



第8章 STC单片机计数器/定时器 原理及实现

何宾

2015.02

本章主要内容

- 计数器/定时器模块概述
- 计数器/定时器寄存器组
- 计数器/定时器工作模式原理和实现

计数器/定时器模块概述

STC15W4K32S4系列单片机设置了5个16位定时器/计数器，即：T0、T1、T2、T3以及T4。这5个16位定时器/计数器可以配置为计数工作模式或者定时工作模式。

- 对于定时器/计数器T0和T1来说，通过特殊功能寄存器TMOD相对应的控制位C/T，选择T0/T1工作在定时器还是计数器模式。
- 对于定时器/计数器T2来说，通过特殊功能寄存器AUXR中相对应的控制位T2_C/T，选择T2工作在定时器还是计数器模式。

计数器/定时器模块概述

- 对于定时器/计数器T3来说，通过特殊功能寄存器T4T3M中相对应的控制位T3_C/T，选择T3工作在定时器还是计数器模式。
- 对于定时器/计数器T4来说，通过特殊功能寄存器T4T3M中相对应的控制位T4_C/T，选择T4工作在定时器还是计数器模式。

计数器/定时器模块概述

对于定时器和计数器来说，其核心部件就是一个做加法运算的计数器，其本质就是对脉冲进行计数。它们的区别在于计数脉冲来源不同：

- **如果计数脉冲来自系统时钟，则为定时方式，此时定时器/计数器每12个时钟或者1个时钟就得到一个计数脉冲，计数值加1。**
- **如果计数脉冲来自单片机外部引脚，则为计数方式，每来一个脉冲，则计数值加1。**

计数器/定时器模块概述



- 定时器/计数器0有4种工作模式：
 - 模式0（16位自动重新加载模式）；
 - 模式1（16位不可重加载模式）；
 - 模式2（8位自动重加载模式）；
 - 模式3（不可屏蔽中断的16位自动重加载模式）。
- 定时器/计数器1没有模式3，其他模式和定时器/计数器0相同。
- 定时器/计数器2的工作模式固定为16位自动重加载模式。
 - 它可以用作定时器，也可以用作串口波特率发生器和可编程时钟输出。
- 定时器/计数器3和4与定时器/计数器2的工作模式相同。

计数器/定时器寄存器组

--定时器/计数器0/1控制寄存器TCON

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。

- 该寄存器位于特殊功能寄存器地址为0x88的位置。
- 当复位后，该寄存器的值为“00000000”。

TCON寄存器各位的含义

比特	B7	B6	B5	B4	B3	B2	B1	B0
名字	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

计数器/定时器寄存器组

--定时器/计数器0/1控制寄存器TCON

■ 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，则由硬件自动的清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。否则，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。

计数器/定时器寄存器组

--定时器/计数器0/1控制寄存器TCON

■ 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，则由硬件自动的清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。

计数器/定时器寄存器组

--定时器/计数器0/1控制寄存器TCON

■ IE1

- 外部中断请求源（INT1/P3.3）标志。当该位为1时，外部中断向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，上升沿或者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。

■ IE0

- 外部中断请求源（INT0/P3.2）标志。当该位为1时，外部中断向CPU发出中断请求。当CPU响应该中断后，由硬件自动清除该位。

■ IT0

- 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。

计数器/定时器寄存器组

--定时器/计数器工作模式寄存器TMOD

定时器/计数器0和定时器/计数器1的定时或者计数功能由TMOD寄存器的控制位C/T进行选择。

TMOD寄存器各位的含义

比特	B7	B6	B5	B4	B3	B2	B1	B0
名字	GATE	C/T	M1	M0	GATE	C/T	M1	M0
作用域	与定时器1有关				与定时器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，才能打开定时器/计数器1。

计数器/定时器寄存器组

--定时器/计数器工作模式寄存器TMOD

■ 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工作在计数器模式下，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。

■ M1和M0 (TMOD.5和TMOD.4)

- 定时器/计数器1模式选择，如下表所示。

计数器/定时器寄存器组

--定时器/计数器工作模式寄存器TMOD

定时器/计数器1模式选择

M1	M0	工作模式
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，停止计数

计数器/定时器寄存器组

--定时器/计数器工作模式寄存器TMOD

- 对于定时器/计数器1来说，存在TH1和TL1它们用于保存计数的初值。
 - TH1保存计数初值的高8位，TL1保存计数初值的低8位。
 - TH1和TL1分别位于特殊功能寄存器地址为0x8D和0x8B的位置。
 - 在复位后，它们值均为“00000000”。

计数器/定时器寄存器组

--定时器/计数器工作模式寄存器TMOD

■ 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，才能打开定时器/计数器0。

■ 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工作在计数器模式下，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。

计数器/定时器寄存器组

--定时器/计数器工作模式寄存器TMOD

■ M1和M0 (TMOD.1和TMOD.0)

□ 定时器/计数器0模式选择，如下表所示。

定时器/计数器0模式选择

M1	M0	工作模式
0	0	16位自动重新加载模式。当溢出时，将RL_TH0和RL_TL0的值自动重新加载到TH0和TL0中
0	1	16位不可自动重新加载模式。即：需要重新写TH0和TL0寄存器
1	0	8位自动重新加载模式。当溢出时，将TH0的值自动重新加载到TL0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/计数器工作模式寄存器TMOD

- 对于定时器/计数器0来说，存在TH0和TL0它们用于保存计数的初值。
 - TH0保存计数初值的高8位，TL0保存计数初值的低8位。
 - TH0和TL0分别位于特殊功能寄存器地址为0x8C和0x8A的位置，
 - 在复位后，它们值均为“00000000”。

