# 第11章 STC单片机计数器/定时器原理及实现

何宾 2019.05

## TCON除了用于控制定时器/计数器T0和T1外,同时也可以锁存T0和T1溢出中断源和外部请求中断源等。

名字	地址	复位值	В7	В6	В5	B4	В3	B2	B1	ВО
TCON	0x88	00000000	TF1	TR1	TF0	TRO	IE1	IT1	IE0	IT0

- TF1,定时器/计数器1的溢出中断标志。
  - ◆ 当允许定时器/计数器1计数后,从初值开始加1计数。
  - ◆ 当产生溢出时,硬件将该位置1。此时,向CPU发出中断请求。如果 CPU响应该中断请求,则由硬件自动清0。
  - ◆ 该位也可通过软件轮询清0。

- TR1, 定时器/计数器1运行控制位。
  - 该位由软件置位和清零。
  - 当工作模式寄存器TMOD的GATE位(第7位)为0,且TR1为1时,允 许其开始计数。当TR1为0时,禁止计数。
  - 当工作模式寄存器TMOD的GATE位(第7位)为1, TR1为1, 且INT1 输入为高电平时, 才允许其开始计数。 (为什么? 内部结构决定)
- TF0, 定时器/计数器0的溢出中断标志。
  - 当允许定时器/计数器0计数后,从初值开始加1计数。
  - 当产生溢出时,硬件将该位置1。此时,向CPU发出中断请求。如果 CPU响应该中断请求,则由硬件自动清0。
  - 该位也可通过软件轮询清0。

- TR0, 定时器/计数器0运行控制位。
  - 该位由软件置位和清零。
  - 当工作模式寄存器TMOD的GATE位(第3位)为0,且TR0为1时,允 许其开始计数。否则,当TR0为0时,禁止计数。
  - 当工作模式寄存器TMOD的GATE位(第3位)为1,TR0为1,且INT0 输入为高电平时,才允许其开始计数。(为什么?内部结构决定)
- IE1,外部中断请求源 (INT1/P3.3)标志。
  - 当该位为1时,外部中断INT1向CPU发出中断请求。当CPU响应该中断后,由硬件自动清除该位。

- IT1, 外部中断源触发控制位。
  - 当该位为0时,上升沿/下降沿均可触发外部中断1;当该位为1时,只有下降沿可以触发外部中断1。
- IEO,外部中断请求源 (INTO/P3.2)标志。
  - 当该位为1时,外部中断INT0向CPU发出中断请求。当CPU响应该中断后,由硬件自动清除该位。
- ITO,外部中断源触发控制位。
  - 当该位为0时,上升沿或者下降沿均可触发外部中断0;当该位为1时, 只有下降沿可以触发外部中断0。

#### TMOD寄存器设置定时器/计数器0/1的定时/计数功能

名字	地址	复位值	В7	В6	В5	B4	В3	В2	В1	ВО
TMOD	0x89	00000000	GATE	C/T	M1	MO	GATE	C/T	M1	MO
	作用域	与为	定时器	1有关		与为	定时器	:0有:	关	

- GATE (TMOD.7) ,该位用于控制定时器/计数器1。
  - 当该位为1时,只有在INT1引脚为高,并且TCON寄存器的TR1位置1时,才能使能定时器/计数器1。
- ■GATE (TMOD.3) ,该位用于控制定时器/计数器0。
  - 当该位置1时,只有在INT0引脚为高,并且TCON寄存器的TR0位置1时,才能使能定时器/计数器0。

- C/T (TMOD.6) ,该位用于控制定时器/计数器1的工作模式。
  - 当该位设置为1时,定时器/计数器1工作在计数器模式,即:对引脚 T1/P3.5外部脉冲计数;
  - 当该位设置为0时,定时器/计数器1工作在定时器模式,即:对内部时钟进行计数。
- ■C/T (TMOD.2),该位用于控制定时器/计数器0的工作模式。
  - 当该位设置为1时,定时器/计数器0工作在计数器模式,即:对引脚 T0/P3.4外部脉冲计数;
  - 当该位设置为0时,定时器/计数器0工作在定时器模式,即:对内部时钟进行计数。

