
项目九 定时器中断控制

陈超然

汕头职业技术学院

课程目录

1. 复习回顾

2. 认识定时/计数器中断

- 什么是定时/计数器
- 定时器中断的过程

3. 定时器中断的控制

- 控制定时器中断的管理员
- 设置定时器中断的步骤

4. 实战任务

- 任务要求和分析
- 仿真软件练习
- 实际开发板练习

5. 实战作业

12MHz晶振的51单片机的机器周期是多少？

- ☒ A 1us
- ☐ B 2us
- ☐ C 1/12 us
- ☐ D 1/6 us

提交

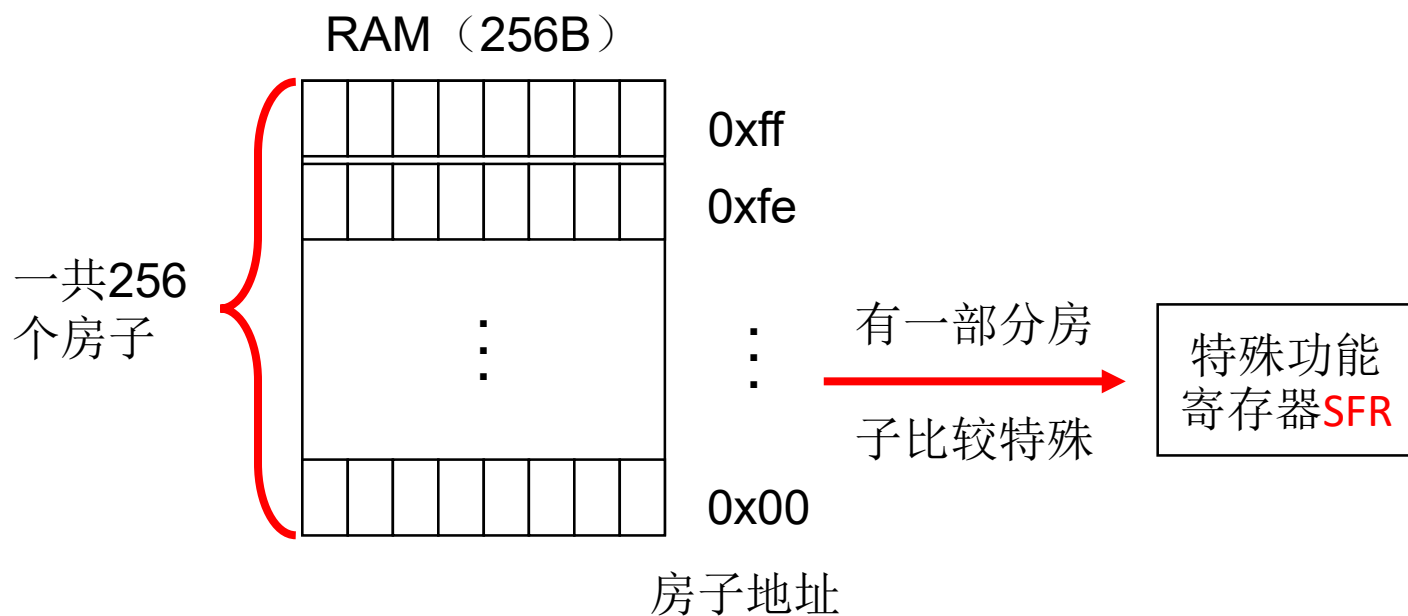
以下哪些是51单片机的中断源？

- ☒ A 外部中断0
- ☒ B 定时/计数器1
- ☐ C 串行口中断1
