微机原理与接口技术

§9 中断系统

主讲人：佘青山

Homepage: https://faculty.hdu.edu.cn/zdhxy/sqs/main.htm
Email: qsshe@hdu.edu.cn
Mob：13758167196
Office: 第二教研楼南楼308室

2024年11月21日

生活中的“中断”例子

什么是中断？ 看书的例子：你正在看书，突然电话铃响了，……

中断就是正常的工作被外部事件打断了。

生活中的中断：

1、什么可以引起中断？生活中示例：有人按了门铃，电话铃响了，你的闹钟铃响了，水烧开了，内急了....等等诸如此类的事件。

把可以引起中断的事件称之为中断源。

2、中断的嵌套与优先级处理：

你正在看书，电话铃突然响了，同时又有人按了门铃，你该先做那样呢？

如果你正在等一个很重要的电话，你一般不会去理会门铃的，而反之，你正在等一个很重要的客人，则可能就不会去理会电话了。如果不是这两者（即不等电话，也不等人上门）你可能会按你的习惯去处理。总之存在优先级问题。

优先级问题不仅发生在两个或以上的中断同时产生的情况，也发生在一个中断已经产生，又有一个中断产生的情况：如你正在接电话，有人按门铃了，或你正在开门与人交谈，又有电话铃了。考虑一下：我们会怎么办？

生活中的“中断”例子

3、中断的响应过程：

- 看书的例子：当有事件产生，① 进入中断之前我们必须先记住现在看第几到页了，或拿一个书签放在当前页的位置，② 然后去处理不同的事情，③ 处理完了继续看书）。
- 接电话的例子：电话铃响了我们要到放电话的地方去，门铃响了，我们要到门那边去，也就是说，不同的中断，我们要在不同的地点处理，而这个地点通常还是固定的，这也和计算机中的中断类似。

4、计算机中的中断

- 计算机执行正常程序时，系统出现某些急需处理的异常情况和特殊请求，CPU暂时中止现在正在执行的的指令，转去对随机发生的更紧迫事件进行处理；处理完后，CPU会自动返回原来的程序继续执行。
- 就如：你正在家中看书，突然电话铃响了，你放下书本，去接电话，和来电话的人交谈，然后放下电话，回来继续看你的书。

中断系统是微控制器最基本最重要的功能模块之一。中断系统使得微控制器具有快速响应和处理突发事件及外设的信息交互请求，使微控制器具有处理多任务的能力，从而有效提高微控制器的实时测控性能。

本章内容分为**5个教学单元**，主要包括中断系统的基本概念，8051微控制器中断系统的组成结构和控制方式，中断的处理过程，汇编中断程序设计方法，以及**利用I/O端口扩展外部中断源的方法**。

内容提要

1. 中断系统概述
2. 8051微控制器的中断系统
3. 中断处理过程
4. 中断程序设计
5. IO端口扩展外部中断源

主要介绍

- 中断的概念
- 中断的作用
- 中断源
- 中断系统的功能

9.1.1 中断的概念

中断：微控制器执行程序过程中，由于**内部**或者**外部**的某种原因，要求MCU**中止**正在运行的程序，转去执行另外一段处理程序，待处理结束后，再回来继续执行被中止了的原程序。这种程序在执行过程中，由于外界的原因而被中间打断的情况称为“**中断**”。

中断是**硬件改变CPU程序运行方向**的一种技术，既和硬件有关，也和软件有关。先进的中断系统能提高MCU实时处理外界异步事件的能力。

- **主程序（调用程序）：**原来运行的程序；
- **中断服务程序：** 中断之后执行的处理程序；
- **断点：**主程序被中断的位置（地址）；
- **中断源：**引起中断的原因或能发出中断申请的来源；
- **中断请求：**中断源要求服务的请求。

9.1.1 中断的概念

中断程序与子程序：

中断程序与子程序有啥区别呢？

- **调用中断服务程序**的过程类似于程序设计中调用子程序的过程。在执行程序的过程中，由于中断源发出中断请求，MCU响应后转去执行一段中断服务程序，相当于在中断发生时刻调用一个子程序。

- 不同点：**子程序的调用是程序预先设计安排好的**；而中断源发出中断请求是**随机的**，所以中断服务程序的调用是**无法预知的**，中断服务程序的调用过程是计算机系统硬件自动完成的。

9.1.2 中断的作用

1. 分时操作

利用中断可以很好协调**快速CPU与慢速外设互相配合**高效地工作。 CPU启动速度较慢的外设后，外设要经过比较长的时间完成一个任务，完成任务后需要CPU服务时，向CPU请求中断。 CPU可命令多个外设同时工作，并在发生中断时及时为外设提供服务。

