

第8章 串行通信

(课时：4学时)

教学目的

- 掌握串行通信基础知识。
- 了解常用的串行通信总线标准。
- 掌握**MCS-51**单片机的串行通信的基本原理。

学习重点和难点

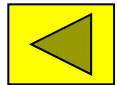
- **MCS-51**单片机串行通信的基本原理。
- 单片机串行通信的应用。

第8章 串行通信

- 8.1 串行通信概述
- 8.2 常用的串行通信总线
- 8.3 MCS-51的串行接口
- 8.4 MCS-51单片机之间的通信
- 本章小结
- 习题

8.1 串行通信概述

- 8.1.1 串行通信的基本方式
- 8.1.2 串行通信的数据传送方式
- 8.1.3 串行通信接口电路

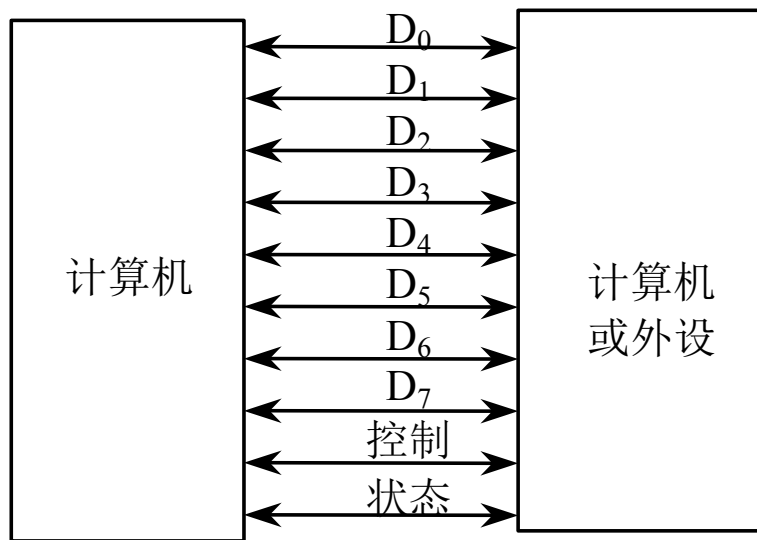


8.1 串行通信概述

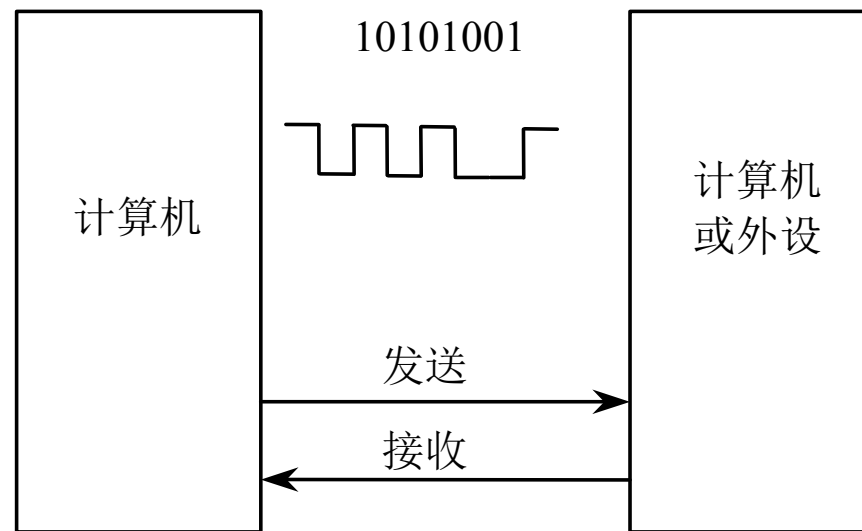
- 在计算机系统中，**CPU**与外部通信的基本方式有两种：
 - 并行通信——数据的各位同时传送。
 - 串行通信——数据一位一位顺序传送。
- 并行通信的特点：各数据位同时传送，传送速度快、效率高。但有多少数据位就需要有多少根数据线，因此传送成本高。在集成电路芯片的内部、同一插件板上各部件之间、同一机箱内各插件板之间等的数据传送都是并行的。并行数据传送的距离通常小于**30米**。
- 串行通信的特点：数据传送按位顺序进行，最少只需一根传输线即可完成，成本低，但速度慢。计算机与远程终端或终端与终端之间的数据传送通常都是串行的。串行数据传送的距离可以从几米到几千公里。
- 串行通信有同步通信和异步通信两种基本方式。

8.1 串行通信概述

基本通信方式示意图



(a) 并行通信



(b) 串行通信

8.1.1 串行通信的基本方式

1. 异步通信方式 (Asynchronous Communication)

- 在异步通信中，数据通常是以字符(或字节)为单位组成字符帧传送的。字符帧由发送端一帧一帧地发送，通过传输线被接收设备一帧一帧地接收。发送端和接收端可以有各自的时钟来控制数据的发送和接收，这两个时钟源彼此独立，互不同步。
- 在异步通信中，接收端是依靠字符帧格式来判断发送端是何时开始发送及何时结束发送。平时，发送线为高电平(逻辑“1”)，每当接收端检测到传输线上发送过来的低电平逻辑“0”(字符帧中起始位)时，就知道发送端已开始发送，每当接收端接收到字符帧中的停止位时，就知道一帧字符信息已发送完毕。

8.1.1 串行通信的基本方式

(1) 字符帧(Character Frame)

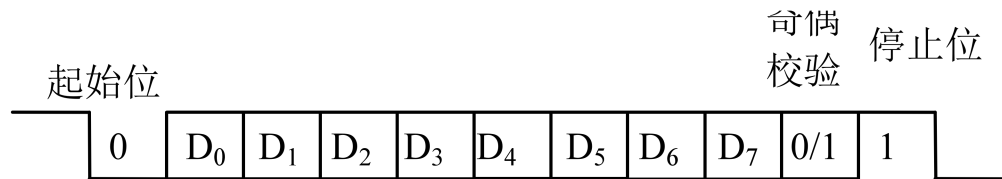
字符帧也叫数据帧，由起始位、数据位、奇偶校验位和停止位等四部分组成。各部分结构和功能分述如下：

- ① 起始位：位于字符帧开头，只占一位，始终为逻辑“0”（低电平），用于向接收设备表示发送端开始发送一帧信息。
- ② 数据位：紧跟起始位之后，用户根据情况可取5位、6位、7位或8位，低位在前高位在后。若所传数据为ASCII字符，则常取7位。
- ③ 奇偶校验位：位于数据位后，仅占一位，用来表征串行通信中采用奇校验还是偶校验，由用户根据需要决定。
- ④ 停止位：位于字符帧末尾，为逻辑“1”（高电平），通常可取1位、1.5位或2位，用于向接收端表示一帧字符信息已发送完毕，也为发送下一帧字符作准备。

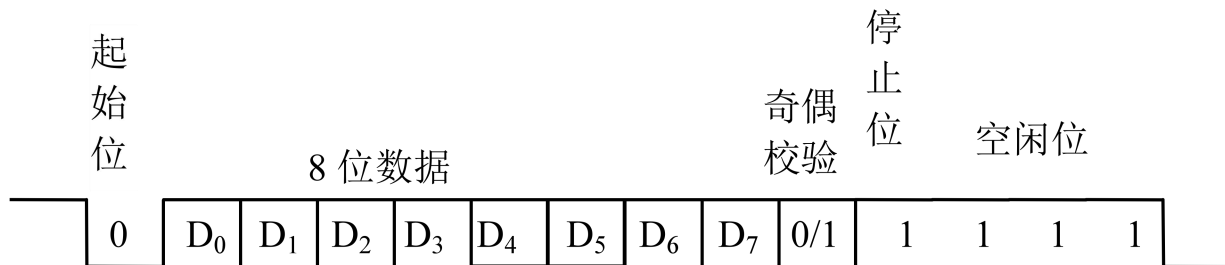
8.1.1 串行通信的基本方式

(1) 字符帧(Character Frame)

在串行通信中，发送端一帧一帧发送信息，接收端一帧一帧接收信息。两相邻字符帧之间可以无空闲位，也可以有若干空闲位，这由用户根据需要决定。



(a) 无空闲位字符帧



(b) 有空闲位字符帧

异步通信的字符帧格式

8.1.1 串行通信的基本方式

(2) 波特率(baud rate)

- 波特率的定义为每秒钟传送二进制数码的位数(亦称比特数), 单位是bit/s, 即位/秒。
- 波特率是串行通信的重要指标, 用于表征数据传输的速度。波特率越高, 数据传输速度越快, 但和字符的实际传输速率不同。字符的实际传输速率是指每秒内所传字符帧的帧数, 和字符帧格式有关。

例如: 波特率为1200 bit/s的通信系统,

若采用上图(a)的字符帧, 则字符的实际传输速率为
 $1200/11=109.09$ 帧/秒

若改用上图(b)的字符帧, 则字符的实际传输速率为
 $1200/14=85.71$ 帧/秒

$$T_d = \frac{1}{1200} = 0.833(\text{ms})$$

8.1.1 串行通信的基本方式

(2) 波特率(baud rate)

- 每位的传输时间定义为波特率的倒数。例如：波特率为1200 bit/s的通信系统，其每位的传输时间应为：

$$T_d = \frac{1}{1200} = 0.833(\text{ms})$$

- 波特率还和信道的频带有关。波特率越高，信道频带越宽。因此，波特率也是衡量通道频宽的重要指标，通常，异步通信的波特率在50bit/s~9600bit/s之间。波特率不同于发送时钟和接收时钟，它通常是时钟频率的1/16或1/64。

$$T_d = \frac{1}{1200} = 0.833(\text{ms})$$

8.1.1 串行通信的基本方式

2. 同步通信(Synchronous Communication)

- 同步通信是一种连续串行传送数据的通信方式，一次通信只传送一帧信息。这里的信息帧和异步通信中的字符帧不同，通常有若干个数据字符，如下图所示。
- 同步字符帧由同步字符、数据字符和校验字符三部分组成。其中，同步字符位于帧结构开头，用于确认数据字符的开始（接收端不断对传输线采样，并把采到的字符和双方约定的同步字符比较，只有比较成功后才会把后面接收到的字符加以存储）；数据字符在同步字符之后，个数不受限制，由所需传输的数据块长度决定；校验字符有1~2个，位于帧结构末尾，用于接收端对接收到的数据字符的正确性的校验。

$$T_d = \frac{1}{1200} = 0.833(\text{ms})$$

8.1.1 串行通信的基本方式

2. 同步通信(Synchronous Communication)

