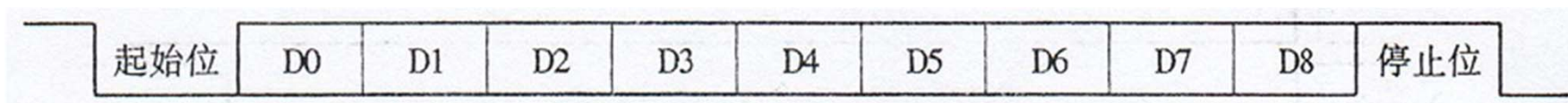


串行口的4种工作方式——方式2和方式3

串行口的工作方式2和方式3都为异步串行通信方式。两者除了波特率的确定方式不一样之外，其余都是相同的。

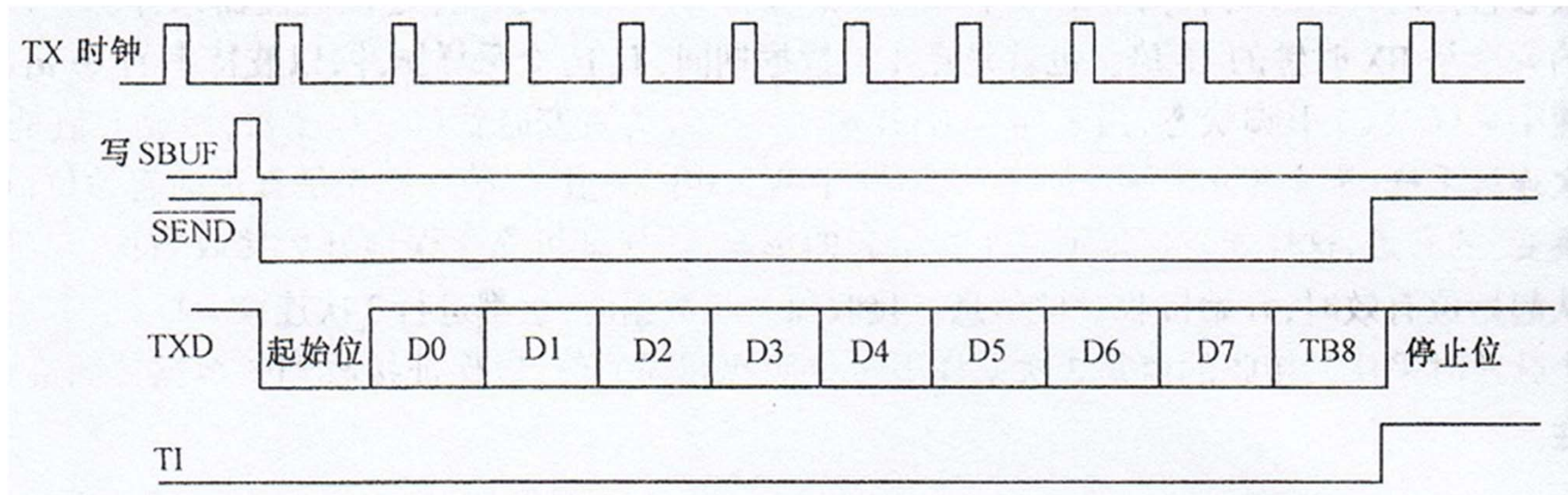
- 11位数据为一帧，1个起始位，8个数据位，1个第9位，1个停止位。发送/接收时，低位在先。
- **RXD**：数据接收端；**TXD**：数据发送端。
- 方式2的波特率 $= f_{osc} \times \frac{2^{SMOD}}{64}$
- 方式3的波特率 $= \frac{2^{SMOD}}{32} \times \text{定时器T1溢出率}$



方式2和方式3的帧格式

串行口的4种工作方式——方式2和方式3

发送过程：先根据通信协议设置TB8（第9位），然后CPU执行将8位数据写入SBUF的指令，启动发送；串行口开始将SBUF中的8位数据和TB8以波特率从TXD引脚输出。TB8发送结束时，TI置1。



