

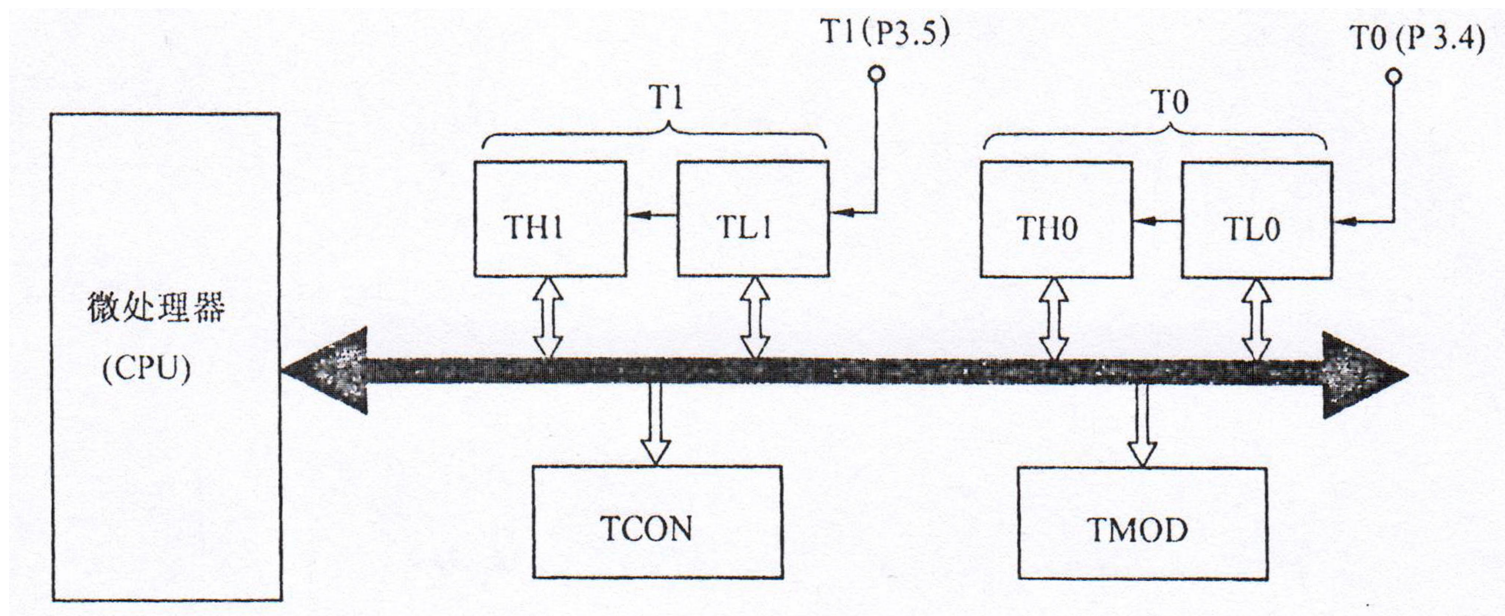
### 定时器/计数器概述

MCS-51的8031系列单片机的内部集成有两个16位的可编程定时器/计数器，即定时器/计数器T0和定时器/计数器T1。T0和T1既可以用于定时，也可以用于对外部事件进行计数，还可以作为串行接口的波特率发生器。

8032系列的单片机还增加了一个定时器/计数器T2，功能比T0和T1更为强大，除了定时/计数功能外，还具有16位重装载模式，加/减计数模式和波特率发生器。

### 定时器/计数器T0与T1的结构

定时器/计数器T0由特殊功能寄存器TH0、TL0构成，定时器/计数器T1由特殊功能寄存器TH1、TL1构成，T0和T1有4种工作方式（方式0、方式1、方式2和方式3、），由特殊功能寄存器TMOD（工作方式控制寄存器）和TCON（定时器/计数器控制寄存器）设定。



### 定时器/计数器T0与T1的结构

- 当用于计数器方式时，计数器对外部事件计数，计数脉冲来自外部输入引脚T0（P3.4）和T1（P3.5）。当外部输入引脚发生1到0的负跳变时，计数器加1。
- CPU在每个机器周期的S5P2拍节对外部计数脉冲进行采样。如果前一个机器周期采样为高电平，后一个机器周期采样为低电平即为一个有效的计数脉冲，在下一个机器周期的S3P1进行计数。
- 由于计数脉冲是在两个机器周期进行的，所以最高计数频率是振荡频率的1/24。

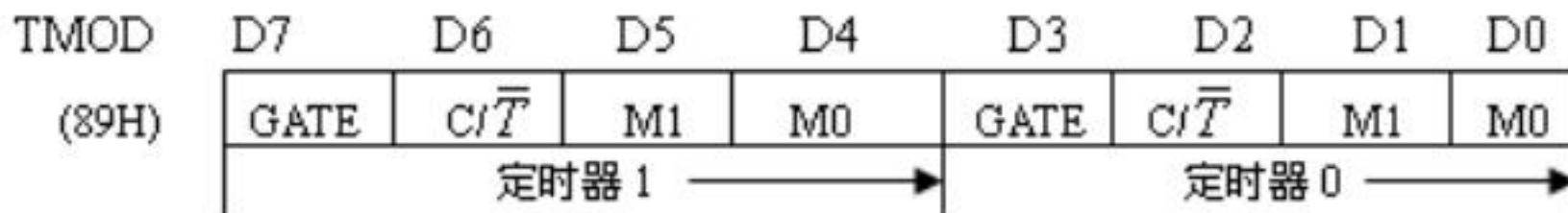
### 定时器/计数器T0与T1的结构

- 当用于定时器方式时，定时器的输入来自内部时钟发生器电路，每个机器周期计数器加1，而1个机器周期包含有12个振荡周期，所以，**定时器的计数频率为晶振频率的1/12**。
- 如果单片机的时钟频率为12MHz，则计数频率为1MHz，即每个微秒（ $\mu s$ ）计数器加1。

### 定时器/计数器T0与T1的结构

#### 工作方式控制寄存器TMOD

- TMOD寄存器用于设定两个定时器/计数器T0和T1的工作方式。  
TMOD的字节地址为89H，不能位寻址，只能用字节指令设置其内容，低4位和高4位分别用于定时器T0和T1的工作方式控制。
- TMOD寄存器的内容如下：



