
RS232 串行接口总线

目 录

第一节 RS232 串行接口系统描述.....	3
第二节 RS232 串行接口拓扑结构.....	3
第三节 物理接口.....	5
3.1 电气特性.....	5
3.2 接口信号.....	6
3.3 机械特性.....	8
第四节 电源.....	8
第五节 RS232 接口协议.....	9
5.1 数据传送格式.....	9
5.2 流控制.....	10
5.3 差错检测.....	10
5.4 差错控制.....	10
第六节 系统配置.....	10
6.1 端口地址和中断.....	10
6.2 串行端口寄存器.....	11
6.3 DLAB.....	14
第七节 数据流模型.....	14
第八节 RS232 串口接口设备.....	15

第一节 RS232 串行接口系统描述

串行接口是微型计算机与外部设备的主要通信接口之一。只需要一条信号线就可以进行单向数据传送。由于线路简单，价格相对较低，目前得到广泛应用。

串行通信接口标准经过使用和发展，目前已经有几种。但都是在 RS-232 标准的基础上经过改进而形成的。所以，以 RS-232C 为主来讨论。RS-232C 标准是美国 EIA(电子工业联合会)与 BELL 等公司一起开发的 1969 年公布的通信协议。它适合于数据传输速率在 0~20000b/s 范围内的通信。这个标准对串行通信接口的有关问题，如信号线功能、电器特性都作了明确规定。由于通行设备厂商都生产与 RS-232C 制式兼容的通信设备，因此，它作为一种标准，目前已在微机通信接口中广泛采用。

在讨论 RS-232C 接口标准的内容之前，先说明两点：

首先，RS-232-C 标准最初是远程通信连接数据终端设备 DTE(Data Terminal Equipment)与数据通信设备 DCE (Data Communication Equipment)而制定的。因此这个标准的制定，并未考虑计算机系统的应用要求。但目前它又广泛地被借来用于计算机(更准确的说，是计算机接口)与终端或外设之间的近端连接标准。显然，这个标准的有些规定及和计算机系统是不一致的，甚至是相矛盾的。有了对这种背景的了解，我们对 RS-232C 标准与计算机不兼容的地方就不难理解了。

其次，RS-232C 标准中所提到的“发送”和“接收”，都是站在 DTE 立场上，而不是站在 DCE 的立场来定义的。由于在计算机系统中，往往是 CPU 和 I/O 设备之间传送信息，两者都是 DTE，因此双方都能发送和接收。

目前，PC 系列微机串行接口采用异步通信方式，按照 RS-232 接口标准进行数据传输。

UART(Universal Asynchronous Receiver/Transmitter)是实现数据字符的串并转换的单元。

第二节 RS232 串行接口拓扑结构

使用串行接口进行通信的器件可以分为两类。一类叫做 DCE(DATA COMMUNICATIONS EQUIPMENT)，另外一类叫做 DTE(DATA TERMINAL EQUIPMENT)。DCE 是类似 MODEM 一类的设备。而 DTE 就是计算机或者是计算终端。图 2.1 是典型的串行通信的拓扑结构图。

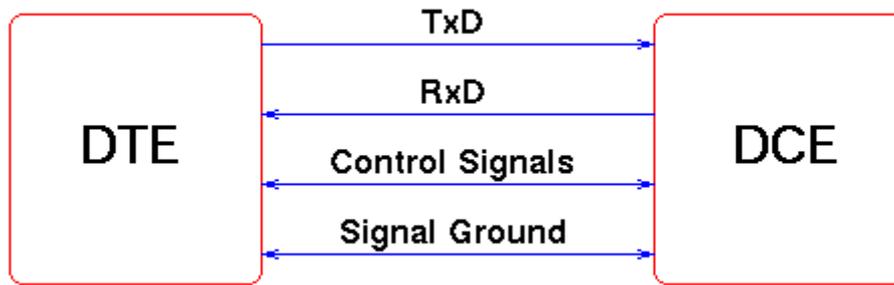


图 2.1 串行通信拓扑结构

尽管微机系统中存在一些单向传输的器件，如只用于发送的游戏杆和只用于接收的打印机。但大多数的情况是全双工，双向的通信。数据在 DTE 和 DCE 之间进行双向传输。

无 MODEM 结构是用来连接两个 DTE 设备的。这通常作为一种较为经济的做法用于联网游戏或者在两台计算机之间使用 ZMODEM 协议或者 XMODEM 协议等进行数据传输。这也可以用于很多微处理器开发系统。图 2.2 是典型的无 MODEM 通信结构图。