方式2和方式3发送时序

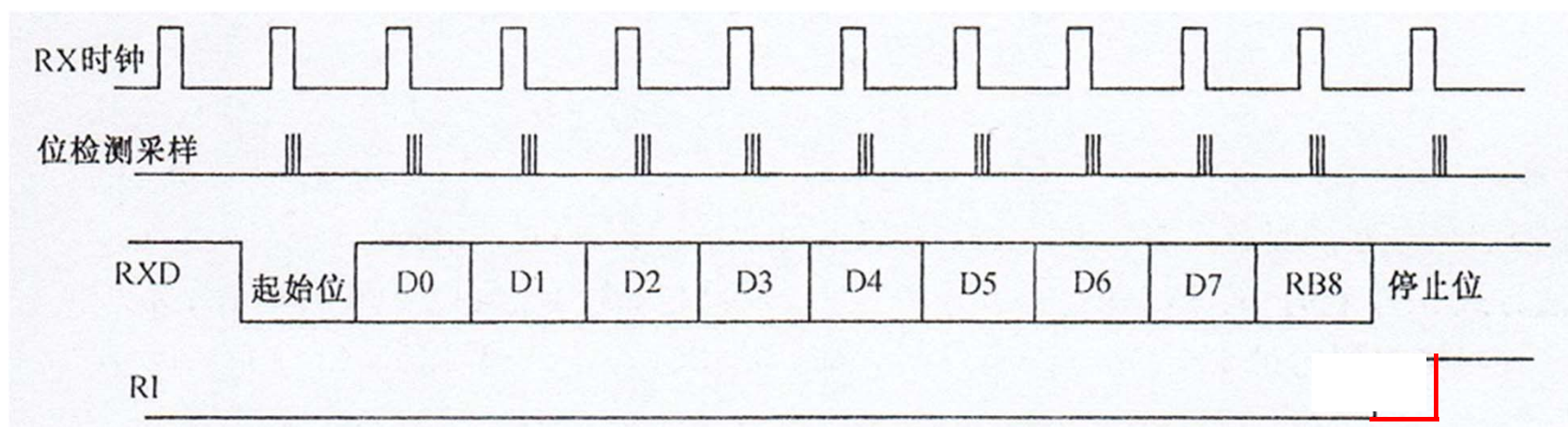
串行口的4种工作方式——方式2和方式3

接收过程：REN=1时允许接收。数据从RXD引入。当检测到起始位时开始接收。接收完一帧，若同时满足下列两个条件时接收有效：

- 1) RI=0; 2) SM2=0或第9位为1。

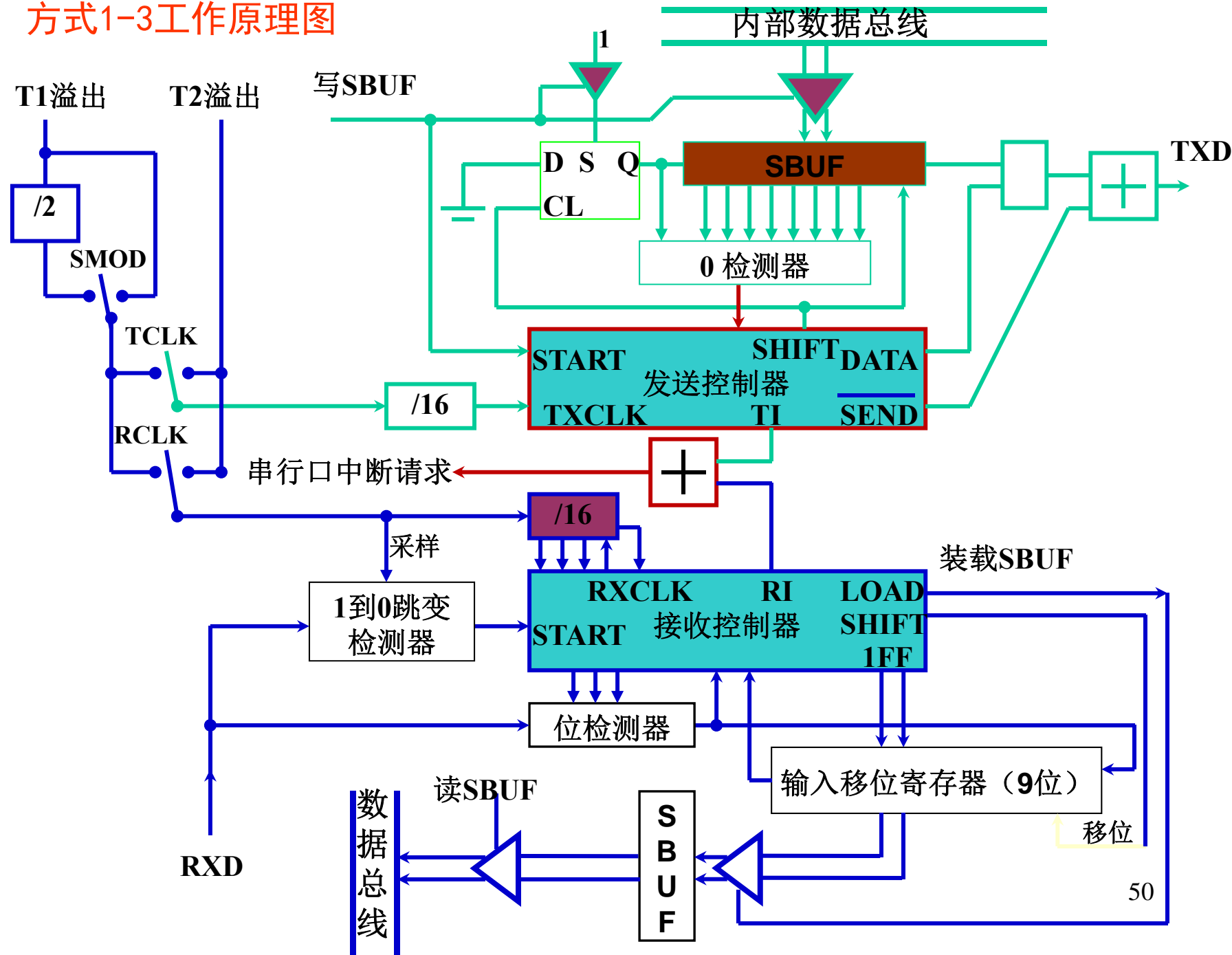
接收有效时，将接收的8位数据装入SBUF，第9位装入RB8，并置RI为1；否则丢弃接收数据，不置位RI。

位检测器的工作过程同方式1。



方式2和方式3接收时序

方式1-3工作原理图



总结：4种工作方式收/发帧的比较

收/发帧长度及结构：

方式	功能	帧长度	帧结构
0	移位寄存器	8	8位数据
1	UART	10	1位起始位（0）+8位数据+1位停止位（1）
2	UART	11	1位起始位（0）+8位数据+1位可编程位（RB8/TB8）+1位停止位（1）
3	UART	11	同方式2

总结：4种工作方式的收/发特点比较

收/发特点及有效接收条件：

方式	TB8、RB8	多机通信	产生最后一次移位脉冲时， 装载SBUF和RB8并置RI=1的 条件
0	不用	不支持	RI=0
1	停止位	不支持	(1) RI=0； (2) SM2=0或 接收到有效的停止位1
2	程控第九数据位	支持	(1) RI=0； (2) SM2=0或 接收到的第九数据位=1
3	程控第九数据位	支持	同方式2

波特率的制定方法

波特率的定义：

串行口每秒钟发送（或接收）的位数，称为波特率。设发送一位所需要的时间为T，则波特率为1/T。

方式0：波特率 = $f_{osc}/12$

方式1：波特率 = $\frac{2^{SMOD}}{32} \times \text{定时器T1溢出率}$

方式2：波特率 = $f_{osc} \times \frac{2^{SMOD}}{64}$

方式3：波特率 = $\frac{2^{SMOD}}{32} \times \text{定时器T1溢出率}$

T1溢出率 = 单位时间内T1的溢出次数 = $1 / (\text{T1的定时时长} t)$ ；

T1的定时时长 t ，就是T1溢出一次所用的时间。

方式1和方式3的波特率和定时器T1的溢出率有关，SMOD可取值0或1。

波特率的制定方法

4种工作方式波特率的比较：

方式	可编程性	影响因素 (f_{osc} 除外)	波特率
0	否	无	$f_{osc}/12$
1	是	PCON. 7、 T1溢出率	$2^{SMOD}/32x$ （定时器T1的溢出率）
2	是	PCON. 7	$f_{osc}/32$ 或 $f_{osc}/64$
3	是	PCON. 7、 T1溢出率	$2^{SMOD}/32x$ （定时器T1的溢出率）

其中：SMOD为TCN. 7， SMOD可取值0或1， f_{osc} 为系统晶振频率。

波特率的制定方法

定时器T1产生波特率的计算

通常定时器T1在作为波特率发生器时，采用定时器工作方式2（自动装初值），即TL1为8位计数器，TH1存放重装初值。T1的溢出率取决于：计数速率和初值。

$$\text{计数速率} = \begin{cases} f_{\text{osc}}/12 & ; \text{（定时器方式）} \\ \text{外部输入时钟频率} & ; \text{（计数器方式）} \end{cases}$$

溢出率的计算公式为：

$$\text{定时器T1的溢出率} = \frac{f_{\text{osc}}/12}{(256 - \text{TH1})}$$

所以，串行方式1、方式3的

$$\text{波特率} = \frac{2^{\text{SMOD}}}{32} \times \frac{f_{\text{osc}}}{12 \times (256 - \text{TH1})}$$

常用的波特率及计数器初值

常用波特率及T1计数器（方式2、自动重装）初值：

波特率	f (MHz)	SMOD	定时器		
			C/T	方式	重新装入值
方式0：1M	12	×	×	×	×
方式2：375K	12	1	×	×	×
方式1、3：62.5K	12	1	0	2	FFH
19.2K	11.0592	1	0	2	FDH
9.6K	11.0592	0	0	2	FDH
4.8K	11.0592	0	0	2	FAH
2.4K	11.0592	0	0	2	F4H
1.2K	11.0592	0	0	2	E8H
110	6	0	0	2	72H
110	12	0	0	1	FEEDH

常用波特率及TH1重装值

波特率计算例

验证9600 bits/s:

取 $f_{osc}=11.0592\text{MHz}$, $\text{SMOD}=0$, T1作波特率发生器, 工作在方式2 (自动重装载模式), 重载值 $\text{TH1}=\text{FDH}$:

