问题：计算机与**MODEM**怎样连接？

—— 通过接口电路——串行接口电路连接。



为了使不同的计算机、外部设备都能正确连接，这个接口其：

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。

目前常用的标准是 **RS-232C** 标准。



（一）串行连接的物理接口标准——（ISO 的 OSI 七层模型中的物理层）

不同的计算机、外部设备之间要进行物理连接，都必须遵守统一的规范，美国电子工业协会（EIA）发布了一系列标准，如：RS-232C、RS-422、RS-423、RS-449等。

其中，RS-232C是目前应用最广泛的一种标准。

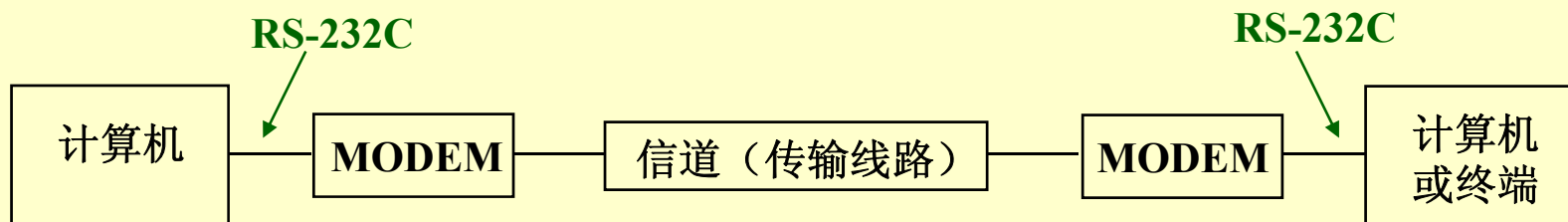
1. RS-232C接口标准

- RS-232C最初是为了使用公用电话网进行数据通信而制定的标准。

在发送端：通过调制解调器将表示为“1”、“0”的高低电平，转换成相应的高低频率的模拟信号，发送到公用电话网。

在接受端：

RS-232C提供了一个利用电话网通过MODEM把远距离设备连接在一起完成通信的技术规范。



- 随着计算机的发展，除了上述连接，也可用 **RS-232C** 使计算机与终端相连。



(1) RS-232C 标准

RS-232C 提出了数据终端设备（**DTE**）和数据通信设备（**DCE**）之间串行传输数据的接口规范，对接口的机械特性、电气特性、功能特性做了规定。

- **机械特性**：标准规定了使用一个**25**针标准连接器（插头座），并对连接器的尺寸、每个针的排列位置做了明确规定。
- **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。可用现成的转换芯片（如**MC1488**、**MC1489**等）转换。

标准规定了**RS-232C**能连接的最大距离为 **15.24 m**（实际可达 **30 m**）。

最大传输速率为 **20k bps**。

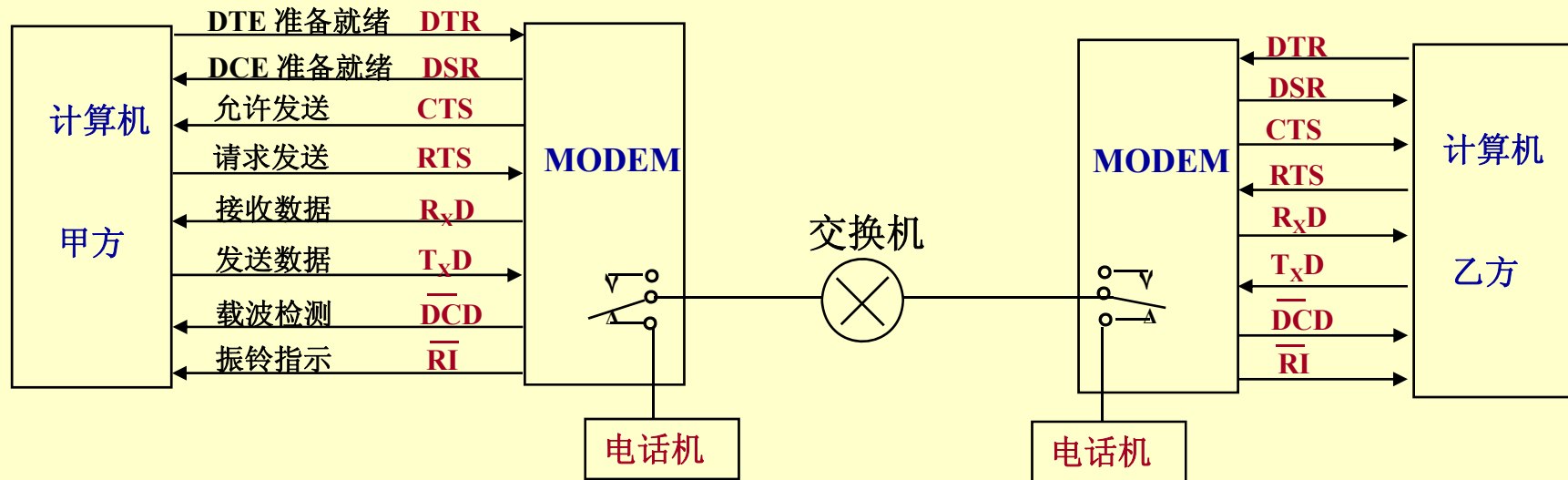


- 功能特性：标准定义了25针连接器中的20条连接线。（下图为常用的信号线）



(2) RS-232C标准接口的应用

—— I. 建立一次通信的过程：



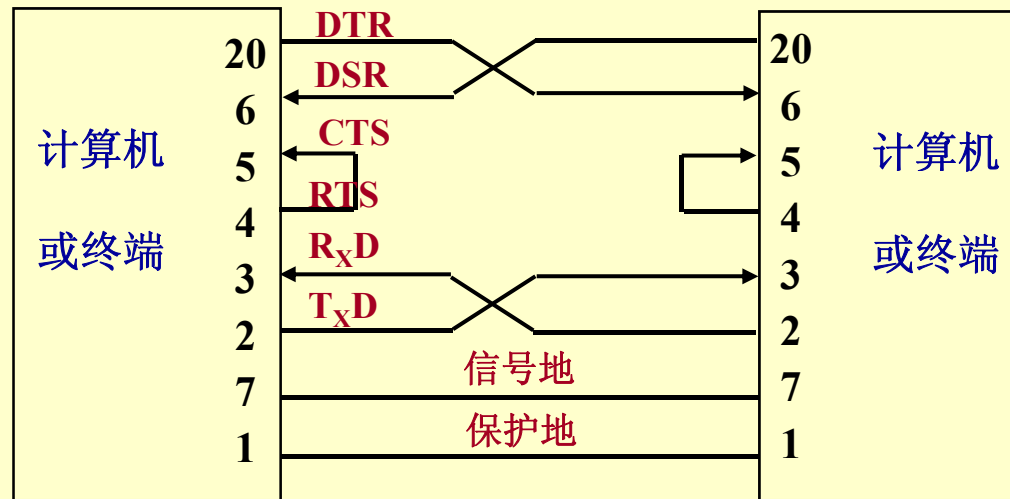
主呼方（甲方）计算机处在准备发送状态，DTR 有效，RTS 有效。

- ① 甲方拨号，呼叫乙方；
- ② 交换机识别所拨号码，使乙方振铃；
乙方MODEM中的振铃检测电路检测到有振铃信号后，
 - 通过 RI ，通知乙方计算机modem已进入收/发工作状态；
 - DCE做好接收数据的准备，DSR和DTR 信号有效，通知modem可以接收/发送数据；接通与外线的连接，起到摘机的作用，交换机停止振铃。
- ③ 乙方准备好后（DTR、RTS、CTS有效），即通过 MODEM 向甲方送一个载波信号；
- ④ 甲方收到此载波信号后，即知乙方已做好准备，使DSR、CTS 有效，并向乙方发送一个载波频率。