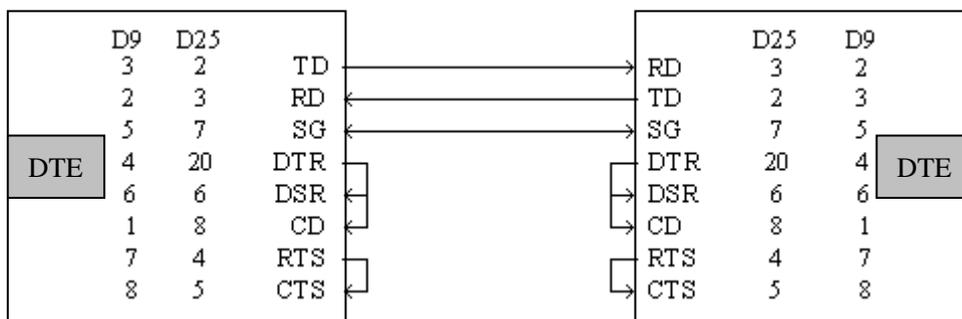


图 2.2 无 MODEM 模式的通信结构图

LOOPBACK 模式的结构在编写串口通信程序的时候使用非常方便。它将发送和接收线连接在一起。因此任何从串口传出的数据会立即回到同一断口的接收端。图 2.3 是典型的 LOOPBACK 模式结构图。

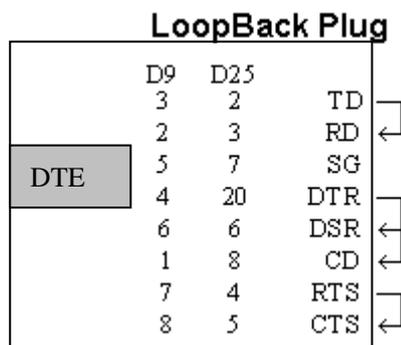


图 2.3 LOOPBACK 通信结构图

第三节 物理接口

RS232 的物理接口包括电气特性和机械特性两部分。

3.1 电气特性

EIA-RS-232C 对电气特性、逻辑电平和各种信号线功能都作了规定。

在 TxD 和 RxD 上:

逻辑 1(MARK)=-3v~-15v

逻辑 0(SPACE)=+3~+15v

在 RTS、CTS、DSR、DTR 和 DCD 等控制线上:

信号有效(接通, ON 状态, 正电压)=+3v~+15v

信号无效(断开, OFF 状态, 负电压)=-3v~-15v

以上规定说明了 RS-232C 标准对逻辑电平的定义。对于数据(信息码): 逻辑“1”(传号)的电平低于-3v, 逻辑“0”(空号)的电平高于+3v; 对于控制信号: 接通状态(ON)即信号有效的电平高于+3v, 断开状态(OFF)即信号无效的电平低于-3v, 也就是当传输电平的绝对值大于 3v 时, 电路可以有效地检查出来, 介于-3~+3v 之间的电压无意义, 低于-15v 或高于+15v 的电压也认为无意义, 因此, 实际工作时, 应保证电平在 $\pm(3\sim 15)v$ 之间。

EIA-RS-232C 与 TTL 转换: EIA-RS-232C 是用正负电压来表示逻辑状态, 与 TTL 以高低电平表示逻辑状态的规定不同。因此, 为了能够同计算机接口或终端的 TTL 器件连接, 必须在 EIA-RS-232C 与 TTL 电路之间进行电平和逻辑关系的变换。实现这种变换的方法可用分立元件, 也可用集成电路芯片。目前较为广泛地使用集成电路转换器件, 如 MC1488、SN75150 芯片可完成 TTL 电平到 EIA 电平的转换, 而 MC1489、SN75154 可实现 EIA 电平到 TTL 电平的转换。MAX232 芯片可完成 TTL \leftrightarrow EIA 双向电平转换, 图 3.1 显示了 1488 和 1489 的内部结构和引脚。MC1488 的引脚(2)、(4, 5)、(9, 10)和(12, 13)接 TTL 输入。引脚 3、6、8、11 输出端接 EIA-RS-232C。MC1489 的 14 的 1、4、10、13 脚接 EIA 输入, 而 3、6、8、11 脚接 TTL 输出。具体连接方法如图 3.2 所示。图中的左边是微机串行接口电路中的主芯片 UART, 它是 TTL 器件, 右边是 EIA-RS-232C 连接器, 要求 EIA 高电压。因此, RS-232C 所有的输出、输入信号都要分别经过 MC1488 和 MC1489 转换器, 进行电平转换后才能送到连接器上去或从连接器上送进来。

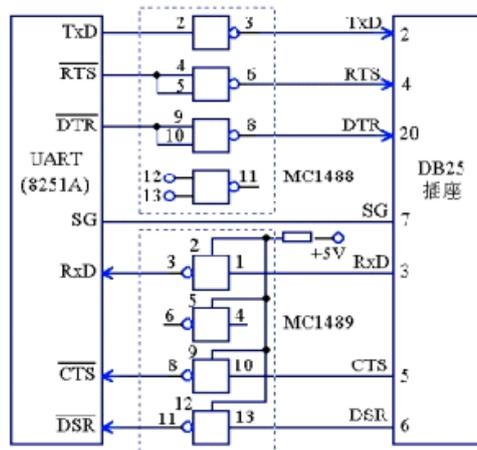


图 3.1 UART 与连接器的连接方法

3.2 接口信号

RS-232C 规标准接口有 25 条线，4 条数据线、11 条控制线、3 条定时线、7 条备用和未定义线，常用的只有 9 根，它们是：

(1) 联络控制信号线：

数据装置准备好 (Data Set Ready-DSR)——有效时 (ON) 状态，表明 MODEM 处于可以使用的状态。

数据终端准备好 (Data Terminal Ready-DTR)——有效时 (ON) 状态，表明数据终端可以使用。

这两个信号有时连到电源上，一上电就立即有效。这两个设备状态信号有效，只表示设备本身可用，并不说明通信链路可以开始进行通信了，能否开始进行通信要由下面的控制信号决定。

