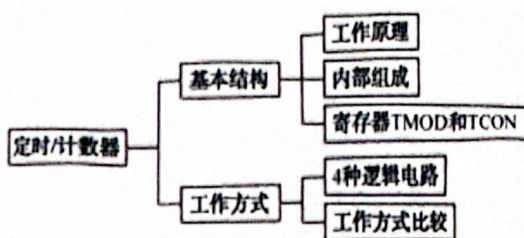


## 5.1 定时/计数器

知识分布网络



扫一扫看定时/计数器工作原理教学视频



扫一扫看定时/计数器工作原理演示文稿

### 5.1.1 定时/计数器的结构

#### 1. 定时/计数器的工作原理

定时/计数器的工作原理如图 5.1 所示。

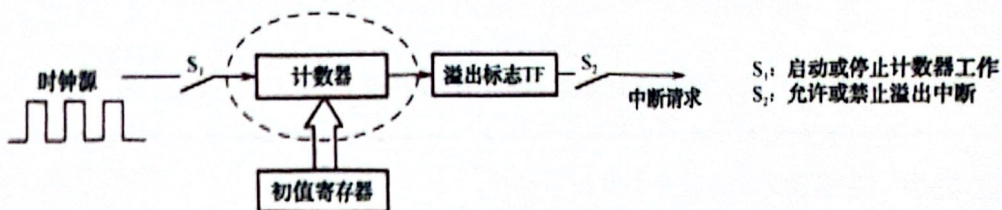


图 5.1 定时/计数器的工作原理



扫一扫看一个完整的数数儿过程微视频

关于定时/计数器的几个相关概念介绍如表 5.1 所示。

表 5.1 定时/计数器相关概念

概 念	说 明
计数器分类	计数器分为加法计数器和减法计数器，前者每来一个计数脉冲，计数值加 1；后者每来一个计数脉冲，计数值减 1
计数器位数	计数器的位数确定了计数器的最大计数个数 $M$ 和计数范围， $n$ 位计数器的最大计数个数 $M=2^n$ ，计数范围是 $0 \sim 2^n - 1$ 。例如 8 位计数器的最大计数个数 $M=256$ ，计数范围是 $0 \sim 255$
计数/定时功能	作为计数器使用时，计数时钟源来自外部信号引脚，记录该外部信号的脉冲个数；作为定时器使用时，计数时钟源来自内部时钟信号。对设定好的内部时钟脉冲个数进行计数所需要的时间就是定时时间
计数器溢出	当 $S_1$ 闭合时，计数器从计数初值开始，对计数脉冲进行加 1 或减 1 计数，当加法计数器计到最大值，或减法计数器计到最小值时，计数器产生溢出，将相应的溢出标志位 TF 置 1。计数器溢出也称为计数器翻转
计数器溢出处理	计数器溢出时，溢出标志位 $TF=1$ ，在程序中可以通过查询该位状态的方法获取计数器状态，查询方式编程参见程序 ex5_2.c；如果 $S_2$ 闭合，TF 还可以向 CPU 申请中断，采用中断方式进行计数溢出处理，中断方式编程参见程序 ex5_1.c
初值计算	假定计数器为 8 位加法计数器，计数脉冲来自内部时钟信号， $f_{\text{计数}}=1 \text{ MHz}$ ，若要定时 $250 \mu\text{s}$ ，计算计数初值的过程如下。 ① 计算计数周期： $T_{\text{计数周期}}=1/f_{\text{计数}}=1/(1 \text{ MHz})=1 \mu\text{s}$ ； ② 计算计数个数： $\text{count}_{\text{计数}}=T_{\text{定时时间}}/T_{\text{计数周期}}=250/1=250$ ；







续表

概念	说明
初值计算	③ 8位加法计数器最大计数个数: $M = 256$ ; ④ 加法计数器初值计算: $X_{\text{初值}} = M - \text{count}_{\text{计数}} = 256 - 250 = 6$
定时/计数器编程	定时/计数器编程包括以下4个步骤: ① 初始化, 确定计数/定时方式、工作方式等; ② 计算并设置计数初值; ③ 启动定时/计数器, 闭合 $S_1$ ; ④ 计数溢出处理 (查询和中断两种方式)



扫一扫看定时功能与计数功能演示文稿



扫一扫看定时功能与计数功能教学视频

## 小问答

问: 假定计数器为减法计数器, 计数初值计算有什么不同呢?