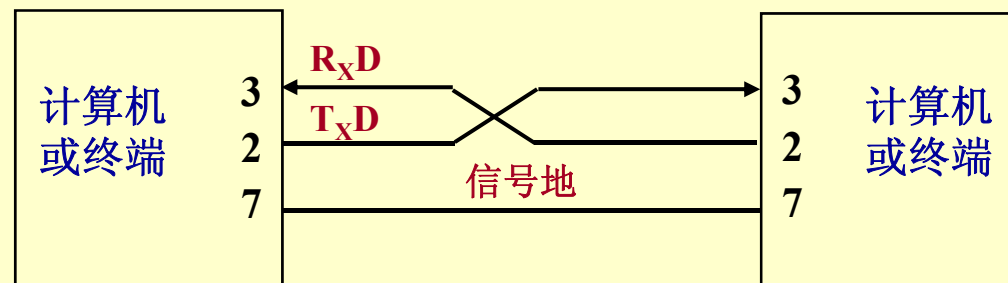
至此，双方已沟通了数据通道，可以开始进行数据通信。



RS-232C标准接口的应用 —— II. 计算机终端之间的 RS-232C 对接



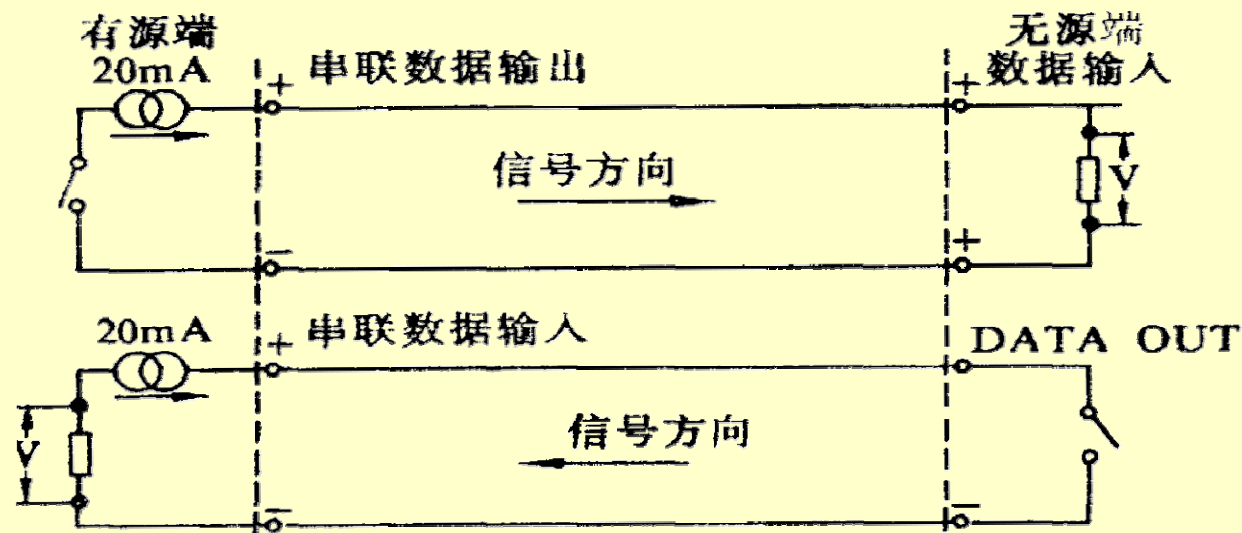
最简单方式



2. 20 mA 电流环接口

20 mA 电流环是一种比较老的接口方法，主要用于电传打字机、纸带读入机及穿孔机中，但其传输距离较远，现在在某些场合仍然使用。

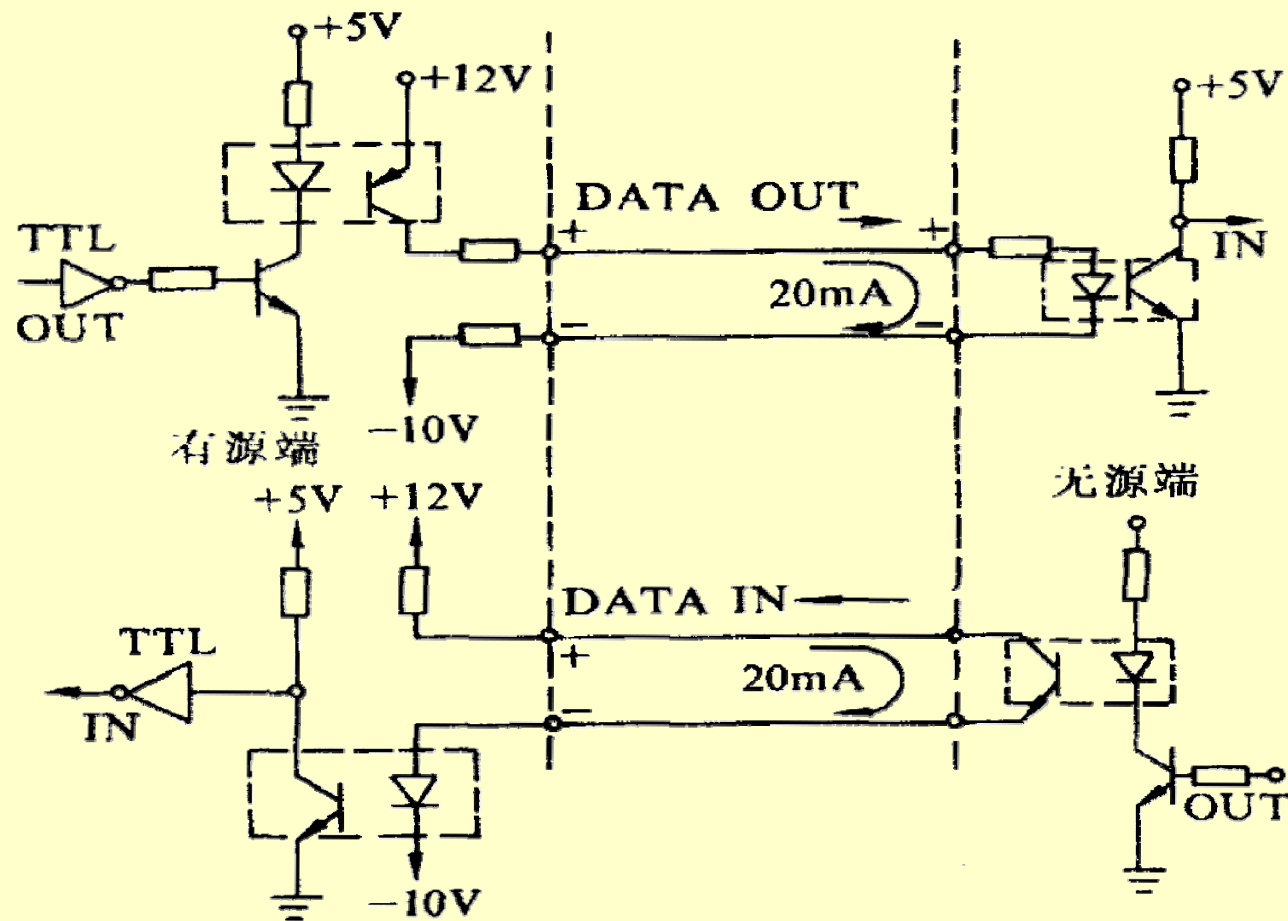
电流回路原理



数据由发送端合上与断开开关传送 { 合上，电流流通，表示为“1”
断开，电流不通，表示为“0”

故可在另一侧检测电流的流通来收集数据信号。

典型电路 —— 光电隔离的电流回路



该电路把开关的闭合与断开变为发光二极管导通和截止，然后由光敏三极管对光信号进行放大与转换，变为有无 **20mA** 电流，在线路上传送。

四、串行传送的实现

串行传送，数据是一位一位传送的：

发送时：将并行数据转换为串行的输出；

接收时：将串行数据转换成并行的，再进行处理。

这样的转换可以用软件、也可以用硬件实现。

1. 软件实现

编写一个发送和接收的程序模块。

例如：CPU的数据通过电传打字机串行输出，一个字符用 7 位 ASC II 码表示，外加一位奇偶校验位，一位起始位，两位停止位，共11位，若电传打字机的速度为 110 波特，且要输出的字符已在 DL 寄存器当中，则串行输出程序如下：



TTYOUT:	MOV CL, 0BH	;设置输出位数
	MOV AL, DL	
	OR AL, AL	;清进位标志C (OR 指令使C=0)
	RCL AL, 1	;C到AL的 bit 0 , 即设置起始位
MORE:	OUT (02H), AL	;输出给TTY
	CALL DELAY	;于110波特相符合
	RCR AL, 1	;恢复原来字符
	STC	;使 C=1 , 设置停止位
	DEC CL	
	JNE MORE	
	RET	