请求发送 (Request to send-RTS)——用来表示 DTE 请求 DCE 发送数据，即当终端要发送数据时，使该信号有效 (ON 状态)，向 MODEM 请求发送。它用来控制 MODEM 是否要进入发送状态。

允许发送 (Clear to send-CTS)——用来表示 DCE 准备好接收 DTE 发来的数据，是对请求发送信号 RTS 的响应信号。当 MODEM 已准备好接收终端传来的数据，并向前发送时，使该信号有效，通知终端开始沿发送数据线 TxD 发送数据。

这对 RTS/CTS 请求应答联络信号是用于半双工 MODEM 系统中发送方式和接收方式之间的切换。在全双工系统中作发送方式和接收方式之间的切换。在全双工系统中，因配置双向通道，故不需要 RTS/CTS 联络信号，使其变高。

接收线信号检出 (Received Line detection-RLSD)——用来表示 DCE 已接通信链路，告知 DTE 准备接收数据。当本地的 MODEM 收到由通信链路另一端 (远地) 的 MODEM 送来的载波信号时，使 RLSD 信号有效，通知终端准备接收，并且由 MODEM 将接收下来的载波信号解调成数字两数据后，沿接收数据线 RxD 送到终端。此线也叫做数据载波检出 (Data Carrier Detection-DCD) 线。

振铃指示 (Ringin-RI)——当 MODEM 收到交换台送来的振铃呼叫信号时，使该信号有效 (ON 状态)，通知终端，已被呼叫。

(2) 数据发送与接收线：

发送数据 (Transmitted data-TxD)——通过 TxD 终端将串行数据发送到 MODEM，(DTE→DCE)。

接收数据 (Received data-RxD)——通过 RxD 线终端接收从 MODEM 发来的串行数据，(DCE→DTE)。

(3) 地线

有两根线 SG、PG——信号地和保护地信号线，无方向。

上述控制信号线何时有效，何时无效的顺序表示了接口信号的传送过程。例如，只有当 DSR 和 DTR 都处于有效 (ON) 状态时，才能在 DTE 和 DCE 之间进行传送操作。若 DTE 要发送数据，则预先将 DTR 线置成有效 (ON) 状态，等 CTS 线上收到有效 (ON) 状态的回答后，才能在 TxD 线上发送串行数据。这种顺序的规定对半双工的通信线路特别有用，因为半双工的通信才能确定 DCE 已由接收方向改为发送方向，这时线路才能开始发送。

232 引脚	CCITT Modem 名称	说明	用途	
			异步	同步

1	101	AA	保护地	设备外壳接地	PE	PE ✓
2	103	BA	发送数据	数据送 Modem	TXD	
3	104	BB	接收数据	从 Modem 接收数据	RXD	
4	105	CA	请求发送	在半双工时控制发送器的开和关	RTS	
5	106	CB	允许发送	Modem 允许发送	CTS	
6	107	CC	数据终端准备好	Modem 准备好	DSR	
7	102	AB	信号地	信号公共地	SG	SG ✓
8	109	CF	载波信号检测	Modem 正在接收另一端送来的信号	DCD	
9			空			
10			空			
11			空			
12			接收信号检测 (2)	在第二通道检测到信号		✓
13			允许发送 (2)	第二通道允许发送		✓
14	118		发送数据 (2)	第二通道发送数据		✓
15	113	DA	发送器定时	为 Modem 提供发送器定时信号		✓
16	119		接收数据 (2)	第二通道接收数据		✓
17	115	DD	接收器定时	为接口和终端提供定时		✓
18			空			
19			请求发送 (2)	连接第二通道的发送器		✓
20	108	CD	数据终端准备好	数据终端准备好	DTR	
21			空			
22	125		振铃	振铃指示	RI	
23	111	CH	数据率选择	选择两个同步数据率		✓
24	114	DB	发送器定时	为接口和终端提供定时		✓
25			空			

3.3 机械特性

连接器: 由于 RS-232C 并未定义连接器的物理特性, 因此, 出现了 DB-25、DB-15 和 DB-9 各种类型的连接器, 其引脚的定义也各不相同。下面分别介绍两种连接器。

(1) DB-25: PC 和 XT 机采用 DB-25 型连接器。DB-25 连接器定义了 25 根信号线, 分为 4 组:

- ①异步通信的 9 个电压信号 (含信号地 SG) 2, 3, 4, 5, 6, 7, 8, 20, 22
- ②20mA 电流环信号 9 个 (12, 13, 14, 15, 16, 17, 19, 23, 24)
- ③空 6 个 (9, 10, 11, 18, 21, 25)
- ④保护地 (PE) 1 个, 作为设备接地端 (1 脚)

