



单片机原理与应用

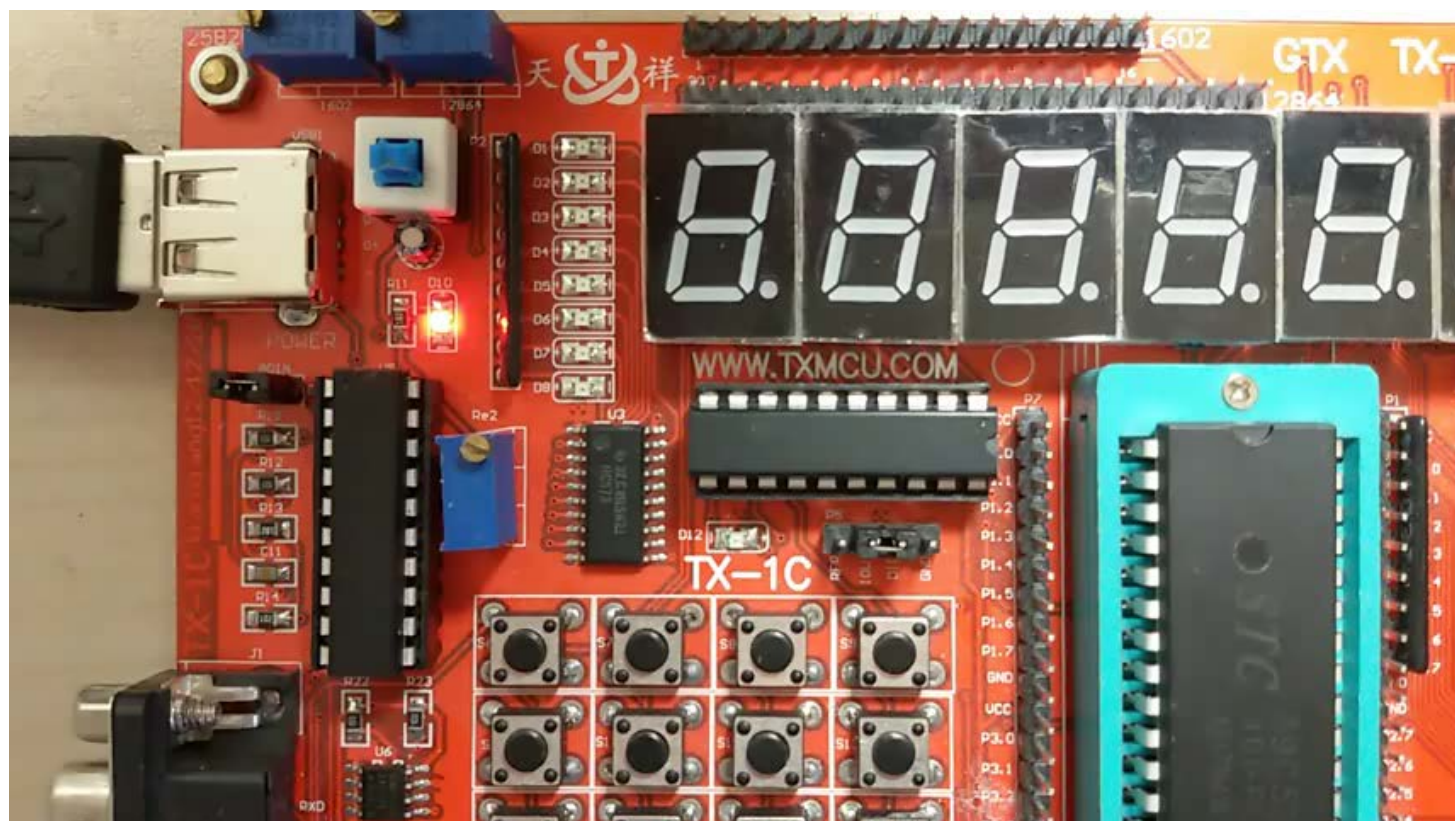
北京航空航天大学
电子信息工程学院
张玉玺

zhangyuxi@buaa.edu.cn

第七章 定时器

■ 本章实现功能

- 单片机定时器控制LED以1s为间隔闪烁



第七章 定时器

■ 如何实现功能

- 定时器工作原理
- 相关寄存器控制
- 工作方式1
 - 流程图
 - 程序代码
 - 烧写
- 其他工作方式对比



第七章 定时器

■ 定时器工作原理

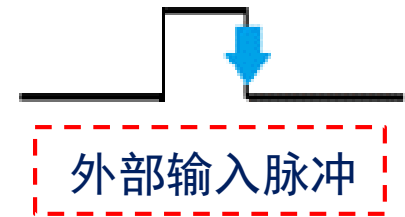
- 具有**定时**和**计数**两种功能
- 可用**T0**、**T1**两个定时器、4种工作方式
- 通过**中断**方式触发，执行中断服务程序

中断源	入口地址	中断号
外部中断0	0003H	0
定时器0溢出	000BH	1
外部中断1	0013H	2
定时器1溢出	001BH	3
串行口中断	0023H	4

第七章 定时器

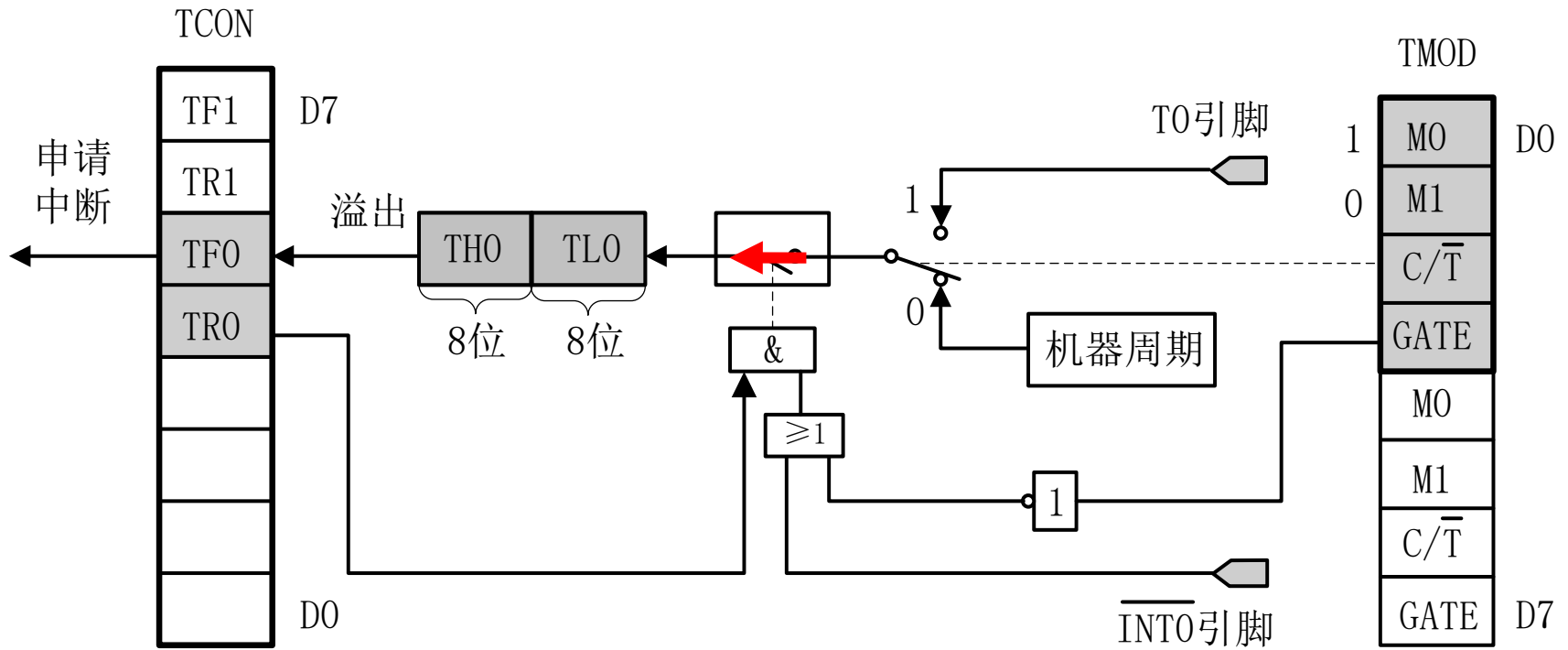
■ 定时器工作原理

- 计数器：对**外部**输入脉冲进行计数
- 定时器：对单片机**内部**机器周期脉冲进行计数，通过计算脉冲个数实现定时
- 两者均对脉冲计数，本质相同，统称**定时器**



