单片机定时/计数器

信息工程学院　　陈梅

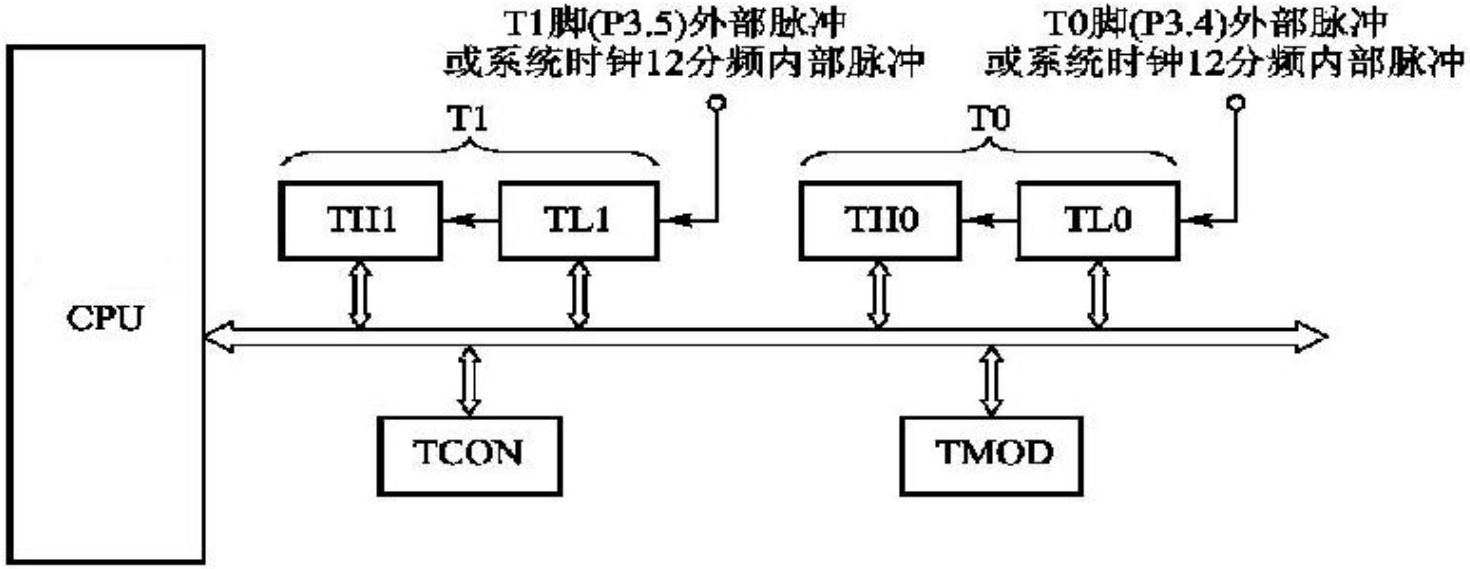
STC89C52单片机定时计数器 要点

要点：

- 定时/计数器的组成与功能
- 定时/计数器的工作模式和工作方式
- 定时/计数器的计数原理
- 定时/计数器的初始化
- 定时/计数器的编程及应用实例

定时/计数器工作原理，初值设置

●工作原理：核心部件是加法计数器，对脉冲进行加法计数，即每来一个脉冲，计数器加1，当加到全1时，再输入一个脉冲，计数器回0的同时产生的溢出信号，使用TCON中TF0或TF1置1，即表示定时时间已到或计数值已满。



◆ 定时/计数器初值由**THx和TLx**存放，从初值开始计数，直到计满溢出为止。

故：**初值X = 最大值M - 计数次数N**

计数原理：初值X = 最大值M - 计数次数N

- ◆最大值M与工作方式有关，由TMOD设置。
- ◆计数次数N与计数脉冲来源有关，由TMOD设置。
- ◆计数初值X存放在THi和TLi中。
- ◆启动定时/计数器工作，由TCON设置。

符号	描述	地址	位地址及其符号 MSB LSB								复位值
TCON	定时器控制寄存器	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000 0000B
TMOD	定时器模式寄存器	89H	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	0000 0000B
TL0	Timer Low 0	8AH									0000 0000B
TL1	Timer Low 1	8BH									0000 0000B
TH0	Timer High 0	8CH									0000 0000B
TH1	Timer High 1	8DH									0000 0000B

只有TCON可位寻址，但TMOD、THx、TLx均不可。

初值X = 最大值M - 计数次数N

◆最大值M与工作方式有关，由TMOD设置。

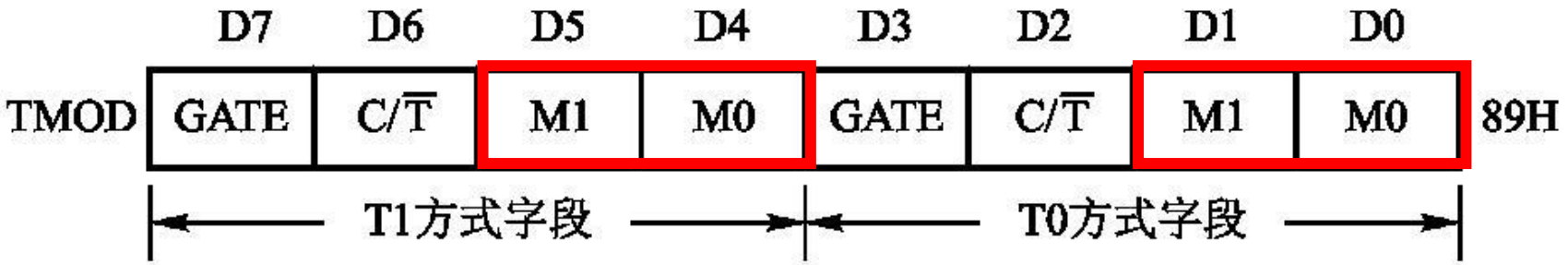


表 6-1　M1、M0 工作方式选择

M1	M0	工 作 方 式
0	0	方式 0，为 13 位定时器/计数器
0	1	方式 1，为 16 位定时器/计数器
1	0	方式 2，8 位的常数自动重新装载的定时器/计数器
1	1	方式 3，仅适用于 T0，此时 T0 分成两个 8 位计数器， T1 停止计数

