



合肥工业大学
HEFEI UNIVERSITY OF TECHNOLOGY



计算机与信息学院
SCHOOL OF COMPUTER SCIENCE AND INFORMATION ENGINEERING
人工智能学院
SCHOOL OF ARTIFICIAL INTELLIGENCE



中断系统

第四次培训：2021-10-31

电子设计创新实验室 周布伟

内容介绍

CONTENTS

一 中断系统的基本概念

二 51单片机的中断系统

三 外部中断的实现

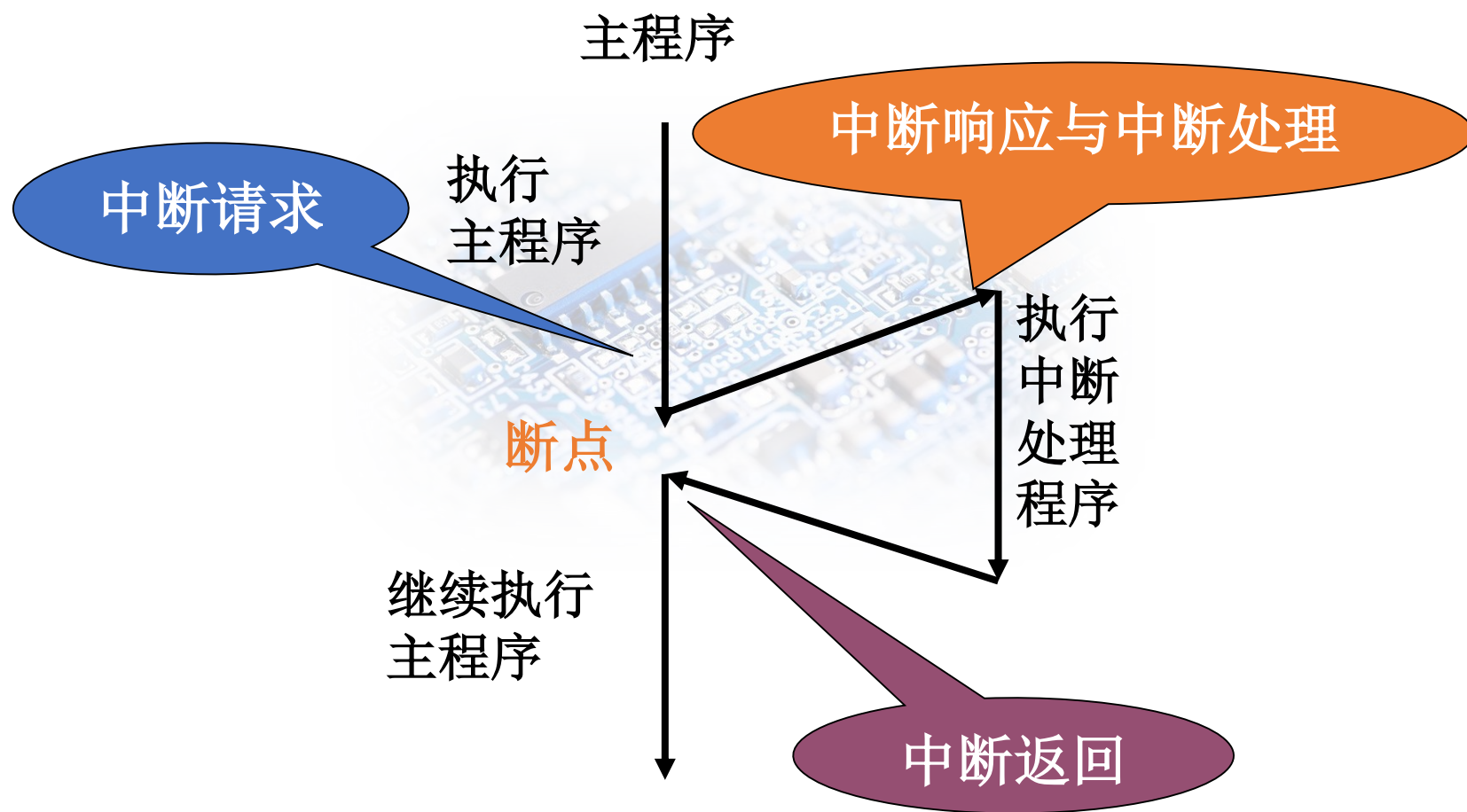
— 中断系统的 基本概念

➤ 中断的引入

```
void main()
{
    LED = 0xff;
    while (1)
    {
        if (key == 0)
        {
            Delay5ms();
            if (key == 0)
            {
                LED = ~LED;
                while(key == 0);
            }
        }
    }
}
```



➤ 中断的定义及相关术语

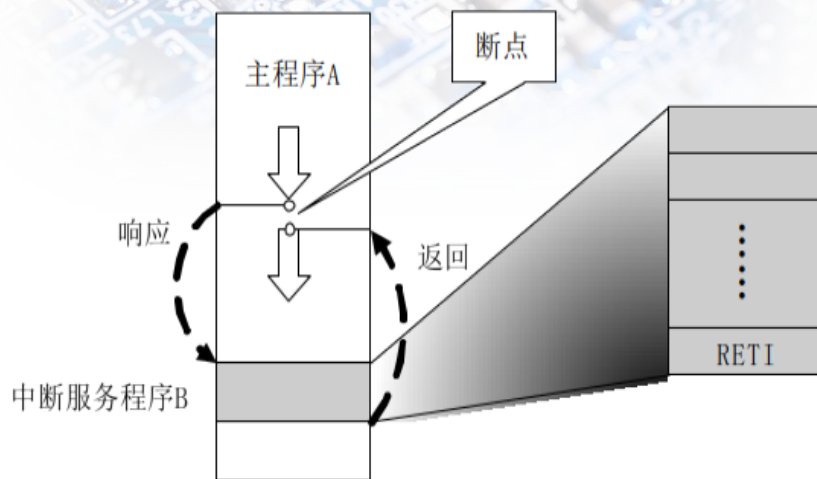


➤ 中断的定义及相关术语

- 计算机(单片机)执行某程序时, 发生了**紧急事件或有特殊请求**, CPU**暂停**某程序的执行, **转而去处理**上述事件或请求, 处理完毕后再**重新执行**某程序的过程叫做中断。
- 具体过程:
 1. CPU在处理某一事件A时, **发生了**另一事件B请求CPU迅速去处理(中断发生);
 2. CPU**暂时中断当前的工作**, 转去处理事件B(中断响应和中断服务);
 3. 待CPU将事件B处理完毕后, 再**回到原来事件A被中断的地方**继续处理事件A(中断返回)。