波特率 = $2^{\text{SMOD}}/32 \times$ (定时器T1的溢出率), 即

$$\begin{aligned}\text{波特率} &= \frac{2^{\text{SMOD}}}{32} \times \frac{f_{osc}}{12 \times [256 - (\text{TH1})]} \\ &= \frac{2^0}{32} \times \frac{11.0592 \times 10^6}{12 \times (256 - 253)} \approx 9599.83(\text{bits} / \text{s})\end{aligned}$$

相对误差: $E=(9600 - 9599.83)/9600 = 0.0018\%$

波特率计算例

例题

若 $f_{osc} = 6\text{MHz}$ ，波特率为2400，设 $SMOD = 1$ ，T1采用定时器工作方式2，则定时/计数器T1的计数初值为多少？并进行初始化编程。

解：
$$X = 256 - 2^{SMOD} \times f_{osc} / (2400 \times 32 \times 12)$$
$$= 242.98 \approx 243 = \text{F3H}$$

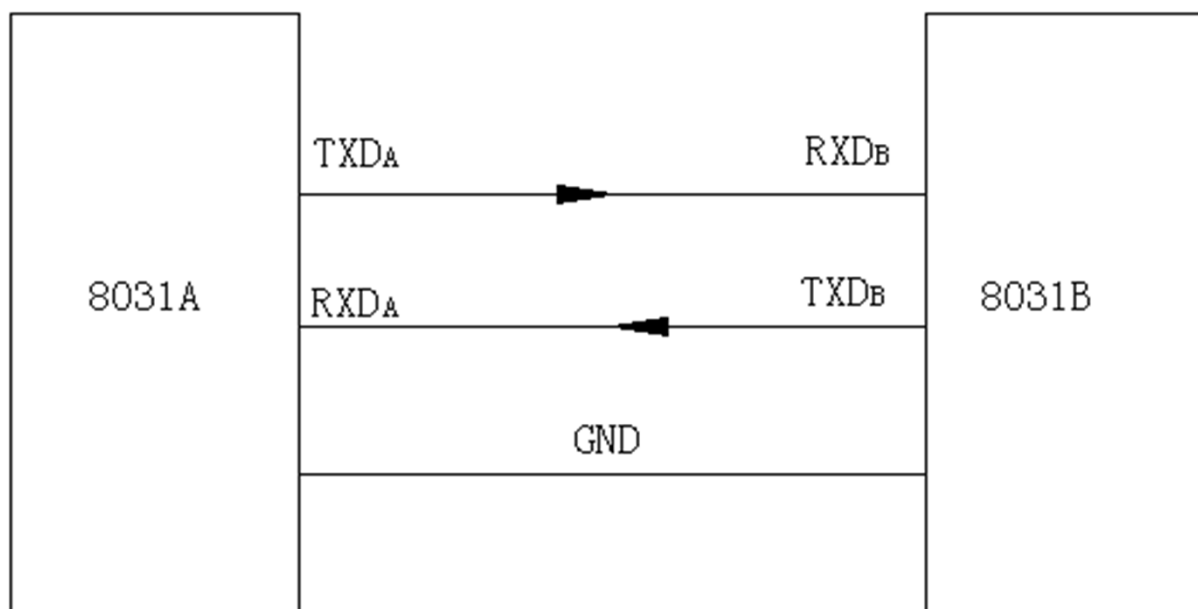
（若 $f_{osc} = 11.0592\text{MHz}$ ，波特率为2400，设 $SMOD = 0$ ，则 $X = \text{F4H}$ 。

```
MOV    TMOD, #20H    ; 设置定时器1方式2
MOV    PCON, #80H     ; 设置波特率倍增, SMOD = 1
MOV    TH1, #0F3H     ; 定时器1赋初值0F3H
MOV    TL1, #0F3H     ;
SETB   TR1            ; 开定时器1
MOV    SCON, #50H     ; 设置串口方式1, 允许接收位REN=1
```