第七章 定时器

■ 定时模式

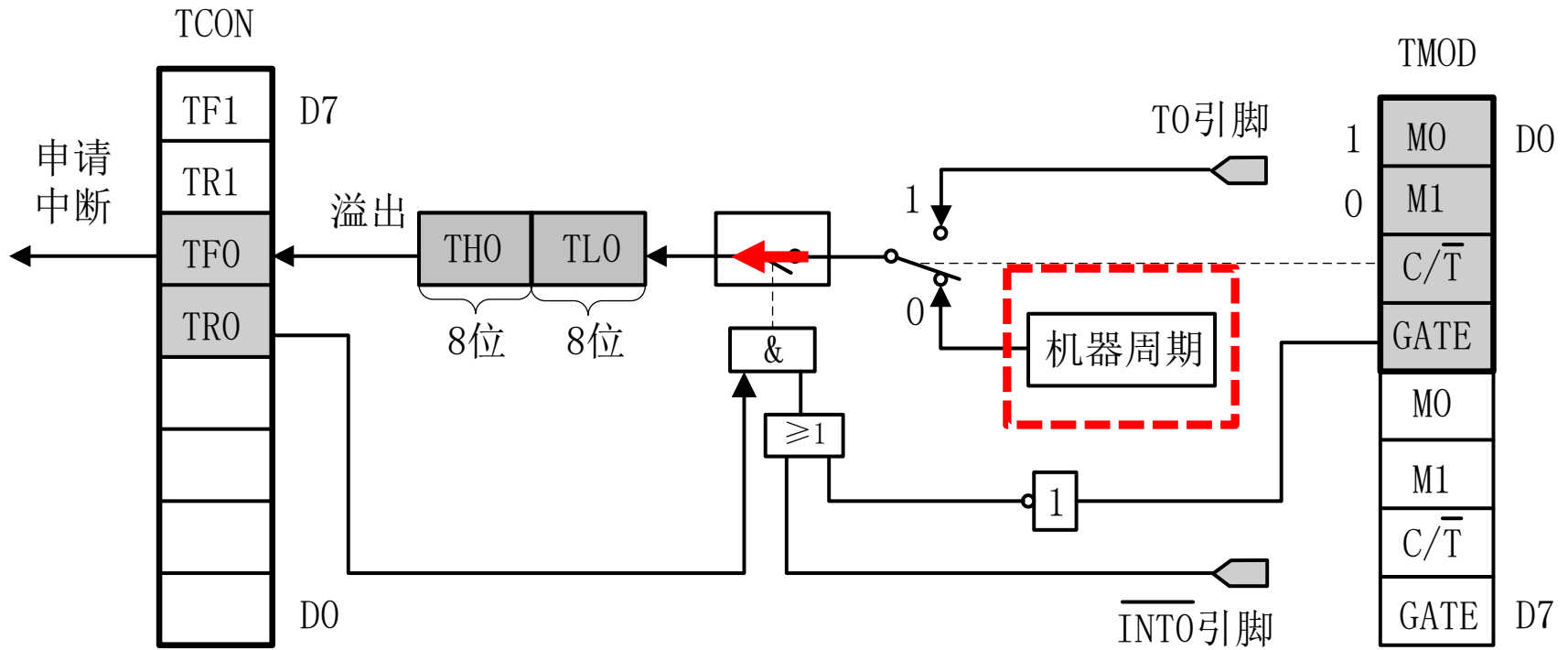


1、启动定时器，开始计时

第七章 定时器

■ 定时模式

机器周期

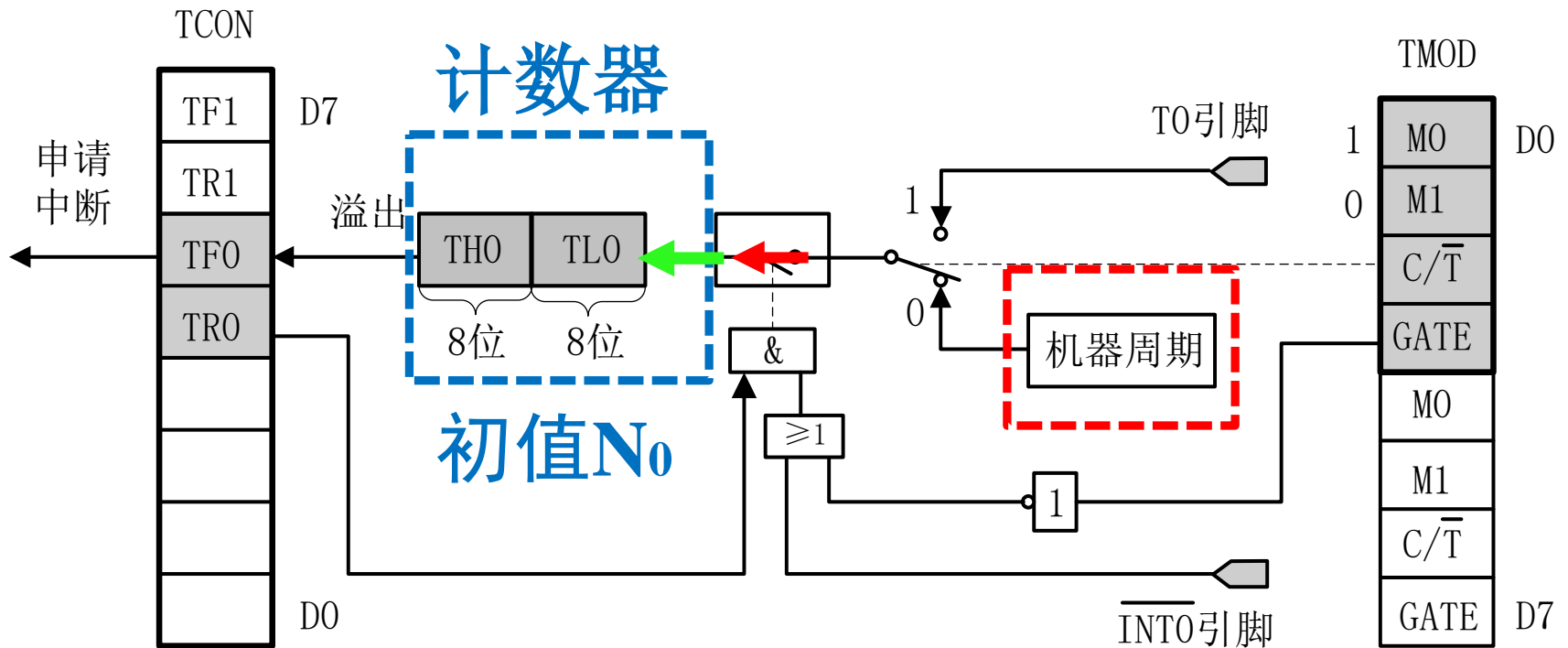


2. 定时模式：接收机器周期脉冲（由系统时钟脉冲经12分频得到）

第七章 定时器

定时模式

机器周期

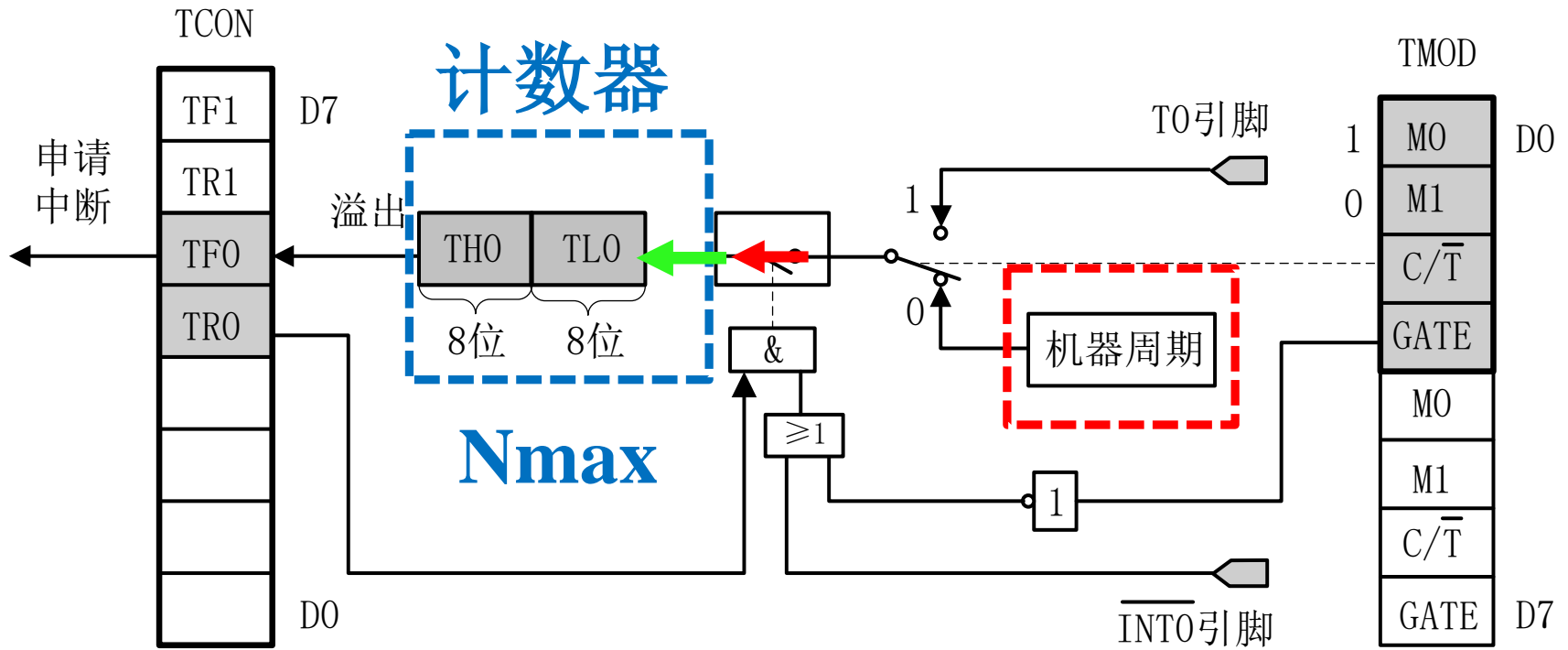


3. 每次脉冲到来，计数器加1

第七章 定时器

■ 定时模式

机器周期

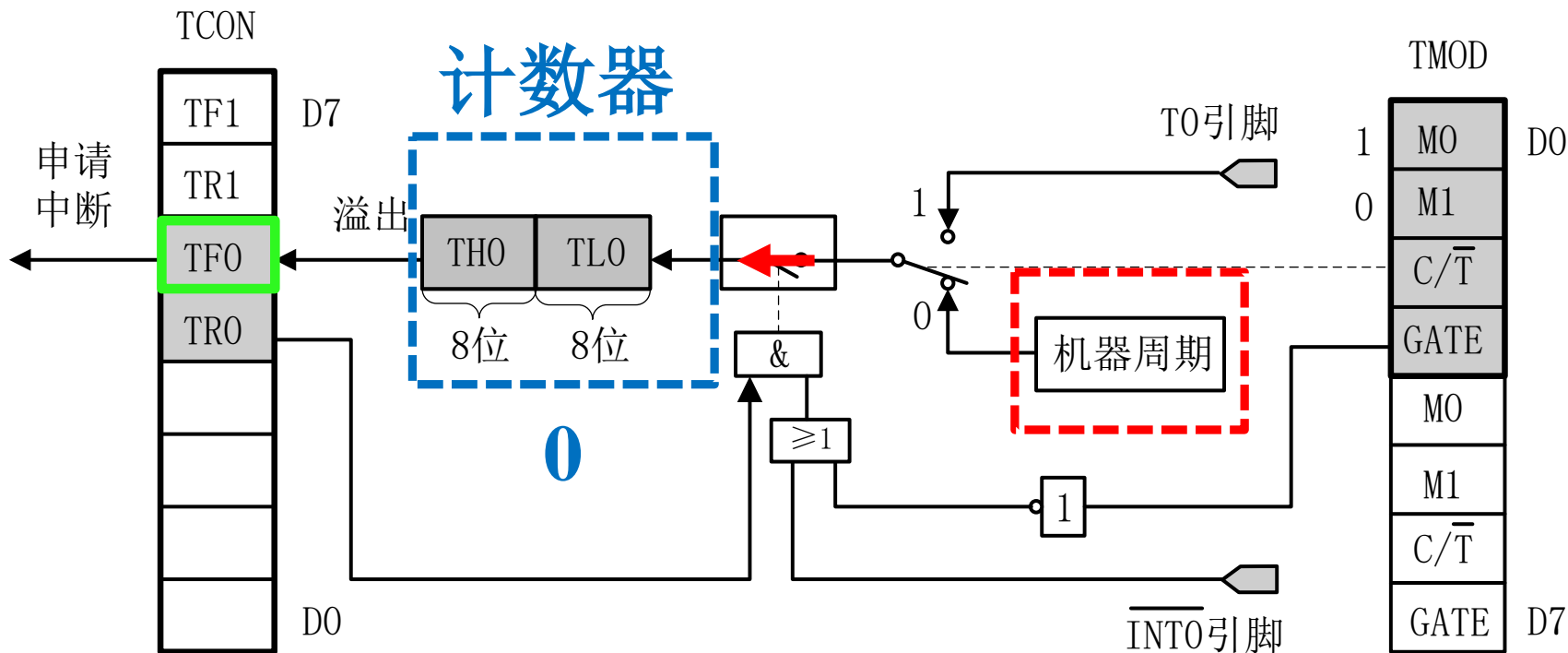
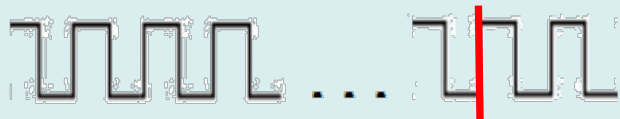


4. 如此循环，直到计数器加到最大值

第七章 定时器

定时模式

机器周期

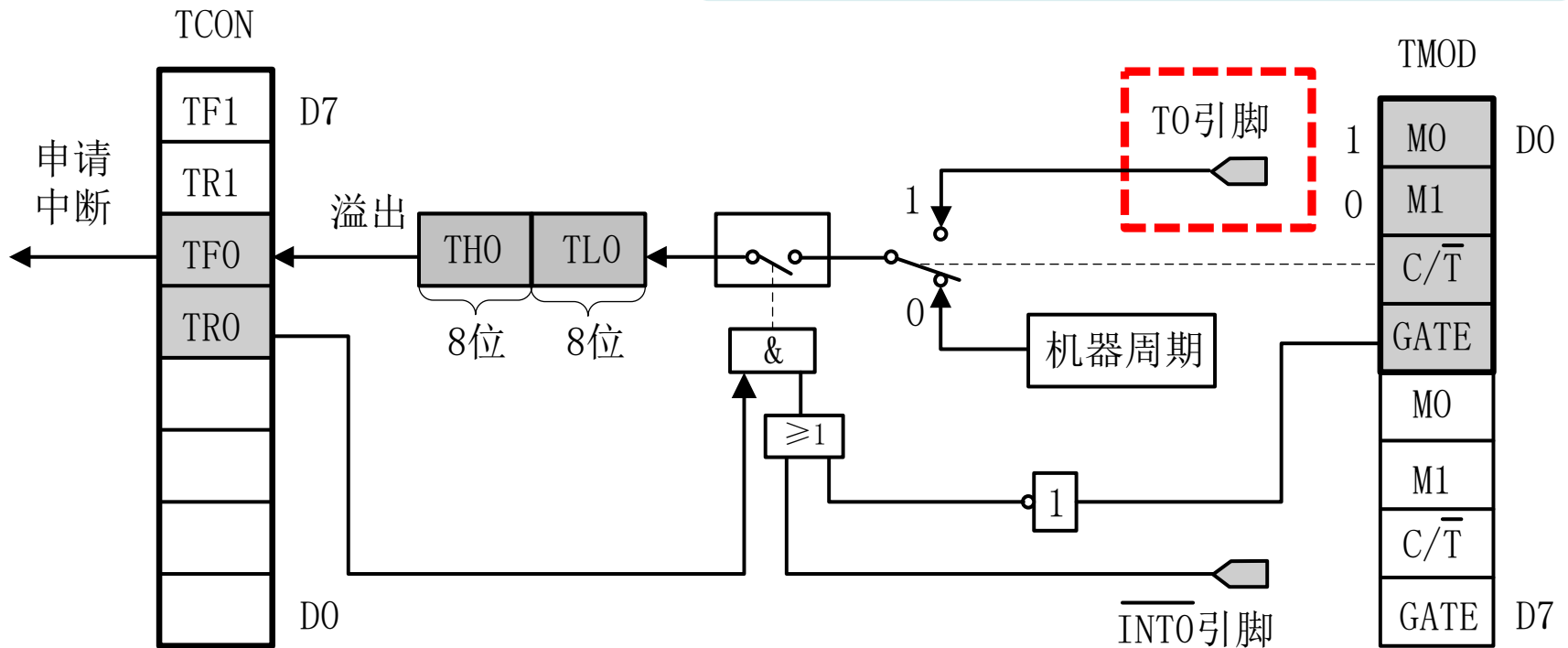
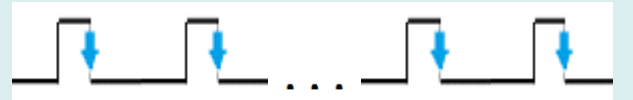