方式0：$M=2^{13}=8192$

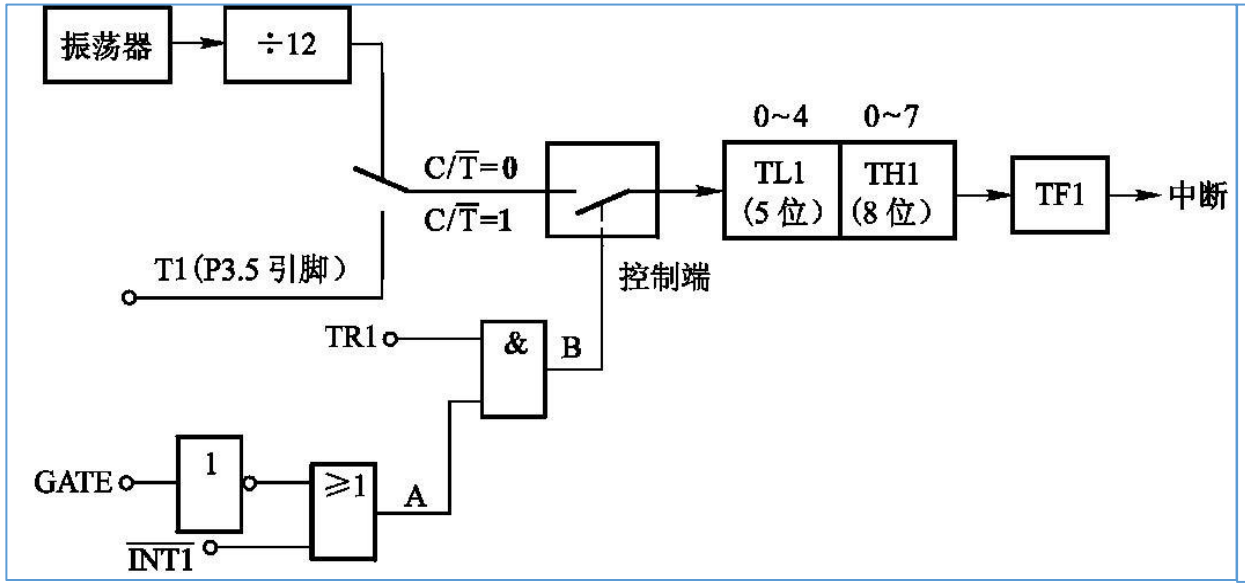
方式1：$M=2^{16}=65536$

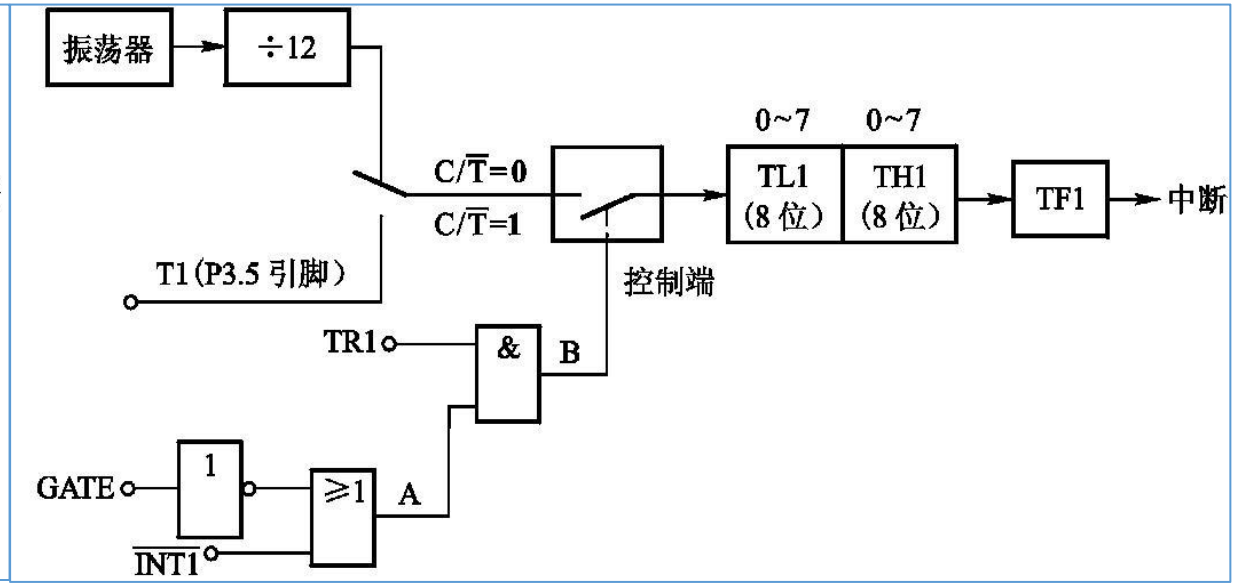
方式2：$M=2^{8}=256$

方式3：$M=2^{8}=256$

定时/计数器的工作方式0和工作方式1

- 方式0-----13位计数器
- 方式1——16位计数器



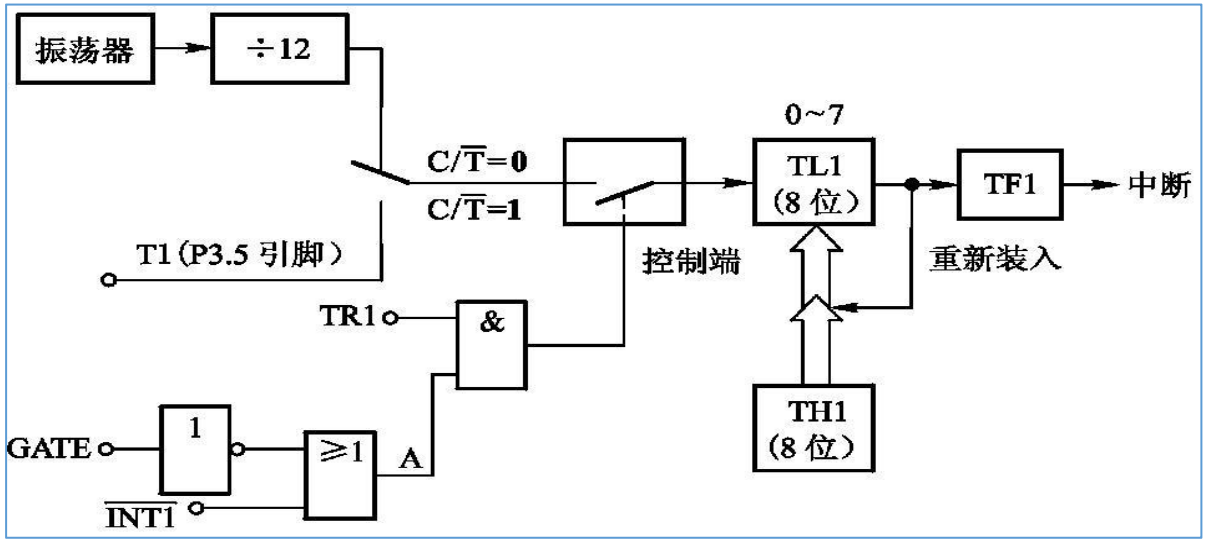


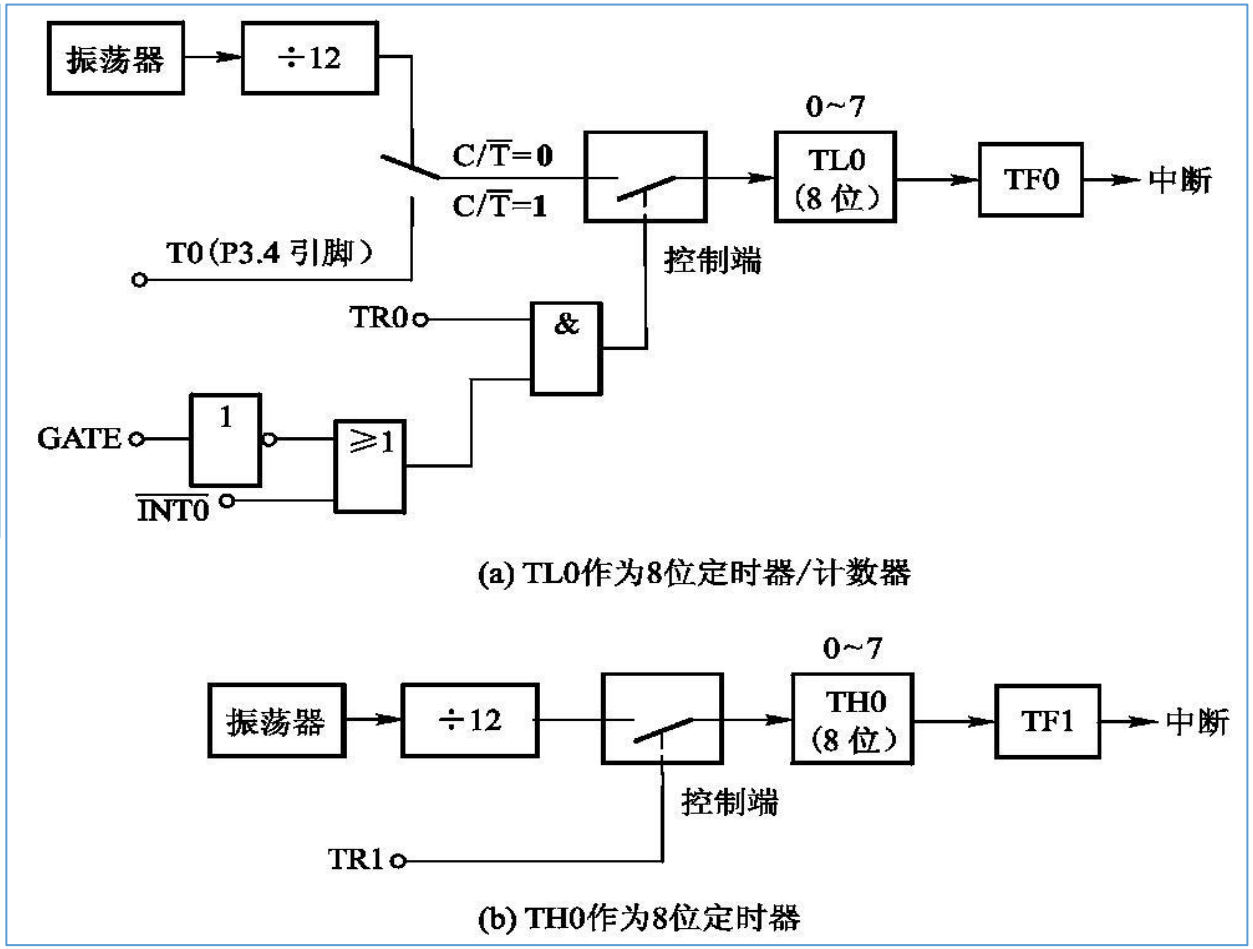
方式0由TLx低5位和THx高8位构成；方式1由THx高8位和TLx低8位构成。

两种方式的差别仅仅在于计数位数不同。

定时/计数器的工作方式2和工作方式3

- 方式2——8位计数器,初值自动重装
- 方式3——分为两个8位定时/计数器





初值X = 最大值M - 计数次数N

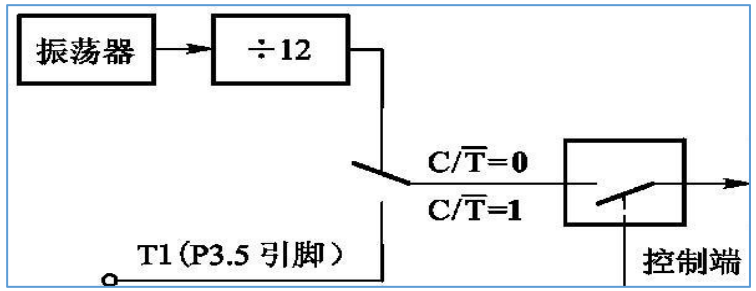
◆ 计数次数N与计数脉冲来源有关，由TMOD设置。

- 根据计数来源的不同，具有定时和计数两种模式，本质是相同的。

– **定时模式：计数脉冲来自单片机内部系统脉冲，**每经过1个机器周期计数值加1，直至计满溢出为止。即通过计机器周期个数来实现，即 **初值X =最大值M－$T_{定时}/T_{机器}$**

– **计数模式：计数脉冲来自外部引脚（P3.4-T0,P3.5-T1),** 每来1个计数脉冲加1，即外部引脚电平跳变一次，则计数器加1。即**初值X=最大值M－计数值N**

	D7	D6	D5	D4	D3	D2	D1	D0	
TMOD	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	89H
	← T1方式字段 →				← T0方式字段 →				



