



微机原理和接口技术

第十一讲 定时器/计数器3



提 纲

1. 定时器计数器概述

2. 定时器计数器的结构与控制

3. 定时器计数器的工作方式

4. 定时器计数器的初始化

5. 计数器的飞读

6. 定时的实现方法

7. 定时方式的应用

8. 计数方式的应用

9. 脉冲宽度的测量

10. 扩展外部中断

11. 实时时钟的设计

提 纲

9. 脉冲宽度的测量



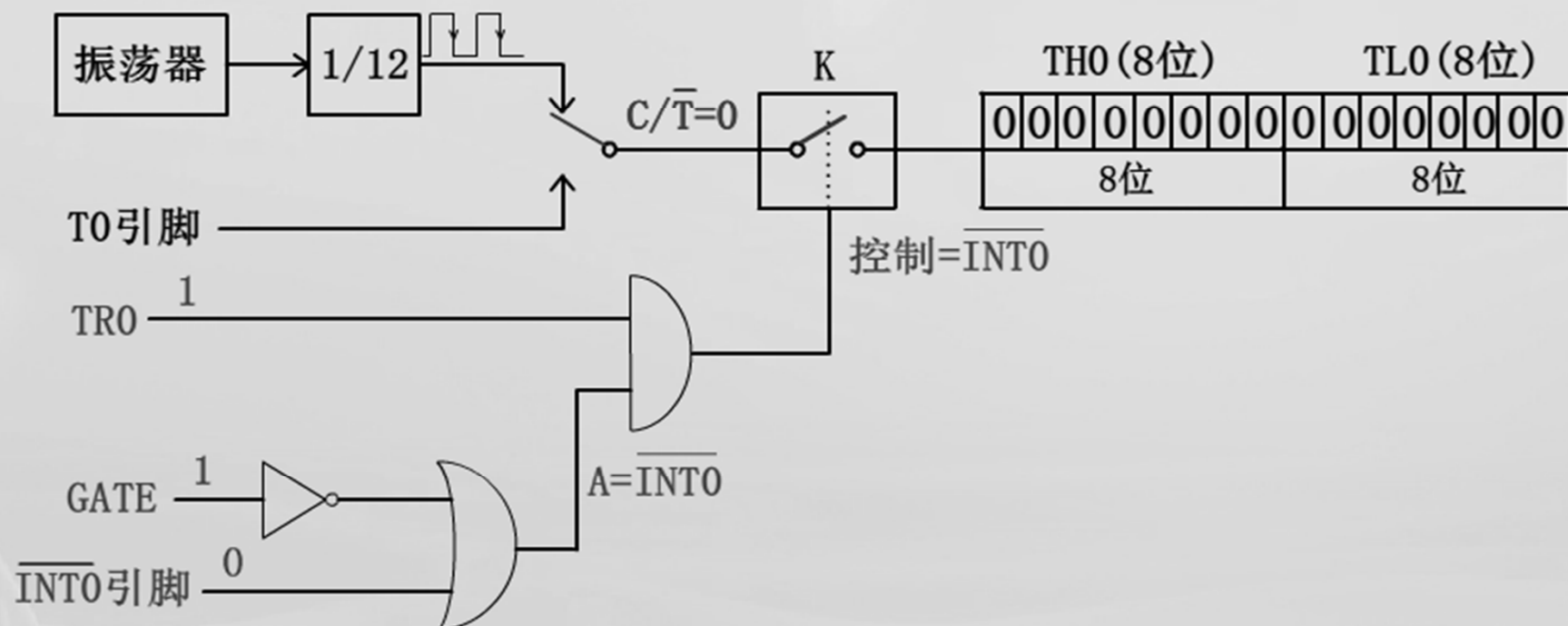
脉冲宽度的测量

在实际应用中，通常需要测量脉冲的频率、周期或高、低电平的宽度。利用经典8051 MCU中的定时器/计数器结合外部中断引脚，可以实现脉冲高电平宽度的测量。

脉冲高电平的测量，需要用到方式寄存器TMOD中的GATE信号，并要将被测脉冲信号连接到INT0或INT1引脚。

脉冲宽度的测量

定时器测量外部脉冲宽度



1. GATE=1, 则 “A” = $\overline{\text{INT0}}$; 当 TR0=1, “控制” = $\overline{\text{INT0}}$;
2. $\overline{\text{INT0}}$ 引脚为 “1”, K 闭合 (开始计数); $\overline{\text{INT0}}$ 引脚为 “0” 时, K 断开 (停止计数); 则 T0 记录的机器周期数, 即为 $\overline{\text{INT0}}$ 引脚的高电平宽度的时间



