

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：

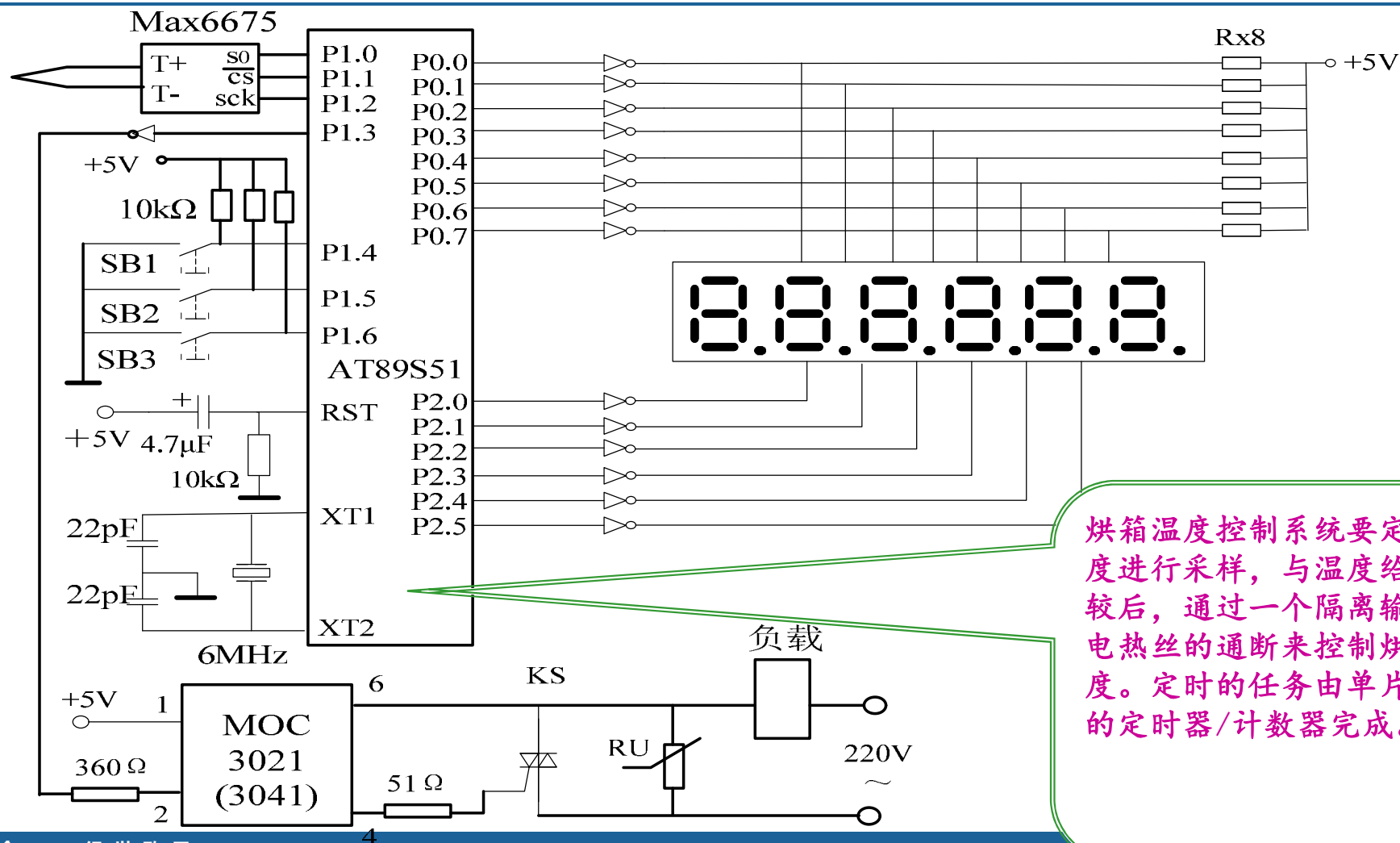
- 1、定时/计数的功能及工作方式是可编程的。即可选它作定时器，也可作计数器，同时，可选择几种工作方式。
- 2、定时/计数值在规定的范围内是编程的。即根据需要，通过软件来设定定时值或计数值。
- 3、当达到定时/计数的设定值时，应向CPU申请中断，以便实现定时/计数控制。

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），既可作为定时器，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。