- 在同步通信中，同步字符可以采用统一标准符式，也可由用户约定。
- 在单同步字符帧结构中，同步字符常采用ASCII码中规定的SYN(即16H)代码；
- 在双同步字符帧结构中，同步字符一般采用国际通用标准代码EB90H。

同步字符	数据 1	数据 2	数据 3		数据 n	校验字符 1	校验字符 2
------	------	------	------	-------	--------	--------	--------

(a) 单同步字符帧结构

同步字符 1	同步字符 2	数据 1	数据 2		数据 n	校验字符 1	校验字符 2
--------	--------	------	------	-------	--------	--------	--------

(b) 双同步字符帧结构

同步通信
的字符帧格式

$$T_d = \frac{1}{1200} = 0.833(\text{ms})$$

8.1.1 串行通信的基本方式

2. 异步通信与同步通信的特点

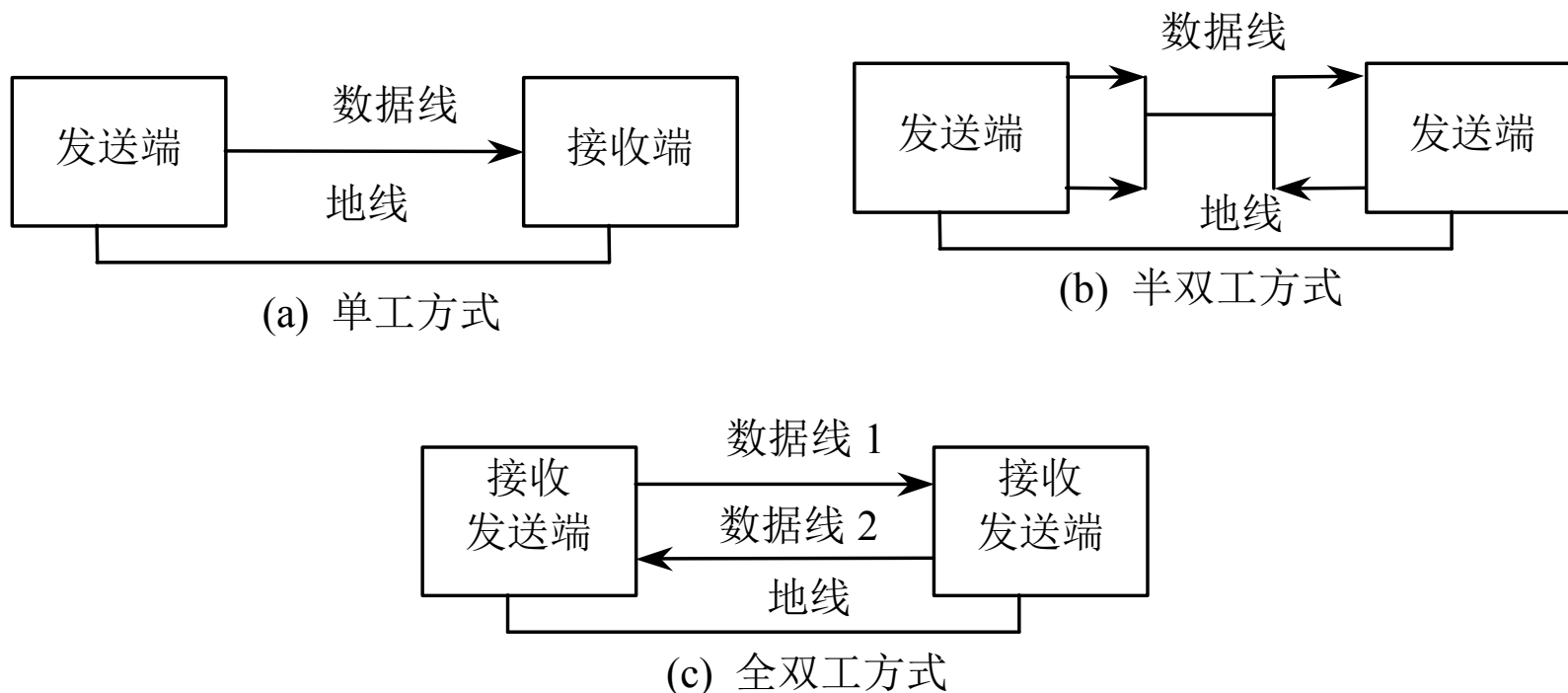
- 异步通信的优点是不需要传送同步脉冲，字符帧长度也不受限制，故所需设备简单。
- 异步通信的缺点是字符帧中因包含有起始位和停止位而降低了有效数据的传输速率。
- 同步通信的优点是传输速率较高，通常可达56000bit/s或更高。
- 同步通信的缺点是要求发送时钟和接收时钟保持严格同步，故发送时钟除应和发送波特率保持一致外，还要求把它同时传送到接收端去。

8.1.2 串行通信的数据传送方式

- 在串行通信中，数据是在两个站之间传送的。按照数据传送方向，串行通信可分为单工、半双工和全双工三种传送方式。如下图所示。
- 单工方式：通信线的一端接发送器，另一端接接收器，它们形成单向连接，只允许数据按照一个固定的方向传送。数据只能单方向传送。
- 半双工方式：系统中的每个通信设备都由一个发送器和一个接收器组成，通过收发开关接到通信线上。数据能够实现双向传送，但任何时刻只能由其中的一方发送数据，另一方接收数据。其收发开关并不是实际的物理开关，而是由软件控制的电子开关，通信线两端通过半双工协议进行功能切换。
- 全双工方式：系统的每端都含有发送器和接收器，数据可以同时两个方向上传送。

8.1.2 串行通信的数据传送方式

- 尽管许多串行通信接口电路具有全双工功能，但在实际应用中，大多数情况下只工作于半双工方式，即两个工作站通常并不同时收发。这种用法并无害处，虽然没有充分发挥效率，但简单、实用。



8.1.3 串行通信接口电路

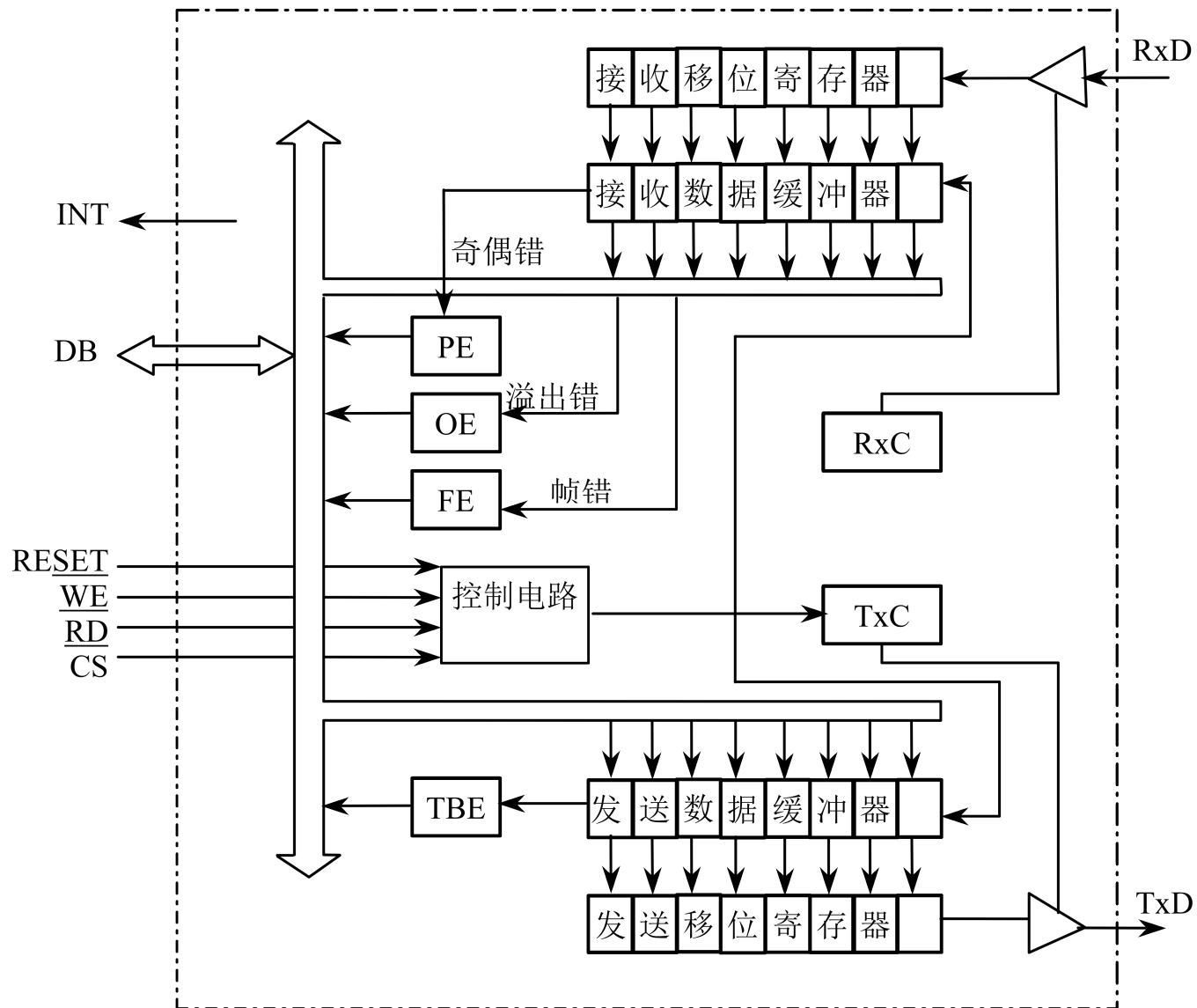
- 串行接口电路芯片的种类和型号很多。能够完成异步通信的硬件电路称为**UART**，即通用异步接收器/发送器 (**Universal Asynchronous Receive/Transmitter**)；能够完成同步通信的硬件电路称为**USRT** (**Universal Synchronous Receive/Transmitter**)；既能异步又能同步通信的硬件电路称为**USART** (**Universal Synchronous Asynchronous Receive/Transmitter**)。
- 从本质上说，所有的串行接口电路都是以并行数据形式与**CPU**接口、而以串行数据形式与外部逻辑接口传送数据。它们的基本功能都是从外部逻辑接收串行数据，转换成并行数据后传送给**CPU**，或从**CPU**接收并行数据，转换成串行数据后输出给外部逻辑。