DELAY PROC	;延时子程序, 延时时间= 1 / 110 秒
------------	-------------------------

由此可见, 串并的转换完全由CPU通过软件来实现, 外部只要增加简单的电平转换电路即可。但这样, CPU负担加重。



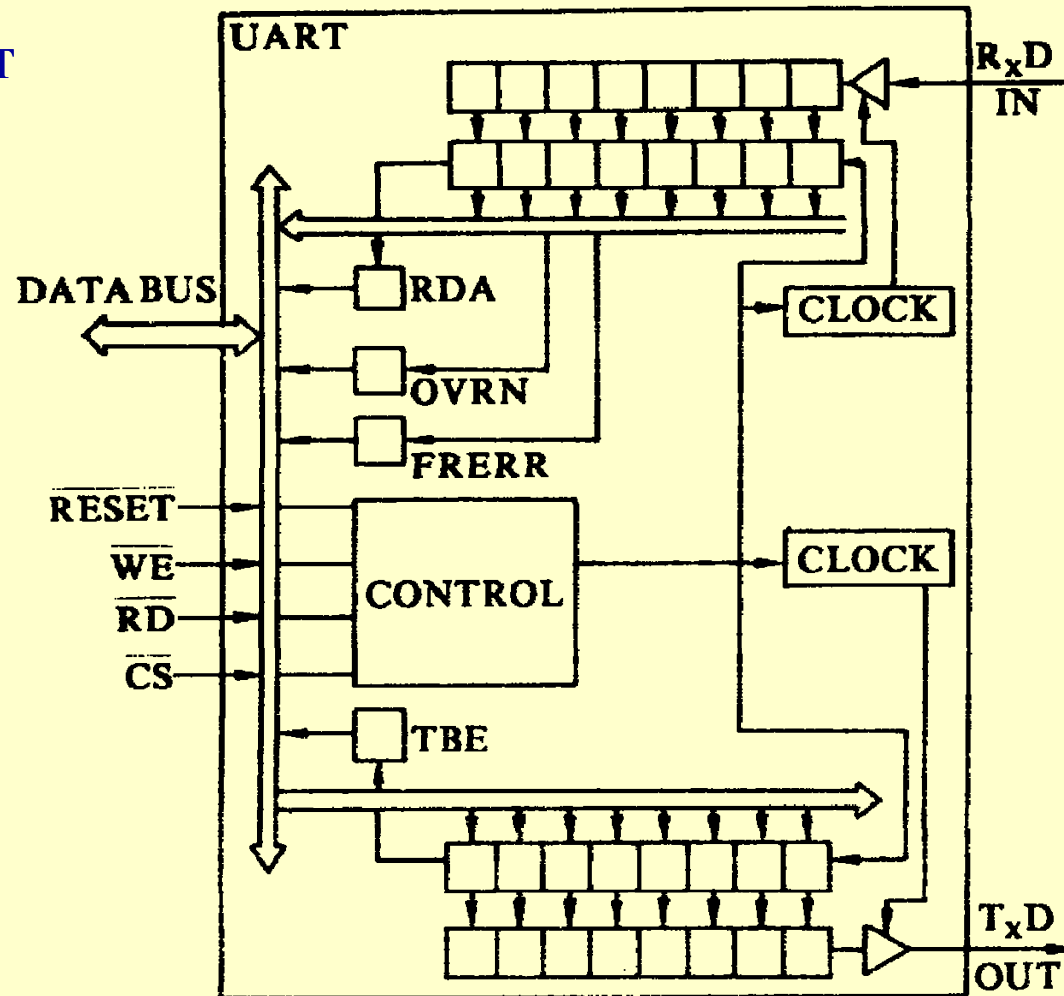
2. 硬件实现

为减轻CPU负担，目前，大多采用串行接口芯片来完成上述的转换工作。

通用异步接收器/发送器 UART

接收时：从R_xD来的串行数据先进入移位寄存器，变成并行的，然后输入给缓冲器，由数据总线输送到CPU。

发送时：从CPU来的并行数据由输出缓冲器接收，然后送至移位寄存器，一位一位地输出至T_xD。



为了控制与监视UART的工作，UART中还有一些控制和状态信息

RDA—— 输入缓冲器满；

TBE—— 输出缓冲器满；

奇偶错误标志（**Parity Error**）—— 在接受时，UART检查接收到的每一个字符的“1”的个数，若不符合要求，则置这个标志，发出奇偶校验错误信息。

帧错误（**Frame Error**）—— 若接收到的字符格式不符合规定，则置出错标志。发出帧错误信息。

溢出（**Overrun Error**）——

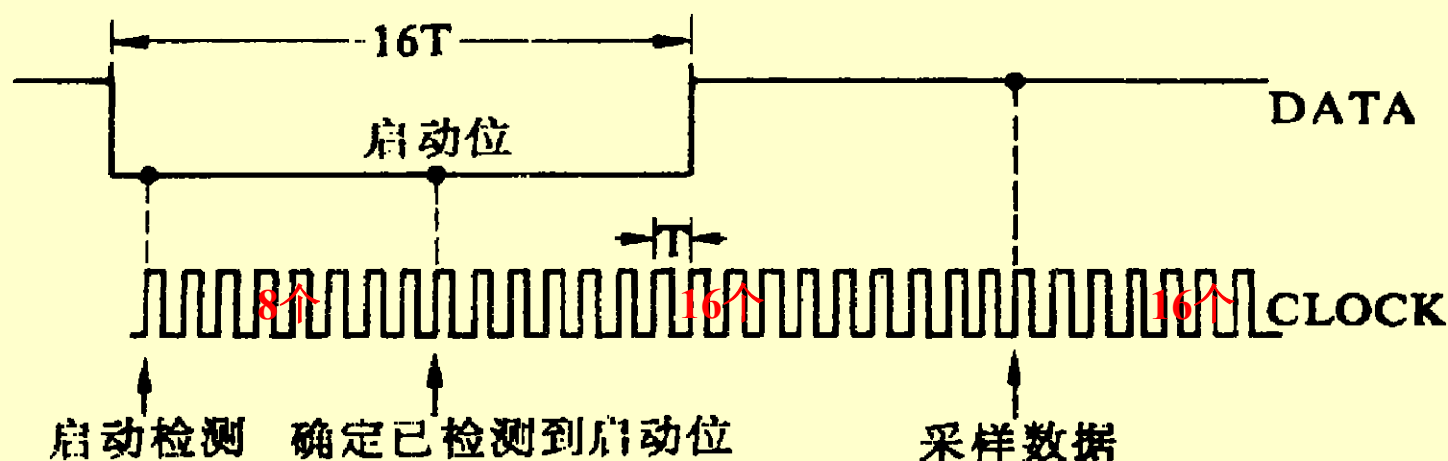


UART 是用外部时钟来和接受的数据同步的

外部时钟的周期 T_C 和数据位的周期 T_d 之间的关系如下：

$$T_C = T_d / K$$

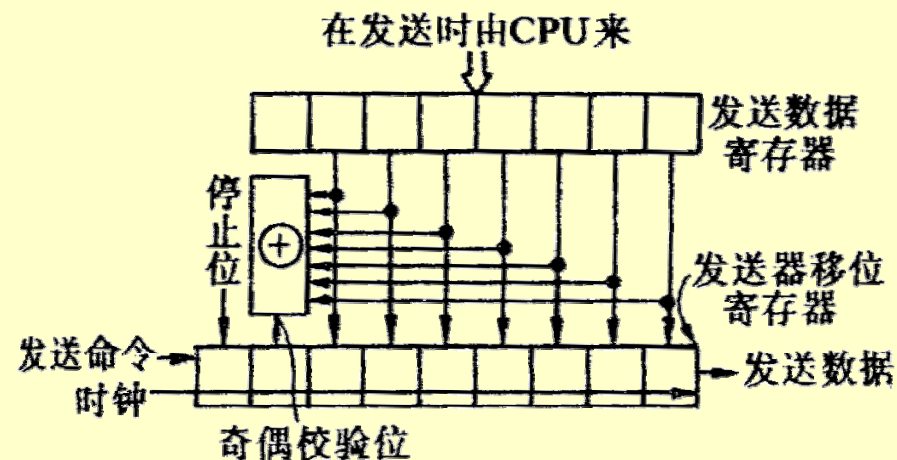
其中， $K=16$ 或 64 。



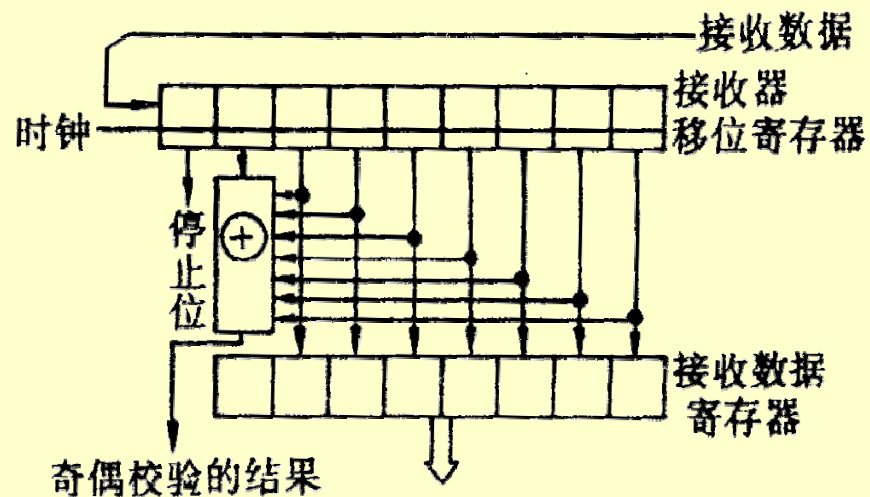
若 $K=16$ ，在每一个时钟周期的上升沿采样接收数据线，若发现了第一个“0”（因数据的起始位为“0”），以后又连续采样到 8 个“0”，则确定其为起始位（而不是干扰信号），以后每隔 16 个周期采样一次数据线，作为输入数据。