## 定时器/计数器T0与T1的结构

### 工作方式控制寄存器TMOD

TMOD	D7	D6	D5	D4	D3	D2	D1	D0
(89H)	GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0
	定时器 1				定时器 0			

◆ **GATE**: 门控位。

**GATE=0**时, 由运行控制位**TRX** ( $X=0$ 或 $1$ ) 启动定时器。

**GATE=1**时, 由外中断请求信号 ( $\overline{INT0}$ 或 $\overline{INT1}$ ) 和**TRX**共同启动定时器。

◆ **C/  $\overline{T}$** : 定时/计数模式选择位。

◆ **C/  $\overline{T}$ =0** 定时工作模式,

◆ **C/  $\overline{T}$ =1** 计数工作模式。计数脉冲来自外部输入引脚**T0** (**P3.4**) 和**T1** (**P3.5**) , 负跳变有效。

## 第6部分 MCS-51的定时器/计数器

### 定时器/计数器T0与T1的结构

#### 工作方式控制寄存器TMOD

TMOD	D7	D6	D5	D4	D3	D2	D1	D0
(89H)	GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0
	定时器 1				定时器 0			

#### ◆ M1M0: 工作方式选择位

M1M0=00 方式0 (13位定时/计数器)

M1M0=01 方式1 (16位定时/计数器)

M1M0=10 方式2 (8位自动重装载定时/计数器)

M1M0=11 方式3 (仅适用于T0, T0分为两个独立8位定时/计数器; T1停止计数)



### 定时器/计数器T0与T1的结构

#### 定时器/计数器控制寄存器TCON

TCON寄存器用于保存定时器/计数器、外部中断请求的中断请求标志。TCON寄存器字节地址是88H，位地址88H~8FH。与定时器T0和T1有关的控制位有4位，与中断有关的控制位有4位。寄存器的内容和位地址表示如下：

TCON	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
(88H)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0



### 定时器/计数器T0与T1的结构

#### 定时器/计数器控制寄存器TCON

TCON	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
(88H)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

◆ TF0和TF1：溢出标志位，也是中断请求标志位。

◆ TR0、TR1：定时器/计数器运行控制位。

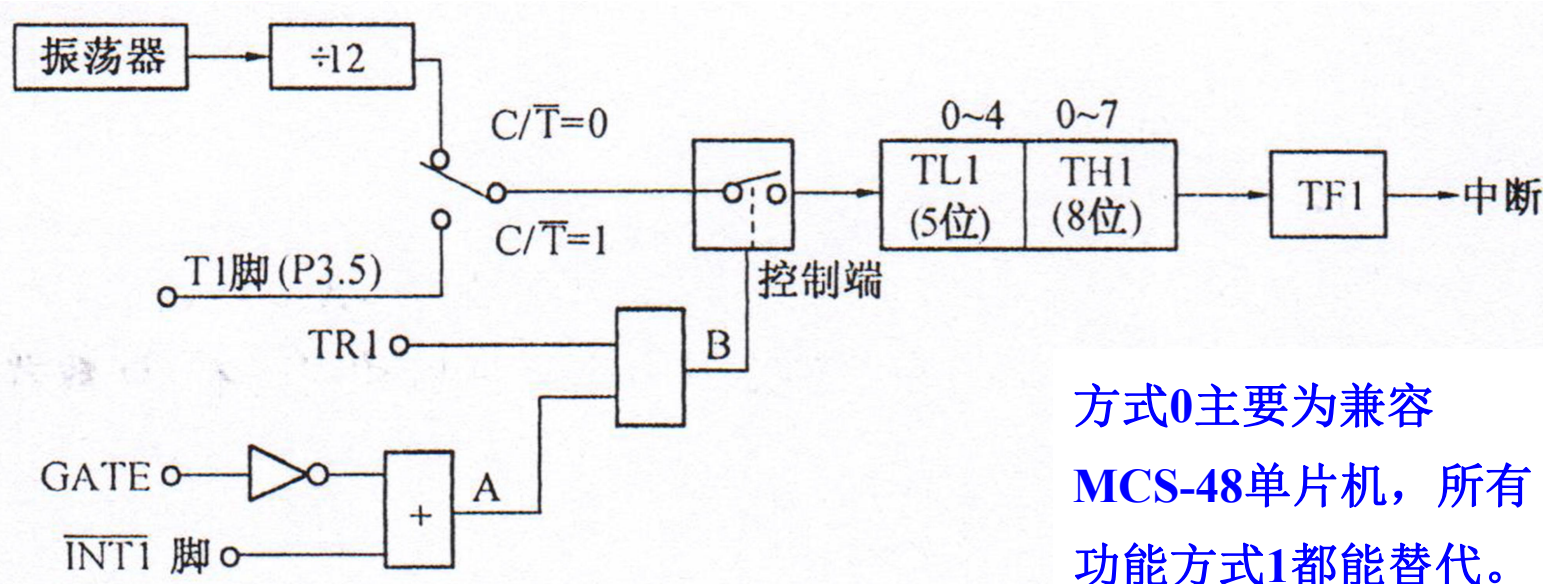
TR0 (TR1) =1      启动定时器/计数器工作

TR0 (TR1) =0      停止定时器/计数器工作

### 定时器/计数器T0与T1的工作方式

#### 定时器/计数器T0与T1的工作方式0

- ◆ 当M1、M0为00时，定时器/计数器被设置为工作方式0。
- ◆ 定时器/计数器被设置为工作方式0时，为13位计数器，由TLx (x=0, 1) 的低5位和THx的高8位所构成。TLx低5位溢出，则向THx进位，THx计数溢出，则置位溢出标志TFx。

