

单片机原理与应用



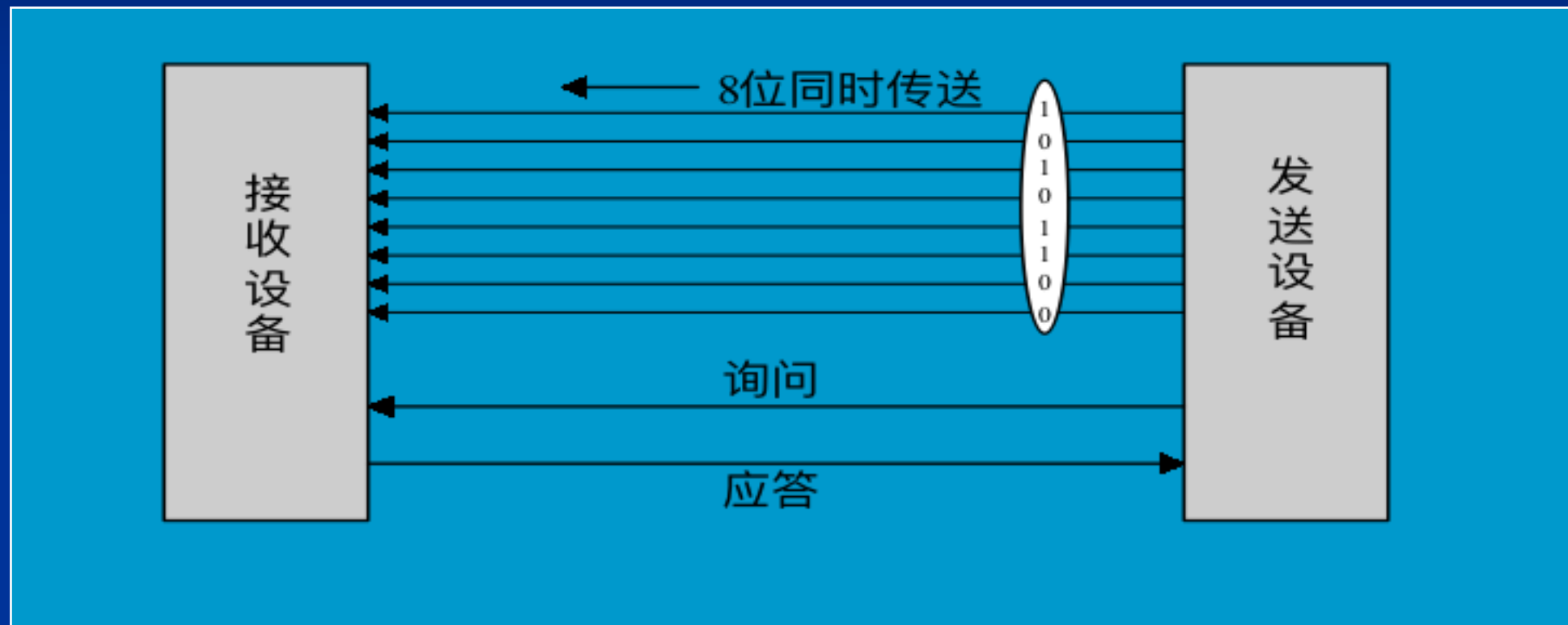
第7章 单片机的串行接口

信息科学与工程学院自动化系

本章内容和要求

- 7.1 串行通信概述
- 7.2 AT89C52单片机的串行接口
- 7.3 串行接口的工作方式
- 7.4 多机通信
- 7.5 波特率设置
- 7.6 串行接口的应用和编程