5. 脉冲再次到来，计数器溢出，数值清零，溢出标志位置1，触发中断

第七章 定时器

■ 计数模式

外部输入



与定时模式区别：计数脉冲来自外部引脚，检测到引脚信号下降沿，计数器加1

第七章 定时器

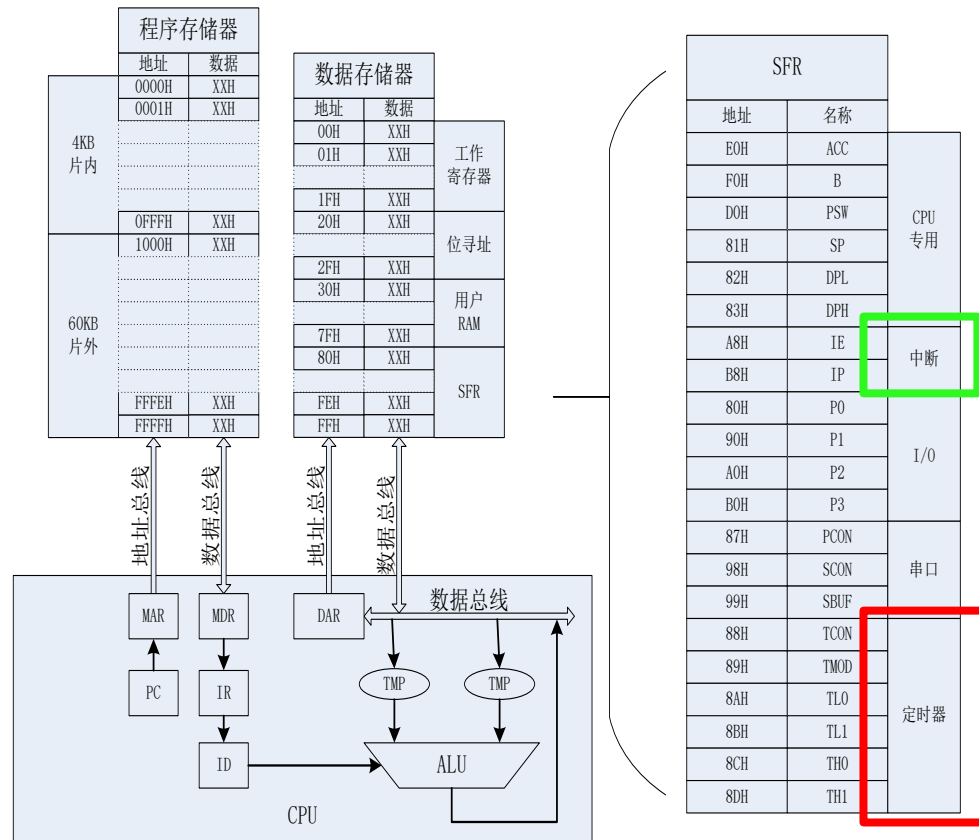
相关寄存器控制

定时器寄存器

- **TMOD**: 89H
- **TCON**: 88H
- **TLO**: 8AH
- **TL1**: 8BH
- **TH0**: 8CH
- **TH1**: 8DH

中断寄存器

- **IE**: A8H
- **IP**: B8H

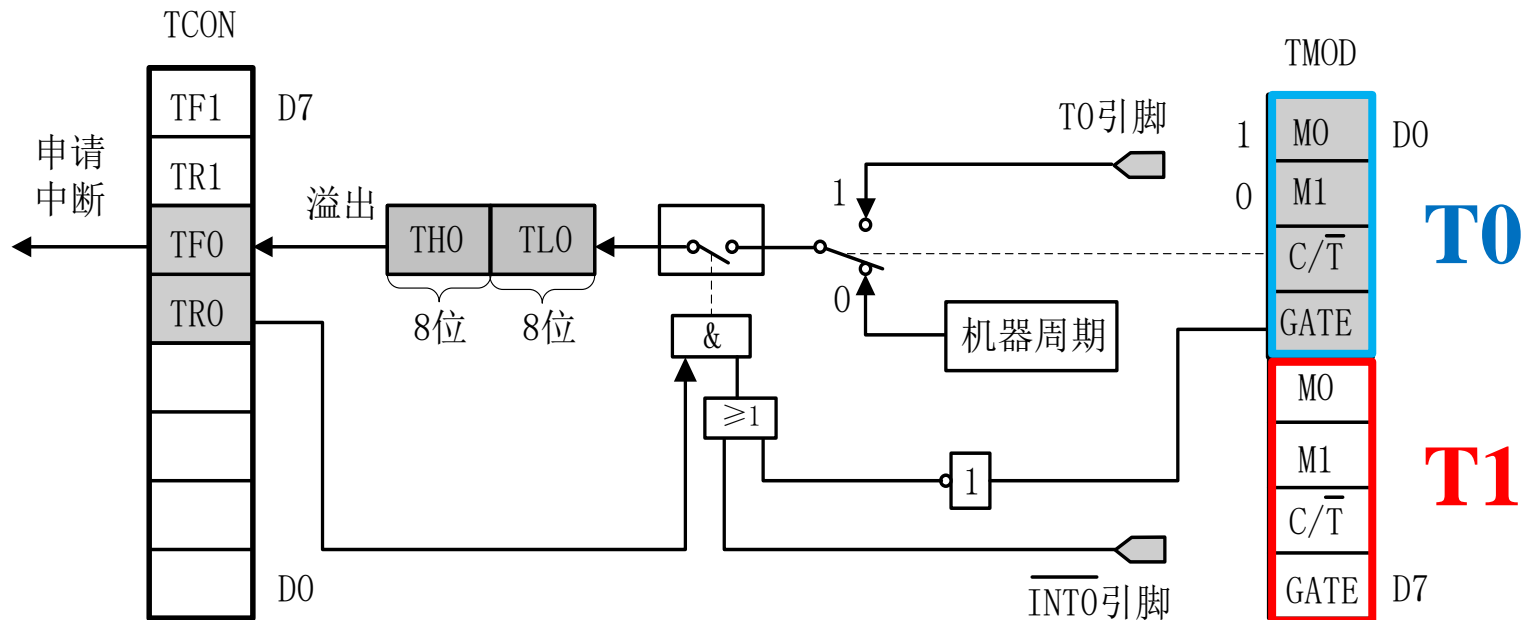


第七章 定时器

■ 定时器寄存器

➤ TMOD工作方式寄存器89H

- 设置定时/计数器的工作方式
- 低四位用于T0，高四位用于T1

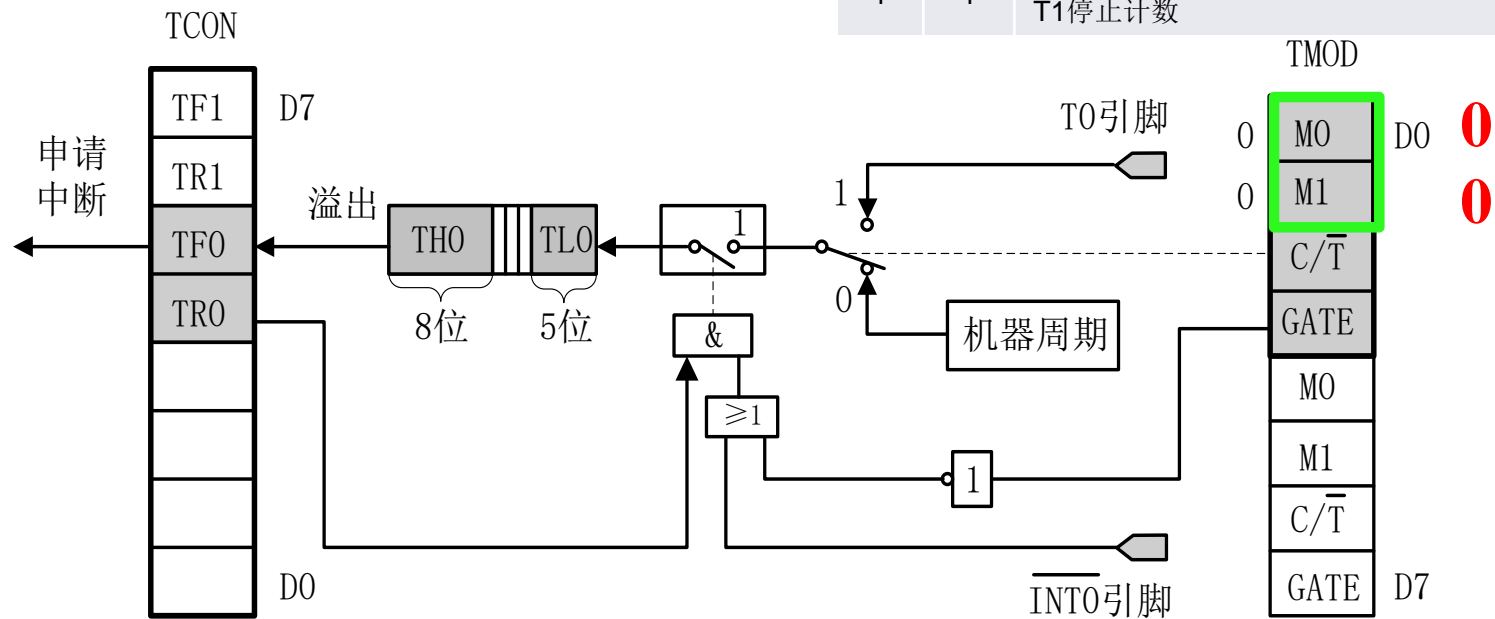


第七章 定时器

■ TMOD工作方式寄存器

➤ 工作方式设定位M1 M0

M1	M0	工作方式
0	0	方式0, 13位定时/计数器
0	1	方式1, 16位定时/计数器
1	0	方式2, 8位初值自动重装的8位定时/计数器
1	1	方式3, 仅适用于T0, 分成两个8位计数器, T1停止计数

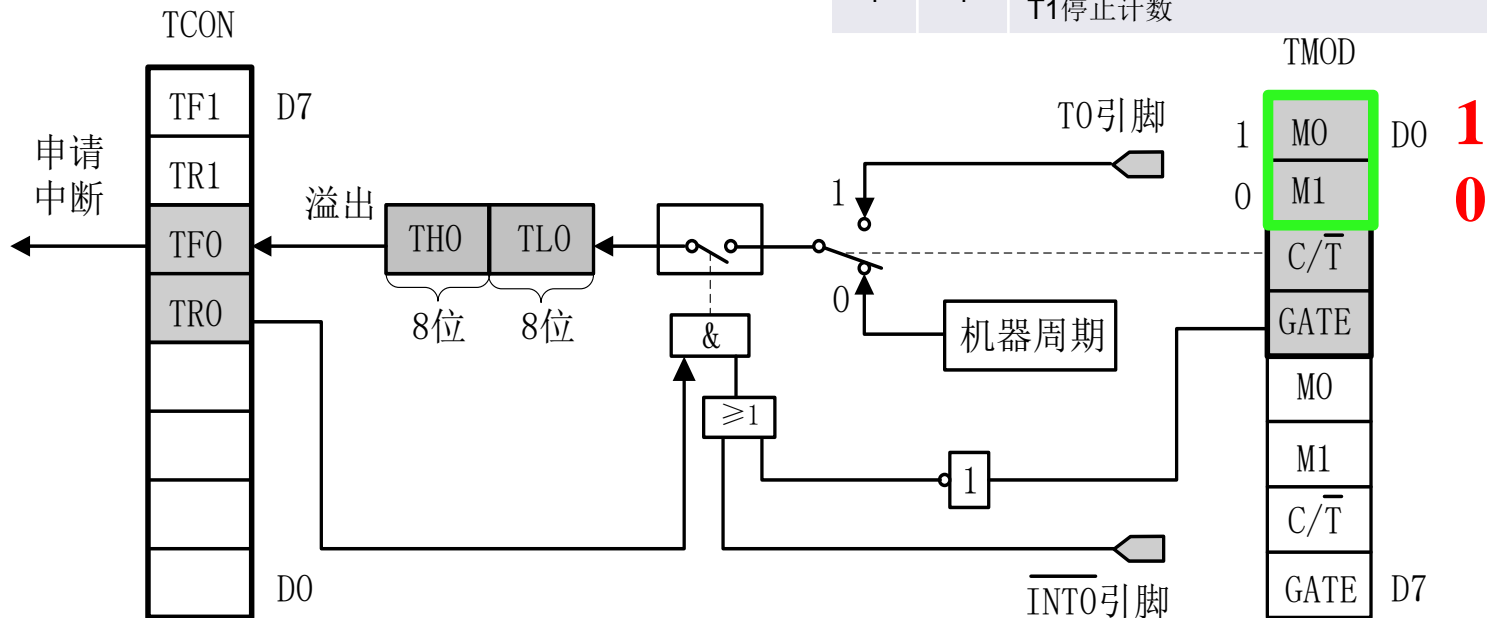


第七章 定时器

■ TMOD工作方式寄存器

➤ 工作方式设定位M1 M0

M1	M0	工作方式
0	0	方式0, 13位定时/计数器
0	1	方式1, 16位定时/计数器
1	0	方式2, 8位初值自动重装的8位定时/计数器
1	1	方式3, 仅适用于T0, 分成两个8位计数器, T1停止计数

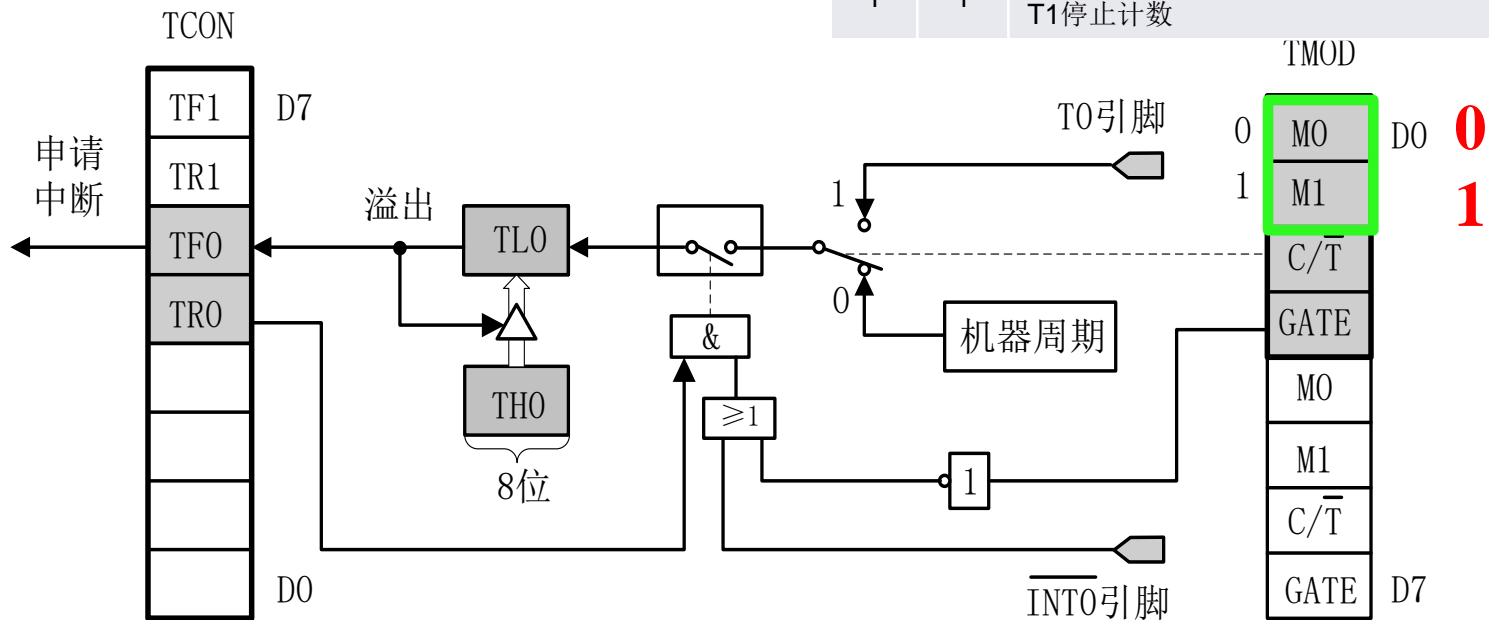


第七章 定时器

■ TMOD工作方式寄存器

➤ 工作方式设定位M1 M0

M1	M0	工作方式
0	0	方式0, 13位定时/计数器
0	1	方式1, 16位定时/计数器
1	0	方式2, 8位初值自动重装的8位定时/计数器
1	1	方式3, 仅适用于T0, 分成两个8位计数器, T1停止计数