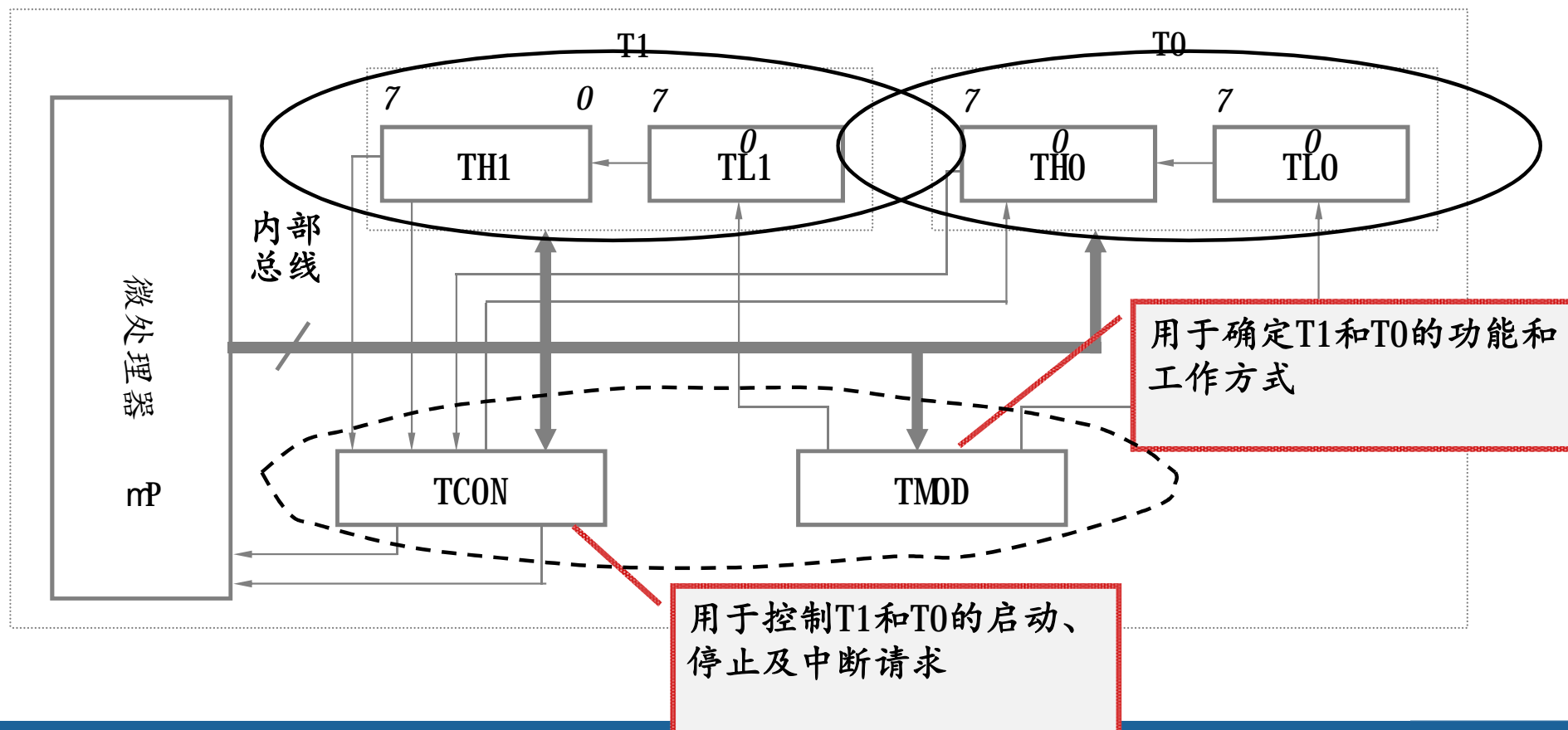
烘箱温度控制系统



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

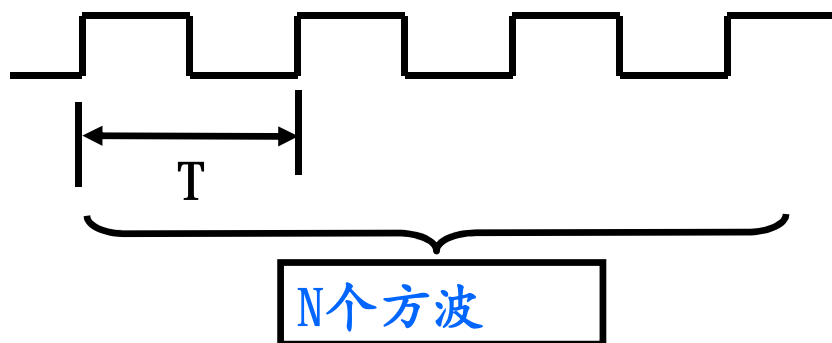
1、定时器/计数器T0、T1的结构



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： $f_{osc} / 12$ ($= 1 / T$) 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 $f_{osc} / 24$ 。

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

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。

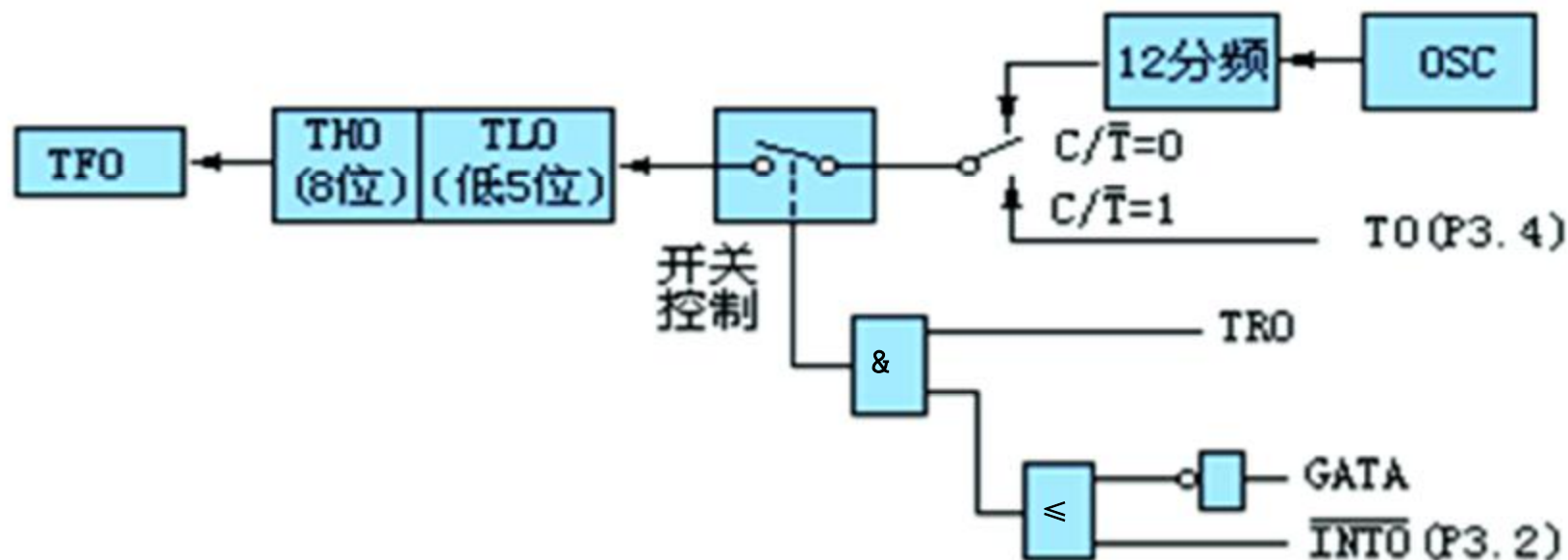
TR0: 功能与TR1类似。

定时器0
控制位

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



当在定时工作方式时，定时器的计数范围是：
 (8198192计数值) ÷ 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 $f_{osc}=6\text{MHz}$ 。

