# 第5章 定时/计数器及串行通信应用

## 5.1 定时/计数器

## 5.2 单片机串行通信

## 问题的提出：

1、在微机应用系统中，普遍用到定时/计数的功能，如对工业过程量信号的定时采样、对流水生产工艺中某一产品的计数（或计件），因此，定时/计数具有广泛的应用背景。

2、在前面的程序设计一章，大家已学习了软件延时（即通过执行一个循环程序进行时间的延迟）实现定时功能，但这种定时具有明显的缺点：一是循环程序执行占用CPU时间，降低CPU的利用率；二是定时时间不精确或计算比较麻烦。

因此，针对应用广泛的定时/计数功能，必须在CPU中用硬件来实现定时/计数功能。

**问题的解决：**

由于定时/计数功能的普遍性应用，特别是在长时间定时或定时精度要求较高的场合，通常选用硬件实现定时，采用硬件实现定时，我们希望：

1、定时/计数的功能及工作方式是可编程的。即可选它作定时器，也可作计数器，同时，可选择几种工作方式。

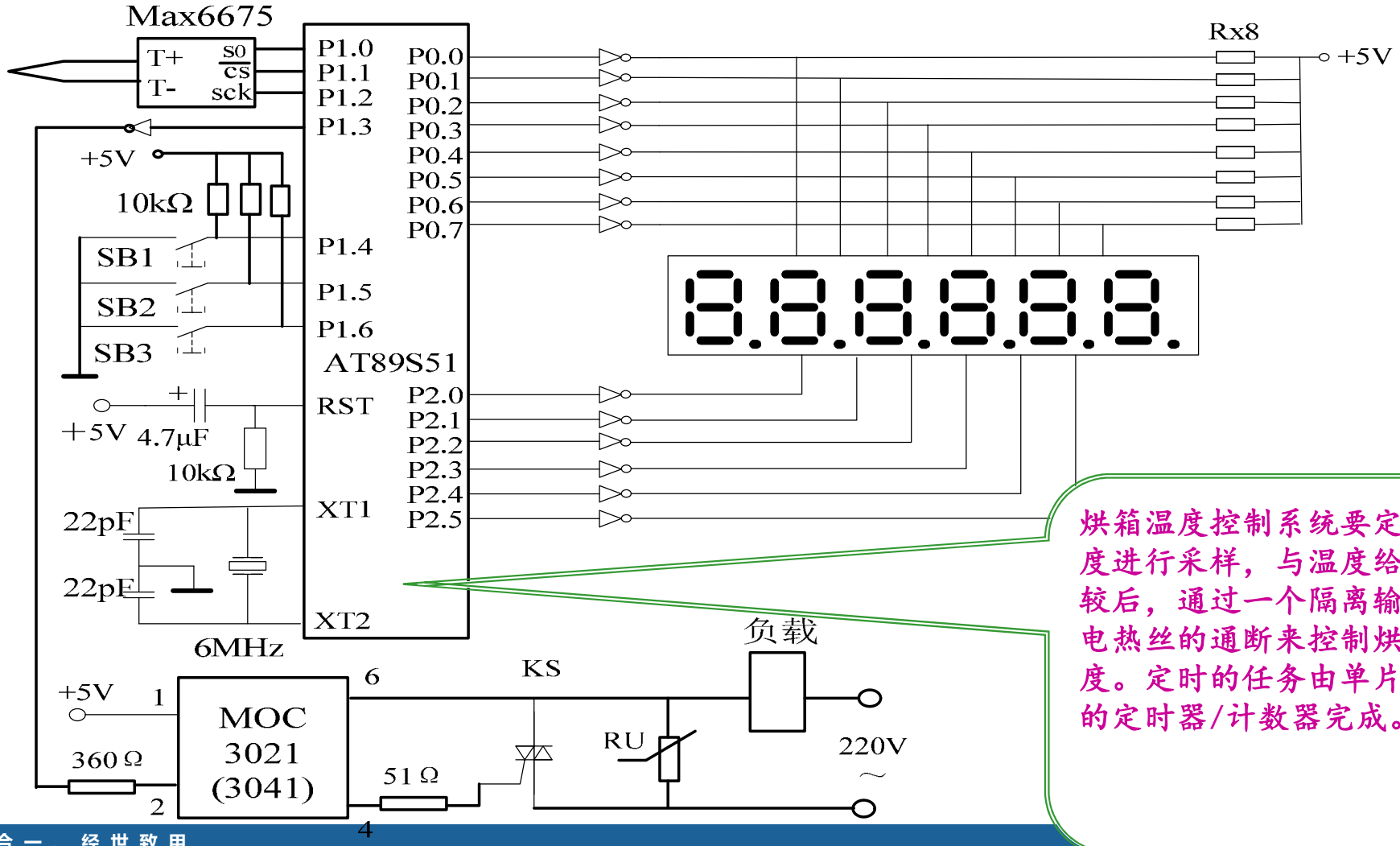
2、定时/计数值在规定的范围内是可编程的。即根据需要，通过软件来设定定时值或计数值。

3、当达到定时/计数的设定值时，应向CPU申请中断，以便实现定时/计数控制。

## 5.1 定时/计数器

- 51系列单片机内部提供2个定时/计数器T0和T1（AT89S系列有3个），既可用作定时器，也可用作计数器，还可作为串行口的波特率发生器。定时/计数器实现软、硬件结合，给应用系统的设计带来很多方便之处。
- 主要作用：对外部脉冲计数、产生精确定时时间、作串行口的波特率发生器。

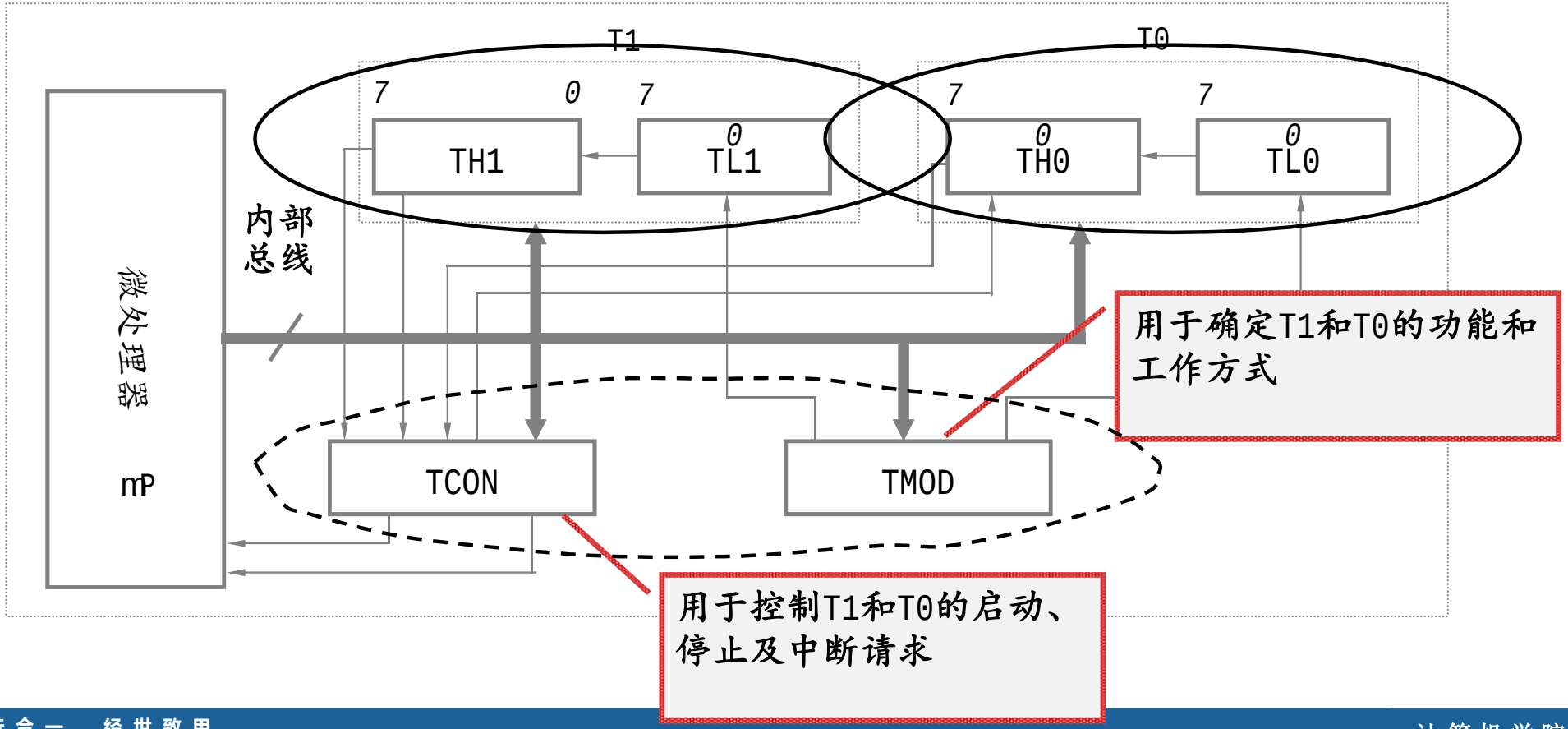
## 烘箱温度控制系统




烘箱温度控制系统要定时对温度进行采样，与温度给定值比较后，通过一个隔离输出控制电热丝的通断来控制烘箱内温度。定时的任务由单片机片内的定时器/计数器完成。

## 5.1.1定时/计数器的定时和计数功能

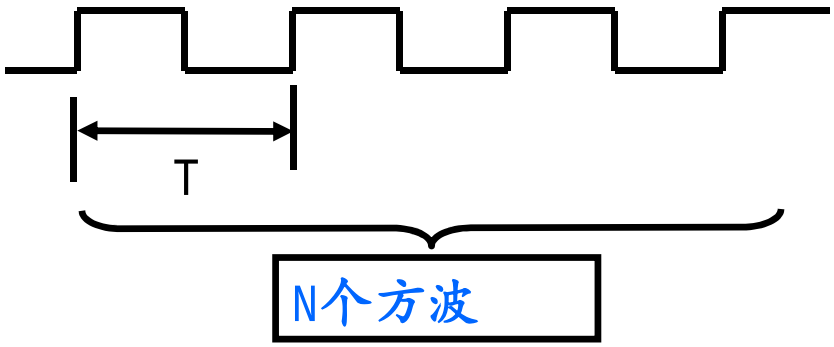
### 1、定时器/计数器T0、T1的结构



## 2、定时与计数的概念

定时：指采用具有精确的固定周期的脉冲信号进行计数，一般是利用内部振荡脉冲，即机器周期：fosc / 12（= 1 / T）作为计数脉冲；

计数：指对外部输入脉冲进行计数；




波形等间隔，次数已定，时间确定
即对机器周期T进行计数。

左图定时时间为 N*T

计数：脉冲不等间隔。

每个下降沿计数一次
确认一次负跳变需两个机器周期，
所以，计数频率最高为fosc / 24。

## 5.1.2 定时/计数器的控制

### 1. 工作方式控制寄存器TMOD

- 用于控制定时器/计数器的功能和工作方式
- TMOD格式：（字节地址89H）

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| GATE | $C/\overline{T}$ | M1 | M0 | GATE | $C/\overline{T}$ | M1 | M0 |
| 定时器1方式字段 | | | | 定时器0方式字段 | | | |

操作模式控制位

GATE：门控制

- 定时方式时，每个机器周期使计数器加1

| M1M0 | 方式 | 计 数 器 配 置 |
|---|---|---|
| 0 0 | 0 | TLx低5位与THx的8位构成13位计数器 |
| 0 1 | 1 | TLx与THx构成16位计数器 |
| 1 0 | 2 | 自动重装初值(THx)的8位(TLx溢出时)计数器 |
| 1 1 | 3 | 仅用于T0，分成两个8位计数器，T1停止计数 |