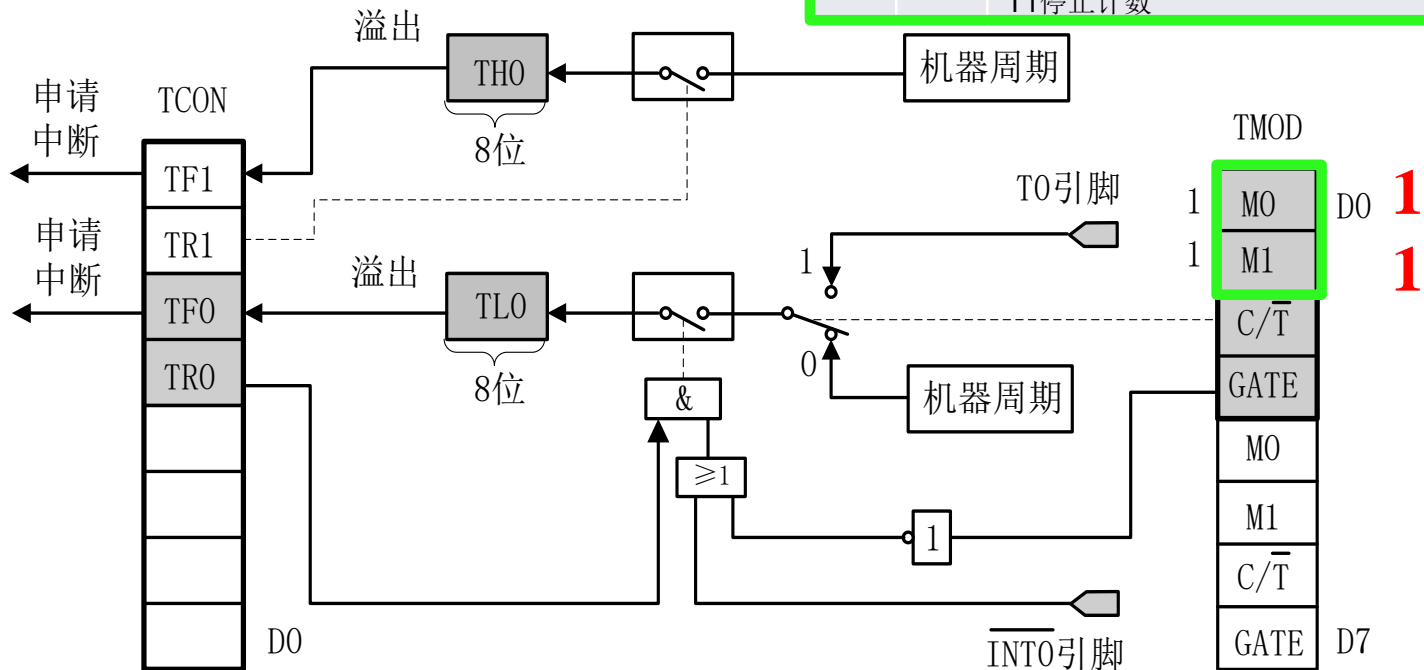


第七章 定时器

■ TMOD工作方式寄存器

➤ 工作方式设定位M1 M0

M1	M0	工作方式
0	0	方式0, 13位定时/计数器
0	1	方式1, 16位定时/计数器
1	0	方式2, 8位初值自动重装的8位定时/计数器
1	1	方式3, 仅适用于T0, 分成两个8位计数器, T1停止计数