UART中的奇偶校验电路

发送时产生奇偶校验位的电路



接收时奇偶校验电路



五、可编程异步通信接口芯片8250

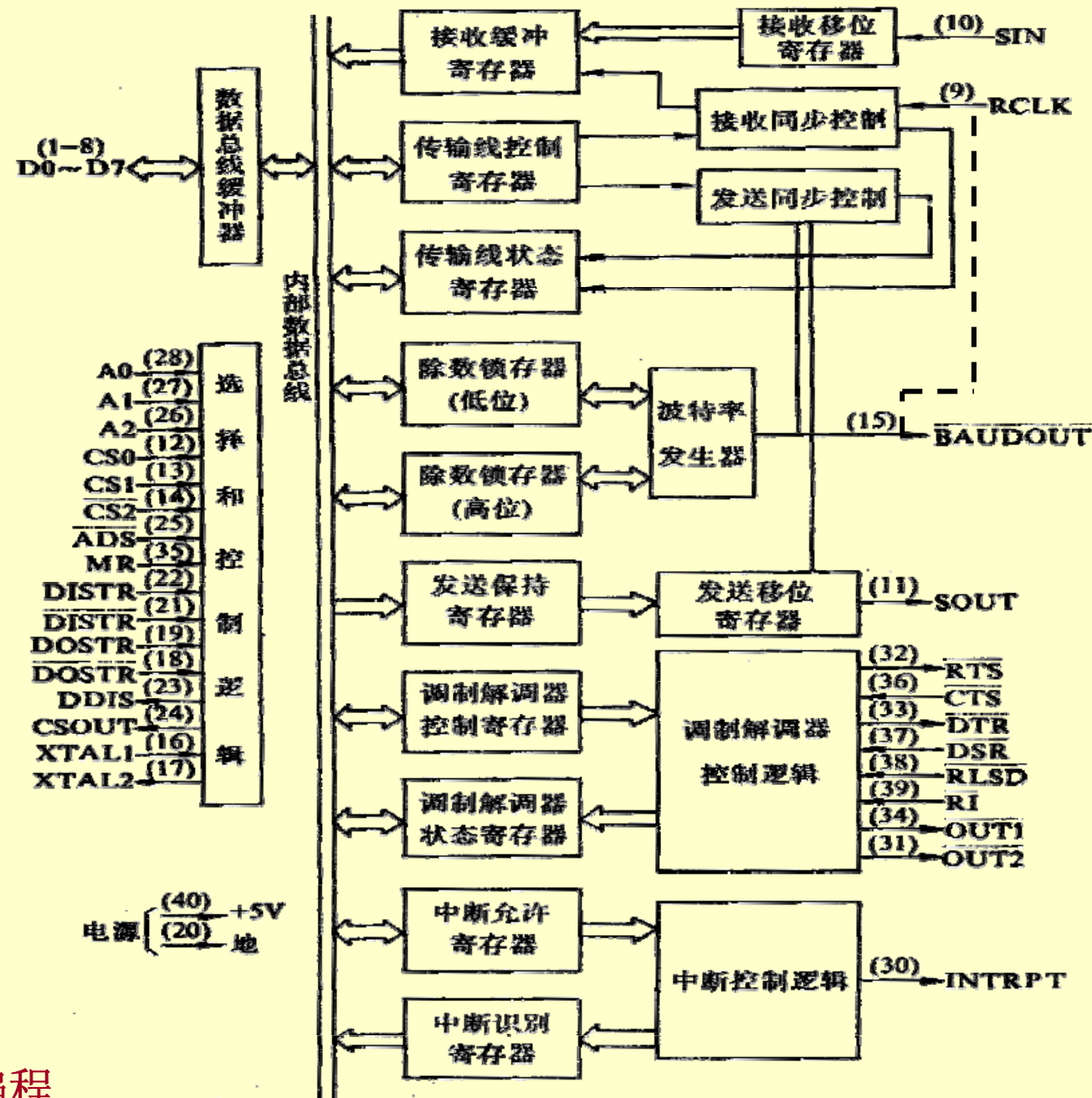
8250 是国家半导体公司推出的一种异步通信控制器。被 **IBM** 等许多计算机公司广泛采用，作为它们生产的异步通信适配器的核心控制芯片。其主要功能是完成串——并的转换。

1. 基本功能

- 完全双工，双缓冲器发送和接受的异步通信接口电路；
- 通信波特率：50 ~ 9600 ， 共分为 15 种；
- 每个字符可传送 5 ~ 8 位；
- 停止位可编程选择为：1、1.5、2 位；
- 可产生中止字符（输出连续的低电平，以通知对方中止通信）；
- 可进行奇偶校验，选择奇校验还是偶校验；
- 出错检测——具有奇偶、溢出和帧错误等检测电路；
- 片内具有优先权中断控制逻辑，具有很强的中断控制能力。



2. 内部结构



3. 初始化编程



(1) 数据总线缓冲器——8250 与系统数据总线的接口，8250 与CPU 之间的数据、命令、状态信息都是通过此缓冲器进行的。

(2) 选择与控制逻辑——8250 与系统地址总线、控制总线的接口，

- 主要功能：
- 片选8250，及选择内部各寄存器；
 - 为8250 提供时钟信号；
 - 控制 8250 芯片的整个工作过程。

主要信号：

DISTR、 $\overline{\text{DISTR}}$ ——数据输入选通信号。有效时，CPU 可从 8250 中读出数据或状态信息。

DOSTR、 $\overline{\text{DOSTR}}$ ——数据输出选通信号。有效时，CPU 可把数据或命令字写入8250的寄存器中。

$\overline{\text{ADS}}$ ——地址选通信号。有效时，将锁存A2、A1、A0 及 $\overline{\text{CS2}}$ 、 $\overline{\text{CS1}}$ 、 $\overline{\text{CS0}}$ 。

DDIS —— 驱动器禁止信号。有效时，禁止挂在 CPU 和 8250 间数据线上的内部收发器动作；

当 CPU 从8250 读取数据时，DDIS 应置为低电平。

CSOUT —— 片选输出信号。有效时，表示 8250 已被 $\overline{\text{CS2}}$ 、 $\overline{\text{CS1}}$ 、 $\overline{\text{CS0}}$ 选中，可以进行数据传输了。

MR —— 主机复位信号。

XTAL_1 、 XTAL_2 —— 外部时钟输入/输出。



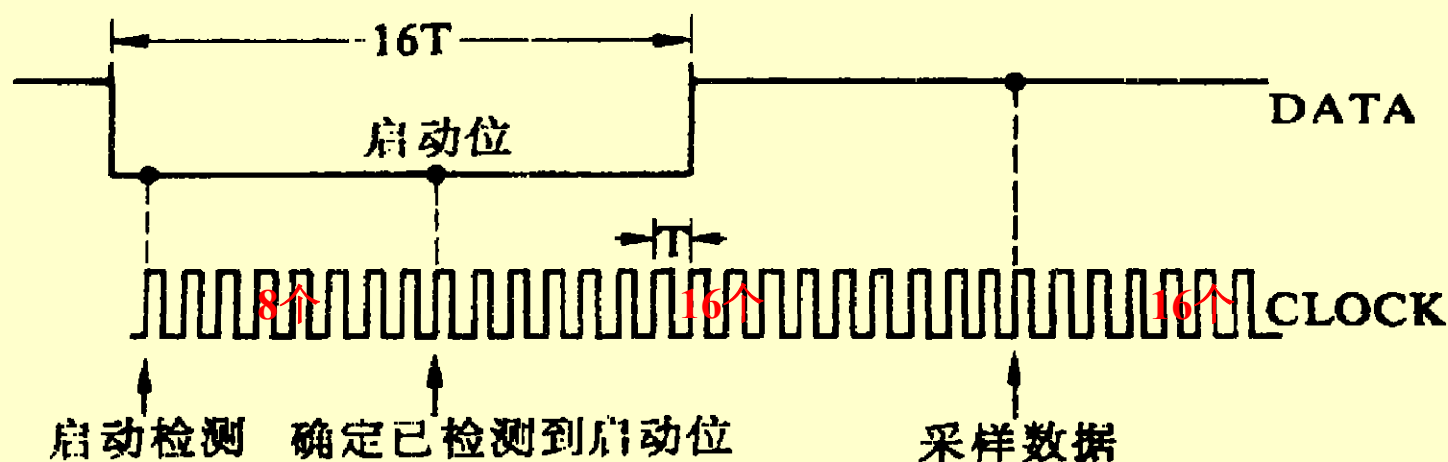
(3) 数据收发部分

① 接收器部分

- 移位寄存器 **RSR** 对 **SIN** 上输入的串行数据进行移位接收，接收控制电路按照规定的数据格式和波特率自动剔除起始位、奇偶位、停止位，将其转换成并行数据，送入接收缓冲器**RBR**。

8250接收时钟**RCLK**使用16倍波特率的时钟信号。**SIN** 变低后，连续测试 8 个 **RCLK** 时钟，若采样到的都是低电平信号，则确认为起始位，然后，每隔16个 **RCLK**时钟，对**SIN** 输入的数据进行采样，直至规定的数据格式结束。

RCLK或由外界提供，或由波特率发生器输出端接入。



- 同时，使 通信线路状态寄存器 **LSR** 的数据接收位置1，可通过中断请求 **CPU** 读取字符。
- **CPU** 读取字符后，使 **LSR** 数据接收位置 0，并复位中断控制逻辑，使 **RSR** 继续接收字符。
- 接收控制电路，在接收数据的同时，还对接收数据的正确性进行监视，如出现奇偶校错误、帧错误等，可请求中断，要求**CPU**处理。