DB-25 型连接器的外形及信号线分配如图 3.3 所示。注意, 20mA 电流环信号仅 IBM PC 和 IBM PC/XT 机提供, 至 AT 机及以后, 已不支持。

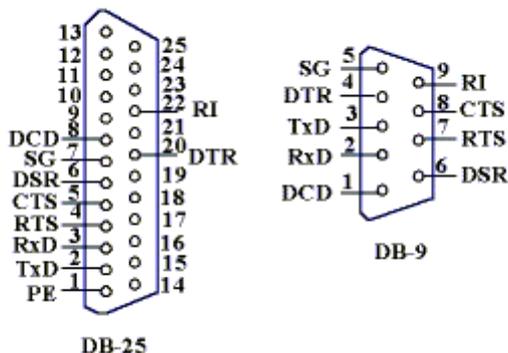


图 3.2 连接器外形和引脚配置

(2) DB-9 连接器

在 AT 机及以后，不支持 20mA 电流环接口，使用 DB-9 连接器，作为提供多功能 I/O 卡或主板上 COM1 和 COM2 两个串行接口的连接器。它只提供异步通信的 9 个信号。DB-25 型连接器的引脚分配与 DB-25 型引脚信号完全不同。因此，若与配接 DB-25 型连接器的 DCE 设备连接，必须使用专门的电缆线。

电缆长度：在通信速率低于 20kb/s 时，RS-232C 所直接连接的最大物理距离为 15m (50 英尺)。

最大直接传输距离说明：RS-232C 标准规定，若不使用 MODEM，在码元畸变小于 4% 的情况下，DTE 和 DCE 之间最大传输距离为 15m (50 英尺)。可见这个最大的距离是在码元畸变小于 4% 的前提下给出的。为了保证码元畸变小于 4% 的要求，接口标准在电气特性中规定，驱动器的负载电容应小于 2500pF。

第四节 电源

无

第五节 RS232 接口协议

串行接口的主要功能是接受来自 CPU 的数据，并转换为串行数据格式发送出去。或者将接收到的串行数据，送交 CPU。图 2.1 示出了串行接口的结构框图。与系统连接的数据总线不但用于传送并行数据，而且接收系统的控制字和向系统发送串行接口的状态。系统送往串行接口的控制字包括发送和接收串行数据的波特率、二进制位串数据格式等。系统读取的状态字包括发送空、接收完成及错误等状态，以便控制发送和接收过程。

发送数据时，CPU 将发送数据送入发送缓冲寄存器。由控制寄存器按照 CPU 设定的格式将并行数据转换为串行数据，加上起始位、奇偶校验位和停止位。转换后并被格式化的二进制位串被送入移位寄存器中，按照选定的波特率发送输出。

接收数据的过程与此相反。控制器按照设定的波特率接收串行位串数据并进行奇偶校验，去掉串行格式中的附加信息，将其转换成并行数据，送入接收缓冲寄存器。无论发送和接收，线路的状态寄存器始终保持着接口的各种状态。通过查询状态寄存器，就可以了解和控制发送和接收过程。

此外，串行接口还包括一个有效的中断系统，无论发送与接收，以及出现的传输错误，都可以引发系统的硬件中断。

5.1 数据传送格式

串行数据通信是把并行数据转换为串行数据按顺序依次发送。接受端再将接受到的串行数据转换成并行数据。串行数据可按两种方式进行，即同步方式和异步方式。同步方式适用于大批量数据通信，每次传送一组数据，在每组数据前加上一定的编制信号，使得双方保持同步。异步数据通信每次发送一个字符，在每个字符的二进制位串前加上启动位作为接收端的同步信号，这样每发送一个字符，双方进行一次同步。

串行异步数据传输有着严格的数据格式和时序关系。在空闲状态，线路呈现出高电平或者“1”状态。传输时，首先发送起始位，接收端接收到起始位，即开始接收过程。在后边的整个二进制位串传输过程中，都是以起始位作为同步时序的基准信号。起始位以“0”表示。紧跟其后的是数据位，根据采用的编码，数据位可能为5位到8位之间。奇偶位可以有也可以没有。处在最后的是停止位，停止位以“1”表示，位数可能是1位、1/2位或者2位。

传送格式中的起始位和停止位在数据传输过程中起着十分重要的作用。通信中有两点影响着数据的正确接收。一是数据发送是随机的，接收端必须随时准备接收数据。另一方面，接收端和发送端不是使用同一个时钟，在通信线路的两端各自具有时钟信号源，虽然可以设定双方的时钟频率一样，但脉冲边沿也不可能一致。脉冲周期、脉冲宽度总有误差。开始发送时，接收端必须准确地检测到起始位的下降沿，使其内部时钟和发送端保持同步。在对每个二进制位接收采样时，采样点应可能接近数据位中点。为了能及时发现起始位的下降边沿，大多数接收器使用高于发送数据波特率十六倍的频率时钟信号进行采样。这样，接收器能够在一个二进制位周期的1/16时间内检测到起始位的下降边。对数据位周期中心采样的偏差也不会超过位周期时间的1/16。