- C/ $\overline{T}$= 1计数方式；

## 2、状态控制寄存器TCON

TCON格式： （字节地址88H）

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |

定时器1控制位

定时器0控制位

TF1:定时器T1溢出中断标志。

- 当定时器1溢出时，由硬件置1；
- 当响应中断转向中断服务程序时由硬件清0。

TR1:定时器T1运行控制位。

- 由软件置位/复位，控制定时器是否运行。

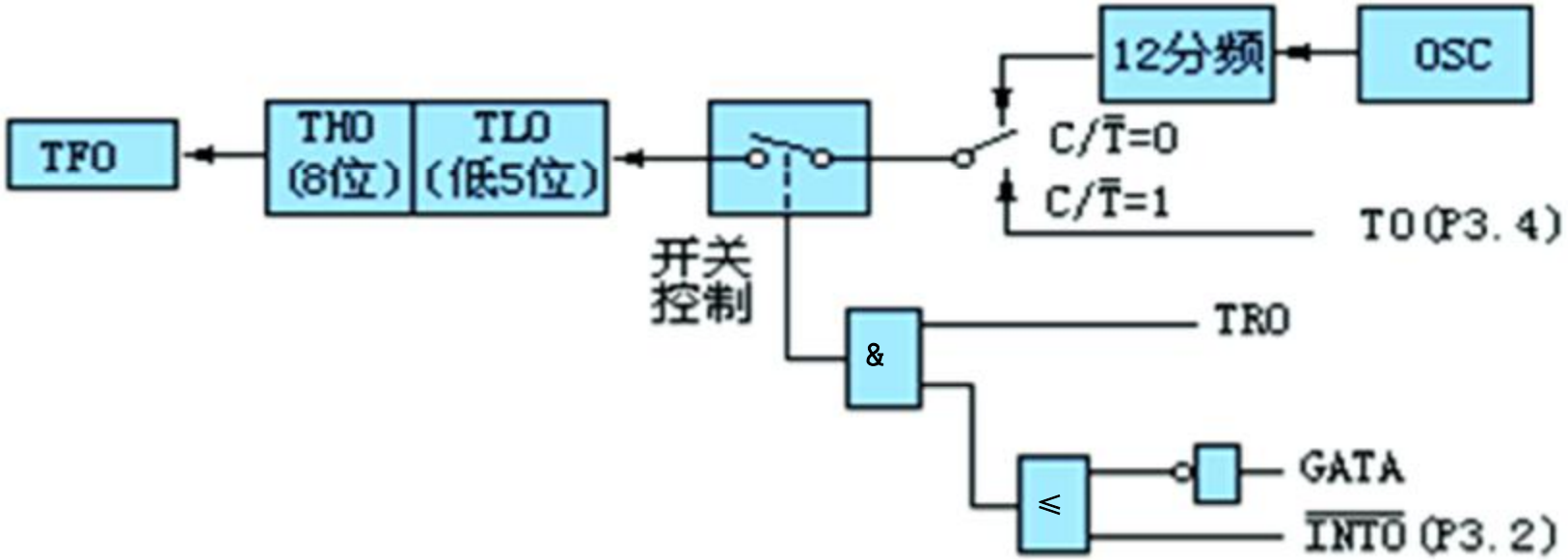
TF0:功能与TF1类似。

TR0:功能与TR1类似。

## 5.1.3 定时/计数器的工作方式

### 1、方式0

T0的等效逻辑结构




当将定时/计数器T0设置为方式0时，定时/计数器的计数范围是：

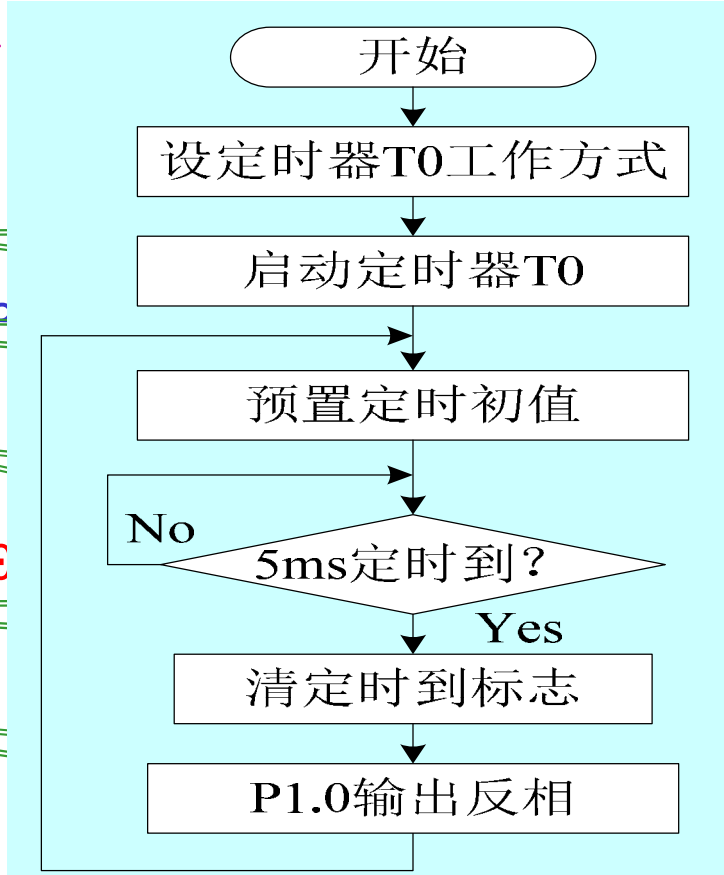
（1～8192）次方晶振周期×12

例1：选用T0操作模式0，用于定时，由P1.0输出周期为10ms的方波。设晶振fosc=6MHz。

编程思路：P1.0输出周期为10ms宽的方波，只要每

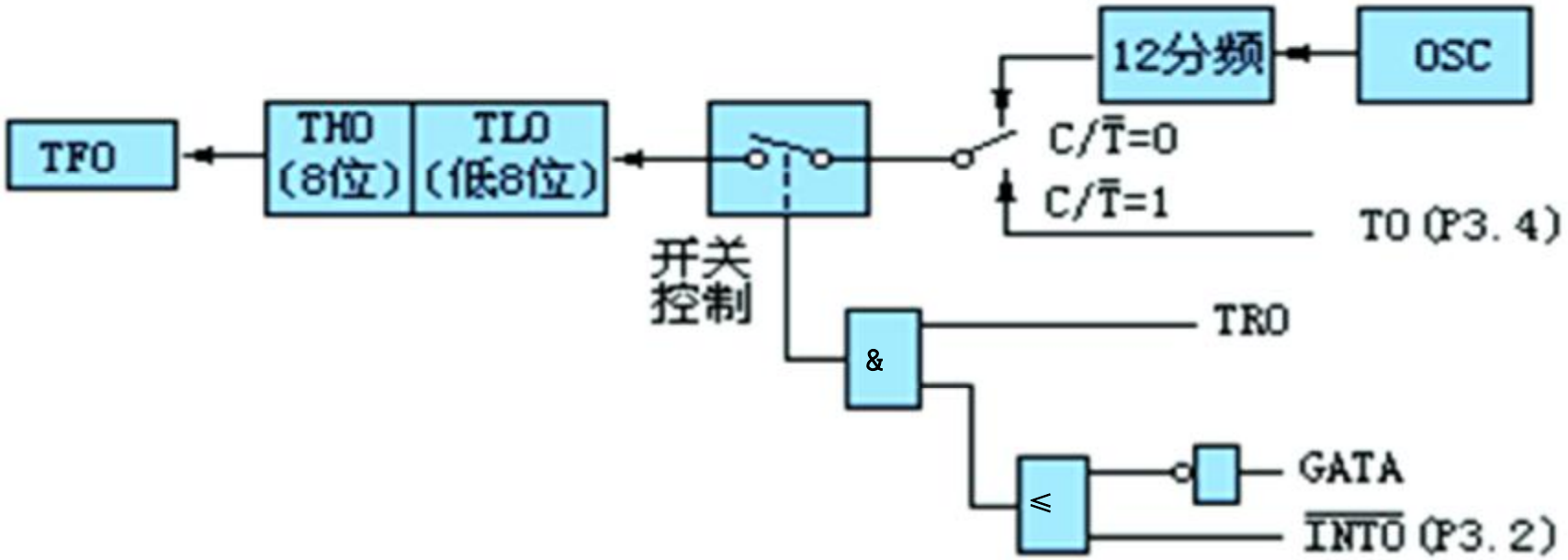
初值为：

```
        ORG   0100H
        MOV   TMOD,#00H
        SETB   TR0
LP1:    MOV   TL0,  #1CH
        MOV   TH0,  #0B1H
LP2:    JBC    TF0, LP3
        AJMP   LP2
LP3:    CPL    P1.0
        SJMP   LP1
```



## 2 、工作方式1

T0的等效逻辑结构




当为定时工作方式1时，计数器的计数范围是为：

(65536~65536-初值)×12个晶振周期

例2：用定时器T1产生一个50Hz的方波，由P1.1输出，仍用程序查询方式，fosc=12MHz。

编程思路：方波周期T=1/50=0.02s=20ms，用T1定时10ms。

$X=2^{16}-10\times10^{3}\times12/12=65536-10000=55536=D8F0H$

```
        ORG   0100H
        MOV   TMOD, #10H
        SETB  TR1
LOOP:   MOV   TH1, #0D8H
        MOV   TL1, #0F0H
        JNB   TF1, $
        CLR   TF1
        CPL   P1.1
        SJMP  LOOP
```

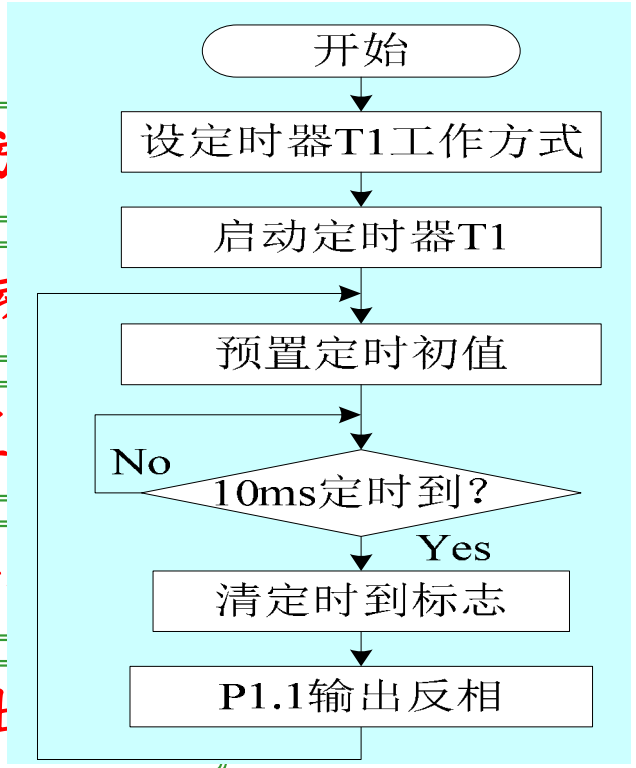
T1方

启动

重置

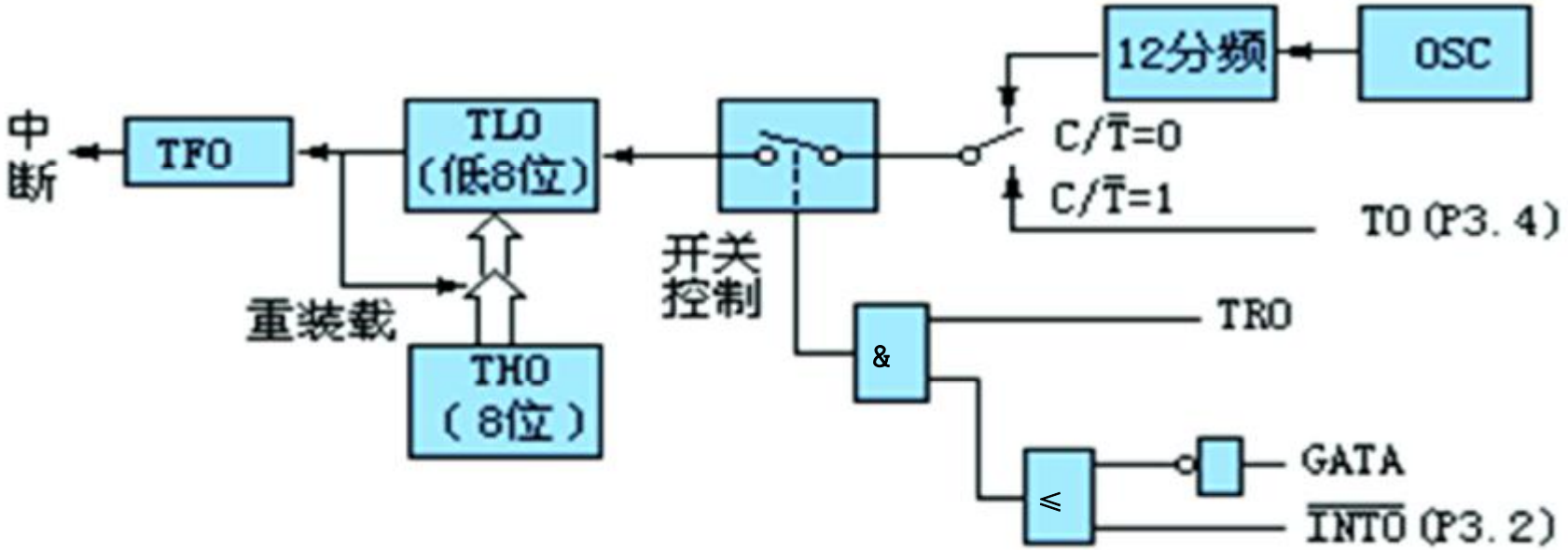
等

输出



## 3、工作方式2

T0的等效逻辑结构




方式2为能自动重置计数器初值的工作方式，TL0为8位计数器，TH0为计数器初值暂存器。

定时时间 T=(256-计数初值)× 晶振周期×12

例3：用定时器1，模式2计数，要求每计满3次，将P1.0端取反。

编程思路：T1工作于计数方式，外部计数脉冲由T1（P3.0）引脚引入，每来一个脉冲，TF1的状态。

计数初值：$X=2^8-3=253$

```
        ORG   0100H
        MOV   TMOD, #60H
        MOV   TH1, #253
        MOV   TL1, #253
        SETB   TR1
LOOP:   JBC    TF1, REP
        SJMP   LOOP
REP:    CPL    P1.0
        SJMP   LOOP
```

T1方式2计数

送初值

启动T1

等待T1计数到

取反

方式2与方式0、1的区别：

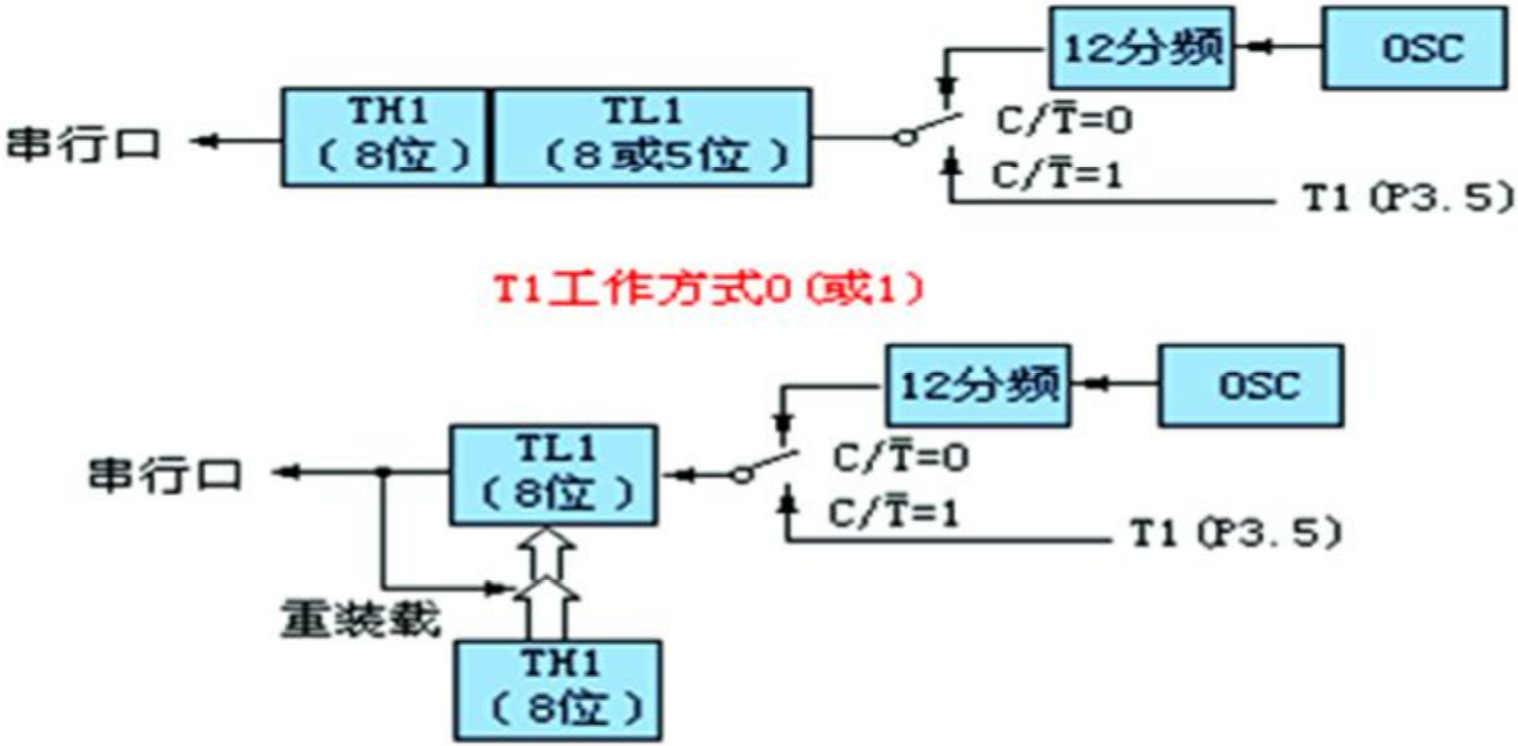
方式0方式1： 计数溢出后，计数器为全0，因而循环定时或循环计数应用时就存在反复设置初值的问题，这给程序设计带来许多不便，同时也会影响计时精度。