答: 对于减法计数器, 计数个数就是计数初值, 即  $X_{\text{初值}} = \text{count}_{\text{计数}}$ 。

## 2. 定时/计数器的组成

51 单片机内部有两个 16 位的可编程定时/计数器, 称为 T0 和 T1, 其逻辑结构如图 5.2 所示。51 单片机的定时/计数器由 T0、T1、工作方式寄存器 TMOD 和控制寄存器 TCON 四部分组成, T0 和 T1 均为加法计数器。下面我们从定时/计数器的工作过程来理解各部分的作用。

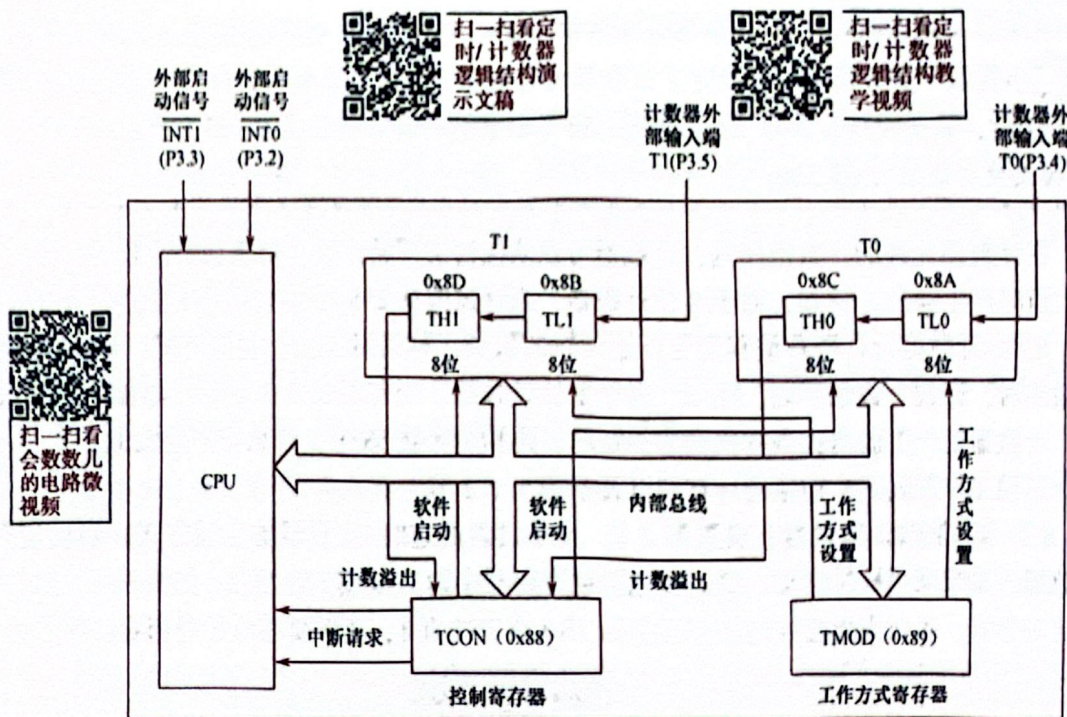


图 5.2 51 单片机定时/计数器逻辑结构

定时/计数器的工作过程如下:

(1) 设置定时/计数器的工作方式。通过对工作方式寄存器 TMOD 的设置, 确定相应的定时/计数器是定时功能还是计数功能, 确定工作方式及启动方法。







### 小问答

问: T0 和 T1 可编程选择为定时功能与计数功能, 二者有什么不同?