图 5.1 串行异步传输格式

5.2 流控制

如果 DTE 到 DCE 的速度是 DCE 到 DCE 速度几倍的话，PC 就可以用 115.200Kbps 的速度给 MODEM 发送数据了。而数据由于缓冲区的溢出早晚会丢失。因此，我们要采用流控制。流控制有两种：硬件流控制和软件流控制。

软件流控制，有时表示位 XON/XOFF，使用两个字符 XON 和 XOFF。XON 常常用 ASCII 码 17 表示，而 XOFF 常用 ASCII 码 19 表示。MODEM 只有很小的缓冲区所以当计算机将他填满时 MODEM 发出一个 XOFF 字符告诉计算机结束发送数据。一旦 MODEM 有空间供更多的数据使用，它就发 XON 字符，计算机就可以发送更多的数据。这种类型的流控制有这样的优点：它不需要额外的信号线，就像字符通过 TD/RD 线传送那样。但是，在慢速的连接中，每个字节需要 10 位，这样会减慢通信的速度。

硬件流控制，也叫做 RTS/CTS 流控制。它用我们串口电缆中的两根线而不是在数据线中传输的额外的字符来实现。硬件流控制不会像 XON/XOFF 那样减慢传输速度。当计算机

希望发送数据时，激活请求输出线。如果 MODEM 有所需要的空间，MODEM 会通过激活允许发送线来回复，然后计算机开始发送数据。如果 MODEM 没有空间，就不会激活允许发送线。

5.3 差错检测

协议定义了完善的差错检测机制，包括通过奇偶检验判定数据的正确性，帧检验判定数据的完整性。我们可以在发送的数据中在一个字的最后一位和停止位之间加入校验位，进行数据完整性的检测。

奇偶检验可以用五种不同的校验位。Mark parity bit 总是被置于逻辑 1；space parity bit 总是被置于逻辑 0；偶校验位通过计算字中的位数，如果结果是偶数，就置为 1；奇校验位则在结果是奇数时置为 1。后面的两种方法提供了一种检测位电平传输错误的方法。要注意的是并不一定要使用校验位，这样就可以在每一个帧内去掉一位，常称为无校验位帧。

帧检验将由硬件自动实现，当设定了相应的帧长度之后，如果系统检测到实际帧宽度与之不符，将会给出错误标志。

5.4 差错控制

RS-232 通过设置接口寄存器的相应的值，接口设备会对不同的差错产生相应的中断信号。系统可以通过相应的中断服务程序进行差错检测，但协议并没有规定相应的差错控制方法，如果要实现该功能，需要用户在客户端自己添加。

第六节 系统配置

6.1 端口地址和中断

Name	Address	IRQ
COM 1	3F8	4
COM 2	2F8	3
COM 3	3E8	4
COM 4	2E8	3

表 6.1 标准端口地址

上表是一般 PC 机均适用的标准端口地址。如果你的机器恰好是带有微通道总线的 IBM P/S2，那么它的地址和中断是另外一种分配。和 LPT 端口一样，COM 口的基址可以从 BIOS 的数据区读取。

Start Address	Function
0000:0400	COM1's Base Address
0000:0402	COM2's Base Address

0000:0404	COM3' s Base Address
0000:0406	COM4' s Base Address

表 6.2 BIOS 数据区的 COM 口地址

上表所示的是 BIOS 数据区的 COM 口地址，每个地址占 2 字节空间。下面这个用 C 编写的范例程序，告诉我们如何从这些位置获得 COM 口的地址。

6.2 串行端口寄存器

寄存器列表

Base Address	DLAB	Read/Write	Abr.	Register Name
+ 0	=0	Write	-	Transmitter Holding Buffer
	=0	Read	-	Receiver Buffer
	=1	Read/Write	-	Divisor Latch Low Byte
+ 1	=0	Read/Write	IER	Interrupt Enable Register
	=1	Read/Write	-	Divisor Latch High Byte
+ 2	-	Read	IIR	Interrupt Identification Register
	-	Write	FCR	FIFO Control Register
+ 3	-	Read/Write	LCR	Line Control Register
+ 4	-	Read/Write	MCR	Modem Control Register
+ 5	-	Read	LSR	Line Status Register
+ 6	-	Read	MSR	Modem Status Register
+ 7	-	Read/Write	-	Scratch Register

表 6.3 寄存器列表

(1) 中断允许寄存器(IER)

中断允许寄存器可能是 UART 所有寄存器中最易理解的。将 BIT 0 设置为高，则允许接收数据中断，当 CPU 读取接收寄存器或 FIFO 中包含的数据，就会产生一个中断。

BIT 1 是发送存储寄存器为空中断允许位，该中断用来提示 CPU 发送缓冲区为空。BIT 2 是接收线路状态寄存器中断允许位，当该寄存器状态变化时，UART 产生一个中断。同样的，BIT 3 是 Modem 状态中断允许位，BIT 4 到 BIT 7 最为简单，因为它们都是保留位。

(2) 中断识别寄存器(IIR)