计数器/定时器寄存器组

--辅助寄存器AUXR

STC15系列单片机在复位后，定时器0、定时器1和定时器2和传统8051一样，都是12分频。

- 可以通过设置新增加的AUXR寄存器来禁止分频，而直接使用SYSclk时钟驱动定时器。
- 该寄存器位于特殊功能寄存器地址为0x8E的位置。
- 当复位后，该寄存器的值为“00000001”。

辅助寄存器AUXR

比特	B7	B6	B5	B4	B3	B2	B1	B0
名字	T0x12	T1x12	UART_M0x6	T2R	T2_C/T	T2x12	EXTRAM	S1ST2

计数器/定时器寄存器组

--辅助寄存器AUXR

■ T0x12

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

■ T1x12

- 定时器1速度控制位。当该位为0时，定时器1为12分频；当该位为1时，定时器1不分频。

注：如果UART1/串口用T1作为波特率发生器，则有T1x12决定UART1/串口1是否分频。

■ 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。

计数器/定时器寄存器组

--辅助寄存器AUXR

■ 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）。

注：对于定时器2来说，只有16位自动重加载模式，TH2保存计数初值的高8位，TL2保存计数初值的低8位。它们分别位于特殊功能寄存器地址为0xD6和0xD7的位置。当复位后，它们的值为“00000000”。

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（UART1）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（UART）的波特率发生器；当该位为1时，选择定时器2作为串口1（UART）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

--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。

- 该寄存器在特殊功能寄存器地址为0x8F的位置。
- 当复位后，该寄存器的值为“x000x000”。

INT_CLKO (AUXR2) 寄存器各位含义

比特	B7	B6	B5	B4	B3	B2	B1	B0
名字	--	EX4	EX3	EX2	--	T2CLKO	T1CLKO	T0CLKO

计数器/定时器寄存器组

--INT_CLKO(AUXR2)

■ EX4

- 外部中断4 (INT4) 允许位。当该位为1时，允许外部中断4产生中断事件；否则，当该位为0时，禁止外部中断4产生中断事件。

■ EX3

- 外部中断3 (INT3) 允许位。当该位为1时，允许外部中断3产生中断事件；否则，当该位为0时，禁止外部中断3产生中断事件。

■ EX2

- 外部中断2 (INT2) 允许位。当该位为1时，允许外部中断2产生中断事件；否则，当该位为0时，禁止外部中断2产生中断事件。

计数器/定时器寄存器组

--INT_CLKO(AUXR2)

■ T0CLKO

□ 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。

□ 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：

$$\{\text{SYSclk}/(65536 - [\text{RL_TH0}, \text{RL_TL0}])\}/2$$

当T0工作在12T（AUXR.7/T0x12=0）模式时，输出频率为：

$$\{(\text{SYSclk}/12)/(65536 - [\text{RL_TH0}, \text{RL_TL0}])\}/2$$

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：

$$\{\text{T0引脚输入时钟频率}/(65536 - [\text{RL_TH0}, \text{RL_TL0}])\}/2$$

计数器/定时器寄存器组

--INT_CLKO(AUXR2)

□ 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：

$$[\text{SYSclk}/(256 - \text{TH0})]/2$$

当T0工作在12T（AUXR.7/T0x12=0）模式时，输出频率为：

$$[(\text{SYSclk}/12)/(256 - \text{TH0})]/2$$

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：

$$[\text{T0引脚输入时钟频率}/(256 - \text{TH0})]/2$$

■ 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

- 将P3.4/T0引脚配置为定时器1的时钟输出T1CLKO允许控制位。
- 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：

$$\{T1引脚输入时钟频率/(65536 - [RL_TH1,RL_TL1])\}/2$$

计数器/定时器寄存器组

--INT_CLKO(AUXR2)

□ 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（ $AUXR.6/T1 \times 12 = 1$ ）模式时，输出频率为：

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

当T1工作在12T（ $AUXR.6/T1 \times 12 = 0$ ）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：

$$[T1引脚输入时钟频率/(256 - TH1)]/2$$

□ 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。

计数器/定时器寄存器组

--INT_CLKO(AUXR2)

■ T2CLKO

- 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：

$$\{\text{SYSclk}/(65536 - [\text{RL_TH2}, \text{RL_TL2}])\}/2$$

当T2工作在12T（AUXR.2/T2x12=0）模式时，输出频率为：

$$\{(\text{SYSclk}/12)/(65536 - [\text{RL_TH2}, \text{RL_TL2}])\}/2$$

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：

$$\{\text{T2引脚输入时钟频率}/(65536 - [\text{RL_TH2}, \text{RL_TL2}])\}/2$$

- 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

--定时器T0和T1中断允许控制寄存器IE

中断控制寄存器IE

- 该寄存器位于特殊功能寄存器地址为0xA8的位置。
- 当复位后，该寄存器的值为“00000000”。

中断控制寄存器IE各位含义

比特	B7	B6	B5	B4	B3	B2	B1	B0
名字	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。

■ ET0

- 定时器/计数器T0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

--定时器T0和T1中断优先级控制寄存器IP

中断优先级控制寄存器IP

- 该寄存器位于特殊功能寄存器地址为0xB8的位置。
- 当复位后，该寄存器的值为“00000000”。

中断控制寄存器IP各位含义

比特	B7	B6	B5	B4	B3	B2	B1	B0
名字	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）。