- ☒ D 外部中断1
- ☒ E 定时/计数器0

提交

51单片机的特殊功能寄存器SFR是放在哪个存储器？

- ☐ A 程序存储器RAM
- ☐ B 数据存储器ROM
- ☐ C 程序存储器ROM
- ☒ D 数据存储器RAM

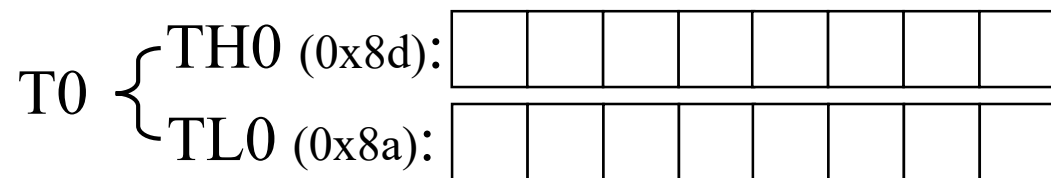


提交

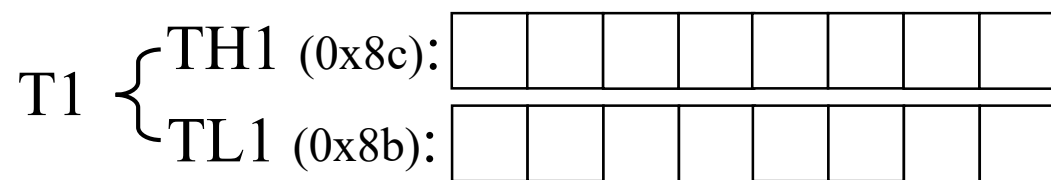
9.2 认识定时/计数器中断

1、什么是定时/计数器

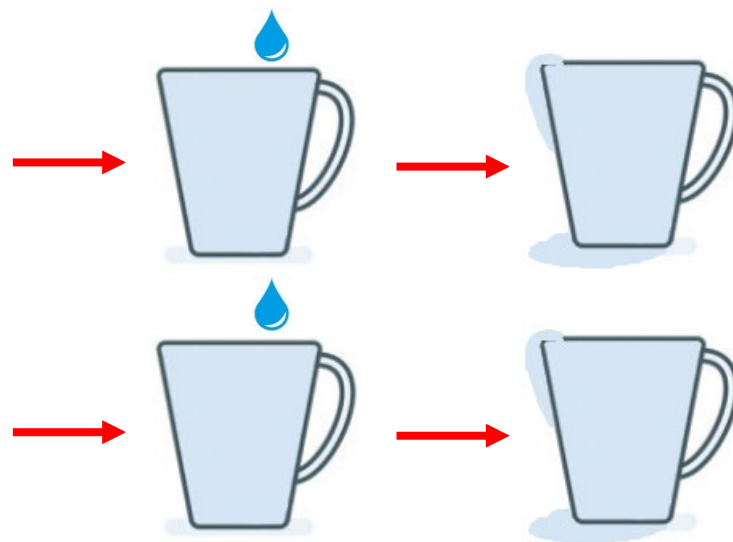
- 定时/计数器0和1是两个16位的SFR：T0和T1
- T0由两个8位的SFR组成：TH0和TL0
- T1由两个8位的SFR组成：TH1和TL1