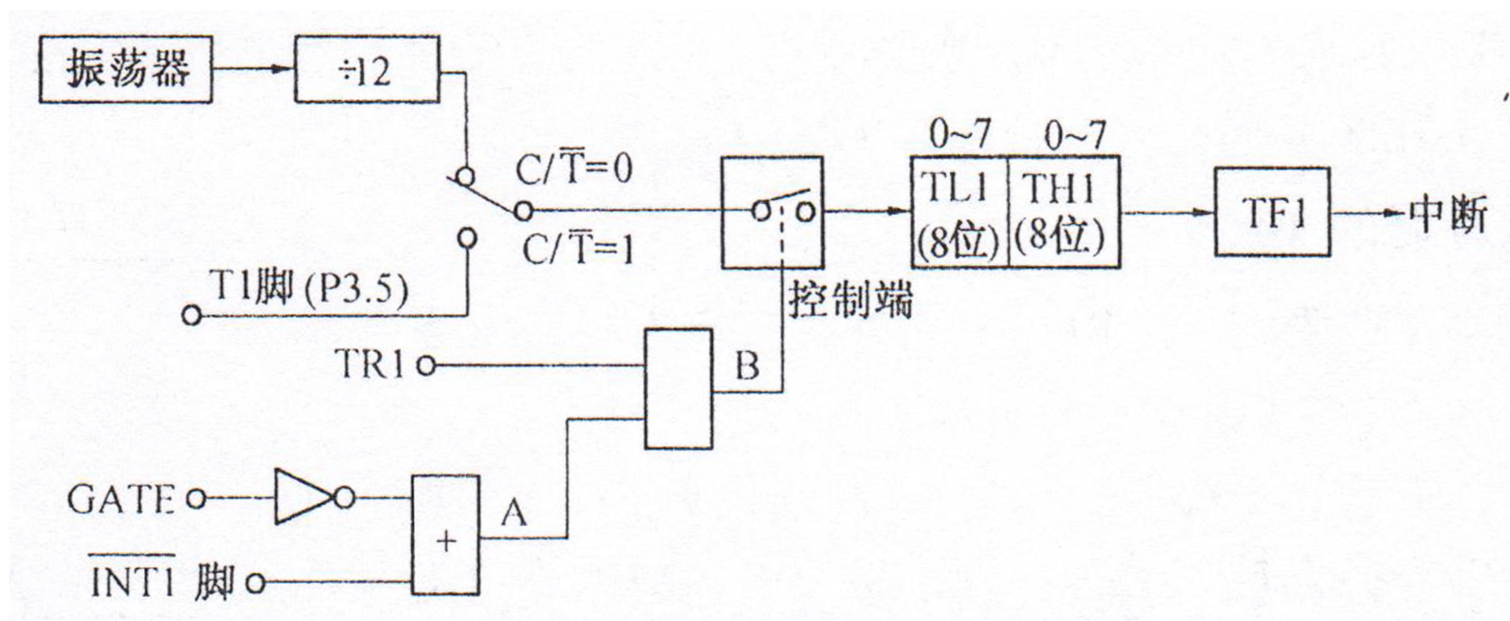


方式0主要为兼容  
MCS-48单片机，所有  
功能方式1都能替代。

### 定时器/计数器T0与T1的工作方式

#### 定时器/计数器T0与T1工作方式1

- ◆ 当M1、M0为01时，定时器/计数器被设置为工作方式1。
- ◆ 定时器/计数器被设置为工作方式1时，为16位计数器，由TLx (x=0, 1) 的低8位和THx的高8位所构成。TLx低8位溢出，则向THx进位，THx计数溢出，则置位溢出标志TFx。



### 定时器/计数器T0与T1的工作方式

#### 定时器/计数器T0与T1工作方式2

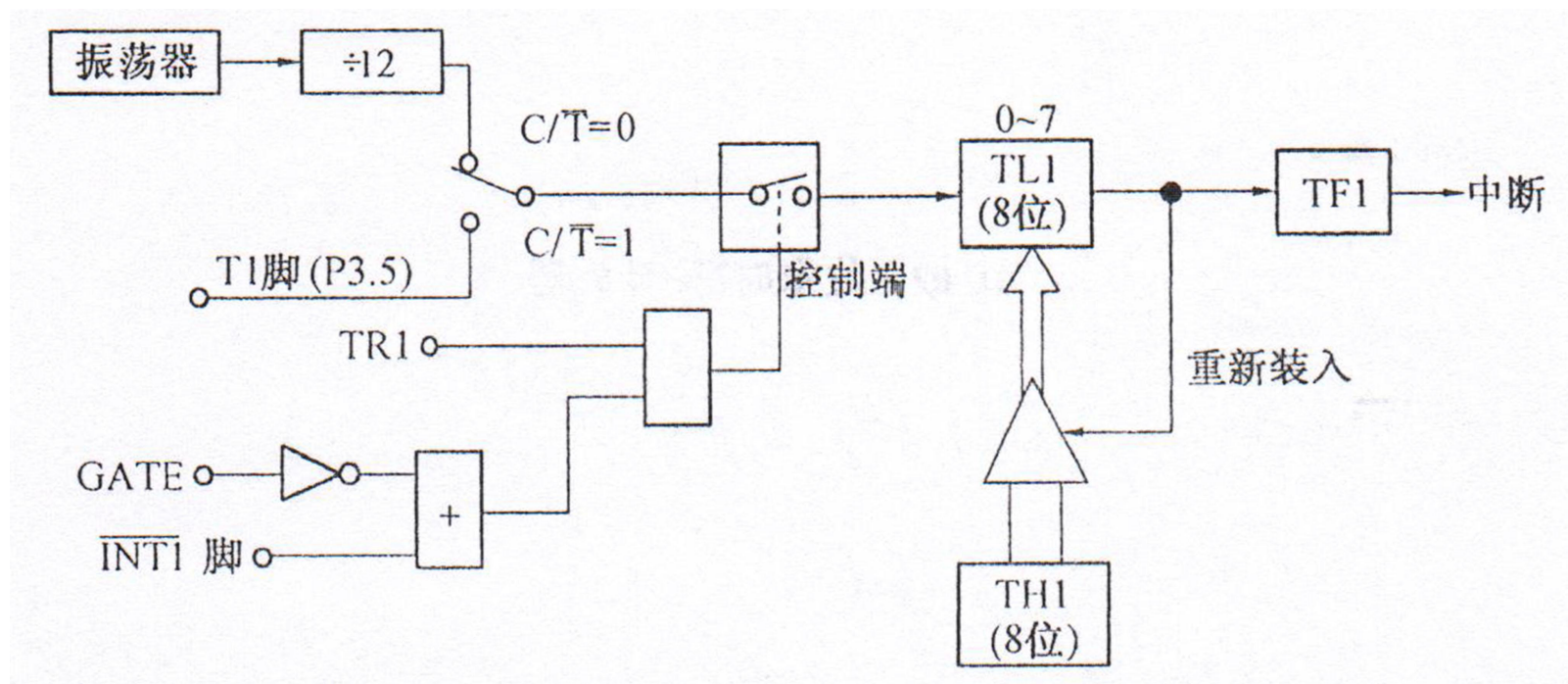
- ◆ 当M1、M0为10时，定时器/计数器被设置为工作方式2。
- ◆ 方式2是**能自动重装计数初值的8位计数器**。TLx作8位计数器用，THx用以保存计数初值。当TLx计数溢出时，将溢出位TFx置1，同时将保存在THx中的计数初值重新自动装入TLx中，继续计数，循环重复不止。
- ◆ 方式2可自动循环计数，省去了软件重新装入初值的麻烦，提高了定时的精度。该模式常用在定时精度高的场合，例如做为串行口的波特率发生器使用。