■ M1和M0 (TMOD.5和TMOD.4) , 定时器/计数器1模式选择。

M1	МО	工作模式选择
0	0	16位自动重新加载模式。当溢出时,将RL_TH1和RL_TL1的值自动重
		新加载到TH1和TL1中
0	1	16位不可自动重新加载模式。即:需要重新写TH1和TL1寄存器
1	0	8位自动重新加载模式。当溢出时,将TH1的值自动重新加载到TL1中
1	1	无效 <b>,</b> 停止计数

■ M1和M0 (TMOD.1和TMOD.0) , 定时器/计数器0模式选择。

M1	МО	工作模式
0	0	16位自动重新加载模式。当溢出时,将RL_THO和RL_TLO的值自动重
		新加载到THO和TLO中
0	1	16位不可自动重新加载模式。即:需要重新写THO和TLO寄存器
1	0	8位自动重新加载模式。当溢出时,将THO的值自动重新加载到TLO中
1	1	不可屏蔽中断的16位自动重装定时器

## 计数器/定时器寄存器组 --定时器/计数器1计数初值寄存器

■ 定时器/计数器1的计数初值寄存器TH1和TL1,它们用于保存定时器/计数器1的计数初值。

名字	地址	复位值	В7	В6	B5	B4	В3	B2	B1	ВО
TH1	0x8D	00000000	计数初值的高8位							
TL1	0x8B	00000000	计数初值的低8位							

## 计数器/定时器寄存器组 --定时器/计数器0的计数初值寄存器

■ 定时器/计数器0的初值寄存器TH0和TL0用于保存定时器/计数器0的计数初值。

名字	地址	复位值	В7	В6	<b>B</b> 5	B4	В3	B2	B1	В0
ТНО	0x8C	00000000	计数初值的高8位							
TL0	0x8A	00000000	计数初值的低8位							

## 计数器/定时器寄存器组 --辅助寄存器AUXR

定时器0/1的速度控制位T0x12/T1x12。

■ 当该位为0时, 定时器0/1是12分频; 当该位为1时, 定时器0/1不分频。

名字	地址	复位值	В7	В6	B5	B4	В3	B2	B1	ВО
AUXR	0x8E	000000 01	T0x12	T1x12	UART_MOx6	T2R	T2_C/T	T2x12	EXTRAM	S1ST2

注: (1) 复位后, STC15系列单片机定时器0、定时器1和定时器2与传统8051一样, 都是12分频。通过设置新的AUXR寄存器可禁止分频, 而直接使用SYSclk时钟驱动定时器。

(2)如果UART1/串口1用T1作为波特率发生器,则由T1x12决定UART1/串口1 是否分频。

## 计数器/定时器寄存器组 --辅助寄存器AUXR

- UART\_M0x6,串口1模式0的通信速率设置位。
  - 当该位为0时,串口1模式0为12分频;当该位为1时,串口1模式0为2分频。
- T2R, 定时器2允许控制位。
  - 当该位为0时,不允许定时器2运行;当该位为1时,允许定时器2运行。
- T2\_C/T, 控制定时器/计数器2的工作模式。
  - 当该位为0时,用作定时器,即:对内部系统时钟进行计数;当该位为1时,用作计数器(对引脚T2/P3.1的外部脉冲进行计数)。

## 计数器/定时器寄存器组 --辅助寄存器AUXR

- T2x12,定时器2速度控制位。
  - 当该位为0时,定时器2是12分频;当该位为1时,定时器2不分频。
- EXTRAM,内部/外部RAM存取控制位。
  - 当该位为0时,允许使用逻辑上在片外、物理上在片内的扩展数据 RAM区;当该位为1时,禁止使用逻辑上在片外、物理上在片内的扩 展数据RAM区。
- S1ST2, 串口1选择定时器2作为波特率发生器的控制位。
  - 当该位为0时,选择定时器1作为串口1的波特率发生器;当该位为1时,选择定时器2作为串口1的波特率发生器,此时释放定时器1,它可以作为独立的定时器使用。