高8位
低8位

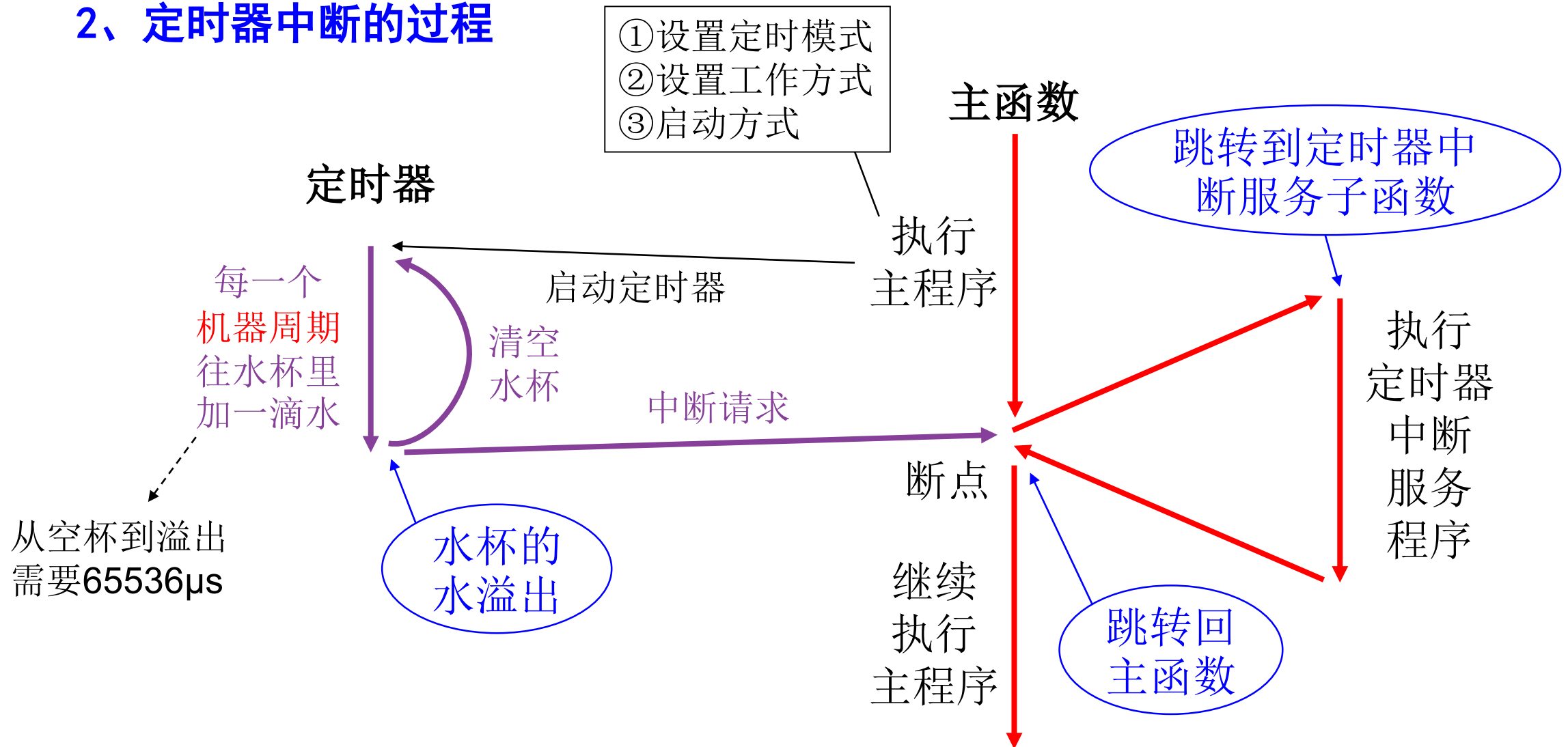


高8位
低8位



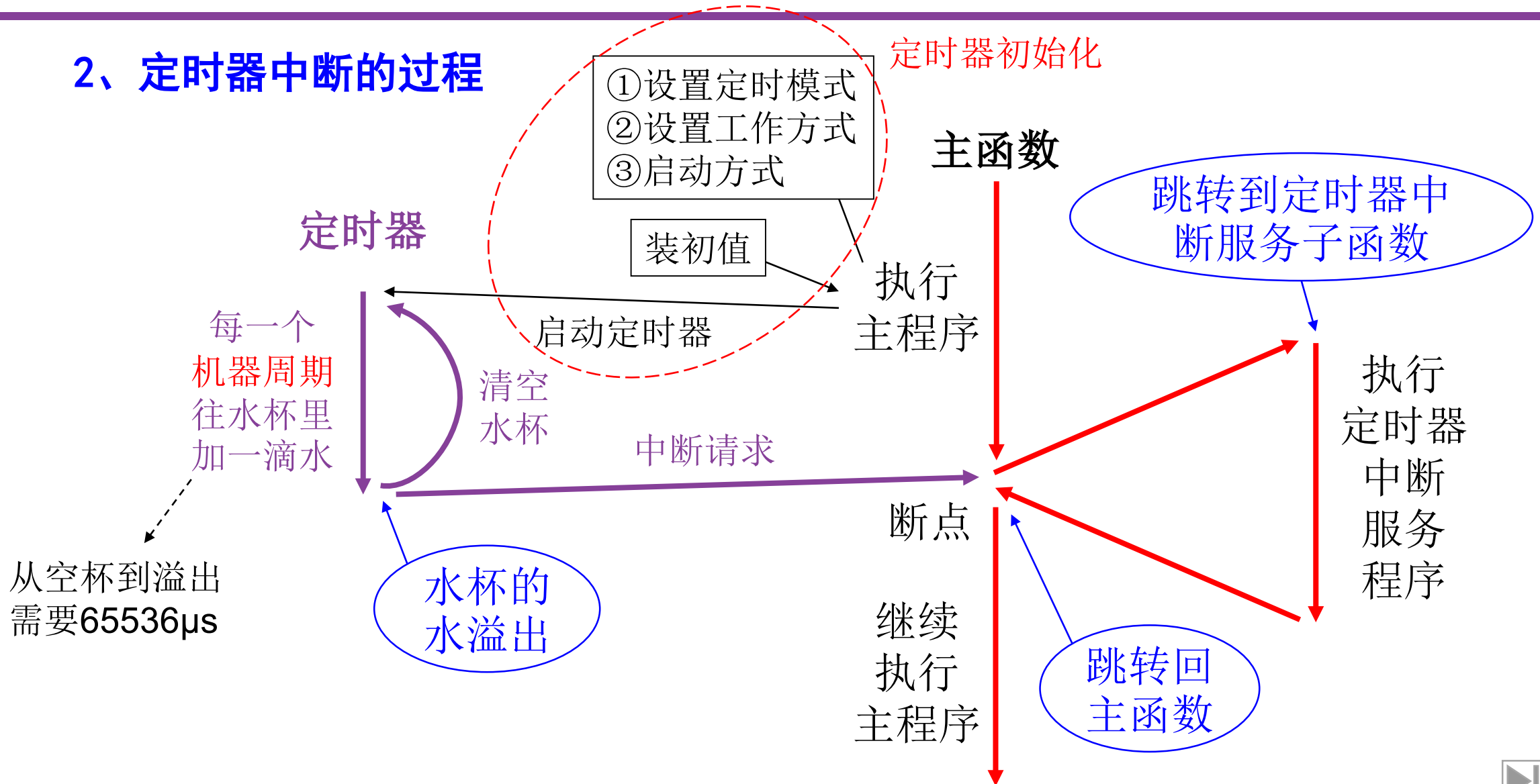
9.2 认识定时/计数器中断

2、定时器中断的过程



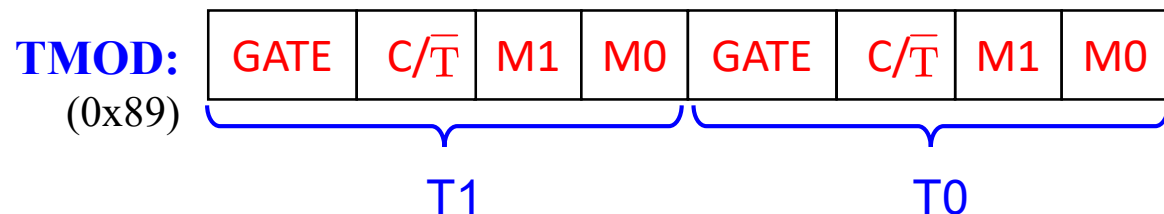
9.2 认识定时/计数器中断

2、定时器中断的过程



9.3 定时器中断的控制

1、控制定时器中断的管理员



- ①设置定时模式
- ②设置工作方式
- ③启动方式

TMOD=0x01

- C/ $\bar{T}$: 定时模式和计数模式 选择位
 - C/ $\bar{T}$ =0: 定时器模式
 - C/ $\bar{T}$ =1: 计数器模式
- M1、M0: 工作方式 选择位
- GATE: 门控制位
 - GATE=0: 启动/停止定时器只由TCON寄存器中的TR0/TR1来控制
 - GATE=1: 启动与停止定时器由TCON寄存器中的TR0/TR1和外部中断引脚上的电平来共同控制