8.1.3 串行通信接口电路

1. 异步通信硬件电路UART工作原理

- 串行发送时，**CPU**可以通过数据总线把8位并行数据送到“发送数据缓冲器”，然后并行送给“发送移位寄存器”，并在发送时钟和发送控制电路控制下通过**TxD**线一位一位地发送出去。起始位和停止位是由**UART**在发送时自动添加上去的。**UART**发送完一帧后产生中断请求，**CPU**响应后可以把下一个字符送到发送数据缓冲器，然后重复上述过程。
- 串行接收时，**UART**监视**RxD**线，并在检测到**RxD**线上有一个低电平(起始位)时就开始一个新的字符接收过程。**UART**每接收到一位二进制数据位后就使“接收移位寄存器”左移一次。连续接收到一个字符后并行传送到“接收数据缓冲器”，并通过中断促使**CPU**从中取走所接收的字符。

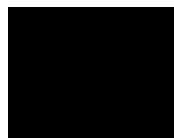
UART硬件框图



8.1.3 串行通信接口电路

2. UART对RxD线的采样

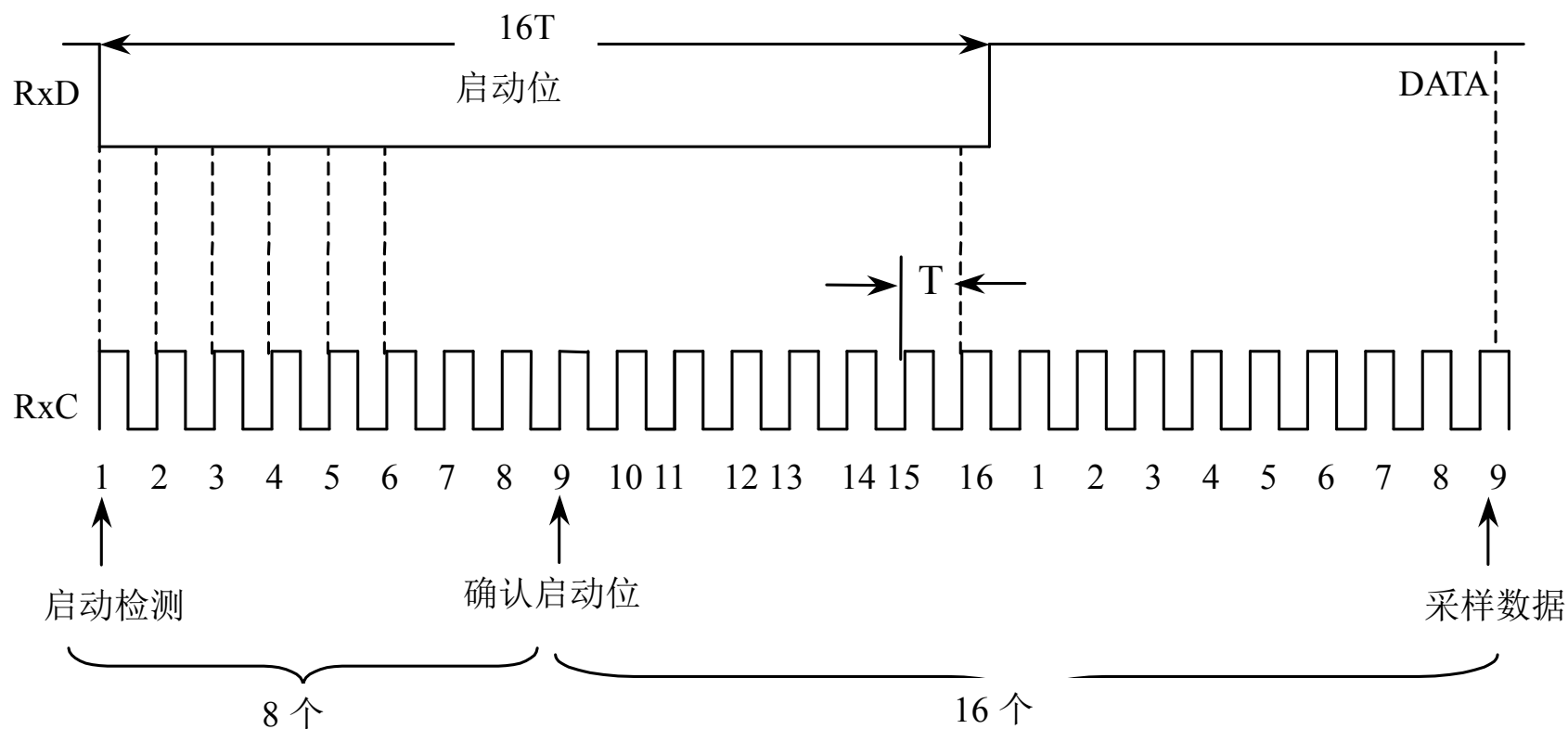
- UART对RxD线的采样是由接收时钟RxC完成的。其周期 T_c 和所传数据位的传输时间 T_d (位速率的倒数)必须满足如下关系:



式中, $K=16$ 或 64 。现以 $K=16$ 来说明UART对RxD线上字符帧的接收过程。

- 平常, UART按RxC脉冲上升沿采样RxD线。当连续采到RxD线上8个低电平(起始位二分之一位置)后, UART便确认对方在发送数据(不是干扰信号)。此后, UART便每隔16个RxC脉冲采样RxD线一次, 并把采到的数据作为输入数据, 以移位方式存入接收移位寄存器。

8.1.3 串行通信接口电路



UART对数据的采样

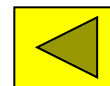
8.1.3 串行通信接口电路

3. 错误校验

- 数据在长距离传送过程中必然会发生各种错误，奇偶校验是一种最常用的校验数据传送错误的方法。
- 奇偶校验分奇校验和偶校验两种。UART的奇偶校验是通过发送端的奇偶校验位添加电路和接收端的奇偶校验检测电路实现的。

8.2 常用的串行通信总线

- 8.2.1 RS-232C总线
- 8.2.2 RS-449、RS-422A、RS-423A及RS-485总线
- 8.2.3 20mA电流环串行接口

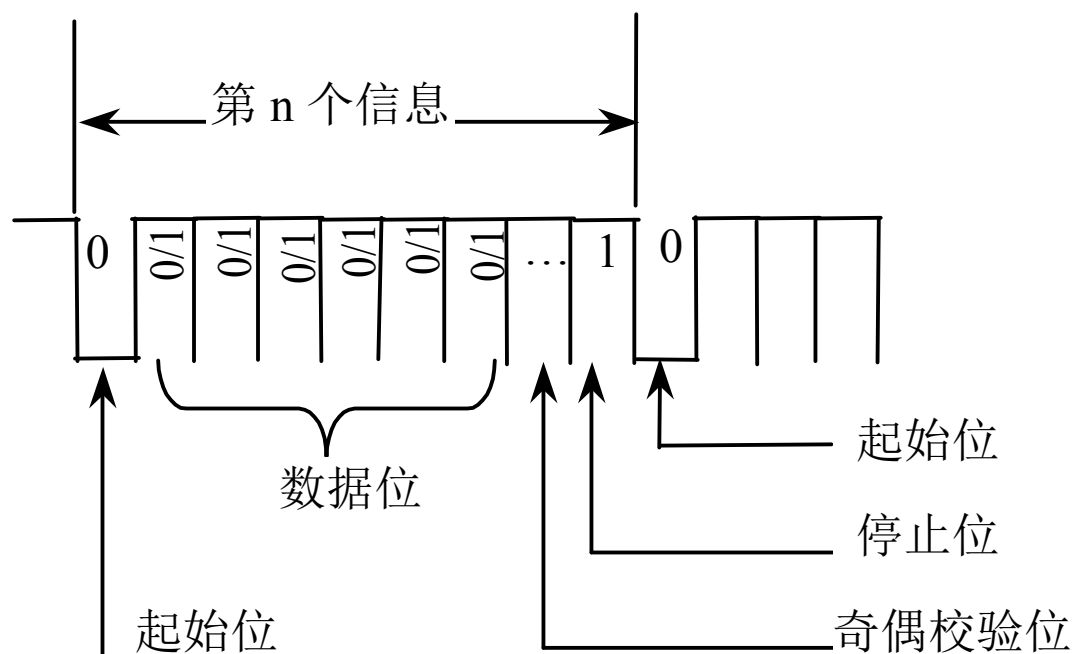


8. 2. 1 RS-232C总线

- **RS-232C**是使用最早、应用最多的一种异步串行通信总线标准，它主要用来定义计算机系统的一些数据终端设备(DTE)和数据电路终接设备(DCE)之间的接口的电气特性。
- **CRT**、打印机与**CPU**的通信大都采用**RS-232C**接口。由于**MCS-51**系列单片机本身有一个异步串行通信接口，因此，该系列单片机用**RS-232C**串行接口总线极为方便。
- **RS-232C**采用按位串行方式。**RS-232C**传递信息的格式标准对所传递的信息规定如下：信息的开始为起始位，信息的结尾为停止位，它可以是一位、一位半或两位；信息本身可以是**5、6、7、8**位再加一位奇偶校验位；如果两个信息之间无信息，则应写“**1**”，表示空。

8.2.1 RS-232C总线

RS-232C数据传输格式



8. 2. 1 RS-232C总线

- **RS-232C**传送的比特率(bit/s)规定为**19200、9600、4800、2400、600、300、150、110、75、50**。
- **RS-232C**接口总线的传送距离一般不超过**15m**。
- 由于**RS-232C**，它使用负逻辑，其低电平“0”在**+5~+15V**之间，高电平“1”在**-5~-15V**之间。因此，**RS-232C**不能和**TTL**电平直接相连，使用时必须加上适当的接口否则将使**TTL**电路烧毁。
- **RS-232C**的**25**条总线中，**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。

8.2.2 RS-449、RS-422A、RS-423A及RS-485总线

1. RS-449

- 1977年公布的电子工业标准接口RS-449，在很多方面可代替RS-232C应用。两者的主要差别是信号在导线上的传输方法不同。RS-232C是利用传输信号与公共地之间的电压差，RS-449接口是利用信号导线之间的信号电压差，可在1219.2m的24-AWG双绞线上进行数字通信，速率可达90000bit/s。RS-449规定了两种接口标准连接器，一种为37脚，一种为9脚。
- RS-449可以不使用调制解调器，它比RS-232C传输速率高，通信距离长；由于RS-449系统用平衡信号差传输高速信号，所以噪声低；它还可以多点或者使用公用线通信，故RS-449通信电缆可与多个设备并联。