## 计数器/定时器寄存器组 --定时器/计数器2计数初值寄存器

■ 定时器/计数器2的初值寄存器TH2和TL2用于保存定时器/计数器2的计数初值。

名字	地址	复位值	B7	В6	<b>B</b> 5	B4	В3	B2	B1	ВО
TH2	0xD6	00000000	计数初值的高8位							
TL2 注・対于完成	0xD7	00000000	白計畫加	□栽描台	计数	<b>女初值</b>	的低8	位		

<del>注:对于定时器2来说,只有16位自动重加载模式</del>

## 计数器/定时器寄存器组 --INT\_CLKO(AUXR2)

■ 通过INT\_CLKO寄存器的T0CLKO、T1CLKO和T2CLKO位,控制T0CLKO/P3.5、T1CLKO/P3.4和T2CLKO/P3.0的时钟输出。

名字	地址	复位值	В7	В6	В5	B4	В3	B2	В1	В0
AUXR2	0x8F	x000x000		EX4	ЕХЗ	EX2		T2CLK0	T1CLKO	TOCLKO
INT_CLKO										

- Exn (n=2、3、4) ,外部中断n (INTn) 允许位。
  - 当该位为1时,允许外部中断n产生中断事件;否则,当该位为0时,禁止外部中断n产生中断事件。



## 计数器/定时器寄存器组 --INT\_CLKO(AUXR2)

- 将P3.5/T1引脚配置为定时器0的时钟输出T0CLKO允许控制位。 当该位为 1时,将P3.5/T1引脚设置为定时器0的时钟输出T0CLKO,输出时钟频率 =T0溢出率/2。
- 口 如果运行在模式0 (16位自动重加载模式) 时,则:
- ①如果工作在定时器模式下,定时器/计数器T0是对内部系统时钟计数,则:

当T0工作在1T (AUXR.7/T0x12=1) 模式时,输出频率为:

{SYSclk/(65536 - [RL\_TH0,RL\_TL0])}/2

当T0工作在12T (AUXR.7/T0x12=0) 模式时, 输出频率为:

{(SYSclk/12)/(65536 - [RL TH0,RL TL0])}/2

②如果工作在计数器模式下,定时器/计数器T0是对外部脉冲输入 (P3.4/T0) 计数,输出时钟频率为:

{TO引脚输入时钟频率/(65536 - [RL\_THO,RL\_TLO])}/2



- 口 如果运行在模式2(8位自动重加载模式)时,则:
- ①如果工作在定时器模式下,定时器/计数器T0是对内部系统时钟计数,则:

当T0工作在1T (AUXR.7/T0x12=1) 模式时,输出频率为:

[SYSclk/(256 - TH0)]/2

当T0工作在12T (AUXR.7/T0x12=0) 模式时, 输出频率为:

[(SYSclk/12)/(256 - TH0)]/2

②如果工作在计数器模式下,定时器/计数器T0是对外部脉冲输入 (P3.4/T0) 计数,输出时钟频率为:

[T0引脚输入时钟频率/(256 - TH0)]/2

口 该位为0时,不允许将P3.5/T1引脚配置为定时器0的时钟输出。

## 计数器/定时器寄存器组 --INT\_CLKO(AUXR2)

#### T1CLKO

- 口 将P3.4/T0引脚配置为定时器1的时钟输出T1CLKO允许控制位。
- 山 当该位为1时,将P3.4/T0引脚设置为定时器1的时钟输出T1CLKO,输出时钟频率=T1溢出率/2。如果运行在模式0(16位自动重加载模式)时,则:
- ①如果工作在定时器模式下,定时器/计数器T1是对内部系统时钟计数,则:

当T1工作在1T (AUXR.6/T1x12=1) 模式时, 输出频率为:

{SYSclk/(65536 - [RL\_TH1,RL\_TL1])}/2

当T0工作在12T (AUXR.6/T1x12=0) 模式时,输出频率为:

{(SYSclk/12)/(65536 - [RL\_TH1,RL\_TL1])}/2

②如果工作在计数器模式下,定时器/计数器T1是对外部脉冲输入 (P3.5/T1) 计数,输出时钟频率为:

{T1引脚输入时钟频率/(65536 - [RL\_TH1,RL\_TL1])}/2



- 口 如果运行在模式2 (8位自动重加载模式) 时,则:
- ①如果工作在定时器模式下,定时器/计数器T1是对内部系统时钟计数,则:

当T1工作在1T (AUXR.6/T1x12=1) 模式时,输出频率为:

[SYSclk/(256 - TH1)]/2

当T1工作在12T (AUXR.6/T1x12=0) 模式时,输出频率为:

[(SYSclk/12)/(256 - TH1)]/2

②如果工作在计数器模式下,定时器/计数器T1是对外部脉冲输入 (P3.5/T1) 计数,输出时钟频率为:

[T1引脚输入时钟频率/(256 - TH1)]/2

口 该位为0时,不允许将P3.4/T0引脚配置为定时器1的时钟输出。



#### **■ T2CLKO**

- 口 将P3.0/T1引脚配置为定时器2的时钟输出T2CLKO允许控制位。
- □ 当该位为1时,P3.0引脚设置为定时器2的时钟输出T2CLKO,输出时钟频率=T2溢出率/2。该定时器只能运行在模式0(16位自动重加载模式),则:
- ①如果工作在定时器模式下,定时器/计数器T2是对内部系统时钟计数,则:

当T2工作在1T (AUXR.2/T2x12=1) 模式时,输出频率为:

{SYSclk/(65536 - [RL\_TH2,RL\_TL2])}/2

当T2工作在12T (AUXR.2/T2x12=0) 模式时,输出频率为:

{(SYSclk/12)/(65536 - [RL TH2,RL TL2])}/2

②如果工作在计数器模式下,定时器/计数器T0是对外部脉冲输入 (P3.1/T2) 计数,输出时钟频率为:

{T2引脚输入时钟频率/(65536 - [RL TH2,RL TL2])}/2

口 该位为0时,不允许将P3.0引脚配置为定时器2的时钟输出。

## 计数器/定时器寄存器组 --中断控制寄存器IE

名字	地址	复位值	B7	В6	B5	B4	В3	B2	B1	В0
IE	0xA8	00000000	EA	ELVD	EADC	ES	ET1	EX1	ET0	EX0

- ET1, 定时器/计数器T1溢出中断允许位。
  - 当该位为1时,允许T1溢出中断;当该位为0时,禁止T1溢出中断。
- ETO, 定时器/计数器TO溢出中断允许位。
  - 当该位为1时,允许T0溢出中断;当该位为0时,禁止T0溢出中断。

## 计数器/定时器寄存器组 --中断优先级控制寄存器IP

比特	地址	复位值	В7	В6	B5	B4	В3	B2	B1	ВО
名字	0xB8	00000000	PPCA	PLVD	PADC	PS	PT1	PX1	PT0	PX0

- PT1,定时器1中断优先级控制位。
  - 当该位为0时,定时器1中断为最低优先级中断(优先级为0);当该位为1时,定时器1中断为最高优先级中断(优先级为1)。
- PT0, 定时器0中断优先级控制位。
  - 当该位为0时,定时器0中断为最低优先级中断(优先级为0);当该位为1时,定时器0中断为最高优先级中断(优先级为1)。

#### 该寄存器用于控制定时器/计数器3和4的工作模式

名字	地址	复位值	В7	В6	B5	B4	В3	B2	B1	В0
T4T3M	0xD1	00000000	T4R	T4_C/T	T4x12	T4CLK0	T3R	T3_C/T	T3x12	T3CLK0

- T4R, 定时器4允许控制位。
  - 当该位为0时,不允许定时器4运行;当该位为1时,允许定时器4运行。
- T4\_C/T,控制定时器/计数器4的工作模式。
  - 当该位为0时,用作定时器,即:对内部系统时钟进行计数;当该位为 1时,用作计数器(对引脚T4/P0.7的外部脉冲进行计数)。