答: T0 或 T1 用做计数器时, 分别对从芯片引脚 T0 (P3.4) 或 T1 (P3.5) 上输入的脉冲进行计数, 外部脉冲的下降沿将触发计数, 每输入一个脉冲, 加法计数器加 1。计数器对外部输入信号的占空比没有特别的限制, 但必须保证输入信号的高电平与低电平的持续时间都在一个机器周期以上。

用做定时器时, 对内部机器周期脉冲进行计数, 由于机器周期是固定值, 故计数值确定时, 定时时间也随之确定。如果 51 单片机系统采用 12 MHz 晶振, 则计数周期为:  $T_{\text{机器周期}} = 1 / (12 \times 10^6 / 12) = 1 \mu\text{s}$ , 这是最短的定时周期。适当选择定时器的初值可获取各种定时时间。

定时/计数器的工作方式有四种: 方式 0、方式 1、方式 2 和方式 3, 参见 5.1.2 节。

定时/计数器的启动方式有两种: 软件启动和硬软件共同启动。从图 5.2 中可以看到, 除了从控制寄存器 TCON 发出的软件启动信号外, 还有两个外部启动信号引脚, 这两个引脚也是单片机的外部中断输入引脚。

(2) 设置计数初值。T0、T1 都是 16 位加法计数器, 分别由两个 8 位专用寄存器组成, T0 由 TH0 和 TL0 组成, T1 由 TH1 和 TL1 组成。TL0、TL1、TH0、TH1 的访问地址依次为 0x8A~0x8D, 每个寄存器均可被单独访问, 因此可以被设置为 8 位、13 位或 16 位的计数器来使用。

**小提示** 计数器的位数确定了计数器的计数范围。8 位计数器的计数范围是 0~255 (0xFF), 其最大计数值为 255。同理, 16 位计数器的计数范围是 0~65 535 (0xFFFF), 其最大计数值为 65 535。

在计数器允许的计数范围内, 计数器可以从任何值开始计数, 对于加 1 计数器, 当计到最大值时产生溢出。例如, 对于 8 位计数器, 当计数值从 255 再加 1 时, 计数值变为 0。

定时/计数器允许用户编程设定开始计数的数值, 称为赋初值。初值不同, 则计数器产生溢出时, 计数个数也不同。例如, 对于 8 位计数器, 当初值设为 100 时, 再加 1 计数 156 个, 计数器就产生溢出; 当初值设为 200 时, 再加 1 计数 56 个, 计数器产生溢出。

不同工作方式下, 初值的计算和设置参见 5.1.2 节。

(3) 启动定时/计数器。根据第 (1) 步中设置的定时/计数器的启动方式, 启动定时/计数器。如果采用软件启动, 则需要把控制寄存器中的 TR0 或 TR1 置 1; 如果采用硬软件共同启动方式, 不仅需要把控制寄存器中的 TR0 或 TR1 置 1, 还需要相应的外部启动信号为高电平。

**小提示** 当设置了定时器的工作方式并启动定时器工作后, 定时器就按被设定的工作方式独立工作, 不再占用 CPU 的操作时间, 只有在计数器计满溢出时才可能中断 CPU 当前的操作。



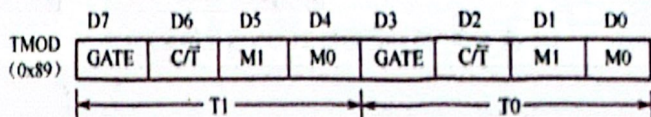




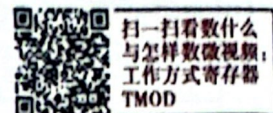
(4) 计数溢出。计数溢出标志位在控制寄存器 TCON 中, 用于通知用户定时/计数器已经计满, 用户可以采用查询方式或中断方式进行操作。

### 3. 定时/计数器工作方式寄存器 TMOD

TMOD 为定时/计数器的工作方式寄存器, 其格式如下:



TMOD 的低 4 位为 T0 的工作方式字段, 高 4 位为 T1 的工作方式字段, 它们的含义完全相同。



(1) M1 和 M0: 工作方式选择位。其含义如表 5.2 所示。

表 5.2 工作方式选择位的含义

M1	M0	工作方式	功能说明
0	0	方式 0	13 位计数器
0	1	方式 1	16 位计数器
1	0	方式 2	初值自动重载 8 位计数器
1	1	方式 3	T0: 分成两个 8 位计数器 T1: 停止计数