工作方式寄存器TMOD　89H　不可位寻址

- **GATE——门控位**

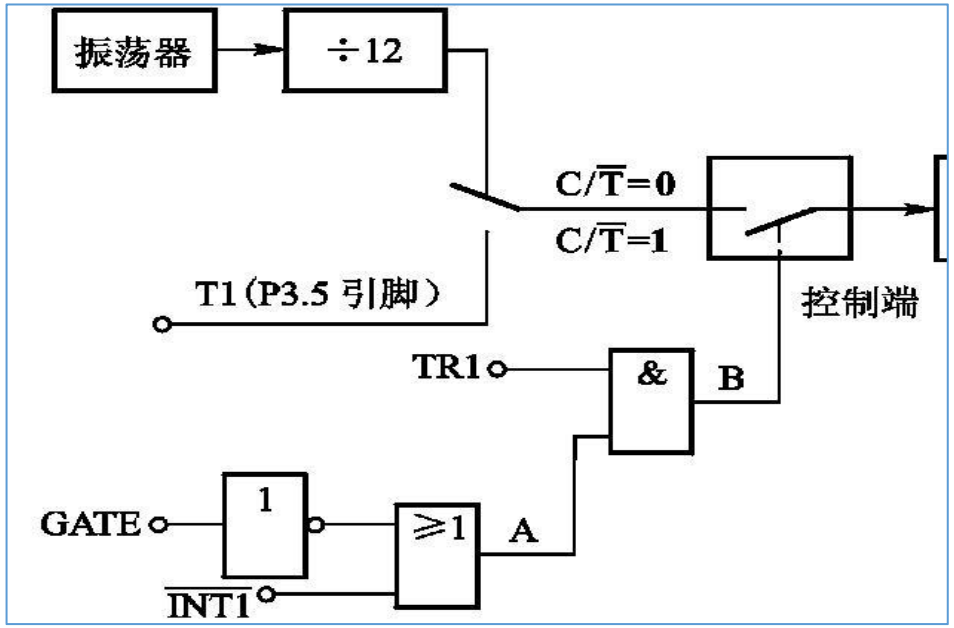
	D7	D6	D5	D4	D3	D2	D1	D0	
TMOD	GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0	89H
	← T1方式字段 →				← T0方式字段 →				

GATE=1时，由外部中断引脚INTx的电平与运行控制位TRx共同控制运行。

即若TR_1=1并且$\overline{INT_1}$=1，允许T1计数；若TR_1=0或$\overline{INT_1}$=0，禁止T1计数

GATE=0时，由运行控制位TRx单独控制运行。

（默认）即若TR_1=1允许计数；若TR_1=0禁止计数



定时/计数控制寄存器TCON　88H 可位寻址

TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

- **TR1、TR0**——计数运行控制位

  为1，启动定时/计数　　SETB　TRx

  为0，停止定时/计数　　CLR　TRx

- **TF1、TF0**——当计数溢出时，该标志位自动置“1”。
  - 使用查询方式编程时，在查询有效后，应使用软件指令及时将该位清“0”。
  - 使用中断方式编程时，在中断请求有效后，进入中断服务程序后由硬件自动清“0”。

定时/计数器的初始化

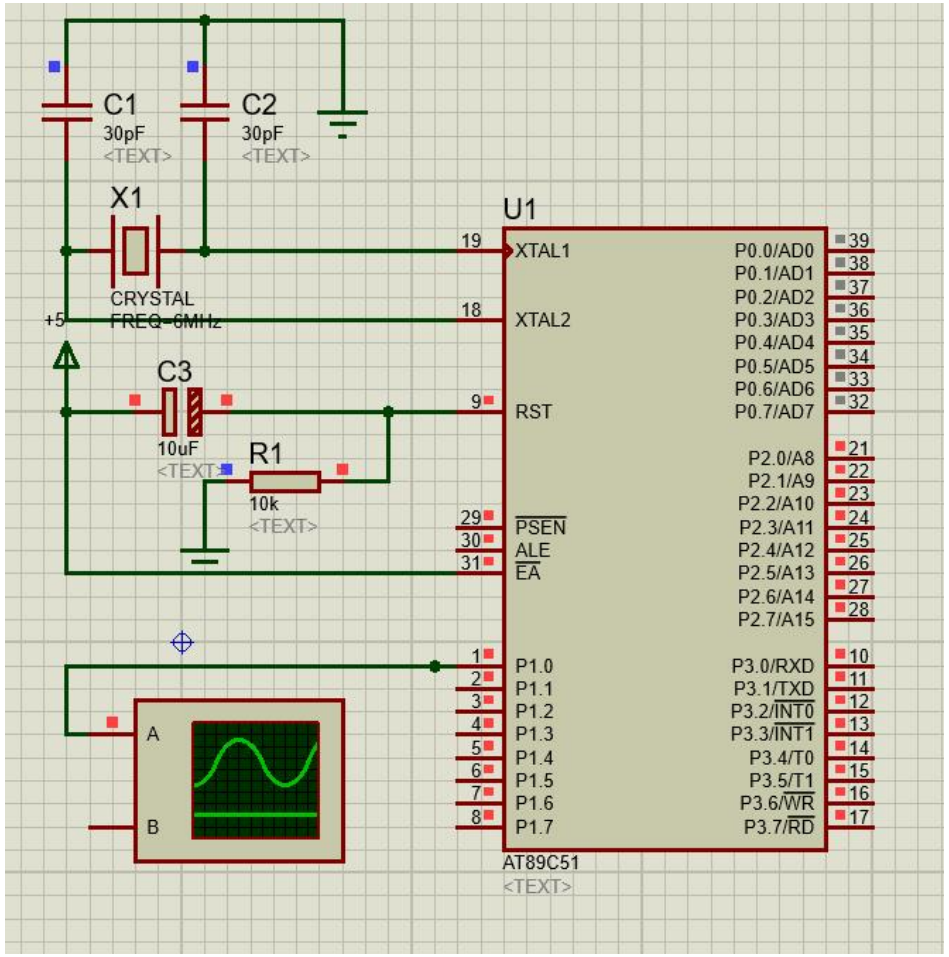
定时/计数器是一种可编程部件，当启动定时/计数器工作后，定时器就会按设定的工作方式独立运行，不占用CPU的时间，除非定时计数溢出，才可能中断CPU当前操作。

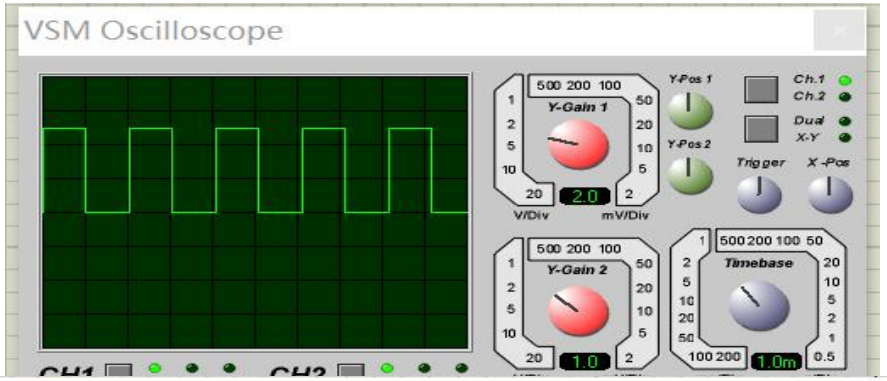
初始化的步骤如下：

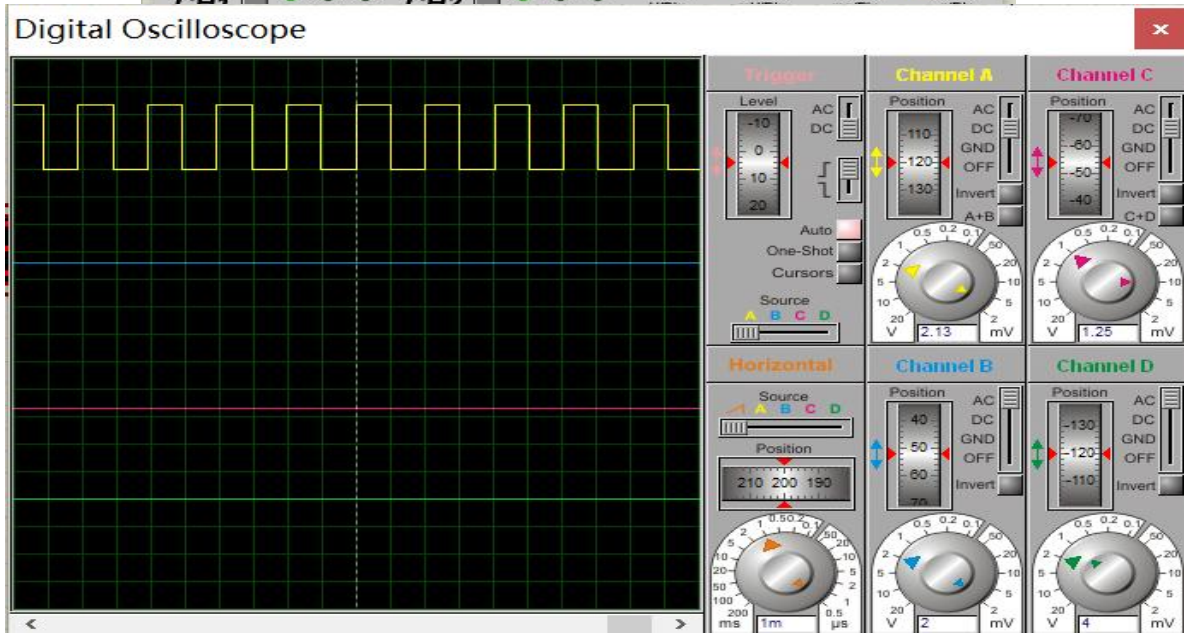
1) 确定计数来源算出计数次数N----------设置TMOD的C/T#
2) 确定工作方式即最大值M----------设置TMOD的M1M0
3) 根据X=M-N计算出初值X----------对THi,Tli赋值
4) 根据需要开放总中断和定时中断---------设置IE的EA和ETx
5) 启动定时器/计数器----------设置TCON中的TRi