- T4x12, 定时器4速度控制位。
  - 当该位为0时,定时器4是传统8051单片机的速度,即: 12分频; 当该位为1时, 定时器4的速度是传统8051单片机速度的12倍, 即: 不分频。
- T4CLKO,定时器4的时钟输出T4CLKO允许控制位。
  - 当该位为1时,将P0.6引脚设置为定时器4的时钟输出T4CLKO,输出时钟频率=T4溢出率/2。

该定时器只能运行在模式0 (16位自动重加载模式),则:

①如果工作在定时器模式下,定时器/计数器T4是对内部系统时钟计数,则:

当T4工作在1T (T4T3M.5/T4x12=1) 模式时, 输出频率为:

{SYSclk/(65536 - [RL\_TH4,RL\_TL4])}/2

当T4工作在12T (T4T3M.5/T4x12=0) 模式时,输出频率为:

{(SYSclk/12)/(65536 - [RL TH4,RL TL4])}/2

②如果工作在计数器模式下,定时器/计数器T4是对外部脉冲输入 (P0.7/T4) 计数,输出时钟频率为:

{T4引脚输入时钟频率/(65536 - [RL\_TH4,RL\_TL4])}/2

口 该位为0时,不允许将P0.6引脚配置为定时器4的时钟输出。

## 计数器/定时器寄存器组 --定时器T4计数初值寄存器

■ 定时器4只有16位自动重加载模式,其计数初值保存在寄存器 TH4和TL4中。

名字	地址	复位值	В7	В6	B5	B4	В3	B2	B1	ВО
TH4	0xD2	00000000	计数初值的高8位							
TL4	0xD3	00000000	计数初值的低8位							

#### T3R

□ 定时器3允许控制位。当该位为0时,不允许定时器3运行;当该位为1时, 允许定时器3运行。

#### T3\_C/T

□ 控制定时器/计数器3的工作模式。当该位为0时,用作定时器,即:对内部系统时钟进行计数;当该位为1时,用作计数器(对引脚T3/P0.5的外部脉冲进行计数)。

#### ■ T3x12

□ 定时器3速度控制位。当该位为0时,定时器3是12分频;当该位为1时,定时器3不分频。

- □ 将P0.4引脚配置为定时器3的时钟输出T3CLKO允许控制位。
- □ 当该位为1时, P0.4引脚设置为定时器3的时钟输出T3CLKO, 输出时钟频率=T3溢出率/2。该定时器只能运行在模式0(16位自动重加载模式),则:
- ①如果工作在定时器模式下,定时器/计数器T3是对内部系统时钟计数,则:

当T3工作在1T (T4T3M.1/T3x12=1) 模式时,输出频率为:

{SYSclk/(65536 - [RL\_TH3,RL\_TL3])}/2

当T3工作在12T (T4T3M.1/T3x12=0) 模式时,输出频率为:

{(SYSclk/12)/(65536 - [RL\_TH3,RL\_TL3])}/2

②如果工作在计数器模式下,定时器/计数器T3是对外部脉冲输入 (P0.5/T3) 计数,输出时钟频率为:

{T3引脚输入时钟频率/(65536 - [RL\_TH3,RL\_TL3])}/2

□ 该位为0时,不允许将P0.4引脚配置为定时器3的时钟输出。

## 计数器/定时器寄存器组 --定时器T3计数初值寄存器

■ 对于定时器3来说,只有16位自动重加载模式,其计数初值保存在寄存器TH3和TL3中。

名字	地址	复位值	В7	В6	<b>B</b> 5	B4	В3	B2	B1	В0
TH3	0xD4	00000000	计数初值的高8位							
TL3	0xD5	00000000	计数初值的低8位							

### 计数器/定时器寄存器组 --中断控制寄存器IE2

#### 该寄存器可用于控制定时器T2~定时器T4的中断

- 该寄存器位于特殊功能寄存器地址为0xAF的位置。
- 当复位后,该寄存器的值为x0000000B。

名字	地址	复位值	В7	В6	B5	В4	В3	B2	B1	ВО
IE2	0xAF	x00000000		ET4	ET3	ES4	ES3	ET2	ESP1	ES2