8.2.2 RS-449、RS-422A、RS-423A及RS-485总线

2. RS-422A

- RS-422A标准给出了RS-449应用中对于电缆、驱动器和接收器的要求，规定了双端电气接口型式，该接口是双端线传送信号。它通过传输线驱动器，把逻辑电平变换成电位差，完成发送端的信息传递；通过传输线接收器，把电位差变换成逻辑电平，实现接收端的信息接收。RS-422A比RS-232C传输距离长、速度快，传输速率最大可达10Mbit/s，在此速率下电缆允许长度为12m。如果采用低速率传输，如90000bit/s，最大传输距离可达1200m。
- RS-422A每个通道要用两条信号线，如果其中一条是逻辑“1”状态，另一条就是逻辑“0”状态。RS-422A电路由发送器、平衡连接电缆、电缆终端负载、接收器等部分组成。在电路中规定只许有一个发送器，可有多个接收器，因此通常采用点对点的通信方式。该标准允许驱动器输出电压为 $\pm 2V \sim \pm 6V$ ，接收器可以检测到的输入信号电压可低到200mV。

8.2.2 RS-449、RS-422A、RS-423A及RS-485总线

3. RS-423A

- RS-422A和RS-423A标准分别给出了在RS-449应用中对电缆、驱动器和接收器的要求。RS-422A给出平衡信号差的规定，RS-423A给出了不平衡信号差的规定。
- RS-423A规定为单端线，而且与RS-232C兼容，参考电平为地，其正信号逻辑电平电压范围为200mV~6V，负信号逻辑电平电压范围为-200mV~-6V，RS-423A驱动器在90m长的电缆上传送数据的最大速率为100kbit/s，若降低到1000bit/s，则允许电缆长度为1200m。RS-423A允许在传送线上连接多个接收器，接收器为平衡传输接收器，因此允许驱动器和接收器之间有电位差。逻辑“1”状态必须超过4V，但不能高过6V；逻辑“0”状态必须低于-4V，但不能低于-6V。
- RS-423A也需要进行电平转换，常用的驱动器和接收器为3691和26L32。

8.2.2 RS-449、RS-422A、RS-423A及RS-485总线

4. RS-485

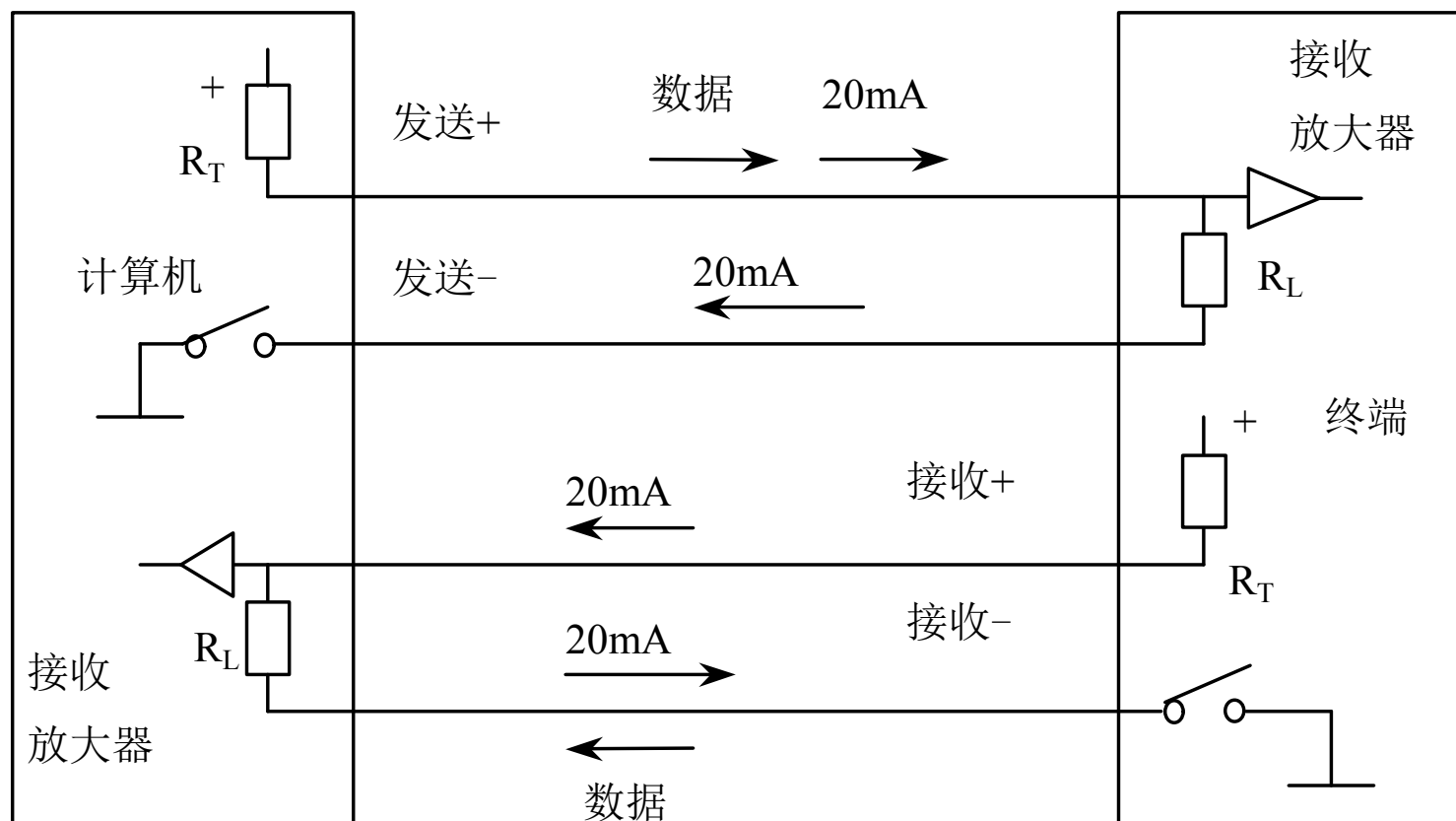
- RS-485是一种多发送器的电路标准，它扩展了RS-422A的性能，允许双导线上一个发送器驱动32个负载设备。负载设备可以是被动发送器、接收器和收发器。RS-485电路允许共用电话线通信。电路结构是在平衡连接电缆两端有终端电阻，在平衡电缆上挂发送器、接收器和组合收发器。RS-485标准没有规定在何时控制发送器发送或接收机接收数据。电缆选择比RS-422A更严格：以失真度(%)为纵轴，电缆的上升时间(t_r)或时间间隔单位(U. I)为横轴，给接收机不同信号电压 U_0 画出不同直线，根据直线选择电缆。
- RS-485最小型由两条信号电路线组成。每条连接电路必须有接地参考点，这电缆能支持32个发送接收器对。为了避免地面漏电流的影响，每个设备一定要接地。电缆应包括连至每个设备地的第三信号参考线。若用屏蔽电缆，屏蔽应接到设备的机壳。

8.2.3 20mA电流环串行接口

- 20mA电流环是目前串行通信中广泛使用的一种接口电路，其原理如下图所示。
- 由图可知，发送正、发送负、接收正、接收负四根线组成一个输入电流回路，一个输出电流回路。当发送数据时，根据数据的逻辑0、1使回路有规律地形成通、断状态(图中用开关示意)。
- 由于20mA电流环是一种异步串行接口标准，所以在每次发送数据时必须以无电流的起始状态作为每一个字符的起始位，接收端检测到起始位时便开始接收字符数据。
- 电流环串行通信接口的最大优点是低阻传输线对电气噪声不敏感，而且易实现光电隔离，因此在长距离通信时要比RS-232C优越得多。