➤ 中断的定义及相关术语

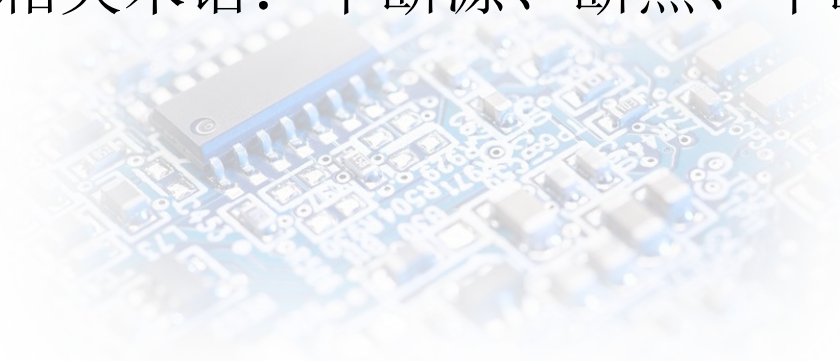
- 引起CPU中断的根源，称为中断源。
- 中断源向CPU提出的中断请求。CPU暂时中断原来的事务A，转去处理事件B。对事件B处理完毕后，再回到原来被中断的地方（即断点），称为中断返回。
- 实现上述中断功能的部件称为中断系统（中断机构）。