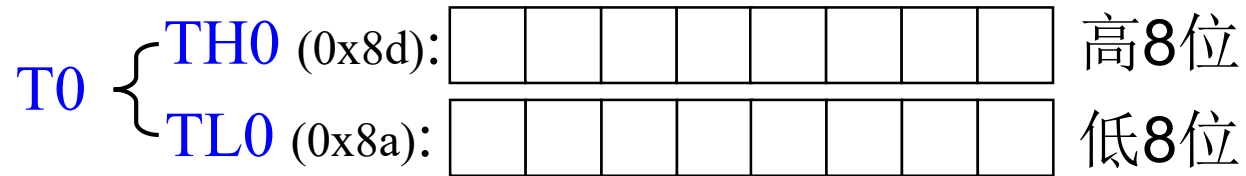
M1、M0	工作方式	说明
00	方式0	13位定时/计数器
01	方式1	16位定时/计数器
10	方式2	两个8位定时/计数器，初值自动装入
11	方式3	两个8位定时/计数器，仅适用于T0



9.3 定时器中断的控制

给定时器装初值

1、控制定时器中断的管理员



$$TH0 = (65536 - X) / 256;$$

$$TL0 = (65536 - X) \% 256;$$

- 假设要先装15536滴水（初值）：
 - $TH0 = 15536 / 256$; （取商）
 - $TL0 = 15536 \% 256$; （取余）
- 假设要定时的值为X:
- 给定时器装的初值为 $65536 - X$
 - $TH0 = (65536 - X) / 256$;
 - $TL0 = (65536 - X) \% 256$;



9.3 定时器中断的控制

1、控制定时器中断的管理员

IE:
(0xa8)