## 第6部分 MCS-51的定时器/计数器

### 定时器/计数器T0与T1的工作方式

#### 定时器/计数器T0与T1工作方式2



### 定时器/计数器T0与T1的工作方式

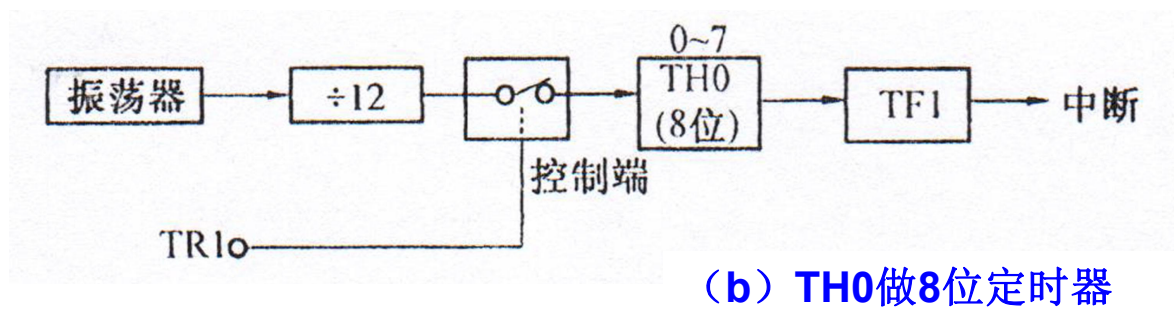
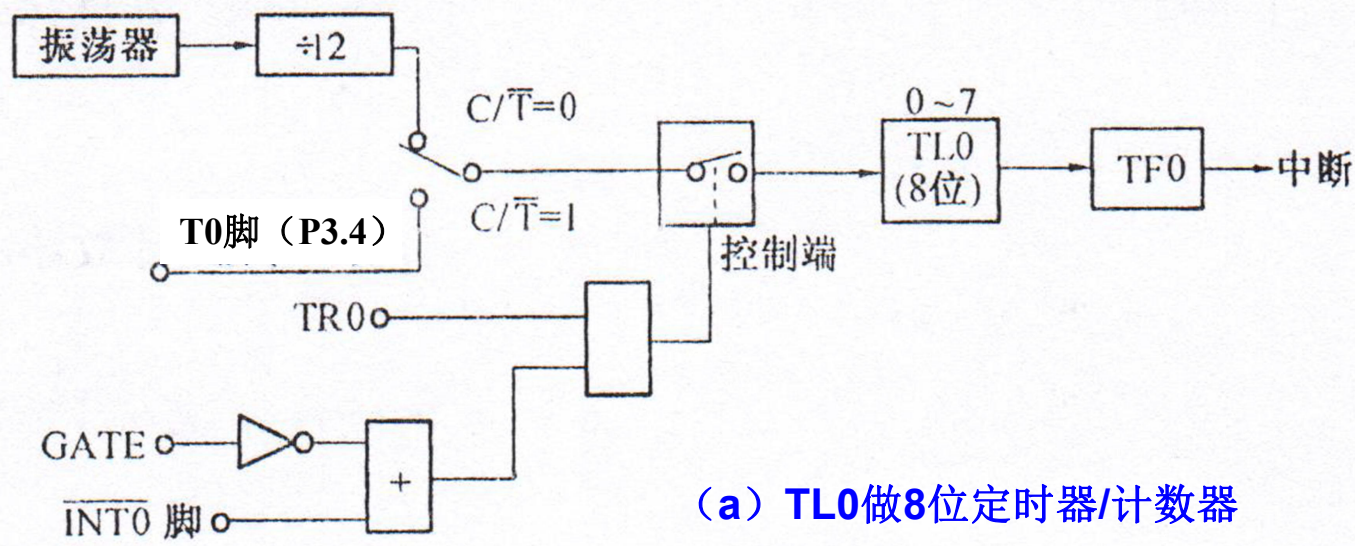
#### 定时器/计数器T0与T1工作方式3

- ◆ 工作方式3只适用于定时器/计数器T0，定时器/计数器T1不能工作在方式3。当TMOD的低2位为11时，定时器/计数器T0被设置为工作方式3。
- ◆ 方式3下，TH0和TL0变成2个独立的计数器。TL0占用了全部的T0控制位（C/T，GATE，TR0，TF0），仍可具有定时/计数功能；TH0只能用于定时方式，运行控制位和溢出标志位则借用定时器T1的TR1和TF1。
- ◆ 在T0设置为方式3工作时，一般是将定时器T1作为串行口波特率发生器，或用于不需要中断的场合。