双机串行异步通信

两台MCS-51单片机可采用三线方式连接，两台单片机的发送端TXD与接收端的RXD交错相连，地线相连。

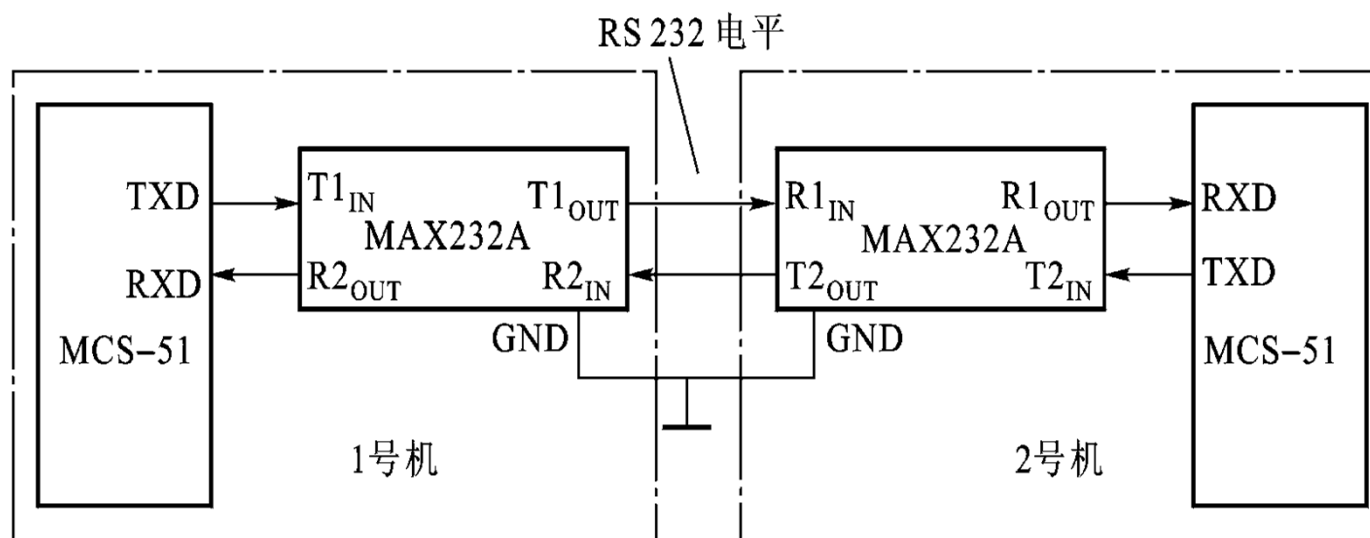
通信距离在几米以内。



两台单片机直接通信

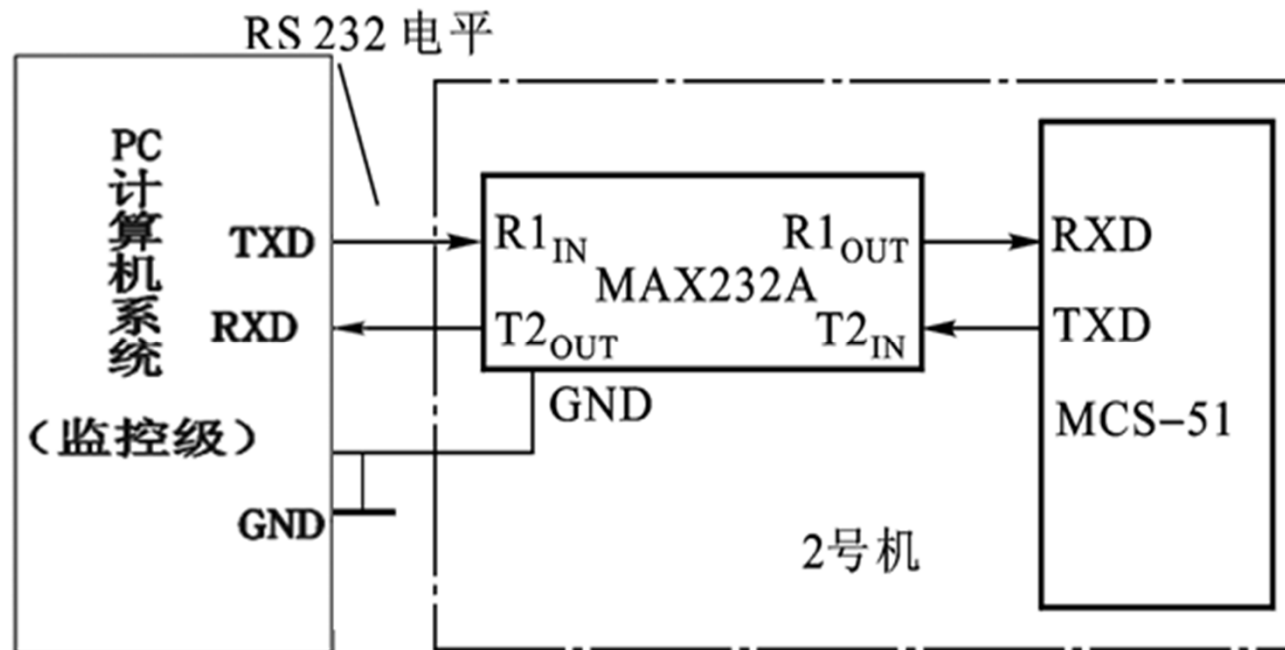
双机串行异步通信

- RS-232定义为一种在低速率串行通讯中增加通讯距离的单端标准。
- 发送端驱动器输出正电平在+5~+15V，负电平在-5~-15V之间。
- 接收器典型的工作电平在+3~+12V与-3~-12V。
- 最大传送距离约50英尺，最高速率为20kb/s。



两台单片机采用RS-232总线通信

双机串行异步通信



单片机和PC机通过RS232总线通信

多机主从通信

多机主从通信原理

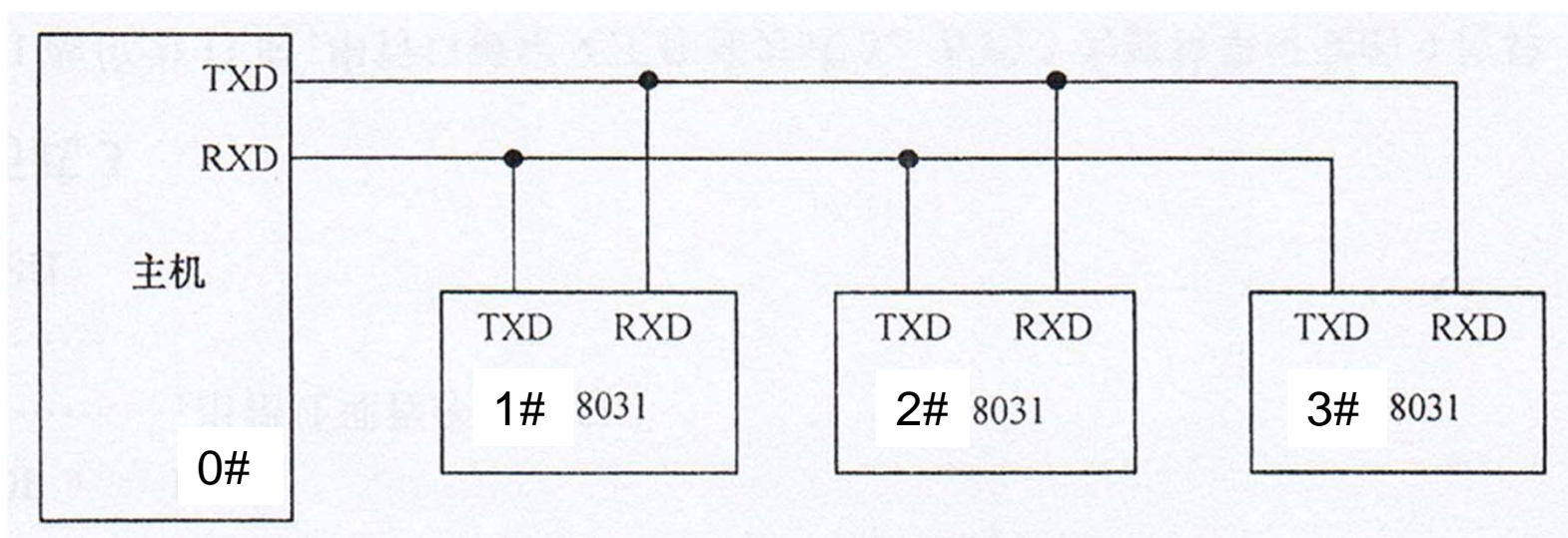
- 串行口控制寄存器SCON中的SM2为多机通信接口控制位；
- 串行口以模式2或3接收时：若SM2=1，则仅当接收到的第9位数据RB8=1时，数据才装入SBUF，置位RI，请求CPU进行处理；
- 当SM2=0时，则接收到一个数据后，不管第9位数据RB8是0还是1，都将数据装入接收缓冲器SBUF并置位RI，请求CPU处理；
- 若发地址帧，置TB8=1；若发数据帧，置TB8=0。

多机主从通信基本步骤

- 设置串行口通信模式2或3；
- 设置所有从机SM2=1；
- 置TB8=1，主机发送地址帧数据；
- 所有从机接收地址数据。若所接收的地址数据与本机地址编码相符合，即置SM2=0，并发应答信号给主机；若不相符合，则SM2依然为1；
- 主机收到应答，置TB8=0，发送数据帧信息；
- 此时，只有SM2=0的从机才能收到数据。

多机主从通信

MCS-51串行通信的方式2和方式3，可以实现同一条串行总线的多机通信过程。



多机主从通信

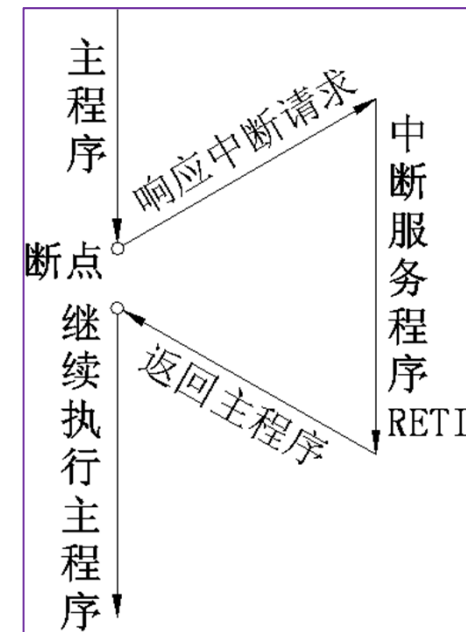
发送/接收字节	发送主机	接收从机	其他从机
地址字节=DA	TB8=1, 数据=DA	① SM2=1, 接收到第9位=1, 产生接收中断; ② DA→SBUF; ③ DA符合本机地址, 将SM2置为0, 准备接收数据字节。	① SM2=1, 接收到第9位=1, 产生接收中断; ② DA→SBUF; ③ DA不符合本机地址, 维持SM2=1。
数据字节1=DD1	TB8=0, 数据=DD1	① SM2=0, 产生接收中断; ② DD1→SBUF。	① SM2=1, 接收到第9位=0; ② 放弃DD1, 不产生接收中断。
.....
数据字节n=DDn	TB8=0, 数据=DDn	① SM2=0, 产生接收中断; ② DDn→SBUF。	① SM2=1, 接收到第9位=0; ② 放弃DDn, 不产生接收中断。
校验字节=CRC	TB8=0, 数据=CRC	① SM2=0, 产生接收中断; ② CRC→SBUF; ③ 检查校验是否正确, 本轮接收结束, SM2=1。	① SM2=1, 接收到第9位=0; ② 放弃CRC, 不产生接收中断。