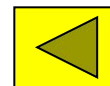
8.2.3 20mA电流环串行接口

20mA电流环原理图



8.3 MCS-51的串行接口

- 8.3.1 MCS-51串行接口结构
- 8.3.2 MCS-51串行口的工作方式
- 8.3.3 MCS-51串行通信的波特率

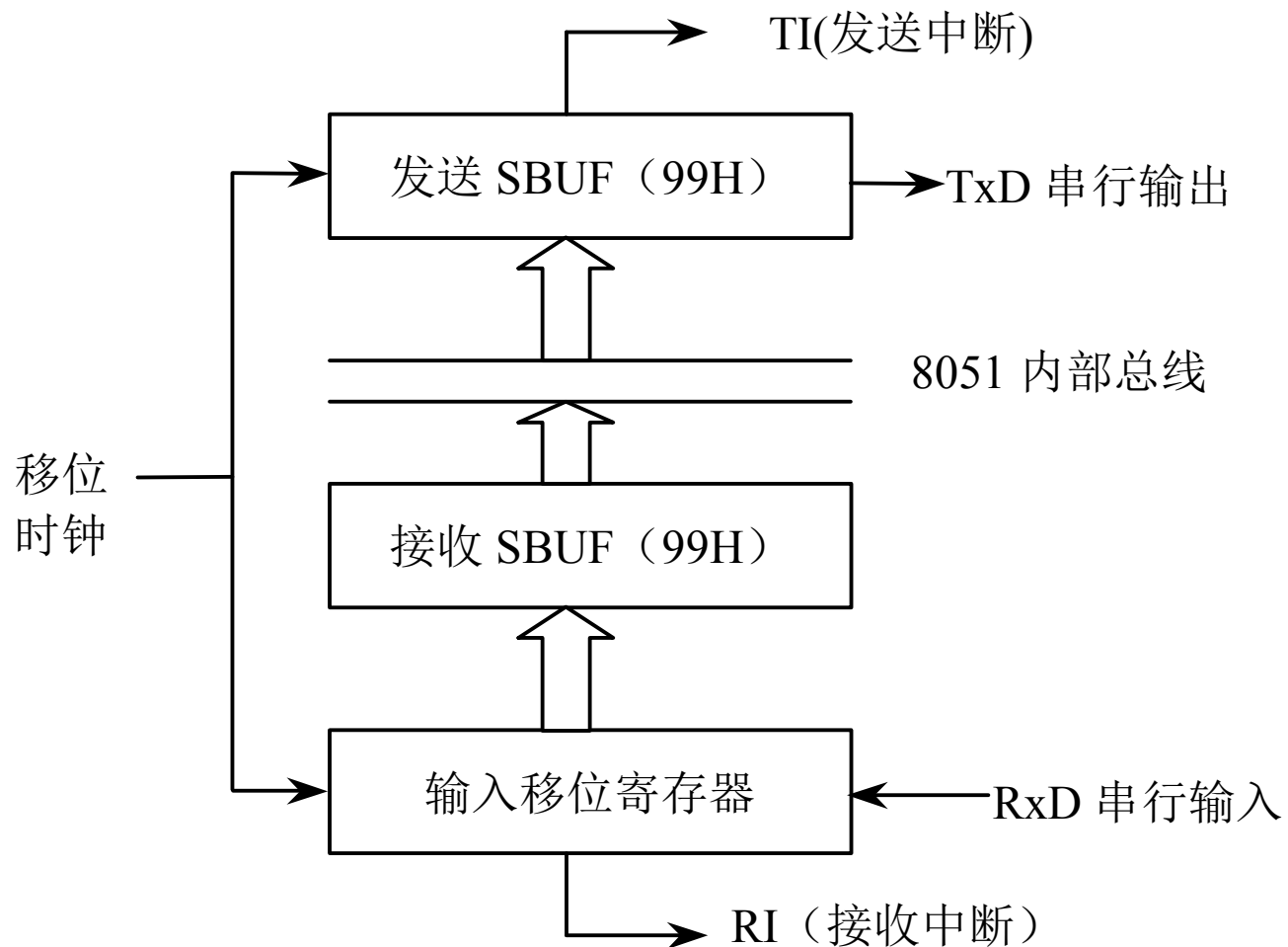


8.3.1 MCS-51串行接口结构

- MCS-51有两个独立的接收、发送缓冲器**SBUF** (属于特殊功能寄存器)，一个用作发送，一个用作接收。发送缓冲器只能写入不能读出；接收缓冲器只能读出不能写入，两者共用一个字节地址(99H)。串行接口结构如下图所示。
- 在发送时，**CPU**由一条写发送缓冲器的指令把数据(字符)写入串行口的发送缓冲器**SBUF** (发)中，然后从**TXD**端一位位地向外发送。与此同时，接收端**RXD**也可一位位地接收数据，直到收到一个完整的字符数据后通知**CPU**，再用一条指令把接收缓冲器**SBUF** (收)的内容读入累加器。可见，在整个串行收发过程中，**CPU**的操作时间很短，使得**CPU**还可以从事其他的各种操作(指工作在中断方式下)，从而大大提高了**CPU**的效率。

8.3.1 MCS-51 串行接口结构

串行接口结构示意图



8.3.1 MCS-51 串行接口结构

1. 串行接口数据缓冲器SBUF

- SBUF是两个在物理上独立的接收、发送缓冲器，可同时发送、接收数据。
- 两个缓冲器只用一个字节地址99H，可通过指令对SBUF的读写来区别是对接收缓冲器的操作还是对发送缓冲器的操作。
- CPU写SBUF，就是修改发送缓冲器；读SBUF，就是读接收缓冲器。
- 串行口对外也有两条独立的收发信号线RxD (P3. 0)、TxD (P3. 1)，因此可以同时发送、接收数据，实现全双工。

8.3.1 MCS-51 串行接口结构

2. 串行口控制寄存器SCON

SCON寄存器用来控制串行口的工作方式和状态，它可以是位寻址。在复位时所有位被清零，字地址为98H。

其格式如下：

位地址	9F	9E	9D	9C	9B	9A	99	98
位符号	SM ₀	SM ₁	SM ₂	REN	TB ₈	RB ₈	TI	RI

8.3.1 MCS-51 串行接口结构

2. 串行口控制寄存器SCON

各位定义如下：

- **SM0、SM1**：串行口工作方式选择位
其状态组合所对应的工作方式为：

SM0	SM1	工作方式	功能说明	波特率
0	0	0	8位同步移位寄存器	$f_{osc}/12$
0	1	1	8位异步收发	由定时器控制
1	0	2	9位异步收发	$f_{osc}/32$ 或 $f_{osc}/64$
1	1	3	9位异步收发	由定时器控制

8.3.1 MCS-51 串行接口结构

2. 串行口控制寄存器SCON

- **SM2：多机通信控制位**

因为多机通信是在方式2和方式3下进行的，因此SM2主要用于方式2和方式3。当串行口以方式2和方式3接收数据时，

- **SM2=1**，则只有在接收到的第9位数据(RB8)为1时才将接收到的前8位数据送入SBUF，并置位RI产生中断请求；否则将接收到的前8位数据丢弃。
- **SM2=0**，则不论第9位数据是0还是1，都将前8位数据装入SBUF中，并产生中断请求。在方式0时，SM2必须为0。

8.3.1 MCS-51 串行接口结构

2. 串行口控制寄存器SCON

- REN: 允许接收控制位
 - REN=0时禁止串行口接收。
 - REN=1时允许串行口接收。
- 该位由软件置位或复位。

8.3.1 MCS-51 串行接口结构

2. 串行口控制寄存器SCON

- **TB8：发送数据位**
 - 在方式2或方式3时，TB8是发送数据的第9位，根据发送数据的需要由软件置位或复位。
 - 可作为奇偶校验位(单机通信)。
 - 可在多机通信中作为发送地址帧或数据帧的标志位。多机通信时，一般约定：发送地址帧时，设置TB8=1；发送数据帧时，设置TB8=0。在方式0和方式1中，该位未用。

8.3.1 MCS-51 串行接口结构

2. 串行口控制寄存器SCON

- **RB8：接收数据位**
 - 在方式2和方式3时，存放接收数据的第9位。
 - 可以是约定的奇偶校验位。
 - 可以是约定的地址/数据标志位，可根据RB8被置位的情况对接收到的数据进行某种判断。在多机通信时，若RB8=1，说明收到的数据为地址帧；RB8=0，说明收到的数据为数据帧。在方式1下，若SM2=0，则RB8用于存放接收到的停止位方式；方式0下，该位未用。

8.3.1 MCS-51 串行接口结构

2. 串行口控制寄存器SCON

- TI：发送中断标志位
用于指示一帧数据发送完否。
 - 在方式0下，发送电路发送完第8位数据时，TI由硬件置位。
 - 在其他方式下，TI在发送电路开始发送停止位时置位，这就是说：TI在发送前必须由软件复位，发送完一帧后由硬件置位。因此，CPU查询TI状态便可知一帧信息是否已发送完毕。

8.3.1 MCS-51 串行接口结构

2. 串行口控制寄存器SCON

- **RI：接收中断标志位**
用于指示一帧信息是否接收完。
 - 在方式1下，RI在接收电路接收到第8位数据时由硬件置位。
 - 在其他方式下，RI是在接收电路接收到停止位的中间位置时置位的，RI也可供CPU查询，以决定CPU是否需要从“SBUF(接收)”中提取接收到的字符或数据。RI也由软件复位。

8.3.1 MCS-51 串行接口结构

2. 串行口控制寄存器SCON

- 在进行串行通信时，一帧发送完后，必须用软件来设置SCON的内容。当由指令改变SCON的内容时，改变的内容在下一条指令的第一个周期的S1P1状态期间才锁存到SCON寄存器中，并开始有效。如果此时已开始进行串行发送，那么TB8送出去的仍是原有的值而不是新值。
- 在进行串行通信时，一帧发送完毕后，发送中断标志置位，向CPU请求中断；当一帧接收完毕时，接收中断标志置位，也向CPU请求中断。若CPU允许中断，则要进入中断服务程序。CPU事先并不能区分是RI请求中断还是TI请求中断，只能在进入中断服务程序后，通过查询来区分，然后进入相应的中断处理。

8.3.1 MCS-51 串行接口结构

3. 电源控制寄存器PCON

PCON寄存器主要是为CHMOS型单片机的电源控制设置的专用寄存器，单元地址为87H，不能位寻址。

其格式如下：

位序	D7	D6	D5	D4	D3	D2	D1	D0
位符号	SMOD	/	/	/	GF ₁	GF ₀	PD	IDL

8.3.1 MCS-51串行接口结构

3. 电源控制寄存器PCON

各位定义如下：

- **SMOD**: 串行口波特率的倍增位。
在HMOS单片机中，该寄存器中除最高位之外，其他位都是虚设的。在单片机工作在方式1、方式2和方式3时，
 - **SMOD=1**，串行口波特率提高一倍。
 - **SMOD=0**，则波特率不加倍。系统复位时**SMOD=0**。
- **GF1、GF0**: 通用标志位，由软件置位、复位。
- **PD**: 掉电方式控制位，**PD=1**，则进入掉电方式。
- **IDL**: 待机方式控制位，**IDL=1**，则进入待机方式。

8.3.2 MCS-51串行口的工作方式

1. 方式0

串行接口工作方式0为同步移位寄存器方式，其波特率是固定的，为 f_{osc} (振荡频率)的1/12。

(1) 方式0发送

数据从TxD引脚串行输出，RxD引脚输出同步脉冲。当一个数据写入串行口发送缓冲器时，串行口将8位数据以 $f_{osc}/12$ 的固定波特率从TxD引脚输出，从低位到高位。发送后置中断标志TI为1，请求中断，在再次发送数据之前，必须用软件将TI清零。

8.3.2 MCS-51串行口的工作方式

1. 方式0

(2) 方式0接收

在满足REN=1和RI=0的条件下，串行口处于方式0输入。此时，RxD为数据输入端，TXD为同步信号输出端，接收器也以 $f_{osc}/12$ 的波特率对RxD引脚输入的数据信息采样。当接收器接收完8位数据后，置中断标志RI=1为请求中断，在再次接收之前，必须用软件将RI清零。

在方式0工作时，必须使SCON寄存器中的SM2位为“0”，这并不影响TB8位和RB8位。方式0发送或接收完数据后由硬件置位TI或RI，CPU在响应中断后要用软件清除TI或RI标志。