## 第6部分 MCS-51的定时器/计数器

### 定时器/计数器T0与T1的工作方式

定时器/计数器T0  
与T1的工作方式3

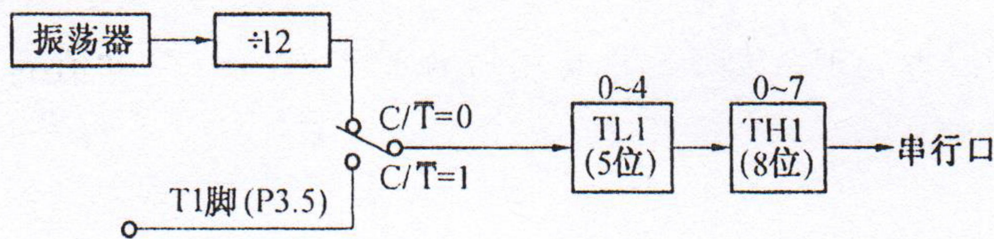




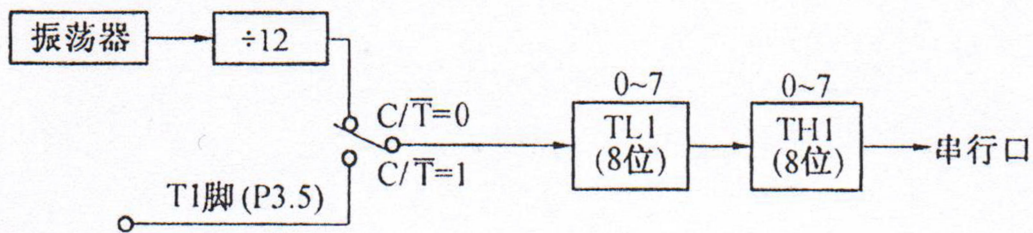
## 第6部分 MCS-51的定时器/计数器

### 定时器/计数器T0与T1的工作方式

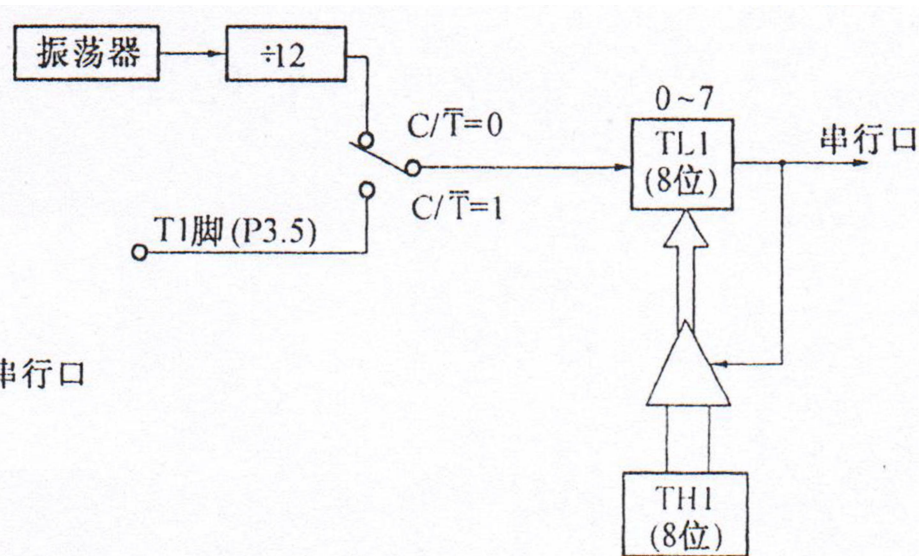
定时器/计数器T0  
与T1的工作方式3



T0为方式3时T1为方式0的工作示意图



T0为方式3时T1为方式1的工作示意图



T0为方式3时T1为方式2的工作示意图

### 定时器/计数器对输入信号的要求

- 定时器/计数器采用定时器模式时，输入信号是内部时钟，每个机器周期产生一个计数脉冲。
- 定时器/计数器采用计数器模式时，输入信号是外部信号，负跳变有效。每个机器周期检测一次外部信号的状态，因此外部计数脉冲最高频率为机器周期对应频率的 $1/2$ ，即振荡频率的 $1/24$ 。
- 定时器/计数器采用计数器模式时，外部输入信号的高、低电平至少保持一个机器周期。

### 定时器/计数器T0与T1的初始化

#### 1. 初始化的步骤:

- 确定工作方式、操作模式、启动控制方式，写入TMOD、TCON。
- 设置定时或计数器的初值，直接将初值写入TH0、TL0或TH1、TL1、TH2、TL2中。
- 根据需要开放CPU和定时 / 计数器的中断，即对IE和IP寄存器编程
- 启动定时器 / 计数器工作：若要求用软件启动，编程时对TCON中的TR0或TR1置位即可启动；若由外部中断引脚电平启动，则对TCON中的TR0或TR1置位后，还需给外引脚（或）加启动电平。

### 定时器/计数器T0与T1的初始化

#### 2. 计数初值的计算

若设最大计数值为 $2^n$ ， $n$ 为计数器位数，各工作方式下的 $2^n$ 值为：

方式0:  $2^n=8192$  ;  $n=13$

方式1:  $2^n=65536$  ;  $n=16$

方式2:  $2^n=256$  ;  $n=8$

方式3:  $2^n=256$  ;  $n=8$ ，定时器T0分成2个独立的8位计数器，所以TH0、TL0的最大计数值均为256。

T0、T1定时器均为加1计数器，当加到最大值（00H或0000H）时产生溢出中断，因此计数器初值X的计算式为： $X=2^n-\text{计数值}$

### 定时器/计数器T0与T1的初始化

#### ◆ 计数模式时

计数模式时，对外部脉冲进行计数，其计数初值： $X=2^n$ -计数值。

#### ◆ 定时模式时

定时模式时，对机器周期进行计数，故计数脉冲频率为 $f_{cont}=f_{osc} / 12$ 、计数周期  $T=1 / f_{cont}$ ，定时模式的计数初值X等于：