EA	-	-	ES	ET1	EX1	ET0	EX0
----	---	---	----	-----	-----	-----	-----

TCON:
(0x88)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

- EA: 总中断 允许位
- ET0: 定时/计数器0 允许位
- ET1: 定时/计数器1 允许位
- TR0: 定时器0运行 控制位（启动/停止）
- TR1: 定时器1运行 控制位（启动/停止）
- TF0: 定时器0溢出 标志位
- TF1: 定时器1溢出 标志位

- ①打开总中断
②打开定时器中断
③启动定时器

启动定时器

EA=1;

ET0=1;

TR0=1;



9.3 定时器中断的控制

2、设置定时器中断的步骤（以定时器T0为例）：

1. 硬件(接线)：单片机最小系统即可
2. 软件(编程)：

① 定时器初始化：

a) 设置定时器0为定时模式、工作方式1：TMOD=0x01

b) 设置定时器0初值：TH0=(65536-?)/256、
TL0=(65536-?)%256 } 装初值

c) 打开总中断：EA=1

d) 打开定时器0中断：ET0=1

e) 启动定时器0：TR0=1

- ①设置定时模式
- ②设置工作方式
- ③启动方式

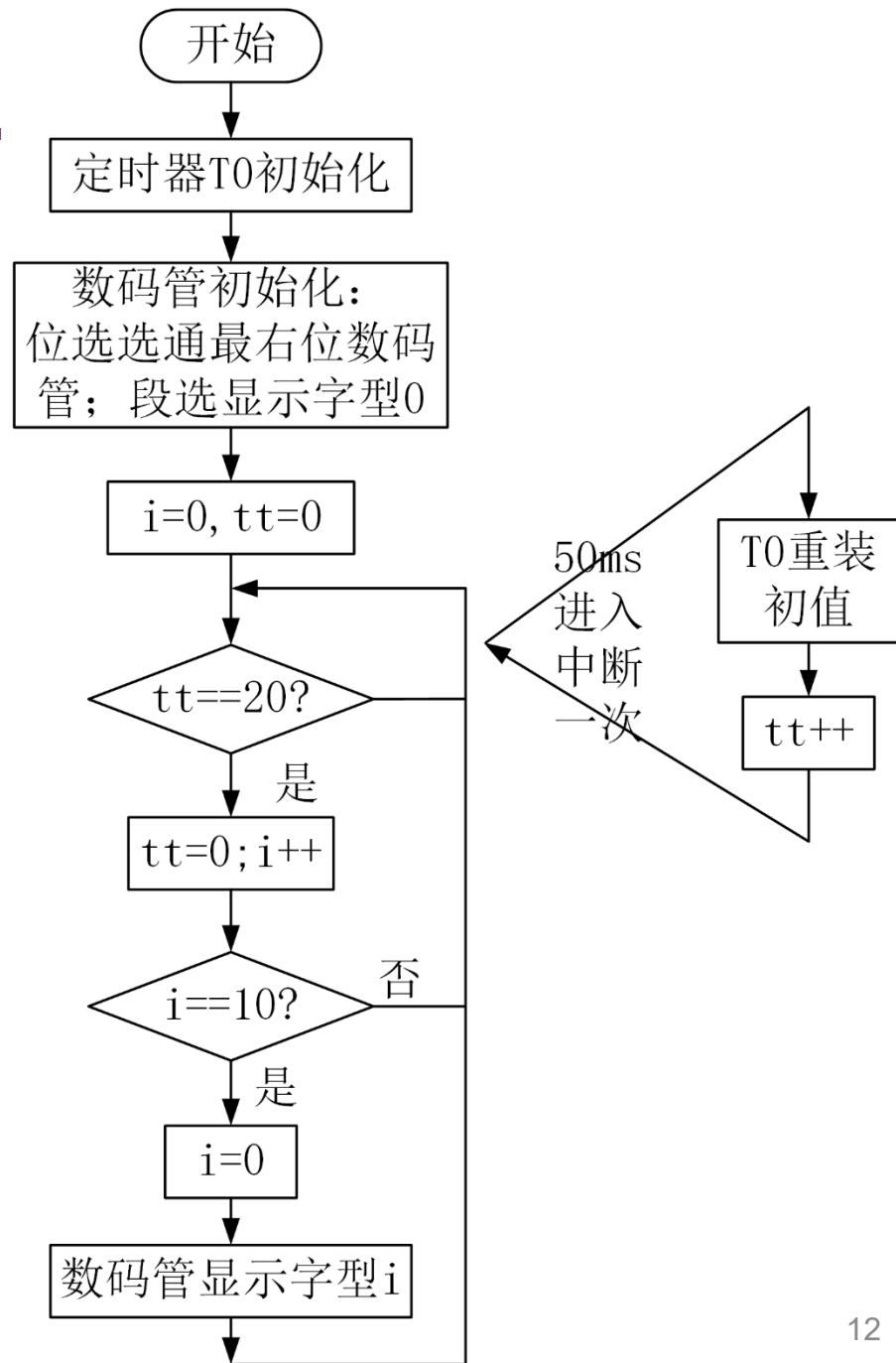
② 定时器0中断服务程序（内部代码别写太长）：

```
void xxx() interrupt 1  
{.....}
```