8.3.2 MCS-51串行口的工作方式

【例1】 利用两片74LS165扩展2个8位并行输入端口。

解：扩展电路如下图所示。

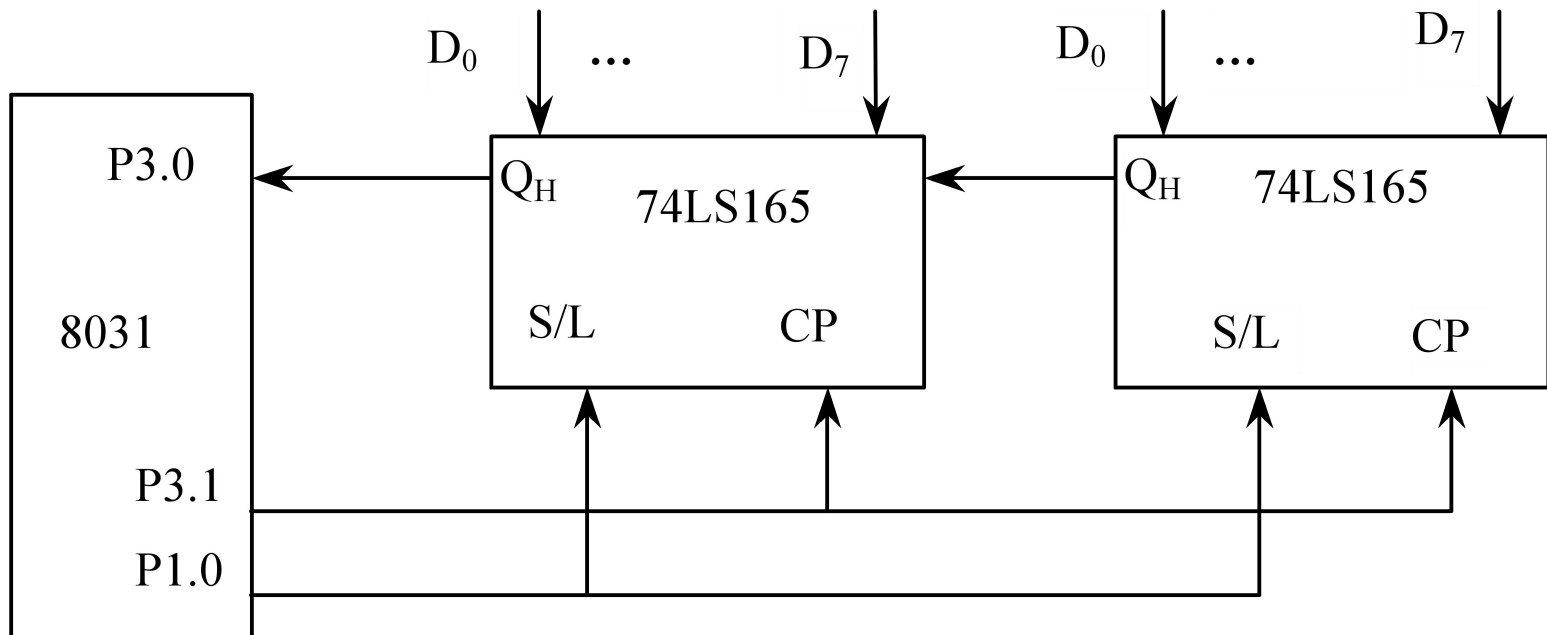
74LS165是8位并行输入、串行输出移位寄存器，RxD为串行输入引脚，与74LS165的串行输出端相连；TxD为移位脉冲输出端，与所有的74LS165芯片移位脉冲输入端相连；用1根I/O线来控制移位与置位。

以两个8位并行口读入20H组字节数据，并把它们转存到内部RAM数据区(首址为30H)的程序清单如下。

程序中用户标志位F0用来标志一组数的前8位与后8位。

8.3.2 MCS-51 串行口的工作方式

利用串行口扩展并行输入口电路



8.3.2 MCS-51串行口的工作方式

参考程序：

	MOV R7, #20H	;设置字节组数
	MOV R0, #30H	;设置内部RAM数据区首址
	SETB F0	;设置读入字节奇偶数标志,第1个8位数为偶数
RCV0:	CLR P1.0	;74LS165置入数据
	SETB P1.0	;允许74LS165串行移位
RCV1:	MOV SCON, #10	;串行口设为方式0
STP:	JNB RI, STP	;等待接收完
	CLR RI	
	MOV A, SBUF	;从串口读入数据
	MOV @R0, A	
	INC R0	;指向数据区下个地址
	CPL F0	;指向第奇数个8位数
	JB F0, RCV2	;读入第偶数个8位数后继续读第奇数个8位数,如读完第奇数个8位数转RCV2
	DEC R7	;读完一组数
	SJMP RCV1	;再读入第奇数个8位数
RCV2:	DJNZ R7, RCV0	;20组数未读完,重新并行置入
		

8.3.2 MCS-51串行口的工作方式

2. 方式1

工作方式1时，串行口被设置为波特率可变的8位异步通信接口。

(1) 方式1发送

串行口以方式1发送数据时，数据位由TxD端输出，1帧信息包含10位，即一位起始位、八位数据位(先低位后高位)和一个停止位“1”。CPU执行一条数据写入发送缓冲器SBUF的指令，就启动发送器发送数据：发送完数据后，就置中断标志TI为1。方式1所传送的波特率取决于定时器T的溢出率和特殊功能寄存器PCON中SMOD的值。

8.3.2 MCS-51串行口的工作方式

(2) 方式1接收

当串行口工作在方式1下，且REN=1时，串行口处于方式1的输入状态。它以所选波特率的16倍的速率对RxD引脚状态采样。当采样到由1到0的负跳变时，启动接收器，接收的值是3次采样中至少两次相同的值，以保证可靠无误。当检测到起始位有效时，开始接收一帧的其余信息。一帧信息包含10位，即一位起始位、八位数据位(先低位后高位)和一个停止位“1”。使用方式1接收时，必须同时满足以下两个条件：即RI=0和停止位为1或SM2=0，若满足条件则接收数据进入SBUF，停止位进入RB8，并置中断请求标志RI为1。若上述两个条件不满足，则接收数据丢失，不再恢复。这时将重新检测RxD上1到0的负跳变，以接收下一帧数据。中断标志也必须由用户在中断服务程序中清零。

8.3.2 MCS-51串行口的工作方式

【例2】 由内部RAM单元20H~3FH取出ASCII码数据，在最高位上加奇偶校验位后由串行口输出，采用8位异步通信，波特率为1200bit/s， $f_{osc}=11.059\text{MHz}$ 。

解：由题意可知，应把串行口置为方式1；采用定时器T1，以方式2工作，作波特率发生器，预置值(TH1)=0E8H。

主程序

```
MOV    TMOD, #20H      ;设T1为模式2
MOV    TL1, #0E8H      ;装入时间常数
MOV    TH1, #0E8H

SETB   TR1              ;启动定时器T1

MOV    SCON, #40H      ;设串行口为方式1

MOV    R0, #20H        ;发送数据首地址

MOV    R7, #32          ;发送首个数

LOOP:  MOV    A, @R0     ;发送数据送累加器A

       ACALL  SPOUT      ;调发送子程序

       INC    R0         ;指向下一个地址

       DJNZ   R7, LOOP
```

串行口发送子程序

```
SPOUT:  MOV    C, P      ;设置奇校验位

        CPL    C

        MOV    ACC.7, C

        MOV    SBUF, A   ;启动串行口发送

        JNB    TI, $     ;等待发送完

        CLR    TI        ;清TI标志,允许再发送

        RET
```

8.3.2 MCS-51串行口的工作方式

8.3.2 MCS-51串行口的工作方式

3. 方式2

串行口工作在方式2下时，被定义为9位异步通信接口。

(1) 方式2发送

发送数据由TxD端输出，一帧信息包含11位，即一位起始位(0)、八位数据位(先低位后高位)、一位附加可控位(1或0)和一位停止位“1”。附加的第9位数据为SCON中的TB8，它由软件置位或清零，可作为多机通信中地址 / 数据信息的标志位，也可作为数据的奇偶校验位。

以TB8作为奇偶校验位，处理方法为数据写入SBUF之前，先将数据的奇偶位写入TB8。CPU执行一条写SBUF的命令后，便立即启动发送器发送，送完一帧信息后，TI被置1，再次向CPU申请中断。因此在进入中断服务程序后，在发送完一帧数据之前，必须将TI清零。发送中断服务参考程序如下。

8.3.2 MCS-51串行口的工作方式

发送中断服务参考程序：

```
PIPL:    PUSH    PSW                ;保护现场
          PUSH    A
          CLR     TI                ;清零发送中断标志
          MOV     A, @R0             ;取数据
          MOV     C, P              ;奇偶位送C
          MOV     TB8, C            ;奇偶位送TB8
          MOV     SBUF, A           ;发送数据
          INC     R0                ;数据指针加1
          POP     A                  ;恢复现场
          POP     PSW
          RETI
```

8.3.2 MCS-51串行口的工作方式

3. 方式2

(2) 方式2接收

当串行口置为方式2，且REN=1时，串行口以方式2接收数据。方式2的接收与方式1基本相似。数据由RxD端输入，接收11位信息，其中，一位起始位(0)、八位数据位(先低位后高位)、一位附加可控位(1或0)和一位停止位“1”。当采样到RxD端由1到0的负跳变，并判断起始位有效后，便开始接收一帧信息，当接收器接收到第9位数据后，当RI=0且SM2=0或接收到的第9位数据为1时，将收到的数据送入SBUF(接收数据缓冲器)，第9位数据送入RB8，并将RI置1；若以上两个条件不满足，接收信息丢失。

若附加的第9位为奇偶校验位，在接收中断服务程序中应作检验。接收中断服务程序参考程序如下。

8.3.2 MCS-51串行口的工作方式

接收中断服务参考程序：

```
PIPL:      PUSH   PSW                      ;保护现场
           PUSH   A
           CLR    RI                      ;清零接收中断标志
           MOV    A, SBUF                  ;接收数据
           MOV    C, P                    ;取奇偶校验位
           JNC    L1                      ;偶校验时转L1
           JNB    RB8, ERR                ;奇校验时RB8为0转出错处理
           SJMP   L2
L1:         JB    RB8, ERR                 ;偶校验时RB8为1转出错处理
L2:         MOV    @R0, A                  ;奇偶校验对时存入数据
           INC    R0                      ;修改指针
           POP    A                      ;恢复现场
           POP    PSW
           RETI                           ;中断返回
ERR:        .....                       ;出错处理子程序(省略)
           RETI                           ;中断返回
```