定时/计数器0或1的工作方式1的应用

【例6.1】STC89C52单片机系统时钟频率F_{CLK}为6MHz，要在$P_{1.0}$引脚上输出1个周期为2毫秒、占空比为50%的方波信号。

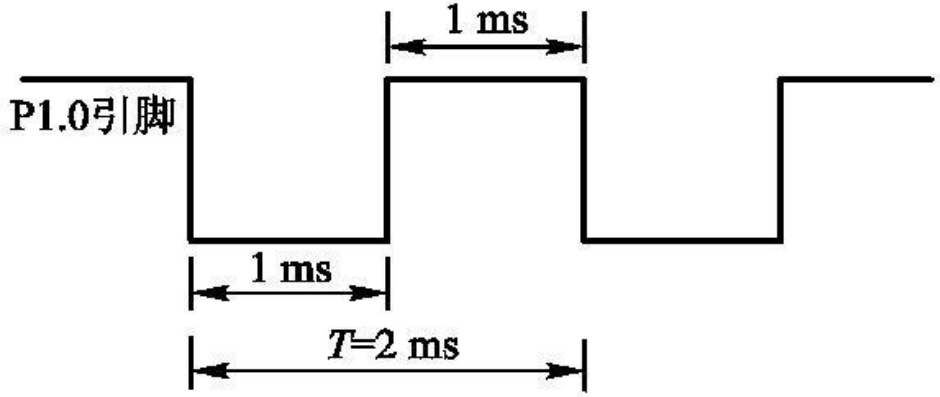






定时/计数器1的工作方式1的应用

【例6.1】STC89C52单片机系统时钟频率F_{CLK}为6MHz，要在$P_{1.0}$引脚上输出1个周期为2毫秒、占空比为50%的方波信号。



- **确定计数来源于内部脉冲N=$T_{定时}$/$T_{机器}$**

每隔1ms电平取反一次即$T_{定时}$=1ms

F_{CLK}为6MHz，$T_{机器}$=12/F_{clk}=12/6MHz= 2μs

计数次数N=1ms/ 2μs=500

方式0计算X=M-N（不常用）

X=2^{13}-500=8192-500= 1E0CH

=0001 1110 0000 1100B

则 TH0=1 1110 000B=0F0H,

TL0=0 1100B =0CH

- 确定工作方式1得最大值M=2^{16}
- 计算初值X=M-N

X=2^{16}-500=65536-500= FE0CH

故 TH0=0FEH，TL0=0CH

采用T0定时器工作，初始化步骤：

1) 设置TMOD的C/T#和M1M0——即TMOD=0000 0001B=01H
2) 设置THx和TLx初值———————— 即TH0=0FEH,TL0=0CH
3) 设置IE的EA和ETx开中断————即EA=1,ET0=1
4) 设置TCON中的TRi启动工作——即TR0=1

初始化指令序列：

```
1)  MOV    TMOD，#01H
2)  MOV    TH0，#0FEH
    MOV    TL0，#0CH
3)  SETB   EA
    SETB   ET0
4)  SETB   TR0
```

定时器方式1

采用中断方式工作

```
        ORG     0000H
        LJMP    MAIN
        ORG     000BH  ;T0入口
        LJMP    T0INT
        ORG     0100H
MAIN:   MOV   SP,#60H        ; 设置堆栈指针
        MOV TMOD,#01H        ; 设置定时器工作方式
        MOV TH0,#0FEH        ; 设置定时初值
        MOV TL0,#0CH
        SETB  EA             ; 开放中断
        SETB  ET0
        SETB  TR0            ; 启动定时器
        SJMP   $             ; 等待中断
T0INT:  CPL P1.0             ;电平取反输出
        MOV TH0,#0FEH        ;重装初值
        MOV TL0,#0CH
        RETI
        END
```

```
#include  <reg52.h>
sbit      P10=P1^0;
void main( )
{
    SP=0x60;          //设置堆栈指钍
    TMOD=0x01;        //设置定时器工作方式
    TH0=0xFE;         //设置定时初值
    TL0=0x0C;
    EA=1;             //开放中断
    ET0=1;
    TR0=1;            //启动定时器
    while(1) {;}      //等待中断
}
void  timer0(void)  interrupt  1
{
    P10=!P10;         //电平取反输
    出TH0=0xFE;       //重装初值
    TL0=0x0C;
}
```

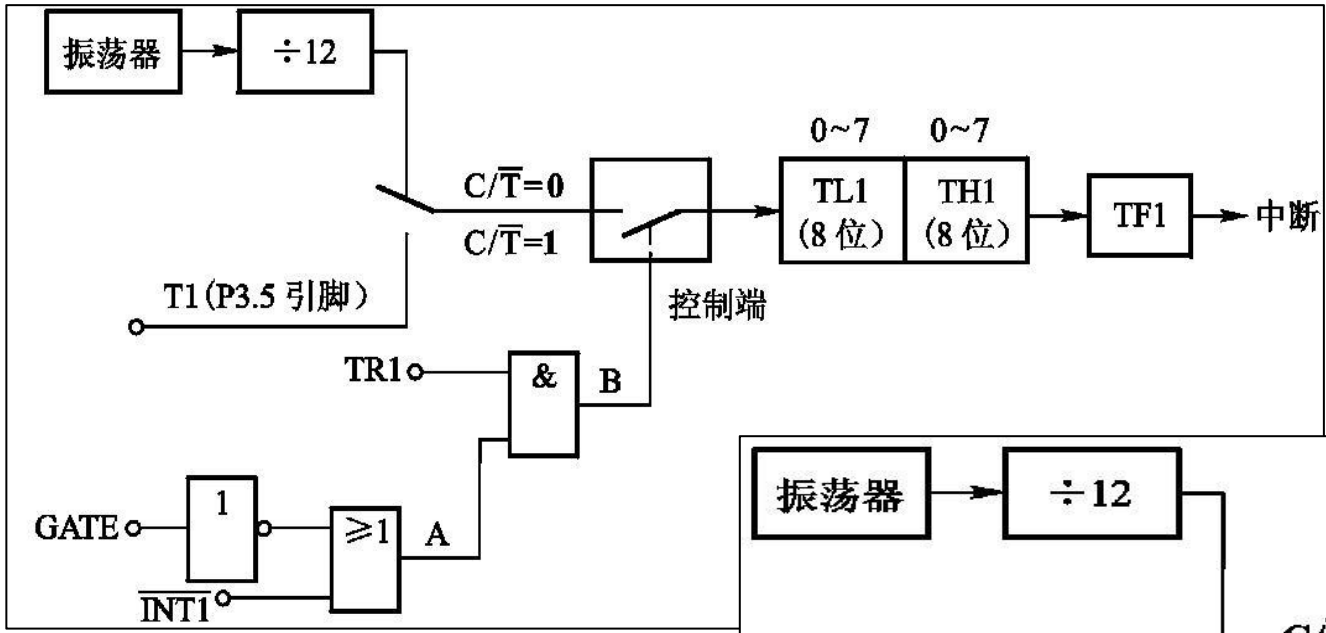
采用中断方式工作

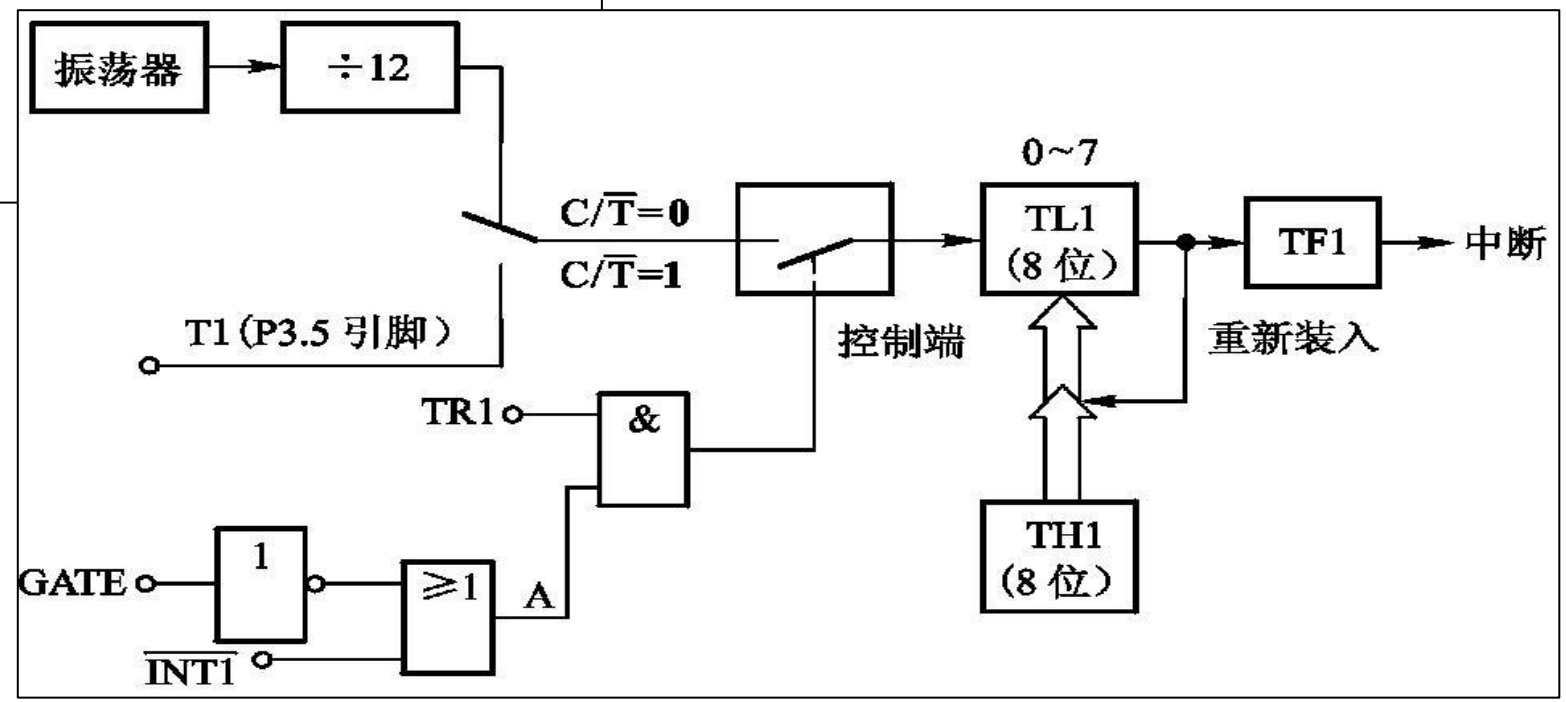
```
        ORG    0000H
        LJMP   MAIN
        ORG    000BH  ;T0入口
        LJMP   T0INT
        ORG    0100H
MAIN:   MOV    SP,#60H          ; 设置堆栈指针
        MOV TMOD,#01H           ; 设置定时器工作方式
        MOV TH0,#0FEH           ; 设置定时初值
        MOV TL0,#0CH
        SETB  EA                ; 开放中断
        SETB  ET0
        SETB  TR0               ; 启动定时器
        SJMP   $                ; 等待中断
T0INT:  CPL P1.0                ;电平取反输出
        MOV TH0,#0FEH           ;重装初值
        MOV TL0,#0CH
        RETI
        END
```