中断识别寄存器是只读寄存器，BIT6 和 BIT7 指示 FIFO 缓冲区的状态。当两位均为 0 时，没有 FIFO 缓冲区被激活。对于 8250 或 16450，只有这个状态。如果 BIT7 为 1，而 BIT 6 为 0，代表 UART 的缓冲区有效但是不能使用。当 16550 UART 的 FIFO 缓冲区出现故障，就会使得 FIFO 不能使用。如果两位均为 1，则代表 FIFO 有效而且完全可用。

BIT4 和 BIT5 是保留位。在 16550 或更高版本中，BIT3 代表超时状态中断。

而 BIT0 代表是否有中断发生。如果有中断发生，由 BIT1 和 BIT2 来显示它的状态。这些中断按照优先级状态工作。线路状态中断具有最高优先级，然后从高往低依次是数据中断、发送寄存器为空中断、Modem 状态中断。

(3) FIFO 控制寄存器(FCR)

FIFO 寄存器是只写寄存器，用来控制 16550 或更高版本上的 FIFO(先进先出)缓冲区。BIT0 是 FIFO 接收和发送操作允许位，将该位置为 0 则使得 FIFO 的所有发送和接收操作无效，因此将失去保存在这些 FIFO 缓冲区里的所有数据。

BIT1 和 BIT2 控制 FIFO 发送和接收的清除。BIT1 负责接收缓冲区，而 BIT2 负责发送缓冲区。将这两位置 1 只会清除 FIFO 中的内容，而不会影响移位寄存器。这两位是自动清除的，因此结束时不需要用户自己置 0。

BIT3 是 DMA 模式选择位，适用于 16550 或更高的版本。在此之后的 BIT4 和 BIT5 都是简单的保留位。

BIT6 和 BIT7 都用来设置接收 FIFO 的触发级。例如，BIT7 设置为 1 而 BIT6 设置为 0，那么触发级则为 8 字节。当接收 FIFO 中有 8 字节的数据，就会产生接收数据中断。参看 IIR。

(4) 线路控制寄存器(LCR)

线路控制寄存器设置通信的基本参数。BIT7 是波特率因子访问锁存位（简称为 DLAB），在“什么是 DLAB”中我们已经讨论过这个概念。BIT6 是 j 间断允许位，该位有效时 TD 线路进入间隔（Spacing）状态，使接收 UART 产生中断。将该位设为 0 将不允许中断。

BIT3、BIT4、BIT5 是奇偶校验选择位。对这三位仔细研究，你会发现 BIT3 控制了奇偶校验。也就是说，如果它置 0，则不使用奇偶校验；置 1 则使用奇偶校验。而 BIT5 可被视作是固定奇偶检验（STICKY PARITY）的控制位。固定奇偶校验其实就是传送和检验的奇偶校验位总是固定的 0 或 1。这种方法用来查错是不太有效的，比如前 4 位出错，而固定奇偶校验位上仍然是正确的预置值，这样就不会产生奇偶校验错误。高位固定奇偶校验（STICKY HIGH PARITY）就是奇偶校验位固定为 1；而低位固定奇偶校验（STICKY LOW PARITY）则用 0 来作为奇偶校验位的值。

如果 BIT5 控制固定奇偶校验（STICKY PARITY），那么当该位置 0，而 BIT3 仍为 1 时，将产生普通的奇偶校验。奇校验指奇偶校验位为 1 或 0，来保证传送的字节中 1 的总个数为奇数。偶校验中的奇偶校验位则保证 1 的个数为偶数。这是较好的查错方法，但也不是完美的，因此软件纠错中常常使用 CRC-32 方法。在这种奇偶校验中，当只有一位发生 0、1 变化，那么就会产生奇偶校验错误；但如果有两位都 0、1 跳变了，那么奇偶校验是正确的，也不会产生奇偶校验错误了。

BIT2 设置停止位的长度。将该位置 0，则产生一位停止位，而将其置 1 则根据字长可分别产生 1.5 或 2 位停止位。注意，接收设备只检查第一位停止位。

BIT0 和 BIT1 设置字长，这是很直接的，现在的字长一般是 8bit。

(5) Modem 控制寄存器(MCR)

Modem 控制寄存器是读写寄存器。BIT5、BIT6、BIT7 是保留位，BIT4 激活回送模式。在回送模式下，发送端串行输出进入标记状态，接收端串行输入被断开。发送端输出回送到接收端输入。DSR, CTS, RI 和 DCD 均断开，DTR, RTS, OUT1 和 OUT2 连接到 Modem 控制输入上，此时 Modem 控制输出管脚无效。在该模式下，发送寄存器中的发送的所有数据都被同一块芯片上的接收电路接收，存于接收寄存器中。这可用来测试 UART 的工作情况。

辅助设备输出 2 可连接可控制 UART-CPU 中断进程的外部电路。辅助设备输出 1 一般是断开的，但有些卡中会用作 1.8432MHZ 和 4MHZ（MIDI）的切换。BIT0 和 BIT1 来控制相应的数据线，例如将 BIT1 置 1 就可激活发送线路。

(6) 线路状态寄存器(LSR)