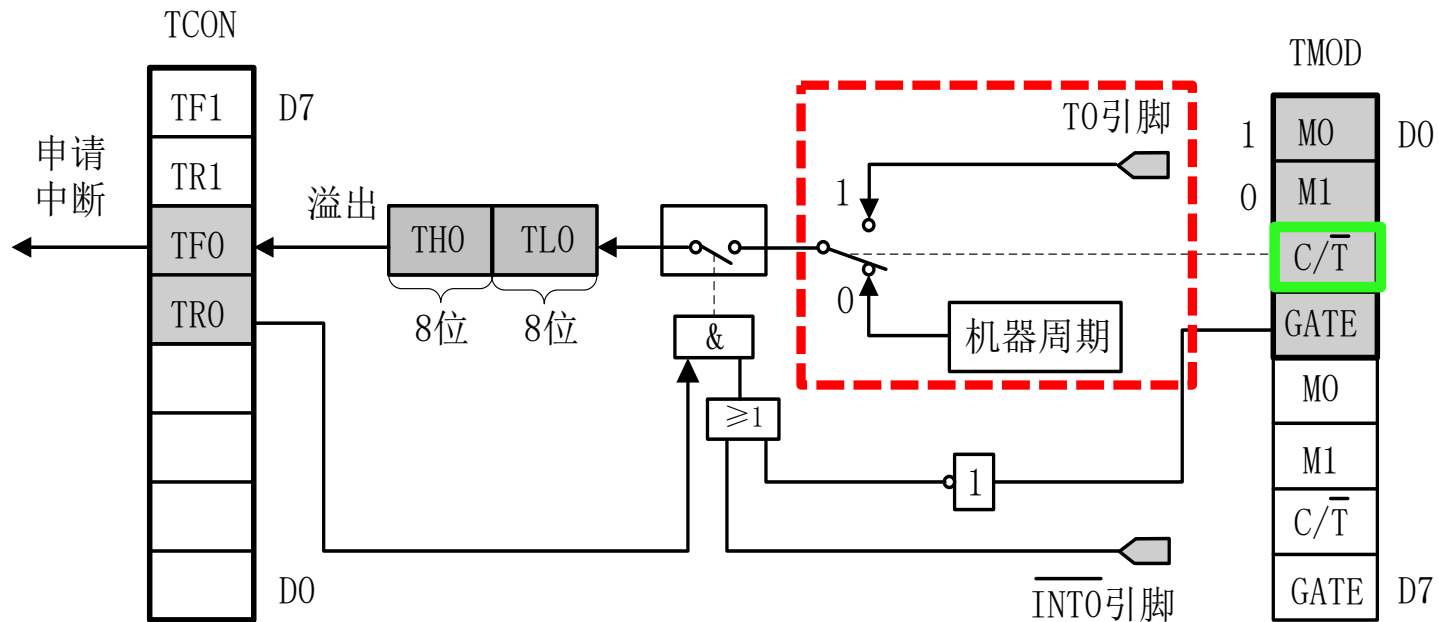


第七章 定时器

■ TMOD工作方式寄存器

➤ 定时/计数位C/T

- 定时模式 C/T=0； 计数模式 C/T=1

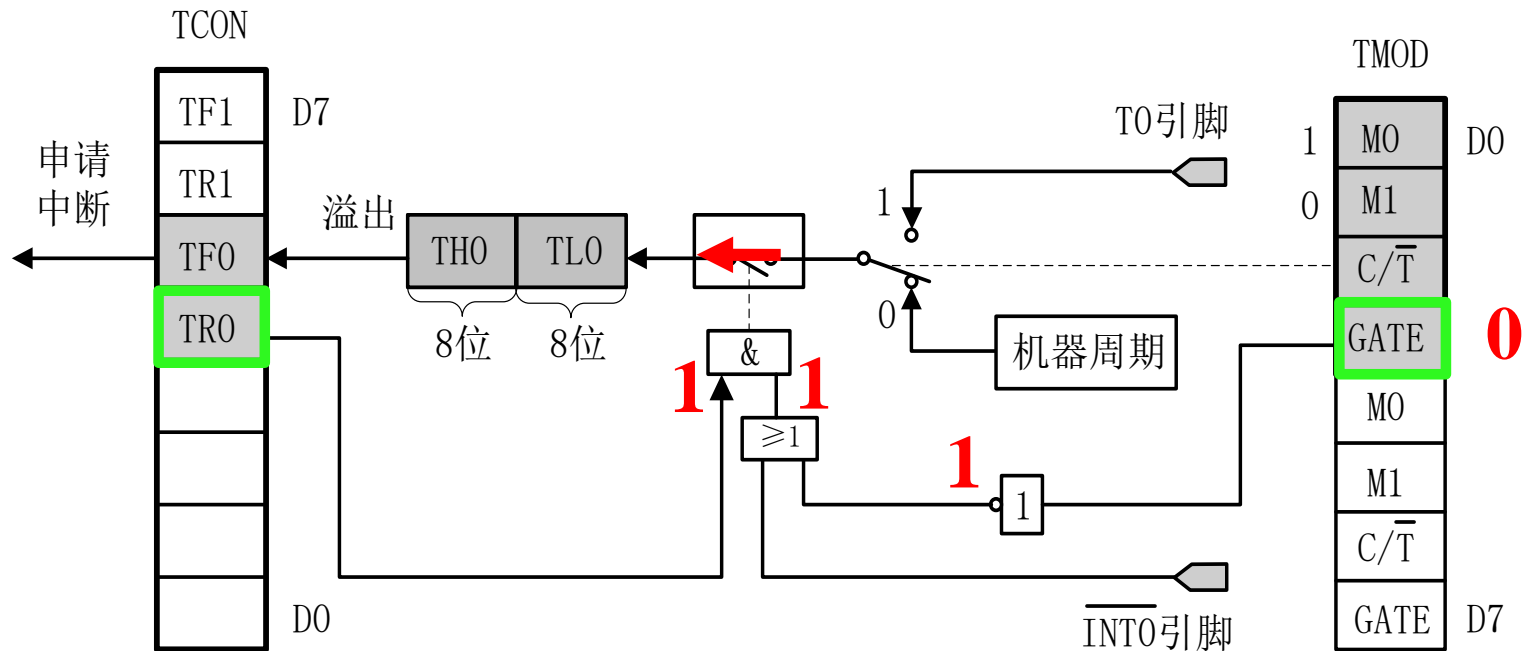


第七章 定时器

■ TMOD工作方式寄存器

➤ 门控位GATE

- GATE=0时，定时器启停仅受TCON寄存器的TR0/TR1位控制

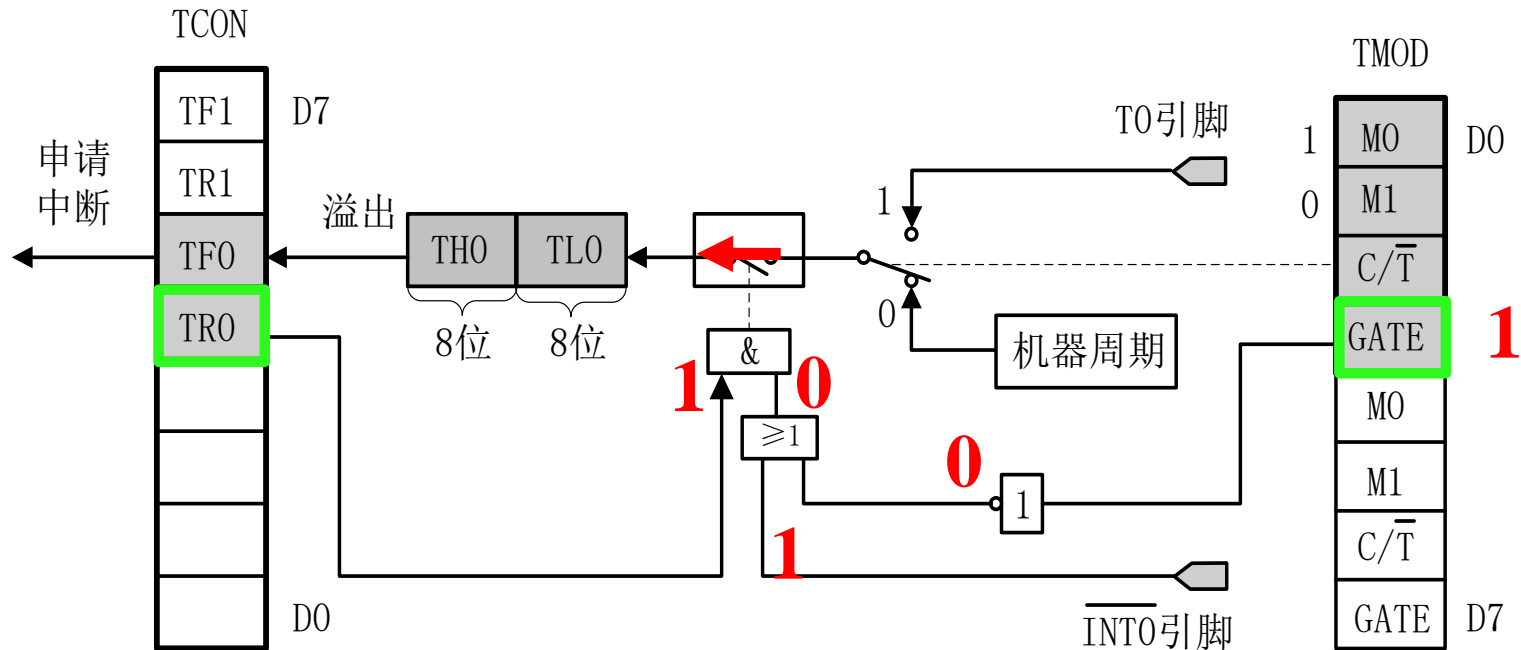


第七章 定时器

■ TMOD工作方式寄存器

➤ 门控位GATE

- GATE=1时，定时器启停受TCON寄存器的TR0/TR1位及外部中断引脚的电平状态控制

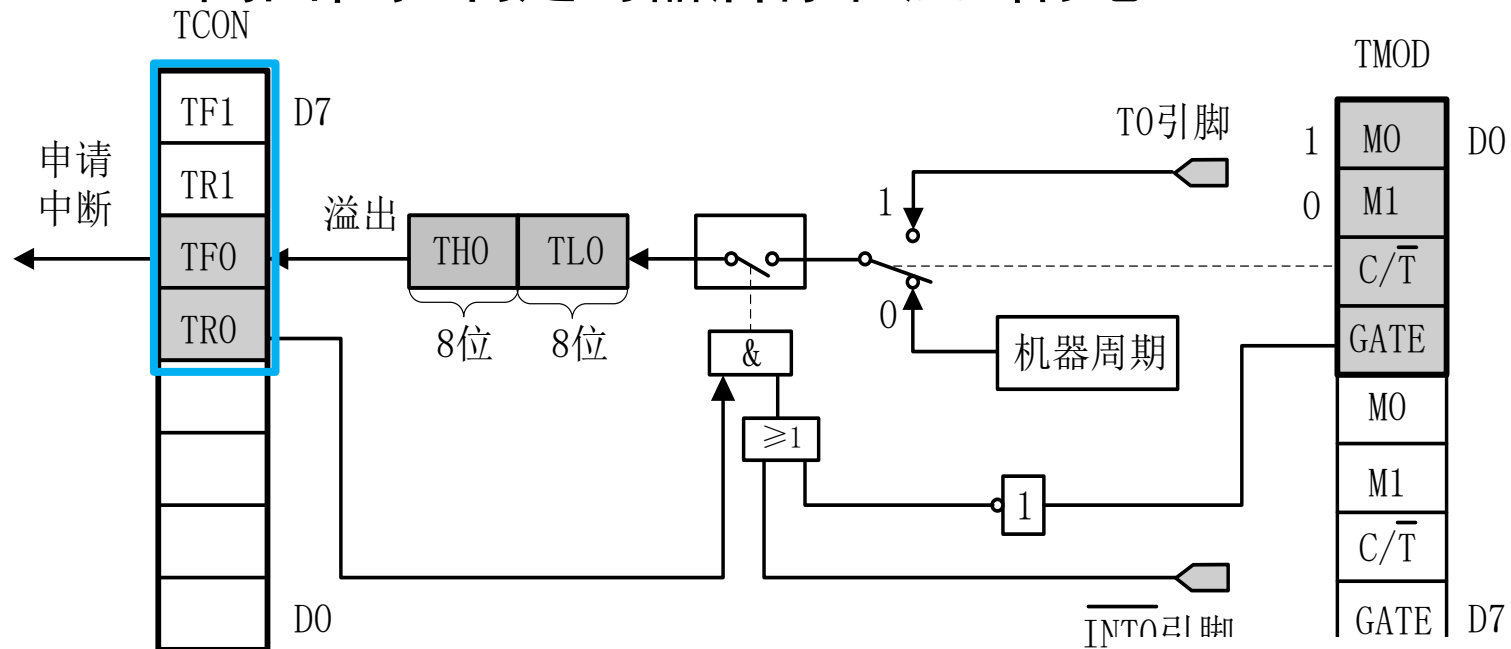


第七章 定时器

■ 定时器寄存器

➤ TCON定时/计数器控制寄存器88H

- 低四位控制外部中断（见第六章）
- 高四位控制定时器启停和溢出标志

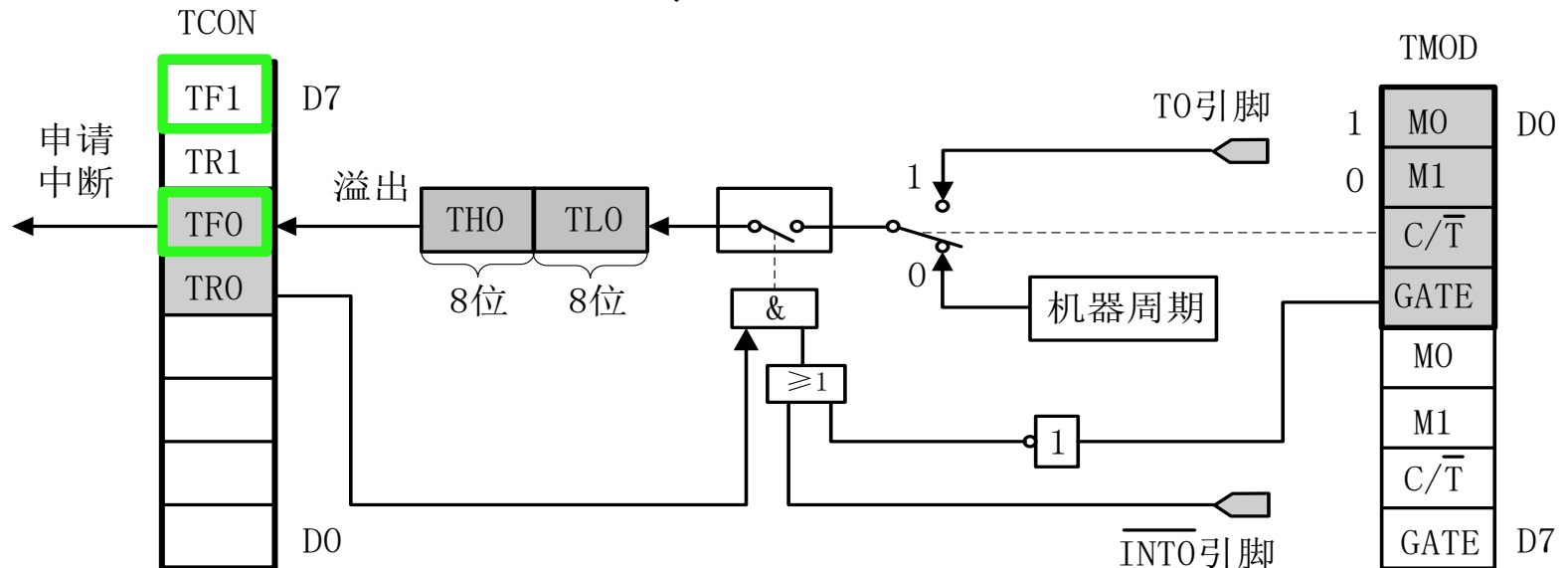


第七章 定时器

■ TCON定时/计数器控制寄存器

➤ 中断请求标志位TF0/TF1

- T0/T1溢出时，硬件自动置TF0/TF1为1
- 若中断方式实现，硬件自动清0
- 若查询方式实现，需要软件手动清0

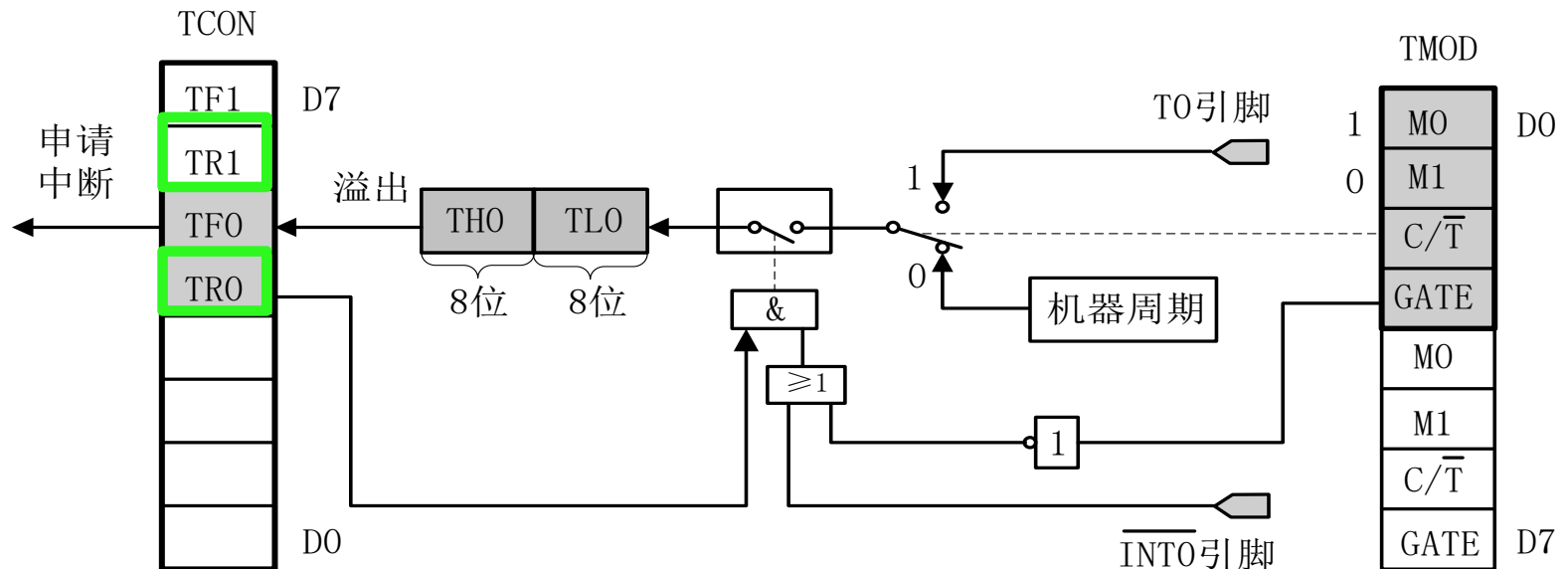


第七章 定时器

■ TCON定时/计数器控制寄存器

➤ 运行控制位TR0/TR1

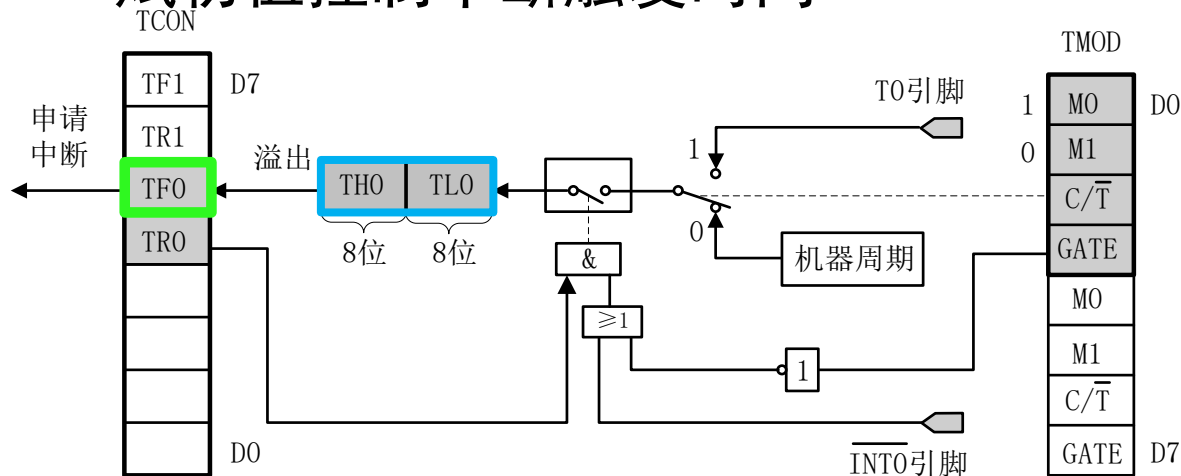
- TR0/TR1置1时，T0/T1开始定时/计数
- TR0/TR1置0时，T0/T1停止定时/计数



第七章 定时器

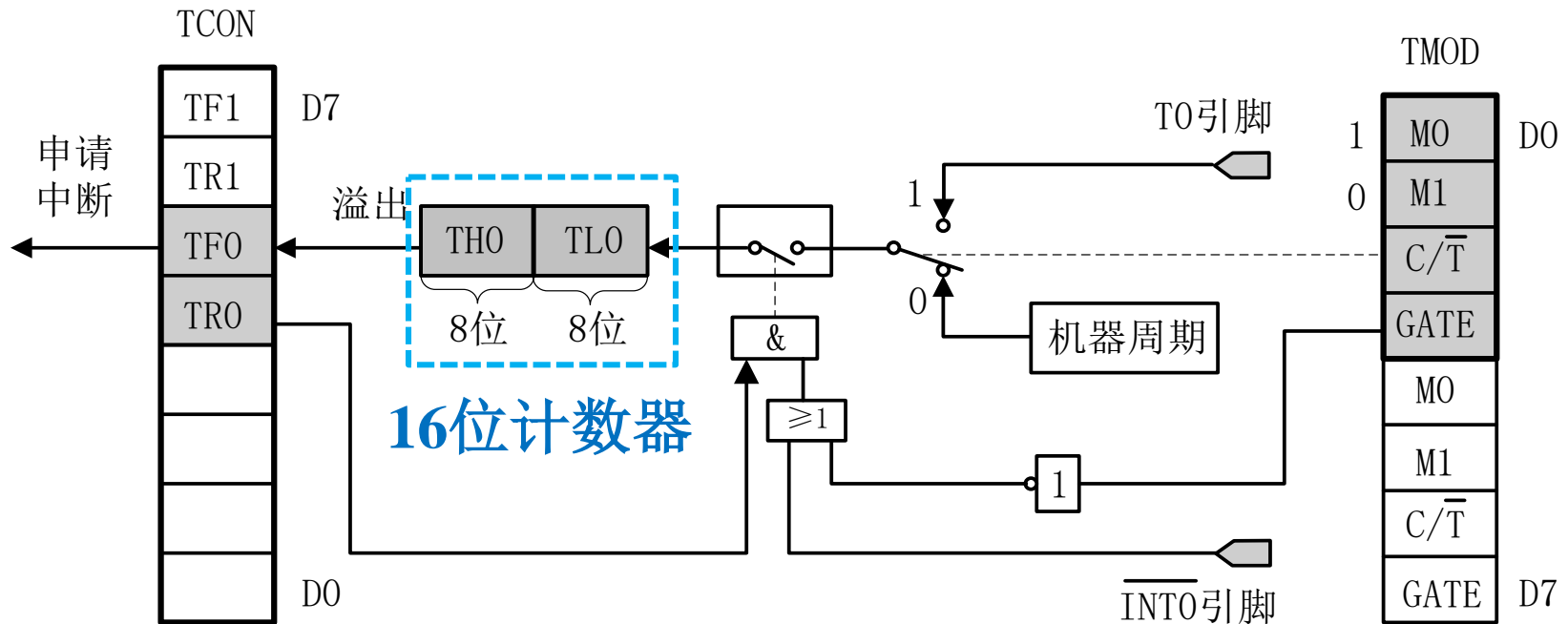
■ 定时器寄存器

- TH0/TL0：定时器0计数器值高/低8位
- TH1/TL1：定时器1计数器值高/低8位
 - 每次计数加1，TLX溢出向THX进位
 - THX溢出，TCON中TFX标志置位，触发中断
 - 赋初值控制中断触发时间