1.9 MCS-51中断系统

中断的概念

当CPU正在处理某事件的时候，外部发生的某一事件请求CPU迅速去处理，于是CPU暂时中止当前的工作，转去处理所发生的事件。中断服务处理完该事件后，再返回到原来被中止的地方继续原来的工作，这样的过程称为中断。

保护现场，恢复现场，中断返回



中断作用：

- 1) 中断系统提高了CPU对外界异步事件的处理能力，解决了快速CPU与慢速的外部设备之间的矛盾，大大提高了CPU的利用率。
- 2) 中断系统使CPU能够及时地处理内部和外部的随机信息，提高了响应的实时性和故障处理能力。

中断系统结构

- ◆ 8031系列单片机有5个中断源——2个外部中断源INT0和INT1，2个片内定时器 / 计数器（T0和T1）溢出中断源，1个片内串行口中断源。
8032系列单片机有6个中断源，增加了一个定时器/计数器T2中断源。
- ◆ 每一个中断源都可独立设置为开/关中断。
- ◆ 分为两级——高级中断和低级中断。其中任何一个中断源的优先级均可由软件设定为高级或低级，能实现两级中断服务程序嵌套。

中断系统结构

5个中断源定义

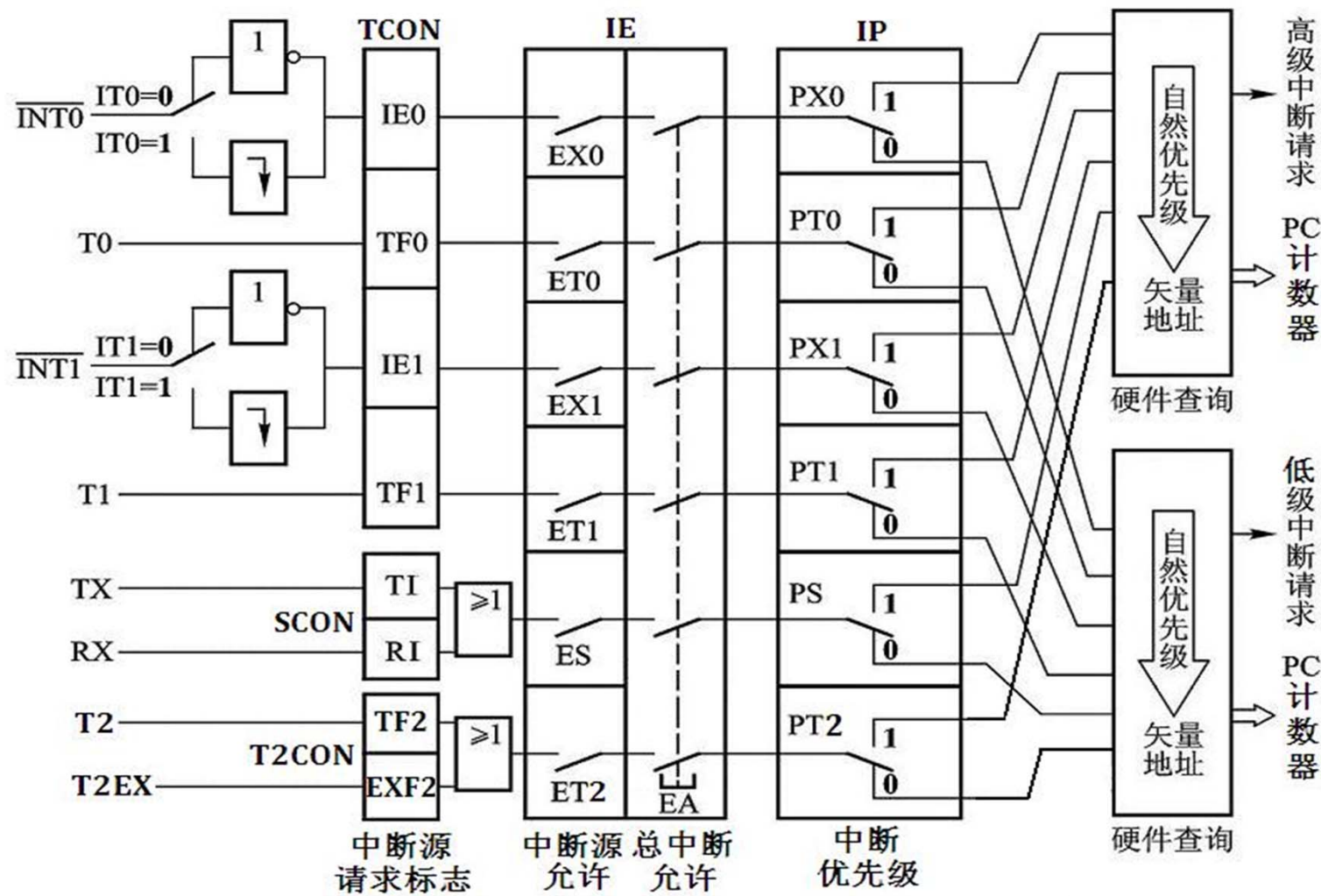
- 1) 外部中断0请求，由P3.2 ($\overline{\text{INT0}}$) 引脚输入。可由用户设定为两种触发方式：
电平触发或边沿触发。一旦输入信号有效，则向CPU申请中断，并且将中断标志IE0置1。
- 2) 外部中断1请求，由P3.3 ($\overline{\text{INT1}}$) 引脚输入。亦可由用户设定为电平触发方式，还是边沿触发方式。一旦输入信号有效，则向CPU申请中断，并将中断标志IE1置1。
- 3) 定时器T0溢出中断请求。当定时器T0产生溢出时，T0中断请求标志TF0置1，请求中断处理。
- 4) 定时器T1溢出中断请求。当定时器T1产生溢出时，T1中断请求标志TF1置1，请求中断处理。
- 5) 串行口发送 / 接收中断请求。当通过串行口发送或接收完一帧串行数据时，串行口中断请求标志TI或RI置1，请求中断处理。

中断系统结构

5个中断源比较：

中断源	请求标志	位地址	中断入口地址	来源	自然优先
外部中断0	IE0	TCON. 1	0003H	外部	1
定时器0	TF0	TCON. 5	000BH	内部	2
外部中断1	IE1	TCON. 3	0013H	外部	3
定时器1	TF1	TCON. 7	001BH	内部	4
串行口中断	TI或RI	SCON. 1 SCON. 0	0023H	内部	5