编程思路: P1.0输出周期为10ms宽的方波, 只要每10ms的方法, 由此可选用T0定时5ms

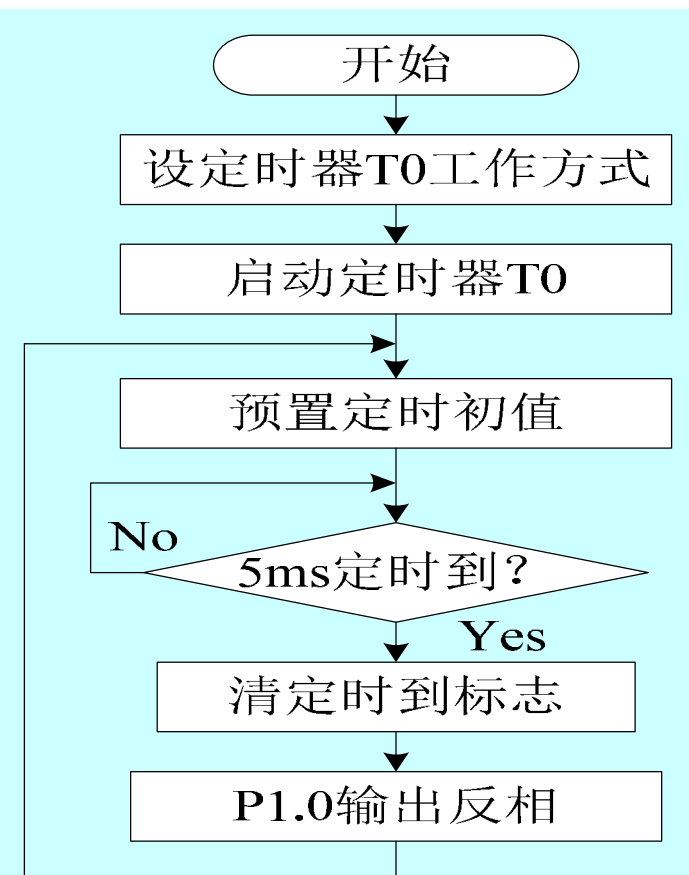
初值为:

```

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
  
```

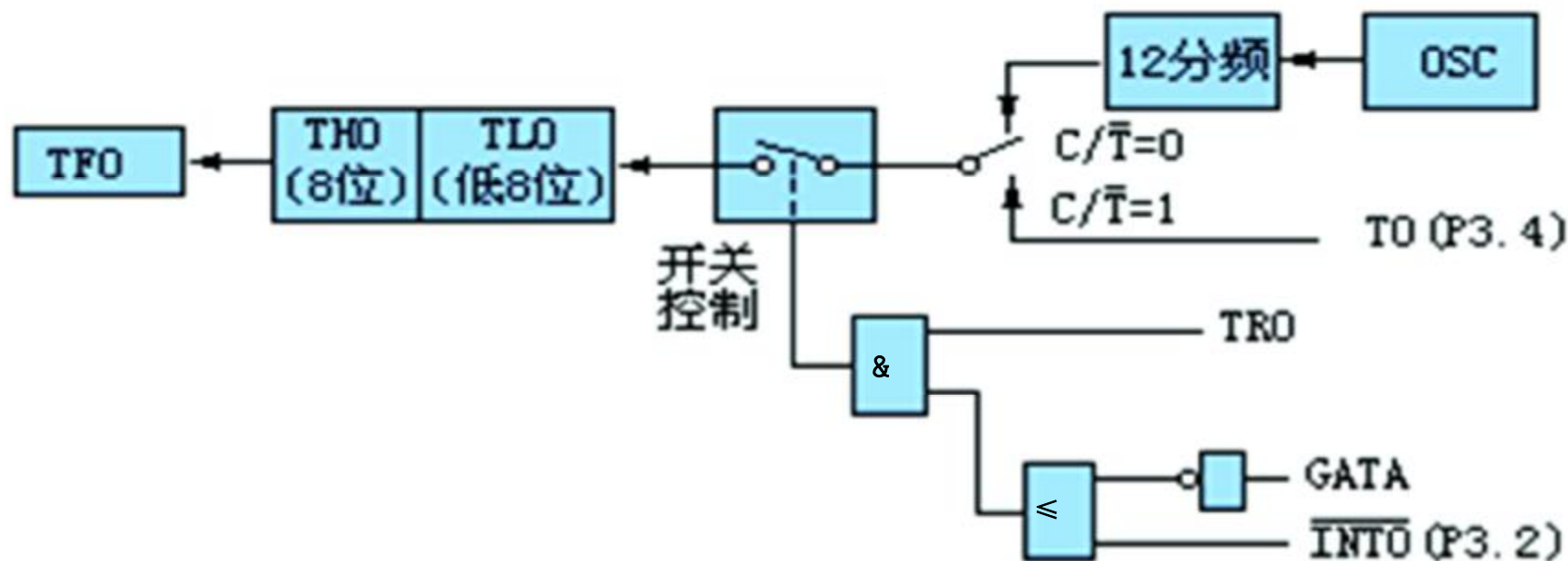
162

110



2、工作方式1

T0的等效逻辑结构



当为定时操作方式时,计数器的计数范围是:

(65536~65535) 或 初值的18次晶振周期×12

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, $f_{osc}=12\text{MHz}$ 。

编程思路: 方波周期 $T=1/50=0.02\text{s}=20\text{ms}$, 用T1定时10ms。

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

```

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

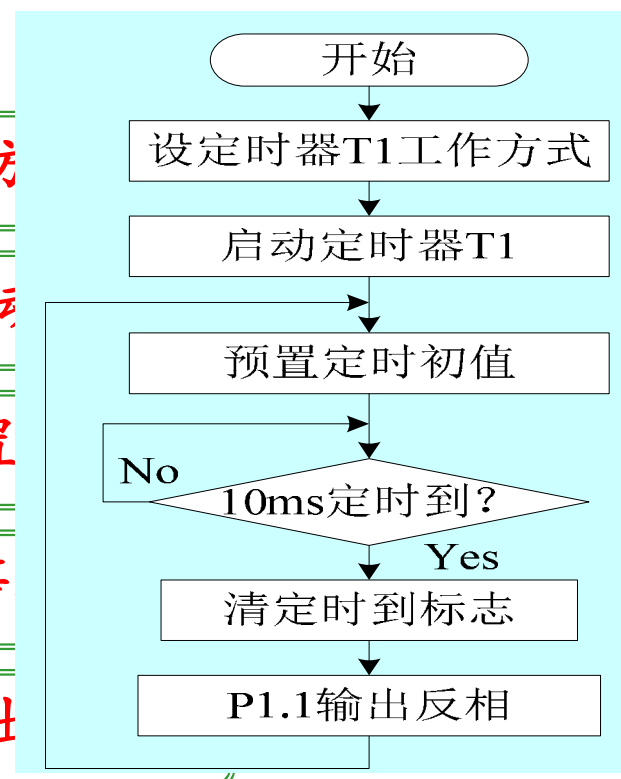
T1方

启动

重置

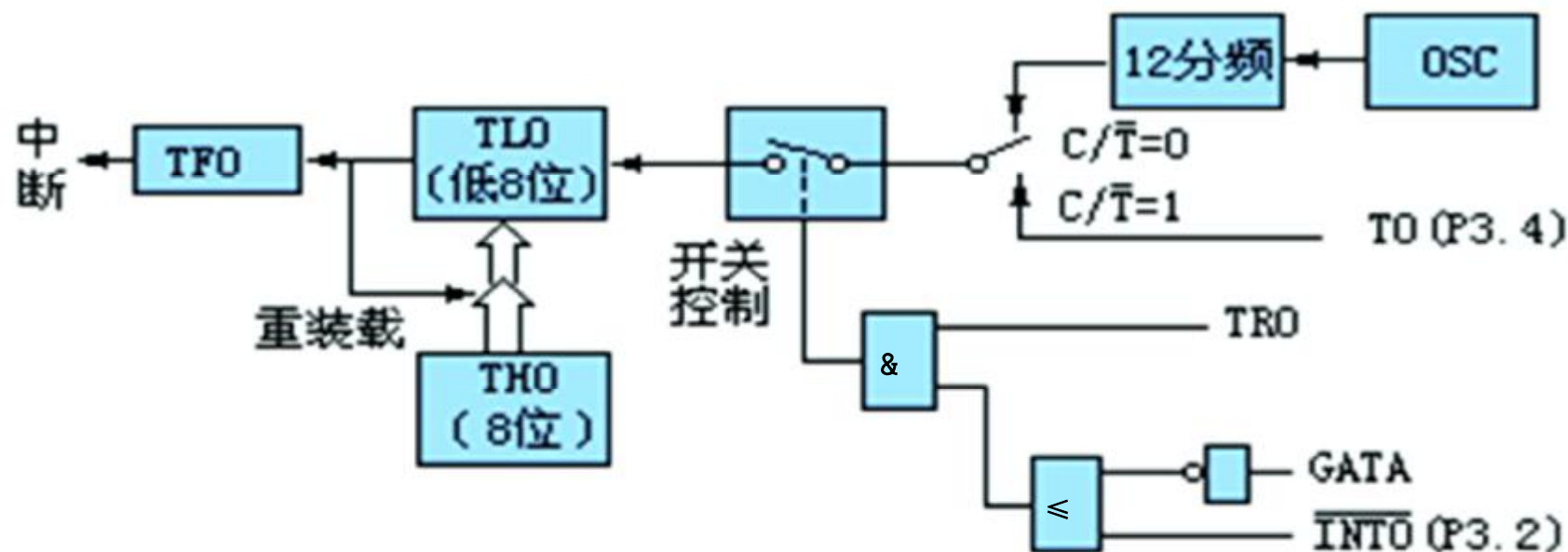
等

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 $T = (256 - \text{计数初值}) \times \text{晶振周期} \times 12$

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，每来一个计数脉冲，TF1的状态。

```

ORG 0100H
MOV TMOD, #60H
计数初值: 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：

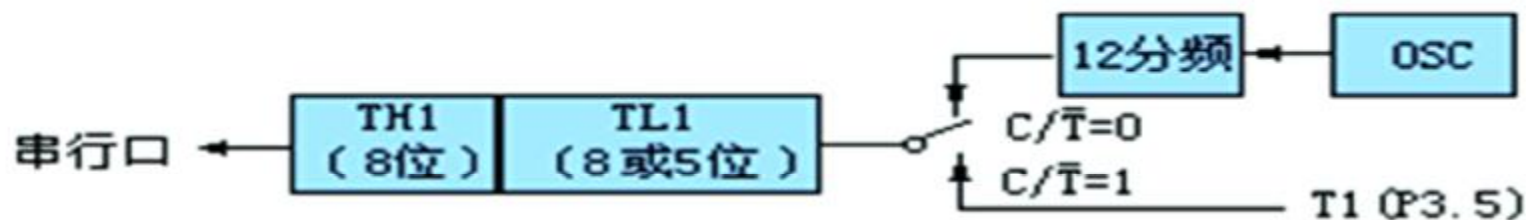
方式0方式1： 计数溢出后，计数器为全0，因而循环定时或循环计数应用时就存在反复设置初值的问题，这给程序设计带来许多不便，同时也会影响计时精度。