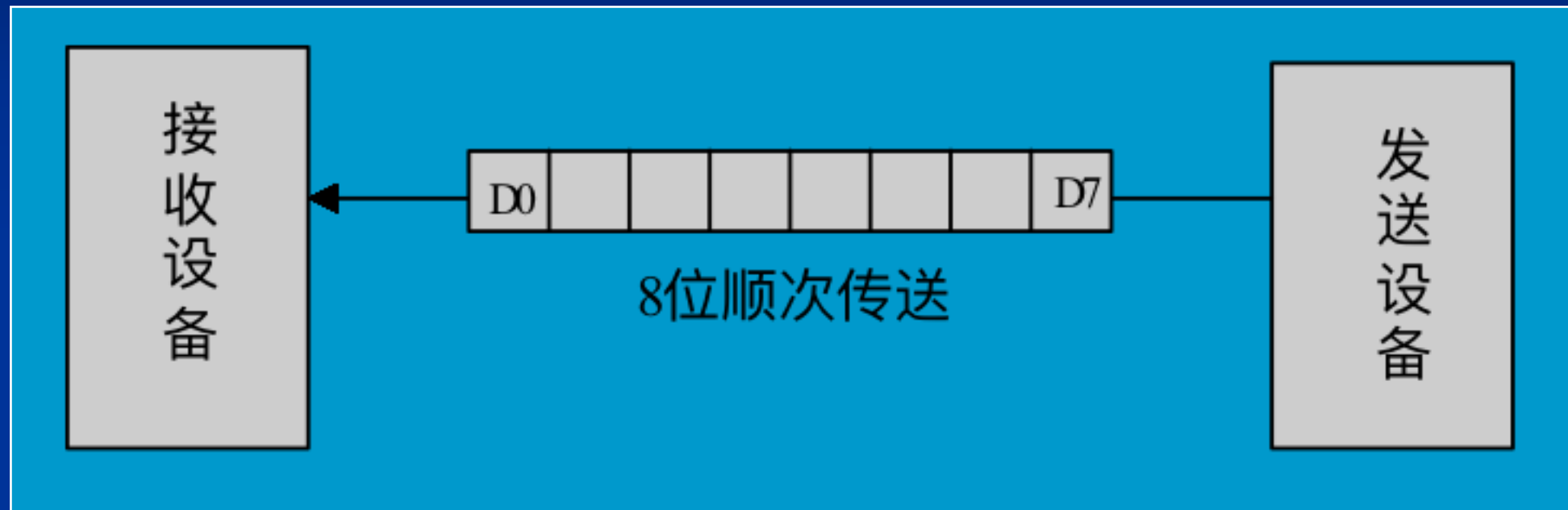
并行通信通常是将数据字节的各位用多条数据线同时进行传送。



并行通信的特点：

控制简单、传输速度快；由于传输线较多，长距离传送时成本高且使系统的抗干扰能力下降。

串行通信是将数据字节分成一位一位的形式在一条传输线上逐个地传送。



串行通信的特点：

传输线少，长距离传送时成本低，且可以利用电话网等现成的设备，但数据的传送控制比并行通信复杂。

串行通讯的传输方式

单工通讯方式



A端为发送站，B端为接收站，数据仅能从A站发至B站

半双工通讯方式



数据可以从A发送到B，也可以由B发送到A。不过同一时间只能作一个方向的传送，其传送方式由收发控制开关K来控制。

全双工通讯方式



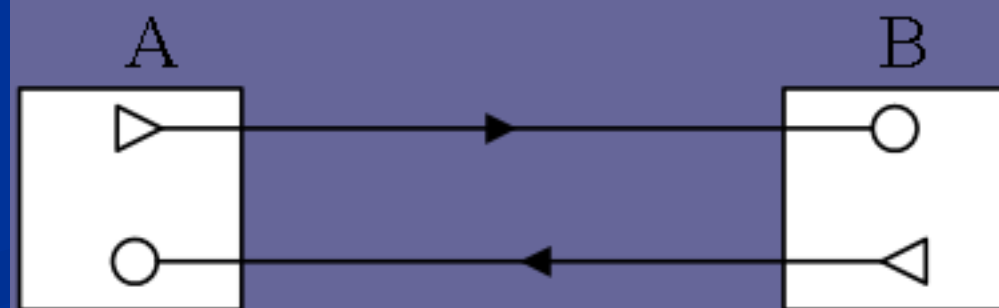
每个站（A、B）既可同时发送，又可同时接收。



(a) 单工通信方式



(b) 半双工通信方式



(c) 全双工通信方式

串行通讯的基本方式

(一).异步通讯

以**字符**为传送单位用起始位和停止位标识每个字符的开始和结束
字符间隔不固定，只需字符传送时同步。

异步通讯常用格式：一个字符帧



异步通讯的双方需要两项约定：

1.字符格式

一帧字符位数的规定：数据位，校验位，起始位和停止位。

2.波特率(位/秒)对传送速率的规定

例：要求每秒传送120个字符，每帧为10位。

解： $B=120 \times 10 = 1200$ 波特 每位0.83ms

数据位传输率 $= 120 \times 8 = 960$ 位/秒

(二).同步通讯

以一串字符为一个传送单位，字符间不加标识位，在一串字符开始用同步字符标识，硬件要求高，通讯双方须严格同步。

数据格式：按不同方式，一帧位数 **8/10/11**发送/接收时，数据皆低位在前。



一帧字符发送/接收结束，置位标志位(TI/RI)并申请SIO中断。

中断控制：中断允许位ES

中断入口：**0023H**

奇偶校验

在发送时，每个字符的最高位置后附加一个奇偶校验位，可为“1”或“0”，以保证整个字符（包括校验位）为“1”的位数为偶数（偶校验）或为奇数（奇校验）。

接收时，按照发送方规定的同样的奇偶性，对接收到的每个字符进行校验。若二者不一致，说明出现了差错。

奇偶校验是一个字符校验一次，针对单个字符进行的。

校验和

数据发送时，发送方对块中数据简单求和，产生一单字节校验字符（校验和）附加到数据块结尾。

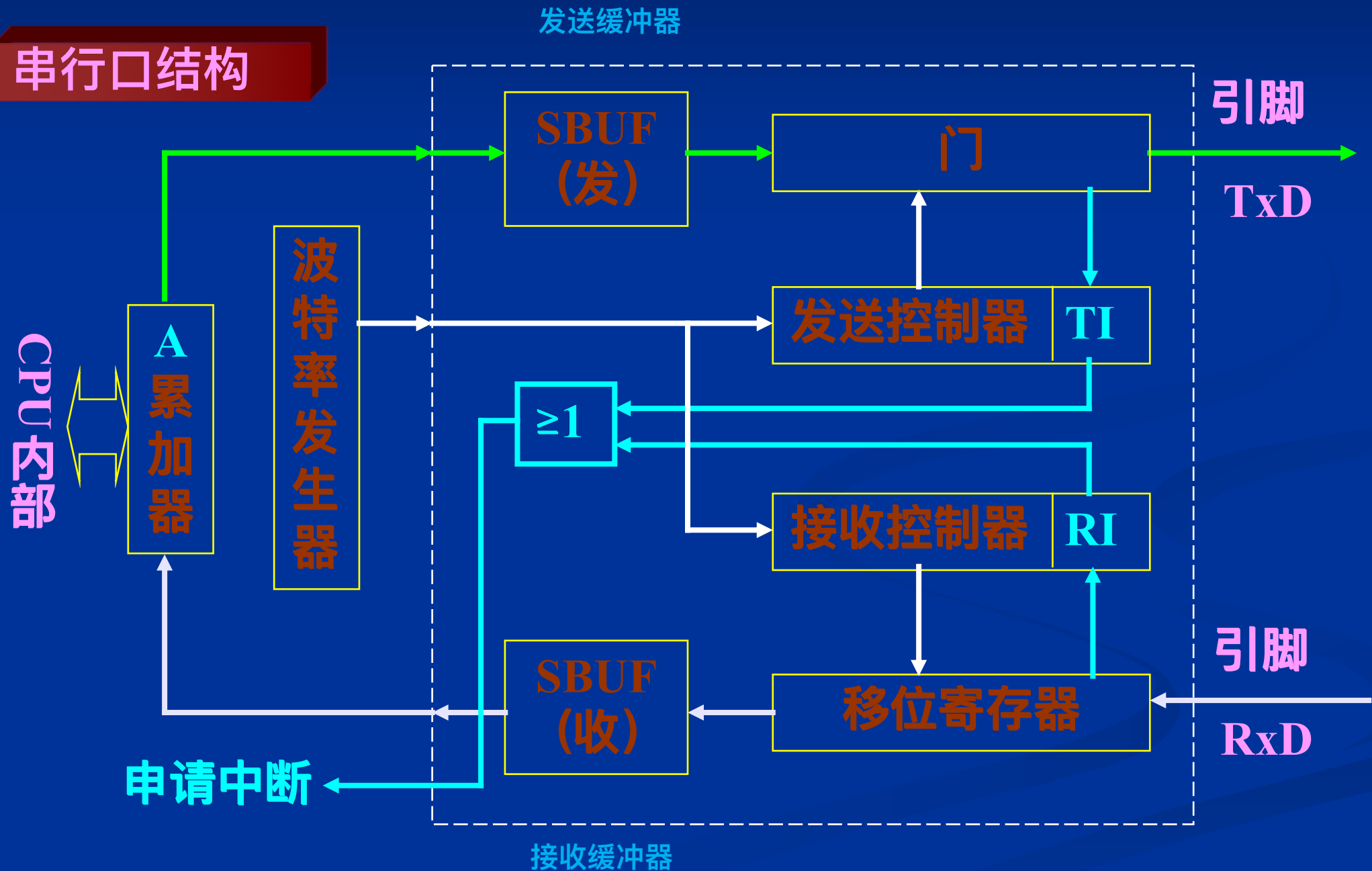
接收方对接收到的数据算术求和后，所得的结果与接收到的校验和字符比较。如果两者不同，即表示接收有错。

但校验和不能检测出排序错。即使数据块是随机、无序地发送，产生的校验和仍然相同。

7.2 AT89C52单片机的串行接口

1个全双工串行接口，可同时进行发送和接收。

串行口结构



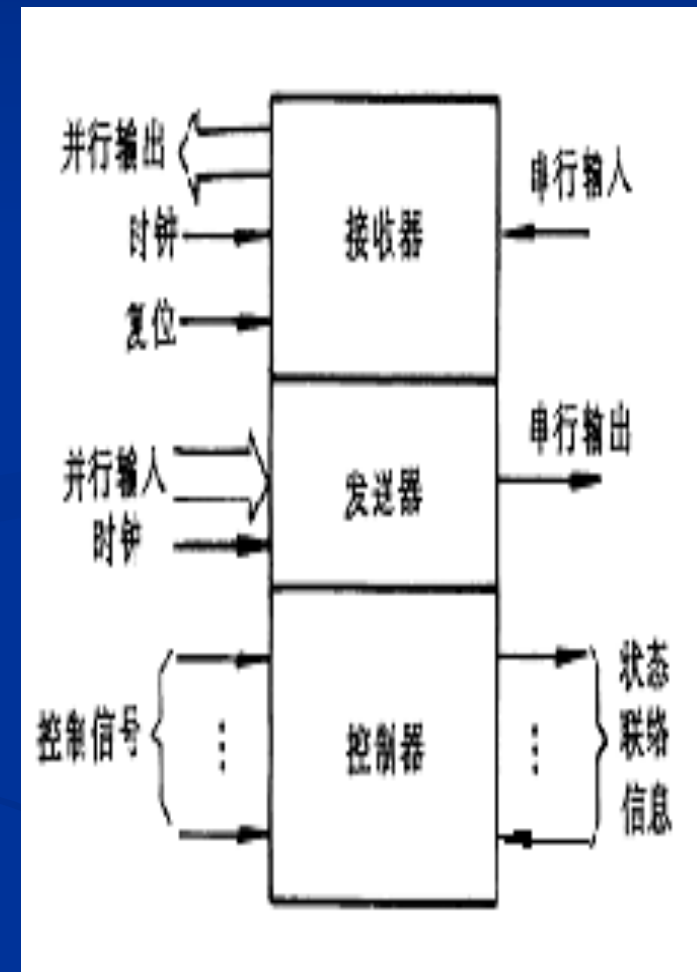
串行接口输入/输出引脚：TXD(P3.1)、RXD(P3.0)

串行接口功能

1.发送器：并→串数据格式转换，添加标识位和校验位，一帧发送结束，设置结束标志，申请中断。

2.接收器：串→并数据格式转换，检查错误，去掉标识位，保存有效数据，设置接收结束标志，申请中断。

3.控制器：接收编程命令和控制参数，设置工作方式：同步/异步、字符格式、波特率、校验方式、数据位与同步时钟比例等。



- 单片机内部有一个**串行数据缓冲寄存器SBUF**，可直接寻址的特殊功能寄存器，**地址99H**，实际是由两个八位寄存器组成，一个作发送缓冲寄存器，一个作接收缓冲寄存器。由读写信号区分，但同一地址**99H**。
- 串行口控制寄存器**SCON**，用于串行数据通信中控制和监视串行口工作状态；
- 电源控制及波特率选择寄存器**PCON**：串行口波特率的倍增控制。

串行接口控制

1.数据缓冲器SBUF

两个物理上**独立的**串行数据缓冲寄存器，发送SBUF和接收SBUF**共用一个地址99H**。

1) **发送SBUF**存放待发送的8位数据，写入SBUF时将同时启动发送。

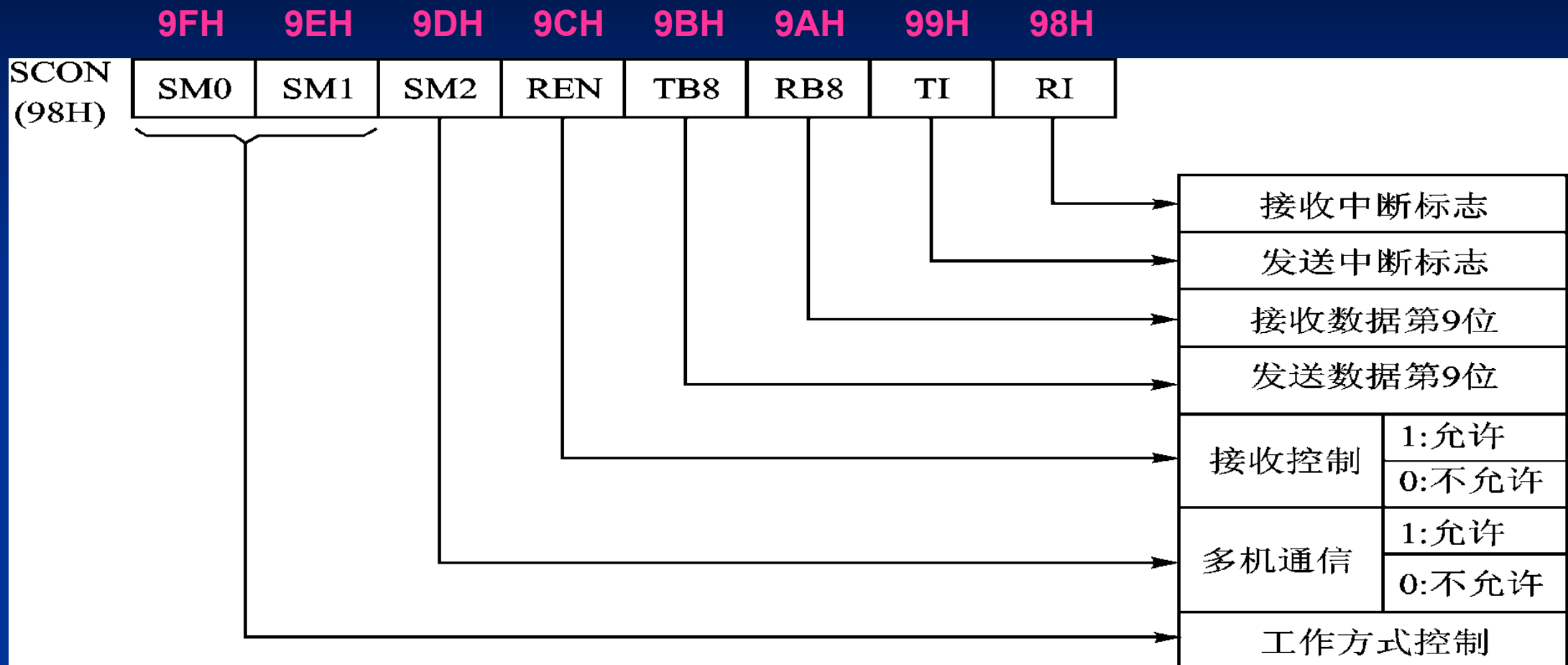
发送指令： MOV SBUF, A

2) **接收SBUF**存放已接收成功的8位数据，供CPU读取。

读取串行口接收数据指令：

MOV A, SBUF

2. 串行口控制/状态寄存器SCON(98H)