▲ 中断在单片机中的实现

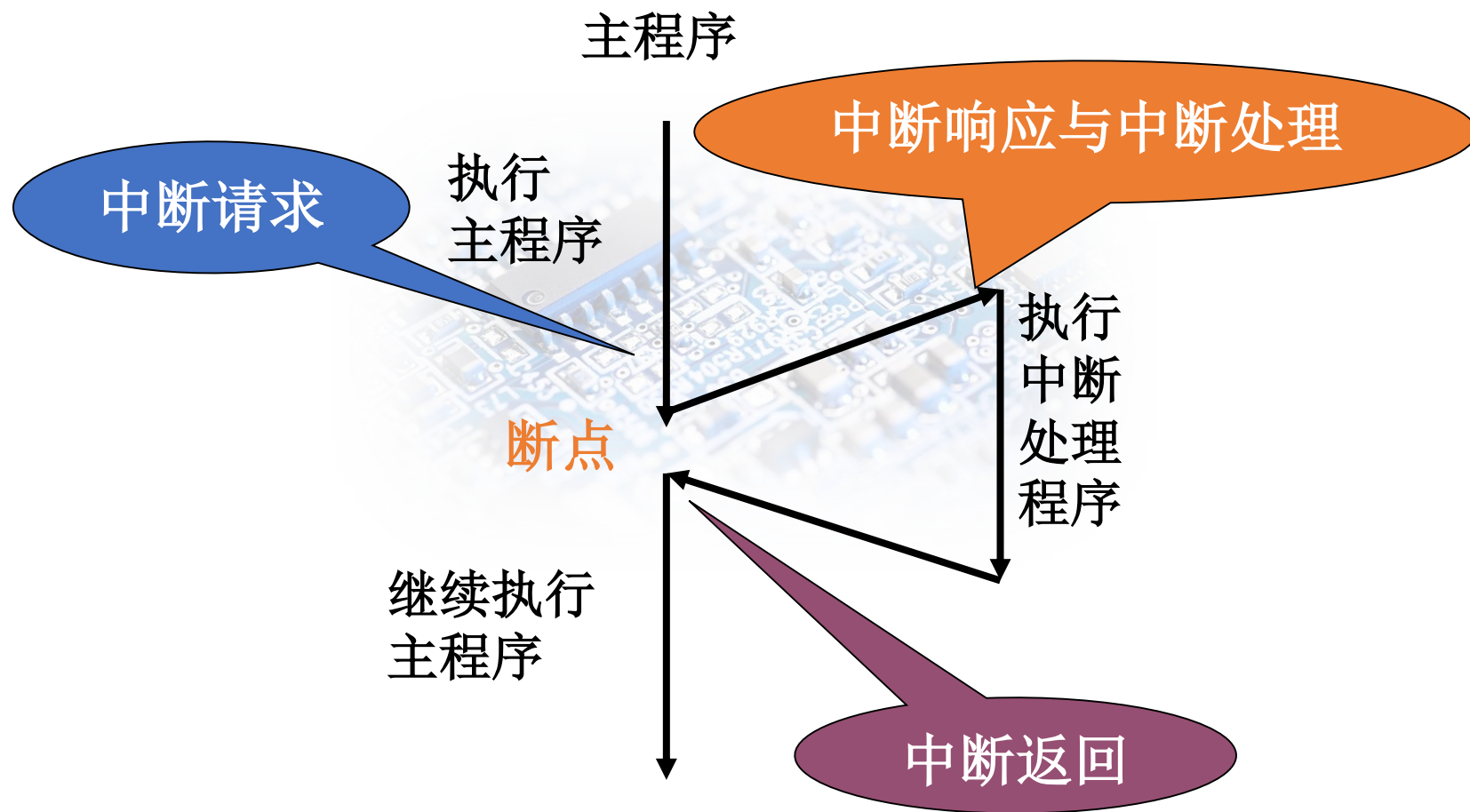
➤ 小结

- 引入中断的意义
- 中断的过程：中断请求、中断响应与中断处理、中断返回
- 中断的定义及相关术语：中断源、断点、中断系统

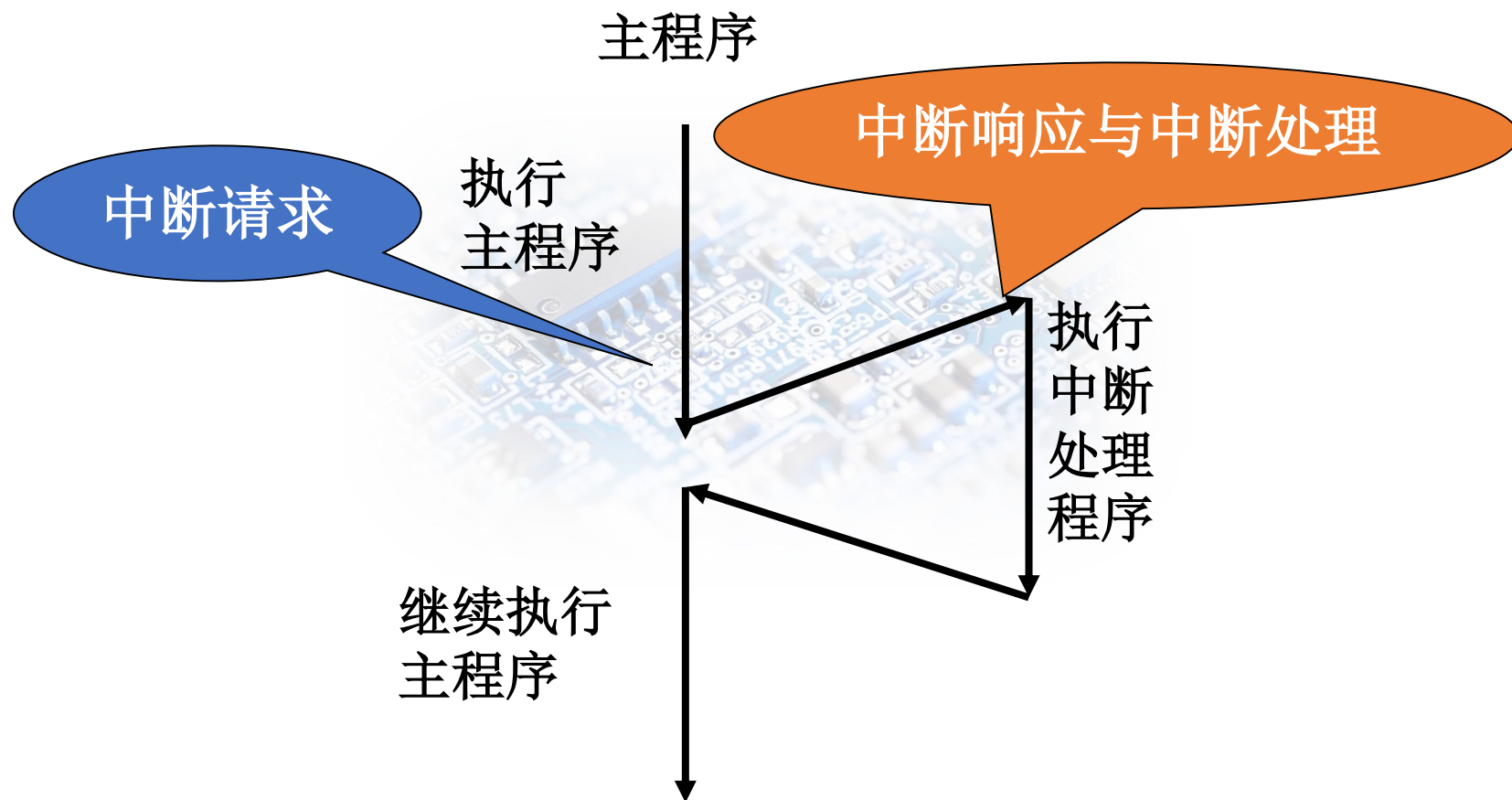


二 51单片机的 中断系统

➤ 51单片机如何实现中断？



➤ 51单片机如何实现中断？



➤ 51单片机如何实现中断?

• P3各引脚**第二功能**定义:

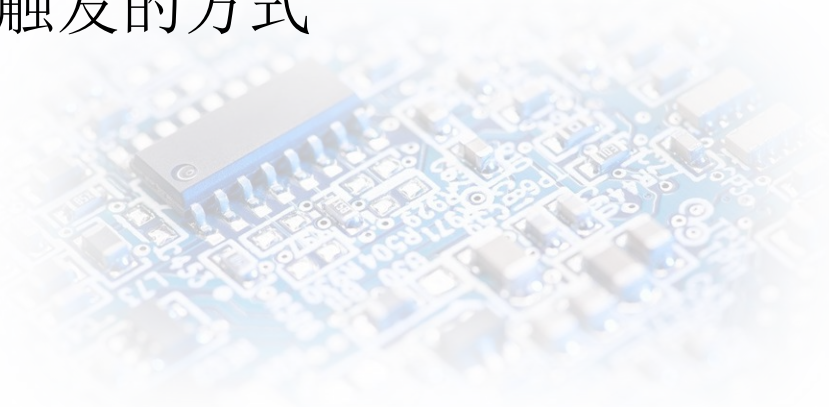
- ◆ P3.0: RXD串行口输入
- ◆ P3.1: TXD串行口输出
- ◆ **P3.2**: INT0外部中断0输入
- ◆ **P3.3**: INT1外部中断1输入
- ◆ P3.4: T0定时器0外部输入
- ◆ P3.5: T1定时器1外部输入
- ◆ P3.6: WR外部写控制
- ◆ P3.7: RD外部读控制

P3.0/RXD	10
P3.1/TXD	11
P3.2/ $\overline{\text{INT0}}$	12
P3.3/ $\overline{\text{INT1}}$	13
P3.4/T0	14
P3.5/T1	15
P3.6/ $\overline{\text{WR}}$	16
P3.7/ $\overline{\text{RD}}$	17