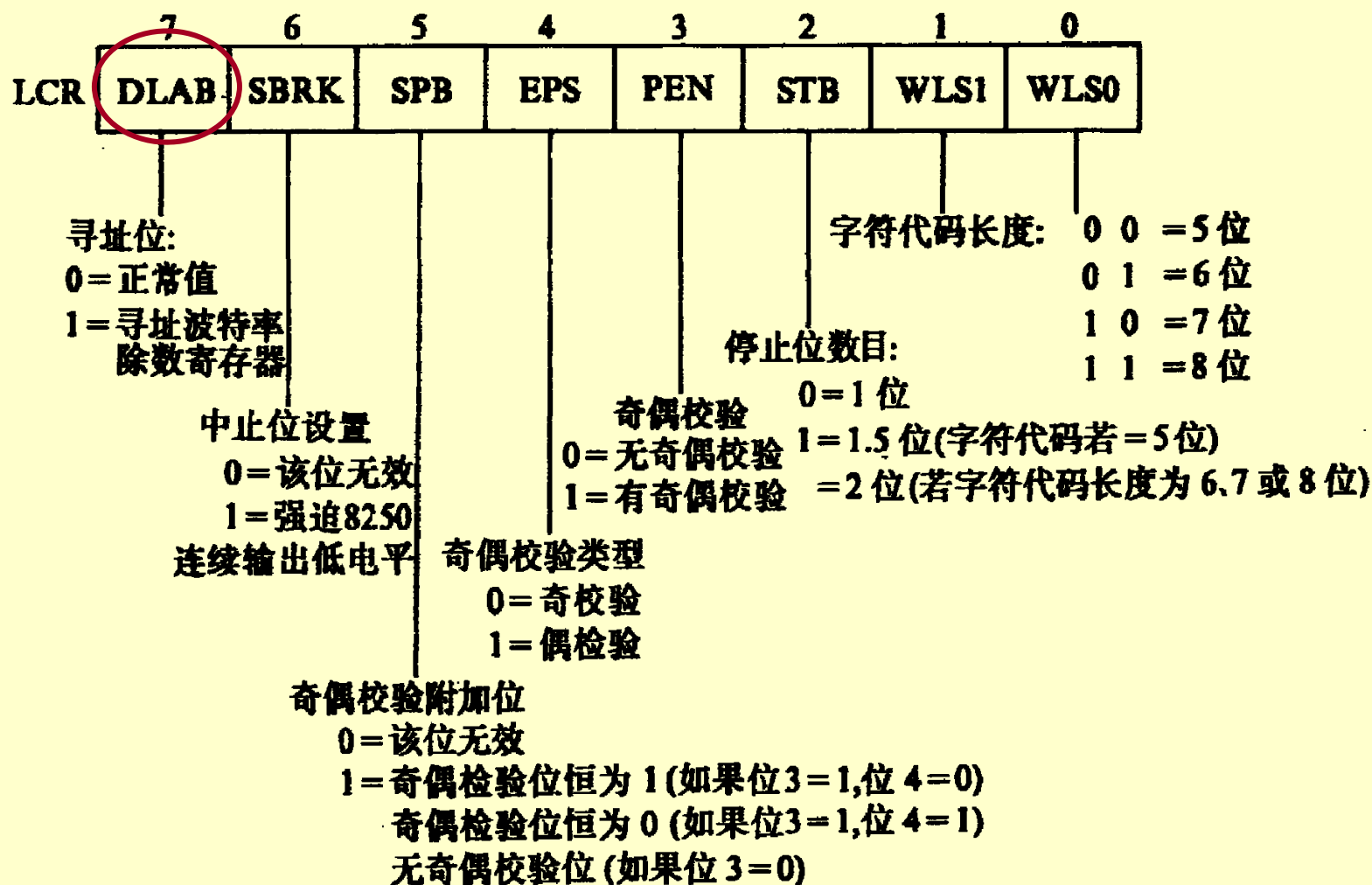
② 发送器部分

- **CPU** 将要发送的数据送入发送保持寄存器 **THR** 中，**8250** 将其送入移位寄存器 **TSR** 中，接收控制电路按照编程设置的数据格式和波特率，加入起始位、奇偶位、停止位，从**SOUT** 中串行输出（低位在前，高位在后）。
- 发送保持寄存器 **THR** 送出数据后，使通信线路状态寄存器 **LSR** 保持寄存器空位置1，可通过中断控制逻辑 向 **CPU** 请求发送下一个字符。
- **CPU** 发送后，使该位置 0，并复位中断控制逻辑。

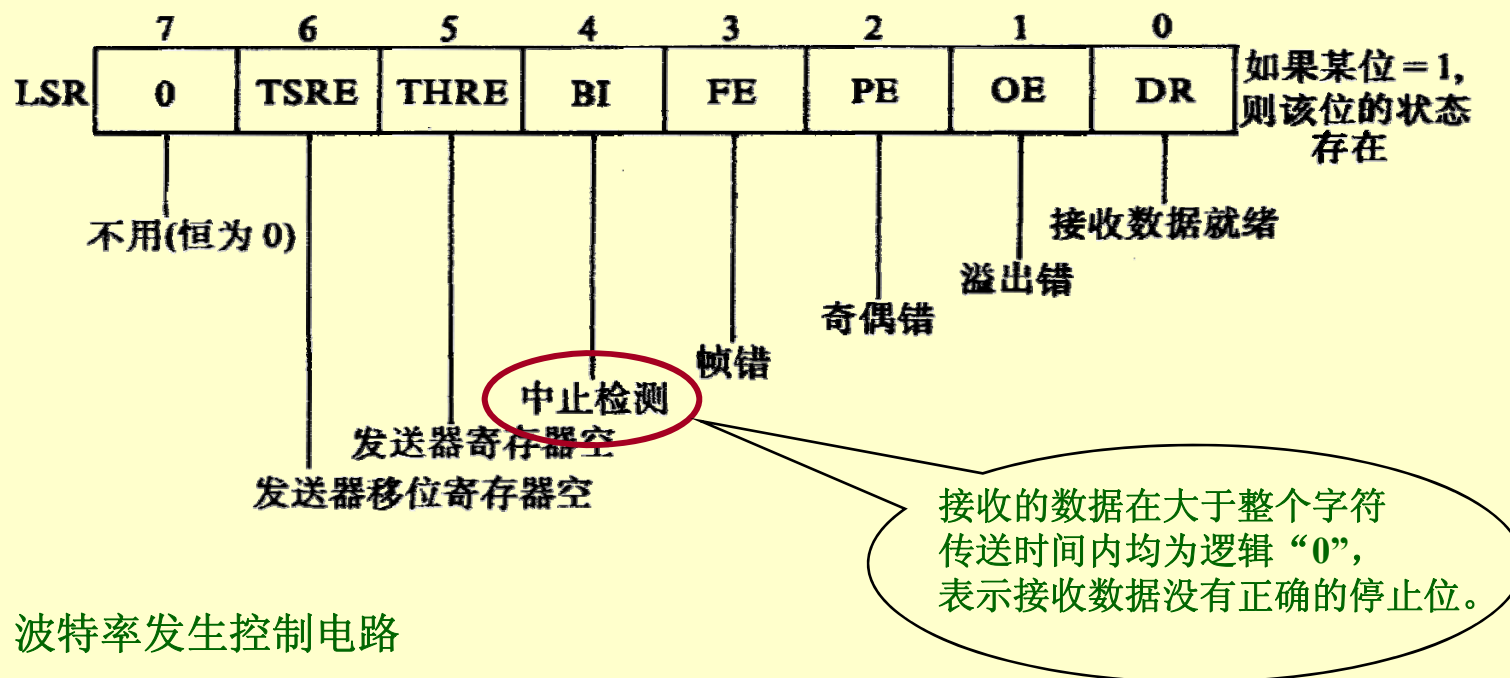


(4) 通信线路控制和状态寄存器

① 通信线路控制寄存器 LCR ——规定通信的数据格式



② 通信线路状态寄存器 LSR —— 向 CPU 提供有关数据传输的状态信息。



(5) 波特率发生控制电路

① 波特率发生器

从 XTAL₁ 引脚接入 1.8432 MHz 的时钟信号，对其分频，产生发送、接收所需的 16*波特率 的时钟信号。分频系数放在除数锁存器中。

$$\frac{1.8432 \text{ MHz}}{\text{除数}} = 16 * \text{波特率}$$

$$\text{除数} = \frac{1.8432 \text{ MHz}}{16 * \text{波特率}}$$



② 除数锁存器 —— 存放分频系数，由两部分组成，高位送入MSB，低位送入LSB。

8250 波特率与分频系数（除数锁存器中的值）的关系。

	波特率除数寄存器的值	
	MSB	LSB
50	09	00
75	06	00
110	04	17
134.5	03	59
150	03	00
300	01	80
600	00	C0
1200	00	60
1800	00	40
2000	00	3A
2400	00	30
3600	00	20
4800	00	18
7200	00	10
9600	00	0C