中断系统结构



中断系统相关寄存器

控制寄存器TCON

TCON是专用寄存器，字节地址为88H，它存放了外部的中断请求标志及T0和T1的溢出中断请求标志。复位后TCON为00H。

	D7	D6	D5	D4	D3	D2	D1	D0	
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	88H
位地址	8FH		8DH		8BH	8AH	89H	88H	

- ◆ **IT0**：选择外部中断0（INT0）触发方式控制位。（0：电平触发，1：边沿触发）
- ◆ **IE0**：外部中断0请求标志位。IE0=1，外部中断0向CPU申请中断。
- ◆ **IT1**：选择外部中断1（INT1）触发方式控制位。（0：电平触发，1：边沿触发）
- ◆ **IE1**：外部中断1请求标志位。IE1=1时，外部中断1向CPU申请中断。
- ◆ **TF0**：定时器T0溢出中断请求标志。T0被启动后，从初始值开始进行加1计数，当产生溢出时置TF0=1，向CPU申请中断。
- ◆ **TF1**：定时器T1溢出中断请求标志，其操作功能与TF0类同。

中断系统相关寄存器

特殊功能寄存器SCON

SCON是串行口控制寄存器，与中断有关的是它的低两位TI和RI。

	D7	D6	D5	D4	D3	D2	D1	D0	
SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	98H
位地址							99H	98H	

- ◆ **TI (SCON.1)**：串行口发送中断请求标志位。CPU每将一个字节写入发送缓冲器SBUF，启动发送完一个串行帧数据时，由硬件置位TI，请求中断。
- ◆ **RI (SCON.0)**：串行口接收中断请求标志位，当允许串行口接收数据时，每接收完一个串行帧数据，由硬件置位RI，请求中断。
- ◆ CPU在响应串行中断时，并不自动清“0”标志位TI和RI，需在中断服务程序中通过软件清“0”。

中断系统相关寄存器

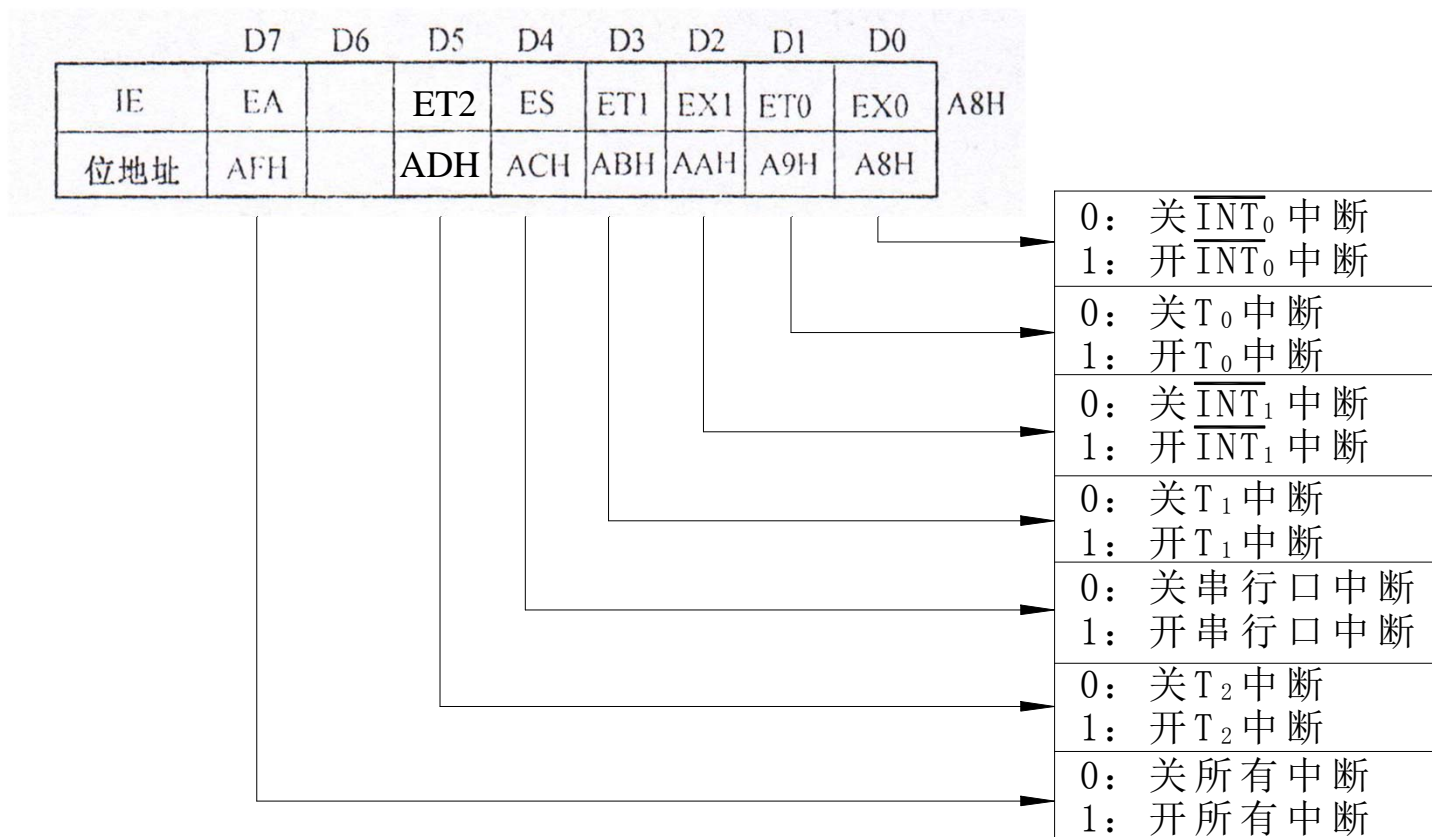
中断标志位一览表

中断标志位		位名称	说 明
TF1	T1溢出中断标志	TCON. 7	T1被启动计数后，从初值开始加1计数，计满溢出后由硬件置位TF1，同时向CPU发出中断请求，此标志一直保持到CPU响应中断后才由硬件自动清0。也可由软件查询该标志，并由软件清0。
TF0	T0溢出中断标志	TCON. 5	T0被启动计数后，从初值开始加1计数，计满溢出后由硬件置位TF0，同时向CPU发出中断请求，此标志一直保持到CPU响应中断后才由硬件自动清0。也可由软件查询该标志，并由软件清0。
IE1	外部中断1中断标志	TCON. 3	IE1 = 1，外部中断1向CPU申请中断。
IT1	中断触发方式控制位	TCON. 2	当IT1 = 0，外部中断1控制为电平触发方式；当IT1 = 1，外部中断1控制为边沿（下降沿）触发方式。
IE0	外部中断0中断标志	TCON. 1	IE0= 1，外部中断0向CPU申请中断。
IT0	中断触发方式控制位	TCON. 0	当IT0= 0，外部中断0控制为电平触发方式；当IT0 = 1，外部中断0控制为边沿（下降沿）触发方式。
TI	串行发送中断标志	SCON. 1	CPU将数据写入发送缓冲器SBUF时，启动发送，每发送完一个串行帧，硬件都使TI置位；但CPU响应中断时并不自动清除TI，必须由软件清除。
RI	串行接收中断标志	SCON. 0	当串行口允许接收时，每接收完一个串行帧，硬件都使RI置位；同样，CPU在响应中断时不会自动清除RI，必须由软件清除。