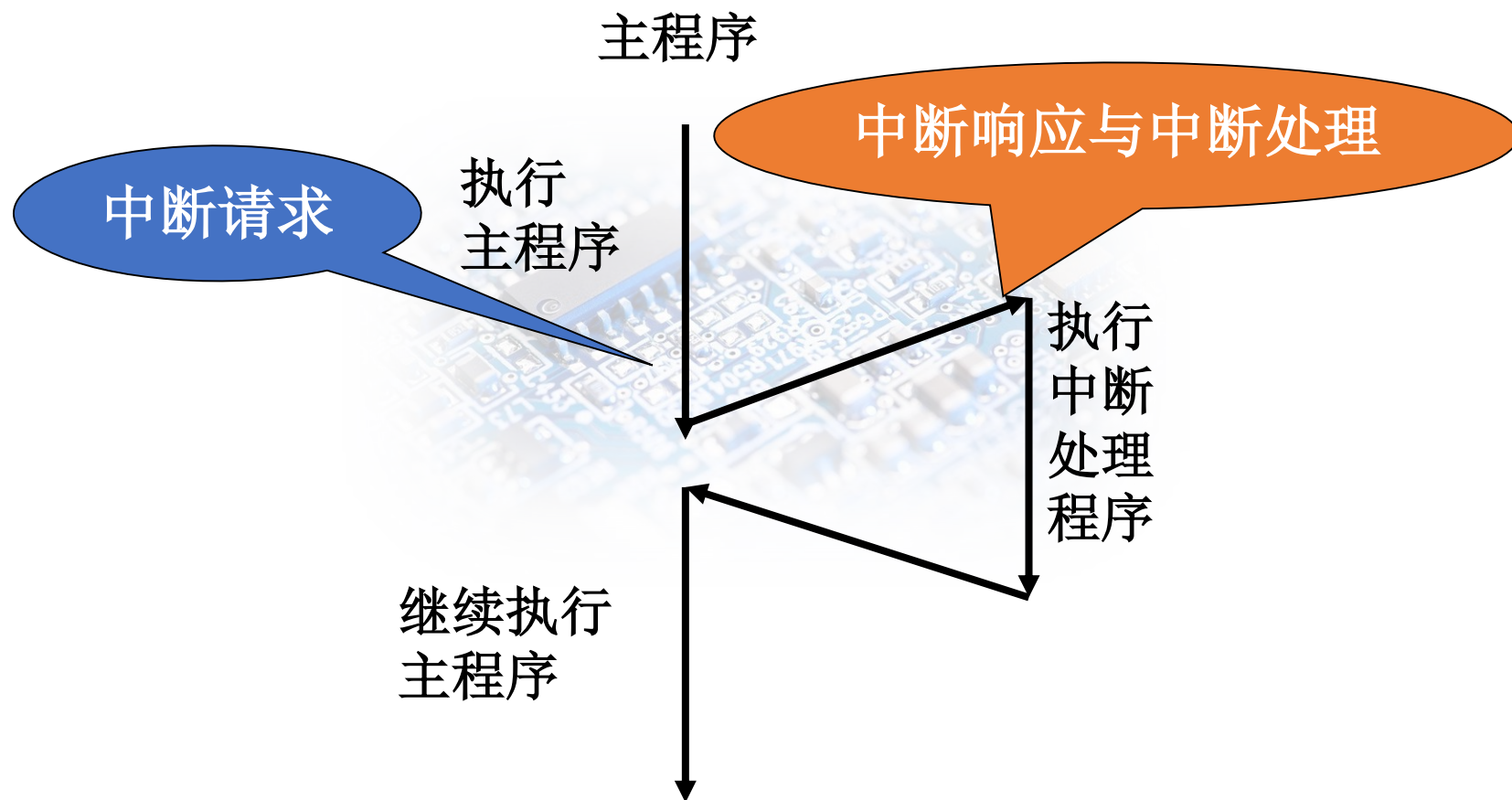
▲ 51单片机的P3引脚

➤ 51单片机如何实现中断？

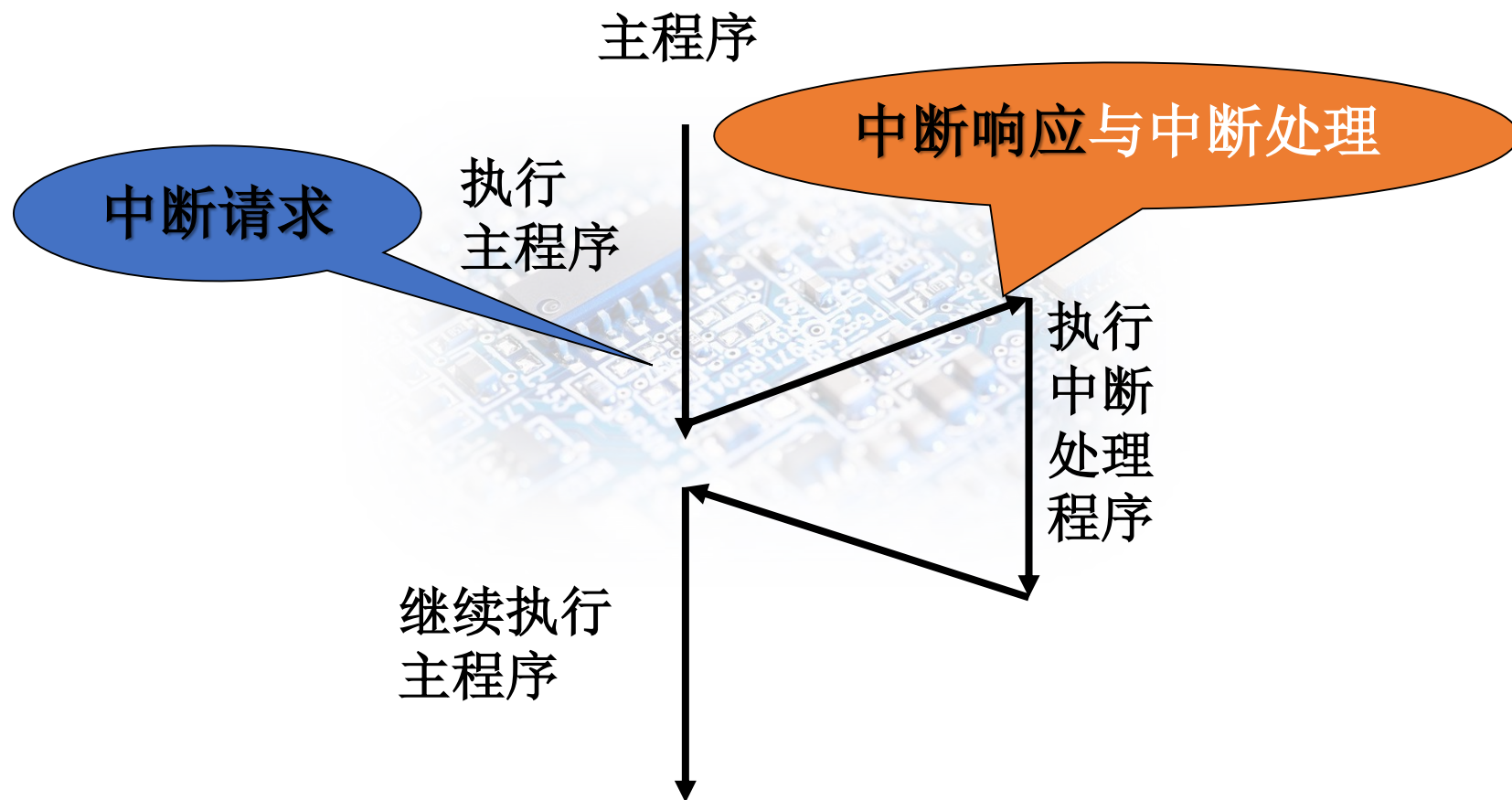
- 中断初始化：
 1. 选择中断的类型：外部中断0/1、定时器0/1中断
 2. 配置中断触发的方式
 3. 打开中断
- 中断处理