8.3.2 MCS-51串行口的工作方式

4. 方式3

方式3为波特率可变的9位异步通信方式，除了波特率有所区别之外，其余都与方式2相同。

8.3.3 MCS-51串行通信的波特率

- 串行口的通信波特率恰到好处地反映了串行传输数据的速率。通信波特率的选用，不仅和所选通信设备、传输距离有关，还受传输线状况所制约。用户应根据实际需要加以正确选用。

1. 方式0下的波特率

在方式0下，串行口通信的波特率是固定的，其值为 $f_{osc}/12$ (f_{osc} 为主机频率)。

8.3.3 MCS-51 串行通信的波特率

2. 方式2下的波特率

在方式2下，通信波特率为 $f_{\text{osc}}/32$ 或 $f_{\text{osc}}/64$ 。用户可以根据PCON中SMOD位的状态来驱使串行口在哪个波特率下工作。选定公式为：

$$\text{波特率} = \frac{2^{\text{SMOD}}}{64} f_{\text{osc}}$$

这就是说，若SMOD=0，则所选波特率为 $f_{\text{osc}}/64$ ；若SMOD=1，则波特率为 $f_{\text{osc}}/32$ 。

8.3.3 MCS-51 串行通信的波特率

3. 方式1或方式3下的波特率

在这两种方式下，串行口波特率是由定时器的溢出率决定的，因而波特率也是可变的。相应公式为：

$$\text{波特率} = \frac{2^{\text{SMOD}}}{32} \cdot \text{定时器 T1 溢出率}$$

定时器T1溢出率的计算公式为：

$$\text{定时器 T1 溢出率} = \frac{f_{\text{osc}}}{12} \left(\frac{1}{2^K - \text{初值}} \right)$$

式中， K 为定时器T1的位数，它和定时器T1的设定方式有关。若定时器T1为方式0，则 $K=13$ ；若定时器T1为方式1，则 $K=16$ ；若定时器T1为方式2或3，则 $K=8$ 。

8.3.3 MCS-51 串行通信的波特率

3. 方式1或方式3下的波特率

注意：

- 其实，定时器T1通常采用方式2，因为定时器T1在方式2下工作时TH1和TL1分别设定为两个8位重装计数器(当TL1从全“1”变为全“0”时，TH1重装TL1)。使用这种方式，不仅可使操作方便，也可避免因重装初值(时间常数初值)而带来的定时误差。
- 由上两式可知，方式1或方式3下所选波特率常常需要通过计算来确定初值，因为该初值是要在定时器T1初值化时使用的。

波特率= $2^{\text{smod}} / 32 \times$ 定时器T1溢出率

T1溢出率= T1计数率/产生溢出所需的周期数

式中T1计数率取决于它工作在定时器状态还是计数器状态。当工作于定时器状态时,T1计数率为 $f_{\text{osc}}/12$;当工作于计数器状态时,T1计数率为外部输入频率,此频率应小于 $f_{\text{osc}}/24$ 。产生溢出所需周期与定时器T1的工作方式、T1的预置值有关。

定时器T1工作于方式0: 溢出所需周期数= $8192-x$

定时器T1工作于方式1: 溢出所需周期数= $65536-x$

定时器T1工作于方式2: 溢出所需周期数= $256-x$

因为方式2为自动重装入初值的8位定时器/计数器模式,所以用它来做波特率发生器最恰当。

当时钟频率选用11.0592MHZ时,取易获得标准的波特率,所以很多单片机系统选用这个看起来“怪”的晶振就是这个道理。

下表列出了定时器T1工作于方式2常用波特率及初值。

常用波特率	Fosc(MHZ)	SMOD	TH1初值
19200	11.0592	1	FDH
9600	11.0592	0	FDH
4800	11.0592	0	FAH
2400	11.0592	0	F4h
1200	11.0592	0	E8h

例如9600 11.0592 0 FDH
波特率= 定时器T1溢出率

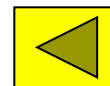
$T1\text{溢出率} = T1\text{计数率} / \text{产生溢出所需的周期数}$

$\text{产生溢出所需的周期数} = 256 - FD(253) = 3$

$SMOD = 0 \quad 11059200 / 12 * 3 * 1/32 = 9600$

8.4 MCS-51单片机之间的通信

- 8.4.1 MCS-51双机通信技术
- 8.4.2 MCS-51多机通信技术

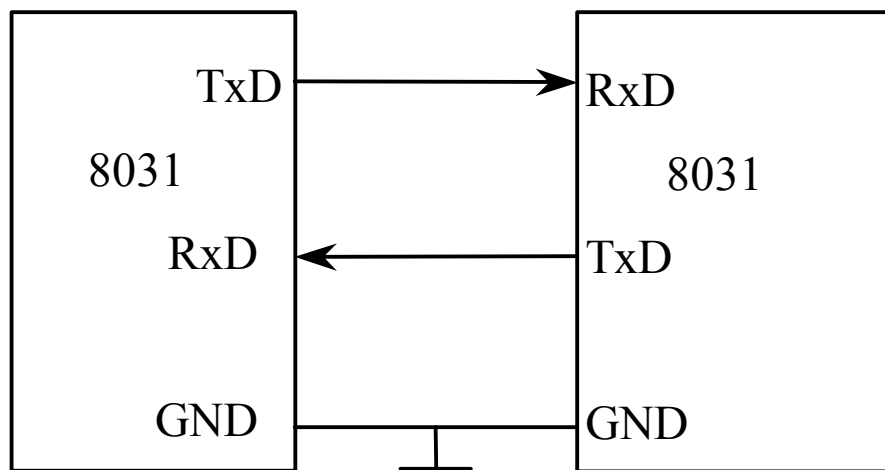


8. 4. 1 MCS-51双机通信技术

- 如果两个8031应用系统相距很近，将它们的串行口直接相连，即可实现双机通信，
- 为了增加通信距离，减少通道及电源干扰，可以在通信线路上采取光电隔离的方法，利用RS-422标准进行双机通信。
- 发送方的数据由串行口TxD端输出，通过74LS05反向驱动，经光电耦合器至四差分驱动器75174的输入端，75174将输入的TTL信号变换成符合RS-422标准的差动信号输出，经传输线(双绞线)将信号传送到接收端。接收方通过三态四差分接收器75175将差分信号转换成TTL电平信号，通过反向驱动后，经光电耦合器到达接收方串行口的接收端。

8.4.1 MCS-51双机通信技术

双机异步通信接口电路



8.4.1 MCS-51双机通信技术

【例3】利用查询方式进行双机通信。

解：甲机发送，乙机接收，双方都用查询方式进行通信的程序编制。两机应用相同的工作方式和波特率。

(1) 甲机发送

编程把甲机片内RAM中70H~9FH单元中的数据从串行口输出。定义以工作方式2发送，TB8作奇偶校验位。其中 $f_{\text{osc}} = 6\text{MHz}$ ，波特率为187.5kbit/s，所以SMOD=1。

(2) 乙机接收

编程使乙机接收到的32字节数据并存入片外2000H~201FH单元中。接收过程要求判断奇偶校验位RB8。若出错，置F0标志为1；正确，则置F0标志为0，然后返回。

$$\text{波特率} = \frac{2^{\text{SMOD}}}{32}$$

8.4.1 MCS-51双机通信技术

发送端参考程序

```
SEN:      MOV    SCON, #80H           ;设置串行口为方式2
          MOV    PCON, #80H           ;SMOD=1
          MOV    R1, #70H             ;设数据块指针
          MOV    R7, #20H             ;设数据块长度
TRS:      MOV    A, @R1               ;取数据
          MOV    C, P
          MOV    TB8, C               ;奇偶位送TB8
          MOV    SBUF, A              ;启动发送
WAIT:     JBC    TI, CONT              ;判断一帧是否发完
          AJMP   WAIT
CONT:     INC    R1                   ;更新数据单元
          DJNZ   R7, TRS              ;循环发送至结束
          RET
```



```
RECE:      MOV    SCON, #80H           ;设置串行口为方式2
           MOV    PCON, #80H           ;SMOD=1
           MOV    DPTR, #2000H         ;设置数据块指针
           MOV    R6, #20H             ;设置数据块长度
           SETB   REN                  ;允许接收

WAIT:      JBC    RI, READ              ;判断一帧是否接收完
           AJMP   WAIT

READ:      MOV    A, SBUF               ;读一帧数据
           JNB    PSW.0, PZ             ;奇偶位P为0,则转
           JNB    RB8, ERR              ;P=1, RB8=0, 则出错
           SJMP   RIGHT                 ;两者全为1, 则正确

PZ:        JB     RB8, ERR              ;P=0, RB8=1, 则出错

RIGHT:     MOVX   @DPTR, A              ;正确, 存放数据
           INC    DPTR                  ;修改指针
           DJNZ   R6, WAIT              ;判断数据块接收完否
           CLR    PSW.5                 ;接收正确, 且接收完, 清F0标志
           RET                           ;返回

ERR:       SETB   PSW.5                 ;出错, 置F0标志为1
           RET                           ;返回
```