(6) 调制解调器电路

连接通信设备，如 **MODEM**，设置 8250 与通信设备间的应答联络信号等。

RTS —— 8250 请求发送 ；

DTR —— 数据终端准备就绪；

CTS —— 允许发送；

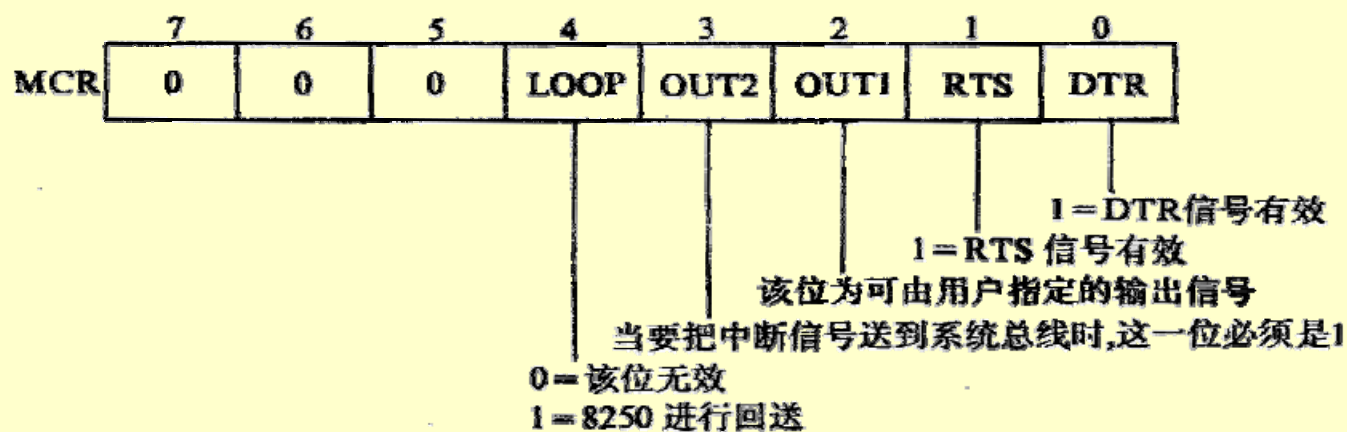
DSR —— **MODEM** 准备就绪；

RLSD —— 载波检测；

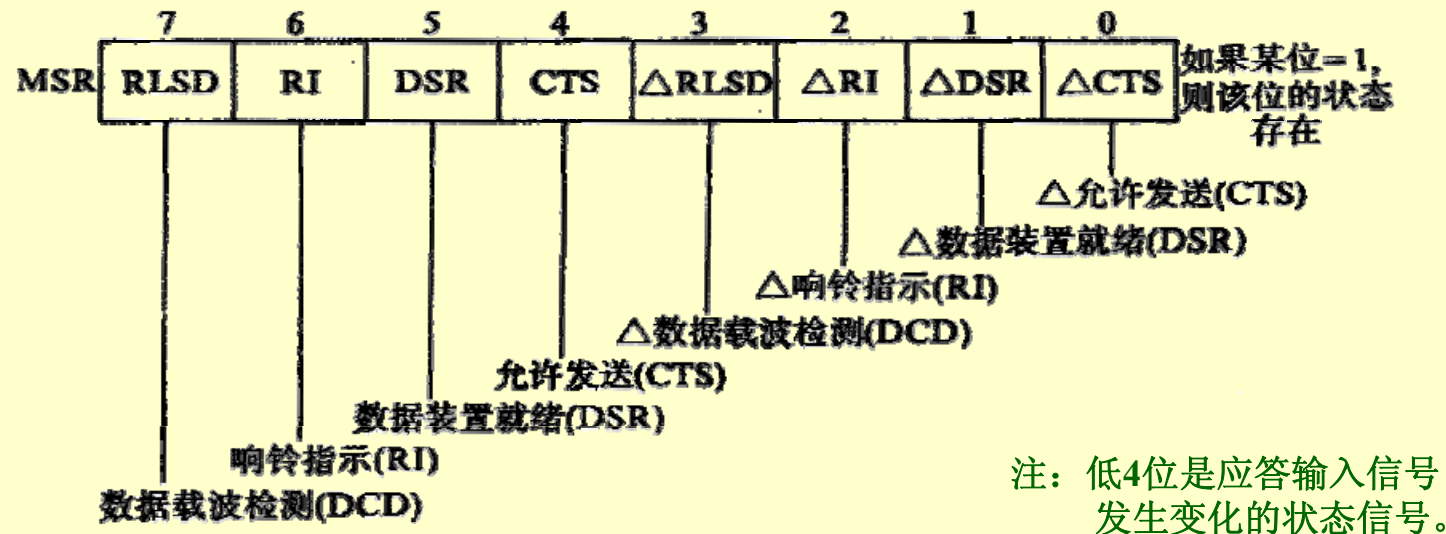
RI —— 振铃检测；

OUT₁、**OUT₂** —— 用户指定的输出端。

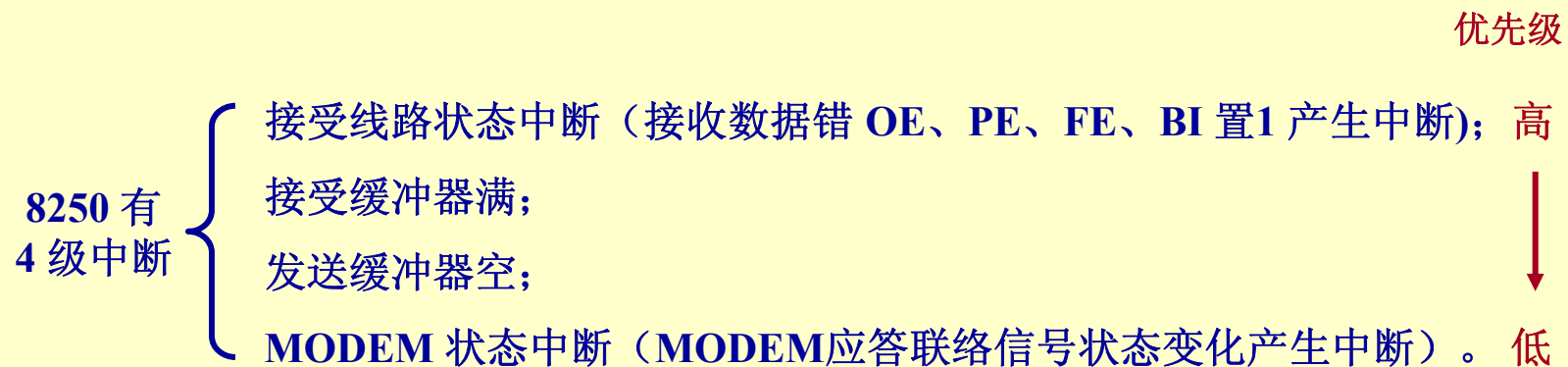
① **MODEM 控制寄存器** —— 用来设置 8250 与通信设备之间的联络应答信号。