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。

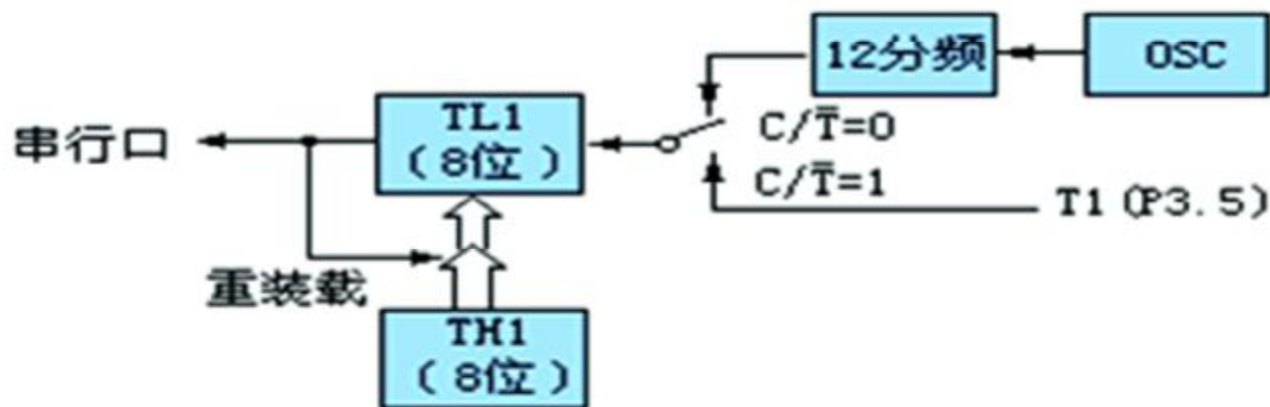
方式2在串口通讯时，常用作波特率发生器。

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：



T1工作方式0 (或1)



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：

$X = M - \text{计数值}$

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
```

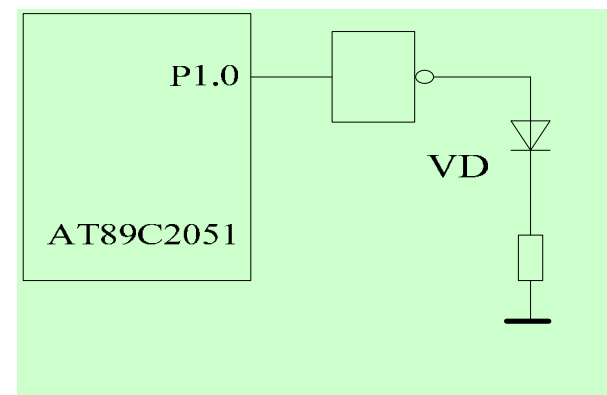
```
MOV    TH0, #4CH
```

```
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 $f_{osc}=12\text{MHz}$ ，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:

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

```
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:

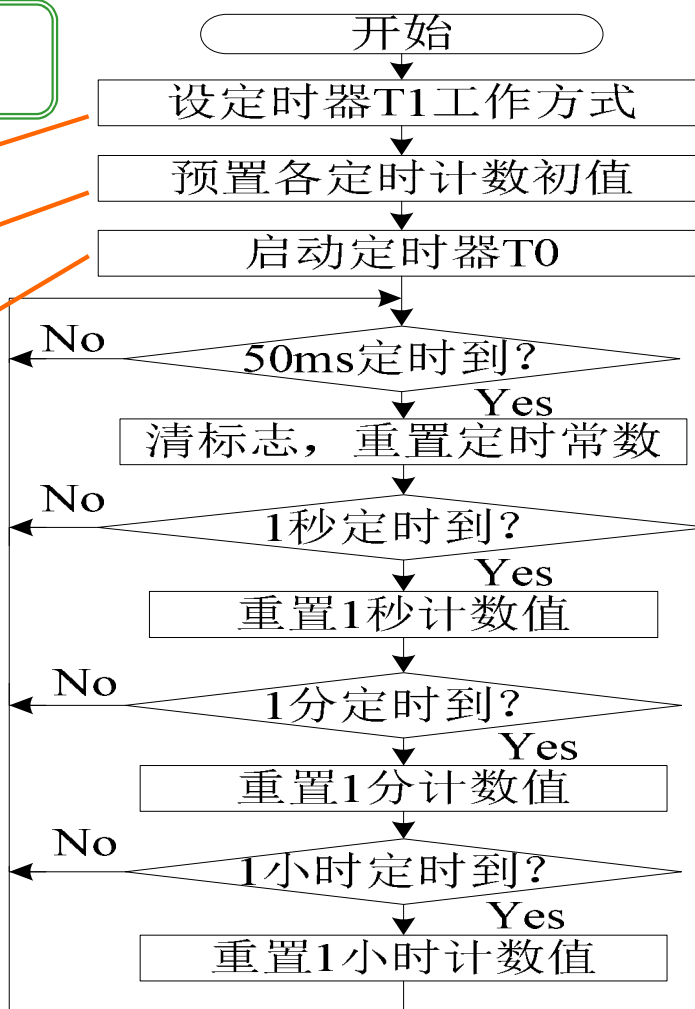
```
ORG 0000H
MOV TMOD,#10H
MOV TH0,#15536 MOD 256
MOV TL0,#15536/256
MOV 50H,#20
MOV 51H,#60
MOV 52H,#60
SETB TR1
```

方式1定时

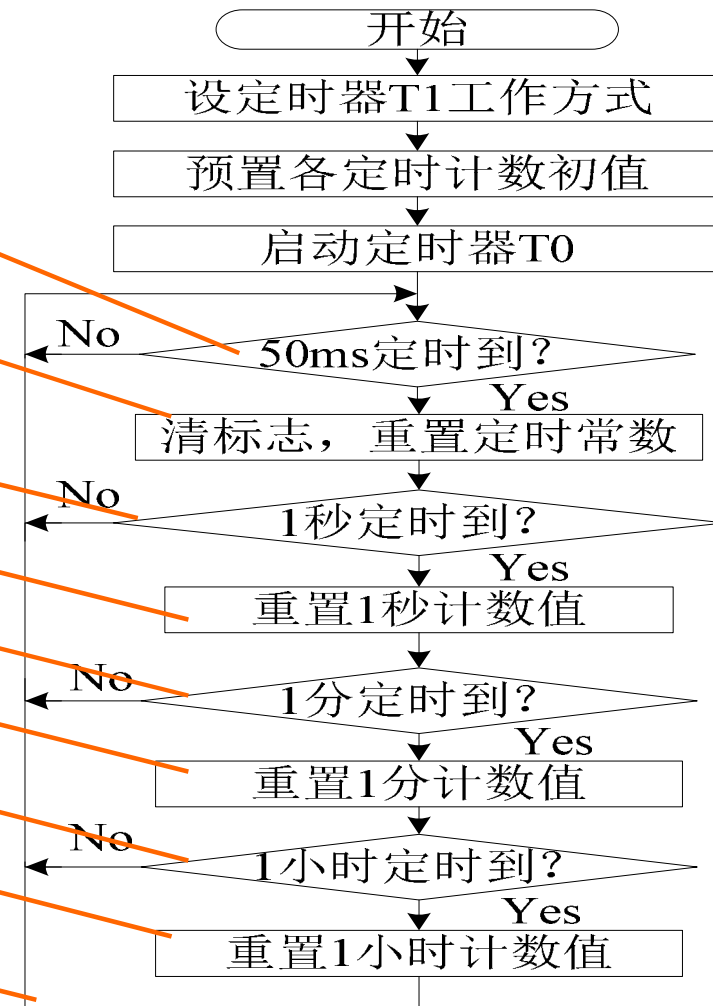
时间常数

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
```