中断系统相关寄存器

中断允许寄存器IE

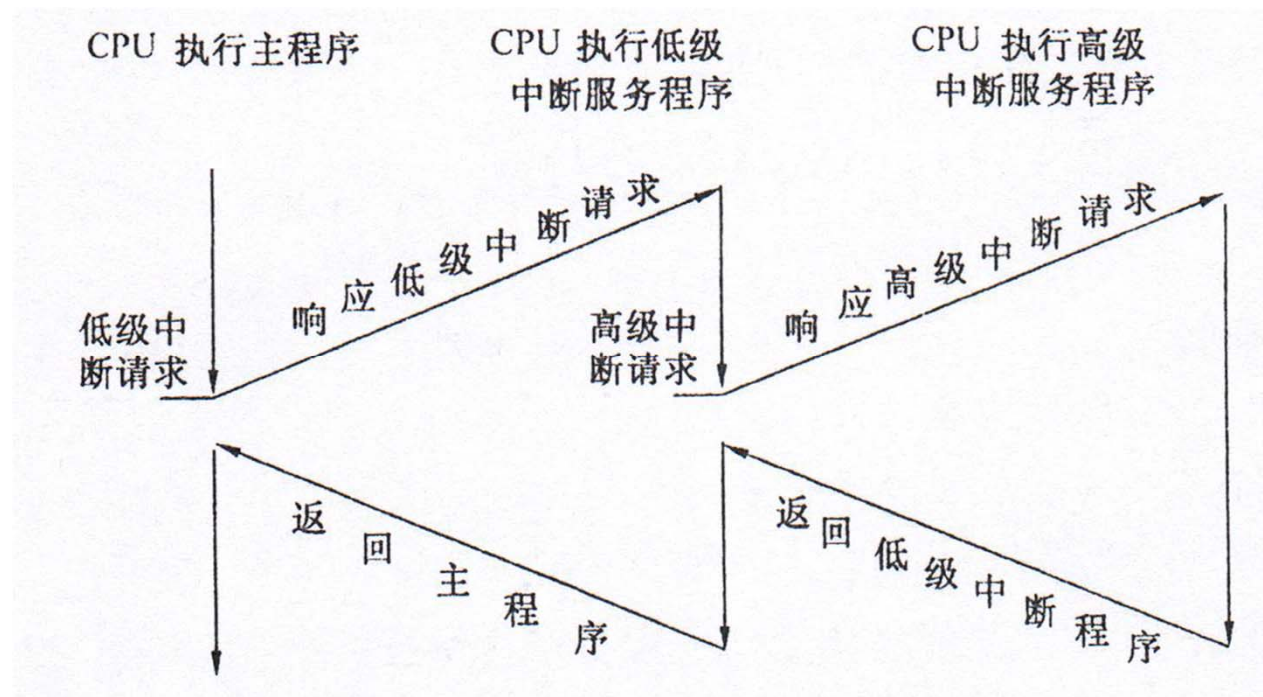
中断允许寄存器IE对中断的开放和关闭实现两级控制，第一级是总的开关中断控制位EA，另一级是每个中断源对应的中断请求允许位。



中断系统相关寄存器

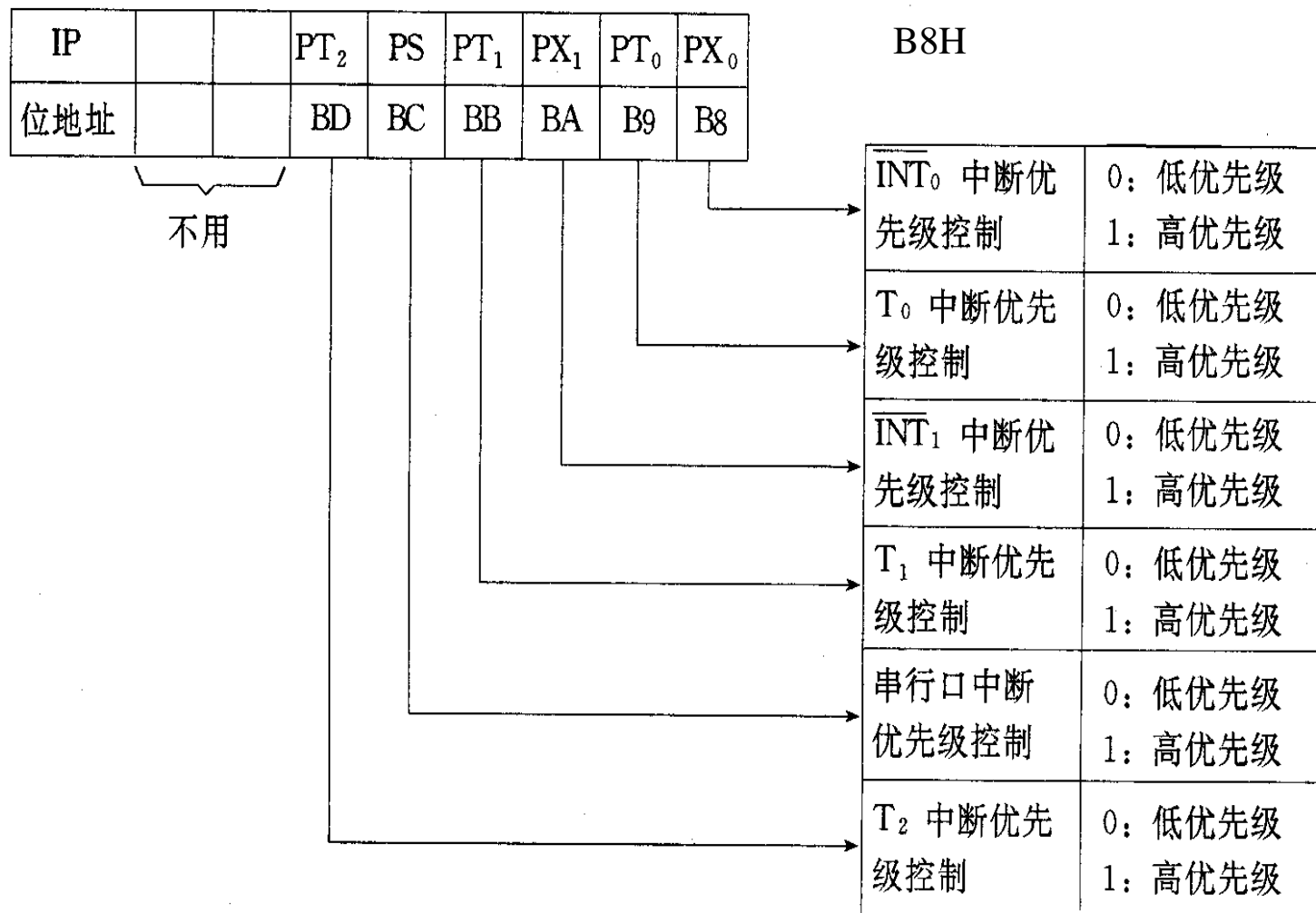
中断优先级寄存器IP

MCS-51单片机的中断系统有两个中断优先级，每一个中断请求源可以由软件设置为高优先级中断或低优先级中断。可实现两级中断嵌套，执行低优先级中断服务程序时，可被高优先级中断请求所中断。