② **MODEM 状态寄存器** ——用来反映 8250 与通信设备之间联络应答信号的当前状态和变化情况。



(7) **中断控制** —— 提供中断控制及优先权判决处理。

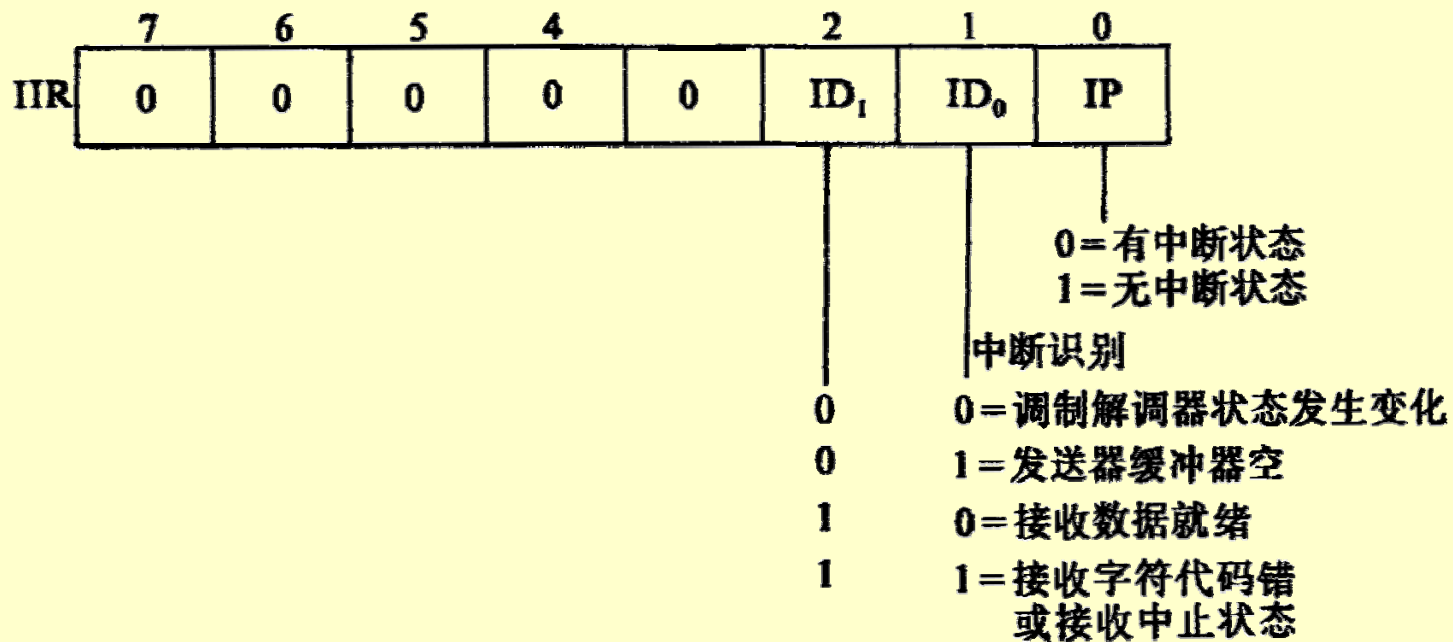


当上述 4 种类型的中断有一种或多种出现时, 8250 便输出 INTPTR 信号。



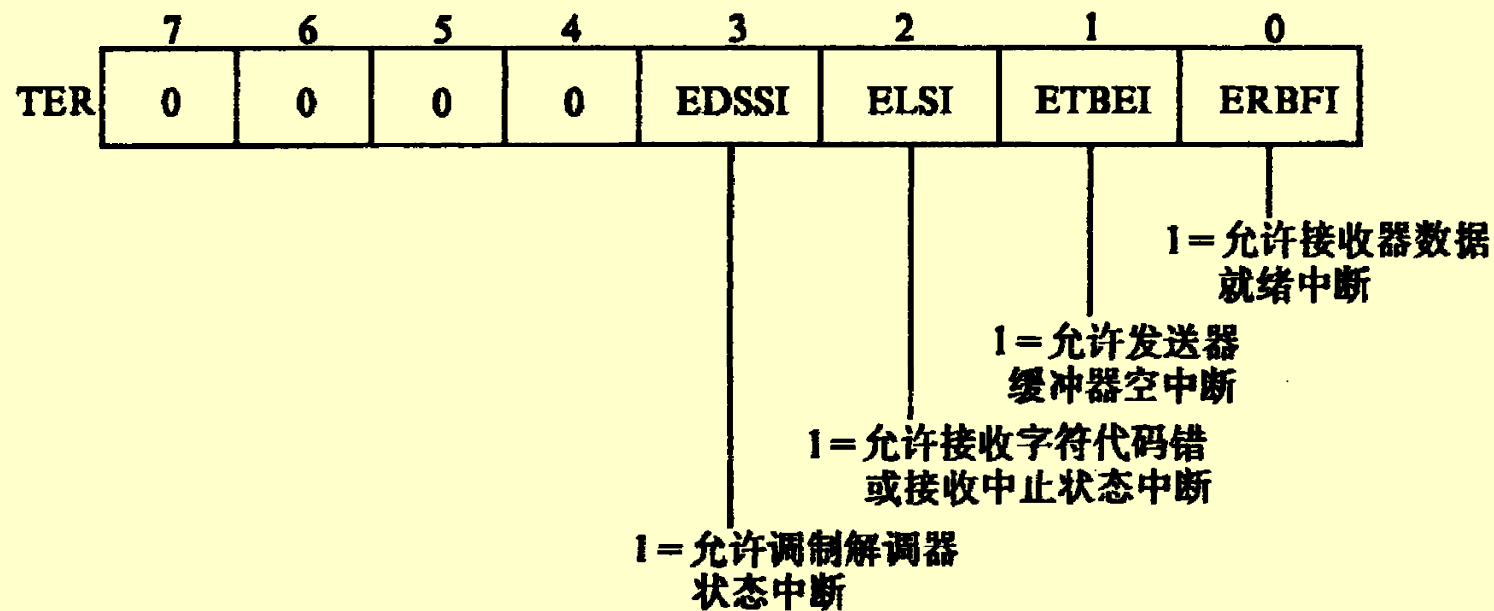
① 中断识别寄存器

中断出现时，为了能具体识别是何种事件引起的中断，设置了中断识别寄存器。它保持正在请求中断的优先级最高的中断类型编码，在这个特定的中断请求由CPU进行服务之前，不接受其它的中断请求。



中断识别寄存器为只读寄存器，其内容随中断源而变化。

② 中断允许寄存器



某位置 “1”，允许中断；

某位置 “0”，屏蔽中断；

中断允许寄存器为读写寄存器。

3. 可存取的内部寄存器及寻址

8250 共有 10 个可存取的内部寄存器

DLAB	A ₂	A ₁	A ₀	寄 存 器
0	0	0	0	接收缓冲器(读),发送缓冲器(写)
0	0	0	1	中断允许
x	0	1	0	中断识别(只读)
x	0	1	1	通信线路控制
x	1	0	0	调制解调器控制
x	1	0	1	通信线路状态
x	1	1	0	调制解调器状态
x	1	1	1	专用
1	0	0	0	除数锁存器(低 8 位)
1	0	0	1	除数锁存器(高 8 位)

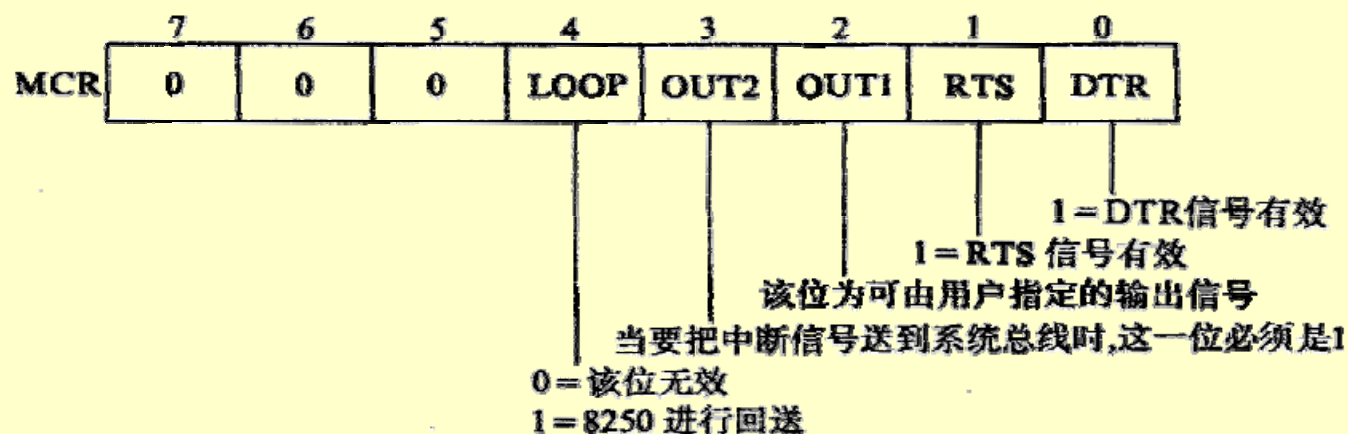
注: **DLAB** 为通信线路控制寄存器**LCR** 的最高位, 设置除数寄存器的值时, 必须先使其为 “1”。



4. 8250 初始化编程

步骤:

- ① 确定波特率 —— 设置除数锁存器 ；
- 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



- 通常，这个寄存器的值设置为 **03H** —— 使 **8250** 输出 **DTR**、**RTS** 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。
- 若要使用中断，则 **OUT₂** 应设置为 “1”，这样，**8250** 中断信号可以通过系统总线送至**8259**中断控制器。



例：要求以 9600 bps 进行异步串行通信，每个字符 7 位，2个停止位，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 $A_2 A_1 A_0$

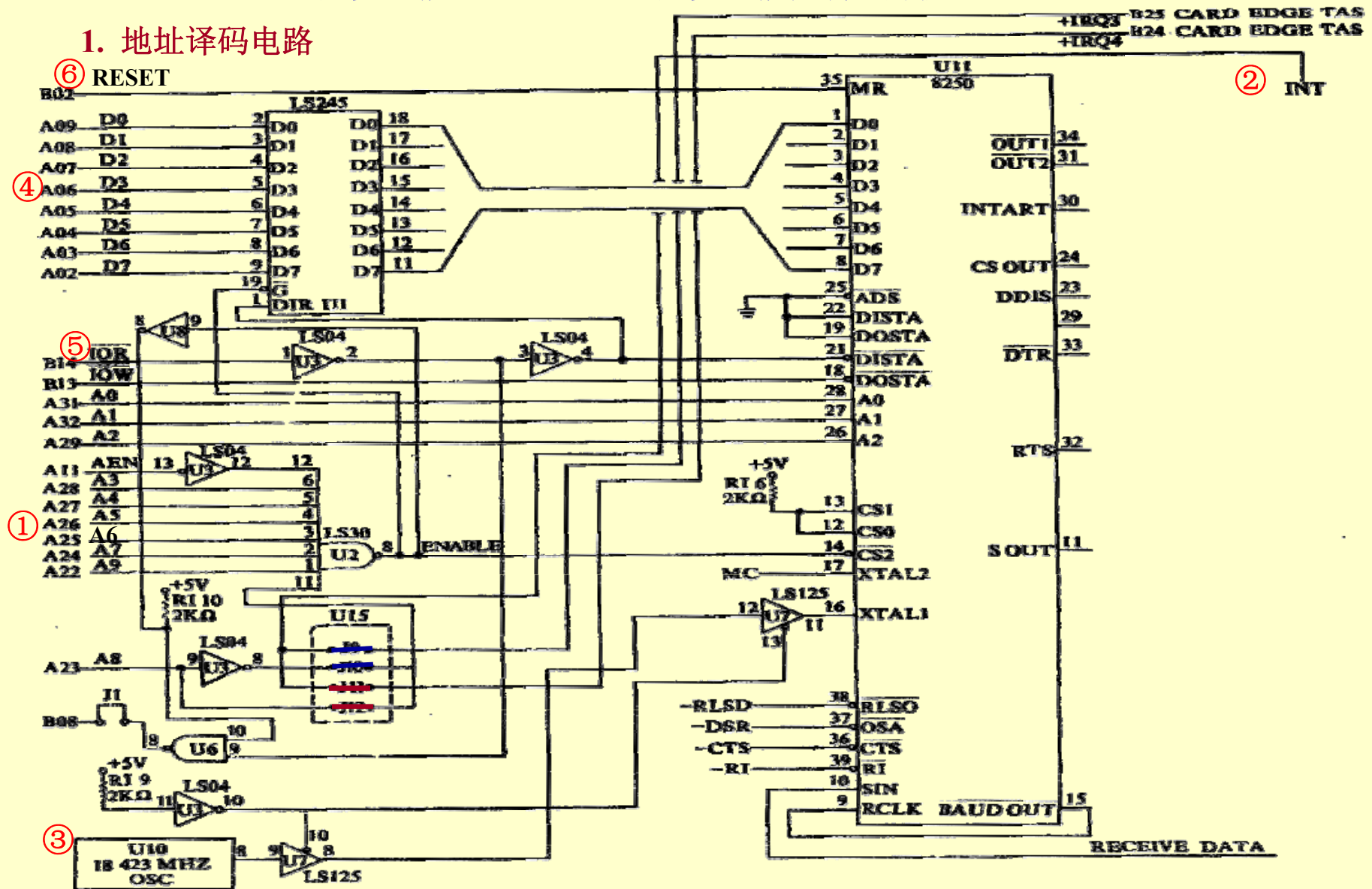
MOV DX, 3FBH	;置除数锁存器（分频系数）
MOV AL, 80H	
MOV DX, AL	;通讯线路控制寄存器最高位置“1”
MOV DX, 3F8H	
MOV AL, 0CH	
OUT DX, AL	;除数低位送入 除数锁存器 LSB（低`8位）
MOV DX, 3F9H	
MOV AL, 0	;除数高位送入 除数锁存器 MSB（高`8位）
OUT DX, AL	
MOV DX, 3FBH	;置通信线路控制寄存器（数据格式）
MOV AL, 00001110B	;7 个字符位，2个停止位，奇校验
OUT DX, AL	
MOV DX, 3F9H	;置中断允许寄存器
MOV AL, 0FH	;允许所有中断
OUT DX, AL	
MOV DX, 3FCH	;置MODEM控制器
MOV AL, 0BH	;使 OUT ₂ 、DTR、RTS 有效
OUT DX, AL	