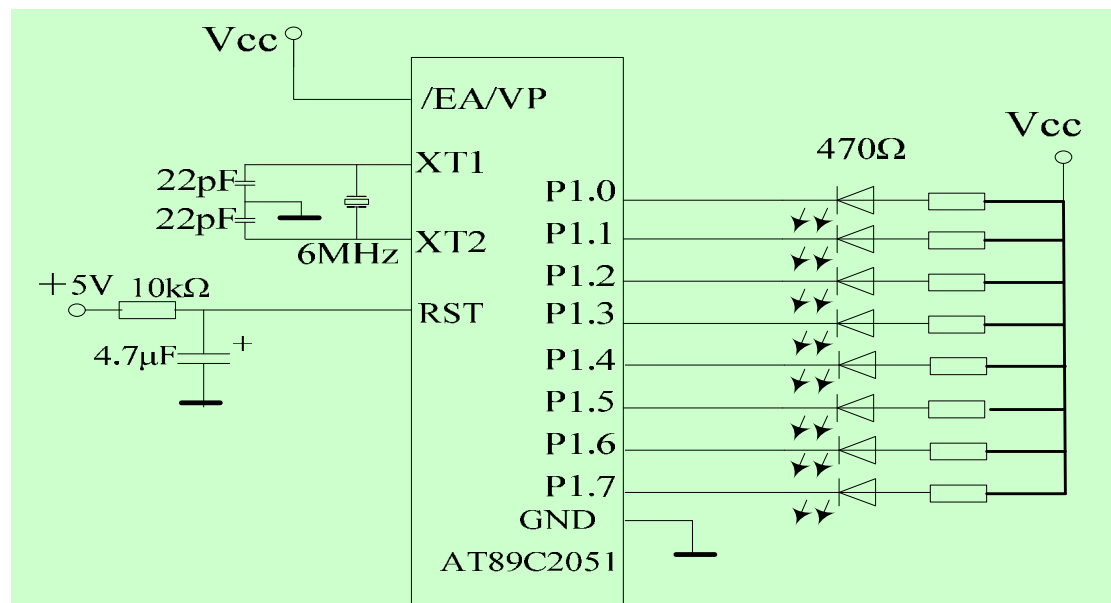


例3：如图开始时P1.0亮，延时0.2秒后左移至P1.1亮，如此左移7次至P1.7亮，再延时0.2秒右移至P1.6亮，如此右移7次后至P1.0亮。

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：

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



主程序:

```
ORG 0000H
MOV TMOD, #01
START: MOV A, #0FEH