➤ 51单片机如何实现中断？



➤ 51单片机如何实现中断？



➤ 配置中断方式：控制寄存器TCON

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

- TF1/TF0：定时器1/定时器0溢出标志位。
- TR1/TR0：定时器1/定时器0运行控制位。
- IE1/IE0：外部中断1/外部中断0请求标志位。
- IT1/IT0：外部中断1/外部中断0触发方式选择位：
 - 当IT1/IT0=0，为低电平触发方式；
 - 当IT1/IT0=1，为下降沿触发方式。

➤ 允许中断：控制中断允许寄存器IE

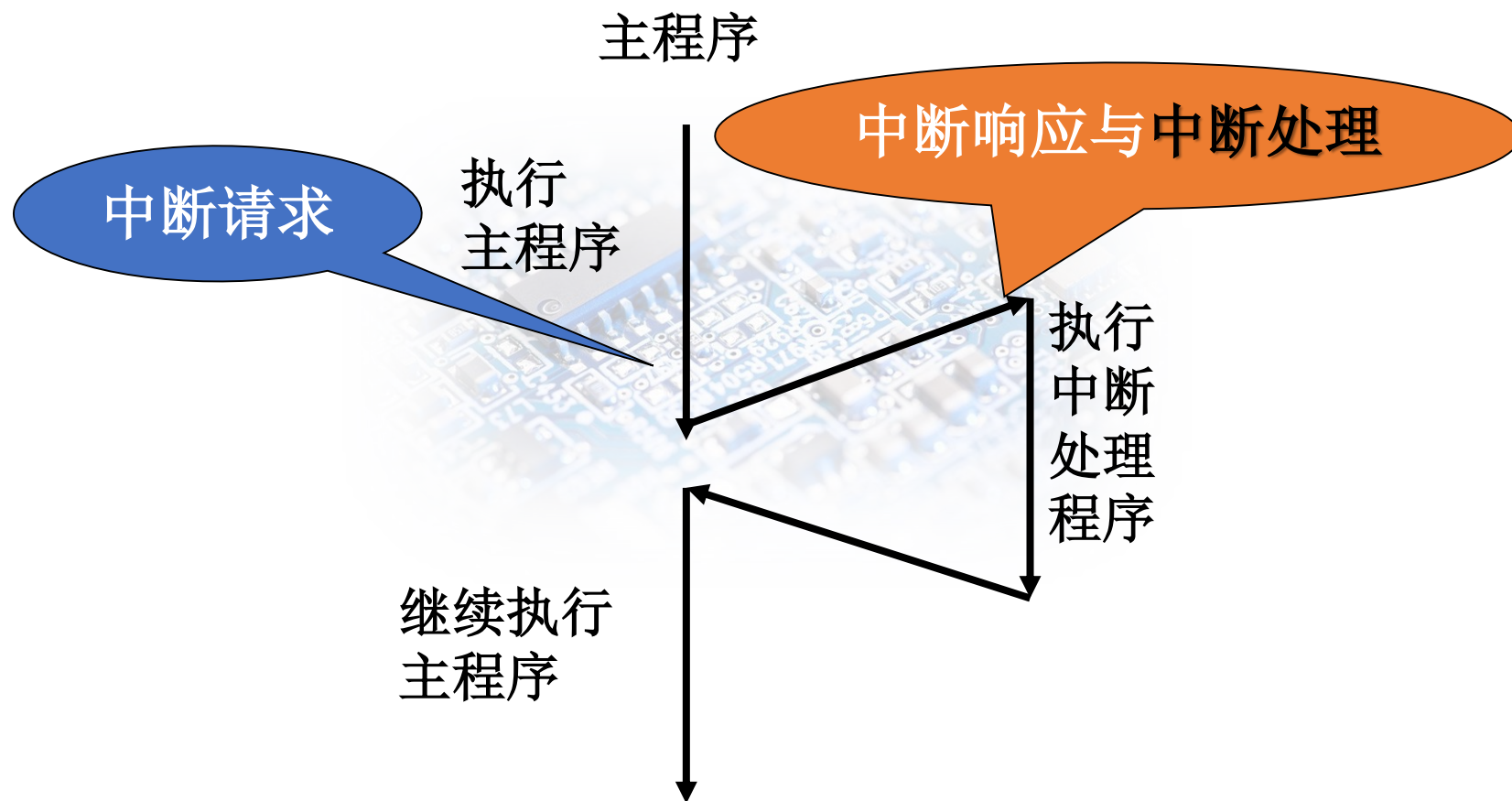
B7	B6	B5	B4	B3	B2	B1	B0
EA	/	/(ET2)	ES	ET1	EX1	ET0	EX0

- EA：全局中断允许位 (=1，打开；=0，关闭，下同)
- ET2：定时器/计数器2中断允许位
- ES：串行口中断允许位
- ET1：定时器/计数器1中断允许位
- EX1：外部中断1中断允许位
- ET0：定时器/计数器0中断允许位
- EX0：外部中断0中断允许位

➤ 代码实现：中断初始化

```
/*主函数*/  
void main()  
{  
    IT0 = 1;           //跳变沿触发方式（下降沿）  
    EX0 = 1;           //打开 INT0 的中断允许  
    EA = 1;            //打开总中断  
    while(1);  
}
```