9.4 实战任务

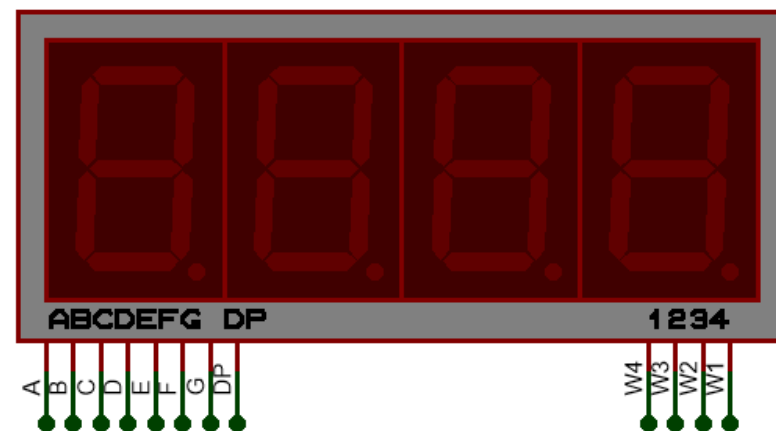
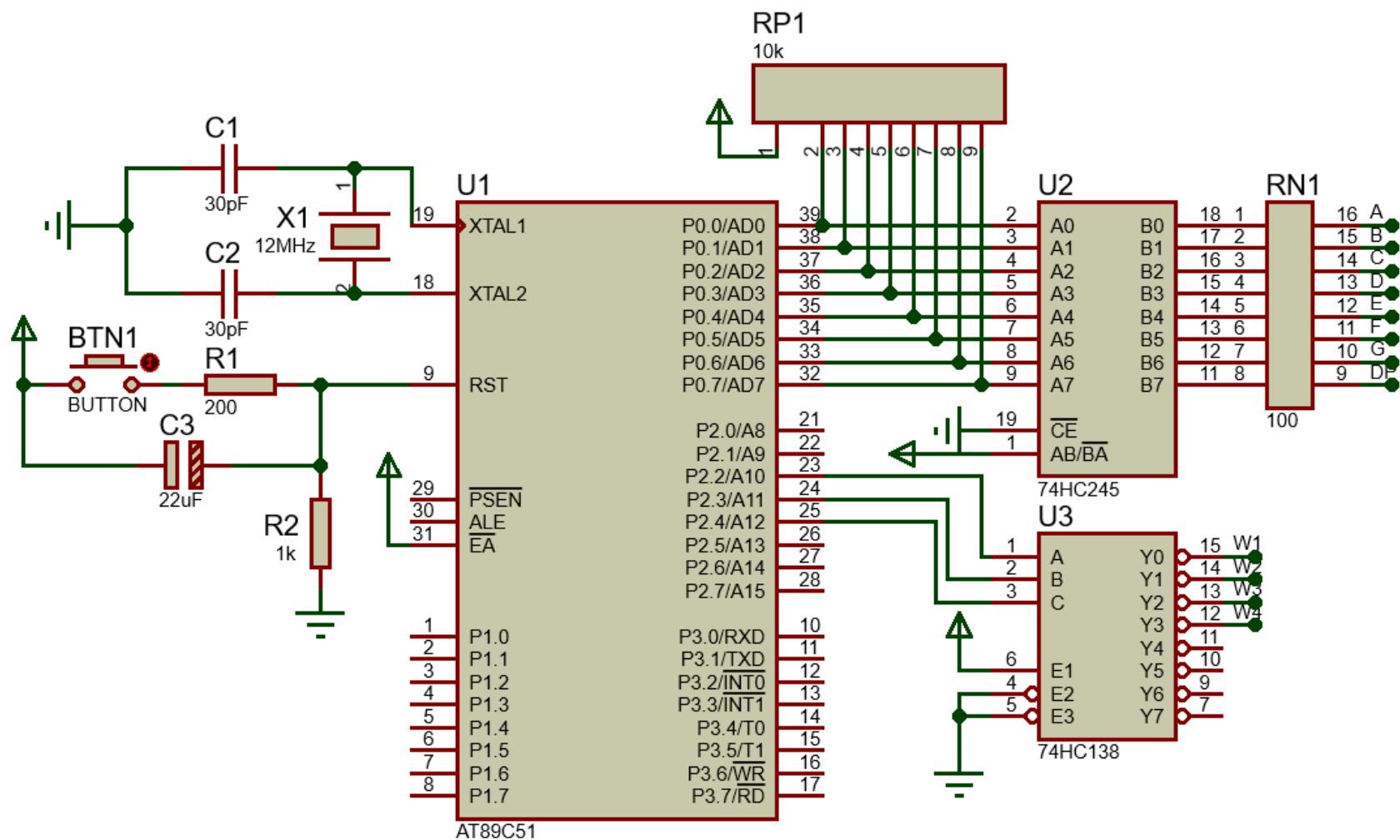
数码管间隔1s循环显示0~9（硬件延时）