12、计数周期  $T=1 / f_{cont}$ ，定时模式的计数初值X等于：

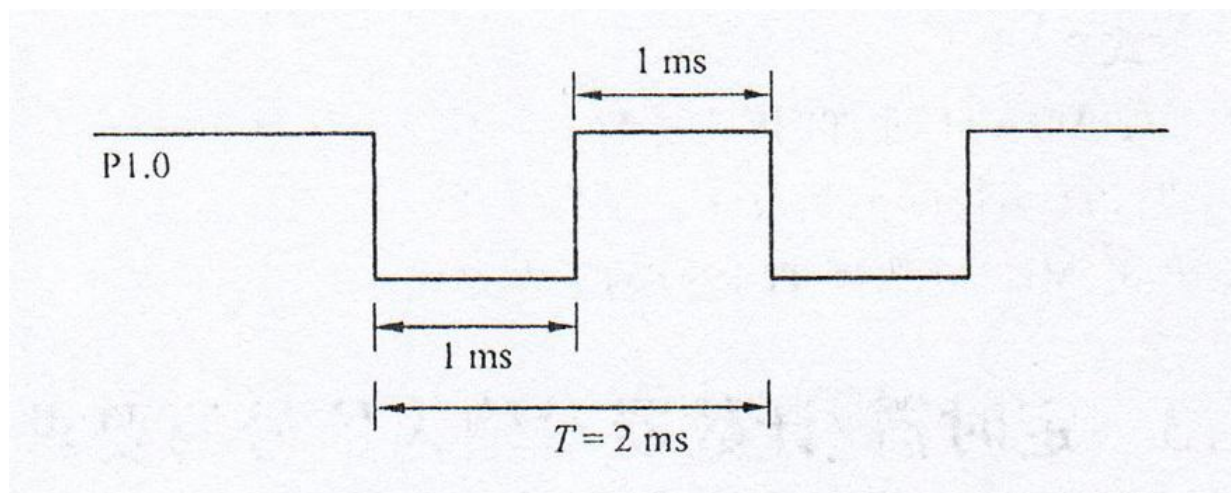
$$X=2^n\text{-计数值}=2^n\text{-}t / T=2^n\text{-} (t \times f_{osc}) / 12$$

注： $f_{osc}$ 单位是MHz，定时时间t的单位是 $\mu s$ 。

### 定时器/计数器T0与T1的编程和应用

#### 例（方式1的应用）

假设系统时钟频率采用6MHz，要在P1.0上输出一个周期为2ms的方波，如图所示。



采用定时器T0，每1ms产生一次中断，中断服务程序程序中对P1.0取反。

### 定时器/计数器T0与T1的编程和应用

例（方式1的应用）

- 计算初值

$$\text{机器周期} = \frac{12}{6 \times 10^6} = 2 \times 10^{-6}(\text{s}) = 2(\mu\text{s})$$

T0的初值X计算：

$$X = 2^n - \text{计数值} = 2^n - t / T = 2^n - (t \times f_{\text{osc}}) / 12$$

$$n=16, \text{计数值} = 1\text{ms} / 2\mu\text{s} = 500$$

$$X = 2^{16} - \text{计数值} = 2^{16} - 500 = 65536 - 500 = 65036 = 0\text{FE}0\text{CH}$$

所以，T0的初值为：TH0=0FEH，TL0=0CH

可以使用字节分离运算符LOW和HIGH获得16位数的低位字节和高位字节，如LOW（65036）=0CH，HIGH（65036）=0FEH。



### 定时器/计数器T0与T1的编程和应用

#### 例（方式1的应用）

- 程序设计  
(中断方式)

```
ORG    0000H
RESET: AJMP  MAIN           ; 转主程序
      ORG    000BH         ; T0中断入口
      AJMP  IT0P           ; 转T0中断处理程序

      ORG    0100H
MAIN:  MOV   SP, #60H       ; 设置堆栈指针
      MOV   TMOD, #01H     ; 设置T0为方式1
      MOV   TL0, #0CH      ; T0置初值
      MOV   TH0, #0FEH
      SETB  EA             ; CPU开中断
      SETB  ET0            ; 允许T0中断
      SETB  TR0            ; 启动T0
      HERE: AJMP  HERE

IT0P:  MOV   TL0, #0CH     ; T0中断服务程序
      MOV   TH0, #0FEH     ; T0重新置初值
      CPL   P1.0           ; P1.0输出方波
      RETI
```

## 第6部分 MCS-51的定时器/计数器

### 定时器/计数器T0与T1的编程和应用

#### 例（方式1的应用）

- 程序设计  
(查询方式)

```

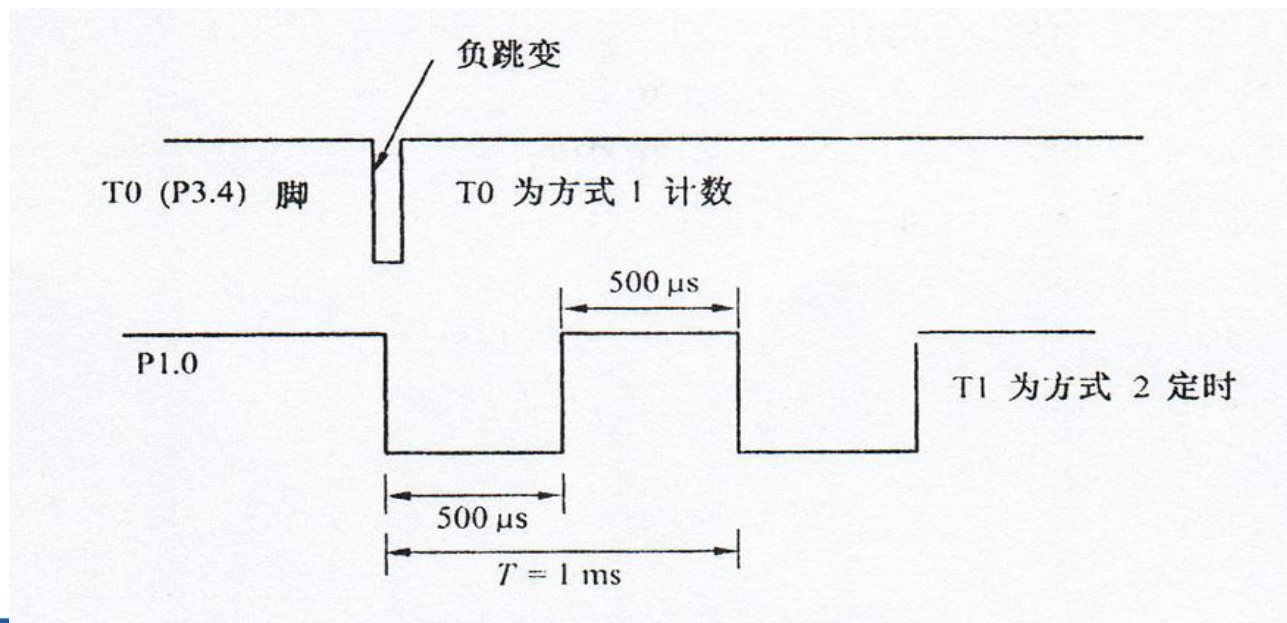
                ORG     0000H
RESET:  AJMP     MAIN                ; 转主程序

                ORG     0100H
MAIN:    MOV     TMOD, #01H          ; 设置T0为方式1
          MOV     TL0, #0CH          ; T0置初值
          MOV     TH0, #0FEH
          SETB    TR0                ; 启动T0
LOOP:    JNB     TF0, LOOP           ; 查询TF0标志
          MOV     TL0, #0CH          ; T0重置初值
          MOV     TH0, #0FEH
          CLR     TF0                ; T0溢出，清溢出标志
          CPL     P1.0               ; P1.0输出方波
          SJMP    LOOP
```