方式2： 具有自动重装载功能，即自动加载计数初值。16位计数器分为两部分，TL0为计数器，TH0作为预置寄存器。当计数溢出时，由预置寄存器TH0以硬件方法自动给计数器TL0重新加载。

方式2在串口通讯时，常用作波特率发生器。

## 4、工作方式3

只有T0能工作于方式3，此时T1的的一些控制位已被T0借用，只能工作在方式0、方式1或方式2下，等效电路参见下图：



## 5 、定时/计数器的初始化和初值确定

为使定时/计数器正常工作，首先必须对定时/计数器进行初始化，然后再开启定时或计数。定时计数器的初始化包括以下内容。

（1）确定工作方式——对TMOD赋值；

如：MOV TMOD,#06H； 设定T0为计数器工作方式。

（2）预置定时计数器中计数的初值——直接写入TH和TL；

```
如：   MOV   TH0,#00H        ；设定计数初值。
       MOV   TL0,#00H
```

（3）根据需要开放定时/计数器的中断——对IE位赋值；

（4）启动定时器/计数器；

```
如：   SETB  TR0
```

**初值的计算方法：**

X=M—计数值
M是定时器的最大计数值。视工作方式不同而不同。

**工作方式0：** 13位定时/计数方式，因此，最多可以计到2的13次方，也就是8192次。

**工作方式1：** 16位定时/计数方式，因此，最多可以计到2的16次方，也就是65536次。

**工作方式2和工作方式3：** 都是8位的定时/计数方式，因此，最多可以计到2的8次方，也说是256次。

预置值计算： 用最大计数量减去需要的计数次数即可。

通过上面的任务，我们掌握了计数程序的编制方法，下面我们再看看定时程序怎样编制。

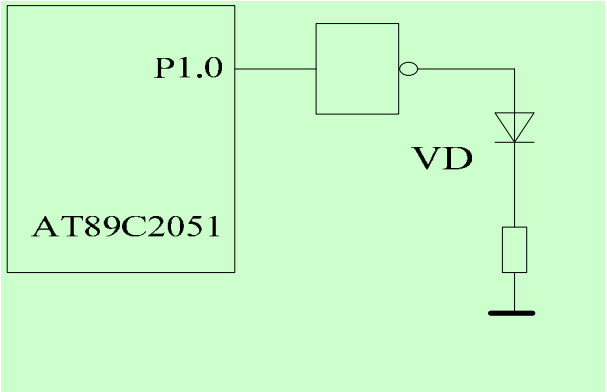
首先我们看一下下面的程序段。

```
MOV     TMOD,#01H
MOV     TL0,#00H
MOV     TH0,#4CH
SETB      TR0
```

以上程序是一个定时初始化程序段，计数方式和它类似。

## 5.1.4 应用举例

例1：已知focs=12MHZ，利用定时器使图中发光二极管VD进行秒闪烁，要求亮50ms，暗50ms，采用中断编程。




编程思路： T0定时每50ms中断一次，中断程序使P1.0反相一次。

初值为：

$X=2^{16}-t\times fosc/12=65536-12\times 50\times 1000/12=15536=3CB0H$

```
        ORG   0000H
        SJMP   START
        ORG   000BH
        LJMP   TIME0
START: MOV   TMOD,#01H
        MOV    TH0, #3CH
        MOV    TL0, #0B0H
        SETB    TR0
        MOV    IE,  #82H
WAIT:  SJMP   WAIT
TIME0: MOV    TH0, #3CH
        MOV    TL0, #0B0H
        CPL     P1.0
        RETI
```

主程序入口

中断程序入口

开放中断与定时器

主程序其他任务

重置初值

改变输出状态

中断返回

例2： 用定时器T1完成时钟秒、分、时的定时。已知晶体振荡频率为12MHz。

分析： 由于机器周期 $T = 1\mu S$ ，采用方式1能定时的最长时间也只有65.536ms。要得到长时间的定时，必须采用软件计数器。例如定时器定时50ms，对50ms计数20次为1秒，对1秒计数60次为1分， 对1分计数60次为1小时。

50ms计数次数＝50000/1＝50000次

定时初值＝65536－50000＝15536

秒计数初值（50H单元）＝20

分计数初值（51H单元）＝60

时计数初值（52H单元）＝60

程序：

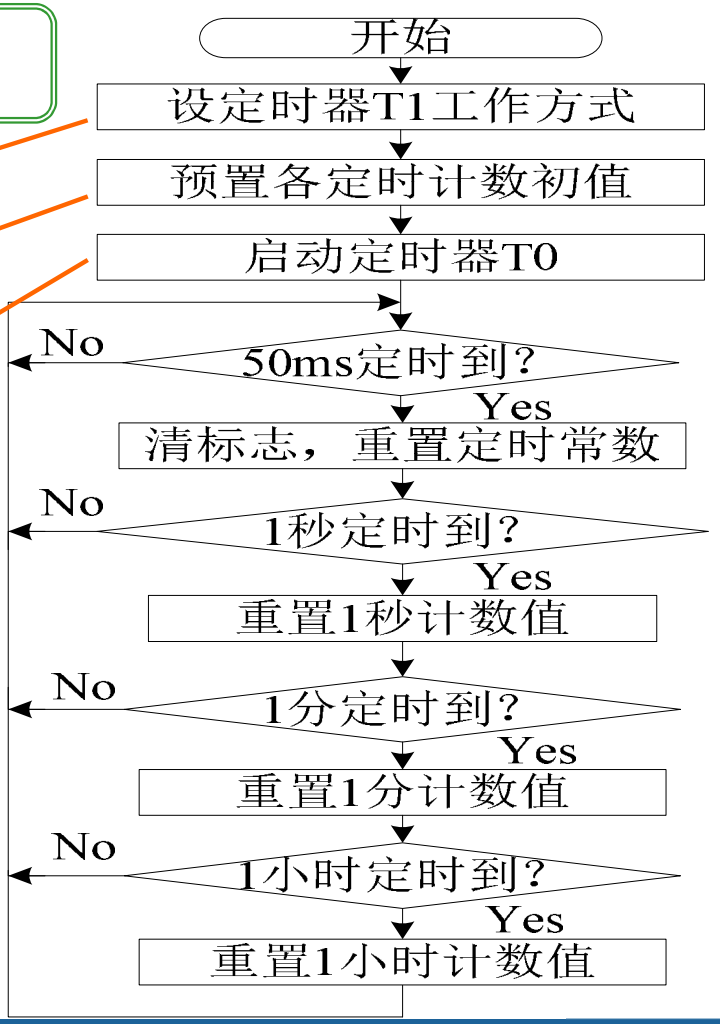
```
ORG   0000H
MOV   TMOD,#10H
MOV   TH0,#15536 MOD 256
MOV   TL0,#15536/256
MOV   50H,#20
MOV   51H,#60
MOV   52H,#60
SETB  TR1
```

方式1定时

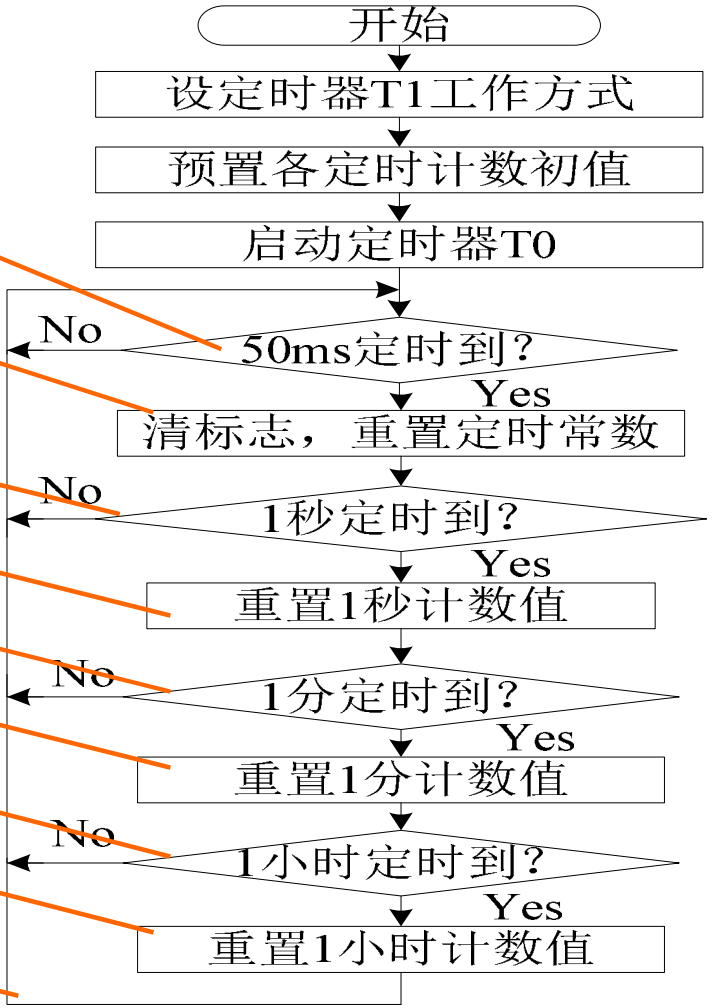
时间常数

软计数常数

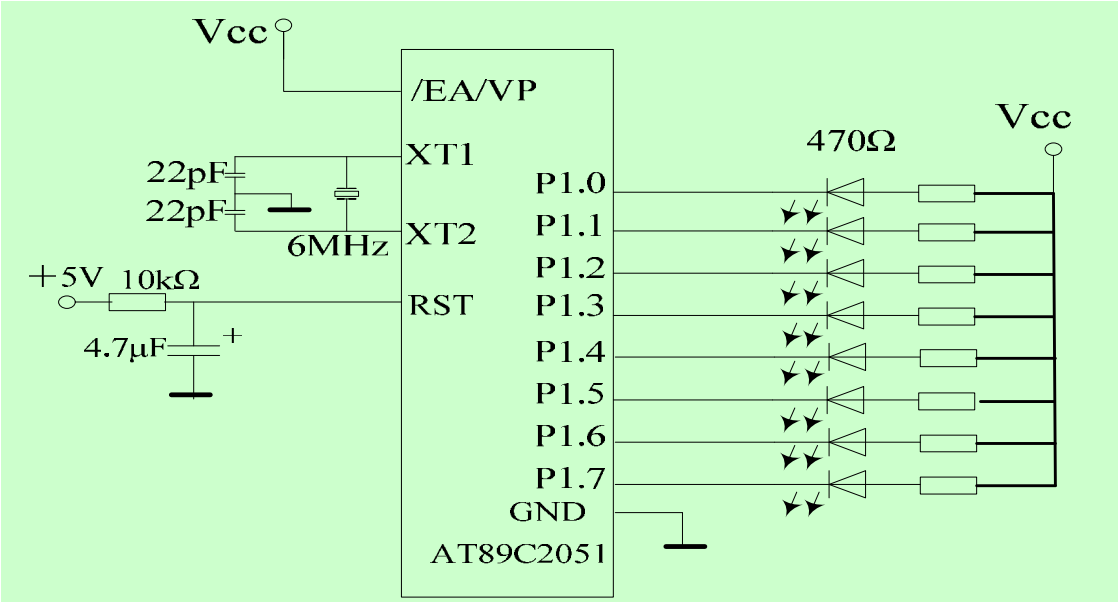
启动定时器



```
L2:   JNB   TF1,L2
      CLR   TF1
      MOV   TL0,#15536 MOD 256
      MOV   TH0,#15536/256
      DJNZ  50H,L2
      MOV   50H,#20
      DJNZ  51H,L2
      MOV   51H,#60
      DJNZ  52H,L2
      MOV   52H,#60
      SJMP  L2
      END
```



例3：如图开始时P1.0亮，延时0.2秒后左移至P1.1亮，如此左移7次至P1.7亮，再延时0.2秒右移至P1.6亮，如此右移7次后至P1.0亮。




编程思路： 延时程序由T0定时10ms，连续延时20次为0.2秒。

初值为：

$X=2^{16}-t\times fosc/12=65536-6\times 10\times 1000/12=60536=0D8F0H$

主程序：

```
        ORG   0000H
        MOV   TMOD, #01
START:  MOV   A, #OFEH
        MOV   R2, #07
LOOP:   MOV   P1，A
        RL    A
        MOV   R3，#20
        LCALL  DELAY
        DJNZ   R2，LOOP
        MOV   R2，#07
LOOP1: MOV   P1，A
        RR    A
        MOV   R3，#20
        LCALL  DELAY
        DJNZ   R2，LOOP1
        LJMP   START
```

方式1定时

点亮第一个灯

左移7次

左移至下一位

延时0.2秒

等待左移结束

右移与左移相同

定时程序：

```
DELAY：SETB   TR0
AGAIN：  MOV   TH0，D8H
         MOV   TL0，#0F0H
LOOP2：JBC    TF0，LOOP3
         LJMP   LOOP2
LOOP3：DJNZ   R3，AGAIN
       CLR    TR0
       RET
       END
```