六、IBM PC/XT 异步通信适配器电路

IBM PC/XT 的异步通信适配器以 8250 异步通信控制器为核心。

1. 地址译码电路



1. \overline{CS}_2 —— A_9 、 $A_7 \sim A_3$ 连至 U_2 。均为高电平信号。

A_8 有两种选择:

① 经反相器、跨接器 J_{10} 连至 U_2 。为低电平。这时信号应为:

A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
1	0	,1	1	1	1	,1			

端口地址范围为: **2F8 ~ 2FFH**

若 J_9 接通, 则本适配器产生的中断请求信号为 **IRQ₃**。

② 经跨接器 J_{12} 直接连至 U_2 。为高电平。这时信号应为:

A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
1	1	,1	1	1	1	,1			

端口地址范围为: **3F8 ~ 3FFH**

若 J_{11} 接通, 则本适配器产生的中断请求信号为 **IRQ₄**。

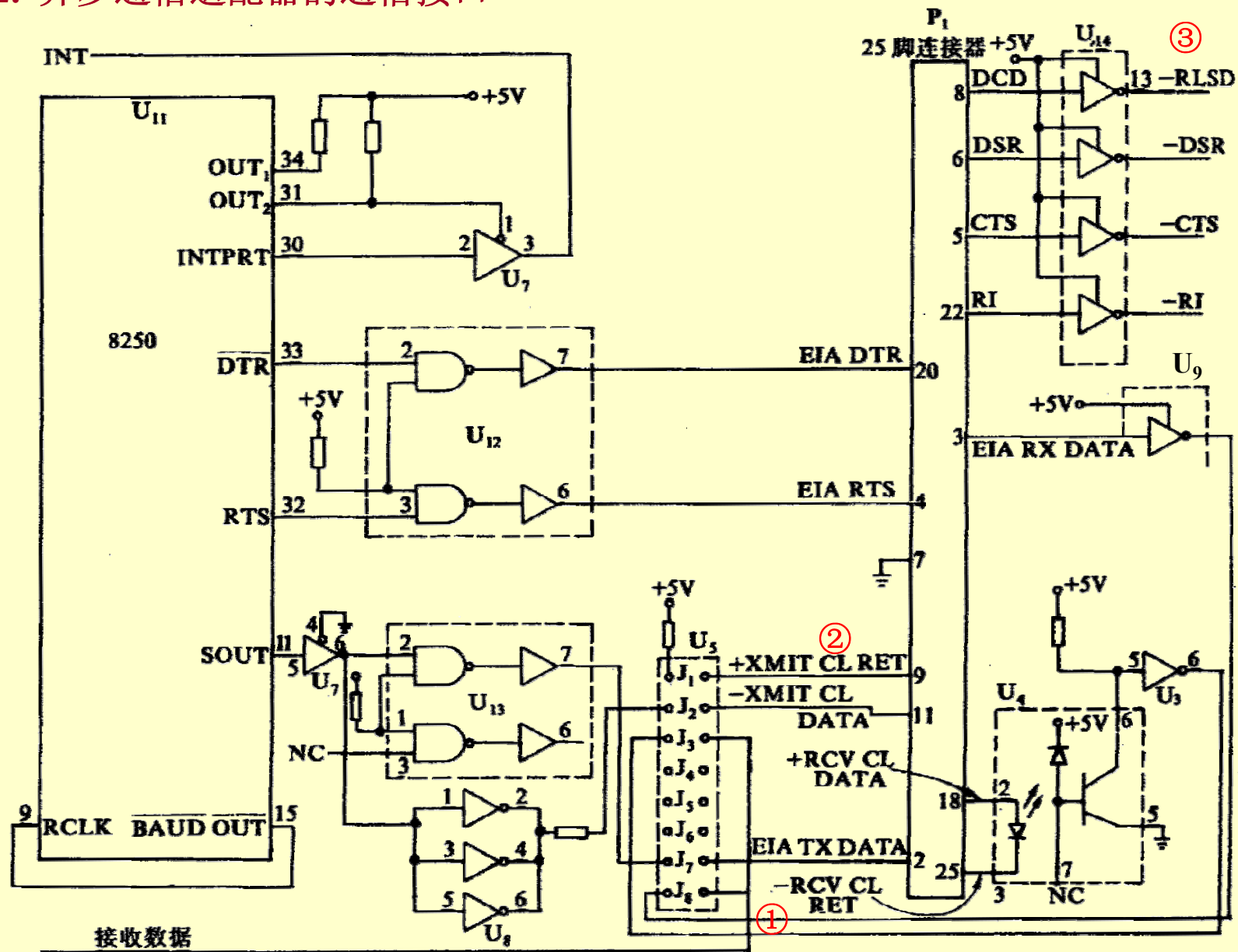
AEN 为低电平信号 (CPU掌握总线)。

2. **XTAL₁** 接至晶体振荡器 U_{10} , 由其提供 **1.8432MHz** 的震荡信号。

3. 数据信号、读写控制信号、复位信号等。



2. 异步通信适配器的通信接口



① RS-232C接口电路:

由U₁₃和U₉组成。当异步通信适配器按RS-232C接口电路工作时，跨接器U₅的J₇、J₈接通。

发送: U₁₃将8250发送的信号SOUT(TTL电平)转换成EIA电平，经U₅的J₇送至25针连接器的输出脚2。

接收: 从25针连接器脚3接收的EIA电平数据经U₉转换成TTL电平，通过J₈送到8250的SIN输入端。

② 20 mA电流环电路: (当异步通信适配器按20 mA电流环接口电路工作时，J₁、J₂接通)

发送: 从SOUT送出的数据经过U₈，产生20mA电流，经J₂、J₁构成电流环路。

接收: 20 mA电流环导致U₄的发光二极管发亮，光敏三极管导通，变成低电平，经U₃反相后，变成TTL电平，送至8250接收端SIN。

③ MODEM 控制信号

通过U₁₄将MODEM四条控制线上的EIA电平转换成TTL电平，送到8250的响应输入上。

通过U₁₂将MODEM输出线上的TTL电平转换成EIA电平，输出到25针连接器上。



3. 异步通信的 BIOS 功能调用——INT 14H

功能

入口参数

出口参数

(1) AH=0 初始化串行口

AL = 初始化参数

AH = 通信线路状态

DX = 串行口号 (0-2)

AL = MODEM状态

AL中需设置参数为:

7	6	5	4	3	2	1	0	
						0	0	代码长
						0	1	5 位
					停止位	1	0	6 位
					0 = 1	1	1	7 位
					1 = 1.5			8 位
					- 2			
								如果代码长为 5 位
								如果代码长为 6 位
								7 位或 8 位
					奇偶校验			
			0	0 = 无				
			0	1 = 奇校验				
			1	0 = 无				
			1	1 = 偶校验				
		波特率						
0	0	0 = 110						
0	0	1 = 150						
0	1	0 = 300						
0	1	1 = 600						
1	0	0 = 1200						
1	0	1 = 2400						
1	1	0 = 4800						
1	1	1 = 9600						



功能

(2) **AH=1** 发送字符

(3) **AH=2** 接收字符

(4) **AH=3** 读串行口状态

入口参数

AL = 欲发送字符

DX = 串行口号 (0-2)

DX = 串行口号 (0-2)

DX = 串行口号 (0-2)

出口参数

AH = 通信线路状态

(AH)₇ = 1 表示传送失败

AH = 通信线路状态

(AH)₇ = 1 表示传送失败

AL = 接收到的字符

AH = 通信线路状态

AL = MODEM状态