中断系统相关寄存器

中断优先级寄存器IP



中断优先级结构

中断优先基本规则

- 1) 低优先级可被高优先级中断，反之则不能。
- 2) 任何一种中断（不管是高级还是低级），一旦得到响应，不会再被同级中断源中断。
- 3) 同时收到几个同级的中断请求时，哪一个中断请求能获得响应，取决于内部的硬件查询顺序（同级顺序）。

中断优先原则：（四句话）

- 1、低级中断，不能打断高级中断；
- 2、高级中断，可以打断低级中断；
- 3、同级中断，不能打断同级中断；
- 4、同级同时中断，查询顺序决定。

由两个内部不可寻址的“优先级状态”激活触发器来实现这一规则。

- 一个指示高优先级中断正在得到服务，阻断后来所有中断请求；
- 一个指示低优先级中断正在得到服务，阻断同级中断，但不阻断高优先级中断。

表 4-1 同一优先级中断的查询次序

中断源	中断级别
外部中断 0	最高
T0 溢出中断	
外部中断 1	
T1 溢出中断	
串行口中断	
T2 中断	
	最低

响应中断请求的条件

中断响应的必要条件：

- 1) CPU开中断，即IE寄存器中的中断总允许位EA=1。
- 2) 该中断源的中断允许位为“1”，即该中断没有被屏蔽。
- 3) 该中断源发出中断请求，即该中断源对应的中断请求标志为“1”。
- 4) 无同级或更高级的中断正在被服务，并且无同级顺序更高的中断源正在请求中断。

右表是各中断服务程序入口地址

通常在各中断服务程序入口地址处，放置一条无条件转移指令，使程序转向中断服务程序。

表 4-2 中断入口地址表	
中断源	中断入口地址
外部中断 0	0003H
定时器/计数器 T0	000BH
外部中断 1	0013H
定时器/计数器 T1	001BH
串行口中断	0023H
定时器/计数器 T2 (T2+EXF2)	002BH

响应中断请求的条件

中断响应受阻的条件（中断封锁条件）：

- 1) CPU正在处理同级或更高优先级的中断；
- 2) 现行机器周期不是正在执行的指令的最后一个机器周期，即现行指令完成前，不响应任何中断请求；
- 3) 正在执行的是中断返回指令RETI或访问专用寄存器IE或IP的指令。
也就是说，在执行RETI或是访问IE、IP的指令后，至少需要再执行一条其他指令，才会响应中断请求。

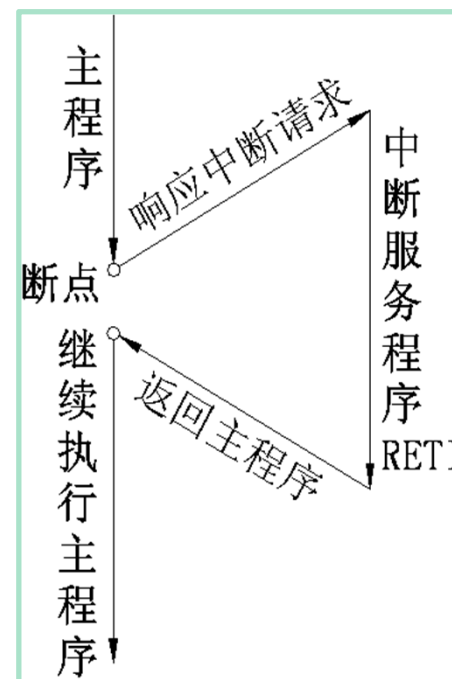
中断查询在每个机器周期中的S5P2重复执行，在S6进行优先级排序。如果某中断标志被置位，但因上述三种情况未能即时响应中断，该中断将继续等待。但是若封锁条件撤消后，该中断的标志位已不再存在（被软件清零），被拖延的中断将不再被响应。

中断响应时序

- 在每一机器周期中，都按顺序检查所有中断源；
- 在每一S6状态时，找到所有已激活中断，并安排好优先权；
- 在下一机器周期的S1状态，只要不受阻，最高优先级的中断便被响应。

中断响应协议

- 中断源提出请求；
- CPU应答，置位“优先级状态”激活触发器，阻断同级和低级中断；
- **PC入栈**
- 程序转向中断入口矢量单元；
- 执行中断服务程序；
- 最后一条指令RETI指令；
- 清除“优先级状态”激活触发器；
- 然后由堆栈**弹出**2字节装入**PC**中。



中断响应与返回过程

响应过程：（假设已使某中断请求标志置1）

- （1）先使相应优先级状态触发器置1
- （2）执行一个子程序的调用
 - 1）硬件清零相应中断请求标志（TI、RI除外）
 - 2）将当前PC内容压入堆栈——保护断点
 - 3）将中断服务子程序入口地址送PC——转移

也称中断
恢复

返回过程：（RETI执行后）

- （1）使相应优先级状态触发器清0
- （2）从堆栈中弹出栈顶的两个字节内容送PC——恢复断点
- （3）CPU接着中断处继续执行原程序

外部中断（ $\overline{\text{INTX}}$ ）

■ 触发方式

- 电平触发，TCON中断方式位ITX=0
- 边沿触发，TCON中断方式位ITX=1

■ 采样

- 电平触发的外部中断，在每个机器周期的S5P2对外部中断引脚采样，若为低电平，则TCON中的中断请求标志IEX被置1，以请求中断。
- 边沿触发的外部中断，在连续两个机器周期的S5P2对外部中断引脚采样，若第一次检测为高电平，第二次为低电平，则TCON中的中断请求标志IEX被置1，以请求中断。

■ 注意事项

- 电平触发，低电平至少保持1个机器周期，且保持中断请求有效直到中断得到响应。
- 边沿触发，高、低电平至少要各保持一个机器周期，才能确保CPU检测到电平的负跳变，而把中断请求标志IEX置1。

MCS-51单片机对于外部中断的响应时间是3~8个机器周期，具体时间取决于中断请求时正在执行的指令。

中断请求的撤销

CPU响应某中断请求后，在中断返回（RETI）前，该中断请求应该撤销，否则会引起另外一次中断。

1) 定时器/计数器T0、T1中断请求的撤销：

中断响应后，硬件自动清除中断请求标志位TF0或TF1，中断请求自动撤销。

2) 外部中断请求的撤销：

边沿方式：中断响应后，硬件自动清除中断请求标志位IE0或IE1，外部中断引脚的负跳沿信号是瞬态过程，不会维持。所以外部中断请求是自动撤销。

电平方式：中断响应后，硬件自动清除中断请求标志位IE0或IE1，但外部中断请求的低电平信号可能继续维持。可在系统中增加硬件电路解决，使得外部中断引脚的电平，从低电平变为高电平。

3) 串行口中断请求的撤销：

中断响应后，硬件不能自动清除TI和RI标志位，因此CPU响应中断后，必须在中断服务程序中，用软件来清除相应的中断标志位，以撤消中断请求。

中断请求的撤除

中断撤除方式一览表

中断源	中断标志	撤除方式
外部中断0	IE0	边沿触发的中断，由硬件自动清除 电平触发的中断，由硬件+软件清除
外部中断1	IE1	
定时器0	TF0	硬件自动清除
定时器1	TF1	
串行接收	RI	必须由软件清除
串行发送	TI	