启动定时器

预置时间常数

等待10mS定时到

等待0.2S定时到

关定时器、返回

例4： 要求当P3.3每来3个脉冲时，P1.0的状态翻转改变一次。

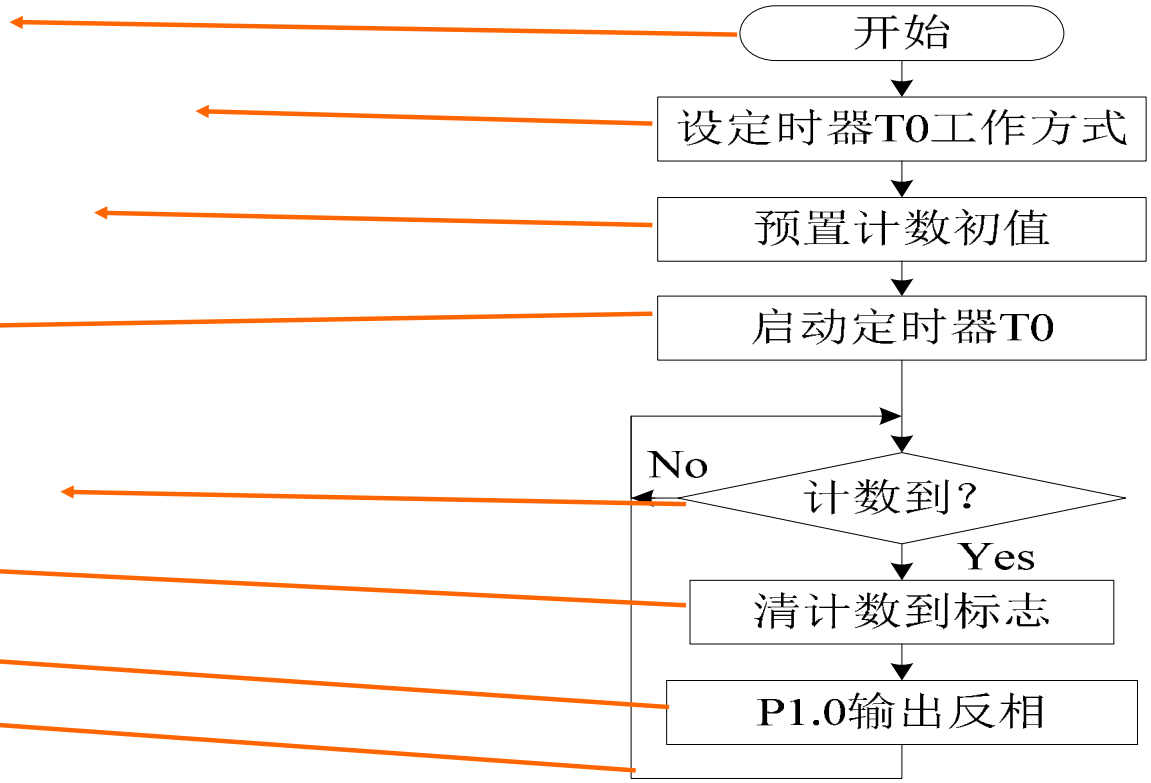
分析： 定时器T0工作在计数方式，计数3次将P1.0输出取反一次。由于计数初值小，因此采用方式2最好。

计数次数＝3次

计数初值＝256－3＝253

程序：

```
        ORG   0000H
        MOV   TMOD,#06H
        MOV   TL0,#253
        MOV   TH0,#253
        SETB  TR0
        MOV   P3,#0FFH
LP2:    JNB   TF0,LP2
        CLR   TF0
        CPL   P1.0
        SJMP  LP2
        END
```



## 5.1.5 定时/计数器T2

### 1、T2的寄存器

1）T2CON控制寄存器

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | $C/\overline{T2}$ | $CP/\overline{RL2}$ |

- TF2 定时器2 溢出标志：

  定时器2 溢出时置位，必须由软件清除。当RCLK或TCLK等于1 时TF2 将不会置位。

- EXF2 定时器2 外部标志：

  当EXEN2＝1且T2EX 的负跳变产生捕获或重装时EXF2置位；定时器2中断开放时EXF2＝1 将使CPU 从中断向量处执行定时器2 中断子程序。EXF2 位必须用软件清零。在递增/递减计数器模式（DCEN＝1 ）中EXF2 不会引起中断。

- RCLK　　接收时钟标志：

  RCLK＝1时，定时器2 的溢出脉冲作为串行口模式1和模式3的接收时钟，RCLK＝0 时，将定时器1 的溢出脉冲作为接收时钟。

- TCLK　　发送时钟标志：

  TCLK＝1时，定时器2 的溢出脉冲作为串行口模式1和模式3的发送时钟，TCLK＝0 时，将定时器1 的溢出脉冲作为发送时钟。

- EXEN2　　定时器2 外部使能标志：

  当其置位且定时器2 未作为串行口时钟时，允许T2EX 的负跳变产生捕获或重装，EXEN2＝0 时，T2EX 的跳变对定时器2 无效。

- TR2　　定时器2 启动/停止控制位：

  置1 时启动定时器。

- C/T2　　定时器/计数器选择：

  置0时定时器2工作在定时状态，置1时工作在计数器状态。

- CP/RL2　　捕获/重装标志：

  置位时，EXEN2＝1时，T2EX 的负跳变产生捕获；清零时，EXEN2＝1则定时器2 溢出或T2EX 的负跳变都可使定时器自动重装。当RCLK＝1 或TCLK＝1 时该位无效且定时器强制为溢出时自动重装。

## 2） T2MOD方式控制寄存器

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | T2OE | DCEN |

- — 不可用，保留将来之用。
- T2OE 定时器2 输出使能位：

  T2OE＝1，允许T2输出； T2OE＝0，禁止T2输出。
- DCEN 向下计数使能位：

  定时器2 可配置成向上/向下计数器， DCEN＝0，T2加1计数；DCEN＝1且T2EX=1时， T2加1计数； DCEN＝1且T2EX=0时， T2减1计数。

## 2、T2的工作方式

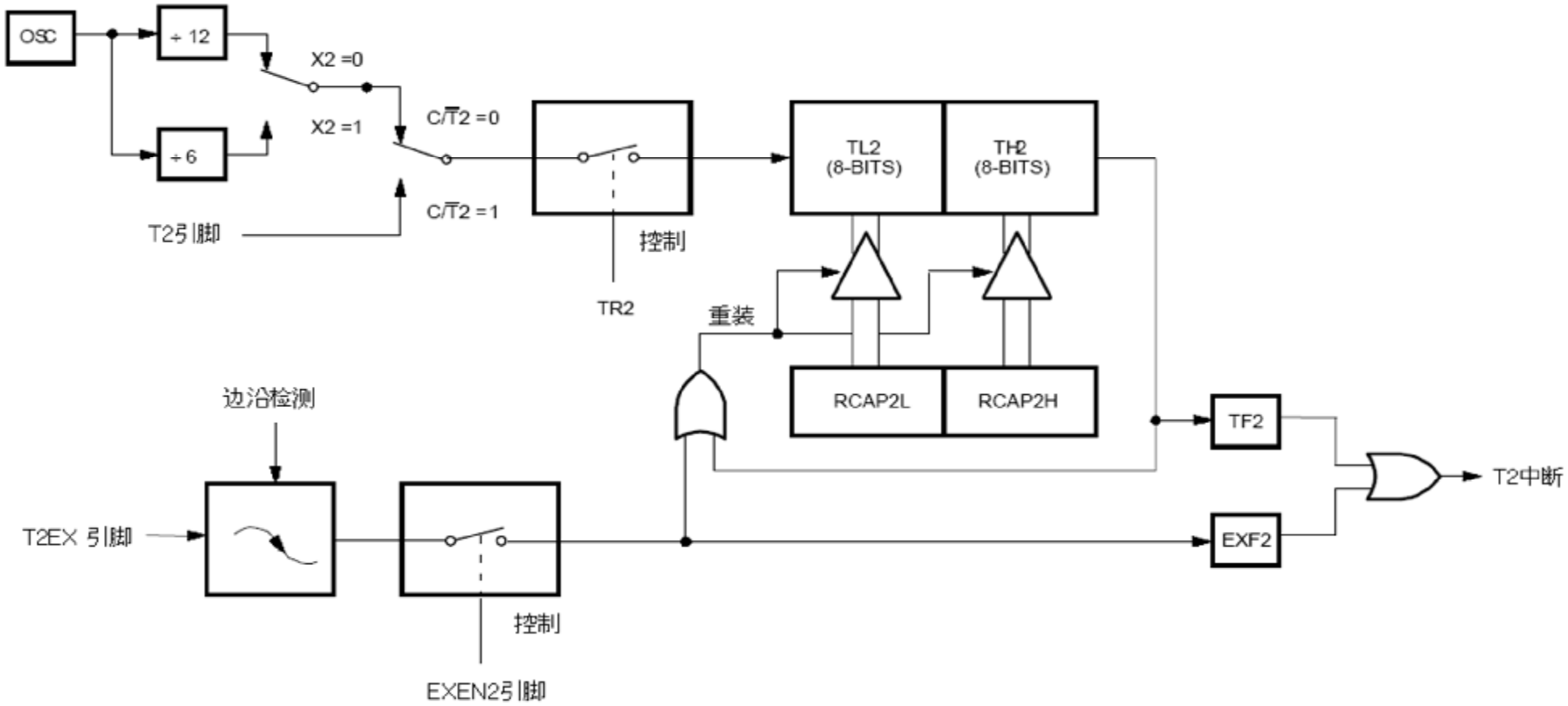
定时器 2 工作方式

| RCLK+TCLK | CP/$\overline{RL2}$ | TR2 | 模式 |
|---|---|---|---|
| 0 | 0 | 1 | 16 位自动重装 |
| 0 | 1 | 1 | 16 位捕获 |
| 1 | X | 1 | 波特率发生器 |
| X | X | 0 | （关闭） |

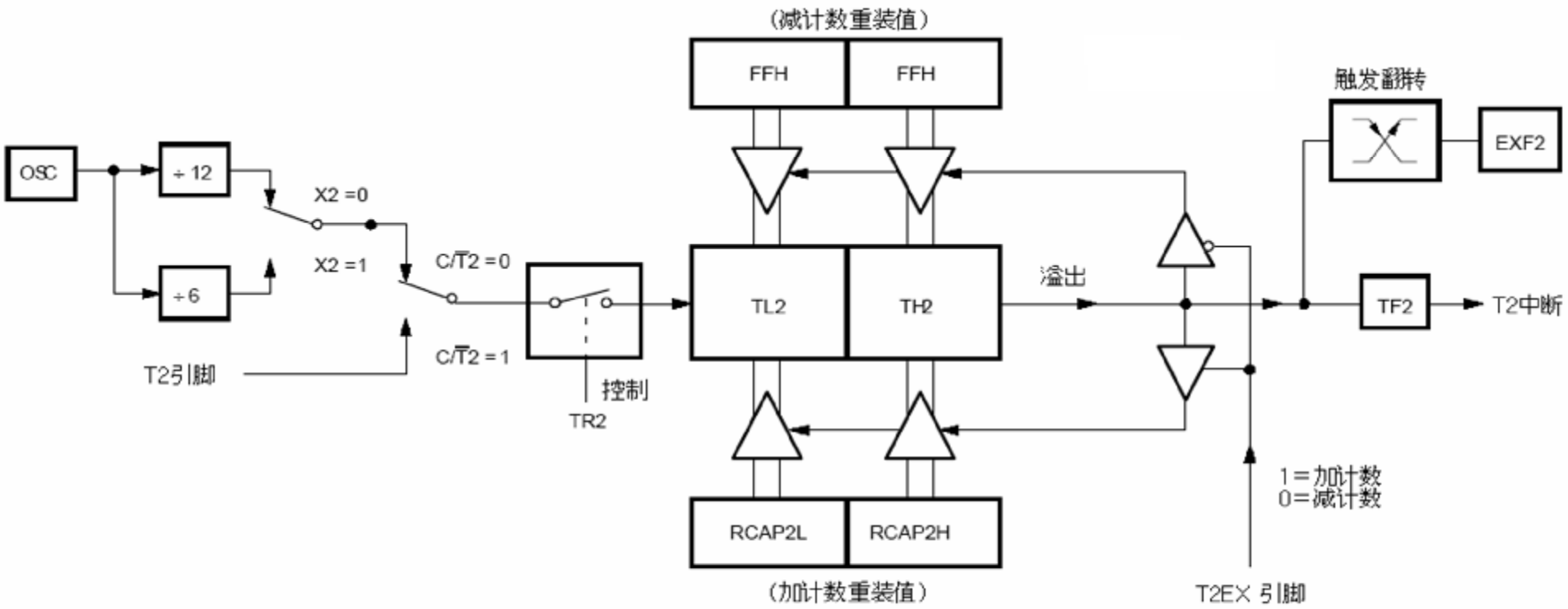
### 1）定时/计数自动重装方式（递增/递减计数器）

- 16 位自动重装模式中，定时器T2 可通过C/T2 配置为定时器/计数器，编程控制递增/递减计数。计数的方向是由DCEN 递减计数使能位确定的，DCEN 位于T2MOD 寄存器 中当，DCEN＝0 时，定时器2 默认为向上计数，当DCEN＝1 时，定时器2 可通过T2EX 确定递增或递减计数。

- 当DCEN＝0 时，定时器2 自动递增计数，在该模式中通过设置EXEN2 位进行选择。如果EXEN2＝0 ，定时器2 递增计数到溢出后将TF2 置位，然后将RCAP2L 和RCAP2H 中的16 位值作为重新装载值装入定时器2，RCAP2L 和RCAP2H 的值是通过软件预设的。如果EXEN2＝1，16 位重新装载可通过溢出或T2EX 从1→0 的负跳变实现，此负跳变同时将EXF2置位，如果定时器2中断被使能，则当TF2 或EXF2置1时产生中断。
- DCEN＝1时，定时器2 可递增或递减计数，此模式允许T2EX 控制计数的方向。当T2EX＝1时定时器2递增计数，计数到0FFFFH 后溢出并置位TF2 ，还将产生中断（如果中断被使能）。定时器2的溢出将使RCAP2L和RCAP2H中的16位值作为重新装载值放入TL2和TH2。当T2EX ＝0 时，将使定时器2递减计数，当TL2 和TH2 计数到等于RCAP2L 和RCAP2H 时，定时器产生溢出，定时器2 溢出置位TF2 ，并将0FFFFH 重新装入TL2 和TH2。当定时器2递增/递减产生溢出时，外部标志位EXF2 翻转，如果需要可将EXF2 位作为第17 位，在此模式中EXF2 标志不会产生中断。