(2) C/T: 功能选择位。C/T=0 时, 设置为定时器工作方式; C/T=1 时, 设置为计数器工作方式。

(3) GATE: 门控位。当 GATE=0 时, 软件启动方式, 将 TCON 寄存器中的 TR0 或 TR1 置 1 即可启动相应定时器; 当 GATE=1 时, 硬软件共同启动方式, 软件控制位 TR0 或 TR1 需置 1, 同时还需 INT0 (P3.2) 或 INT1 (P3.3) 为高电平才可启动相应定时器, 即允许外中断 INT0、INT1 启动定时器。

**小提示** TMOD 不能位寻址, 只能用字节操作来设置定时器的的工作方式, 高 4 位定义 T1, 低 4 位定义 T0。复位时, TMOD 所有位均清零。

任务 5-1 中设置 T1 为软件启动方式、定时功能、工作方式 1, 则 GATE=0、C/T=0、M1 M0=01, 因此, 高 4 位应为 0001; T0 未用, 低 4 位可随意置数, 但低两位不可为 11 (因在工作方式 3 时, T1 将停止计数), 一般将其设为 0000。因此, 采用下面语句设置定时/计数器的工作方式:

```
TMOD=0x10; //设置 T1 为工作方式 1
```

### 4. 定时/计数器控制寄存器 TCON

定时/计数器控制寄存器 TCON 的作用是控制定时器的启动、停止, 标识定时器的溢出和中断情况。TCON 的格式如下:







TCON	0x8F	0x8E	0x8D	0x8C	0x8B	0x8A	0x89	0x88
(0x88)	TF1	TR1	TF0	TH0	IE1	IT1	IE0	IT0

各位的含义如表 5.3 所示。



扫一扫看定  
时/计数器  
TCON寄存  
器演示文稿



扫一扫看数数儿的  
开始与结束微  
视频: 控制寄存  
器TCON

表 5.3 控制寄存器 TCON 各位的含义

控制位	位名称	说 明
TF1	T1 溢出中断标志	TCON. 7 当 T1 计数满产生溢出时, 由硬件自动置 TF1 = 1。在中断允许时, 该位向 CPU 发出 T1 的中断请求, 进入中断服务程序后, 该位由硬件自动清零。在中断屏蔽时, TF1 可做查询测试用, 此时只能由软件清零
TR1	T1 运行控制位	TCON. 6 由软件置 1 或清零来启动或关闭 T1。当 GATE = 1, 且 INT1 为高电平时, TR1 置 1 启动 T1; 当 GATE = 0 时, TR1 置 1 即可启动 T1
TF0	T0 溢出中断标志	TCON. 5 与 TF1 相同
TH0	T0 运行控制位	TCON. 4 与 TR1 相同
IE1	外部中断 1 ( $\overline{\text{INT1}}$ ) 请求标志位	TCON. 3
IT1	外部中断 1 触发方式选择位	TCON. 2
IE0	外部中断 0 ( $\overline{\text{INT0}}$ ) 请求标志位	TCON. 1
IT0	外部中断 0 触发方式选择位	TCON. 0

TCON 中的低 4 位用于控制外部中断, 与定时/计数器无关, 将在第 5.2 节中介绍。当系统复位时, TCON 的所有位均清零。

TCON 可以进行位操作, 溢出标志位清零或启动定时器都可以用位操作语句, 例如:

```
TR1=1;           //启动 T1
TF1=0;           //T1溢出标志位清零
```



**小提示** 任务 5-1 中程序 ex5\_2.c 采用查询溢出标志位 TF1 方式确认 50 ms 定时时间到, 查询语句如下:

```
while (!TF1);      //TF1 由 0 变 1, 定时时间到
TF1=0;             //查询方式下, TF1 必须由软件清零
```