SM0, SM1: 串行接口工作方式选择位

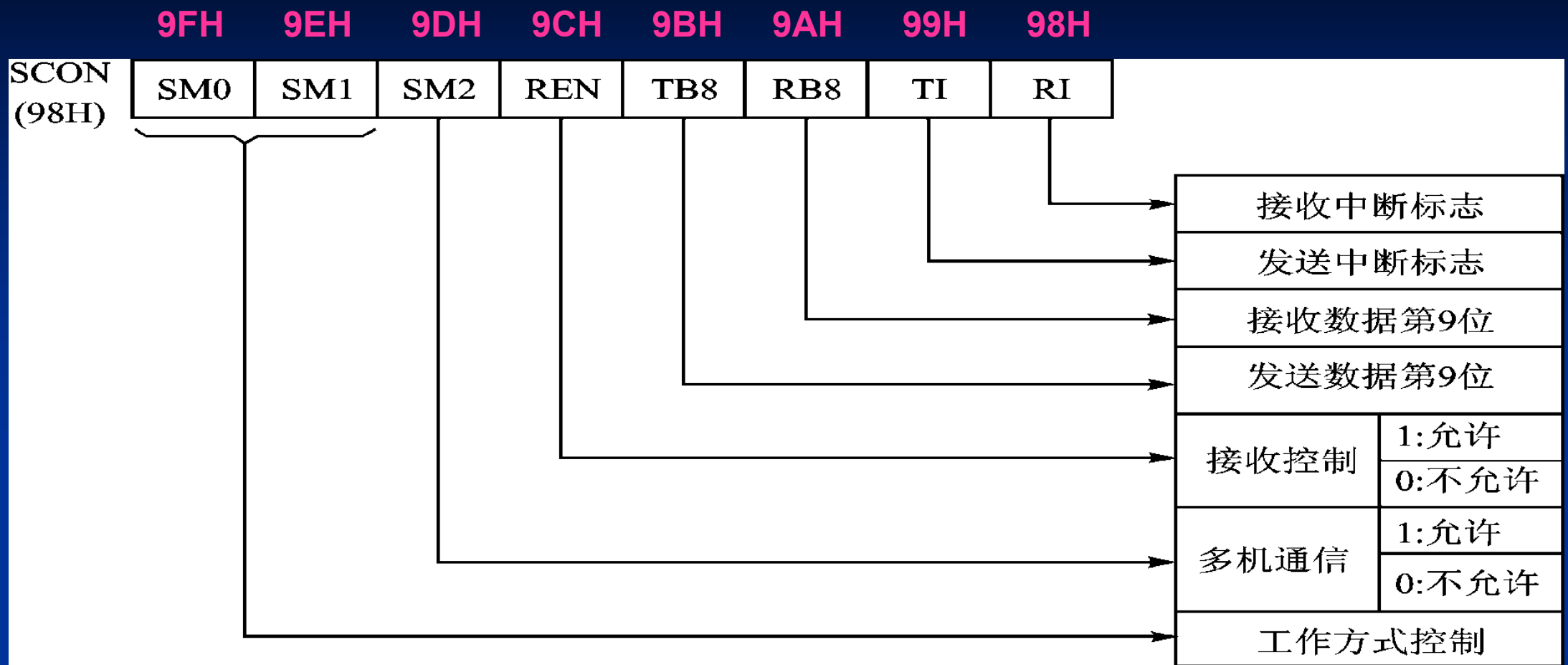
0 0 方式0: 8位移位寄存器

0 1 方式1: 10位异步通信接口 (波特率可变)

1 0 方式2: 11位异步通信接口 (波特率固定)

1 1 方式3: 11位异步通信接口 (波特率可变)

SM2: 方式2和方式3的多机通信控制位。



REN: 允许接收控制位, REN=1, 允许接收; REN=0, 禁止接收。

TB8: 发送的第9位数据位, 可用作校验位和地址/数据标识位

RB8: 接收的第9位数据位或停止位, 方式0时不使用。

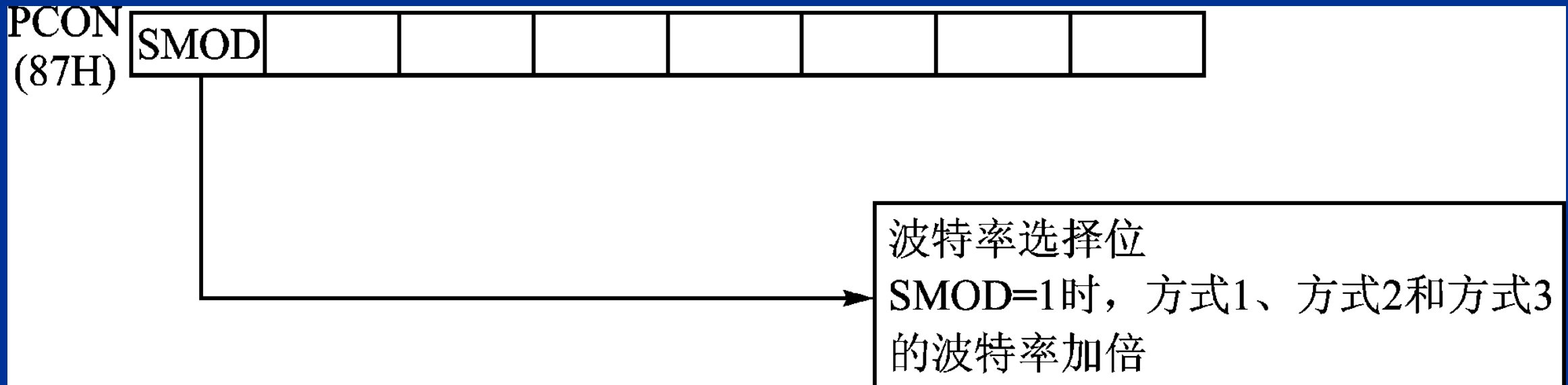
TI: 发送中断标志, 发送一帧结束, TI=1, 必须软件清零

RI: 接收中断标志, 接收一帧结束, RI=1, 必须软件清零

3. 节电控制寄存器（电源控制寄存器）

PCON (87H)

PCON中只有SMOD位与串行口工作有关：不可位寻址



SMOD (PCON.7) 波特率倍增位。

在串行口方式1、方式2、方式3时，波特率与SMOD有关，当SMOD=1时，波特率提高一倍。复位时，SMOD=0。

7.3 串行接口的工作方式

SM0, SM1: 串行口4种工作方式

0 0 方式0: 8位同步移位寄存器

波特率固定: $B = f_{osc}/12$

0 1 方式1: 10位异步通信接口

波特率可变, 按公式计算

1 0 方式2: 11位异步通信接口

波特率固定: $B = (2^{SMOD}/64) \times f_{osc}$

1 1 方式3: 11位异步通信接口

波特率可变, 按公式计算

波特率设置

■ 方式0:

波特率固定为振荡器频率 f_{osc} 的1/12,

$$B = f_{osc}/12$$

且不受PCON寄存器中SMOD位的影响。

■ 方式2:

波特率与振荡器频率 f_{osc} 和PCON中的SMOD位有关,

$$B = (2^{SMOD}/64) \times f_{osc}$$

当SMOD=0: $B = (1/64) \times f_{osc}$;

当SMOD=1: $B = (1/32) \times f_{osc}$ 。

波特率设置

■ 方式1、3: $B=(2^{\text{SMOD}}/32) \times \text{T1溢出率}$

T1为方式2的时间常数: $C = 2^8 - t/T$

溢出时间: $t = (2^8 - C)T = (2^8 - C) \times 12 / f_{\text{osc}}$

T1溢出率: $1/t = f_{\text{osc}} / [12 \times (2^8 - C)]$

∴ 波特率 $B = (2^{\text{SMOD}} / 32) \times f_{\text{osc}} / [12 \times (2^8 - C)]$

串行口方式1、3, 根据波特率选择T1工作方式, 计算时间常数。

$\text{TH1}、\text{TL1} = C = 2^8 - f_{\text{osc}} / 12 \times 2^{\text{SMOD}} / (32 \times B)$

(注意: T1工作于方式2)