脉冲宽度的测量

测量方法：

- 设置为定时模式 ($C/\bar{T}=0$)，工作方式1，定时初值为0000H；
- 外部脉冲连接至INTi引脚；
- 设置GATE=1，在INTi为低电平时，令TRi=1；这时，定时器/计数器的启停受INTi引脚电平控制；高电平开始计数，低电平停止计数；

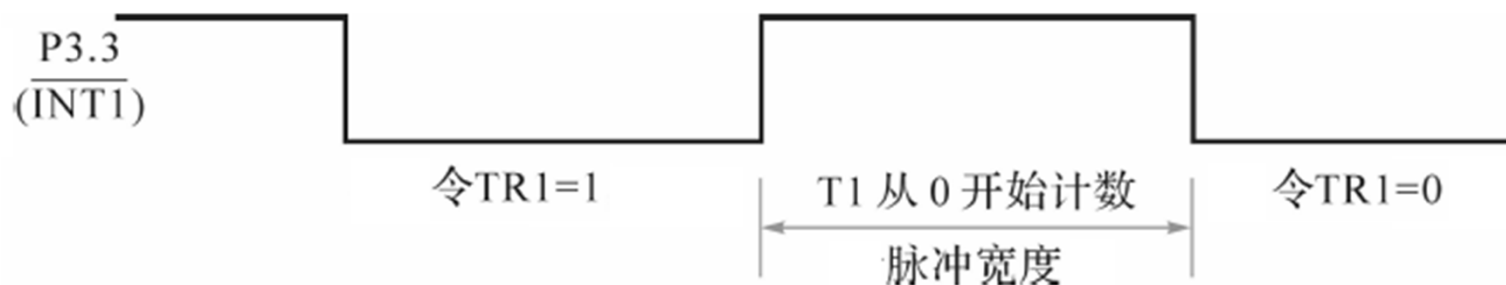
如何保证从高电平的开始那一刻开始计数？而不是从高电平中途计数？

- 停止计数时，读取Ti16位寄存器的内容，即为INTi引脚高电平期间累计的机器周期数，即高电平的宽度。

脉冲宽度的测量

例6-8：利用T1，测量INT1引脚上的高脉冲宽度。设晶振为12MHz，高脉冲宽度小于65ms。

分析：选择T1定时模式，工作方式1，定时初值设置为0。首先在INT1低电平期间设置TR1=1（为INT1控制T1的启停做准备），当INT1引脚从0变1时，就启动了T1工作（计数器从0开始累计），当INT1引脚变低时停止计数。此时T1寄存器中的计数值即为正脉冲的宽度（机器周期数）。所能测量到的最大脉冲宽度为65535 μ s。





脉冲宽度的测量

汇编程序:

```
MOV    TMOD,#90H      ;设置方式1、GATE=1
CLR     TR1
MOV     TL1,#0
MOV     TH1,#0
JB      P3.3, $        ;判断INT1是否为低电平（需要在低电平时，令TR1=1）
SETB    TR1            ;在INT1=0时，置TR1=1，使计数器的启停受控于INT1
JNB     P3.3, $        ;等待INT1变高电平，使计数器开始计，开始脉宽测量
JB      P3.3, $        ;等INT1变为低电平，使计数器停止计数，结束脉宽测量
CLR     TR1            ;置TR1=0，为下次测量做准备
MOV     A, TL1         ;读取16位计数器累计的机器周期数，高低8位分别送B、A寄存器
MOV     B, TH1
SJMP    $              ;一次测量结束
```


提 纲

10. 扩展外部中断



扩展外部中断

外部中断外展方法：

- 把T0、T1设置为计数方式，选用工作方式2，设置初值为FFH，允许中断。
- 外部中断源连接到T0、T1引脚，则在其引脚上发生负跳变时（产生一个下降沿），定时器/计数器加1，就会产生溢出而向CPU请求中断。
- 利用这个特性可以把T0（P3.4）和T1（P3.5）两个引脚扩展为外部中断请求引脚，中断的触发条件是下降沿触发，中断标志为T0、T1的溢出标志TF0和TF1。
- 其作用和功能与外部中断INT0、INT1完全相同。这样就把经典8051 MCU的外部中断源扩展到了4个。