线路状态寄存器是只读寄存器。BIT7 是接收 FIFO 出错位。当 FIFO 中的字节有一次或以上的间断、奇偶校验错误或帧错误发生，该位就为高。

当 BIT6 置高，发送存储寄存器和移位寄存器都为空。UART 的存储寄存器保存将要发送

的并行数据的下一字节。移位寄存器用于将该字节转换为串行数据，这样可以在一根线路上传送数据。当 BIT5 置高，只有发送存储寄存器为空。这两者之间有什么区别呢？BIT6 为高，发送存储寄存器和移位寄存器都为空，没有任何串行转换，因此发送数据线上没有任何动作。而 BIT5 为高只是发送存储寄存器为空，另一个字节可以被数据端口上，就可能发生使用移位寄存器的串行转换。

当接收数据线的逻辑 0（空）状态维持时间超过发送一个完整字所需时间，就会产生间断中断（BIT4）。这包括开始位、数据位和停止位的时间。

当最后一位不是停止位，就会发生帧中断错误（BIT3），超时错误可能会导致这个问题。使用无效的 Modem 连接两台机器或协议分析器，当数据发送的速率和你 UART 设置的接收速率不匹配时，通常会产生帧错误。

当你的程序不能足够快的从端口读取数据，往往会发生超时运行错误。如果寄存器中进来的字节没有及时的被读走，此时如果恰好又接收到另一字节，那么最后的字节将被丢失，从而导致运行超时错误。

BIT0 代表数据是否就绪，也就是说，一个字节已被 UART 接收，并存放在接收缓冲区内等待被读取。

(7) MODEM 状态寄存器(MSR)

Modem 状态寄存器的 BIT0 清除发送变量（delta，指增量），也就是说寄存器最后一次读操作以后，清除发送线有所变化。BIT1 和 BIT3 类似于 BIT0。BIT1 表示数据设置就绪线路上的状态变化，BIT3 代表数据载波检测线路的状态变化。BIT2 代表环指示器追踪沿，当环指示器从低到高跳变时置位。

BIT4 到 BIT7 指示读取数据时数据线的当前状态。BIT7 代表载波检测，BIT6 代表环指示器，BIT5 代表数据设置就绪，BIT4 指示清除发送线路状态。

(8) 擦除寄存器 (Scratch Register)

擦除寄存器不是用于通信的，而是留出一字节的数据空间。它实际是用来判断 UART 是 8250/8250B 还是 8250A/16450，这种功能现在已很少用到，因为 AT 从来就没有 8250/8250B 的设计，因为它不能应付总线速度。

6.3 DLAB

你可能注意到在上面的寄存器列表里有 DLAB 一栏，当 DLAB 设置为 1 或 0 时，一些寄存器会有所变化。这就是为什么 UART 能在 8 个端口地址上带有 12 个寄存器（包括 Scratch Register）的原因。DLAB 是 Divisor Latch Access Bit（波特率因子访问锁存位）的缩写。当线路控制寄存器将 DLAB 设置为 1，就有两个可设置通信速度（以比特率计算）寄存器可供使用。

UART 将会带有一个震荡频率在 1.8432 MHz 左右的晶体。现在的 UART 只有一个 16 分频计数器，将进入的时钟信号简单的 16 分频。如果有 1.8432 MHz 的时钟信号，我们就可以得到 115.2KHZ 的信号，这样 UART 就可以达到 115.2Kbps 的传送和接收速率。这对于一些快速 Modem 和可传递这种速度的设备而言是很有利的，但却这会使得其他设备不能够进行通信。因此 UART 配备了可编程的波特率发生器，由两个寄存器对其控制。

举个例子来说，我们只想要 2400bps 的通信速率，必须将 115.2K 进行 48 分频来得到一个适用的 2400HZ 的时钟。在这里分频因子 48 保存在由 DLAB 控制的两个寄存器中。这个因子可以是 0-65535 中的任何数字（可 16 位存储即可）。UART 数据总线只有 8 位，所以要用两个寄存器，当 DLAB = 1 时，第一个寄存器(基址 + 0)保存波特率因子的低字节，而另一个寄存器(基址+ 1)波特率因子的高字节。

以下是一些更常见的通信速率及其波特率因子（包括高低字节）。注意所有的波特率因子都是以 16 进制表示的。

Speed (BPS)	Divisor (Dec)	Divisor Latch High Byte	Divisor Latch Low Byte
50	2304	09h	00h
300	384	01h	80h
600	192	00h	C0h
2400	48	00h	30h
4800	24	00h	18h
9600	12	00h	0Ch
19200	6	00h	06h
38400	3	00h	03h
57600	2	00h	02h
115200	1	00h	01h

表 6. 4: 常用波特率因子表

第七节 数据流模型

无

第八节 RS232 串口接口设备

通用异步接收器/发送器 (UART) 是一种支持 RS-232C 标准的可编程串行接口芯片。较低档次的 PC 机采用 INS 8250 芯片, AT 机采用与 8250 兼容的 NS 16450 芯片。PC386 机、486 机往往使用定制的包括并行接口等其它功能的专用芯片, 但其基本功能相同。图 4.1 是 16550C 的芯片外观及管脚排列, 通过其特性可以了解异步串行接口的功能和应用。