- 任务要求：用定时器中断让共阴极4位数码管的最右位数码管间隔1秒循环显示0~9
- 任务分析：
 - ① 定时器定时1s后进入中断子函数，对数码管进行加1显示
 - ② 题目要求定时1s（ $1 \times 10^6 \mu s$ ），而定时器每次最多只能定时 $65536 \mu s$ ，要进行多少次定时不好算。
 - ③ 因此，每次定时50ms（ $50000 \mu s$ ），定时20次，就刚好1s



9.4 实战任务

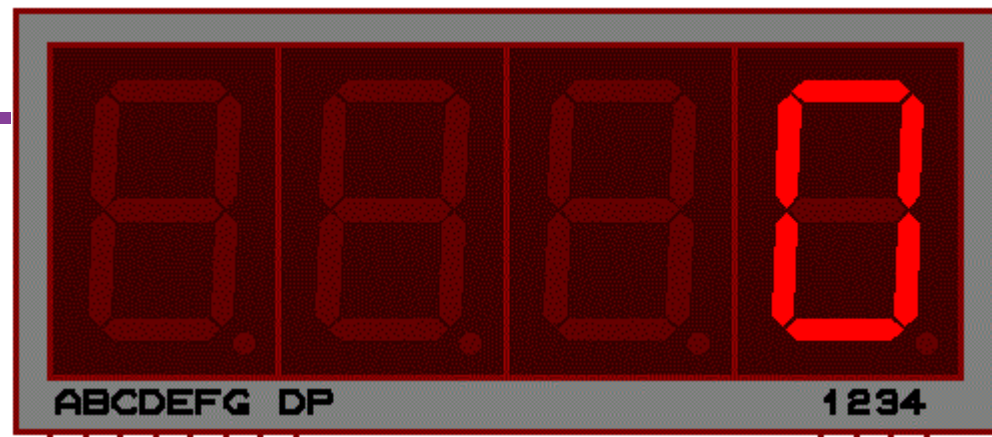
仿真步骤1：画电路图（与开发板相同）



9.4 实战任务

仿真步骤2：编程 & 仿真

```
#include <regx51.h>
#define uint unsigned int
#define uchar unsigned char
//共阳极数码管的字型码表
uchar code table[]={0x3f,0x06,0x5b,
0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,
0x77,0x7c,0x39,0x5e,0x79,0x71};
//定义全局变量，tt为进入定时器0中断的次数
uchar tt=0;
//中断法
void main(void)
{
    uchar num=0;
    //定时器0的初始化
    TMOD=0x01;                //设置定时器0为工作方式1
    TH0=(65536-50000)/256;    //装初值，取模
    TL0=(65536-50000)%256;    //装初值，取余
    EA=1;                    //开总中断
    ET0=1;                  //开定时器0中断
    TR0=1;                  //启动定时器0，开始定时
    //数码管初始化
    P2_2=0;P2_3=0;P2_4=0;    //位选：选通最右边数码管
    P0=table[0];            //段选：先显示“0”
```