➤ 51单片机如何实现中断？



➤ 代码实现：中断处理

```
/*外部中断0处理函数*/  
void Int0() interrupt 0  
{  
    //...  
}
```

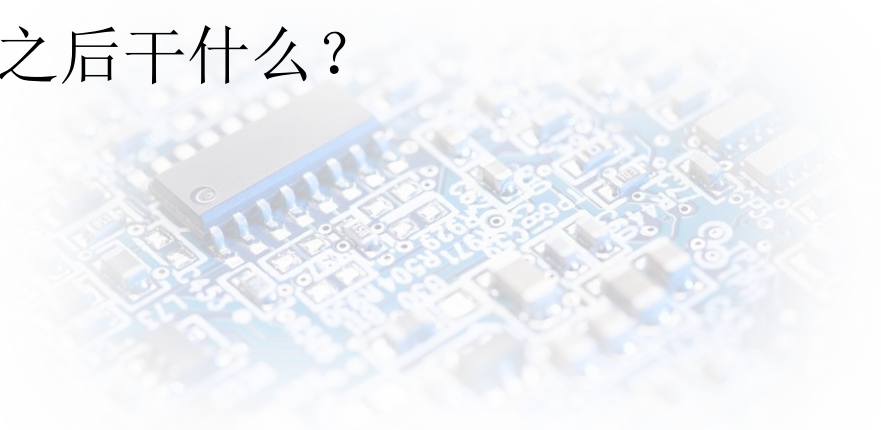


➤ 中断编号

中断源	中断编号
外部中断0	0
T0溢出中断	1
外部中断1	2
T1溢出中断	3
串行口中断	4

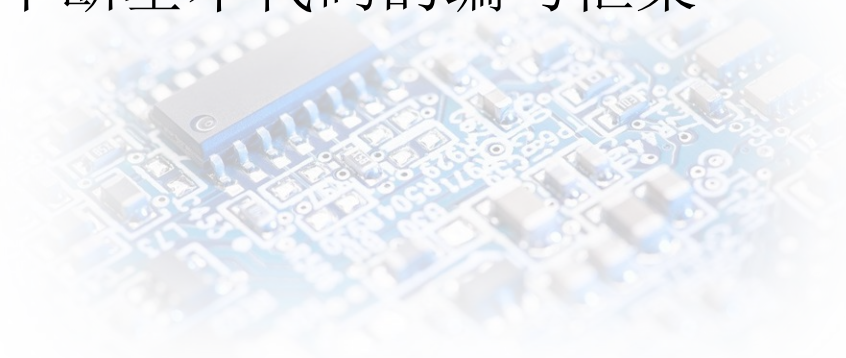
➤ 代码实现

- 你想使用的中断是哪个？选择相应的中断号；
- 你所希望的触发条件是什么？
- 你希望在中断之后干什么？



➤ 小结

- 51单片机中断的实现方式：外部中断、定时器中断
- 51单片机配置外部中断相关的寄存器
- 51单片机外部中断基本代码的编写框架



三 外部中断 的实现

➤ 函数的基本概念与结构

```
#include <reg52.h>
```

```
int Delay_ms(unsigned int x)
```

```
{
```

```
    // delay_1ms * x
```

```
    return 0;
```

```
}
```

函数名（用标识符命名），
引用函数的标志，得到结果。

函数参数说明

函数的类型，返回值的类型

```
void main(void)
```

```
{
```

```
    Delay_ms(5);
```

```
    while (1);
```

```
}
```

➤ 函数的基本概念与结构

- 函数定义时要确定如下四点：

函数的名称、函数的类型、函数的参数、函数的功能

- 关于返回值的几点说明：

1. 函数可以通过一个return语句返回一个值，也可以不返回
返回值，此时应在定义函数时用void类型加以说明；
2. 函数中可以出现多个return语句，遇到一个return 语句，则返回值，且返回调用函数，继续执行；
3. 为了确保参数和返回值类型正确，一般须在函数调用前对其类型和参数的类型加以说明，该说明称之为原型声明。

➤ 代码实现

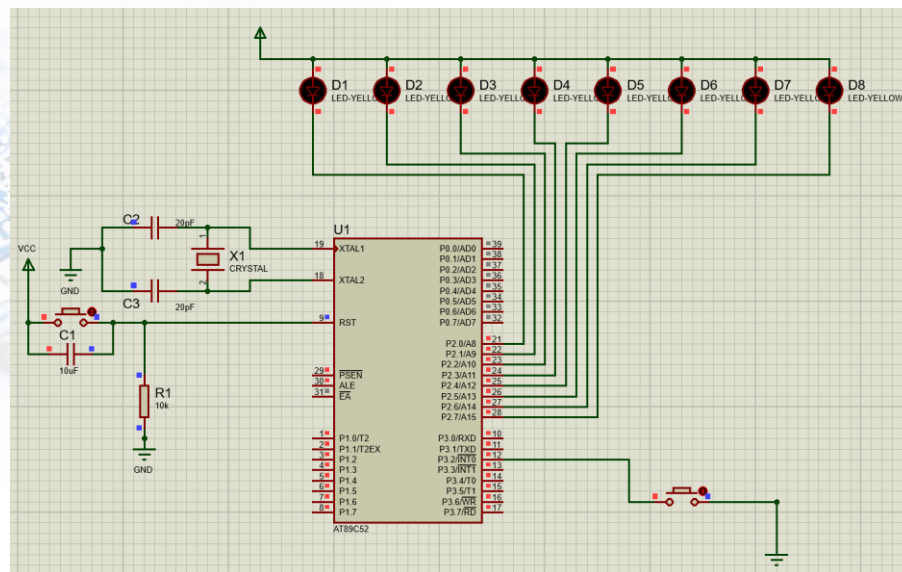
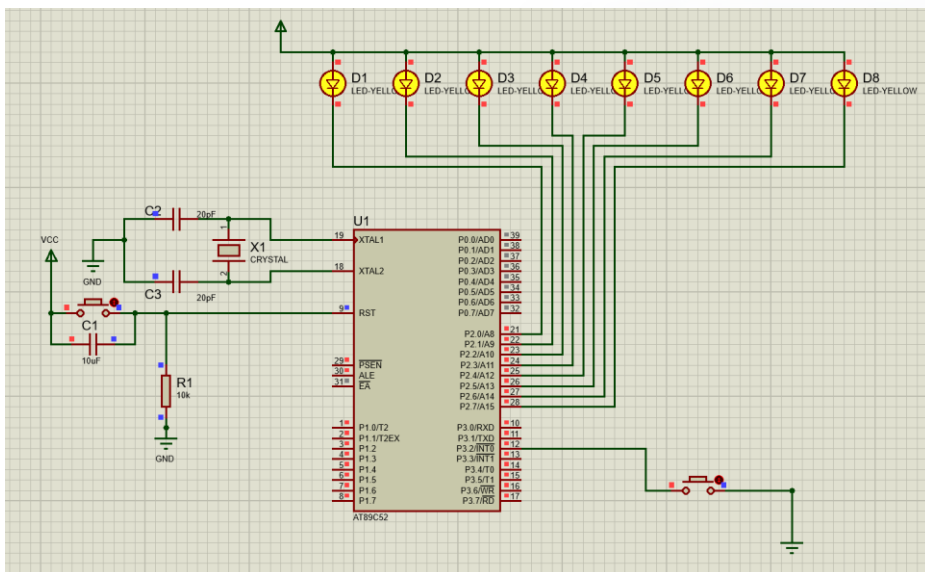
```
/*外部中断0初始化函数*/
void Int0Init()
{
    IT0 = 1;           //跳变沿触发方式（下降沿）
    EX0 = 1;           //打开 INT0 的中断允许
    EA = 1;            //打开总中断
}

/*主函数*/
void main()
{
    Int0Init();
    while(1);
}

/*外部中断0处理函数*/
void Int0() interrupt 0
{
    //...
}
```