👉 溢出率: T1溢出的频繁程度, 即: T1溢出一次所需时间的倒数。

T1 经常被用做串行口的波特率发生器

(1) 方式0：8位同步移位寄存器方式。用于扩展并行I/O接口。

1) .一帧8位，无起始位和停止位,低位在先，高位在后。

2) .RXD：数据输入/输出端。

TXD：同步脉冲输出端，每个脉冲对应一个数据位。

3) .波特率 $B = f_{osc}/12$

如： $f_{osc}=12\text{MHz}$ ， $B=1\text{MHz}$ ， 每位数据占 $1\mu\text{s}$ 。

4) .发送过程：

写入SBUF，启动发送，SBUF中的串行数据由RXD逐位移出，一帧发送结束，TI=1。

接收过程：

REN=1且RI=0，启动接收，串行数据由RXD逐位移入寄存器，并装载到SBUF中，一帧接收完毕，RI=1。

(2) 方式1

8位数据异步通讯方式

1) .一帧**10位**：8位数据位，1个起始位(0)，1个停止位(1)。

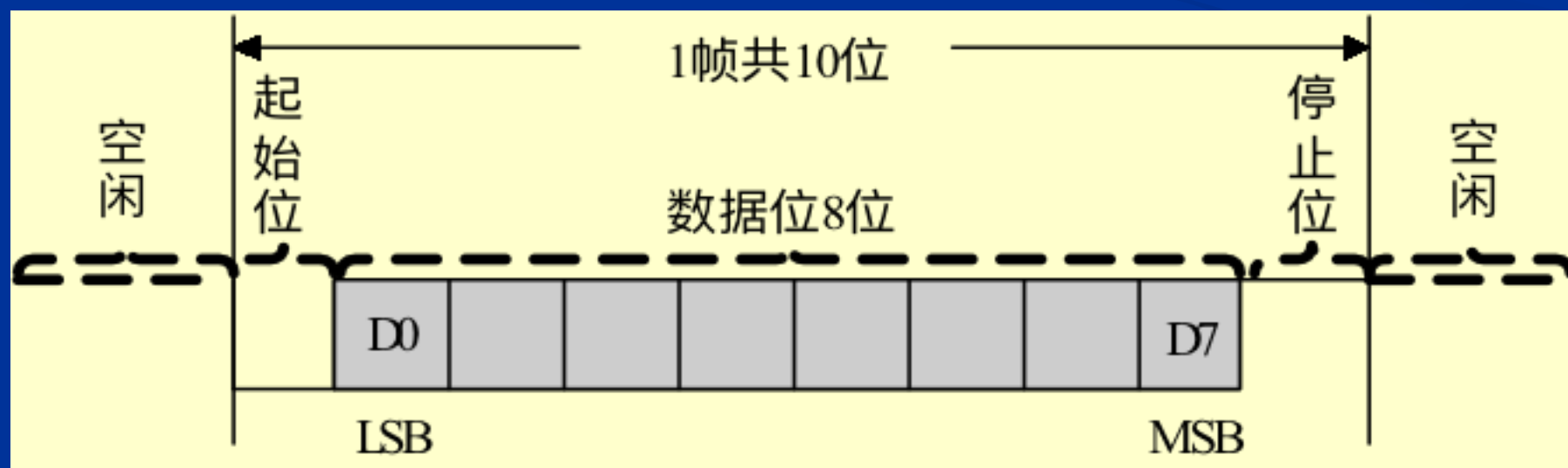
2) .**RXD**：接收数据端。 **TXD**：发送数据端。

3) .波特率可变：用T1作为波特率发生器，

$$B=(2^{SMOD}/32)\times T1 \text{溢出率。}$$

4) .**发送**：写入SBUF，同时启动发送，一帧发送结束，TI=1。

接收：REN=1，允许接收。接收完一帧，若**RI=0且停止位为1** (或SM2=0)，将接收数据装入SBUF，**停止位装入RB8**，并使RI=1；否则丢弃接收数据，不置位RI。



(3) 方式2和方式3

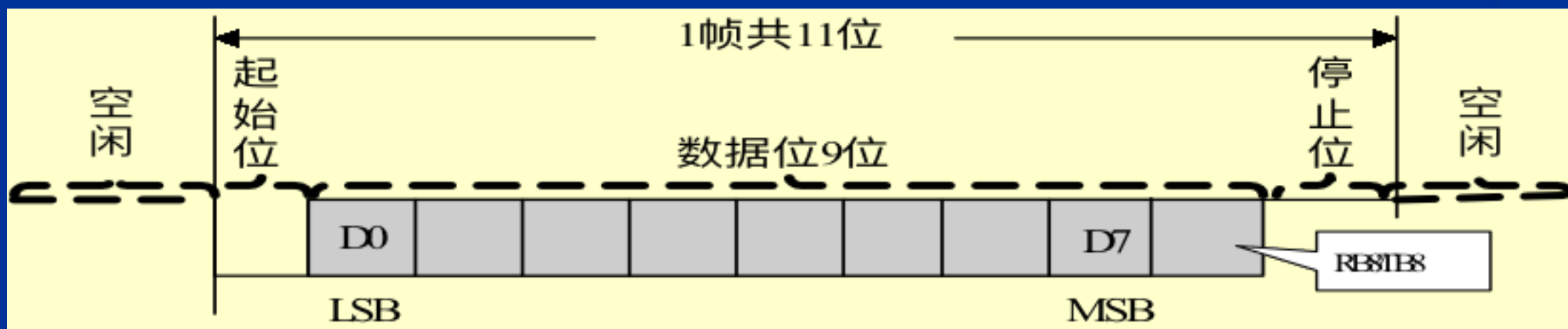
9位数据异步通讯方式。

1. 一帧为**11位**：9位数据位，1个起始位(0)，1个停止位(1)。
第9位数据位在**TB8/RB8**中，常用作校验位和多机通讯标识位。

2. **RXD**：接收数据端，**TXD**：发送数据端。

3. 波特率： 方式2： $B = (2^{\text{SMOD}}/64) \times f_{\text{osc}}$ 。

方式3： $B = (2^{\text{SMOD}}/32) \times T1 \text{溢出率}$ 。



4. **发送**：先装入TB8，写入SBUF并启动发送，发送结束，TI=1。

接收：REN=1，允许接收。接收完一帧，若**RI=0且第9位为1** (或**SM2=0**)，将接收数据装入接收SBUF，**第9位装入RB8**，使RI=1，向CPU请求中断；否则丢弃接收数据，不置位RI。

装入 SBUF和 RB8以及置位 RI的信号只有在产生最后一个移位脉冲且同满足下列两个条件,才会产生:

① RI=0;

② SM2 =0 或接收到的第 9 位数据为“1”。

上述两个条件中任一个不满足,所接收的数据帧就会丢失,不再恢复。两者都满足时,第 9 位数据装入 RB8,前 8 位数据装入 SBUF。

注意:与方式1不同,方式 2 和 3 中装入 RB8 的是第 9 位数据,而不是停止位。

串行口相关的SFR

☞ **SBUF** 两个同名的接收/发送缓冲寄存器

➡ 缓存发送/接收数据（8位）

☞ **SCON** 串行口控制寄存器

☞ **PCON** 电源控制寄存器

} ➡ 设置工作方式

☞ **TMOD** 定时方式寄存器

☞ **TH1, TL1** 定时器1初值寄存器

☞ **TCON** 定时器控制寄存器

} ➡ 设置波特率

串行接口的初始化

在使用串行口进行数据通信之前，应对其进行**初始化**，主要是设置产生波特率的T1、串行口控制和中断控制。具体步骤如下：

- 确定T1的工作方式（编程TMOD寄存器）；
- 计算T1的初值，装载TH1、TL1；
- 启动T1（编程TCON中的TR1位）：**SETB TR1;**

对T1不要开中断！！

- 确定串行口控制（编程SCON寄存器）：
 - 对SCON选工作方式
 - 对PCON设波特率加倍位“SMOD”(缺省值=0)
 - 如果是接收数据, 要先置“1”REN位

串行口中断方式工作时，要进行中断设置（编程IE、IP寄存器）。

7.3~7.6 初始化与应用

串行口初始化编程格式：

```
SIO: MOV  SCON, #控制状态字; 写方式字且TI=RI=0
      (MOV  PCON, #80H)      ; 波特率加倍
      ( MOV  TMOD, #20H )    ; T1作波特率发生器
      ( MOV  TH1, #X )      ; 选定波特率
      ( MOV  TL1, #X )
      ( SETB TR1)           ; 启动T1工作
      ( SETB EA)             ; 开串行口中断
      ( SETB ES)
```

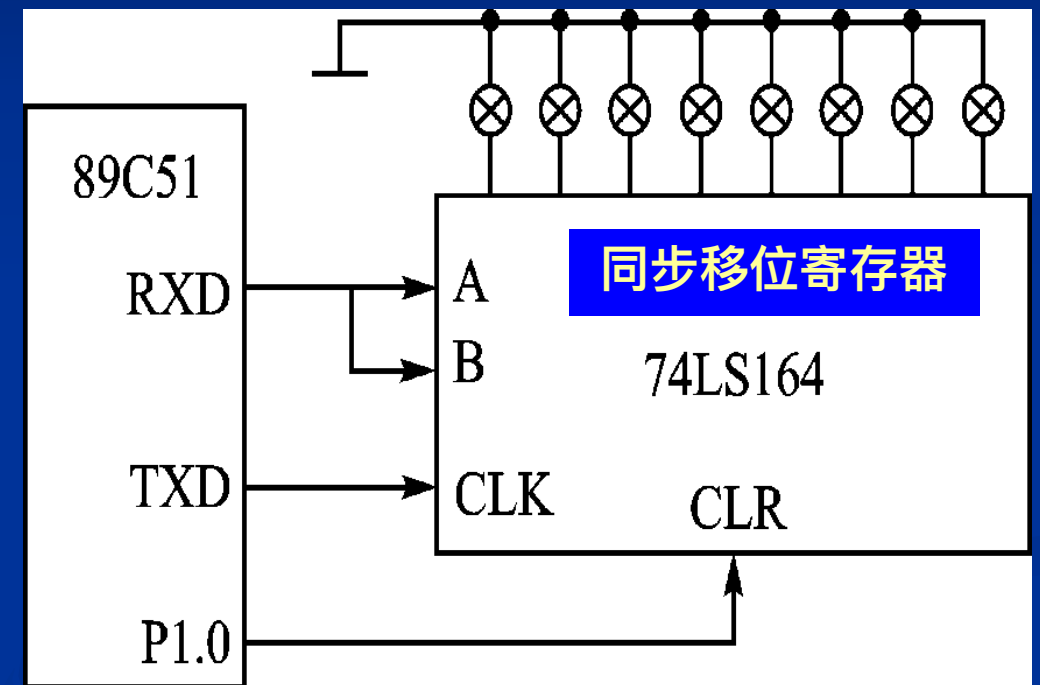
1. 串行口方式0的应用

- MCS-51单片机串行口基本上是**异步通信接口**，但在方式0时是同步操作。外接串入——并出或并入——串出器件，可实现**I/O的扩展**。
- 串行口方式0的数据传送可以采用**中断**方式，也可用**查询**方式。无论哪种方式，都要借助于TI或RI标志。
- 在串行口**发送**时，或者靠TI置位后引起中断申请，在中断服务程序中发送下一组数据；或者通过查询TI的值，只要TI为0就继续查询，直到TI为1后结束查询，进入下一个字符的发送。
- 在串行口**接收**时，由RI引起中断或对RI查询来决定何时接收下一个字符。在开始串行通信前，要先对SCON寄存器初始化，进行工作方式的设置。在方式0中，SCON寄存器的初始化只是简单地把00H送入SCON就可以了。