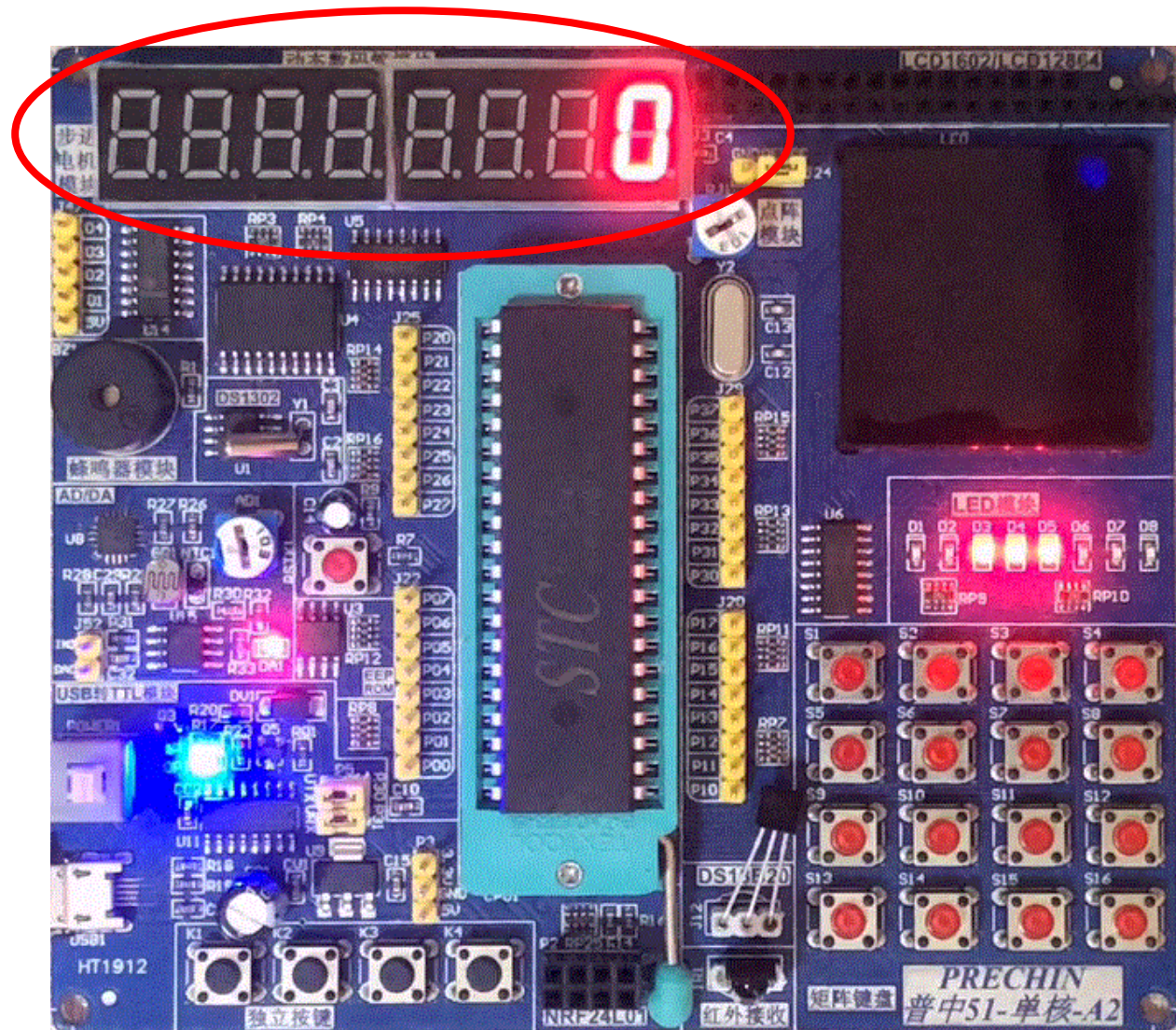


```
while(1)
{
    if(tt==20)
    {
        tt=0;
        num++;
        if(num==10)
            num=0;
        P0=table[num];
    }
}
//定时器0子函数：重装初值，50ms
void timer0() interrupt 1
{
    TH0=(65536-50000)/256;
    TL0=(65536-50000)%256;
    tt++;        //进入定时器0中断的次数加一
}
```

9.4 实战任务

实际开发板练习

- 由于仿真练习的电路图设计与实际开发板原理图相同，故直接将用于仿真的程序直接下载到实际开发板，观察实际效果如右图。



9.5 实战作业

1. 用定时器中断做8个LED流水灯，间隔0.5s。
2. 用两个定时器中断控制蜂鸣器发声：定时器中断0控制频率，定时器中断1控制同个频率持续的时间，间隔300ms依次输出1、10、50、100、200、400、800、1k（Hz）的方波。