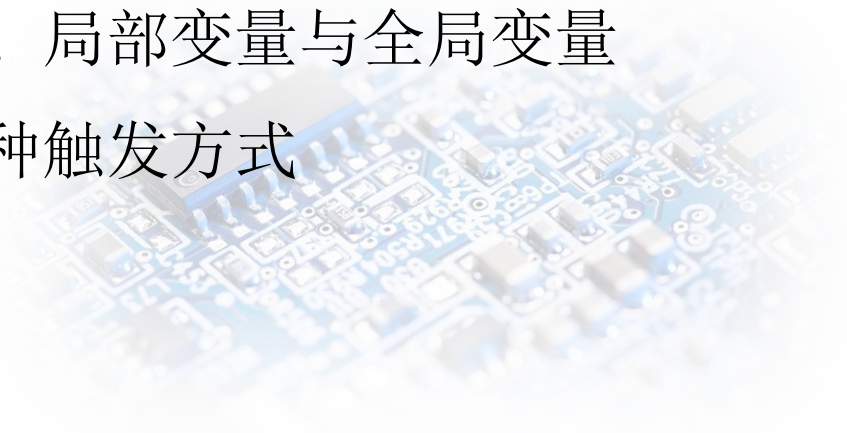
➤ 课堂例题

- 通过一个独立按键实现外部中断0控制8位LED小灯的同时间隔500MS的闪烁开始与停止。



▲LED闪烁

➤ 小结

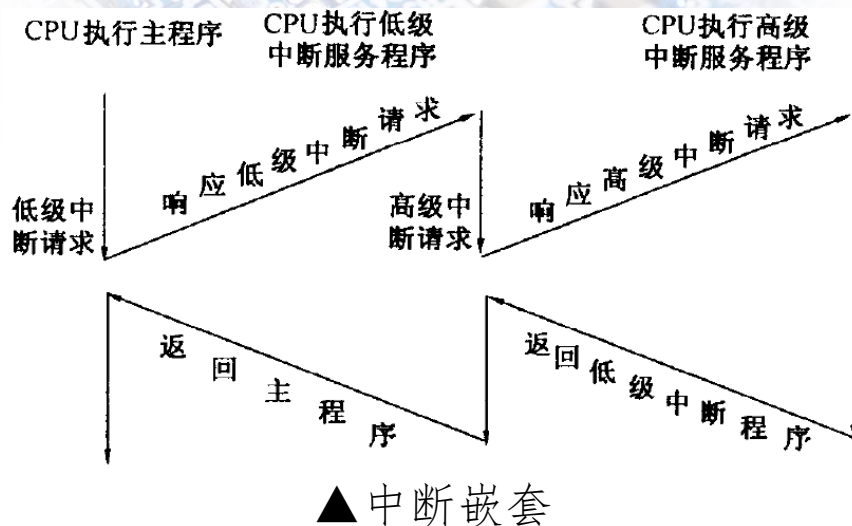
- 中断系统代码编写的基本框架与流程
 - 代码的注释与调试
 - 函数及其参数、局部变量与全局变量
 - 外部中断的两种触发方式
- 

➤ 中断优先级

中断源	中断入口	中断级别
外部中断0	0	最高
T0溢出中断	1	↓
外部中断1	2	
T1溢出中断	3	
串行口中断	4	
		最低

➤ 中断优先级

- 对同时发生多个中断申请时：
 - 不同优先级的中断同时申请(很难遇到)：先高后低
 - 相同优先级的中断同时申请(很难遇到)：按序执行
 - 正处理低优先级中断又接到高级别中断：高打断低
 - 正处理高优先级中断又接到低级别中断：高不理低



➤ 中断与计算机

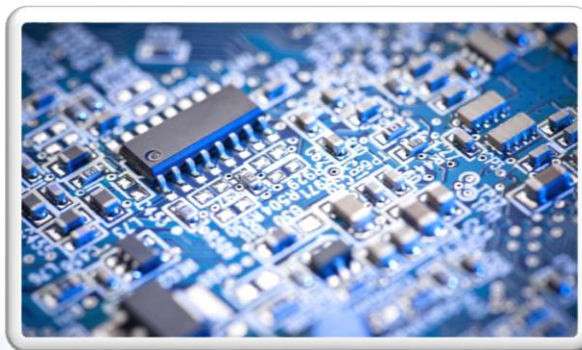
- 数据传送的双方平时各自做自己的工作，一旦甲方要求与乙方进行数据传送，就主动发出信号提出申请，乙方接到申请后若同意传送，安排好当前的工作，再响应与甲方发生数据传送。完事后，回去继续做打断前的工作。
- 中断功能强弱是计算机性能优劣的重要标志：
 - 提高CPU效率
 - 解决速度矛盾
 - 实现并行工作
 - 应付突发事件