扩展外部中断

例：利用定时器/计数器T0扩展外部中断。

```
汇编程序:      ORG 0000H
                  LJMP MAIN          ;转主程序
                  ORG 000BH          ;T0中断入口
                  LJMP T0INT         ;转T0中断服务

                  ORG 0100H
MAIN:  MOV  TMOD, #06H              ;计数模式, 工作方式2
        MOV  TL0, #0FFH            ;设置初值
        MOV  TH0, #0FFH            ;设置重装载初值
        SETB EA                    ;CPU中断开放
        SETB ET0                   ;T0中断开放
        SET  TR0                   ;启动T0
        SJMP $                     ;模拟主程序

                  ORG 2000H          ;中断程序
T0INT: ...
        RETI
```

提 纲

11. 实时时钟的设计



实时时钟的设计

1. 如何获得1s定时

对于16位的工作方式1，其最长定时时间是65536个机器周期，无法直接获得1s的定时，而需要定时器的硬件定时和软件计数相结合来实现。

可以利用定时器产生**50ms**定时，设置定时模式、工作方式1，用一个软件计数器（如一个工作寄存器或存储单元）用于累计50ms的个数，该软件计数器初值设为0，当从其从0累计20时，表示已有20个50ms，即1s定时时间到。

定时器的50ms定时，可以采用查询方式也可以用中断方式，通常用中断方式。



实时时钟的设计

2. 实时时钟的设计思路

- 在内部RAM中设置4个单元如43H-40H分别作为软件计数器（存放50ms个数、秒、分、时）的存储单元。
- 在50ms中断服务程序中，50ms个数加1，累加到20时表示1s到，再把50ms个数清0，秒单元内容加1，满60秒时，秒单元清0，分钟单元加1，满60分钟时，分钟单元清0，时单元加1，满24小时时，小时单元清0。
- 重复上述定时和运算过程，即可实现不断运行的实时时钟。
- 如果在时分秒的存储单元中，预先存入当前实际的时分秒，则运行实时时钟程序，就可以设计出一个电子钟；软件计数器中的时分秒，也可以通过按键输入，即可实现用按键设置或修改时间。