采用查询方式工作

```
        ORG    0000H
        LJMP   MAIN

        ORG    0100H
MAIN:
        MOV    TMOD, #01H
        MOV    TH0, #0FEH
        MOV    TL0, #0CH

        SETB   TR0
LOOP:   JNB    TF0, LOOP
        CLR    TF0;查询方式手动清除TF0
        CPL    P1.0
        SJMP   MAIN

        END
```

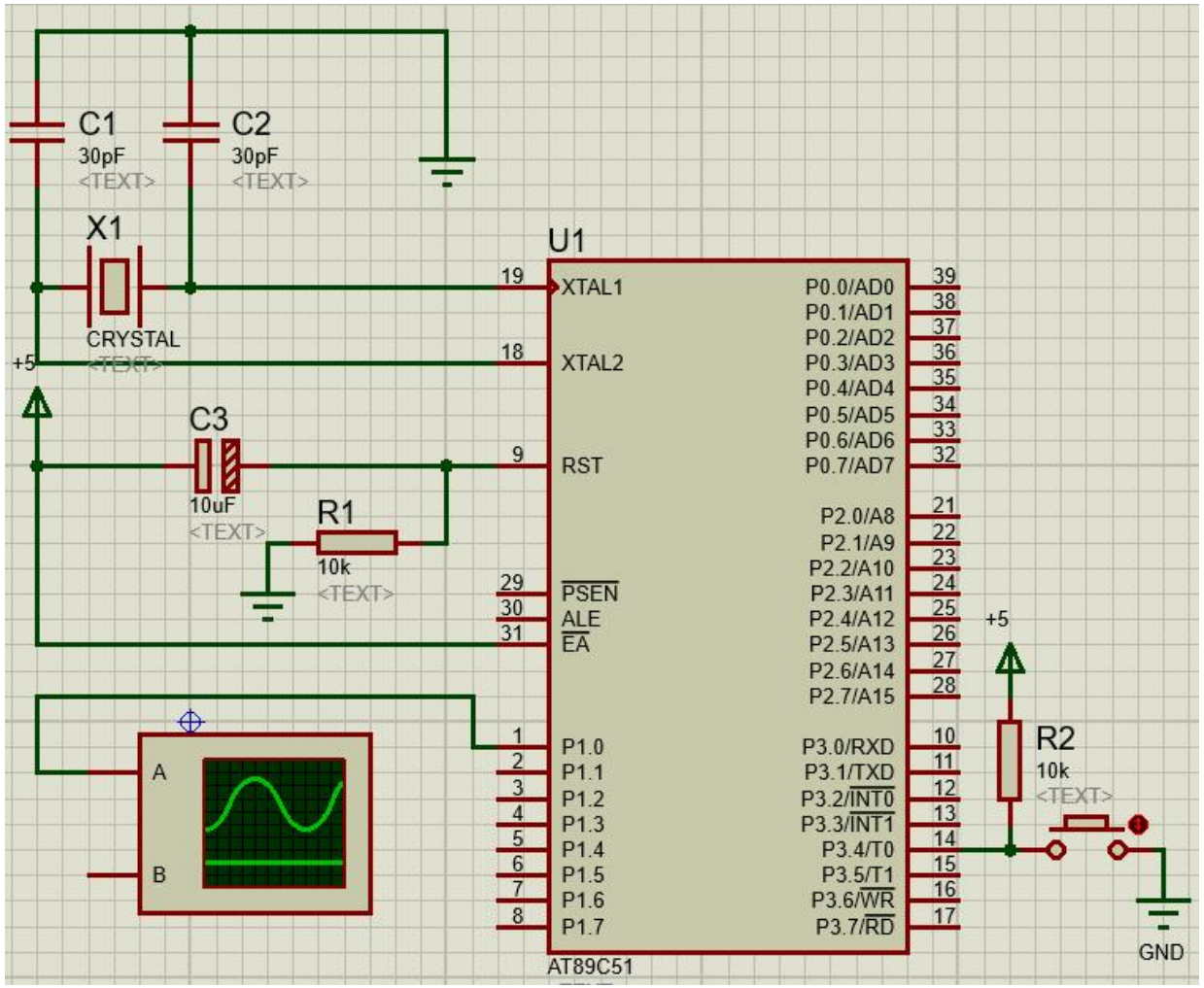
定时/计数器工作方式2与方式1对比

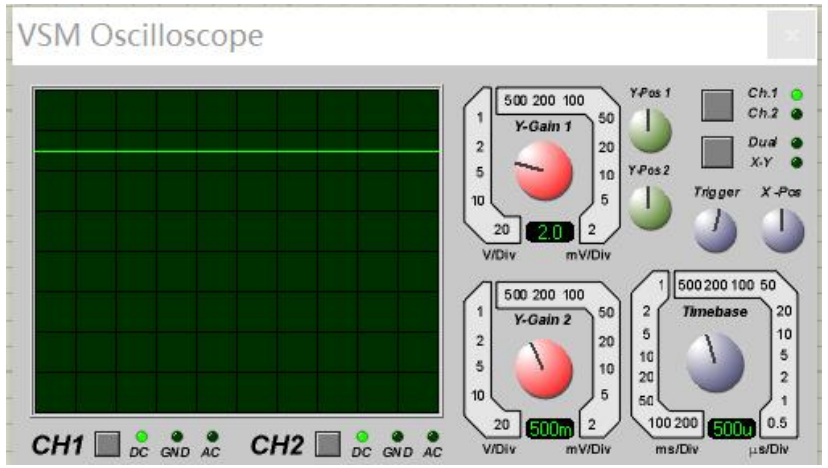


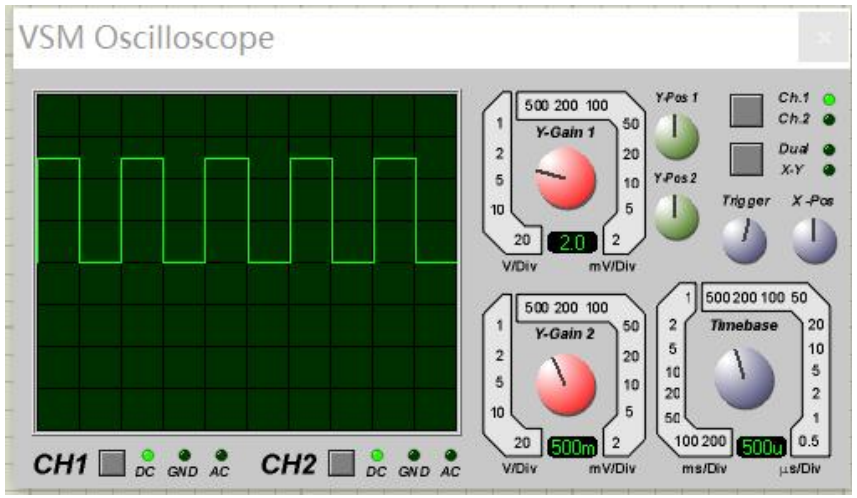


定时/计数器工作方式2的应用

【例6-3】将T0（P3.4）引脚上发生负跳变信号作为P1.0引脚产生方波的启动信号。要求P1.0脚上输出周期为1ms的方波，如图6-5所示（系统时钟6MHz）。

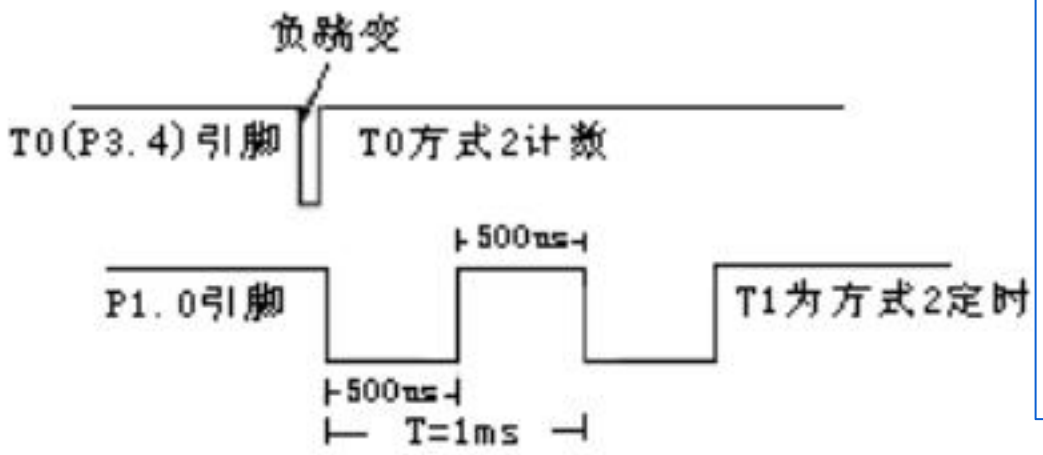






定时/计数器工作方式2的应用

【例6-3】将T0（P3.4）引脚上发生负跳变信号作为P1.0引脚产生方波的启动信号。要求P1.0脚上输出周期为1ms的方波，如图6-5所示（系统时钟6MHz）。



- **确定T0和T1计数来源**
  - ✓ T0计数来源于外部引脚N=1
  - ✓ T1计数来源于内部脉冲N=$T_{定时}/T_{机器}$
    =500μs/ 2μs=250
    (F_{CLK}为6MHz，$T_{机器}=12/F_{clk}=12/6MHz= 2\mu s$)

- 确定T0工作方式得最大值M
  T0方式2计数，$M=2^8$
- **计算初值X**=M-N=2^8-1
  X=256-1 = 255=0FFH
  故 TH0=0FFH，TL0=0FFH