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。

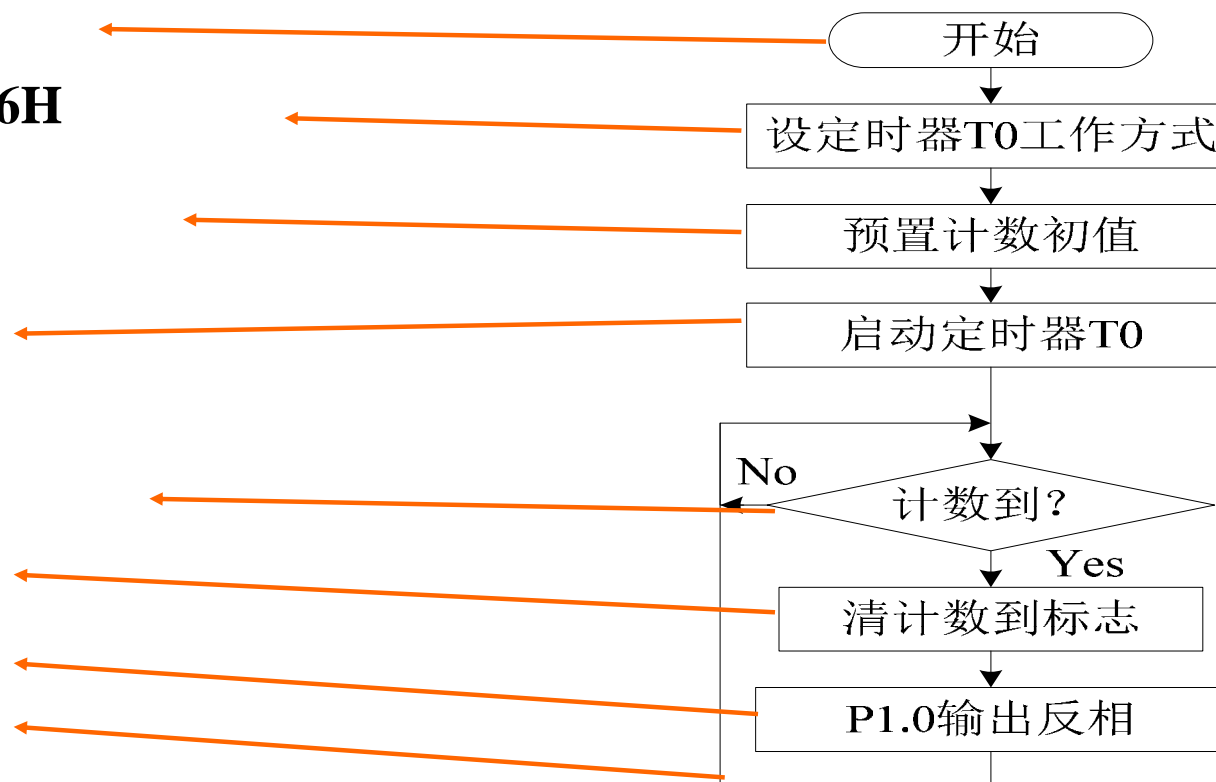
分析: 定时器T0工作在计数方式，计数3次将P1.0输出取反一次。由于计数初值小，因此采用方式2最好。

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

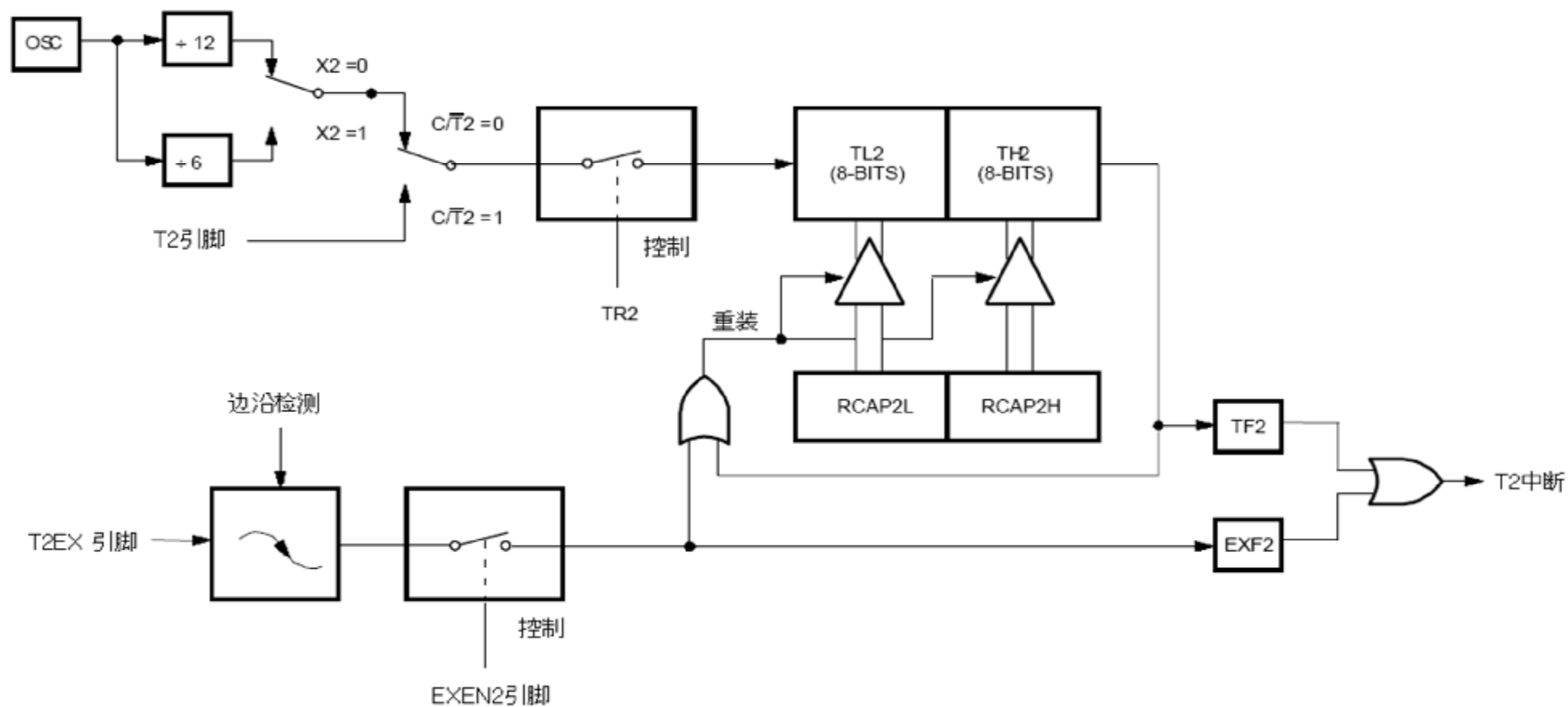
定时器 2 工作方式

RCLK+TCLK	CP/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)