例如：外设**ADC转换结束**或**有按键按下**时向CPU请求中断，CPU中断正在执行的主程序转去执行中断服务程序，**读取AD转换结果**或**扫描按键得到键值**，中断处理完成后CPU返回主程序继续执行原程序，外设继续工作。

2. 实时处理

在微机系统中，依靠中断技术能实现实时控制。即控制对象发出实时操作请求时，通过中断可以使CPU快速地作出响应。

3. 故障处理

如发生电源掉电、通信故障等，可以向CPU请求中断，以便及时作出处理。

9.1.3 中断源

发出中断请求的内部功能模块或外部设备来源一般统称为“**中断源**”。

1. 外部设备

单片机的输入/输出设备，如A/D、打印机、按键等，可通过接口电路向CPU申请中断。

2. 内部设备

定时器定时时间到或**计数个数到**请求中断；**串行口发送**完一帧或**接收**到一帧数据的中断请求。

3. 故障源

如**掉电**故障、**硬件**故障、**运算错误**、**程序运行**故障等请求中断，使得CPU能够以中断方式及时处理发生的故障。

4. 控制对象

微控制器的控制对象，如电压、温度等检测量**超过上下限**时，继电器、开关**动作**时，向CPU请求中断。

9.1.4 中断系统的功能

1. 中断的允许和禁止

即**开中断**和**关中断**，根据需要能够**用指令控制中断的开放**和**关闭**。

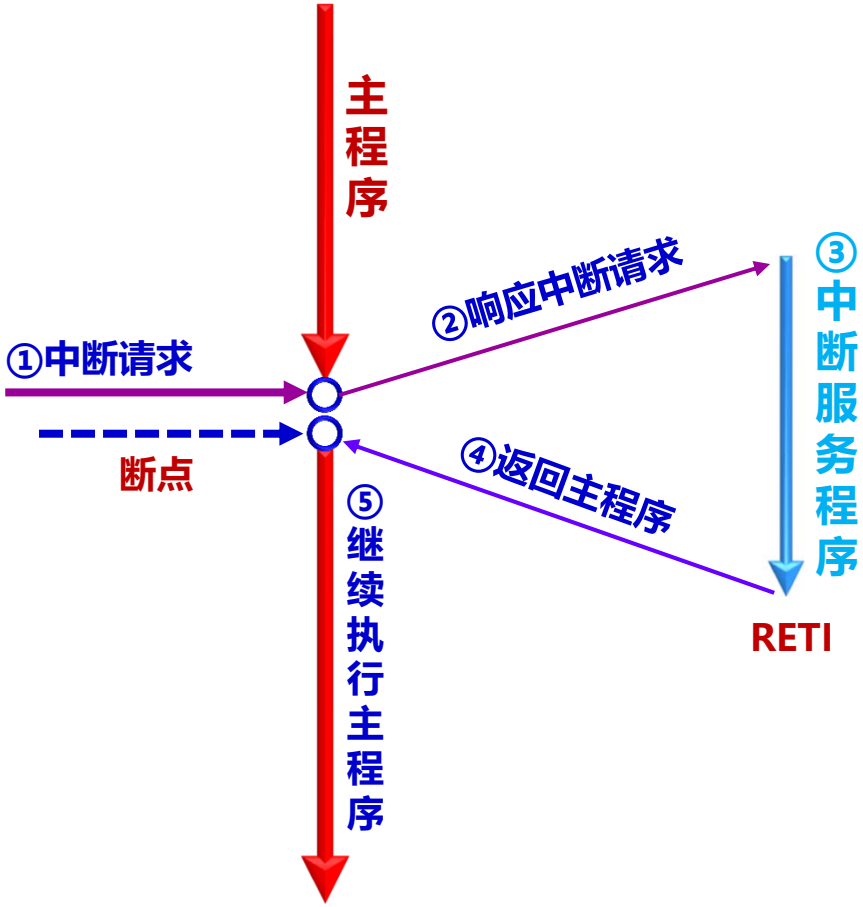
只有在开中断情况下，CPU才能响应中断源的请求。

2. 中断响应和返回

- **保护断点**：CPU在现行指令执行完毕、转到中断程序前，把断点处的PC值（即**下一条指令的地址**）压入堆栈（由**硬件自动完成**）。
- **保护现场**：用户在编写中断服务程序时，须对中断程序中用到的**工作寄存器**和**SFR**等内容进行保护。
- **恢复现场**：在中断返回前，恢复保护的内容。
- **恢复断点**：中断服务程序的最后一条指令必须为**中断返回指令RETI**，其功能是自动恢复断点地址送到PC，使CPU返回到断点处继续执行主程序。