- 确定T0工作方式得最大值M
  T1方式2定时，$M=2^8$
- **计算初值X**=M-N=$2^8-T_{定时}/T_{机器}$
  X=2^8-500μs/2μs=256-250 = 06H
  故 TH1=06H，TL1=06H

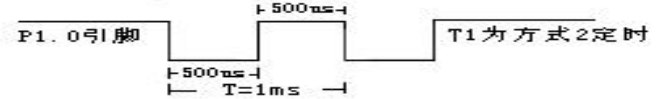
采用T0方式2计数，T1方式2定时工作，初始化步骤：

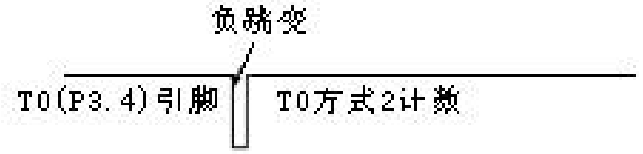
1) 设置TMOD的C/T#和M1M0———即TMOD=0010 0110B=26H
2) 设置THx和TLx初值———即TH0=0FFH,TL0=0FFH;TH1=06H,TL1=06H
3) 设置IE的EA和ETx开中断————即EA=1,ET0=1,ET1
4) 设置TCON中的TRi启动工作———即TR0=1,TR1=1

初始化指令序列：

```
1)  MOV   TMOD, #26H
2)  MOV   TH0, #0FFH
    MOV   TL0, #0FFH
    MOV   TH1, #06H
    MOV   TL1, #06H
3)  SETB  EA
    SETB  ET0
    SETB  ET1
4)  SETB  TR0
    SETB  TR1
```

先仿照例题6-1实现P1.0引脚的方波






```
        ORG   0000H
        LJMP  MAIN
        ORG   000BH   ;T0入口
        LJMP  T0INT
        ORG   001BH   ;T1入口
        LJMP  T1INT
        ORG   0100H
MAIN:   MOV   SP,#60H       ; 设置堆栈指针
        MOV   TMOD,#26H     ; 设置定时器工作方式
        MOV   TH0,#0FFH     ; 设置定时初值
        MOV   TL0, #0FFH
        MOV   TH1,#06H
        MOV   TL1, #06H
        SETB  EA            ; 开放中断
        SETB  ET0
        SETB  ET1
        SETB  TR0           ; 启动定时器
        SETB  TR1
        SJMP  $             ; 等待中断
T0INT:  CLR   TR0
        SETB  TR1
        RETI
T1INT:  CPL P1.0            ;电平取反输出
        RETI
        END
```

```
#include  <reg52.h>
sbit      P10=P1^0;
void main( )
{
   SP=0x60;          //设置堆栈指针
   TMOD=0x26;        //设置定时器工作方式
   TH0=0xFF;         //设置定时初值
   TL0=0xFF;
   TH1=0x06;
   TL1=0x06;
   EA=1;             //开放中断
   ET0=1;
   ET1=1;
   TR0=1;            //启动定时器
   while(1) {;}      //等待中断
}
void  timer0(void)  interrupt  1
{
   TR0=0;
   TR1=1;
}
void  timer1(void)  interrupt  3
{
   P10=!P10;         //电平取反输出
}
```

```
        ORG     0000H
        LJMP    MAIN            ;复位入口转主程序
        ORG     000BH
        LJMP    TOINT           ;转T0中断服务
        ORG     001BH
        LJMP    T1INT           ;转T1中断服务

        ORG     0100H
MAIN:   MOV     SP,#60H
        MOV     TMOD,#26H       ;T0T1初始化
        MOV     TL0,#0FFH       ;T0置初值
        MOV     TH0,#0FFH
        MOV     TL1,#06H        ;T1置初值
        MOV     TH1,#06H
        SETB    EA              ;CPU开放中断
        SETB    ET1             ;允许T1中断
        SETB    ET0             ;允许T0中断
        SETB    TR0             ;启动T0
        ;SETB   TR1             ;不启动T1
        SJMP    $

TOINT:  CLR     TR0             ;T0中断服务
        SETB    TR1
        RETI
T1INT:  CPL     P1.0            ;T1中断服务
        RETI
        END
```

```
#include<reg52.h>
sbit  P10=P1^0;
void  main() {
      SP=0X60;
      TMOD=0x26;
      TL0=0xff;
      TH0=0xff;
      TL1=0x06;
      TH1=0x06;
      EA=1;
      ET0=1;
      ET1=1;
      TR0=1;
      while(1){;
       }
}

void timer0xint(void) interrupt 1 {
      TR0=0;
      TR1=1;
}

void timer1Tint(void) interrupt 3 {
      P10=!P10;
}
```

问题1：此题中的T0可以用方式1计数吗？

问题2：此题中T0的初值是否可以任意？

问题3：定时器T1的初值为什么没有重装？

定时/计数器T0/T1的应用举例

【例6.2】设STC89C52单片机系统时钟频率f_{CLK}为6MH_Z，请编出利用定时/计数器T0在P1.0引脚上产生周期为2秒，占空比为50%的方波信号的程序.

- **确定计数来源得到计数值N**

  T0内部脉冲产生1s定时实现P1.0引脚上电平取反。

  $N=T_{定时}/T_{机器}=1s/\ 2\mu s=1000000/2=500000$

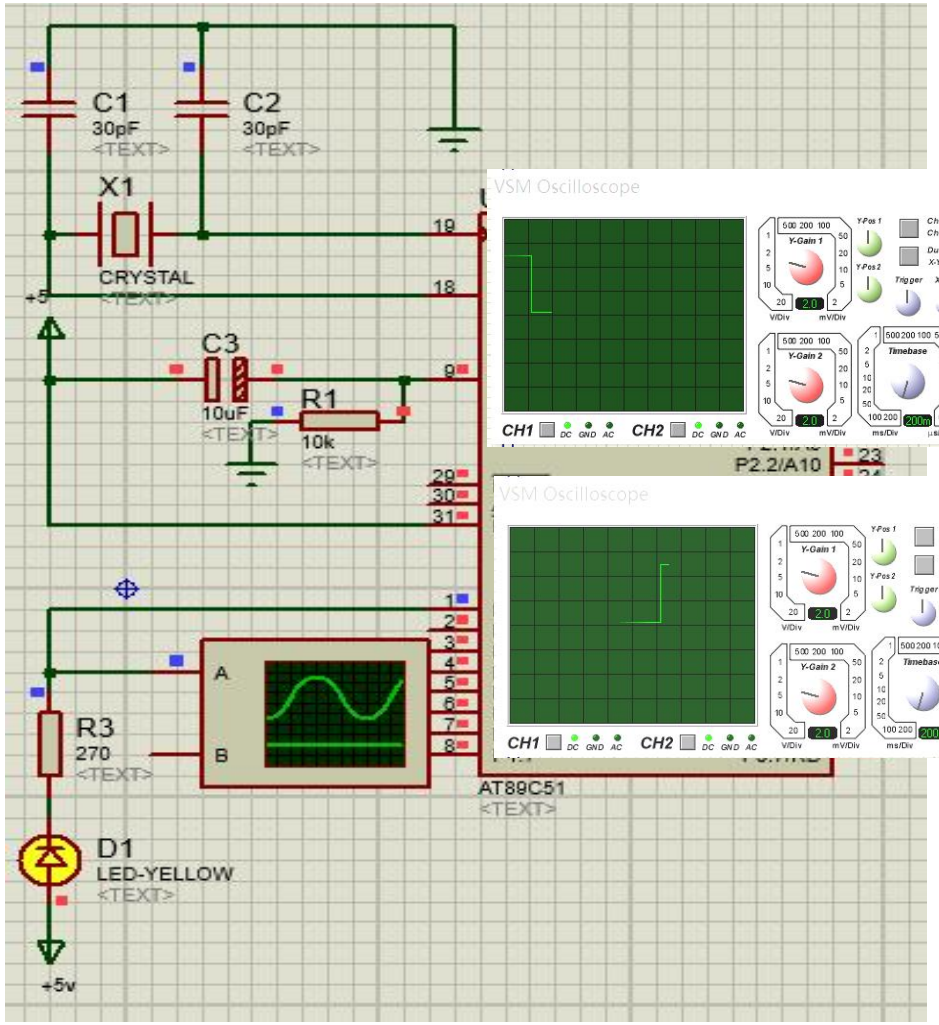
  (F_{CLK}为6MHz，$T_{机器}=12/F_{clk}=12/6MHz=\ 2\mu s$)

- **确定工作方式得最大值M**

  最大值2^{16} = 65536 < 500000

即当Fosc=6MHz时，各工作方式中方式1最长定时=$(2^{16}-0)*2us=65536*2us\approx131ms$，达不到1s定时要求。

因此，选方式1每定时100ms中断一次，判断中断次数到10次否？如果没到，则次数减1返回继续定时。如果到了10次则为1s，则实现电平取反。



定时/计数器T0/T1的应用举例

【例6.2】设STC89C52单片机系统时钟频率f_{CLK}为6MH_Z，请编出利用定时/计数器T0在P1.0引脚上产生周期为2秒，占空比为50%的方波信号的程序.