例1：用89C51串行口外接74LS164
串入——并出移位寄存器扩展8位
并行口；8位并行口的每位都接一个发光二极管，要求发光二极管从左到右以一定延迟轮流显示，并不断循环。设发光二极管为共阴极接法，如图所示。

解：设数据串行发送采用中断方式，
显示的延迟通过调用延迟程序
DELAY来实现。

移位寄存器74LS164实现：
串行→并行的数据转换



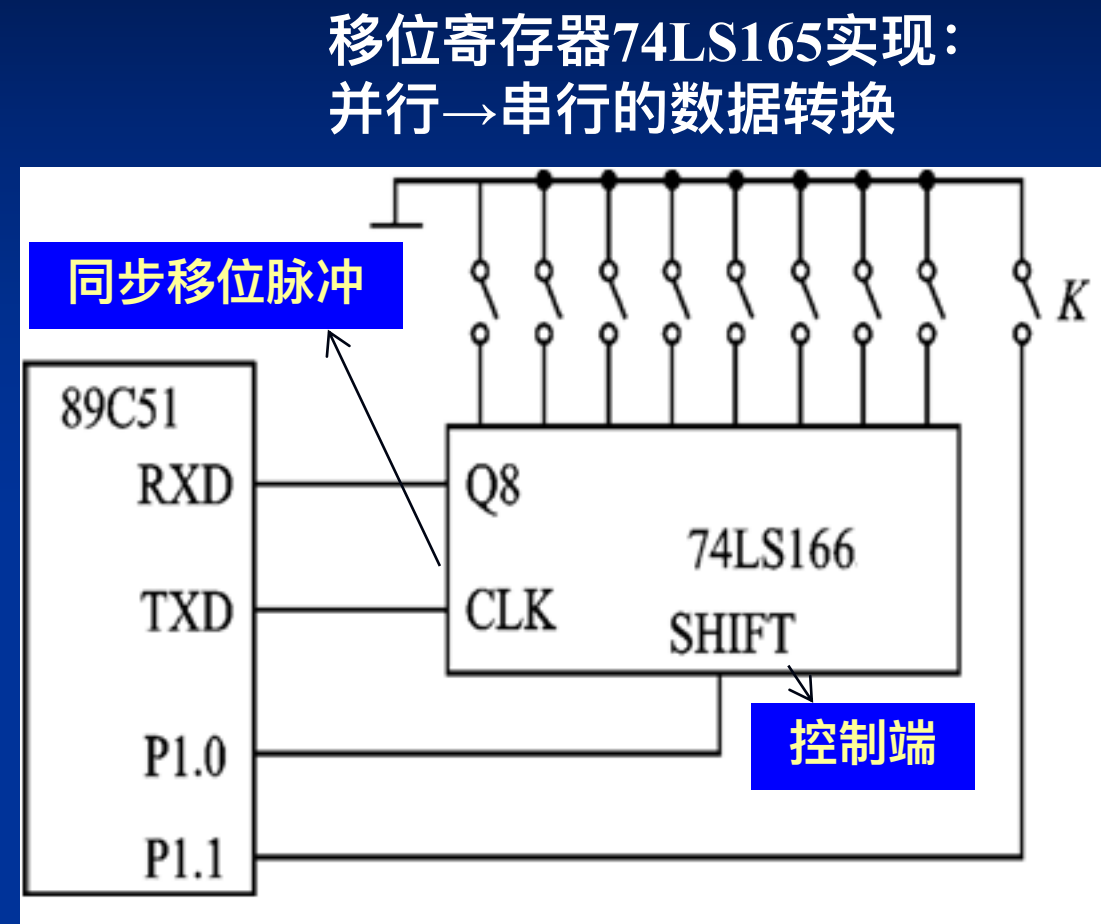
CLR=0，关闭并行输出，
打开串行输入；

CLR=1，打开并行输出，
关闭串行输入。

程序清单：

ORG 0023H	;串行口中断入口
AJMP SBR	;转入串行口中断服务程序
ORG 2000H	;主程序起始地址
MOV SCON ,#00H	;串行口方式0初始化
MOV A ,#80H	;最左一位发光二极管先亮
CLR P1.0	;关闭并行输出
MOV SBUF, A	;开始串行输入
LOOP : SJMP LOOP	;等待中断
SBR : SETB P1.0	;启动并行输出
ACALL DELAY	;显示延迟一段时间
CLR TI	;清发送中断标志
RR A	;准备右边一位显示
CLR P1.0	;关闭并行输出
MOV SBUF , A	;再一次串行输入
RETI	;中断返回

例2：用89C51串行口外加**并入——串**出移位寄存器165或166扩展8位输入，输入数据由8个开关提供，另有一个开关K提供联络信号。当K=0时，表示要求输入数据，输入的8位为开关量，提供逻辑模拟子程序的输入信号。如图所示。



解： 串行口方式0的接收要用SCON寄存器中的**REN**位作为开关来控制。因此，初值化时，除了设置工作方式之外，还要使**REN**位为**1**，其余各位仍然为0。

SHIFT=1，打开并行输入数据，
关闭串行输出；
SHIFT=0，打开串行移位输出，
关闭并行输入。

对RI采用**查询**方式来编写程序，当然，先要查询开关K是否闭合。

程序清单：

START:	MOV	SCON , #10H	;	串行口方式0初始化
	JB	P1.1 , \$;	开关K未闭合，等待
	SETB	P1.0	;	P/S=1,并行置入数据
	CLR	P1.0	;	P/S=0,开始串行移
位				
	JNB	RI , \$;	查询RI
	CLR	RI	;	查询结束，清RI
	MOV	A , SBUF	;	读数据到累加器
	ACALL	LOGSIM	;	进行逻辑模拟
	SJMP	START	;	准备下一次模拟

- 用方式0外加移位寄存器来扩展8位**输出口**时，要求移位寄存器带有输出控制，否则串行移位过程也会反映到并行输出口；另外，输出口最好再接一个寄存器或锁存器，以免在输出门关闭时（CLR=0）输出又发生变化。
- 用方式0加上并入——串出移位寄存器可扩展一个8位并行**输入口**。移位寄存器必须带有预置/移位的控制端，由单片机的一个输出端子加以控制，以实现先由8位输入口置数到移位寄存器，然后再串行移位从单片机的串行口输入到接收缓冲器，最后再读入到CPU中。

2. 串行口方式1的发送和接收

例1： MCS-51串行口按双工方式收发ASCII字符，最高位用来作奇校验位，要求传送的波特率为1200b/s。编写有关的通信程序。

解： 7位ASCII码加1位奇校验共8位数据，故可采用串行口方式1。

MCS-51单片机的奇偶校验位P是当累加器A中1的数目为奇数时， $P=1$ 。如果直接把P的值放入ASCII码的最高位，恰好成了偶校验，与要求不符。因此，要把P的值取反以后放入ASCII码最高位，才是要求的奇校验。

- 双工通信要求收、发能同时进行。实际上，收、发操作主要是在**串行接口**进行，CPU只是把数据从接收缓冲器读出和把数据写入发送缓冲器。
- 数据传送用**中断**方式进行，响应中断以后，通过检测是**RI置位**还是**TI置位**来决定CPU是进行发送操作还是接收操作。
- 发送和接收都通过调用子程序来完成。
- 设发送数据区的首地址为20H，接收数据区的首地址为40H，fosc为6MHz，**通过计算可知**：定时器的初装值应为F3H。定时器T1采用工作模式2，可以避免计数溢出后用软件重装定时初值的工作。

程序清单:

主程序

```

MOV    TMOD ,#20H    ; 定时器1设为方式2
MOV    TL1 , #0F3H   ; 定时器初值
MOV    TH1 ,#0F3H    ; 8位重装值
SETB   TR1           ; 启动定时器1
MOV    SCON ,#50H    ; 设置为方式1,
                        ; REN=1

MOV    R0 ,#20H      ; 发送数据区首址
MOV    R1 ,#40H      ; 接收数据区首址
SETB   ES
SETB   EA

ACALL  SOUT           ; 先输出一个字符

SJMP   $             ; 等待中断
    
```

中断服务程序

```

ORG    0023H        ; 串行口中断入口
AJMP   SBR1          ; 转至中断服务程序
ORG    0100H

SBR1:
    JB   RI , SEND   ; RI=1,为接收中断
    ACALL SIN        ; TI=1,为发送中断
    SJMP NEXT        ; 转至统一的出口

SEND:
    ACALL SOUT        ; 调用发送子程序

NEXT:
    RETI             ; 中断返回
    
```

发送子程序

SOUT: CLR RI

MOV A, @R0 ;取发送数据到A

MOV C, P ;奇偶标识赋予C

CPL C ;奇校验

MOV ACC.7, C ; A中的最高
位为奇校验

INC R0 ;修改发送数
据指针

MOV SBUF, A ;发送ASCII码

RET ;返回

接收子程序

SIN: CLR TI

MOV A, SBUF ;读出接收缓冲
区内容

MOV C, P ;取出校验位

JNC ERR ;奇校验错

ANL A, #7FH ;删去校验位

MOV @R1, A ;读入接收缓
冲区

INC R1 ;修改接收数
据指针

RET ;返回

- 在主程序中已初始化REN=1，则允许接收。以上程序基本上具备了全双工通信的能力，但不能说很完善。
- 例如，在接收子程序中，虽然检验了奇偶校验位，但没有进行出错处理；另外，发送和接收数据区的范围都很有限，也不能满足实际需要。但有了一个基本的框架之后，逐渐完善还是可以做到的。