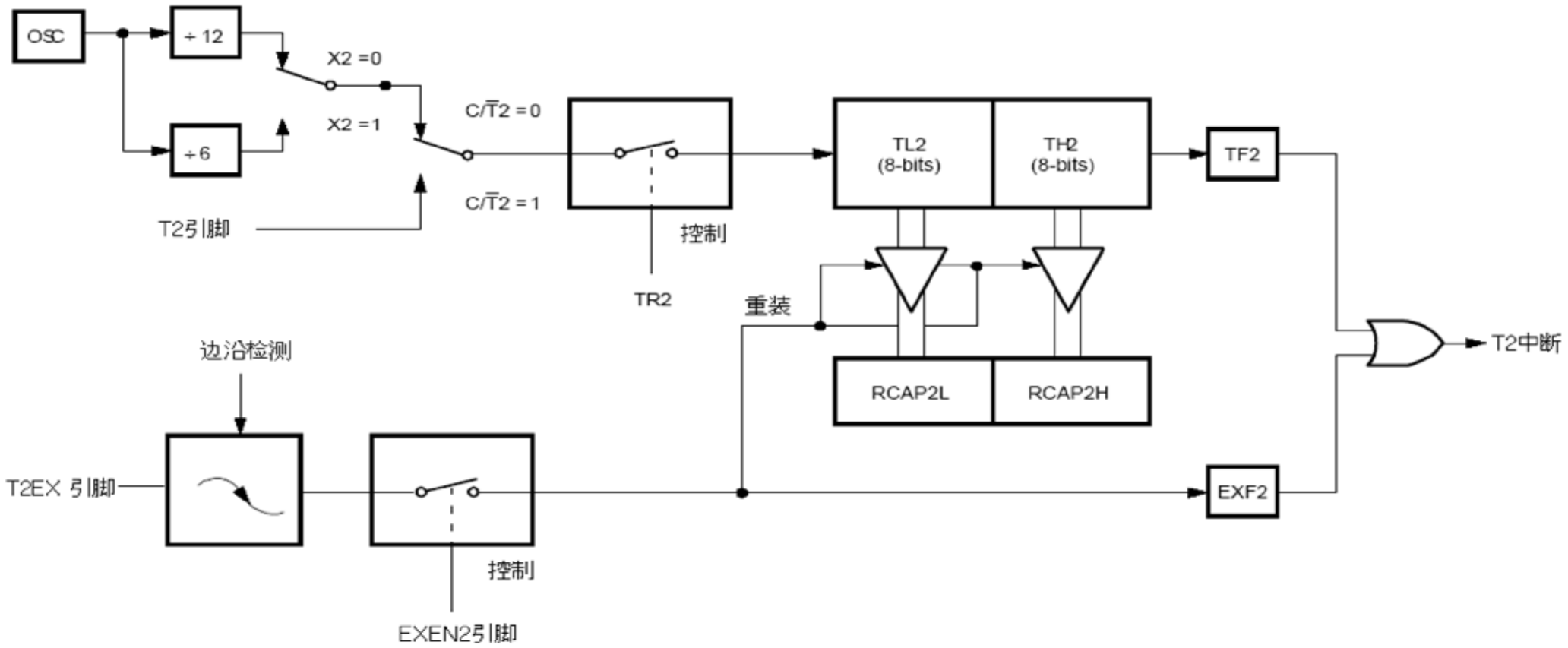

定时器 2 自动重装模式（DCEN=0）




定时器 2 自动重装模式（DCEN=1）

## 2）定时/计数捕获方式

- 在捕获模式中通过T2CON 中的EXEN2 设置两个选项。如果EXEN2＝ 0 ，定时器2 作为一个16 位定时器或计数器（由T2CON 中C/T2 位选择），溢出时置位TF2（定时器2 溢出标志位），该位可用于产生中断（通过使能IE 寄存器中的定时器2 中断使能位）。如果EXEN2＝1，与以上描述相同，但增加了一个特性，即外部输入T2EX 由1 变0 时，将定时器2 中TL2 和TH2 的当前值各自捕获到RCAP2L 和RCAP2H。另外T2EX 的负跳变使T2CON 中的EXF2 置位，EXF2 也象TF2 一样能够产生中断，其向量与定时器2溢出中断地址相同，定时器2 中断服务程序通过查询TF2 和EXF2 来确定引起中断的事件。
- 捕获模式如图所示，在该模式中TL2 和TH2 无重新装载值。甚至当T2EX 产生捕获事件时，计数器仍以T2EX 的负跳变或振荡频率的1/12计数。


OSC
÷12
÷6
X2 =0
X2 =1
C/T2 =0
C/T2 =1
T2引脚
控制
TR2
TL2
(8-bits)
TH2
(8-bits)
TF2
重装
RCAP2L
RCAP2H
T2中断
边沿检测
T2EX 引脚
EXF2
控制
EXEN2引脚

定时器 2 捕获模式

### 3）波特率发生器方式

- 寄存器T2CON 的位TCLK 和或RCLK 允许从定时器1 或定时器2 获得串行口发送和接收的波特率。当TCLK=0 时定时器1作为串行口发送波特率发生器，当TCLK=1时定时器2 作为串行口发送波特率发生器。RCLK对串行口接收波特率有同样的作用，通过这两位串行口能得到不同的接收和发送波特率，一个通过定时器1 产生，另一个通过定时器2 产生。
- 定时器2工作在波特率发生器模式时与自动重装模式相似，当TH2 溢出时，波特率发生器模式使定时器2 寄存器重新装载来自寄存器，寄存器RCAP2H 和RCAP2LR 的值由软件预置。当串行口工作于模式1 和模式3 时，波特率由下面给出的定时器2 溢出率所决定：

$$\text{串行口方式 1和3波特率} = \frac{\text{T2 溢出率}}{16} = \frac{\text{振荡频率}}{32 \times (65536 - X)}$$

式中的X为16位无符号数（ 即RCAP2H 和RCAP2L的16位的值）。

## 4）可编程时钟输出方式

- 当C/T2＝0（T2工作在定时器方式）时，置位T2OE则定时器T2可从P1.0输出占空比为1:1的时钟信号。时钟输出频率为：

$$\text{时钟输出频率} = \frac{\text{振荡频率}}{4 \times (65536 - X)}$$

式中的X为16位无符号数（ 即RCAP2H 和RCAP2L的16位的值）。

## 5.1.6 定时器T3—WDT监视定时器

### 5.6.1 WDT的功能和应用特点

当系统的CPU部位受到干扰信号的作用时，将使系统失控。最典型的故障是破坏程序计数器PC的状态值。导致程序在地址空间内“乱飞”，或者陷入死循环。而我们对这种情况的处理主要有这么几种方法：（1）指令冗余技术；（2）软件陷阱技术；（3）看门狗技术。

看门狗是利用一个专门的定时器，来监控主程序的运行，也就是说在主程序的正常运行过程中，我们要在看门狗定时时间到之前对定时器进行复位，如果出现死循环，或者说PC指针不能回来，那么看门狗得不到复位，其定时时间到后就会产生一个信号使单片机复位，程序重新开始运行。

要激活或复位89S51看门狗，只需向看门狗复位寄存器WDTRST（地址为A6H）顺序写入1EH和E1H即可。

```
MOV     0A6H,#1EH     ;先送1EH
MOV     0A6H,#0E1H    ;后送E1H
```

注意事项：

1. 89S51的看门狗必须由程序激活后才开始工作。所以必须保证CPU有可靠的上电复位。否则看门狗也无法工作。
2. 看门狗使用的是CPU的晶振。在晶振停振的时候看门狗也无效。
3. 89S51只有14位计数器，在16383个机器周期内必须至少喂狗（看门狗复位）一次，而且这个时间是固定的，无法更改。当晶振为12M时每16个毫秒需喂狗一次。89S52只有13位计数器，在8191个机器周期内必须至少喂狗一次。当晶振为12M时每8个毫秒需喂狗一次.

# 5.2 单片机串行通信

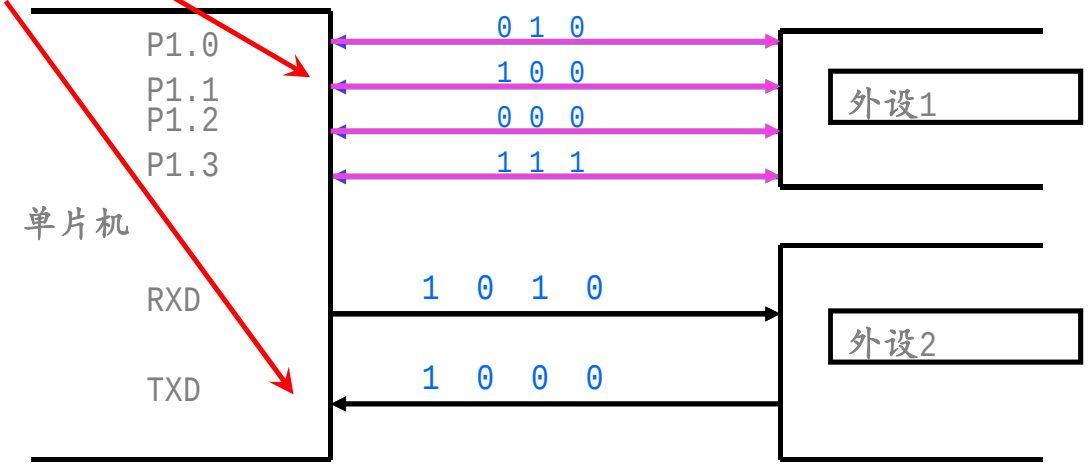
## 5.2.1 串行通信概述

### 1. 数据通信的传输方式

【并行通信和串行通信】

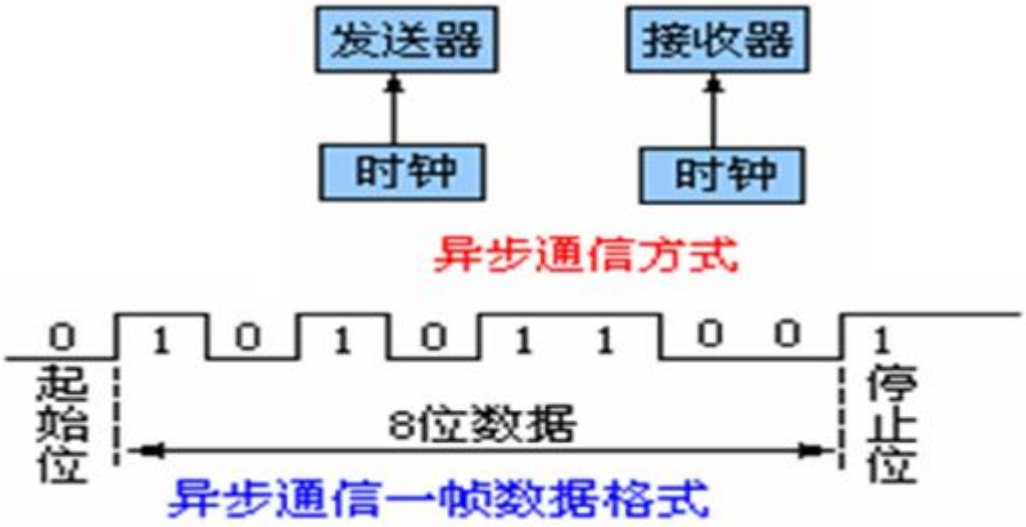
并行通信：数据的各位同时送出。占用I/O多，速度快。

串行通信：数据的各位逐位送出。线路简单，速度慢。



## 2. 串行数据通信两种形式

【异步通信】在这种通信方式中，接收器和发送器有各自的时钟，它们的工作是非同步的，异步通信用一帧来表示一个字符，其内容如下：一个起始位，接着是若干个数据位 和一个停止位。




特点：

a) 无需传送同步脉冲

b) 发送/接收端的两个时钟源可彼此独立

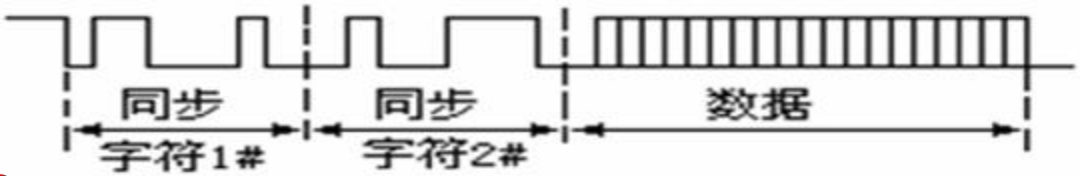
c) 字符帧长度也不受限

d) 设备简单化

e) 每帧含有起始位和停止位，降低了有效数据的传输速率

【同步通信】发送器和接收器由同一个时钟源控制，同步传输方式去掉了起始位和停止位，只在传输数据块时先送出一个同步头（字符）标志即可。在没有信息要传输时，要填上空字符，因为同步传输不允许有间隙。

同步通信方式



同步通信数据格式

特点：

- 同步数据帧由同步字符和数据组成
- 传输速率较高（可56kbps以上）
- 要求发送/接收时钟始终保持严格同步

### 3. 串行数据通信的传输速率

串行数据传输速率有两个概念，即每秒转送的位数bps（Bit per second）和每秒符号数—波特率（Band rate），波特率也叫比特率。在具有调制解调器的通信中，波特率与调制速率有关。

**举例：**设有一帧信息，1个起始位、8个数据位、1个停止位，传输速率为每秒960个字符。求波特率。

**解：**（1＋8＋1）×960 = 9600 b/s = 9600波特。

4 .串行数据传送方向

单工通讯： 数据单向传送。

半双工通讯：数据可分时双向传送。

全双工通讯：可同时进行发送和接收。


5 .串行接口功能

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。

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。

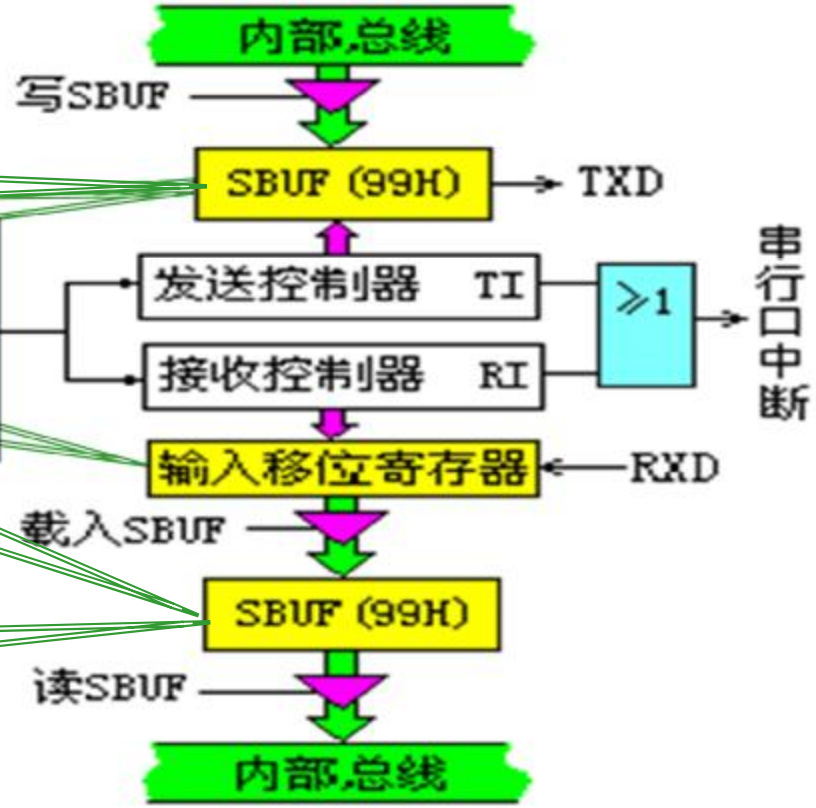
控制器：接收编程命令和控制参数，设置工作方式：同步/异步、字符格式、波特率、校验方式、数据位与同步时钟比例等。