计数器/定时器寄存器组



---定时器T4和定时器T3控制寄存器T4T3M

该寄存器用于控制定时器/计数器3和4的工作模式

- 该寄存器位于特殊功能寄存器地址为0xD1的位置。
- 当复位后，该寄存器的值为“00000000”。

定时器T4和定时器T3控制寄存器T4T3M各位含义

比特	B7	B6	B5	B4	B3	B2	B1	B0
名字	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）。

如需原始PPT文件请点击此处

<http://www.gpnewtech.com/ppt>

计数器/定时器寄存器组



---定时器T4和定时器T3控制寄存器T4T3M

■ 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

- 将P0.6引脚配置为定时器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。

计数器/定时器寄存器组



---定时器T4和定时器T3控制寄存器T4T3M

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（ $T4T3M.5/T4 \times 12 = 1$ ）模式时，输出频率为：

$$\{SYSclk / (65536 - [RL_TH4, RL_TL4])\} / 2$$

当T4工作在12T（ $T4T3M.5/T4 \times 12 = 0$ ）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：

$$\{T4引脚输入时钟频率 / (65536 - [RL_TH4, RL_TL4])\} / 2$$

□ 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和定时器T3控制寄存器T4T3M

- 对于定时器4来说，只有16位自动重加载模式
 - 其计数初值保存在TH4和TL4寄存器中。
 - TH4保存计数初值的高8位，TL4保存计数初值的低8位。
 - 它们分别位于特殊功能寄存器地址为0xD2和0xD3的位置。
 - 当复位后，它们的值为“00000000”。

计数器/定时器寄存器组

---定时器T4和定时器T3控制寄存器T4T3M

■ 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。

计数器/定时器寄存器组



--定时器T4和定时器T3控制寄存器T4T3M

■ T3CLKO

- 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（ $T4T3M.1/T3 \times 12 = 1$ ）模式时，输出频率为：

$$\{SYSclk / (65536 - [RL_TH3, RL_TL3])\} / 2$$

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

$$\{(SYSclk / 12) / (65536 - [RL_TH3, RL_TL3])\} / 2$$

计数器/定时器寄存器组

---定时器T4和定时器T3控制寄存器T4T3M

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：

$$\{T3\text{引脚输入时钟频率}/(65536 - [RL_TH3, RL_TL3])\}/2$$

□ 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。

■ 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TH3和TL3寄存器中。

□ TH3保存计数初值的高8位，TL3保存计数初值的低8位。

□ 它们分别位于特殊功能寄存器地址为0xD4和0xD5的位置。

□ 当复位后，它们的值为“00000000”。

计数器/定时器寄存器组

--定时器T2、T3和T4的中断控制寄存器IE2

该寄存器的某些位可以用于控制定时器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“x0000000”。

中断控制寄存器IP各位含义

比特	B7	B6	B5	B4	B3	B2	B1	B0
名字	--	ET4	ET3	ES4	ES3	ET2	ESP1	ES2

■ ET4

- 定时器4中断允许位。当该位为1时，允许定时器4产生中断；当该位为0时，禁止定时器4产生中断。

计数器/定时器寄存器组

—定时器T2、T3和T4的中断控制寄存器IE2

■ ET3

- 定时器3中断允许位。当该位为1时，允许定时器3产生中断；当该位为0时，禁止定时器3产生中断。

■ ET2

- 定时器2中断允许位。当该位为1时，允许定时器2产生中断；当该位为0时，禁止定时器2产生中断。

■ ES4

- 串口4中断允许位。当该位为1时，允许串口4产生中断；当该位为0时，禁止串口4产生中断。

计数器/定时器寄存器组

--定时器T2、T3和T4的中断控制寄存器IE2

■ ES3

- 串口3中断允许位。当该位为1时，允许串口3产生中断；当该位为0时，禁止串口3产生中断。

■ ES2

- 串口2中断允许位。当该位为1时，允许串口2产生中断；当该位为0时，禁止串口2产生中断。

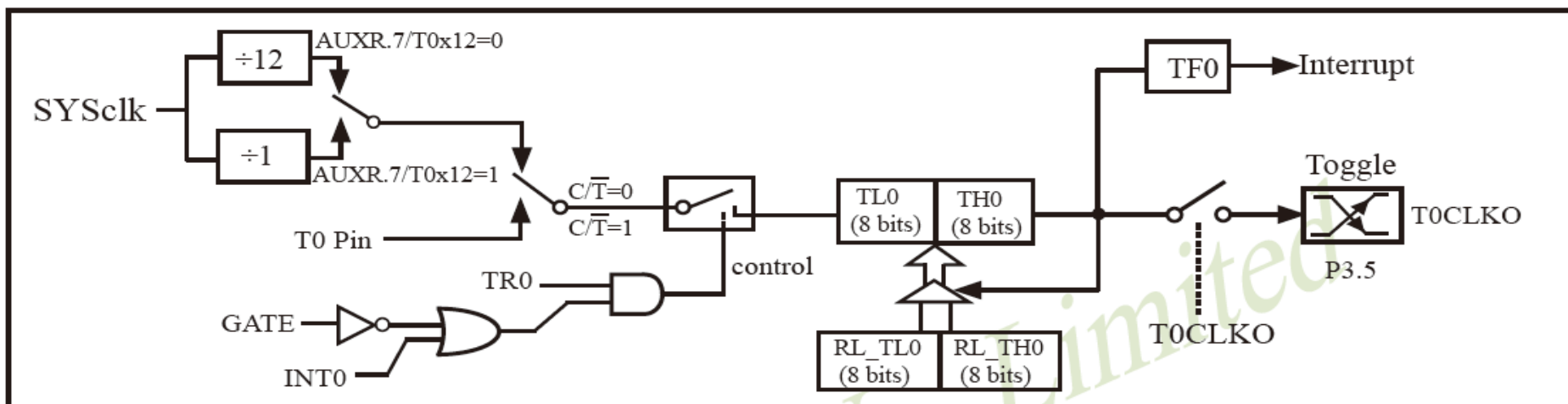
■ ESPI

- 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。

计数器/定时器工作模式原理和实现

--定时器/计数器0工作模式

模式0（16位自动重加载模式）



■ GATE、INT0和TR0之间的关系

- GATE、INT0和TR0之间的关系，决定定时器/计数器是否能正常工作。这三个信号通过逻辑或门和逻辑与门产生control信号，用于控制内部SYSClk信号或者外部脉冲通过T0引脚是否能接入该定时器/计数器。