### 5.1.2 定时/计数器的工作方式

如表 5.2 所示, 工作方式寄存器 TMOD 中的 M1 和 M0 位用于选择四种工作方式, 逻辑电路如图 5.3 所示。





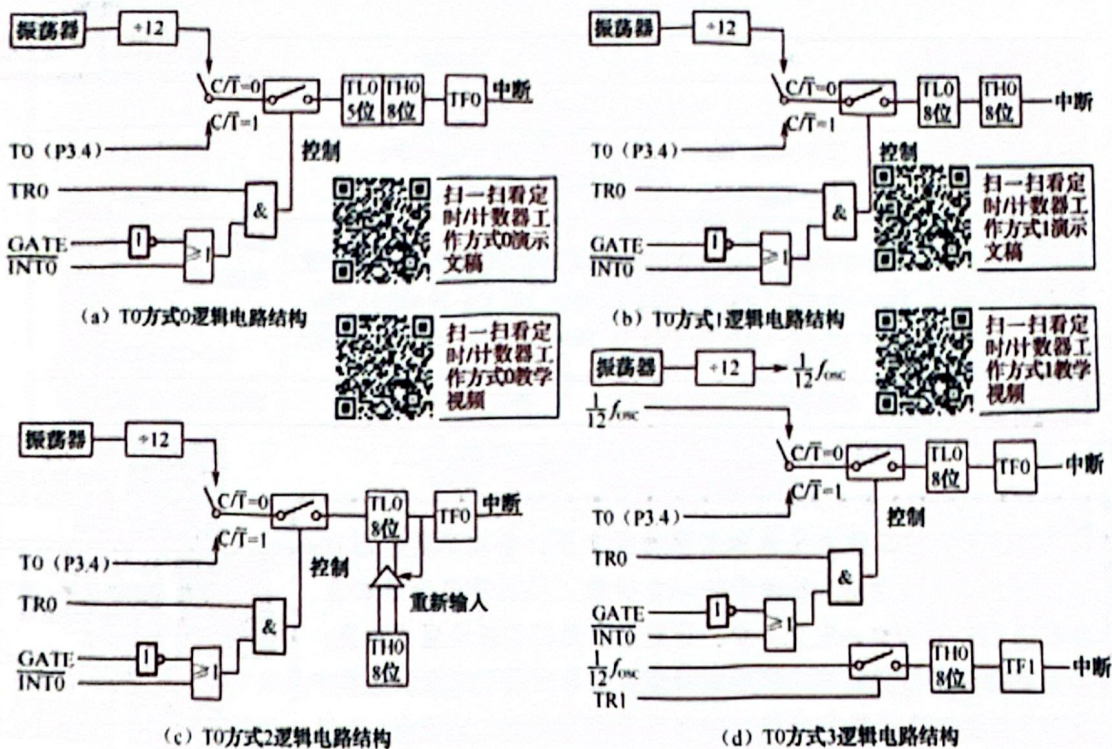


图 5.3 定时/计数器四种工作方式的逻辑电路

### 小问答

问：在图 5.3 的四种工作方式中， $C/\bar{T}$  和 GATE 的作用分别是什么？

答：当  $C/\bar{T}=0$  时，多路开关连接 12 分频器输出，T0 为定时功能，对机器周期计数。

当  $C/\bar{T}=1$  时，多路开关与 T0 (P3.4 引脚) 相连，外部计数脉冲由 T0 脚输入，当外部信号电平发生由 1 到 0 的负跳变时，计数器加 1，T0 为计数功能。

当 GATE=0 时，或门被封锁， $\overline{INT0}$  信号无效。或门输出常 1，打开与门，TR0 直接控制 T0 的启动和关闭。TR0=1，接通控制开关，T0 从初值开始计数直至溢出。TR0=0，则与门被封锁，控制开关被关断，停止计数。

当 GATE=1 时，与门的输出由  $\overline{INT0}$  的输入电平和 TR0 位的状态来确定。若 TR0=1 则与门打开，外部信号电平通过  $\overline{INT0}$  引脚直接开启或关闭 T0，当  $\overline{INT0}$  为高电平时，允许计数，否则停止计数；若 TR0=0，则与门被封锁，控制开关被关闭，停止计数。

定时器 T0 和 T1 都可以设置为工作方式 0、1 和 2，可以用做定时/计数功能，主要用法如表 5.4 所示。

表 5.4 定时器工作方式比较

工作方式	工作方式 0	工作方式 1	工作方式 2
计数位数	13 位定时/计数器	16 位定时/计数器	8 位定时/计数器
计数寄存器	THi 高 8 位，TLi 低 5 位	THi 高 8 位，TLi 低 8 位	TLi