16550C 实现的是对从外围器件或者 modem 接受来的数据进行串并转换和对从 CPU 接受来得数据进行并串转换。CPU 可以随时读取芯片的状态, 包括完整的 modem 控制能力和一个处理器中断系统可用于减少通信连接中对系统管理的占用。

16550C 包含一个可编程的波特率发生器。他可以对输入的参考时钟加以一个分量, 给内部的发送逻辑电路提供一个 16 倍的发送时钟。这个 16 倍时钟也可以用于接受端逻辑。16550C 可以用在一种替换模式下 (FIFO 模式)。通过对接受和发送的字符进行缓存, 可以减轻 CPU 的过度的软件开销。发送和接收的 FIFO 可以存储 16 个字节, 接收端的 FIFO 对每个字节可以另外包含三个错误状态位。在 FIFO 模式下, 可以选择自动流控制的机制, 它可以减轻软件的负担, 并且通过 RTS 输出和 CTS 输入信号自动控制数据流提高系统的效率。下图是典型的 UART 结构框图。

functional block diagram

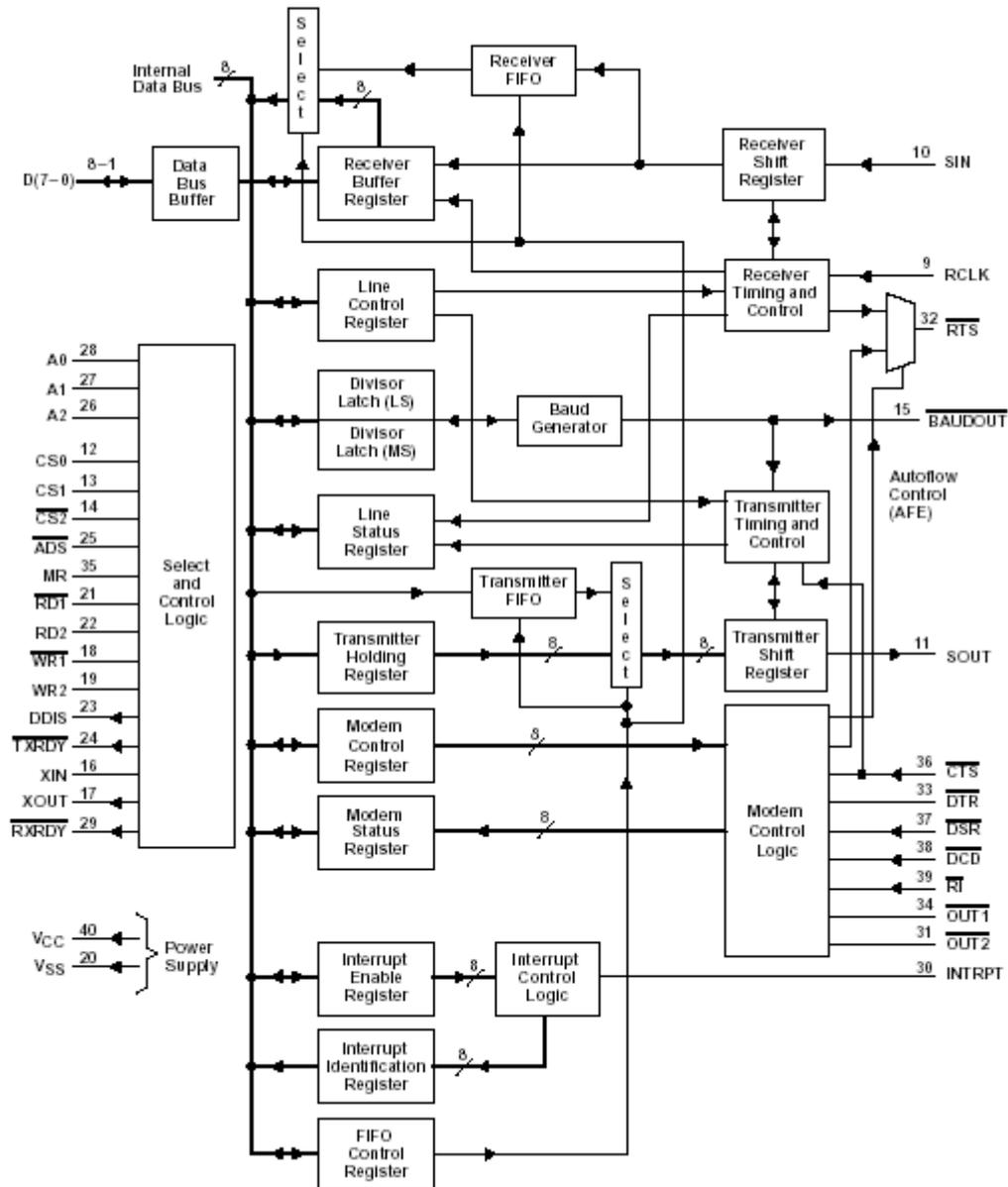


图8.1 UART结构框图