### 定时器/计数器T0与T1的编程和应用

**例（方式2的应用）：**当T0（P3.4）引脚上发生负跳变时，从P1.0引脚上输出一个周期为1ms的方波，如图所示（设系统时钟为6MHz）。

- T0设置为方式1的计数器模式，T1设置为方式2的定时器模式。
- T0初值设为0FFFFH，外部计数输入端（P3.4）发生一次负跳变即引发T0溢出中断。
- 一旦T0产生中断，在T0中断服务程序中启动T1，持续产生周期为1ms的方波。



### 定时器/计数器T0与T1的编程和应用

例（方式2的应用）

- 计算初值

$$\text{机器周期} = \frac{12}{6 \times 10^6} = 2 \times 10^{-6}(\text{s}) = 2(\mu\text{s})$$

T0的初值=0FFFFH

T1的初值X计算：

$$X = 2^n - \text{计数值} \quad n=8, \text{计数值} = 500\mu\text{s} / 2\mu\text{s} = 250$$

$$X = 2^8 - \text{计数值} = 2^8 - 250 = 256 - 250 = 6 = 06\text{H}$$

所以，T1的初值为：TH1=06H，TL1=06H

## 定时器/计数器T0与T1的编程和应用

### 例（方式2的应用）

- 程序设计

```
ORG 0000H
RESET: LJMP MAIN          ; 转主程序
      ORG 000BH          ; T0中断入口
      AJMP IT0P          ; 转T0中断处理程序
      ORG 001BH          ; T1中断入口
      AJMP IT1P          ; 转T1中断处理程序

      ORG 0100H
MAIN:  MOV SP, #60H        ; 设置堆栈指针
      MOV TMOD, #25H      ; T1为方式2, T0为方式1计数器
      MOV TL0, #0FFH      ; T0置初值
      MOV TH0, #0FFH
      MOV TL1, #06H       ; T1置初值
      MOV TH1, #06H
      SETB EA             ; CPU开中断
      SETB ET0            ; 允许T0中断
      SETB TR0            ; 启动T0计数
      HERE: AJMP HERE
```

### 定时器/计数器T0与T1的编程和应用

例（方式2的应用）

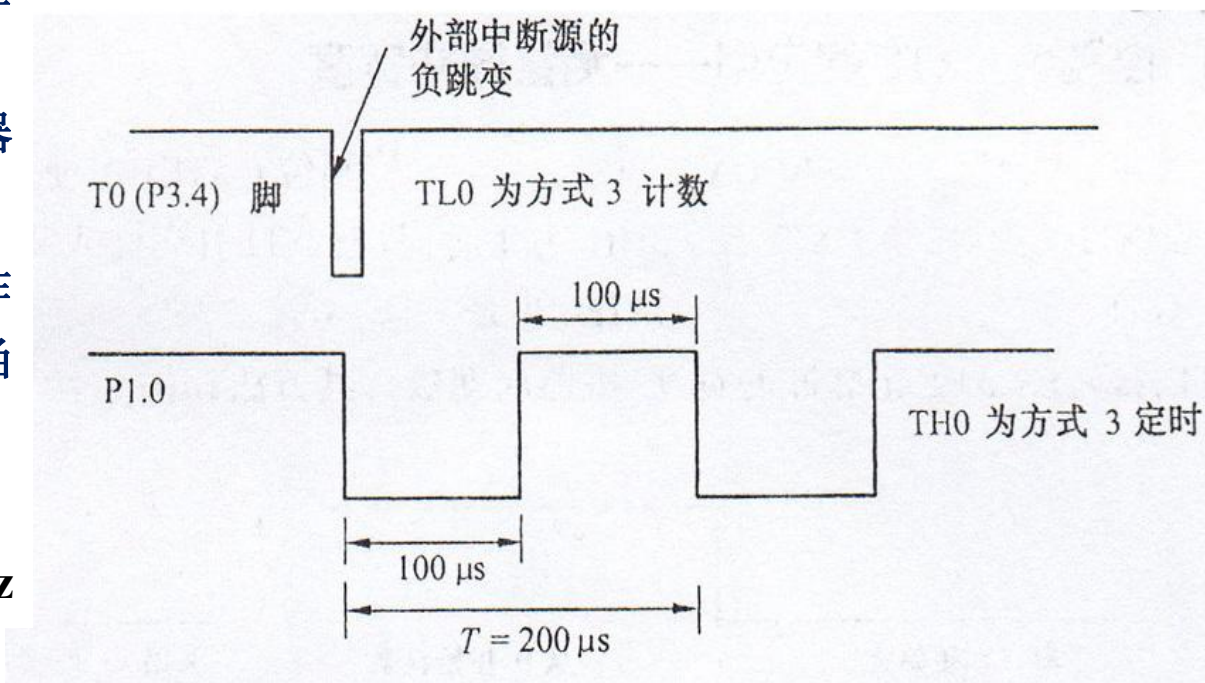
- 程序设计

IT0P:	CLR	TR0	; 停止T0计数
	SETB	ET1	; 允许T1中断
	SETB	TR1	; 启动T1
	CLR	P1.0	; P1.0输出低电平
	RETI		
IT1P:	CPL	P1.0	; T1中断服务程序，输出方波
	RETI		

### 定时器/计数器T0与T1的编程和应用

**例（方式3的应用）：** 假设某MCS-51应用系统的两个外部中断源已被占用，设置定时器T1工作在方式2，作波特率发生器使用。现要求增加一个外部中断源，并控制P1.0引脚输出一个5kHz的方波。假设系统时钟为12MHz。

- T0设置为方式3，TL0工作在方式3的计数器模式，TH0工作在方式3的定时器模式。
- T0外部计数引脚（P3.4）作为新增的外部中断输入，当检测到该引脚上负跳变时，TL0溢出产生中断。
- TH0定时控制P1.0输出5kHz的方波。