第七章 定时器

THX/TLX初值计算（以方式1：16位计数器为例）

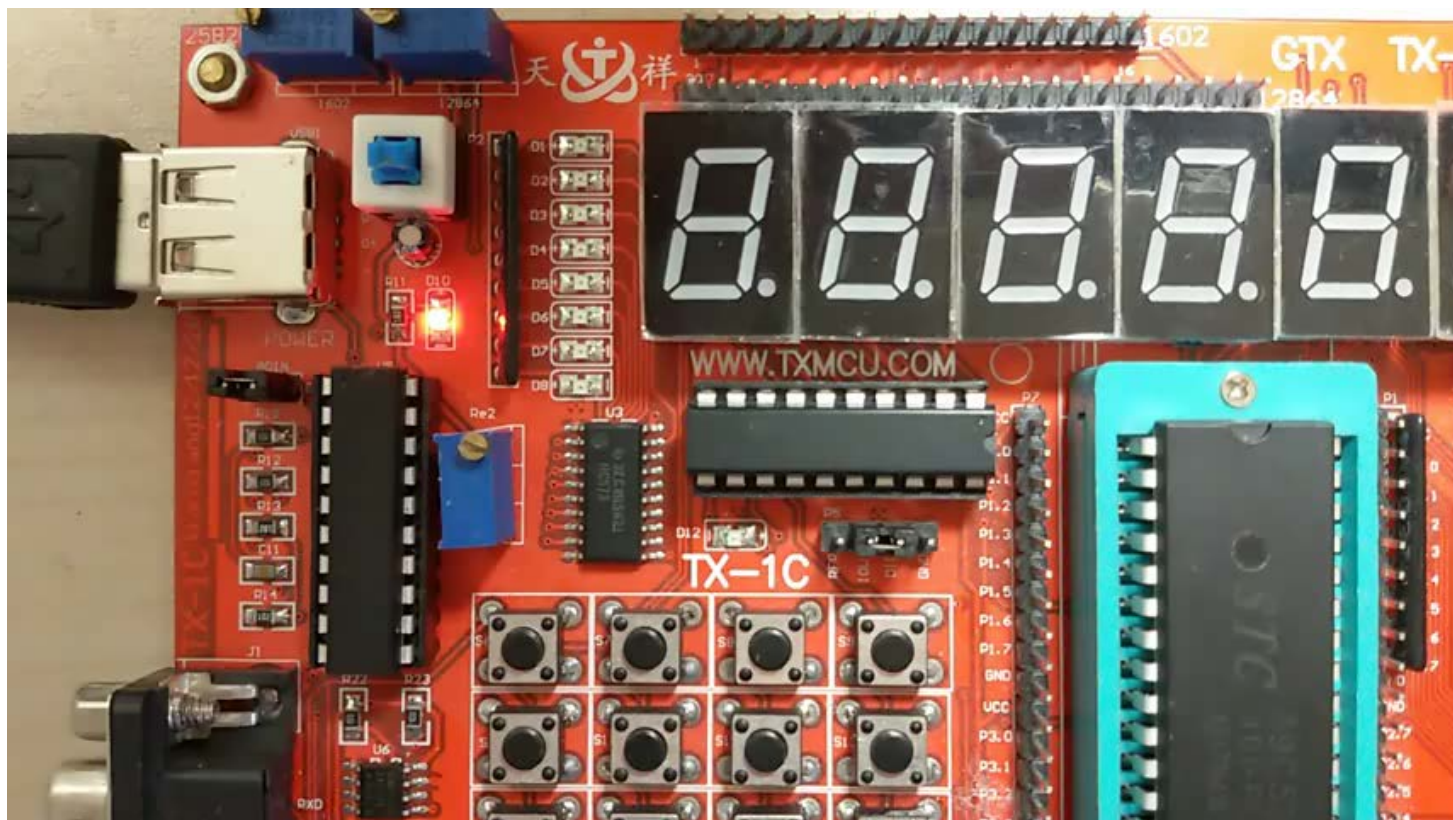


欲使计数器计数 N 次后溢出，
计数初值应为： $2^{16}-N$

第七章 定时器

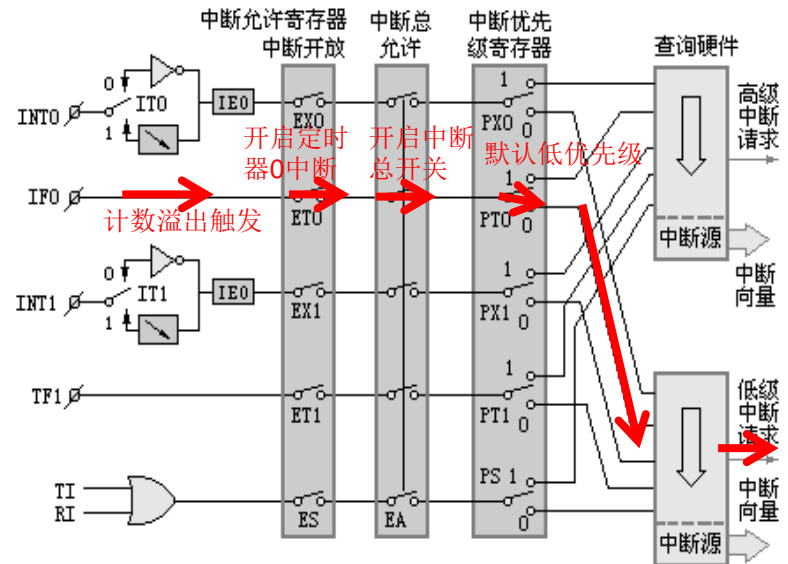
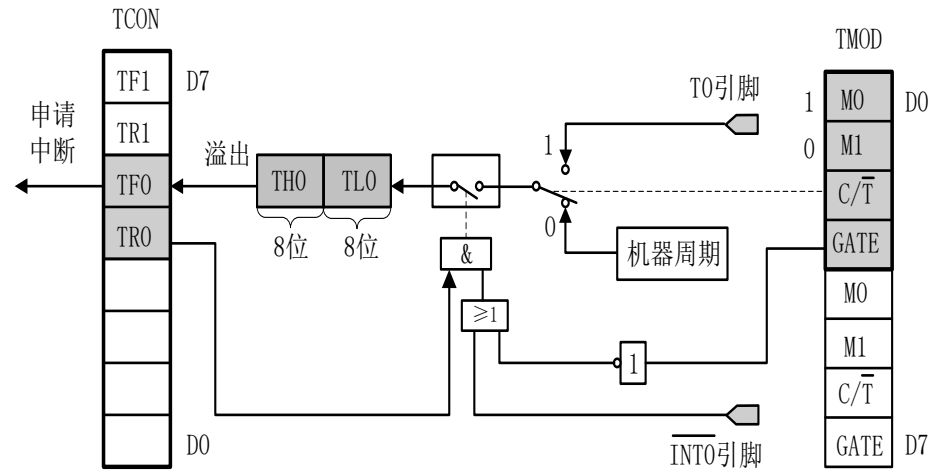
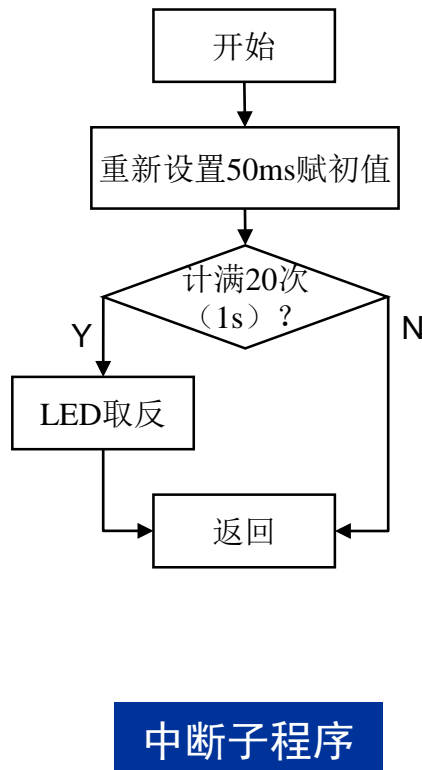
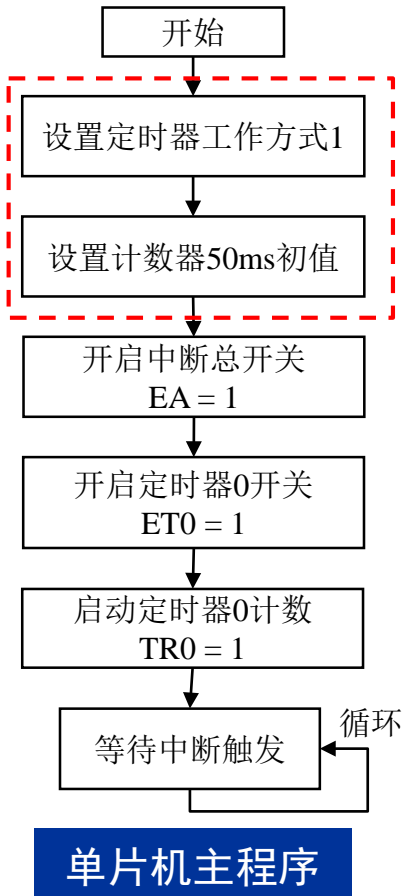
■ 本章实现功能

- 单片机定时器控制LED P1.0以1s为间隔闪烁

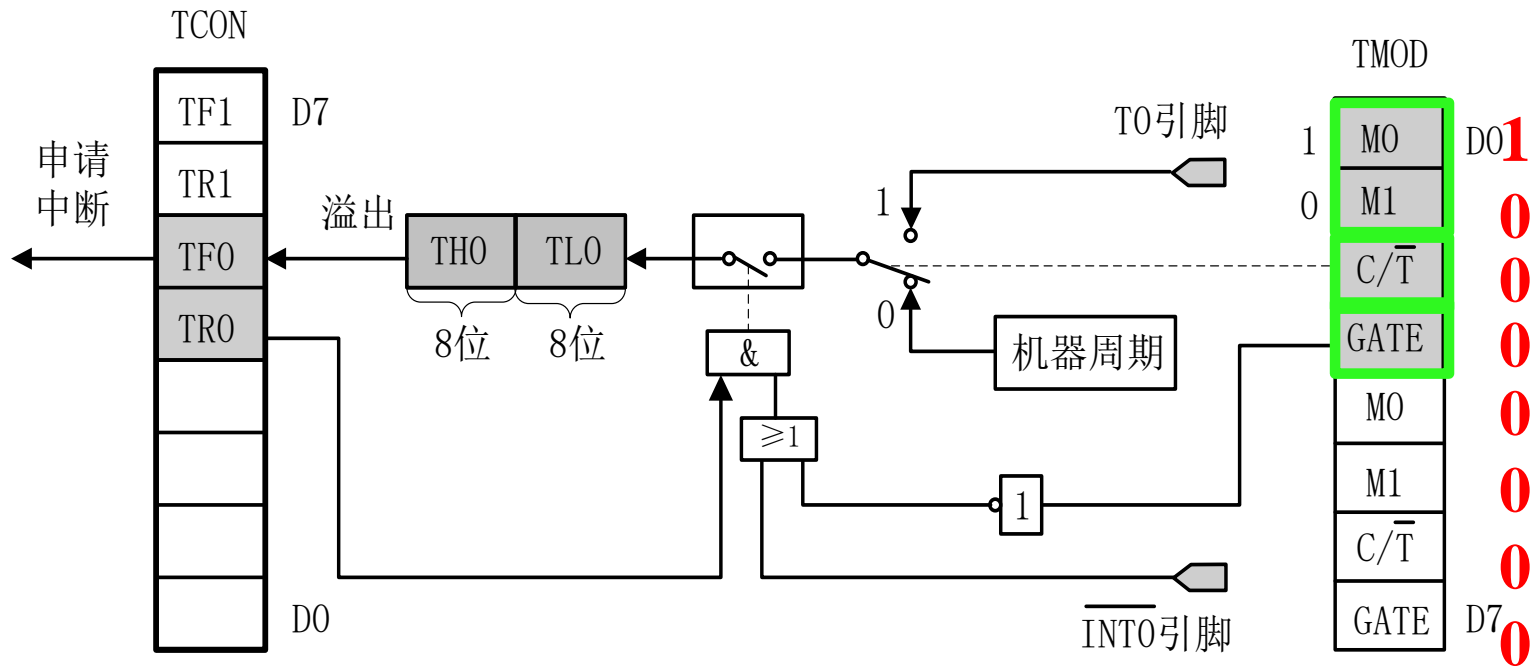


第七章 定时器

流程图 (方式1)

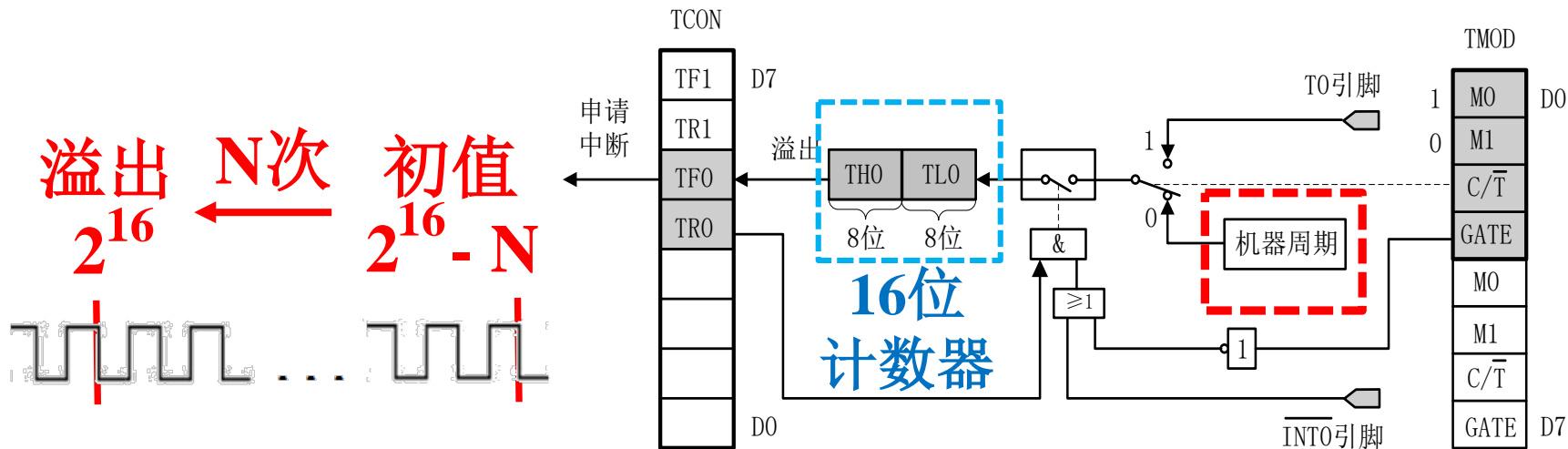


第七章 定时器



采用定时器0，工作方式1：16位定时/计数器
 设置 M1=0, M0=1
 定时模式，设置 C/T=0
 无特殊要求，设置 GATE=0，用TR0即可控制启停
 因此，TMOD=0x01

第七章 定时器



工作方式1，定时50ms，计算计数器初值：
 晶振频率为11.0592 MHz，12分频后，机器周期为1.08507 μ s。
 每过一次机器周期，计数器+1；
 定时时间50ms=50000 μ s，计数个数 $N=50000/1.08507=46080$ ；
 定时初值= $2^{16}-46080=19456=4c00H$ ；
 因此，TH0=0x4c，TL0=0x00

思考：为什么无法直接定时1s？怎么解决？

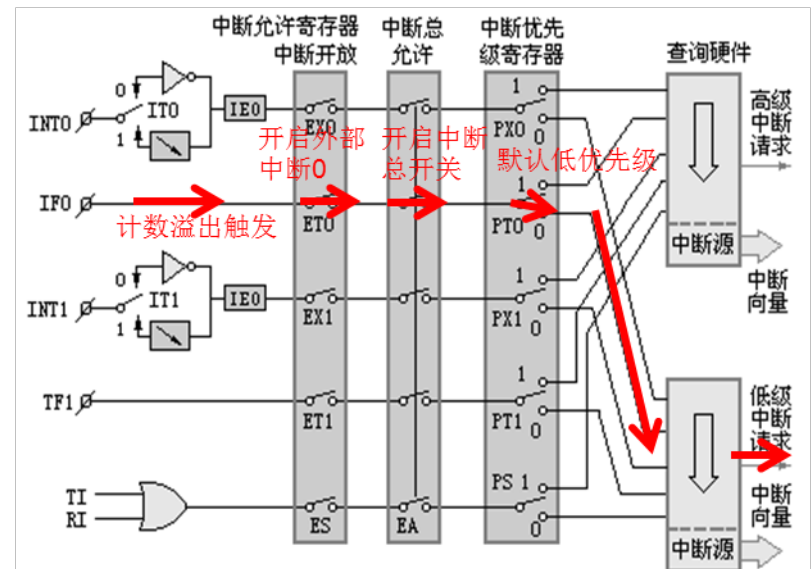
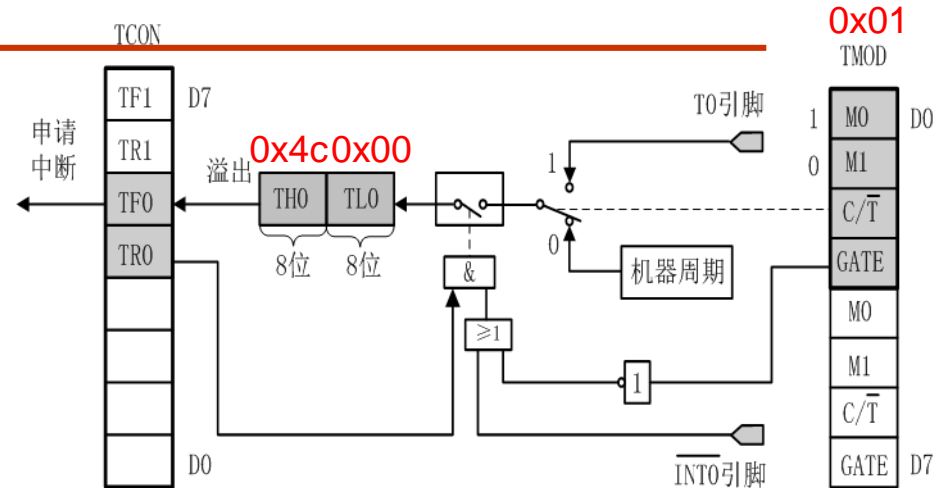
第七章 定时器