例2：采用查询方式由串行口发送带奇偶校验位的数据块。

解：本例由内部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发出。采用8位数据位异步通信方式，波特率为1200b/s， $f_{osc}=11.059\text{MHz}$ 。

由要求可知，应把串行口设置为方式1，采用T1模式2作为波特率发生器，预置值（TH1）=0E8H。

```
MOV  TMOD ,#20H           ;设置定时器1为模式2
MOV  TL1 ,#0E8H           ;初值，波特率为1200b/s
MOV  TH1 ,#0E8H
SETB TR1                  ;启动T1运行
MOV  SCON ,#01000000B     ;设置串行口为方式1
MOV  R0 ,#20H
MOV  R7 ,#32              ;数据块长度
LOOP: MOV  A ,@R0
      ACALL SP-OUT
      JNB  P ,ERROR        ;传输出错处理，由SP-OUT中
                           ;“CPL C” 结果决定
      INC  R0
      DJNZ R7 ,LOOP
      .....
```

串行口发送子程序（奇校验）；

SP-OUT: MOV C,PSW.0 ;设置奇校验位，校验位P=1

 为奇校验

 CPL C ;奇校验（无此指令位偶校验）

 MOV ACC.7,C ;ACC.7补0或1

 MOV SBUF,A ;启动串行口发送过程

 JNB TI,\$

 CLR TI ;清TI标志，允许再发送

 RET

ERROR: (略)

例3：由串行口接收带奇偶校验位的数据块。

解：采用查询方式，本例与上例相呼应，接收器把接收到的32B数据存放在20H-3FH单元内，波特率同上，若奇偶校验出错则置进位位为1。

程序清单：主程序；

MOV SCON ,#01010000B	;设串口方式1，允许接收
MOV TMOD ,#20H	;设置定时器T1为模式2
MOV TL1 ,#0E8H	;初值，波特率为1200b/s
MOV TH1 ,#0E8H	
SETB TR1	;启动T1运行
MOV R0 ,#20H	
MOV R7 ,#32	;数据块长度
LOOP:	
ACALL SP-IN	;调接收一帧子程序
JC ERROR	;由SP-IN中“CPL C”结果决定
MOV @R0 ,A	;存放接收的数据
INC R0	
DJNZ R7 ,LOOP	

.....

接收一帧子程序;

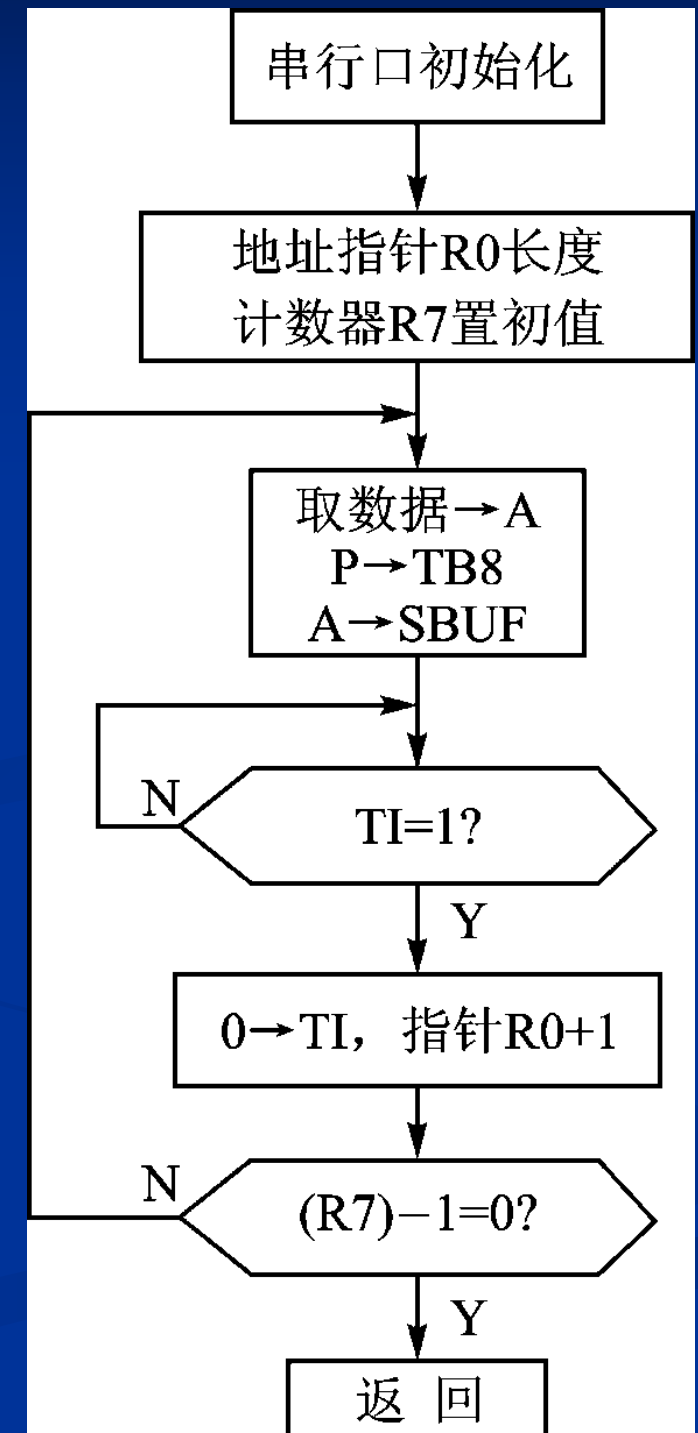
```
SP-IN:      JNB RI,$           ;RI由硬件置位
             CLR RI           ;软件清除RI
             MOV A,SBUF
             MOV C,P           ;检查奇校验位
             CPL C             ;置C为主程序“JC ERROR”用
             ANL A,#7FH        ;去掉奇校验位
             RET
ERROR:      (略)
```

3. 串行口方式2、方式3的发送和接收

- 串行口方式 2与方式 3基本一样（只是波特率设置不同），接收 / 发送 11位信息：开始为1位起始位（0），中间8位数据位，数据位之后为1位程控位（由用户置SCON的TB8决定），最后是1位停止位（1）。只比方式1多了一位程控位。

例1：用第9个数据位作奇偶校验位，编制串行口方式2的发送程序。

解：设计一个发送程序，将片内RAM 50H~5FH中的数据串行发送；串行口设定为方式2状态，TB8作奇偶校验位。在数据写入发送缓冲器之前，先将数据的奇偶位P写入TB8，这时，第9位数据作奇偶校验用。



程序清单如下:

```

TRT:  MOV SCON , #80H      ;方式2设定
      MOV PCON , #80H      ;取波特率为fosc/32
      .....
      MOV R7 , #10H        ;数据长度10H→R7
      MOV R0 , #50H        ;数据块首地址
LOOP: MOV A , @R0          ;取数据→A
      MOV C , PSW.0        ;P→TB8
      MOV TB8 , C          ;注意发送顺序
      MOV SBUF , A         ;数据→SBUF,启动发送
WAIT: JBC  TI , CONT       ;判断发送中断标志
      SJMP WAIT
CONT:  INC  R0
      DJNZ R7 , LOOP
      RET
    
```

例2：编制一个串行口方式2接收程序，并核对奇偶校验位。

解：根据上面介绍的特点，在方式2、方式3的发送过程中，将数据和附加在TB8中的奇偶位一块发向对方。因此，作为接收的一方应设法取出该奇偶位进行核对，相应的接收程序段为：

```

.....
RRR:  MOV SCON ,#90H      ; 选方式2, 并允许接收 (REN=1)
LOOP: JBC RI ,RECEIV      ; 等待接收数据并清RI
      SJMP LOOP
RECEIV: MOV A ,SBUF        ; 将接收到的字符取出后, 送到ACC。注意,
                          ; 传送指令影响PSW, 产生接收端的奇偶值
      JB PSW.0 ,ONE        ; 判断接收端的奇偶值
      JB RB8 ,ERR          ; 判断发送端的奇偶值
      SJMP RIGHT
ONE:   JB RB8 ,RIGHT
      SJMP ERR
RIGHT: .....             ; 接收正确
      .....
ERR:   .....             ; 接收有错

```

- 当接收到一个字符时, 从SBUF转移到ACC中时会产生接收端的奇偶值, 而保存在RB8中的值为发送端的奇偶值, 两个奇偶值应相等, 否则接收字符有错。发现错误要及时通知对方重发。

例3：编制一个发送程序，将片内RAM中50H—5FH的数据串行发送。串行口设定为工作方式2，TB8作奇偶校验位。

解：在数据写入发送SBUF之前，先将数据的奇偶标志P写入TB8，此时，第9位数据便可作奇偶校验用。可采用查询和中断两种方式发送。

(1) 采用查询方式的程序段

```

    ORG 0000H
    AJMP MAIN      ;上电，转向主程序
    ORG 0100H      ;主程序
MAIN:  MOV SCON ,#80H ;设工作方式2
       MOV PCON ,#80H ;取波特率为fosc/32
       .....
       MOV R0 ,#50H  ;首址50H送R0
       MOV R7 ,#10H  ;数值长度送R7
LOOP:  MOV A ,@R0    ;取数据
       MOV C ,PSW.0  ;P→C
       MOV TB8 ,C    ;奇偶标志送TB8
       MOV SBUF ,A   ;发送数据
WAIT:  JBC TI ,CONT
       AJMP WAIT     ;等待中断标志TI=1
CONT:  INC R0
       DJNZ R7 ,LOOP  ;数值尚未发送完， 继续发送下一个数据
       SJMP $