采用T0方式1定时工作，初始化步骤：

1) 设置TMOD的C/T#和M1M0——即TMOD=0000 0001B=01H
2) 设置THx和TLx初值————————即TH0=3CH,TL0=0B0H;
3) 设置IE的EA和ETx开中断————即EA=1,ET0=1
4) 设置TCON中的TRi启动工作——即TR0=1

- **重新确定T0内部脉冲产生计数值N**

  T0产生100ms

  $N=T_{定时}/T_{机器}$ =100ms/ 2μs

  =100000/2=50000

  (F_{CLK}为6MHz，$T_{机器}=12/F_{clk}=12/6MHz=2\mu s$)

- **确定T0工作于方式1最大值M**

  $M=2^{16}$

- **计算初值X**

  $X=M-N=2^{16}-T_{定时}/T_{机器}$

  $=2^{16}-50000=65536-50000=3CB0H$

  故 TH0=3CH，TL0=0B0H

```
1) MOV  TMOD,#0000 0001B
2) MOV  TH0,#3CH
   MOV  TL0,#0B0H
3) SETB  EA
   SETB  ET0
4) SETB  TR0
```

先仿照例题6-1实现P1.0引脚100ms方波

```
        ORG   0000H
        LJMP  MAIN
        ORG   000BH   ;T0入口
        LJMP  T0INT
        ORG   0100H
MAIN:   MOV   SP,#60H        ; 设置堆栈指针
        MOV   R7,#0AH
        MOV   TMOD,#01H      ; 设置定时器工作方式
        MOV   TH0,#3CH       ; 设置定时初值
        MOV   TL0, #0B0H
        SETB  EA             ; 开放中断
        SETB  ET0
        SETB  TR0            ; 启动定时器
        SJMP  $              ; 等待中断
T0INT:  MOV   TH0,#3CH
        MOV   TL0,#0B0H
        DJNZ  R7,RETN
        CPL   P1.0           ;电平取反输出
        MOV   R7,#0AH
RETN:   RETI
        END
```

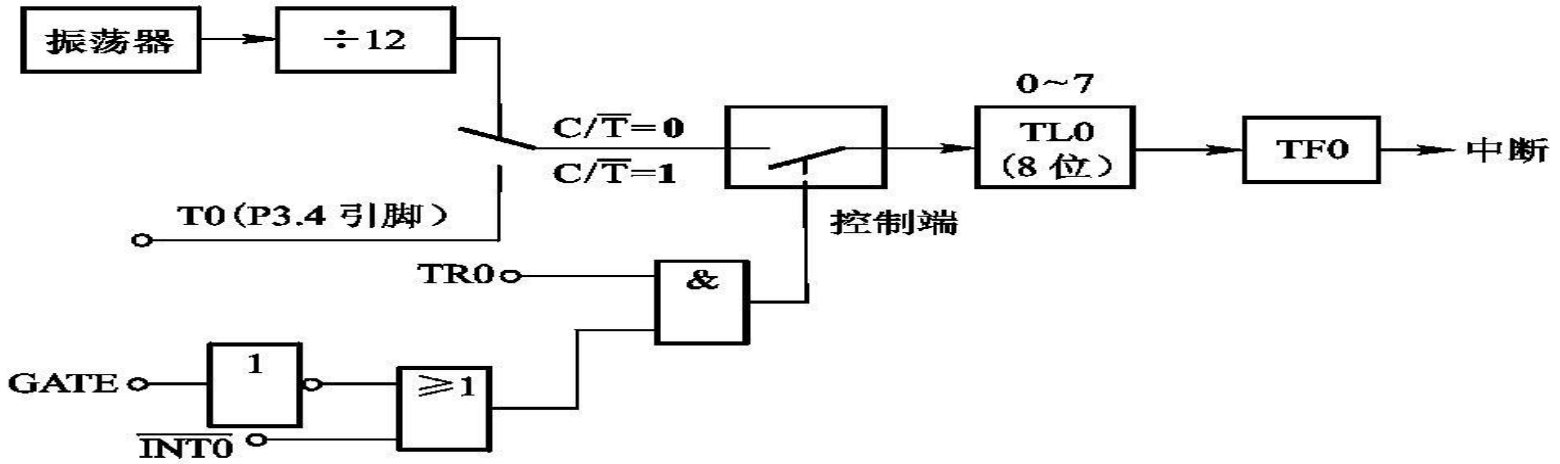
```
#include  <reg52.h>
#define  unsigned char   uchar
uchar    COUNT;
sbit     P10=P1^0;
void main( )
{   SP=0x60;              //设置堆栈指针
    COUNT=0x0A;
    TMOD=0x01;            //设置定时器工作方式
    TH0=0x3C;             //设置定时初值
    TL0=0xB0;
    EA=1;                 //开放中断
    ET0=1;
    TR0=1;                //启动定时器
    while(1) {;}          //等待中断
}
void  timer0(void)  interrupt  1
{     TH0=0x3C;
      TL0=0xB0;
      if(COUNT!=0)   COUNT=COUNT-1;
      else{
              P10=!P10;
              COUNT=0x0a;
       }
}
```

```
        ORG     0000H
        LJMP    MAIN
        ORG     000BH
        LJMP    ITOP

        ORG     0100H
MAIN:MOV    SP,#60H
        MOV     R7,  #0aH
        MOV     TMOD,#01H
        MOV     TH0,#3CH
        MOV     TL0,#0B0H
        SETB    ET0
        SETB    EA
        SETB    TR0
        SJMP    $

ITOP:MOV    TL0,#0B0H
        MOV     TH0,#3CH
        DJNZ    R7,LOOP
        CPL     P1.0
        MOV     R7,#0aH
LOOP:RETI
        END
```

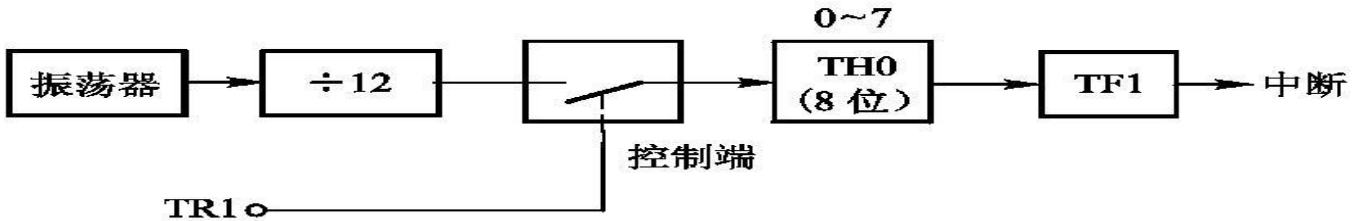
```
#include<reg52.h>
#define uchar unsigned char
uchar COUNT=0;
sbit  P10=P1^0;
void  main(){
  SP=0x60;      /*设置堆栈指针 */
  TMOD=0x1;     /*设置T0定时方式1*/
  TL0=0xb0;     /*装初值*/
  TH0=0x3c;
  IE=0x82;      /*开总中断和定时中断*/
  TR0=1;        /*启动定时器0计数*/
  COUNT=0xa;    /*软件计数初值=10 */
  while(1){ ;  /*踏步等待中断*/
  }
}
/*******定时器0中断函数*********/
void timer0int(void) interrupt 1 {
     TL0=0x0b;              /*重装初值*/
     TH0=0x3c;
     switch(COUNT){         /*判断10次到了吗？*/
       case 0:{             /*1s定时到P1.1求反*/
                 P10= ! P10;
                 COUNT=0xa;
                 break;}
       default:{            /* 没到1s计数值减1*/
                 COUNT=COUNT-1;
                 break;}
     }
}
```

定时/计数器的工作方式3

- **方式3——是为增加一个8位定时/计数器而设**
  - **T1不能工作在方式3，**一般情况下，当T1用作串行口波特率发生器还需要两个定时/计数器时，**T0**才设置为**方式3，T0**将TL0和TH0作为两个独立的**定时/计数器**。
  - **方式3只适用于T0，TL0可作为定时器和计数器，使用T0的原有资源；TH0只能作为定时器且使用T1的资源。**