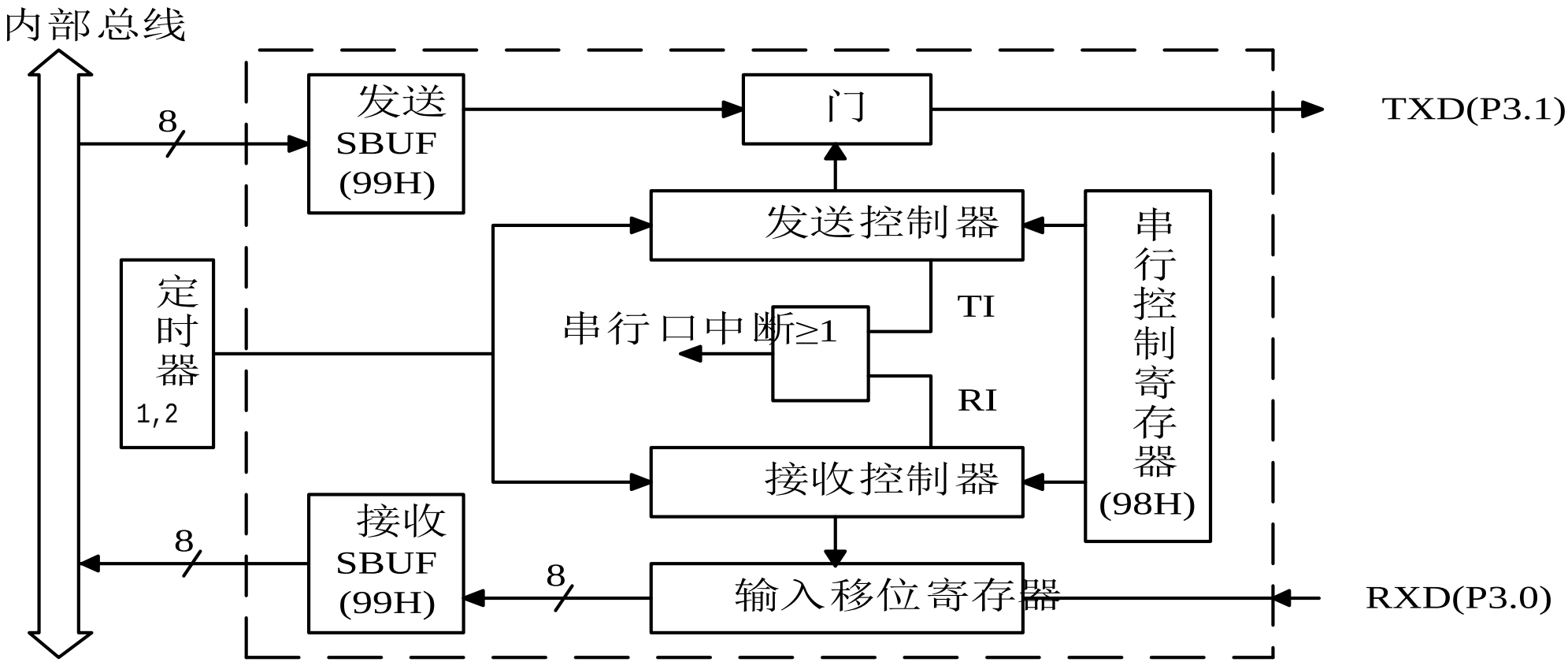
## 5.2.2 80C51串行口简介

### 1 串行口结构与工作原理

发送缓冲器SBUF只能写入不能读出。

接收缓冲器SBUF只能读出不能写入。

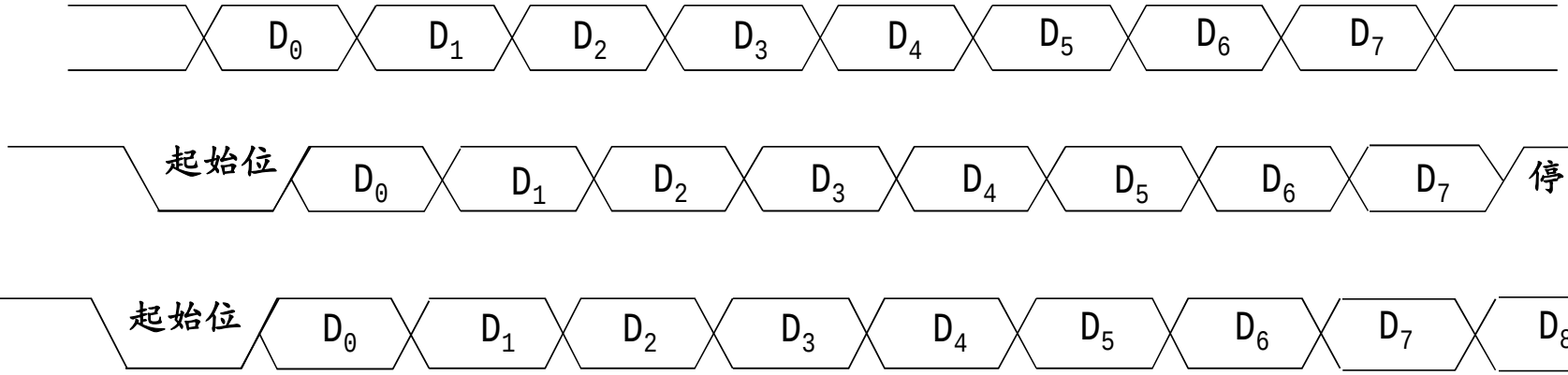



内部总线
8
发送
SBUF
(99H)
门
TXD(P3.1)
发送控制器
定时器
1,2
串行口中断
≥1
TI
RI
串行控制寄存器
(98H)
接收控制器
8
接收
SBUF
(99H)
8
输入移位寄存器
RXD(P3.0)

串行口结构示意图

串行接口输入/输出引脚：TXD(P3.1)、RXD(P3.0)

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)并申请串行口中断。

中断控制：中断允许位ES

中断入口：0023H

## 2 串行口控制

| SCON | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
| 位地址 | 9FH | 9EH | 9DH | 9CH | 9BH | 9AH | 99H | 98H |

SM0、SM1：工作方式选择位（四种工作方式）

SM2：多机通信控制位

REN：允许接收控制位

TB8：发送的第九位数

RB8：接收的第九位数

| SM0 | SM1 | 工作方式选择 | |
|---|---|---|---|
| 0 | 0 | 方式0 | 移位寄存器 |
| 0 | 1 | 方式1 | 8位UART(10) |
| 1 | 0 | 方式2 | 9位UART(11) |
| 1 | 1 | 方式3 | 9位UART(11) |

TI / RI ：发送/接受中断请求标志位，该标志在中断服务程序中也必须由软件清零。

## 3 波特率的设置

在串行通信中，收发双方对传送的数据速率，即波特率要有一定的约定。80C51单片机的串行口通过编程可以有4种工作方式。其中，方式0和方式2的波特率是固定的，方式1和方式3的波特率可变，由定时器1的溢出率决定（AT89S系列还可由T2的溢出率决定），下面加以分析。

### 1) 方式0和方式2

在方式0中，波特率为时钟频率的1/12，即 $f_{osc}/12$ ，固定不变。在方式2中，波特率取决于PCON中的SMOD值，当SMOD=0时，波特率为 $f_{osc}/64$ ；当SMOD =1时，波特率为 $f_{osc}/32$ 。即：

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

2） 方式1和方式3

80C51在方式1和方式3下，波特率由定时器1的溢出率和SMOD共同决定。即：

$$\text{方式1和方式3波特率}=\frac{2^{SMOD}}{32}\times \text{T1的溢出率}$$

当定时器1做波特率发生器使用时，通常是工作在方式2，即自动重装载的8位定时器。设计数的预置值（初始值）为X，那么每过256-X个机器周期，定时器溢出一次。为了避免因溢出而产生不必要的中断，此时应禁止T1中断。波特率为：

$$\text{方式1和方式3波特率}=\frac{2^{SMOD}}{32}\times\frac{f_{osc}}{12\times(256-X)}$$

定时器1产生的常用波特率

| 波特率/(b/s) | $f_{osc}$/MHz | SMOD | 定时器1 | | |
|---|---|---|---|---|---|
| | | | C/$\overline{T}$ | 模式 | 初始值 |
| 方式0：1 | 12 | × | × | × | × |
| 方式2：375 k | 12 | 1 | × | × | × |
| 方式1、3：62.5 k | 12 | 1 | 0 | 2 | FFH |
| 19.2 k | 11.0592 | 1 | 0 | 2 | FDH |
| 9.6 k | 11.0592 | 0 | 0 | 2 | FDH |
| 4.8 k | 11.0592 | 0 | 0 | 2 | FAH |
| 2.4 k | 11.0592 | 0 | 0 | 2 | F4H |
| 1.2 k | 11.0592 | 0 | 0 | 2 | E8H |
| 137.5 k | 11.986 | 0 | 0 | 2 | 1DH |
| 110 | 6 | 0 | 0 | 2 | 72H |
| 110 | 12 | 0 | 0 | 1 | FEEBH |

## 5.2.3 串行通信应用实例

### 1、 方式0

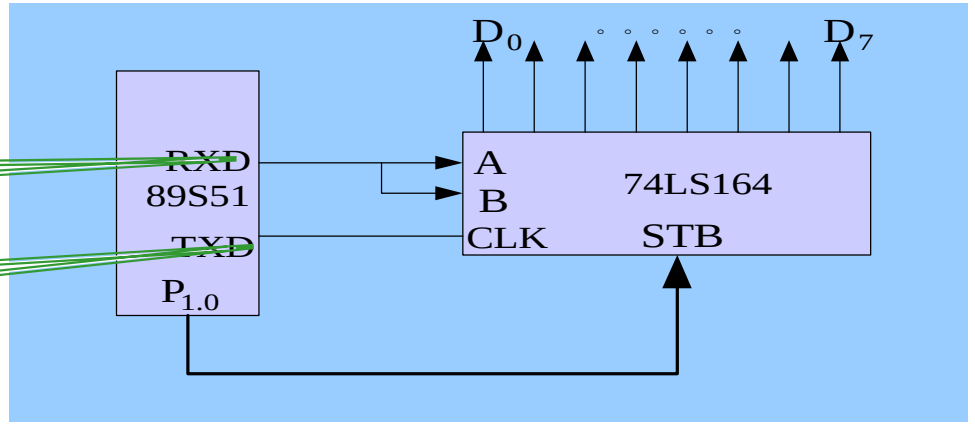
**数据输出（发送）**

当数据写入SBUF后，数据从RXD端在移位脉冲（TXD）的控制下，逐位移出。当8位数据全部移出后，TI由硬件置位，发生中断请求。若CPU响应中断，则从0023H单元开始执行串行口中断服务程序。

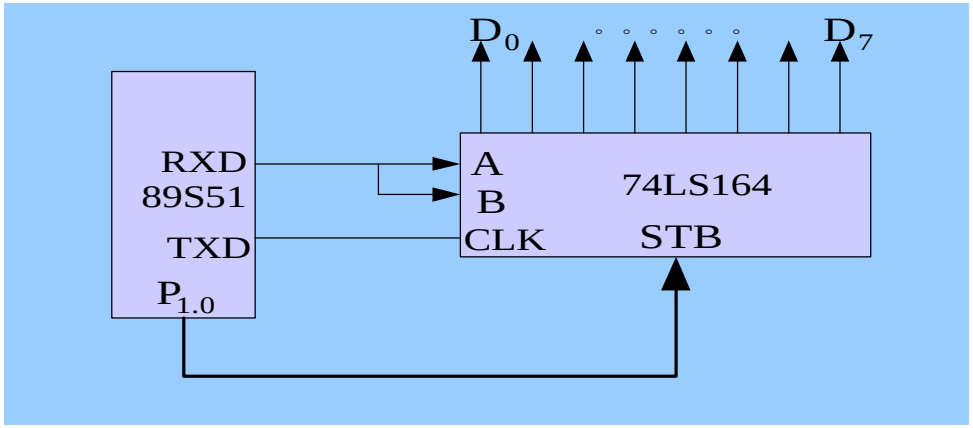
例：数据由74LS164并行输出，其接口逻辑如图所示，[illegible]，74LS164能完成数据的串并转换。

数据输出

时钟输出




接口逻辑




```
        MOV   SCON，#00H
        SETB   P1.0
        MOV   A，#DATA
        MOV   SBUF，A
WAIT：  JNB   TI，WAIT
        CLR   TI
        CLR   P1.0
```