程序代码（方式1）

```

#include <reg51.h>
sbit led = P1^0;
unsigned char num; //计数全局变量

void main() //主程序
{
    TMOD = 0x01; //设置定时器0工作方式
    TH0 = 0x4C; //设置计数器的初值
    TL0 = 0x00;
    EA = 1; //开中断总开关
    ET0 = 1; //开定时器0中断
    TR0 = 1; //定时器0开始计时
    while(1); //循环，等待中断发生
}
    
```



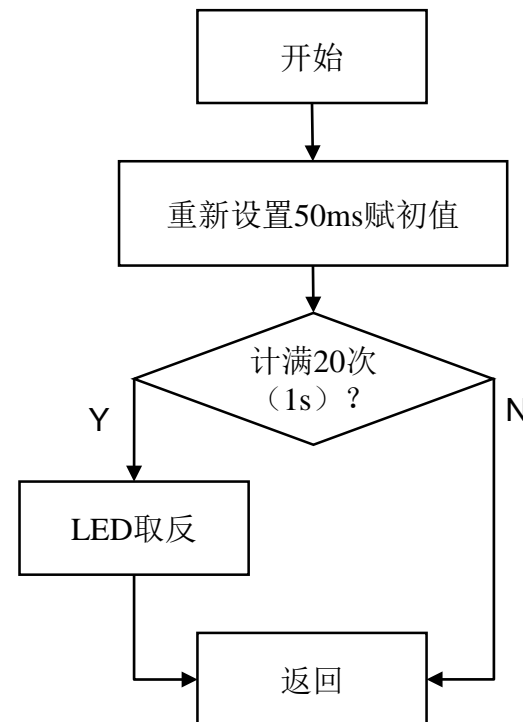
第七章 定时器

■ 程序代码（方式1）

```

//中断服务程序，中断号为1
void TIMER0() interrupt 1
{
    //重装初值，保证每次定时时间相同
    TH0 = 0x40;
    TL0 = 0x00;
    num++;
    if(num==20) //计到20次，说明1s时间到
    {
        num=0;
        led=~led;
    }
}
    
```

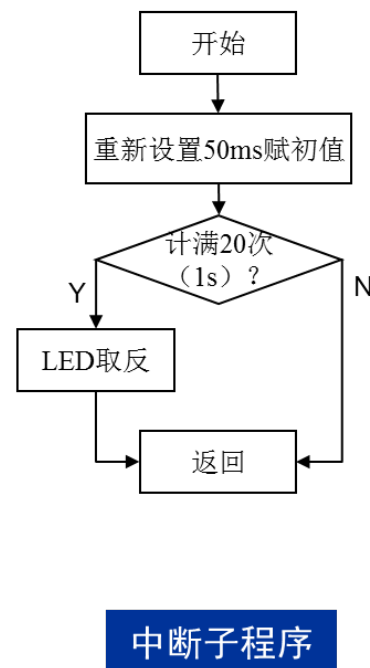
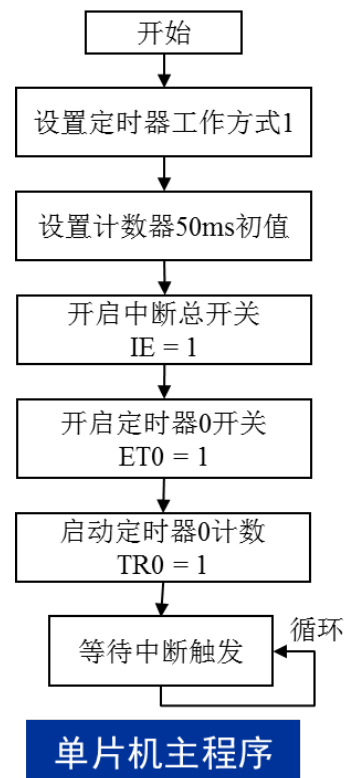
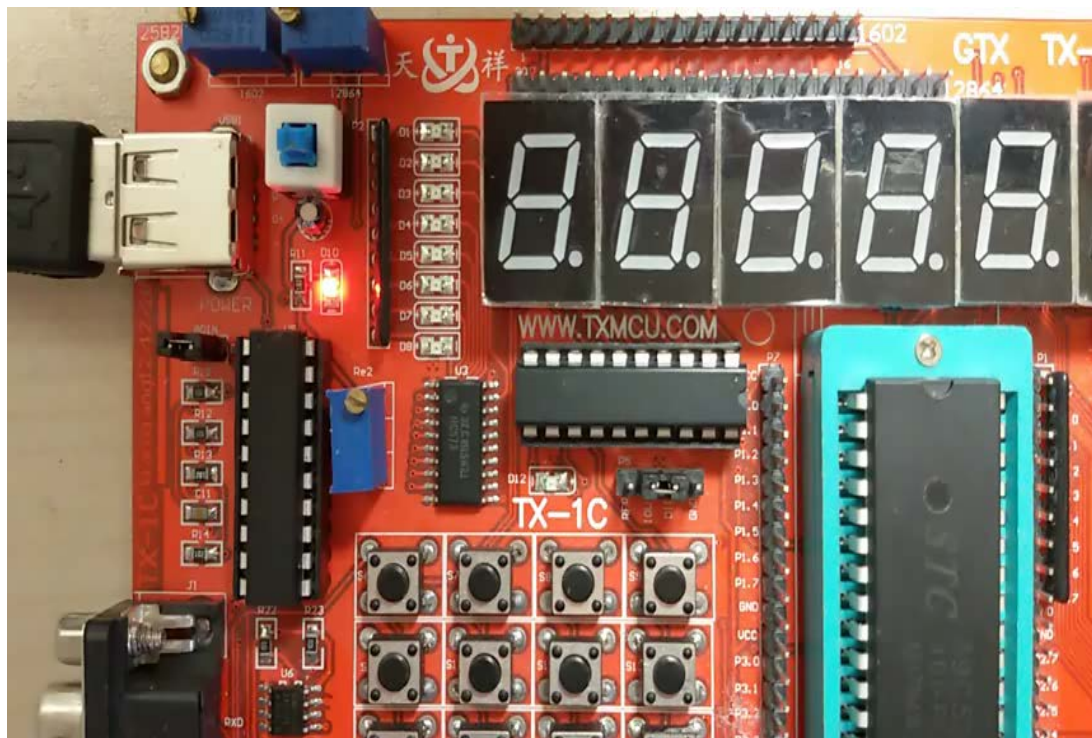
中断源	入口地址	中断号
外部中断0	0003H	0
定时器0溢出	000BH	1
外部中断1	0013H	2
定时器1溢出	001BH	3
串行口中断	0023H	4



第七章 定时器

■ 烧写（方式1）

➤ 观察现象：



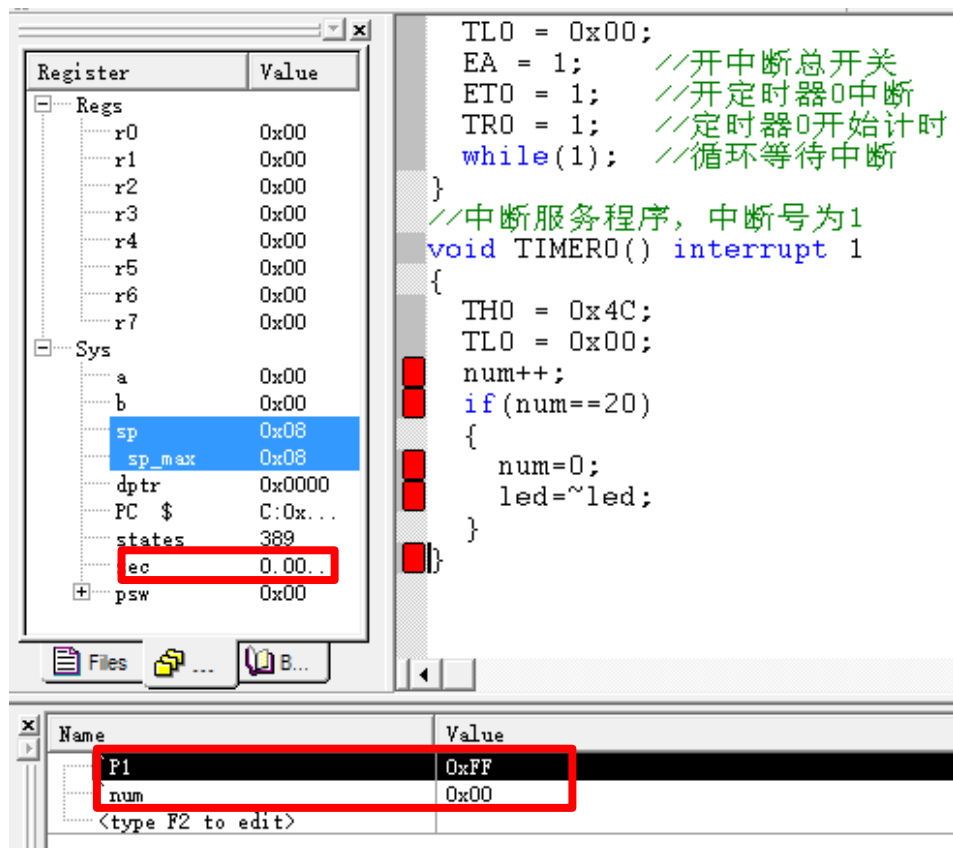
第七章 定时器

■ 调试

➤ 在中断程序中设置断点

➤ 观察：

- 中断触发时间间隔
- 计数变量num
- 端口P1.0电平



Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x08
sp_max	0x08
dptr	0x0000
PC \$	C:0x...
states	389
ec	0.00...
psw	0x00

```

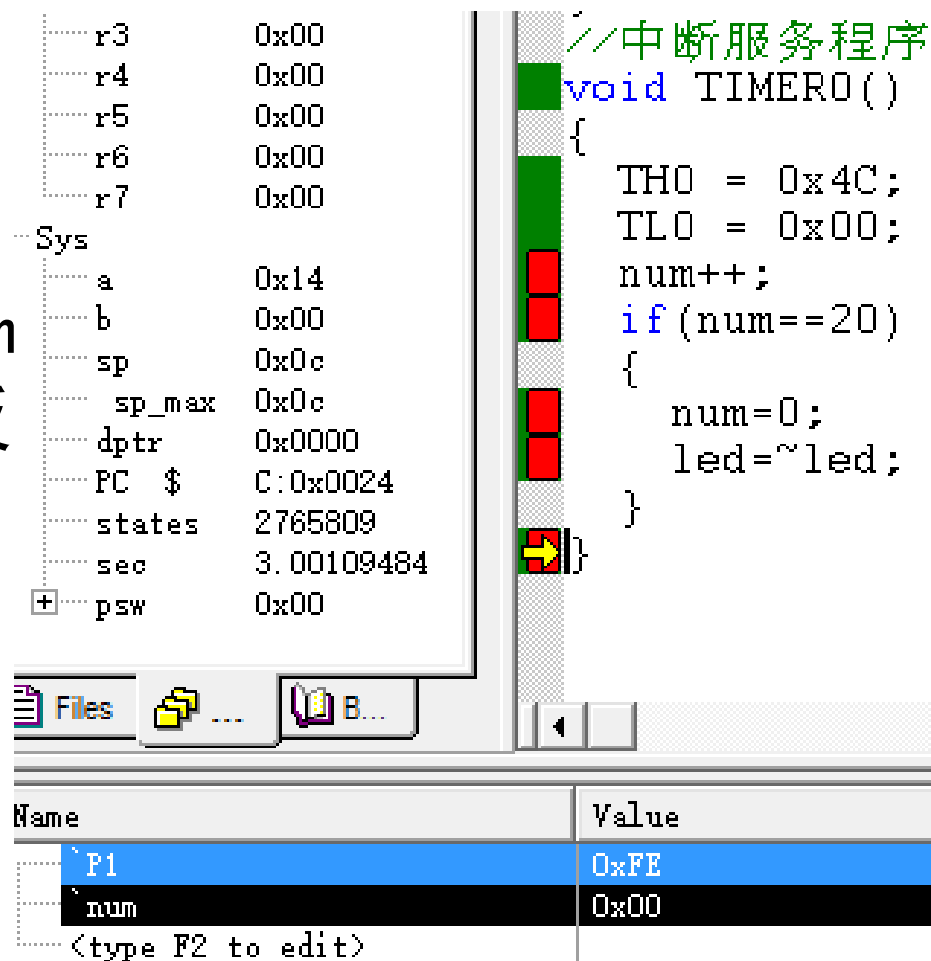
TLO = 0x00;
EA = 1; //开中断总开关
ETO = 1; //开定时器0中断
TRO = 1; //定时器0开始计时
while(1); //循环等待中断
}
//中断服务程序, 中断号为1
void TIMER0() interrupt 1
{
    TH0 = 0x4C;
    TLO = 0x00;
    num++;
    if(num==20)
    {
        num=0;
        led=~led;
    }
}
    
```

Name	Value
P1	0xFF
num	0x00
<type F2 to edit>	

第七章 定时器

■ 调试

- 每0.05s (50ms) 中断一次, num加1
- num增加到20时, num清零, P1.0电平取反 (寄存器P1交替出现0xFF和0xFE)
- 即LED每1s改变一次亮灭



The screenshot shows a debugger interface with a register window on the left and a code window on the right. The register window displays the following values:

r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
--- Sys	
a	0x14
b	0x00
sp	0x0c
sp_max	0x0c
dptr	0x0000
PC \$	C:0x0024
states	2765809
sec	3.00109484
+ psw	0x00

The code window shows the following C code for the interrupt service routine:

```
//中断服务程序
void TIMERO()
{
    TH0 = 0x4C;
    TLO = 0x00;
    num++;
    if (num==20)
    {
        num=0;
        led=~led;
    }
}
```