- ETn(n=4、3、2) ,定时器4/3/2中断允许位。
  - 当该位为1时,允许定时器4/3/2产生中断;当该位为0时,禁止定时器4/3/2产生中断。

## 计数器/定时器寄存器组 -- 中断控制寄存器IE2

- ESn(n=4、3、2) , 串口4/3/2中断允许位。
  - 当该位为1时,允许串口4/3/2产生中断;当该位为0时,禁止串口4/3/2 产生中断。
- ESPI, SPI中断允许位。
  - 当该位为1时,允许SPI产生中断;当该位为0时,禁止SPI产生中断。

#### 【例】定时器/计数器0自动加载模式C语言描述的例子

该例子将通过定时器生成一个频率为1Hz的时钟,并通过单片机P3.5端口输出。

```
#include "reg51.h"

#define TIMS 3036

sfr AUXR =0x8E;

sfr AUXR2 =0x8F;

sfr CLK_DIV=0x97;

void timer_0() interrupt 1

P46=!P46;
P47=!P47;

//E时器/计数器0的计数初值

//定时器/计数器0的计数初值

//声明AUXR2寄存器的地址为0x8F

//声明CLK_DIV寄存器的地址为0x97

//声明定时器/计数器0中断服务程序

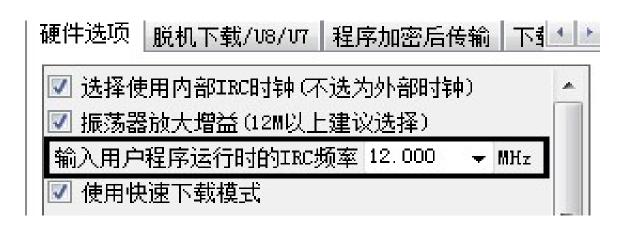
//P4.6端口取反
//P4.7端口取反
```

main()

```
CLK DIV=0x03;
               //CLV DIV=3,将主时钟8分频后作为SYSclk
               //TIMS低八位给定时器计数初值寄存器TL0
TL0=TIMS;
               //TIMS高八位给定时器计数初值寄存器TH0
TH0=TIMS>>8;
               //AUXR最高位置0, SYSclk/12作定时器时钟
AUXR&=0x7F;
AUXR2 = 0x01;
              //AUXR2最低位置1, P3.5端口输出T0CLKO
TMOD=0x00;
               //定时器0工作模式为16位自动重加载模式
              //设置P4.6初值为0, 灯亮
P46=0;
              //设置P4.7初值为0, 灯亮
P47=0;
               //启动定时器/计数器0
TR0=1;
               //使能定时器/计数器0中断
ET0=1;
               //使能CPU全局中断,允许中断请求
EA=1;
while(1);
               //无限循环
```

#### 下面对该设计进行验证和分析, 步骤如下:

- 打开STC-ISP软件,将IRC频率设置为12.000MHz。
- 下载设计到STC提供的学习板上的单片机中。
- 打开示波器,将探头连接到学习板上P3.5端口上(注意:探头
  - 一定要和板子共地)



#### 【例】定时器/计数器0自动加载模式C语言描述的例子

该例子将通过外部中断0控制定时器1的工作过程。

```
main()
                //将TIMS的低8位赋值给TL0
 TL0=TIMS;
                //将TIMS的高8位赋值给TH0
 TH0=TIMS>>8;
 AUXR&=0x7F;
                // AUXR最高位置0, SYSclk/12作定时器时钟
                //设置GATE为1, 定时器0与INT0引脚有关
 TMOD=0x08;
               //设置P4.6初值为0, 灯亮
 P46=0:
               //设置P4.7初值为0, 灯亮
 P47=0;
 TR0=1;
               //启动定时器/计数器0
               //允许定时器/计数器0中断
 ET0=1;
               //CPU允许响应中断请求
 EA=1;
 while(1);
```