计数器/定时器工作模式原理和实现

--定时器/计数器0工作模式

GATE、INT0和TR0之间的关系

GATE	INT0	TR0	功能
0	0	0	不启动定时器
0	0	1	启动定时器
0	1	0	不启动定时器
0	1	1	启动定时器
1	0	0	不启动定时器
1	0	1	不启动定时器
1	1	0	不启动定时器
1	1	1	启动定时器

■ AUXR.7/T0x12比特位

□ 当该位为0时，通过开关将SYSclk/12后得到的时钟接入到定时器/计数器0中；当该位为1时，通过开关将SYSclk直接接入到定时器/计数器0中。

计数器/定时器工作模式原理和实现

--定时器/计数器0工作模式

■ C/T比特位

- 当该位为0时，将内部的时钟引入到定时器/计数器0中；当该位为1时，将T0引脚上的外部脉冲信号引入定时器/计数器0。

■ TF0比特位

- 当该位为1时，该模块产生中断。

■ T0CLKO比特位

- 当该位为1时，将定时器/计数器0产生的时钟送给P3.5引脚；与P3.5引脚断开。此时，P3.5引脚作为普通I/O使用。

计数器/定时器工作模式原理和实现

--定时器/计数器0工作模式

【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 TMS 3036

sfr AUXR = 0x8E;
sfr AUXR2 = 0x8F;
sfr CLK_DIV = 0x97;

void timer_0() interrupt 1
{
    P46=!P46;
    P47=!P47;
}
```

//定时器/计数器0的计数初值
//声明AUXR寄存器的地址为0x8E
//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

计数器/定时器工作模式原理和实现

--定时器/计数器0工作模式

main()

{

```
CLK_DIV=0x03;           //CLK_DIV=3，将主时钟8分频后作为SYSclk
TL0=TIMS;               //TIMS低八位给定时器计数初值寄存器TL0
TH0=TIMS>>8;           //TIMS高八位给定时器计数初值寄存器TH0
AUXR&=0x7F;             //AUXR最高位置0，SYSclk/12作定时器时钟
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
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;                  //使能定时器/计数器0中断
EA=1;                   //使能CPU全局中断，允许中断请求
while(1);               //无限循环
```

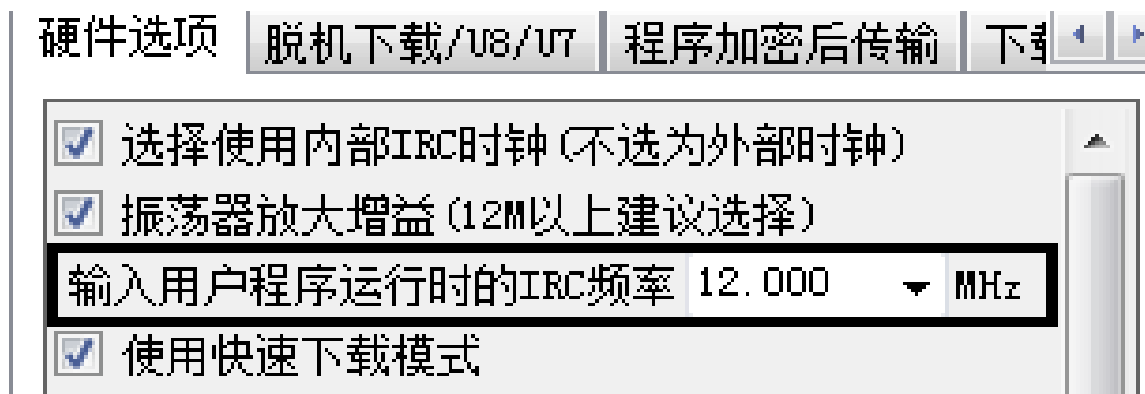
}

计数器/定时器工作模式原理和实现

--定时器/计数器0工作模式

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）



计数器/定时器工作模式原理和实现

--定时器/计数器0工作模式

【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。

```
#include "reg51.h"
#define TIMS 3036
sfr AUXR = 0x8E;           //声明AUXR寄存器的地址为0x8E
void timer_0() interrupt 1  //声明定时器中断
{
    P46=!P46;               //P4.6端口取反
    P47=!P47;               //P4.7端口取反
}
```

计数器/定时器工作模式原理和实现

--定时器/计数器0工作模式

```
main()
```

```
{
```

```
    TL0=TIMS;           //将TIMS的低8位赋值给TL0
    TH0=TIMS>>8;        //将TIMS的高8位赋值给TH0
    AUXR&=0x7F;         //AUXR最高位置0，SYSclk/12作定时器时钟
    TMOD=0x08;          //设置GATE为1，定时器0与INT0引脚有关
    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);
```