At the bottom, a variable window shows the current values of P1 and num:

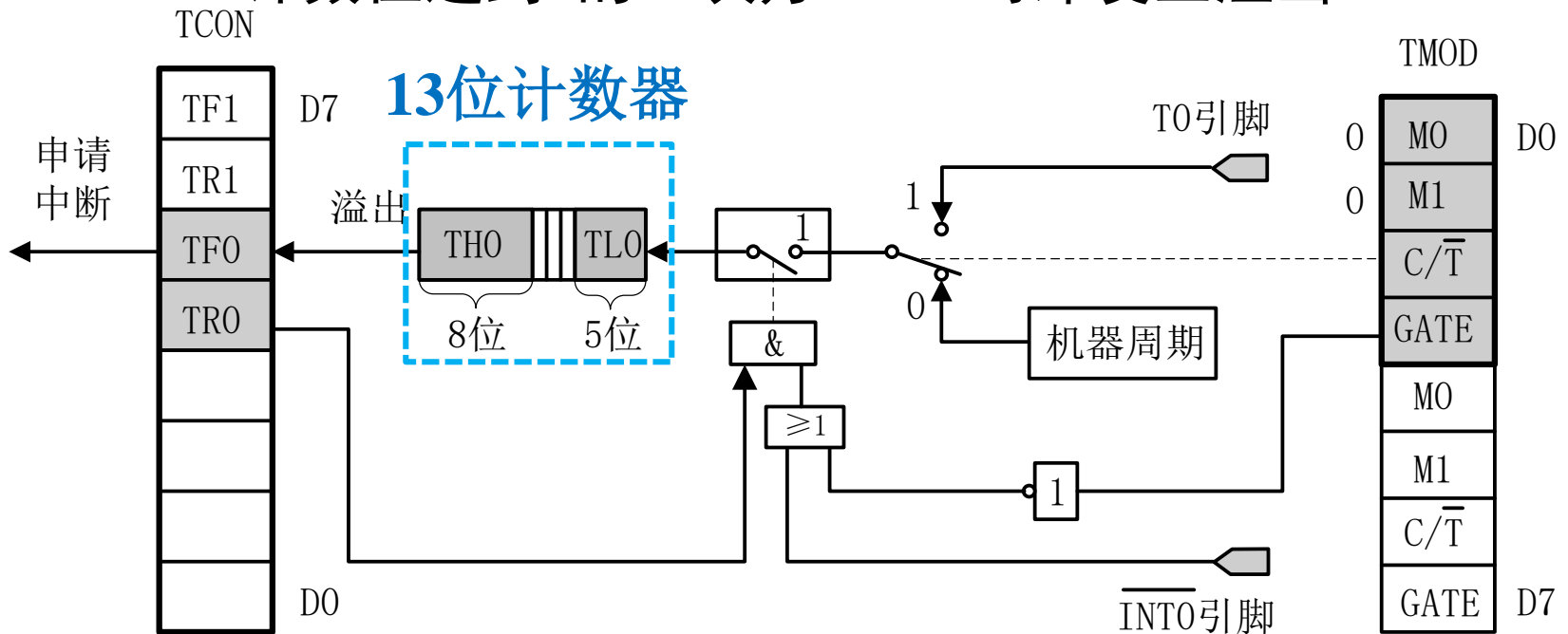
Name	Value
P1	0xFE
num	0x00
<type F2 to edit>	

第七章 定时器

其它工作方式对比

方式0

- 13位计数器，包括TLX的低5位和THX的8位
- 计数值达到2的13次方=8192时即发生溢出





第七章 定时器

- 考虑方式0和方式1重装初值的步骤
 - 操作繁琐
 - 设置初值需要时间，会造成定时不准确

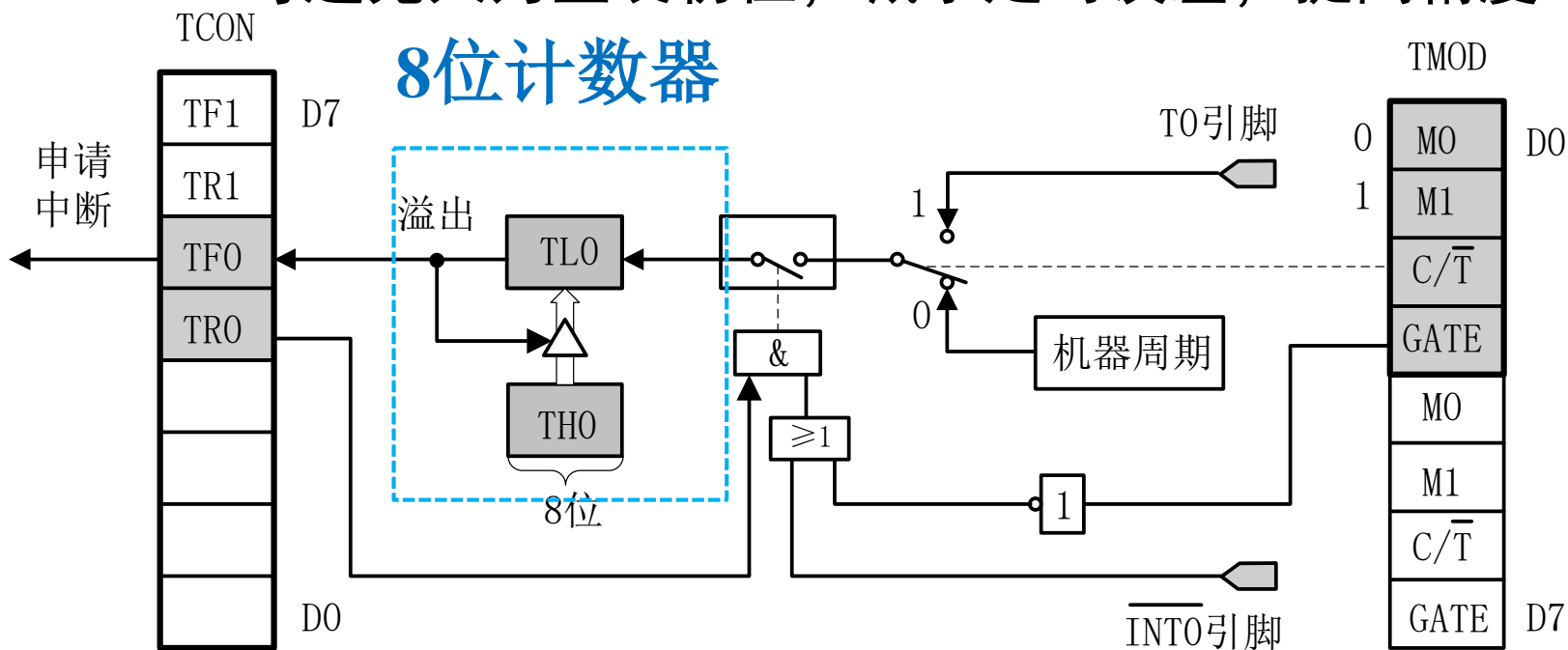
```
void TIMER0() interrupt 1
//中断服务程序，中断号为1
{
    TH0 = 0x4C; //重装初值
    TL0 = 0x00;
    ...
}
```

第七章 定时器

其它工作方式对比

方式2

- 自动重装初值的8位定时/计数器
- 可避免人为重装初值，减小定时误差，提高精度



第七章 定时器

■ 其它工作方式对比

➤ 方式2

- 中断服务程序对比:

方式2

```
void 函数名() interrupt 2  
{  
    语句;  
}
```

无需重装初值

方式1或0

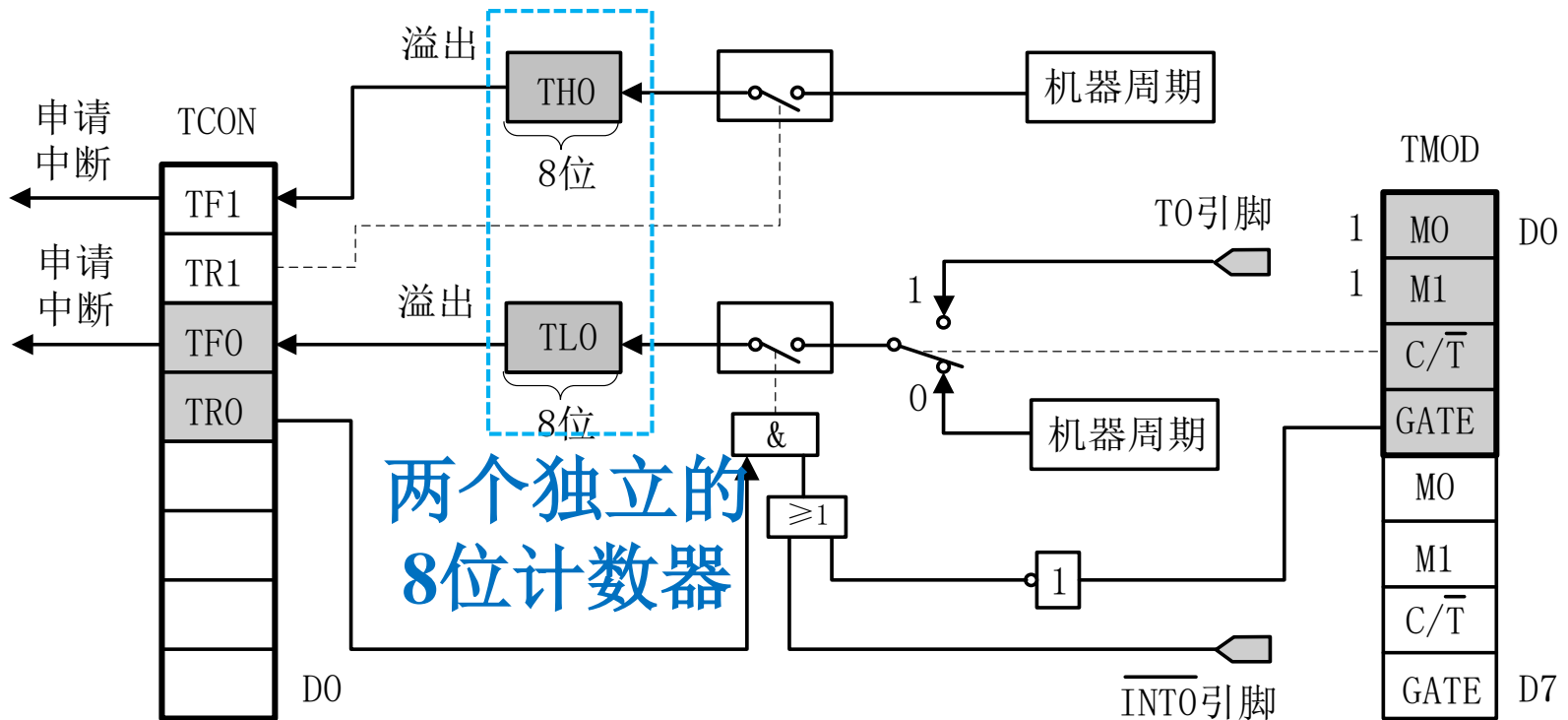
```
void 函数名() interrupt 0或1  
{  
    TH0 = ...; //重装初值  
    TL0 = ...;  
    语句;  
}
```

第七章 定时器

其它工作方式对比

方式3

- 将T0分为两个独立的8位计数器TH0和TL0



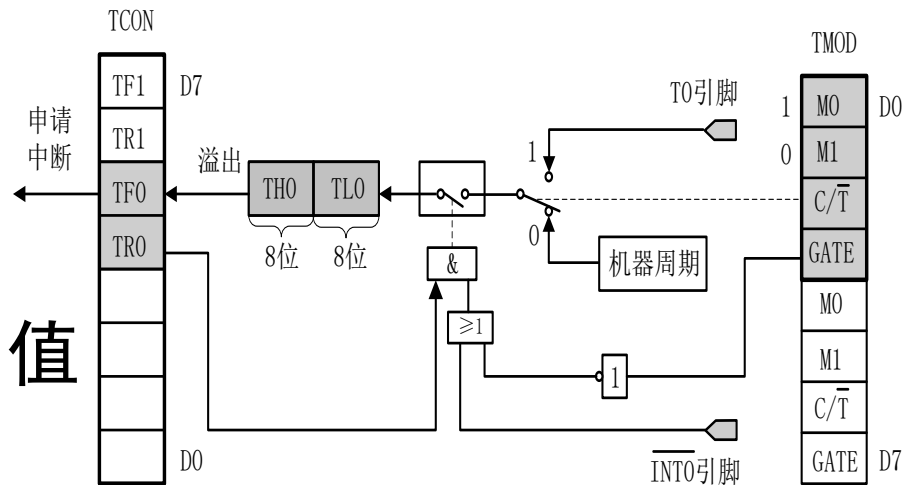
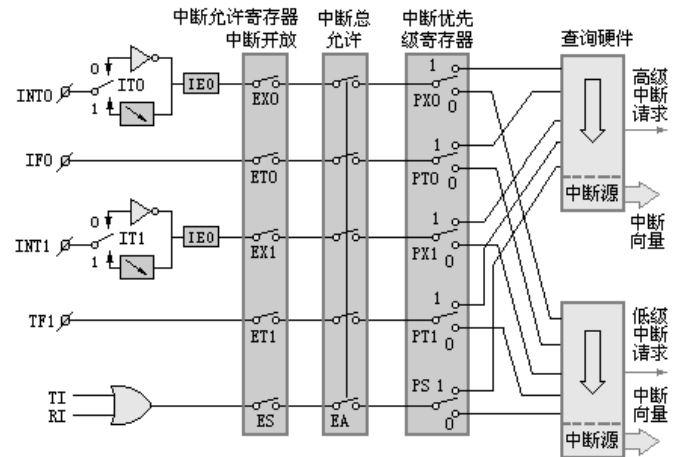
小结

■ 定时器

- 工作原理
- 工作方式
- 与中断的关系

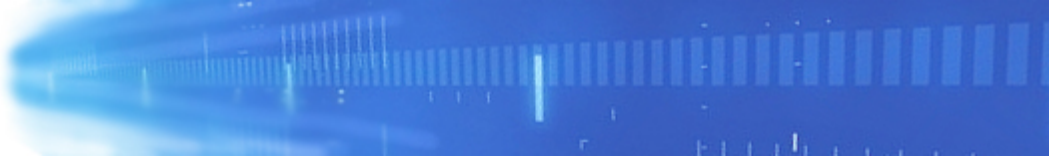
■ 定时器相关寄存器

- TMOD设置工作方式
- TCON设置定时器启停
- THX/TLX设置计数器初值
 - 初值的计算





北京航空航天大学
BEIHANG UNIVERSITY



再见