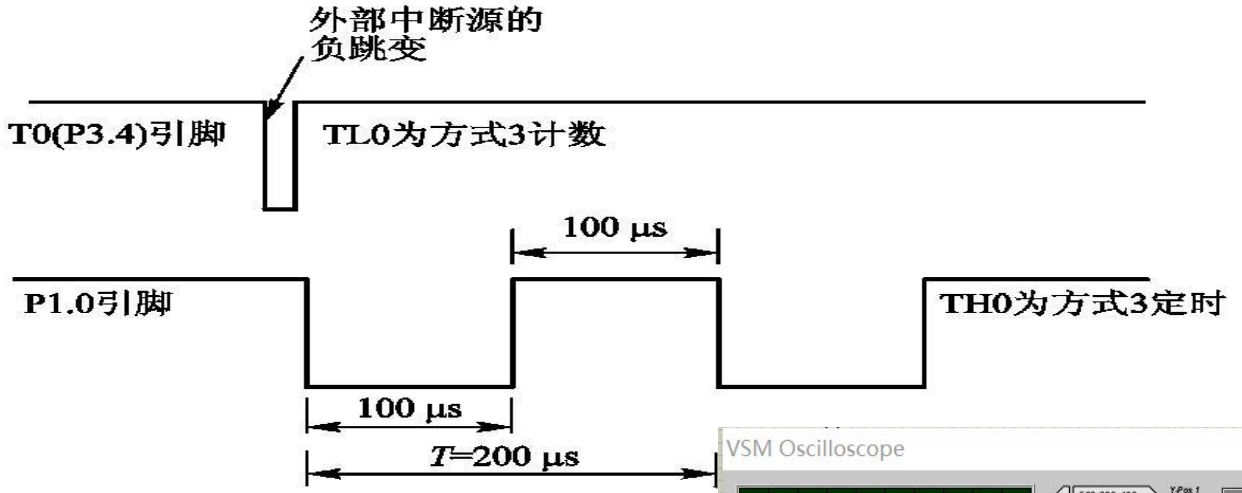
(a) TL0作为8位定时器/计数器

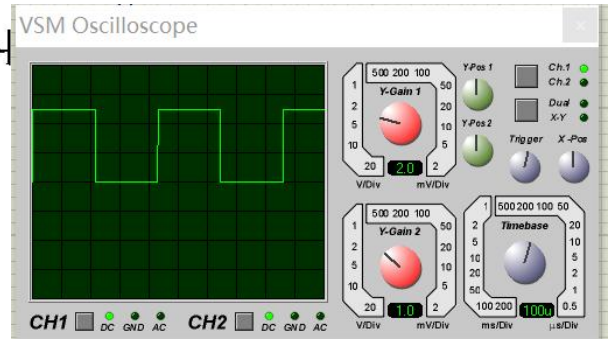


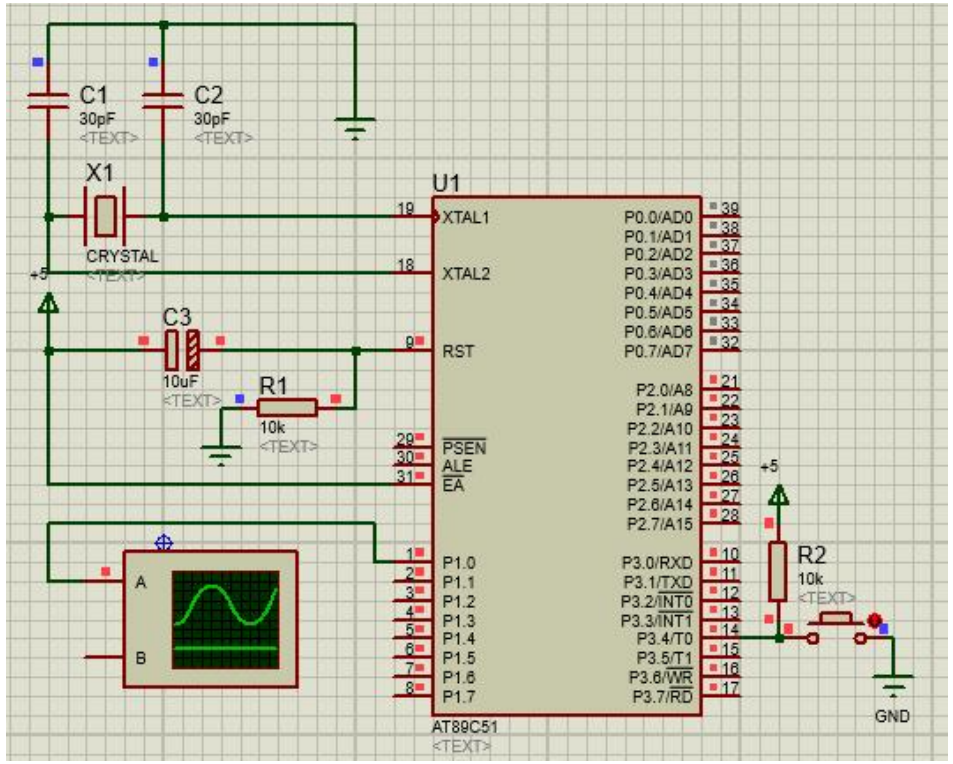
(b) TH0作为8位定时器

方式3的应用举例(可分组讨论)

【例6.4】假设某STC89C52应用系统的2个外部中断源已被占用，因定时器T1工作在方式0作波特率发生器占用。现要求增加1个外部中断源，并控制P1.0引脚输出一个频率5KHz(即周期为200us)的方波， f_{osc}=12MHZ。



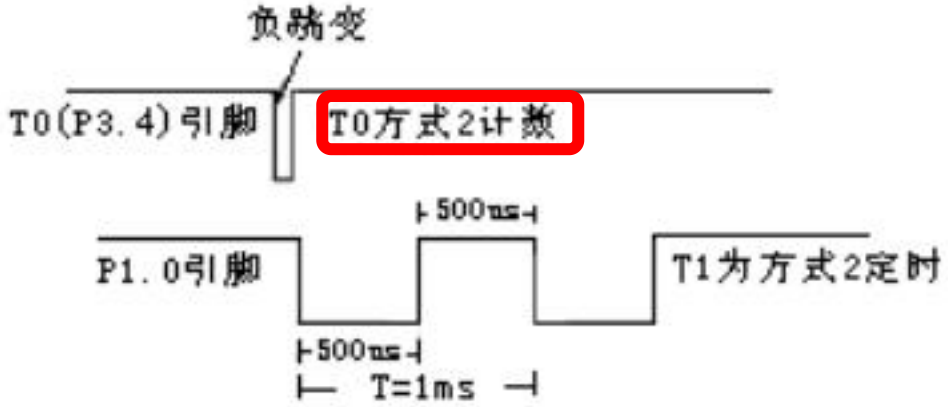




分析：当T1做波特率发生器被占用，要实现外部中断启动周期波形的功能，还需要两个定时计数器，一个用于产生负脉冲，一个用于产生周期波形。因此考虑到使用方式3，将T0分成两个使用。

定时/计数器工作方式2&方式3的应用比较

【例6-3】将T0（P3.4）引脚上发生负跳变信号作为P1.0引脚产生方波的启动信号。要求P1.0脚上输出周期为1ms的方波，如图6-5所示（系统时钟6MHz）。



T0方式2计数，计算X=M-N=2^8-N

X=2^8-1=256-1 = 255=0FFH

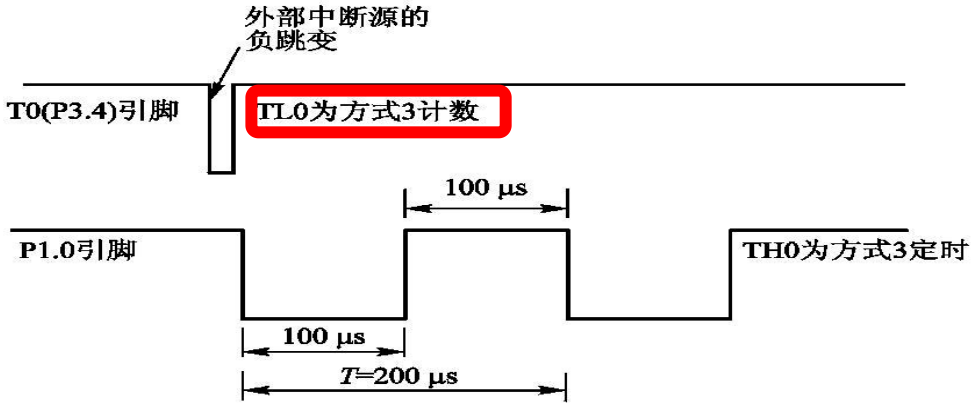
故 TH0=0FFH，TL0=0FFH

T1方式2定时，计算X=M-N=2^8-$T_{定时}/T_{机器}$

X=2^8-500μs/2μs=256-250 = 06H

故 TH1=06H，TL1=06H

【例6.4】fosc=12MHZ，2个外部中断源被占用,定时器T1做波特率发生器被占用。现将TL0外部脉冲作为外部中断触发P1.0引脚输出周期为200us的方波.



TL0方式3计数，计算X=M-N=2^8-N

X=2^8-1=256-1 = 255=0FFH

故 TL0=0FFH

TH0方式3定时，计算X=M-N=2^8-$T_{定时}/T_{机器}$

X=2^8-100μs/1μs=256-100 =156H

故 TH1=9CH

例6-4：方式3的应用

```
        ORG     0000H
        LJMP    MAIN
        ORG     000BH    ;TL0方式3中断入口用T0的资源
        LJMP    TL0INT
        ORG     001BH    ;TH0方式3中断入口用T1的资源
        LJMP  TH0INT
        ORG    0100H     ;主程序入口
MAIN:   MOV   SP,#60H
        MOV   TMOD，#07H  ;T0方式3，T1方式0定时
        MOV   TL0，#0FFH  ;置TL0初值
        MOV   TH0，#9CH   ;置TH0初值
        SETB   EA         ;开中断
        SETB   ET0        ;开T0中断
        SETB   ET1        ;开T1中断
        SETB  TR0         ;启动T0计数
HERE:   AJMP  HERE
TL0INT: MOV   TL0,#0FFH;重装TL0初值
        SETB  TR1       ;启动TH0定时
        RETI

TH0INT: MOV   TH0,#9CH ;重装TH0初值
        CPL   P1.0      ;P1.0输出求反
        RETI
        END
```

例6-3：方式2的应用

```
        ORG   0000H
        LJMP  MAIN
        ORG   000BH          ;T0的中断入口
        LJMP  T0XINT
        ORG   001BH          ;T1的中断入口
        LJMP  T1TIME
        ORG   0030H          ;主程序入口
MAIN: MOV   SP,#60H
        MOV   TMOD,#26H ;T0方式2计数T1方式2定时
        MOV   TL0,#0FFH    ;T0置初值
        MOV   TH0,#0FFH
        MOV   TL1,#06H     ;T1置初值
        MOV   TH1,#06H
        SETB   EA    ;开总中断
        SETB   ET0   ;开T0中断
        SETB   ET1   ;开T1中断
        SETB  TR0    ;启动T0计数
HERE:AJMP  HERE
T0XINT:  CLR       TR0    ;停止T0计数
         SETB      TR1    ;启动T1定时
         RETI
T1TIME: CPL        P1.0   ;P1.0取反
         RETI
         END
```

例6-4：方式3的应用

```
#include<reg52.h>
sbit  P10=P1^0;
void  main(){
  SP=0X60;
  TMOD=0x07;
  TL0=0xff;
  TH0=0x9c;
  EA=1;
  ET0=1;
  ET1=1;
  TR0=1;
  while(1){
  }
}
void timer0int(void) interrupt 1{
    TL0=0xff;
    TR1=1;
}
void timer1int(void) interrupt 3{
    TH0=0x9c;
    P10=!P10;
}
```

例6-3：方式2的应用

```
#include<reg52.h>
sbit  P10=P1^0;
void  main() {
    SP=0X60;
    TMOD=0x26;
    TL0=0xff;
    TH0=0xff;
    TL1=0x06;
    TH1=0x06;
    EA=1;
    ET0=1;
    ET1=1;
    TR0=1;
    while(1){;
     }
}

void timer0xint(void) interrupt 1 {
    TR0=0;
    TR1=1;
}

void timer1Tint(void) interrupt 3 {
    P10=!P10;
}
```

第6讲　定时/计数器要点

重点

- 定时/计数器相关的SFR各位功能和作用
- 定时/计数器的4种工作方式的特点

- 定时与计数的区别
- 计数原理X初值= M最大值-N计数次数
- 最大计数值和最长定时时间
- 定时计数器的初始化
- 工作方式1和工作方式2的应用