```
}
```

计数器/定时器工作模式原理和实现

--定时器/计数器0工作模式

【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"
#define TMS 3036          //定义TMS的值为3036
void timer_0() interrupt 1 //声明定时器/计数器0中断
{
    P46=!P46;             //P4.6端口取反
    P47=!P47;             //P4.7端口取反
}
main()
{
    TL0=TMS;               //TMS低8位赋值给TL0寄存器
    TH0=TMS>>8;           //TMS高8位赋值给TH0寄存器
    TMOD=0x04;            //配置成计数器16位重加载模式
```

计数器/定时器工作模式原理和实现

--定时器/计数器0工作模式

```
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);
}
```

计数器/定时器工作模式原理和实现

--定时器/计数器0工作模式

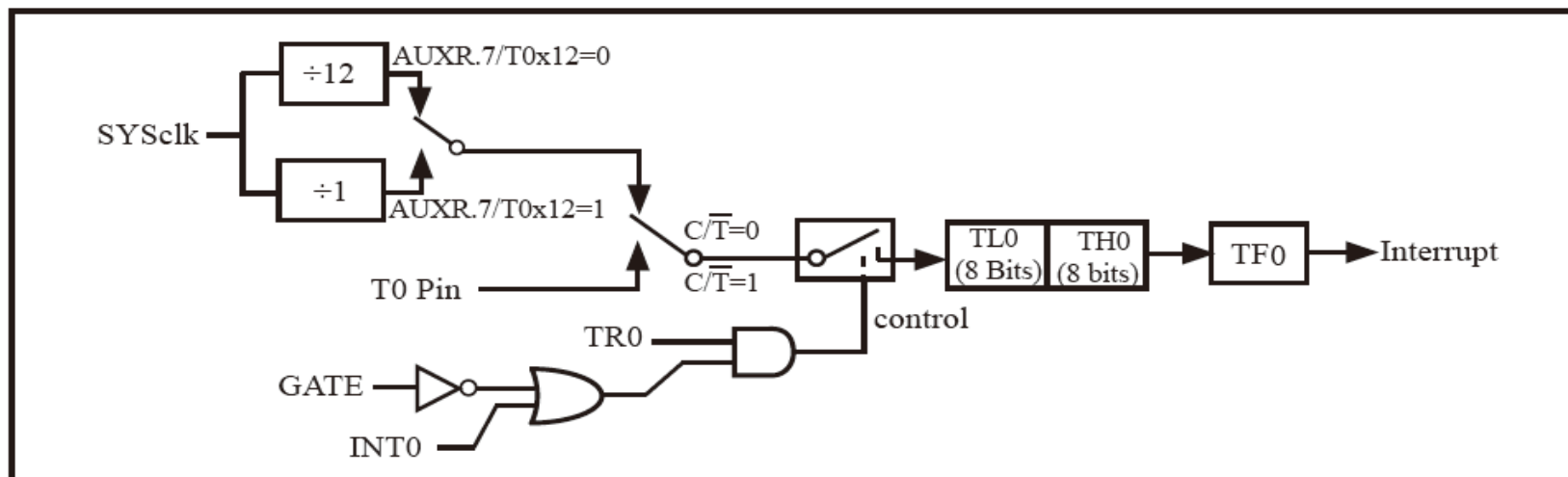
下面对该设计进行验证和分析，步骤如下：

- **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0工作模式

--模式1（16位不可自动重加载模式）

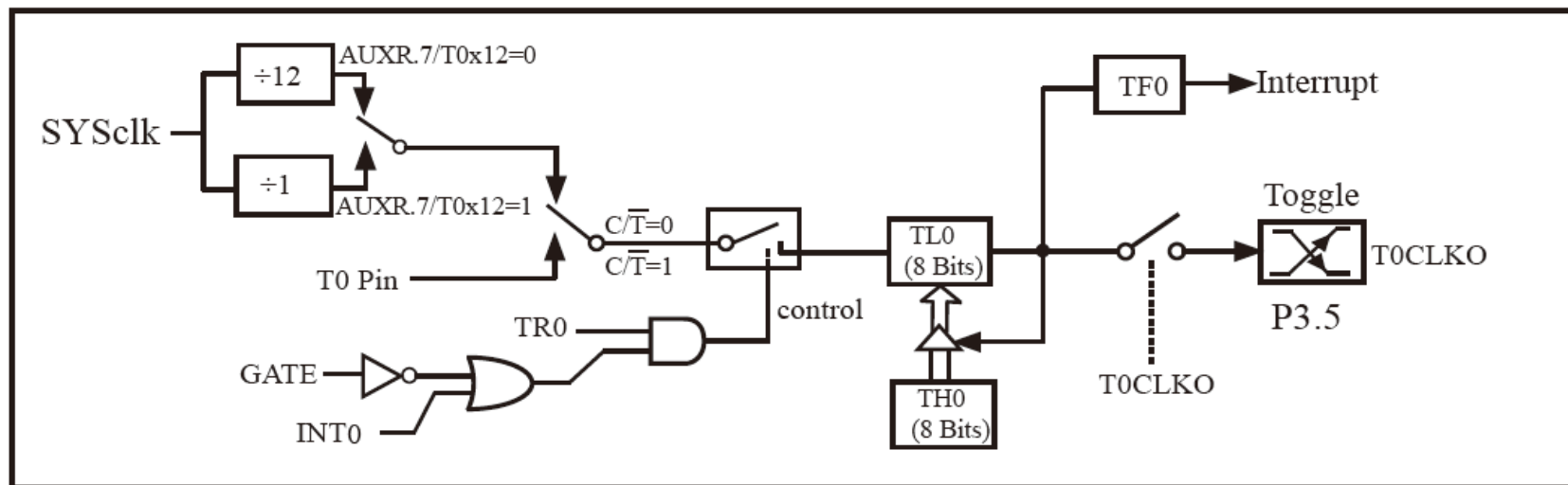
除了不能自动重加载16位计数初值外，以及没有T0CLKO输出外，定时器/计数器0模式1和模式0结构基本相同。