#### 【例】定时器/计数器0自动加载模式C语言描述的例子

该例子将实现对外部脉冲进行计数功能。

```
#include "reg51.h"
                        //定义TIMS的值为3036
#define TIMS 3036
                        //声明定时器/计数器0中断
void timer 0() interrupt 1
                       //P4.6端口取反
   P46=!P46;
                       //P4.7端口取反
   P47=!P47;
main()
                        //TIMS低8位赋值给TL0寄存器
  TL0=TIMS;
                        //TIMS高8位赋值给TH0寄存器
  TH0=TIMS>>8;
                        //配置成计数器16位重加载模式
  TMOD=0x04;
```

```
      P46=0;
      //设置P4.6初值为0, 灯亮

      P47=0;
      //设置P4.7初值为0, 灯亮

      TR0=1;
      //启动定时器/计数器0

      ET0=1;
      //允许定时器/计数器0中断

      EA=1;
      //CPU允许响应中断请求

      while(1);
```

#### 下面对该设计进行验证和分析, 步骤如下:

- 打开STC-ISP软件,将IRC频率设置为12.000MHz。
- 下载设计到STC提供的学习板上的单片机中。
- 打开信号源,信号源输出为TTL/CMOS。将信号源的输出连接 到STC学习板的P34端口上(注意:信号源和STC学习板共地)。

## 定时器/计数器2工作模式 --应用实例

#### 【例】定时器/计数器2自动加载模式C语言描述的例子

```
#include "reg51.h"
#define TIMS 3036
                         //声明AUXR寄存器的地址0x8E
sfr AUXR = 0x8E;
                         //声明IE2寄存器的地址0xAF
sfr IE2 = 0xAF;
                         //声明TH2寄存器的地址0xD6
sfr TH2 = 0xD6;
                         //声明TL2寄存器的地址0xD7
sfr TL2 = 0xD7;
                         //声明CLK DIV寄存器的地址0x97
sfr CLK DIV=0x97;
                        //声明定时器/计数器2中断服务程序
void timer 2() interrupt 12
                         //P4.6端口取反
   P46=!P46;
                         //P4.7端口取反
   P47=!P47;
```

## 定时器/计数器2工作模式 --应用实例

```
main()
                 //主时钟8分频
  CLK DIV=0x03;
                 //TIMS低8位赋值给TL2寄存器
 TL2=TIMS;
                //TIMS高8位赋值给TH2寄存器
 TH2=TIMS>>8;
                //启动定时器/计数器2, 定时器工作模式, 不分频
 AUXR = 0x10;
               //P4.6端口置0, 灯亮
 P46=0;
               //P4.7端口置0, 灯亮
 P47=0;
                //允许定时器2中断
 IE2|=0x04;
                //CPU允许响应中断请求
 EA=1;
 while(1);
```

## 定时器/计数器3工作模式 --应用实例

#### 【例】定时器/计数器3自动加载模式C语言描述的例子

```
#include "reg51.h"
                         //定义TIMS的值
#define TIMS 3036
                          //声明CLK DIV寄存器的地址
sfr CLK DIV=0x97;
                         //声明IE2寄存器的地址
sfr IE2 = 0xAF;
                         //声明TH3寄存器的地址
sfr TH3 = 0xD4;
                         //声明TL3寄存器的地址
sfr TL3 = 0xD5;
                         //声明T4T3M寄存器的地址
sfr T4T3M = 0xD1;
                         //声明定时器/计数器3的中断服务程序
void timer 3() interrupt 19
                         //P4.6端口取反
   P46=!P46;
                         //P4.7端口取反
   P47=!P47;
```

## 定时器/计数器3工作模式 --应用实例

```
main()
                //主时钟8分频,作为系统时钟
  CLK DIV=0x03;
                //TIMIS的低8位赋值给TL3寄存器
 TL3=TIMS;
                //TIMS的高8位赋值给TH3寄存器
 TH3=TIMS>>8;
                //启动定时器/计数器3,工作模式定时器
 T4T3M=0x08;
                //P4.6置0, 灯亮
 P46=0;
                //P4.7置0, 灯亮
 P47=0;
 IE2|=0x20;
                //允许定时器/计数器3中断请求
                //CPU允许响应中断请求
 EA=1;
 while(1);
```