9.1.4 中断系统的功能

中断响应及处理过程



9.1.4 中断系统的功能

3. 中断优先级与中断嵌套

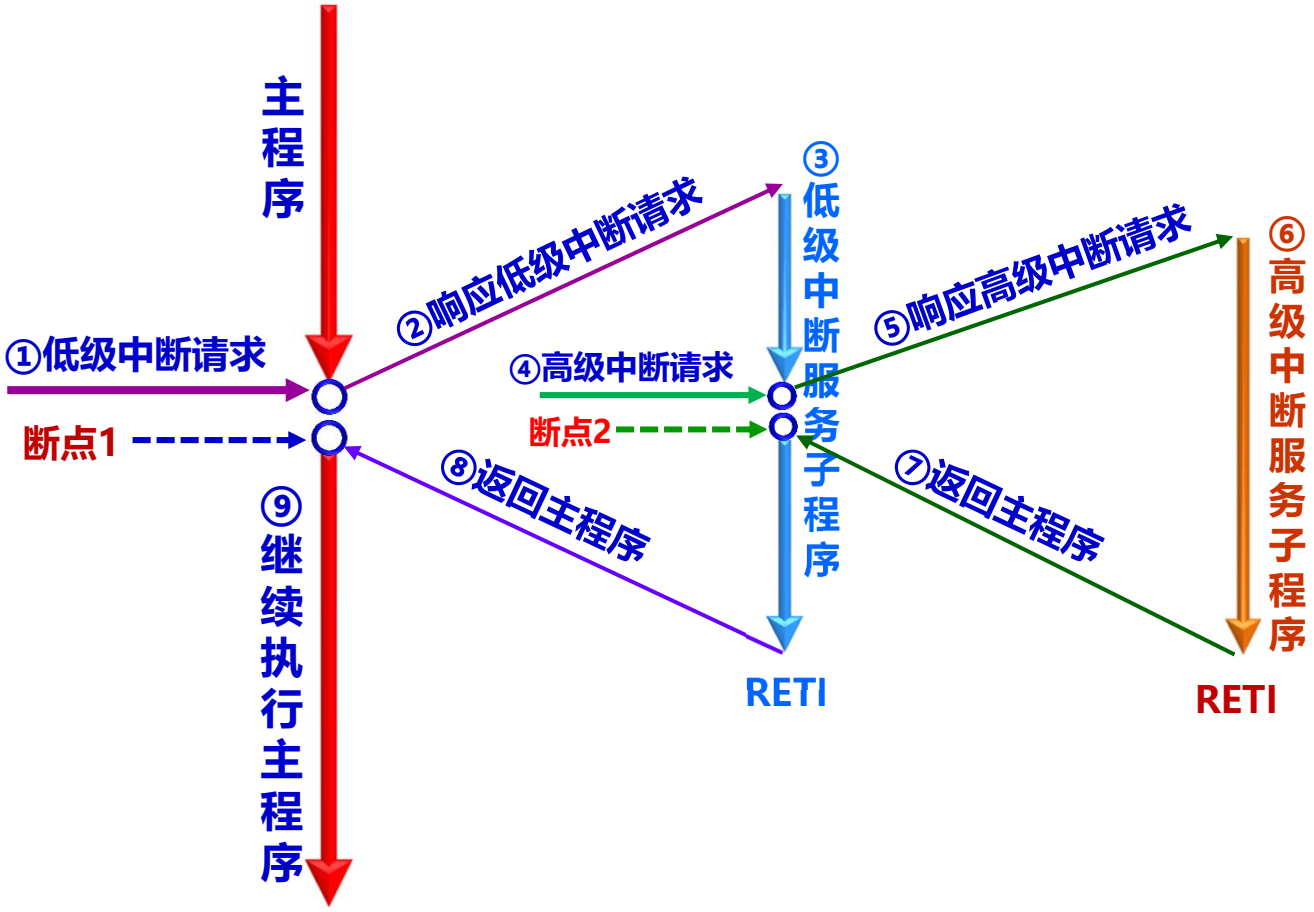
- **中断优先级：**当有多个中断源同时请求中断时，CPU会根据各中断源的优先级别，首先响应优先级高的中断请求，当该中断程序执行完毕返回主程序后，再响应优先权较低的中断源，实现中断优先级的控制。这个过程是MCU的中断系统自动完成的。

- **中断嵌套：**当CPU正在执行低级的中断服务程序时，若有高级中断源申请中断，则能够停下低级中断服务程序转去执行高级中断源的服务程序，实现中断嵌套，并能逐级正确返回。