定时器/计数器0工作模式

--模式2（8位自动重加载模式）

除了自动重加载8位计数初值外，定时器/计数器0模式2和模式0结构基本相同。



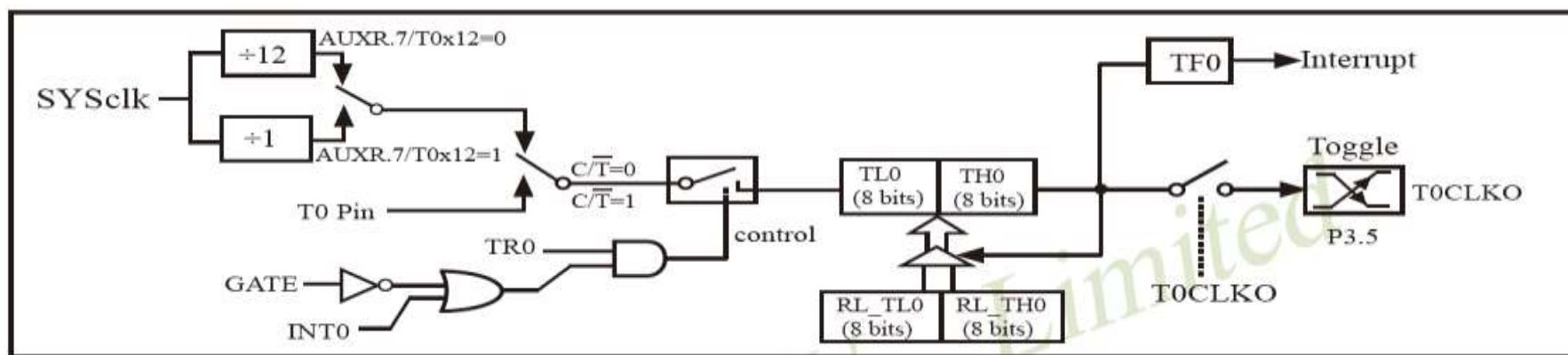
定时器/计数器0工作模式



模式3（不可屏蔽中断16位自动重载）

定时器/计数器0模式3和模式0结构基本相同。

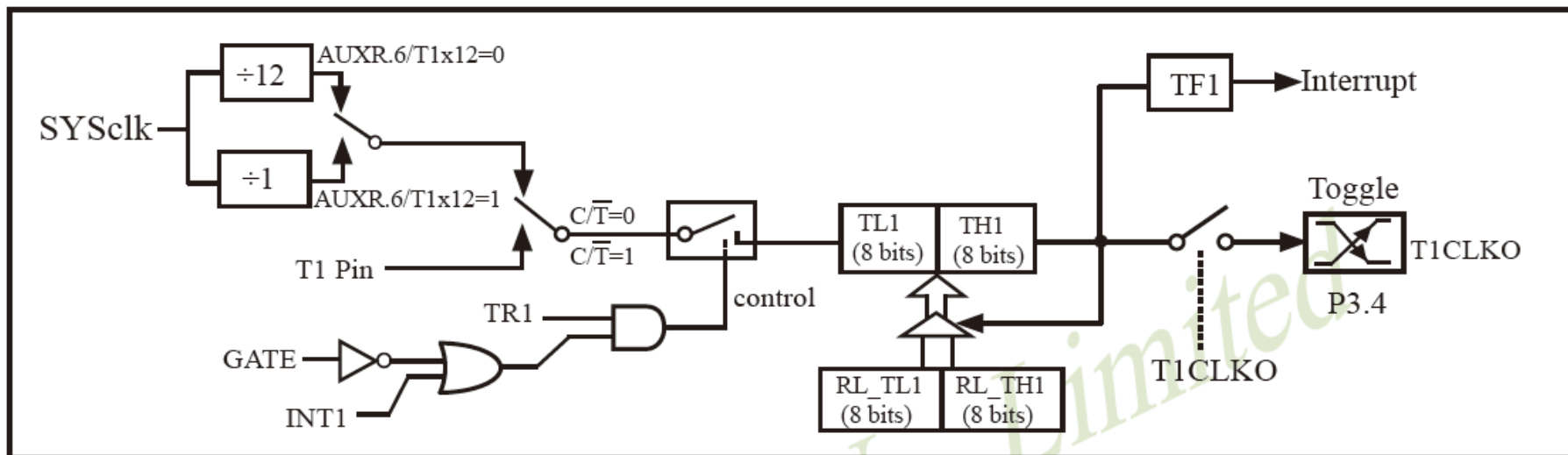
- 不同点在于：当工作在模式3时，只需允许ET0/IE.1（定时器/计数器0中断允许位），而不需要允许EA/IE.7（总中断使能位）就能打开定时器/计数器0的中断。一旦在该模式下的定时器/计数器0中断被打开（ET0=1），那么中断是不可屏蔽的，该中断的优先级也是最高的。