```


(2) 采用中断方式的程序段

```

ORG 0000H
AJMP MAIN ;上电, 转向主程序
ORG 0023H ;串行口的中断入口地址
AJMP SERVE ;转向中断服务程序
ORG 0100H ;主程序
    
```

MAIN:

```

MOV SCON, #80H
MOV PCON, #80H
.....
MOV R0, #50H
MOV R7, #0FH
SETB ES ;允许串行口中断
SETB EA ;CPU允许中断
MOV A, @R0
MOV C, PSW.0 ;P→C
MOV TB8, C
MOV SBUF, A ;发送第一个数据
SJMP $
    
```

SERVE:

```

CLR TI ;清除发送中断标志
INC R0 ;修改数据地址
MOV A, @R0
MOV C, PSW.0 ;P→C
MOV TB8, C
MOV SBUF, A ;发送数据
DJNZ R7, ENDT ;判断数据块发送完否
; 若未发 送完, 则转ENDT
CLR ES ;若发送完, 则禁止串行口中断
ENDT:
RETJ ;中断返回
END
    
```

7.4 多机通信

- 掌握多机通信的原理及实现过程，能设计简单的多机通信系统及程序。
- 本节难点：SM2、TB8、RB8在通信过程中的作用

多机通信的应用领域



典型应用：校园卡系统、智能楼宇、自动温控蔬菜大棚、大型粮库、工厂自动化；

7.4.1 多机通信原理

利用单片机串行口工作于方式2、方式3可实现多机通信。
通信是以主机发送信息，从机接收信息开始。

问题1：多机通信时主机如何寻找从机？

问题2：如何区分地址和数据信息？

串行口控制寄存器SCON中的SM2为多机通信接口控制位。
串行口工作于方式2或3。

SM2 — — — — — 多机通信位；

TB8/RB8 — — — — 地址/数据区分位；

=1：地址 / =0：数据

主机：SM2 = 0

从机：

■ 若SM2 = 1：

则仅当接收到的第9位数据RB8为1时，数据才装入SBUF，置位RI，请求CPU对数据进行处理；

如果接收到的第9位数据RB8为0，则不产生中断标志RI，信息丢失，CPU不做任何处理。

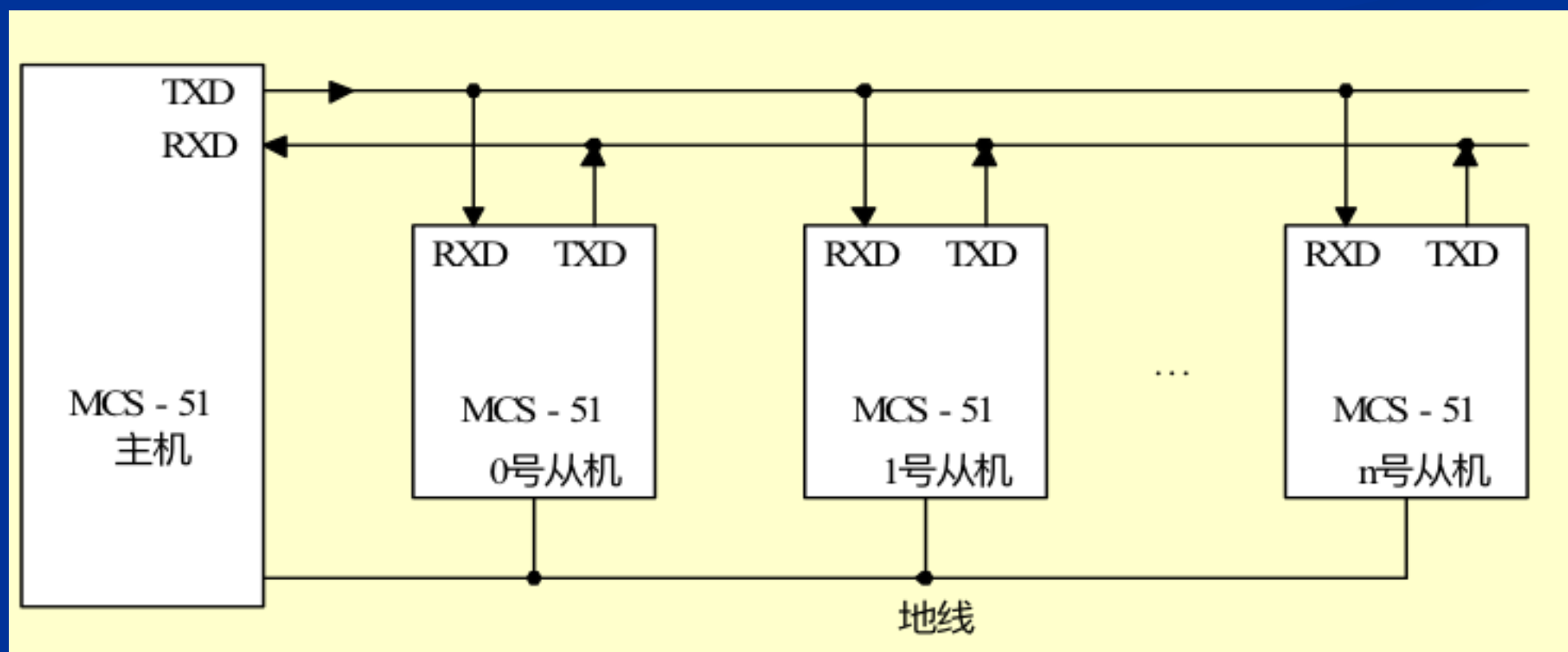
■ 当SM2 = 0时：

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并置位中断标志RI，请求CPU处理。

关键是SM2与TB8/RB8的关系！

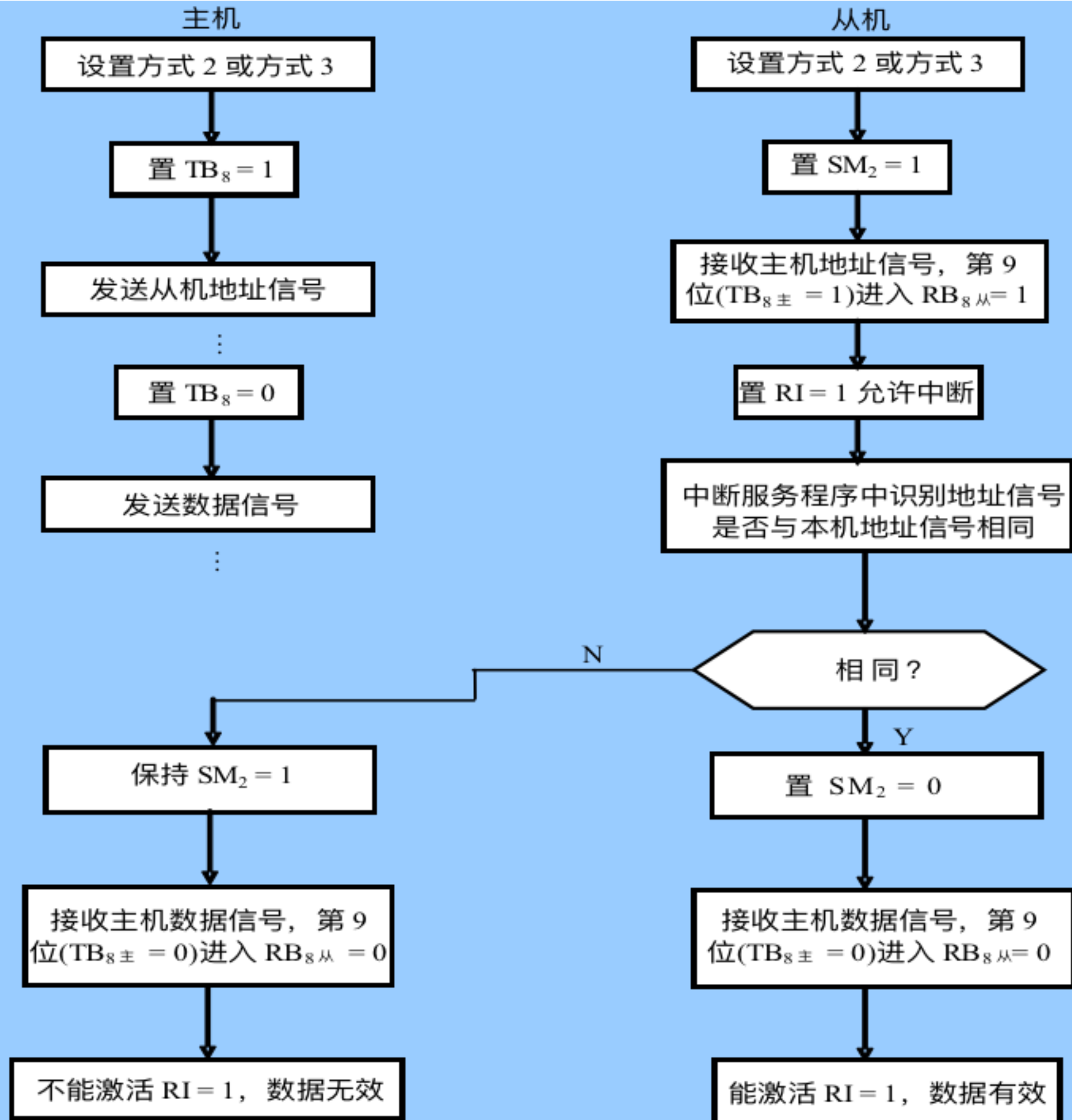
SM2、RB8与从机动作关系

SM2	RB8	从机动作
1	0	此时不能接收数据
1	1	能收到主机发的数据（地址信息）
0	0	进入串口中断，对接收的数据进行处理
0	1	进入串口中断，对接收的数据进行处理