➤ 定时器与定时器中断

- 软件延时的不足或坏处？
 - 准确延时
 - 同时控制多个外设
- 如何配置定时器中断？

B7	B6	B5	B4	B3	B2	B1	B0
EA	/	/(ET2)	ES	ET1	EX1	ET0	EX0

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



感谢倾听
The End

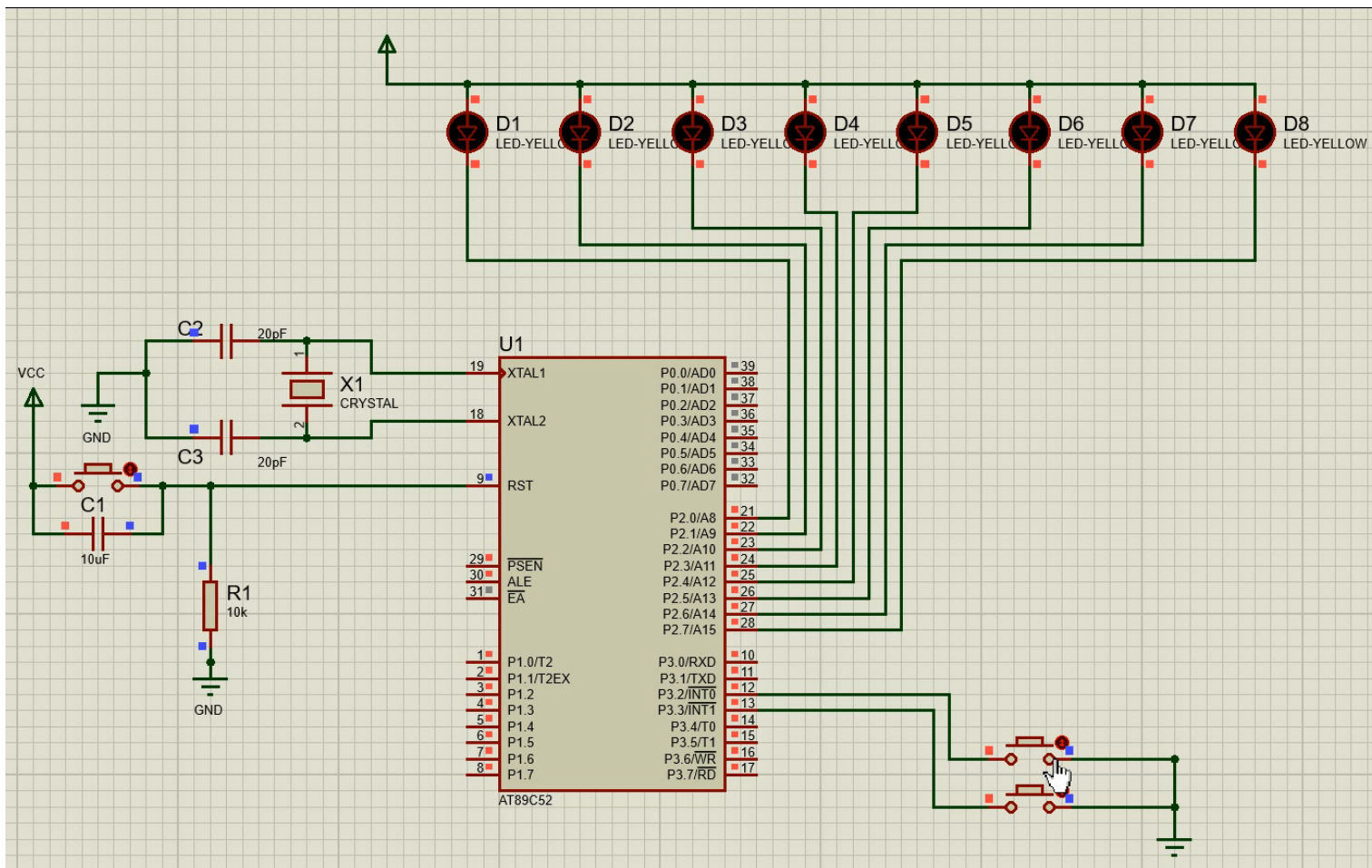
➤ 中断系统的实现 (100分)

1. (40分+10分) 回顾以往培训的作业或其他课外实验，选择其中至少一个程序使用外部中断实现，并选择一个程序绘制流程图，视数量和难度酌情加分。
2. (50分) 通过两个独立按键实现外部中断0和外部中断1分别控制8位LED中某一位小灯的闪烁与移位，并对当前功能可能存在的不合理的地方进行适当改进，代码编写符合规范且须有适当的注释。
3. (10分) 简答题：
 - ①(4分) 谈谈你对中断的认识？
 - ②(6分) 结合自身代码的调试过程，谈谈在51单片机实现外部中断时需要注意什么？
4. (+20分) 预习或提前学习并实现定时器中断：
 - ①(2分) 51单片机定时器中断有哪几种实现方式？
 - ②(3分) 实现51单片机定时器中断之前需要配置哪些寄存器？
 - ③(15分) 编程实现定时器控制LED按照0.5秒时间间隔闪烁。

➤ 中断系统的实现-说明与要求:

1. 第一题需要在程序文件(.c)开头通过注释介绍实现的功能;
2. 本次作业11月6日(星期六) 23:59截止, 作业以“**学号-姓名-第四次作业**”格式命名发送至413732041@qq.com, 邮件标题与文件名相一致;
3. 根据**题号**创建文件夹, 并统一打包, 主要包含程序文件(.c) 运行文件(.hex) 仿真文件(.pdsprj)、流程图(.pdf或图片格式)、运行视频、简答题(.doc/.docx/.txt/.pdf/图片格式)等, 有实物的同学不需要提交仿真文件;
4. 视频要求清晰呈现实验现象, 最好压缩到**10M以内**;
5. 流程图可以**手绘或软件绘制**, 具体规范查阅相关资料。

➤ 中断系统的实现-说明与要求:



▲ 第二题效果演示

➤ 中断系统的实现-说明与要求:

➤ 2021217100-周百威-第四次作业

名称

自行命名代替

➤ 1-外部中断实现1602清屏与流程图

➤ 2-LED控制

➤ 3-简答题

➤ 4-预习 (选做)

▲第四次培训作业文件命名规范(供参考)

➤ 中断系统的实现-评分细则

1. (40分+10分) 外部中断实现与流程图：
 - ① (25分) 外部中断实现；
 - ② (15分) 流程图：无并行，有大循环，有中断，符合规范。
2. (50分) LED控制：
 - ① (10分) 注释与编写规范；
 - ② (25分) 正确实现LED的闪烁与移位控制；
 - ③ (15分) 改正不合理的地方。
3. (10分) 简答题：字数不限，能充分表达意思即可
4. (+20分) 预习

若出现雷同的情况，视情节轻则扣分，重则此次作业0分处理。

➤ 参考资料

1. 电子设计创新实验室. 中断系统[EB/OL]. [2020-11-29].
2. 电子设计创新实验室. 第五次培训[EB/OL]. [2019-12-07].
3. 宣善立. 第六章_函数[EB/OL]. [2020-11].
4. 普中科技. 中断系统[EB/OL]. [2019-09].
5. 普中科技. 外部中断[EB/OL]. [2019-09].