方式0

选通74LS164

启动发送

等待发送结束

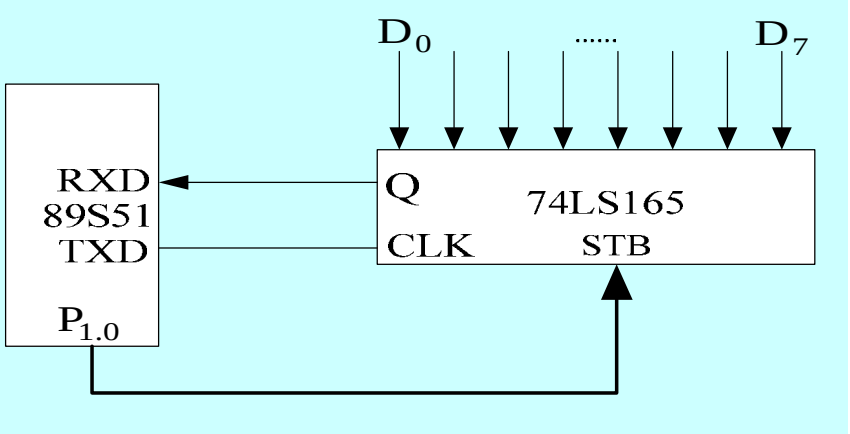
清除发送标志

关74LS164

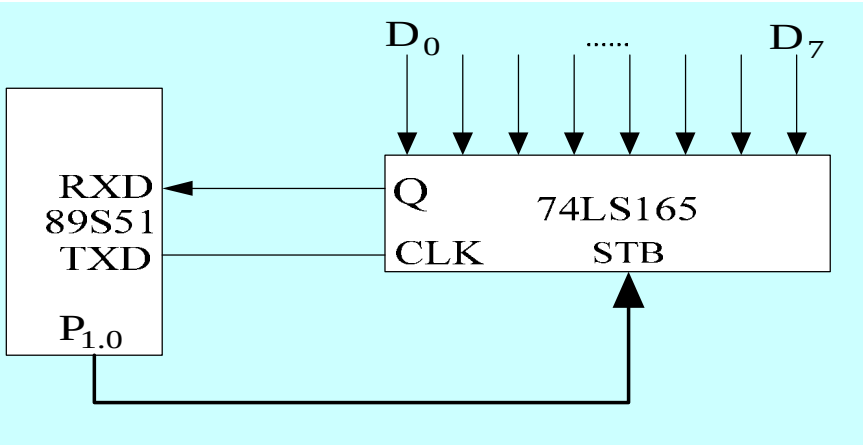
数据输入（接收）

要实现接收数据，必须首先把SCON中的允许接收位REN设置为1。当REN设置为1时，数据就在移位脉冲的控制下，从RXD端输入。当接收到8位数据时，置位接收中断标志位RI，发生中断请求。

例：数据通过外接74LS165输入，其接口逻辑如图所示。由逻辑图可知，通过外接74LS165，串行口能够实现数据的并行输入。




外接移位寄存器输入




```
RQ：CLR   P1.0
    SETB  P1.0
    MOV   SCON，#10H
    JNB   RI，$
    CLR   RI
    MOV   A，SBUF
```

读入并行数据

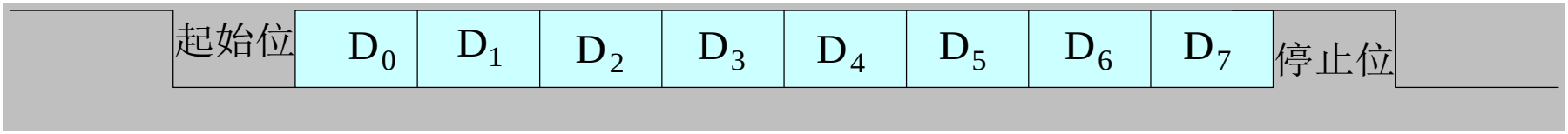
方式0、允许接收

等待数据到

清除接收标志
读数据

## 2、串行工作方式1

方式1为10位为一帧的异步串行通信方式。其帧格式为1个起始位、8个数据位和1个停止位，如下图所示。




### 数据输出（发送）

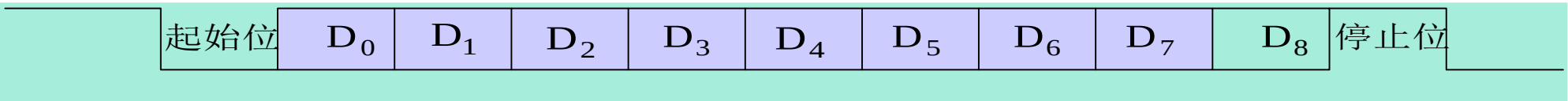
数据写入SBUF后，开始发送，此时由硬件加入起始位和停止位，构成一帧数据，由TXD串行输出。输出一帧数据后，TXD保持在高电平状态下，并将TI置位，通知CPU可以进行下一个字符的发送。

### 数据输入（接收）

当REN=1且接收到起始位后，在移位脉冲的控制下，把接收到的数据移入接收缓冲寄存器（SBUF）中。当RI=0，且停止位为1或SM2=0时，停止位进入RB8位，同时置中断标志RI；否则信息将丢失。所以，方式1接收时，应先用软件清除RI或SM2标志。

## 3、串行工作方式2和方式3

方式2和方式3都为11位为一帧的异步串行通信方式。其帧格式为1个起始位、9个数据位和1个停止位，如下图所示。方式2与方式3只在波特率设置上由所不同。




方式2和方式3在8个数据位上增加了第9个数据位（$D_8$），其功能由用户确定，是一个可编程位。

在发送数据时，应先在SCON的$TB_8$位中把第9个数据位的内容准备好。这可使用如下指令完成：

```
SETB    TB8     ; TB8位置“1”
CLR     TB8     ; TB8位置“0”
```

串行口把接收到的前8个数据位送入SBUF，而把第九数据位送入RB8。

方式例：

| SCON | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
| 位地址 | 9FH | 9EH | 9DH | 9CH | 9BH | 9AH | 99H | 98H |

```
TT:    MOV  SCON，#80H
       MOV  A，#DATA
       MOV  C，PSW.0
       MOV  TB8，C
       MOV  SBUF，A
LOOP：JBC   TI，NEXT
       SJMP  LOOP
NEXT：…
```

方式2

TB8为偶校验位

启动发送

等待发送结束

例：接收一个数据字节并做奇偶校验的程序段。

```
RR:    MOV   SCON，#90H
```

方式2，允许接收

| SCON | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
| 位地址 | 9FH | 9EH | 9DH | 9CH | 9BH | 9AH | 99H | 98H |

```
       JB    PSW.0，ONE
       JB    RB8，ERR
       SJMP  REXT
ONE:   JNB   RB8，ERR
REXT:  …
ERR:   …
```

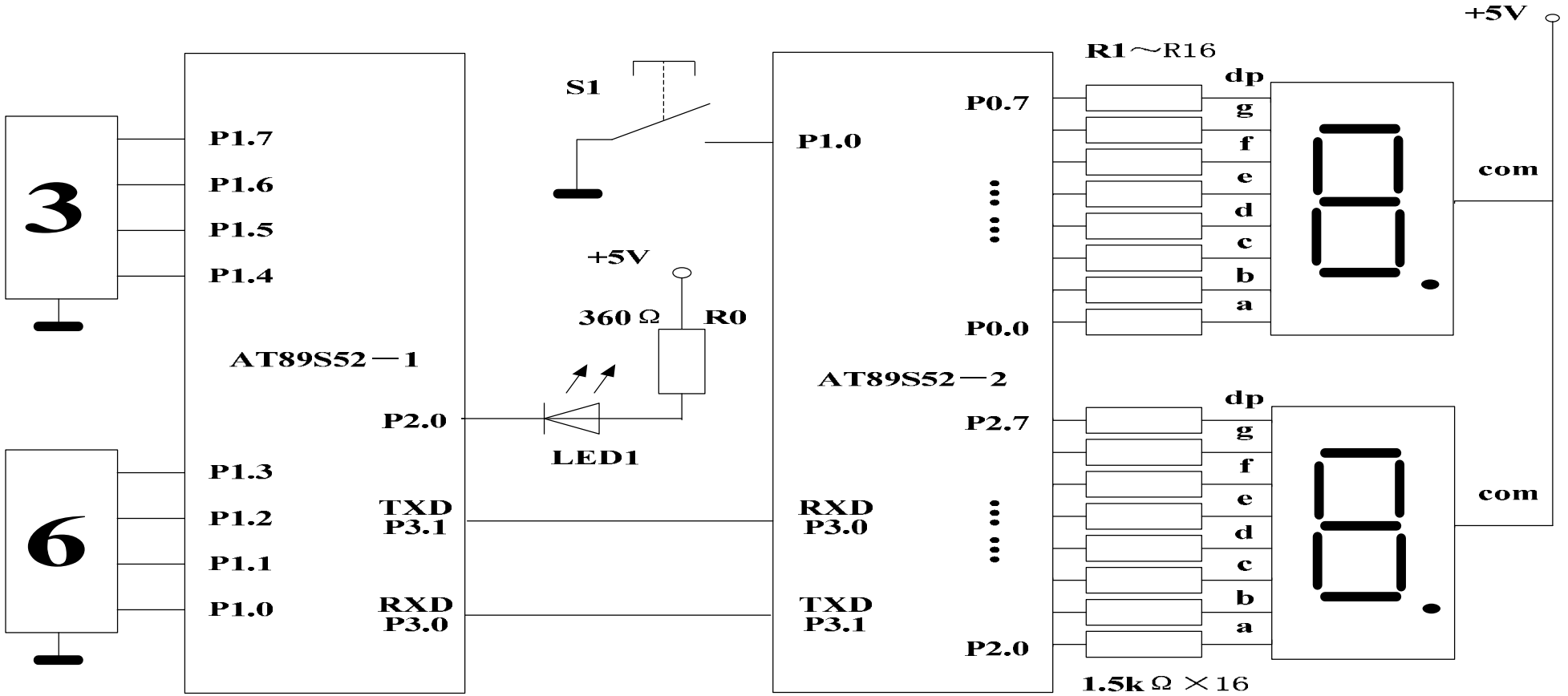
奇偶校验

9位数据1的个数为偶数则发送正确

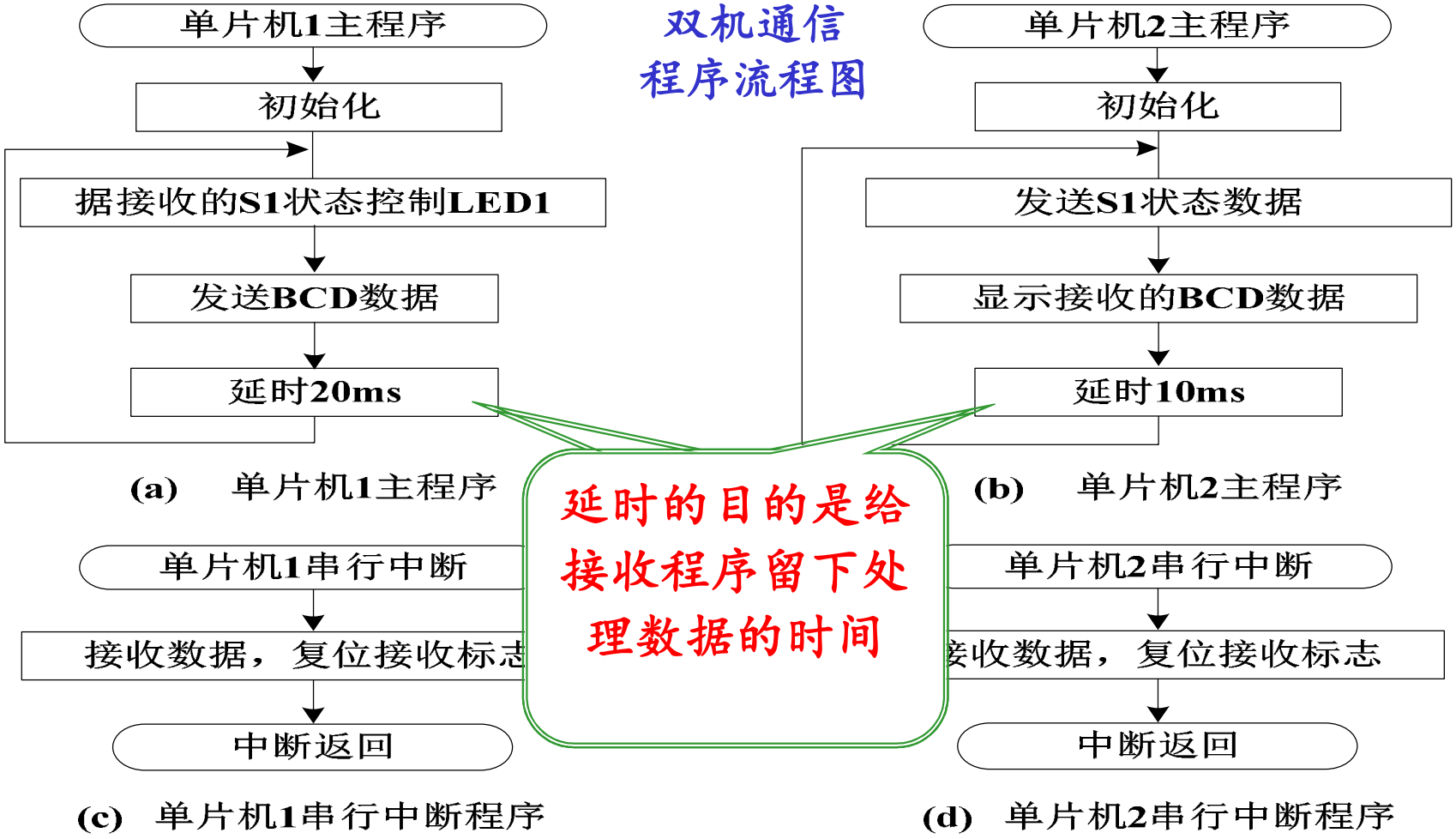
## 4、点对点串行双机通信系统的设计

**任务：**设计一个单片机双机通信系统，单片机1从P1口输入2位BCD数据并定期将该数据发送给单片机2，单片机2接收到单片机1的数据后显示在两位数码管上；单片机1根据接在单片机2上按钮S1的状态控制发光二极管LED1的亮灭。

**分析：**任务中单片机要求发送与接收的数据都只有一个字节，只要掌握了串行口相关特殊功能寄存器的设置方法，采用查询方式或者中断方式发送接收数据都较简单。任务要求从单片机1的P1口输入2位BCD数据，可以使用BCD拨码盘实现。


+5V
R1～R16
dp
g
f
e
d
c
b
a
com
S1
P0.7
P1.0
P0.0
3
P1.7
P1.6
P1.5
P1.4
AT89S52—1
+5V
360Ω
R0
P2.0
LED1
P1.3
P1.2
P1.1
P1.0
6
TXD
P3.1
RXD
P3.0
AT89S52—2
P2.7
RXD
P3.0
TXD
P3.1
P2.0
1.5kΩ×16