定时器/计数器1工作模式

--模式0（16位自动重加载模式）

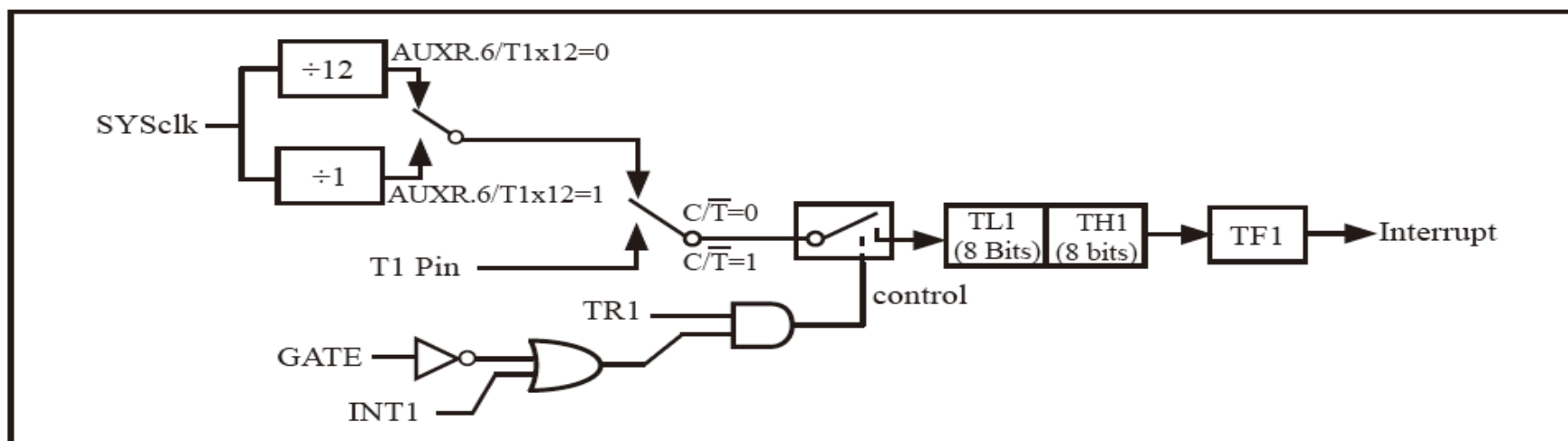
定时器/计数器1工作模式0内部结构



定时器/计数器1工作模式

--模式1（16位不可自动重加载模式）

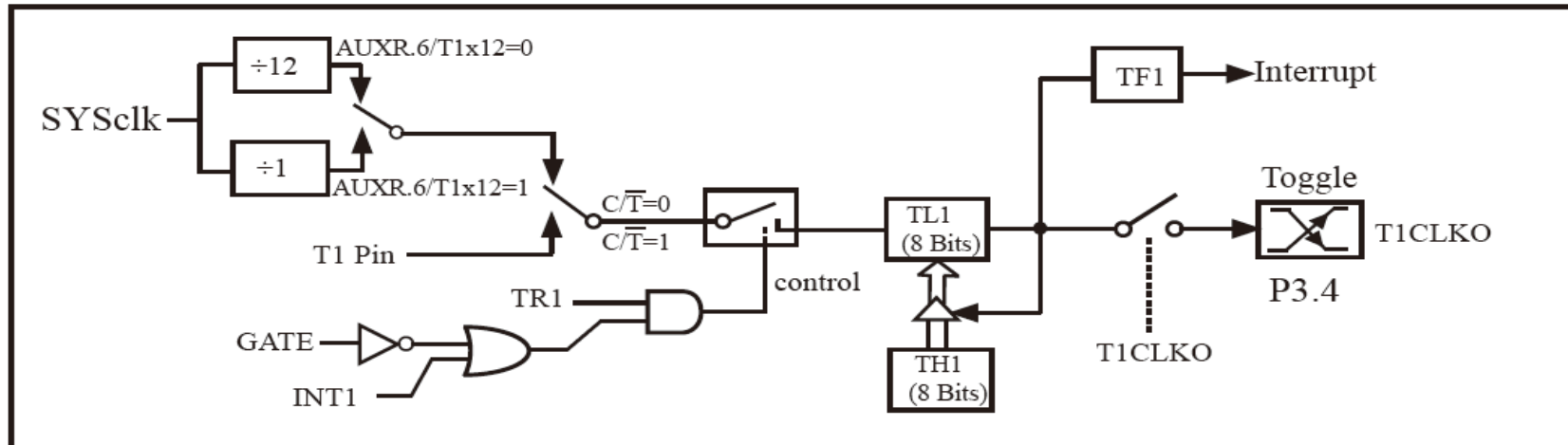
除了不能自动重加载16位计数初值外，以及没有T1CLKO输出外，定时器/计数器1模式1和模式0结构基本相同。