续表

最大计数值 M	8 192	65 536	256
初值计算公式	$X_{\text{初值}} = M - T_{\text{定时时间}} / T_{\text{机器周期}}$		
初值设置	$THi = X_{\text{初值}} / 32;$ $TLi = X_{\text{初值}} \% 32;$	$THi = X_{\text{初值}} / 256;$ $TLi = X_{\text{初值}} \% 256;$	$THi = X_{\text{初值}};$ $TLi = X_{\text{初值}};$
初值设置举例	假定定时时间为 5 ms, 初值设置: $THi = (8\ 192 - 5\ 000 / 1) / 32;$ $TLi = (8\ 192 - 5\ 000 / 1) \% 32;$	假定定时时间为 50 ms, 初值设置: $THi = (65\ 536 - 50\ 000 / 1) / 256;$ $TLi = (65\ 536 - 50\ 000 / 1) \% 256;$	假定定时时间为 250 $\mu\text{s}$ , 初值设置: $THi = 256 - 250 / 1;$ $TLi = 256 - 250 / 1;$
特点	初值不可自动重载	初值不可自动重载	初值可以自动重载

注: 表中 i 表示 0 或 1, 晶振频率假定为 12 MHz,  $T_{\text{机器周期}} = 1 \mu\text{s}$ 。

**小提示** 在工作方式 0 和工作方式 1 下, 每次计数溢出后, 计数器自动复位为 0, 要进行新一轮计数, 必须重置计数初值, 在程序 ex5\_1.c 和 ex5\_2.c 中, 计数溢出后都重新设置了初值。

重新设置初值影响定时时间精度, 又导致编程麻烦。工作方式 2 具有初值自动装载功能, 适合用于较精确的定时场合。

以 T0 为例, 在工作方式 2 下, TLO 用做 8 位计数器, TH0 用来保持初值。编程时, TLO 和 TH0 必须由软件赋予相同的初值。一旦 TLO 计数溢出, TFO 将被置位, 同时, TH0 中保存的初值自动装入 TLO, 进入新一轮计数, 如此重复循环不止。

只有 T0 可以设置为工作方式 3, T1 设置为工作方式 3 后不工作。T0 在工作方式 3 时的工作情况如下:

T0 被分解成两个独立的 8 位计数器 TLO 和 TH0。

TLO 占用 T0 的控制位、引脚和中断源, 包括  $\overline{C/T}$ 、GATE、TR0、TFO 和 T0 (P3.4) 引脚、 $\overline{INT0}$  (P3.2) 引脚。可定时也可计数, 除计数位数不同于工作方式 0 外, 其功能、操作与工作方式 0 完全相同。

TH0 占用 T1 的控制位 TF1 和 TR1, 同时还占用了 T1 的中断源, 其启动和关闭仅受 TR1 控制。TH0 只能对机器周期进行计数, 可以用做简单的内部定时, 不能用做对外部脉冲进行计数, 是 T0 附加的一个 8 位定时器。

**小提示** 当 T0 在工作方式 3 时, T1 仍可设置为方式 0、方式 1 或方式 2。但由于 TR1、TF1 和 T1 的中断源已被 T0 占用, 因此, 定时器 T1 仅由控制位  $\overline{C/T}$  切换其定时或计数功能。当计数器计满溢出时, 只能将输出送往串行口。在这种情况下, T1 一般用做串行口波特率发生器或不需要中断的场合。因 T1 的 TR1 被占用, 当设置好工作方式后, T1 自动开始计数; 当送入一个设置 T1 为工作方式 3 的方式字后, T1 停止计数。



扫一扫看定时/计数器初值计算演示文稿



扫一扫看从哪里数起微视频: 设置定时器计数器的初始值



扫一扫看定时功能应用实例演示文稿



扫一扫看定时功能应用实例操作训练



扫一扫看定时/计数器工作方式 3 演示文稿



扫一扫看计数功能应用实例演示文稿



扫一扫看计数功能应用实例操作训练



扫一扫看工作各有千秋微视频: 工作方式 0、工作方式 2 和工作方式 3