### 定时器/计数器T0与T1的编程和应用

例（方式3的应用）

- 计算初值

$$\text{机器周期} = \frac{12}{12 \times 10^6} = 1 \times 10^{-6} (\text{s}) = 1(\mu\text{s})$$

TL0的初值=0FFH

TH0的初值X计算：

$$X = 2^n - \text{计数值} \quad n=8,$$

5khz的方波周期是200 $\mu\text{s}$ ，TH0的定时时间为半周期100 $\mu\text{s}$ 。

TH0的计数值=100 $\mu\text{s}$ /1 $\mu\text{s}$ =100。

$$X = 2^8 - \text{计数值} = 2^8 - 100 = 256 - 100 = 156 = 9\text{CH}$$

所以，TH0的初值为：TH0=9CH

## 第6部分 MCS-51的定时器/计数器

### 定时器/计数器T0与T1的编程和应用

例（方式3的应用）

#### ● 程序设计

```
ORG    0000H
RESET: LJMP    MAIN                ; 转主程序
ORG    000BH                      ; T0中断入口
AJMP    TL0INT                    ; 转TL0中断处理程序
ORG    001BH                      ; T1中断入口，被TH0占用
AJMP    TH0INT                    ; 转TH0中断处理程序

ORG    0100H
MAIN:   MOV     SP, #60H           ; 设置堆栈指针
        MOV     TMOD, #27H        ; T0为方式3计数，T1为方式2定时
        MOV     TL0, #0FFH        ; T0置初值
        MOV     TH0, #9CH
        MOV     TL1, #data        ; T1置波特率常数初值
        MOV     TH1, #data
        MOV     IE, #97H          ; 设置中断允许（关闭TH0中断）
        MOV     TCON, #15H        ; 启动TL0工作（T1自动启动）
        ... ..
LOOP:   调用其他主程序功能
        SJMP    LOOP
```

### 定时器/计数器T0与T1的编程和应用

#### 例6.5（方式3的应用）

- 程序设计

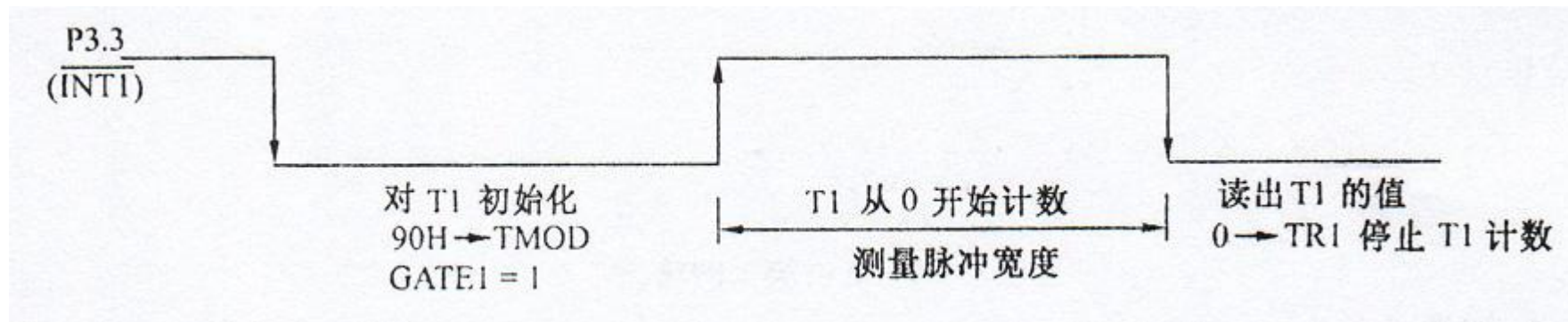
```
TL0INT: CLR    TR0      ; TL0中断服务，停止TL0计数
          SETB   ET1      ; 允许TH0中断
          SETB   TR1      ; 启动TH0
          CLR    P1.0     ; 信号输出低电平
          RETI
```

```
TH0INT: CPL    P1.0      ; TH0中断服务程序，输出方波
          MOV    TH0, #9CH ; 重装初值
          RETI
```

### 定时器/计数器T0与T1的编程和应用

#### 测量脉冲宽度（门控制位GATE的应用）

采用T1，利用门控制位GATE使T1的启动受 $\overline{\text{INT1}}$ （P3.3）的控制，实现对 $\overline{\text{INT1}}$ （P3.3）上正脉冲宽度的测量。



## 第6部分 MCS-51的定时器/计数器

### 定时器/计数器T0与T1的编程和应用

测量脉冲宽度（门控制位GATE的应用）

ORG	0000H	
RESET:	LJMP	MAIN ; 转主程序
ORG	0100H	
MAIN:	MOV	SP, #60H ; 设置堆栈指针
	MOV	TMOD, #90H ; T1为方式1定时, GATE位=1
	MOV	TL1, #00H ; T1置初值
	MOV	TH1, #00H
LOOP:	JB	P3.3, LOOP ; 等待 $\overline{\text{INT1}}$ 进入低电平
	SETB	TR1 ; 如果 $\overline{\text{INT1}}$ 进入低电平, 启动T1, 等待 $\overline{\text{INT1}}$ 进入高电平后T1开始计数
LOOP1:	JNB	P3.3, LOOP1 ; 等待 $\overline{\text{INT1}}$ 正跳变
LOOP2:	JB	P3.3, LOOP2 ; T1开始计数, 等待 $\overline{\text{INT1}}$ 负跳变
	CLR	TR1 ; 停止T1计数
	MOV	R0, #BUF0 ; BUF0为存储区地址
	MOV	@R0, TL1 ; TL1计数值送BUF0单元
	INC	R0
	MOV	@R0, TH1 ; TH1计数值送BUF0+1单元
HERE:	调用其他处理程序	
	SJMP	HERE

### 作业

一、16版教材 P131-第五部分“编程”的第1题。

二、16版教材 P131-第五部分“编程”的第2题。

(提示：占空比是1:10，高电平为40 $\mu$ s，波形如下图)