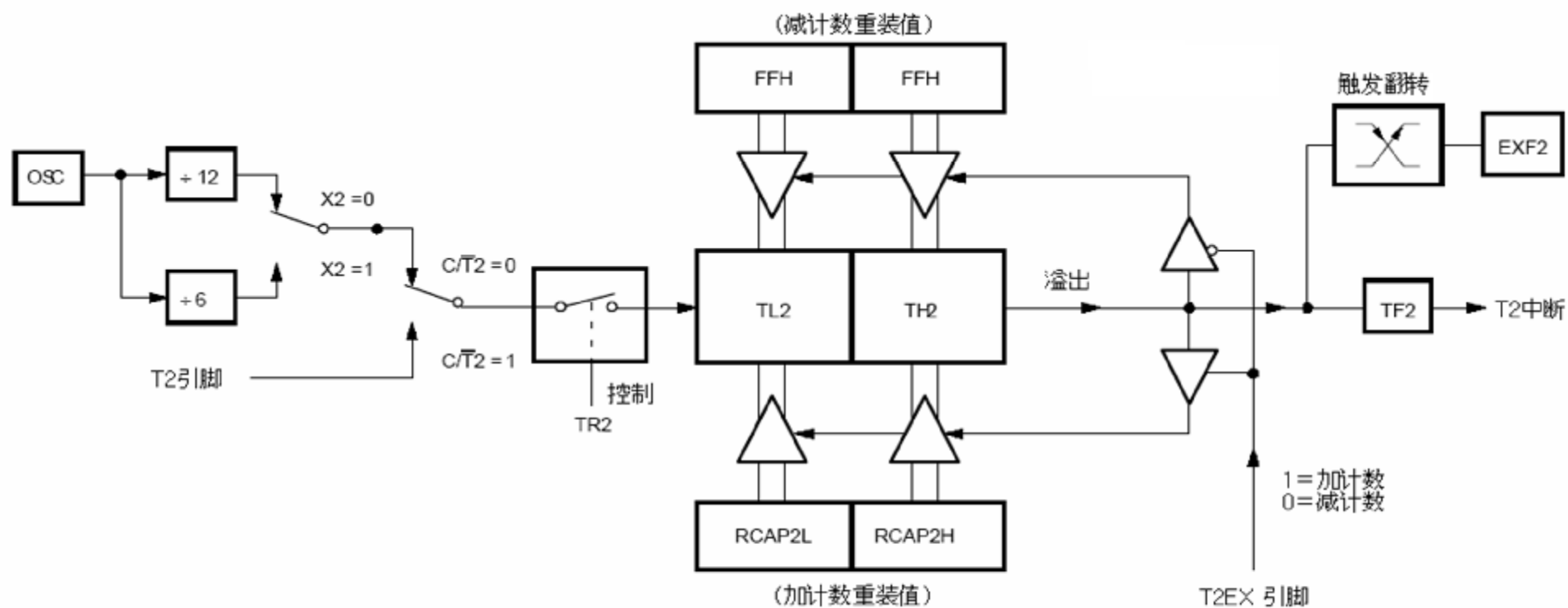
1) 定时/计数自动重装方式 (递增/递减计数器)

- 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。



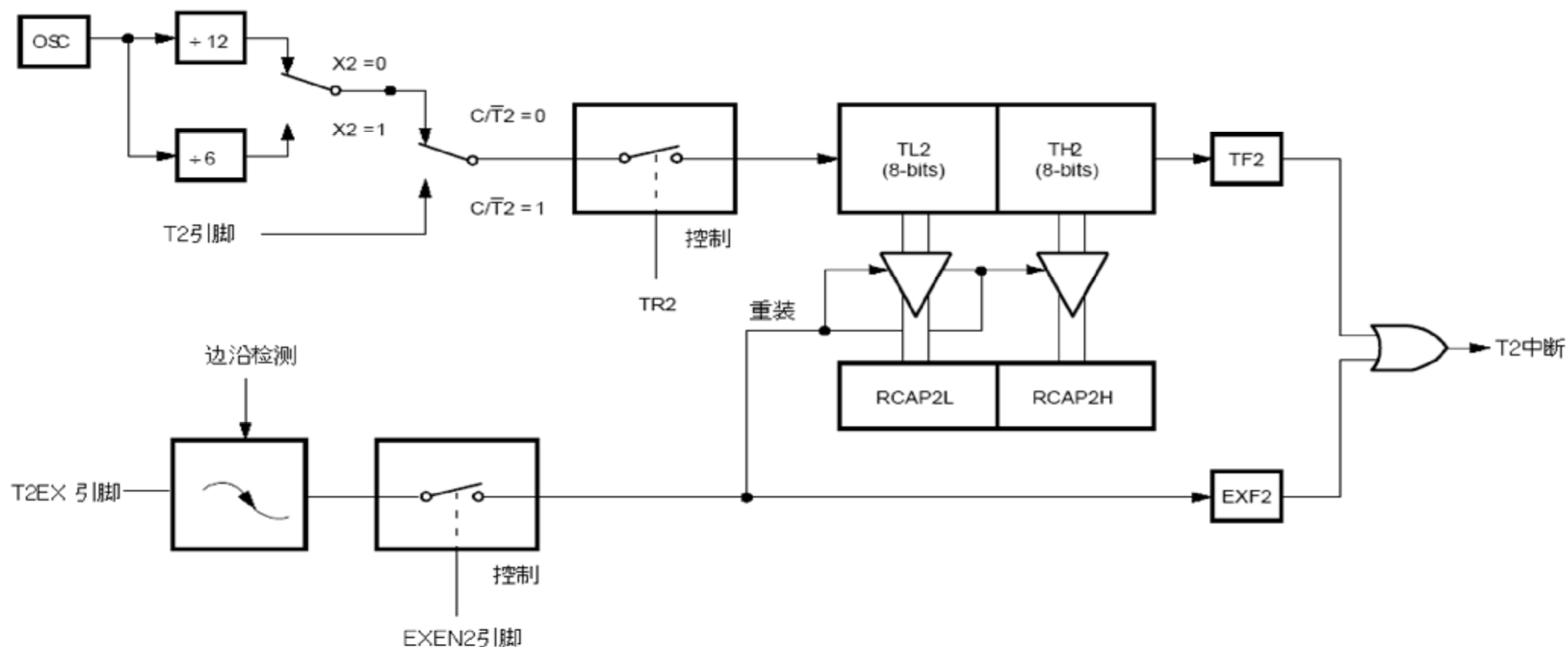
定时器 2 自动重装模式 (DCEN=0)



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。



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。