定时器/计数器1工作模式

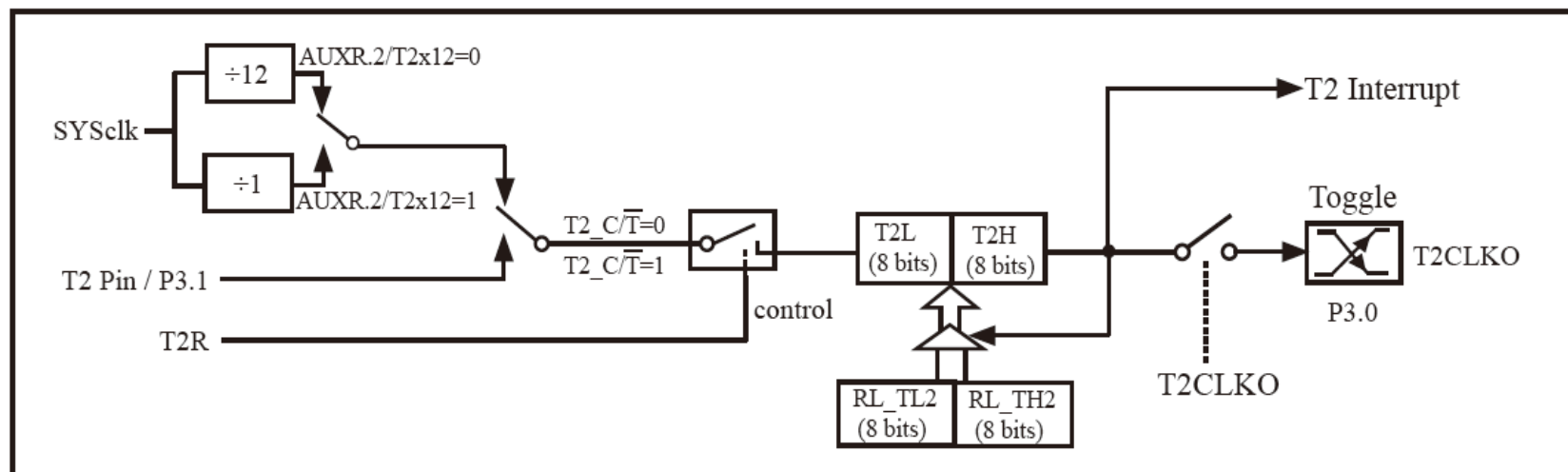
--模式2（8位自动重加载模式）

除了自动重加载8位计数初值外，定时器/计数器1模式2和模式0结构基本相同。



定时器/计数器2工作模式

定时器/计数器2只有16位自动重加载模式



定时器/计数器2工作模式

【例】定时器/计数器2自动加载模式C语言描述的例子

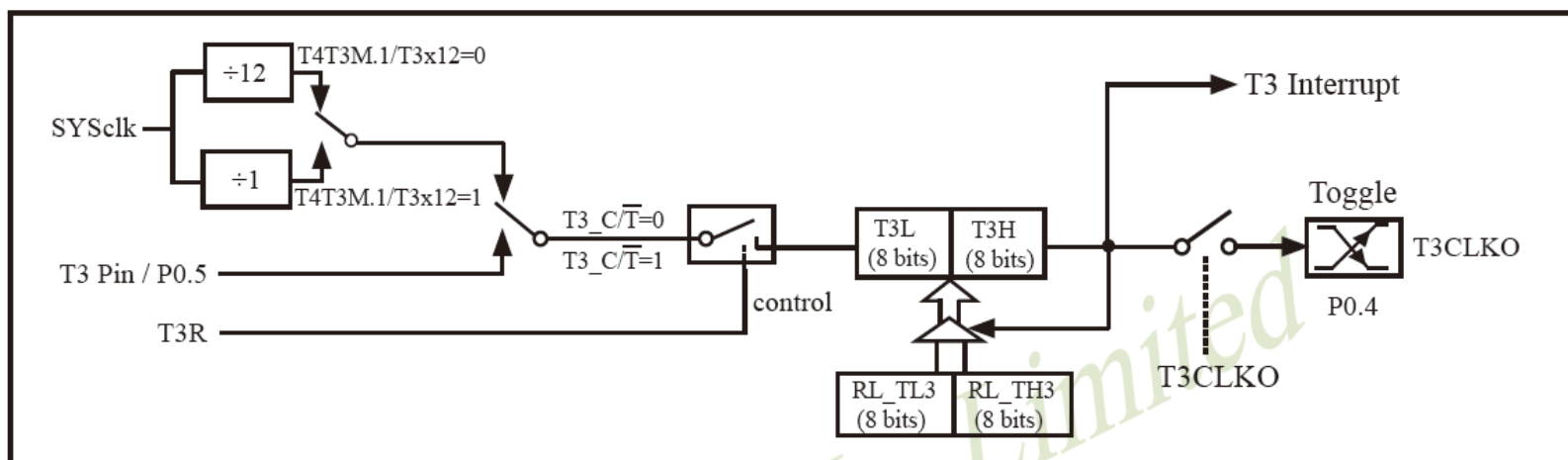
```
#include "reg51.h"
#define TIMS 3036
sfr AUXR =0x8E;           //声明AUXR寄存器的地址0x8E
sfr IE2 =0xAF;            //声明IE2寄存器的地址0xAF
sfr TH2 =0xD6;            //声明TH2寄存器的地址0xD6
sfr TL2 =0xD7;            //声明TL2寄存器的地址0xD7
sfr CLK_DIV=0x97;         //声明CLK_DIV寄存器的地址0x97
void timer_2() interrupt 12 //声明定时器/计数器2中断服务程序
{
    P46=!P46;              //P4.6端口取反
    P47=!P47;              //P4.7端口取反
}
```

定时器/计数器2工作模式

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

定时器/计数器3工作模式

定时器/计数器3只有16位自动重加载模式



定时器/计数器3工作模式

【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"

#define TMS 3036

sfr CLK_DIV=0x97;
sfr IE2 =0xAF;
sfr TH3 =0xD4;
sfr TL3 =0xD5;
sfr T4T3M =0xD1;
void timer_3() interrupt 19
{
    P46=!P46;
    P47=!P47;
}
```

//定义TMS的值

//声明CLK_DIV寄存器的地址

//声明IE2寄存器的地址

//声明TH3寄存器的地址

//声明TL3寄存器的地址

//声明T4T3M寄存器的地址

//声明定时器/计数器3的中断服务程序

//P4.6端口取反

//P4.7端口取反

定时器/计数器3工作模式

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

计数器/定时器工作模式原理和实现

--定时器/计数器4工作模式

定时器/计数器4只有16位自动重加载模式