实时时钟的设计

例：用定时器T0设计一个能够计算时、分、秒的时钟。设晶振频率为12MHz。

主程序流程：



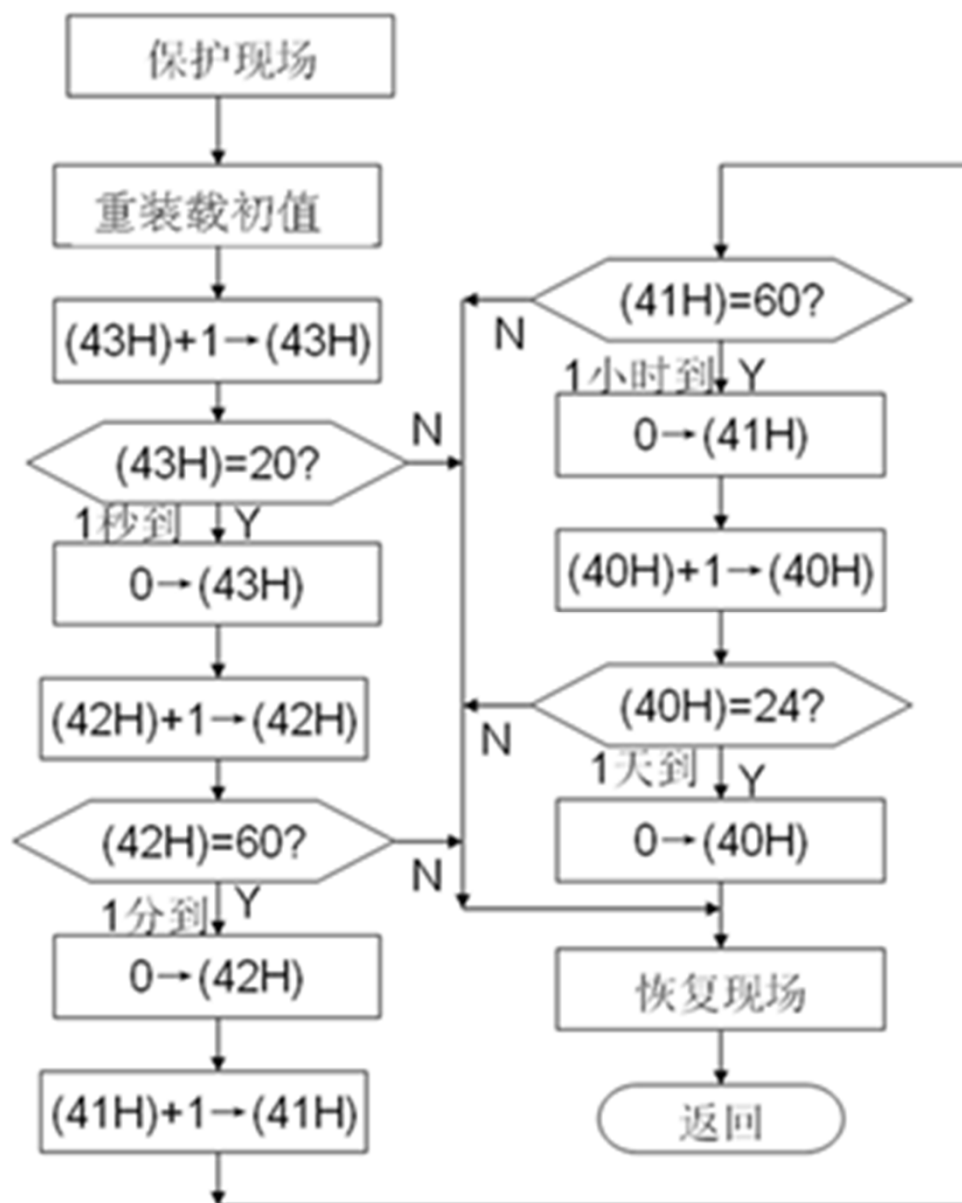
主程序包括：

- T0、中断和软件计数器初始化；
- 启动T0工作；
- 然后执行其他功能程序，本例中为反复调用显示子程序，显示实时时钟；同时随时相应50ms定时中断。

实时时钟的设计

中断服务程序
流程：

中断服务程序：
实现50ms、秒、
分、时的计时处
理和实时时钟的
更新。





实时时钟的设计

```
汇编程序:      ORG      0000H
                  SJMP     MAIN          ; 转主程序
                  ORG      000BH          ; T0中断入口
                  LJMP     T0INT         ; T0中断程序

                  ORG      0030H
MAIN:            MOV      TMOD, #01H      ; 设T0为方式1
                  CLR      A
                  MOV      40H, A         ; "时" 单元清 "0"
                  MOV      41H, A         ; "分" 单元清 "0"
                  MOV      42H, A         ; "秒" 单元清 "0"
                  MOV      43H, A         ; "50ms" 单元清 "0"
                  SETB     ET0            ; T0中断允许
                  SETB     EA            ; CPU中断允许
                  MOV      TH0, #3CH      ; 设置定时初值
                  MOV      TL0, #0B0H
                  SETB     TR0            ; 启动T0
HERE:            LCALL     DISPLAY        ; 循环调用显示程序 (以及其它程序)
                  SJMP     HERE          ; ( 也可1s调用一次显示程序)
```



实时时钟的设计

```
TOINT: PUSH    PSW                ;T0中断服务
        PUSH    ACC
        MOV     TH0, #3CH        ;重装载定时初值
        MOV     TL0, #0B0H
        INC     43H              ;50ms个数+1
        MOV     A, 43H
        CJNE    A, #20, RETURN    ;是否到1秒, 未到则返回
        MOV     43H, 00H         ;50ms个数清0
        MOV     A, #01H          ;秒数+1
        ADD     A, 42H
        DA      A                ;秒数进行十进制调整
        MOV     42H, A           ;保存修改后的秒数
        CJNE    A, #60, RETURN    ;是否到60秒, 未到则返回
        MOV     42H, #00H        ;计满60秒, 秒数清0
        MOV     A, #01H          ;分数+1
        ADD     A, 41H
        DA      A                ;分数进行十进制调整
        MOV     41H, A           ;保存修改后的分数
        CJNE    A, #60, RETURN    ;是否到60分, 未到则返回
        MOV     41H, #00H        ;满60分, 分数清0
        MOV     A, #01H          ;时数+1
        ADD     A, 40H
        DA      A                ;时数进行十进制调整
        MOV     40H, A
        CJNE    A, #24, RETURN    ;是否到24小时, 未到则返回
        MOV     40H, #00H        ;到24小时, 时数清0
```

中断服务程序

```
RETURN: POP     ACC              ;恢复现场
        POP     PSW
        RETI                    ;中断返回
        END
```

Thank you!