通信过程

- ▶ 所有从机的SM2=1，处于只接收地址帧状态
所有同学坐着听讲
- ▶ 主机发地址信息，以第9位TB8=1表示发送的是地址
老师喊同学名字
- ▶ 从机接收到地址帧后与本机地址比较
所有同学听到名字与自己的名字比较
- ▶ 被寻址从机SM2=0，其它从机SM2=1不变
名字相符的同学站起来
- ▶ 主机发数据信息，对已被寻址的从机因SM2=0，可以接收主机发来的信息。其余从机因SM2=1不理睬主机。
老师与站起的同学进行提问回答课堂交流
- ▶ 被寻址的从机SM2置1.主机可另发地址帧与其它从机通信时。
回答问题的同学坐下



7.4.2 多机通信系统设计

设多机通信系统主、从机晶振为6MHz，波特率为2400bps，以方式3进行多机串行通信。每次通信时主机先发送对应的从机地址，然后发一个字节数据。



图1 多机通信系统硬件结构

主机软件设计----初始化子程序

```
INIT:  MOV TMOD, #20H
        MOV TH1, #0F9H;   波特率为2400bps
        MOV TL1, #0F9H
        MOV SCON, #0D0H;  方式3, SM2=0,
                           允许接收
        SETB  TR1
```

由于主机要发送地址信息，所以设置TB8=1

主机软件设计-----通信子程序流程

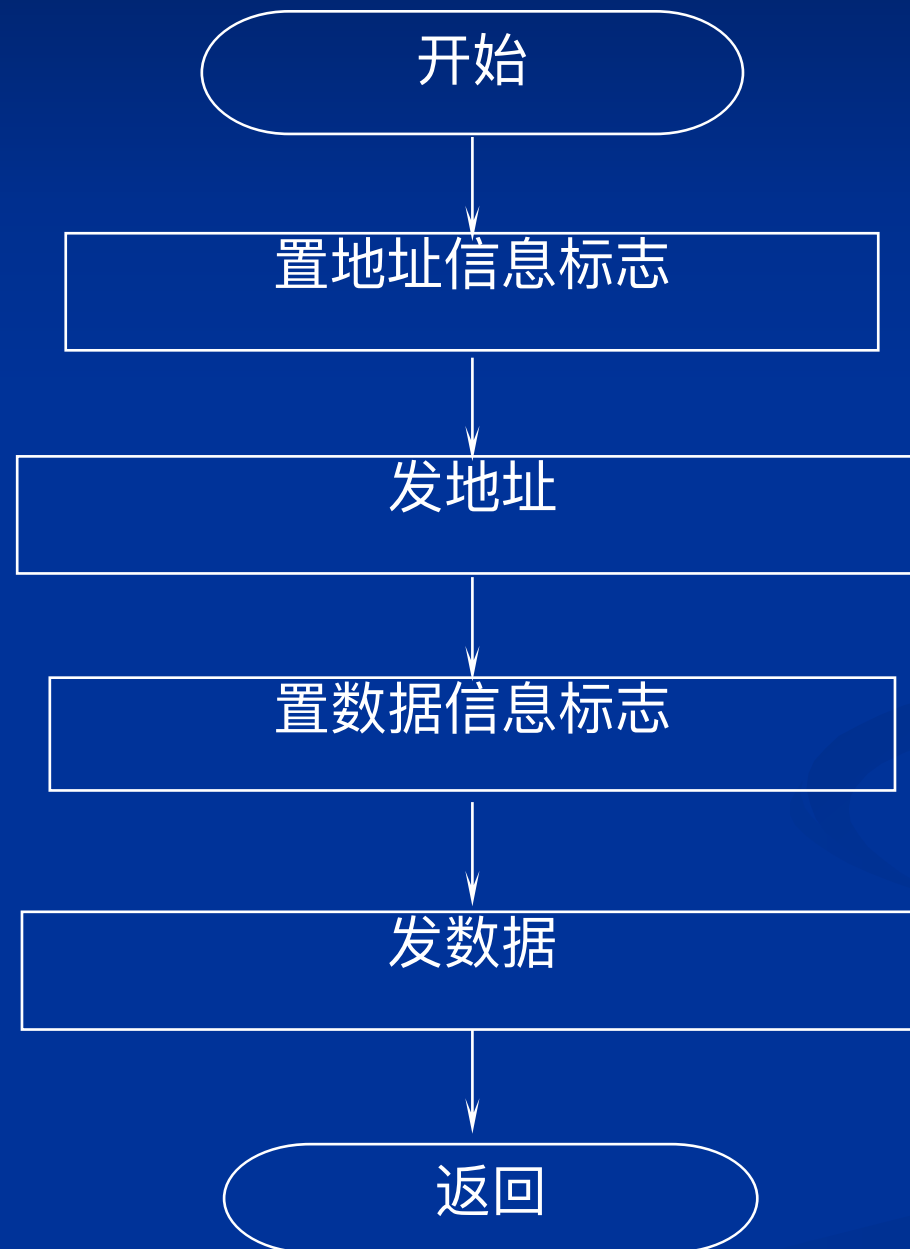


图2 主机通信子程序流程图

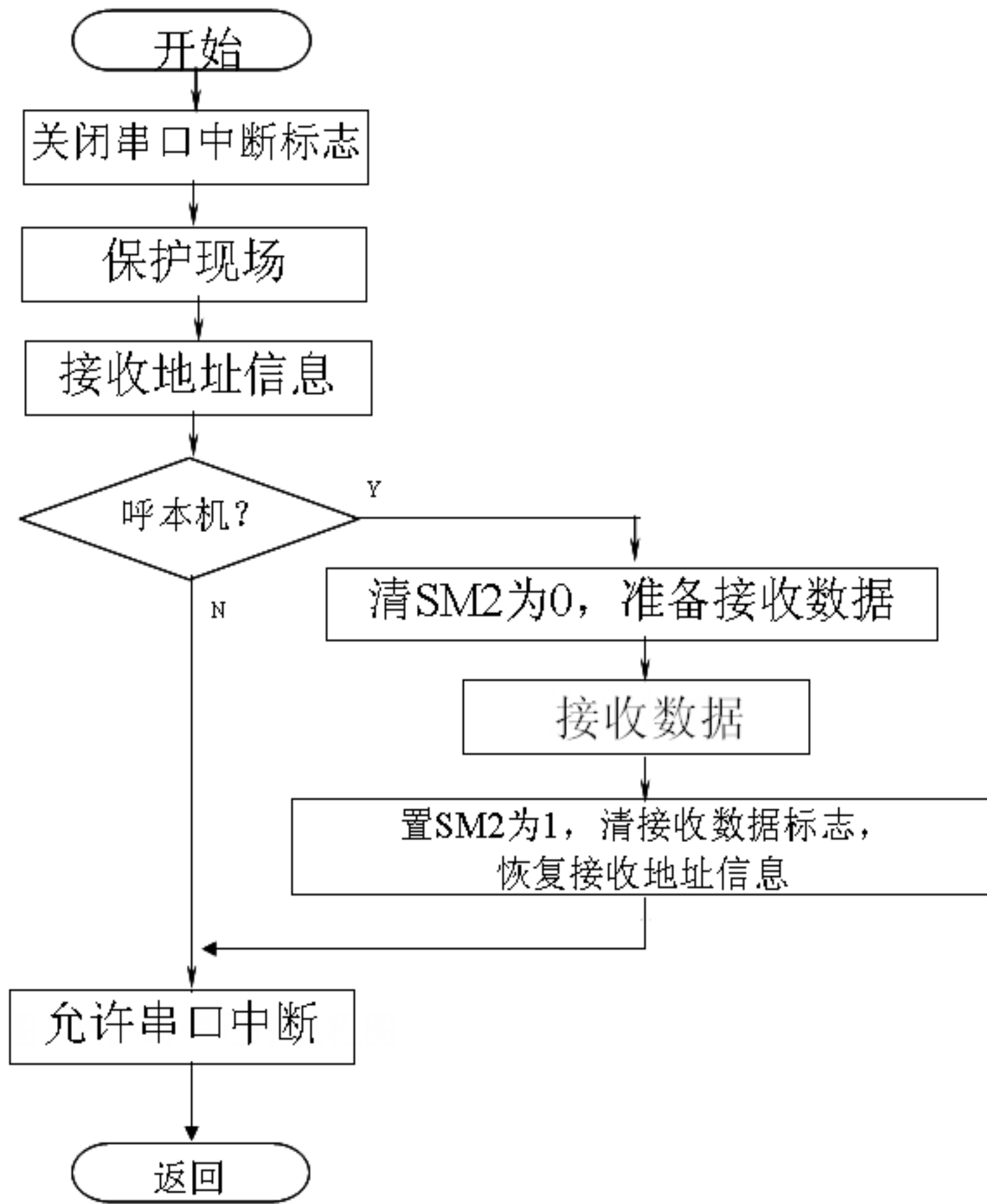
从机软件设计-----初始化程序

```
INIT: MOV  TMOD, #20H
      MOV  TH1, #0F9H    ;设置波特率为2400
      MOV  TL1, #0F9H
      MOV  SCON, #0F0H   ;方式3, SM2=1, 多机通讯
      SETB TR1
      SETB ES            ;允许串行口中断
      SETB EA
WAIT: SJMP WAIT          ;听号
```

当主机向各从机发送地址帧时,

从机RB8=1, RI=1, 各从机响应中断。

从机通信子程序



从机串行中断通信程序

```
SIO:  CLR    RI
      PUSH  PSW          ; 现场保护
      PUSH  ACC
      MOV   A, SBUF      ; 接收串行数据
      XRL   A, #ADDR     ; 判断是否与本站地址相符
      JNZ   BACK        ; 不相符则返回
      CLR   SM2          ; 相符则SM2为0
      CLR   ES           ; 关串口中断, 避免中断嵌套
      MOV   A, SBUF      ; 接收数据
WAIT: JNB   RI, WAIT
      CLR   RI
      MOV   @R0, A       ; 保存数据
      SETB  SM2          ; 恢复多机通信,
      SETB  ES           ; 开串口中断
BACK: POP   ACC          ; 恢复现场
      POP   PSW
      RETI
```

**本章结束，
谢谢！**