双机通信电路原理图


双机通信
程序流程图
单片机1主程序
初始化
据接收的S1状态控制LED1
发送BCD数据
延时20ms
(a) 单片机1主程序
单片机2主程序
初始化
发送S1状态数据
显示接收的BCD数据
延时10ms
(b) 单片机2主程序
延时的目的是给
接收程序留下处
理数据的时间
单片机1串行中断
接收数据，复位接收标志
中断返回
(c) 单片机1串行中断程序
单片机2串行中断
接收数据，复位接收标志
中断返回
(d) 单片机2串行中断程序

```
/* 单片机1通信程序 ************************************/
#include <AT89X52.H>
unsigned char S1;          // 存放接收数据
sbit  LED1=P2^0;

/* 单片机1串行口接收中断服务子程序 *****************/
void serial(void)  interrupt 4 using 1
{
  if (RI)
  {
    RI=0;                  // 接收中断标志位复位
        S1=SBUF;           // 接收发送端传送来的数据
  }
}
```

```
void main(void)     /* 单片机1主函数 ************************************/
{
  TMOD=0x20; TR1=1;      // T1为方式2
  TH1=0xFA; TL1=0xFA;    // 波特率为4800bps
  IE=0x90;               // 开启串行口中断
  SCON=0x50;             // 设串口为方式1、允许串行口接收
  while (1)
```

| SCON | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
| 位地址 | 9FH | 9EH | 9DH | 9CH | 9BH | 9AH | 99H | 98H |

```
    if (S1==0x55) LED1=0;// 根据S1控制LED
          else LED1=1;
    delay20ms();         // 20ms延时程序，没有编写
  }
}
```

```
/* 单片机2通信程序 ************************************/
#include <AT89X52.H>
unsigned char code table[]={0xc0,0xf9,0xa4,0xb0,0x99,
                                        0x92,0x82,0xf8,0x80,0x90};
unsigned char tmp;              // 存放接收数据
sbit  S1=P1^0;
/* 显示程序 ******************************************/
void display2(unsigned char x)      // BCD数显示程序
{
  P2=table[x>>4];               // 显示十位
  P0=table[x&0x0f];                 // 显示个位
}
```

```
/* 单片机2串行口中断服务子程序 ************************/
void serial(void)  interrupt 4 using 1
{
  if (RI)
  {
    RI=0;                          // 接收中断标志位复位
        tmp=SBUF;          // 接收发送端传送来的数据
  }
}
```

```
void main(void)     /* 单片机2主函数 ************************************/
{
  TMOD=0x20; TR1=1;      // T1为方式2
  TH1=0xFA; TL1=0xFA;    // 波特率为4800bps
  IE=0x90;               // 开启串行口中断
  SCON=0x50;             // 设串口为方式1、允许串行口接收
  while(1)
  {
    if (S1) SBUF=0xaa;
            else SBUF=0x55;
          while(!TI);    // 等待发送数据结束
          TI=0;          // TI复位
    display2(tmp);       // 显示BCD数据
    delay10ms();
  }
}
```

## 5、串行总线RS232与RS485

- 接口的物理结构
- 接口的电气特性
- 通讯距离的长短
- 能否支持多点通讯
- 通讯线的差别
- 传输数据的最大波特率

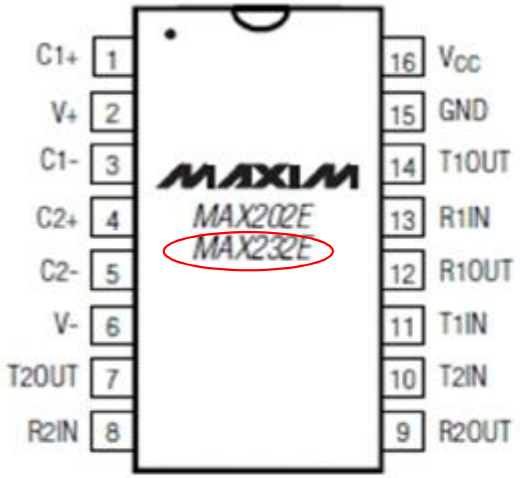
# （1）接口的物理结构

## RS232：

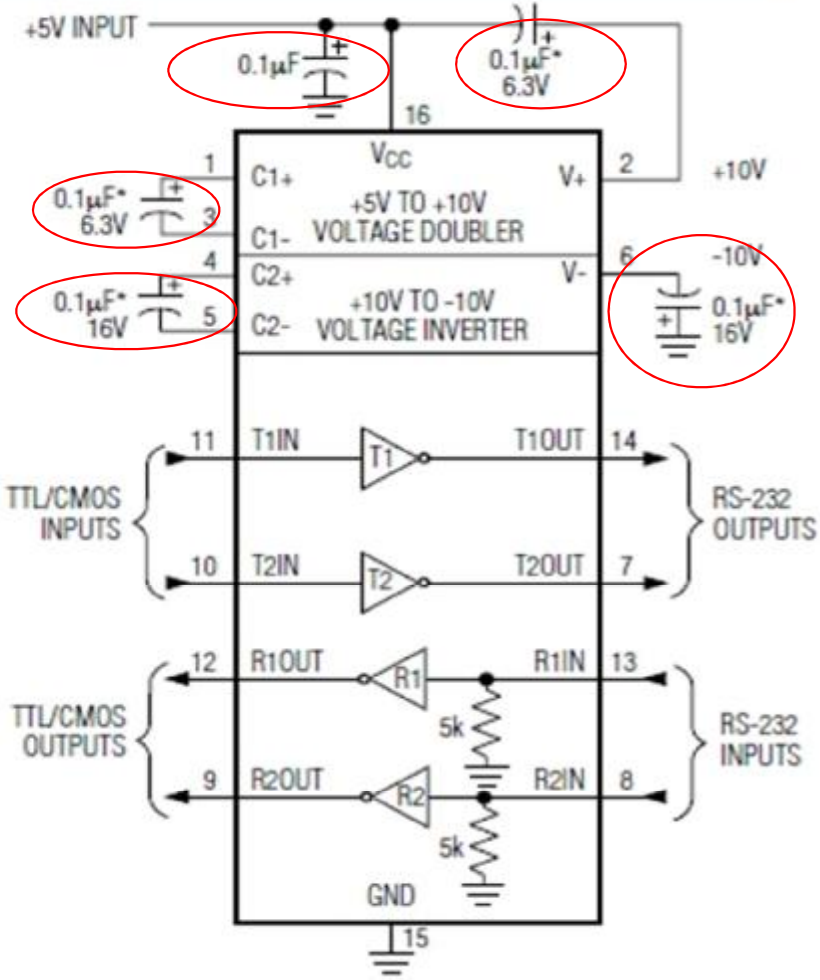
RS232接口连接器使用型号为DB-25的25芯插头座。一些设备与PC机连接的RS-232接口,因为不使用对方的传送控制信号，只需三条接口线，即“发送数据”、“接收数据”和“信号地” 。所以采DB-9的9芯插头座，传输线采用屏蔽双绞线。

DB-9的9芯插头座

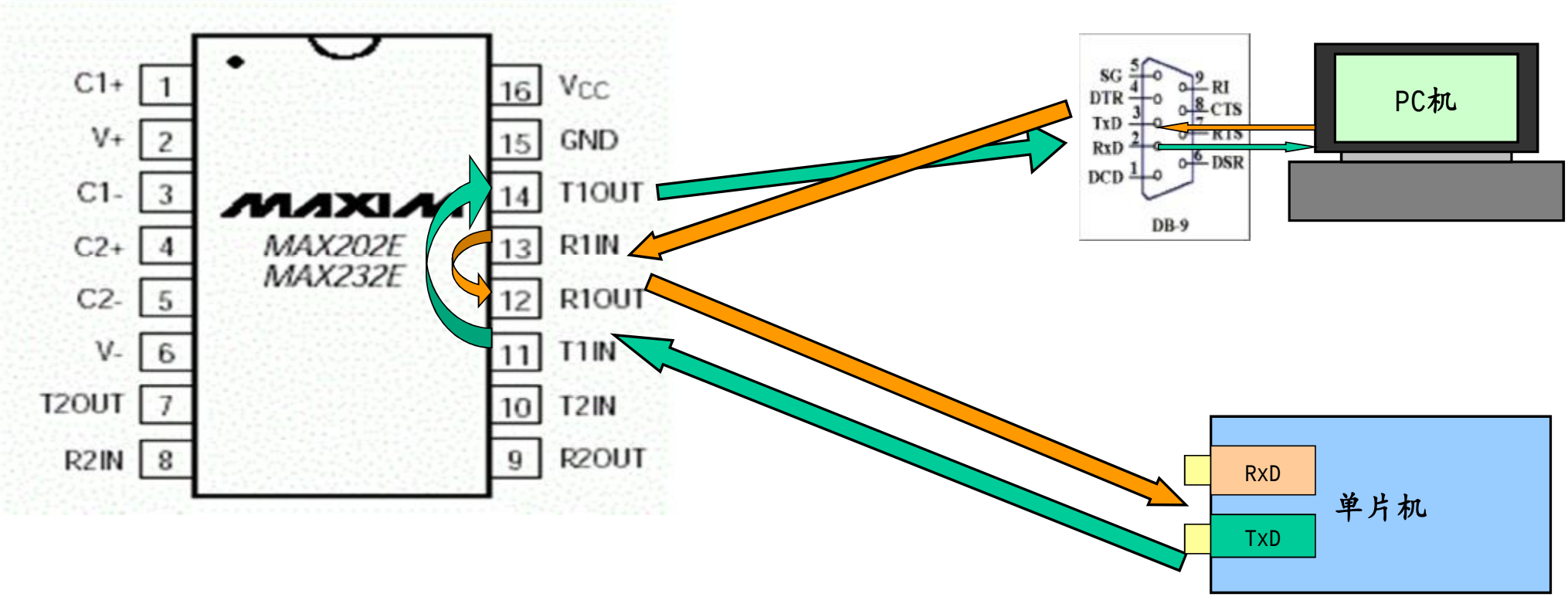
TOP VIEW




DIP/SO/TSSOP




PIN NUMBERS ON TYPICAL OPERATING CIRCUIT REFER TO DIP/SO/TSSOP PACKAGE, NOT LCC.
* 1.0µF CAPACITORS, MAX232E ONLY.

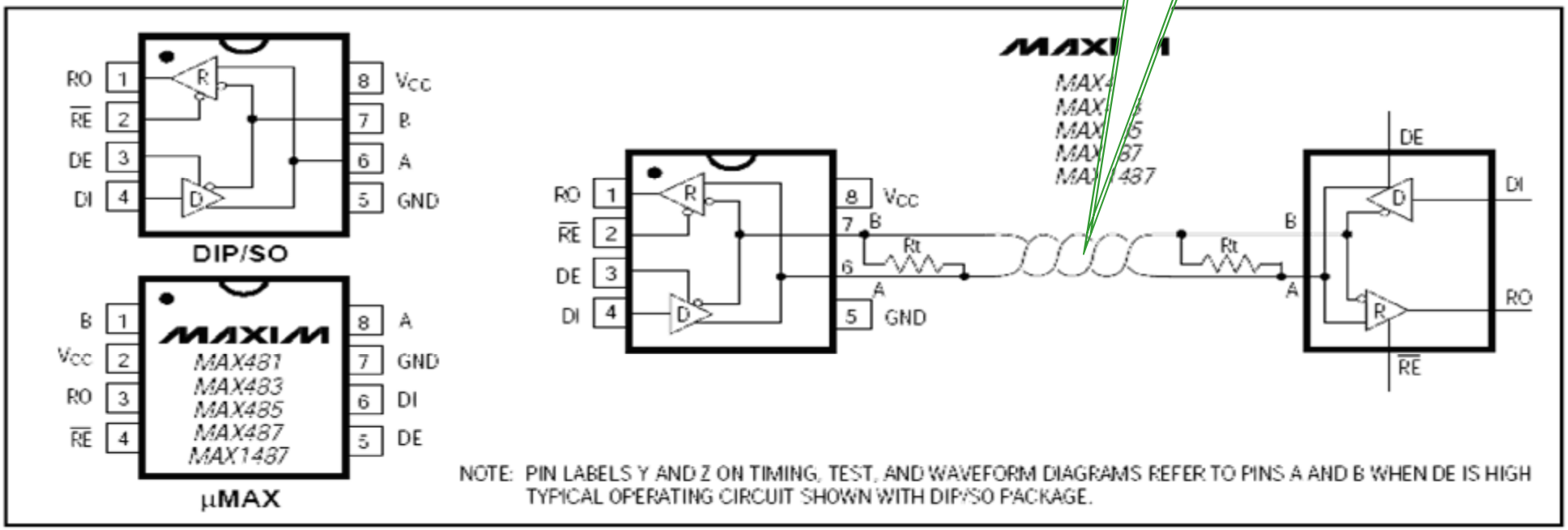



TTL/CMOS数据从T1IN、T2IN输入转换成RS-232数据从T1OUT、T2OUT送到电脑DB9插头；
DB9插头的RS-232数据从R1IN、R2IN输入转换成TTL/CMOS数据后从R1OUT、R2OUT输出。

# RS485：

双绞线或屏蔽线

RS485无具体的物理形状，根据工程的实际情况而采用相应的接口。



# (2) 接口的电气特性

## RS232：传输电平信号

接口的信号电平值较高（信号“1”为“-3V至-15V”，信号“0”为“3至15V”），易损坏接口电路的芯片，又因为与TTL 电平(0~“<0.8V”，1~“>2.0V”)不兼容故需使用电平转换电路方能与TTL电路连接。

抗干扰能力差。

# RS485：传输差分信号

逻辑“1”以两线间的电压差为+（2—6） V表示；逻辑“0”以两线间的电压差为-（2—6）V表示。接口信号电平比RS-232降低了，就不易损坏接口电路的芯片，且该电平与TTL电平兼容，可方便与TTL 电路连接。

抗干扰能力强。

# （3）通讯距离

RS232传输距离有限，最大传输距离标准值为15米，实际上也只能用在25米左右。

RS485最大传输距离标准值为120米，实际上可达 3000米。

# （4）多点通讯支持

- RS232接口在总线上只允许连接1个收发器，不能支持多站收发能力。

- RS485接口在总线上是允许连接多达128个收发器。即具有多站通讯能力，这样用户可以利用单一的RS485接口方便地建立起设备网络。

# （5） 通信线

- RS232可以采用三芯双绞线、三芯屏蔽线等。
- RS485可以采用两芯双绞线、两芯屏蔽线等。

双绞线的内部结构：
两线相绞




电缆线的内部结构：
两线平行

# （6）传输数据的最大波特率

- RS232传输速率较低，最高波特率为19200bps。虽然传输速度很慢，但在很多场合还是很实用。
- RS485的数据最高传输速率为10Mbps 。传输速度快了很多！

# 05 END

中南大学 CENTRAL SOUTH UNIVERSITY

# THANKS


## 《单片机与接口技术》

主讲人：李刚