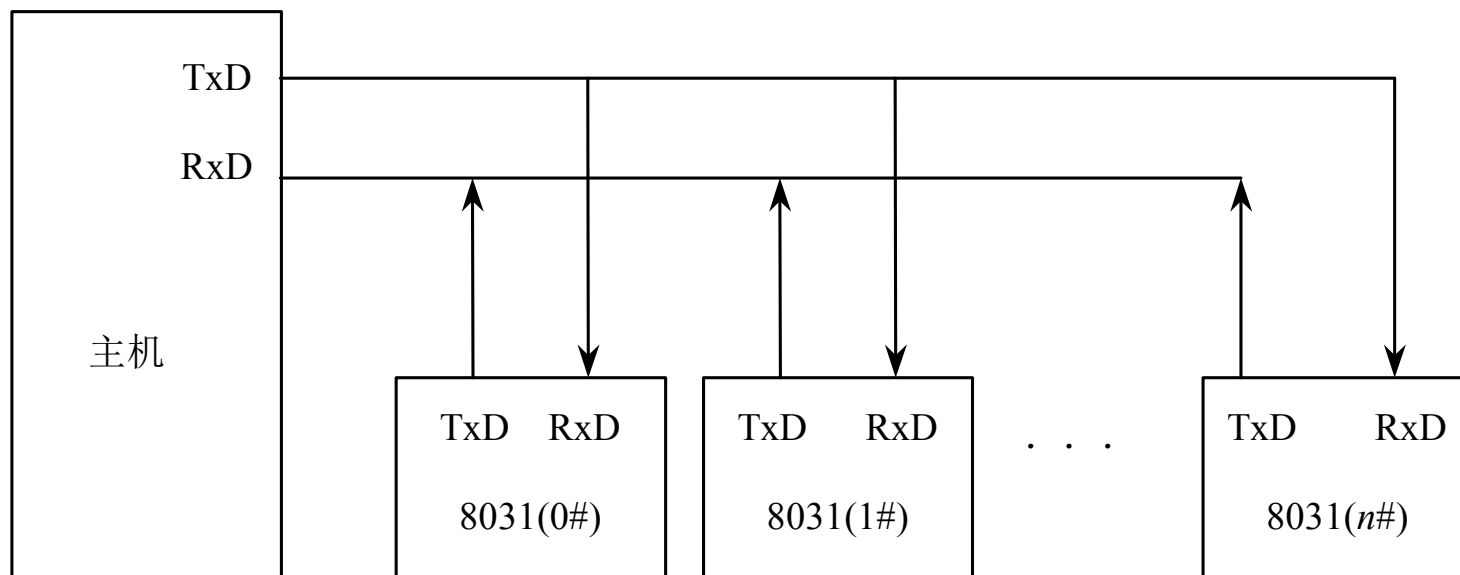
8. 4. 1 MCS-51双机通信技术

【例4】 利用中断方式进行双机通信。

程序请参见教材

8.4.2 MCS-51多机通信技术

通常**MCS-51**的多机通信采用主从式多机通信方式。在这种方式中，只有一台主机，有多台从机。主机发送的信息可以传到各个从机或指定的从机，各从机发送的信息只能被主机接收。其连接电路如下图所示。



8.4.2 MCS-51多机通信技术

1. 多机通信原理

- 要保证主机与所选择的从机实现可靠的通信，必须保证通信接口具有识别功能。MCS-51串行控制寄存器中的SM2就是为了满足这一要求而设置的多机控制位。多机控制原理如下：
 - 若SM2=1（在串行口以方式2或方式3接收时），表示置多机通信功能位，这时出现两种可能的情况：
 - 接收到的第9位数据为1，数据装入SBUF，并置RI=1，向CPU发出中断请求。
 - 接收到的第9位数据为0，不产生中断，信息将被丢失。
 - 若SM2=0，则接收到的第9位信息无论是0还是1，都产生RI=1的中断标志，接收到的数据装入SBUF。根据这个功能，便可实现多个MCS-51系统的串行通信。

8.4.2 MCS-51多机通信技术

1. 多机通信原理

多机通信的过程如下：

- (1) 使所有从机的**SM2=1**，处于准备接收一帧地址数据的状态。
- (2) 主机设第**9**位数据为**1**，发送一帧地址信息，与所需的从机进行联络。
- (3) 从机接收到地址信息后，各自将其与自己的地址相比较，对于地址相符的从机，使**SM2=0**，以接收主机随后发来的所有信息；对于地址不相符的从机，仍保持**SM2=1**状态，对主机随后发来的数据不理睬，直至发送新的一帧地址信息。
- (4) 主机发送控制指令与数据给被寻址的从机。一帧信息的第**9**位置零，表示发送的是数据或控制指令。

8.4.2 MCS-51多机通信技术

2. 多机通信实现方法

(1) 有关通信协议的几条规定

- ① 从机地址为**00H**~**FEH**，即允许接**255**台从机。
- ② **FFH**为一条控制指令，使所有从机都处于**SM2=1**的状态。
- ③ 其余控制指令有：**00H**—接收指令；**01H**—发送指令。这两条指令是作为数据发送的。
- ④ 从机状态字格式如下：

D7	D6	D5	D4	D3	D2	D1	D0
ERR	0	0	0	0	0	TRDY	RRDY

ERR=1，表示收到非法指令；

TRDY=1，表示发送准备完毕；

RRDY=1，表示接收准备完毕。

8.4.2 MCS-51多机通信技术

2. 多机通信实现方法

(2) 软件设计

主机和从机程序都以子程序形式出现，所不同的是在这里设从机采用中断的方式。

- 在调用主机子程序之前，应先准备好**R0**、**R1**、**R2**、**R3**和**R4**中的参数。
- 主机中寄存器的分配：
 - **R0**: 主机接收数据块首址;
 - **R1**: 主机发送数据块首址;
 - **R2**: 被寻址的从机地址;
 - **R3**: 主机发出的指令;
 - **R4**: 数据块长度。
- 主机串行口设为方式**3**，允许接收，并置**TB8**为**1**，故控制字为**11011000B=D8H**。

8. 4. 2 MCS-51多机通信技术

2. 多机通信实现方法

(2) 软件设计

- 从机采用中断的方式，即接收到地址帧后就进行串行口中断申请。CPU响应后，进入中断服务程序。
- 从机串行口的初始化、定时器初始化和中断初始化等都不包括在内，这些内容应在从机主程序中完成。
- 从机串行口中断服务参考程序中SLAVE为本机地址，并用F0和PSW.1作为本从机发送和接收准备就绪的状态位。
- 从机串行口中断服务参考程序中R0、R1分别为从机发送和接收数据区地址指针，R2、R3分别为发送和接收数据块长度。

	MOV SCON, #0D8H	; 串行口初始化
MSIO1:	MOV A, R2	
	MOV SBUF, A	; 启动发送, 发送从机地址
	JNB RI, \$	; 等待从机回答
	CLR RI	; 从机回答后清RI
	MOV A, SBUF	; 取从机地址
	XRL A, R2	; 核对地址
	JZ MSIO3	; 相符转MSIO3
MSIO2:	MOV SBUF, #0FFH	; 准备重发地址
	SETB TB8	; 设地址帧标志
	SJMP MSIO1	; 重发地址
MSIO3:	CLR TB8	; 准备发送指令
	MOV SBUF, R3	; 发出指令
	JNB RI, \$	; 等待从机回答
	CLR RI	; 从机回答后清RI
	MOV A, SBUF	; 取应答信息
	JNB ACC.7, MSIO4	; 核对指令接收是否出错
	SJMP MSIO2	; 指令出错, 重发

8.4.2 MCS-51多机通信技术

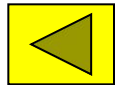
MSIO4:	CJNE R3,#00H, MSIO5	;若为发送指令,则转发送程序段
	JNB ACC.0, MSIO2	;从机未准备好接收,重新联络
STX:	MOV SBUF, @R1	;从机准备好,开始发送
	JNB TI, \$	;等待发送结束
	CLR TI	;清TI,准备发送下一帧数据
	INC R1	;修改指针
	DJNZ R4, STX	;未完,继续
	RET	;发送完,返回
MSIO5:	JNB ACC.1, MSIO2	;从机未准备好发送,重新联络
SRX:	JNB RI, \$	;等待接收完毕
	CLR RI	;清RI,为下次接收准备
	MOV A, SBUF	;取出收到的数据
	MOV @R0, A	;存入数据
	INC R0	;修改指针
	DJNZ R4, SRX	;未完,继续
	RET	;接收完毕,返回

SSIO:	CLR RI	;清RI标志
	PUSH A	;保护现场
	PUSH PSW	
	SETB RS1	;取1区工作寄存器
	CLR RS0	
	MOV A, SBUF	
	XRL A, #SLAVE	;核对是否收到本机地址
	JZ SSIO1	;地址相符转SSIO
RETURN:	POP PSW	;不是呼叫本机,恢复现场,返回
	POP A	
	RETI	
SSIO1:	CLR SM2	;准备接收数据/指令
	MOV SBUF, #SLAVE	;向主机发回本机地址
	JNB RB8, SSIO2	;RB8=0,是数据/指令,转SSIO2
	SETB SM2	;是复位指令,返回
	SJMP RETURN	
SSIO2:	MOV A, SBUF	;取指令
	CJNE A, #02H, NEXT	;检查指令是否合法
Next:	JC SSIO3	;是合法指令,转SSIO3
	CLR TI	
	MOV SBUF, #80H	;是非法指令,发出出错信息,返回
	SJMP RETURN	

SSIO3:	JZ CMOD	;是接收指令,转接收模块
CMD1:	JB F0,SSIO4	;发送准备好,转SSIO4
	MOV SBUF,#00H	;未准备好,发TRDY=0,返回
	SJMP RETURN	
SSIO4:	MOV SBUF,#02H	;发TRDY=1,准备发送
	CLR TI	
LOOP1:	MOV SBUF,@R0	;开始循环发送,直至发送完毕
	JNB TI,\$	
	CLR TI	
	INC R0	
	DJNZ R2,LOOP1	
	SETB SM2	
	SJMP RETURN	
CMOD:	JB PSW.1,SSIO5	;接收准备好,转SSIO5
	MOV SBUF,#00H	;未准备好,发RRDY=0,返回
	SJMP RETURN	
SSIO5:	MOV SBUF,#01H	;发RRDY=1,准备接收
LOOP2:	JNB R1,\$	
	CLR RI	
	MOV @R1,SBUF	;开始循环接收,直至接收完毕
	INC R1	
	DJNZ R3,LOOP2	
	SJMP RETURN	

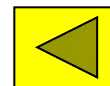
本章小结

- 计算机之间的通信有并行通信和串行通信两种方式。异步串行通信接口主要有RS-232C、RS-449及20mA电流环等几种标准。
- MCS-51系列单片机内部有一个全双工的异步串行通信I/O口，该串行口的波特率和帧格式可以编程设定。MCS-51串行口有四种工作方式：方式0、1、2、3。帧格式有10位、11位。方式0和方式2的传送波特率是固定的，方式1和方式3的传送波特率是可变的，由定时器的溢出率决定。
- 单片机与单片机之间以及单片机与PC机之间都可以进行通信，异步通信程序通常采用两种方法：查询法和中断法。



习 题

1. 串行数据传送与并行数据传送相比的主要优点和用途是什么？
2. 简述MCS-51单片机串行口四种工作方式的接收和发送数据的过程。
3. 串行口有几种工作方式？各工作方式的波特率如何确定？
4. 若晶体振荡器频率为11.0592MHz，串行口工作于方式1，波特率为4800bit/s，写出用T1作为波特率发生器的方式控制字和计数初值。
5. 简述利用串行口进行多机通信的原理。
6. 使用8031的串行口按工作方式1进行串行数据通信，假定波特率为2400bit/s，以中断方式传送数据，请编写全双工通信程序。



Q & A?

Thanks!