如果新的中断请求的优先级与正在处理的中断是同级别或低一级，则CPU暂时不响应这个新中断申请，直到正在处理的中断服务程序执行完毕，才会给予响应。

9.1.4 中断系统的功能

中断嵌套（响应及处理过程）



内容提要

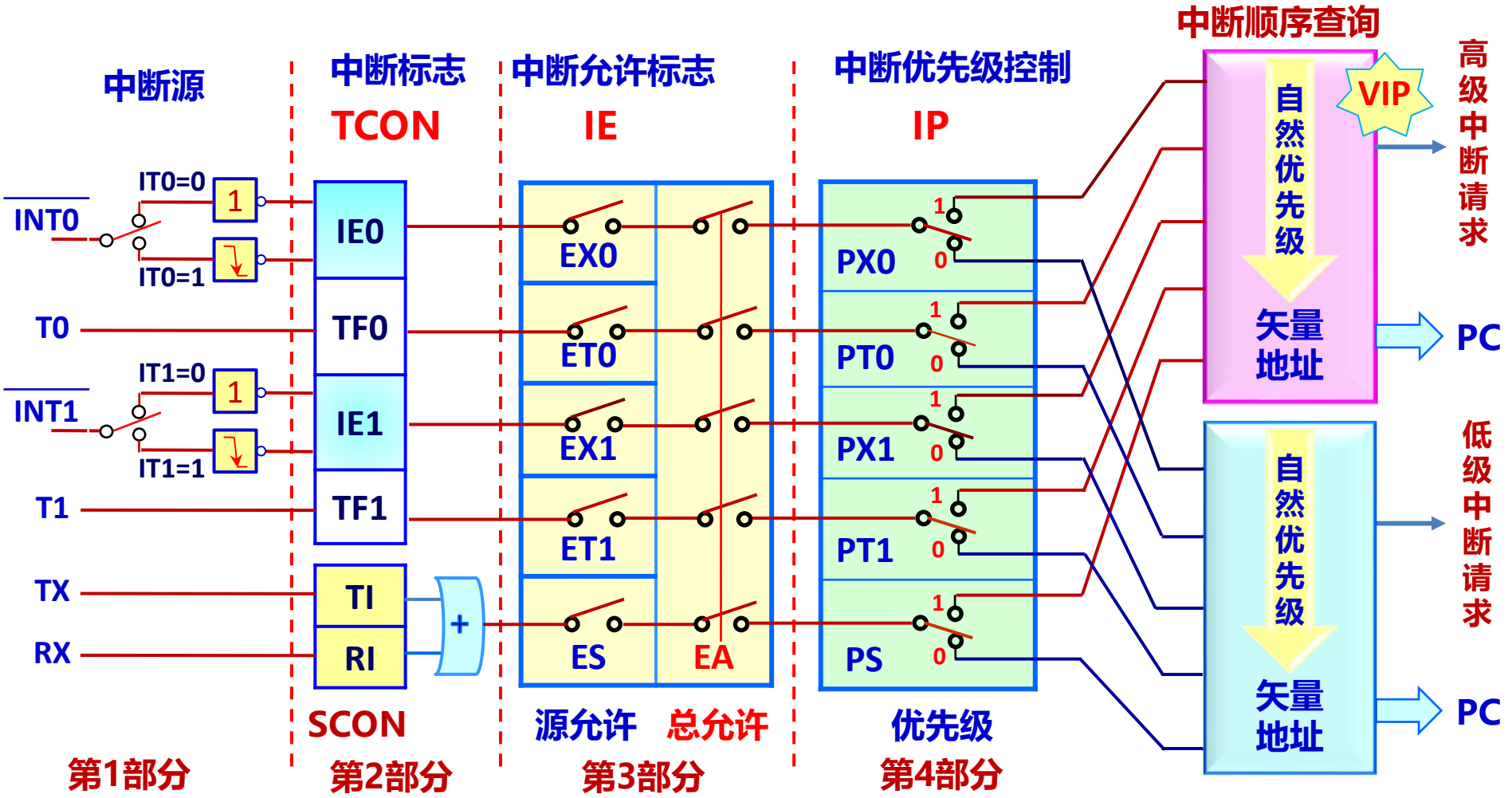
1. 中断系统概述
2. 8051微控制器的中断系统
3. 中断处理过程
4. 中断程序设计
5. IO端口扩展外部中断源

主要介绍8051微控制器中断系统的组成结构，包括5个中断源、与中断系统相关的4个特殊功能寄存器以及功能。

9.2.1 中断系统的结构

1. 中断系统结构图

由5个中断源、4个SFR（TCON、SCON、IE、IP）、中断查询逻辑电路等组成。



9.2.1 中断系统的结构

2. 中断源

8051微控制器有**5个中断源**：外部中断（2个）、定时器/计数器中断（2个）和串行口中断（1个）。

(1) 外部中断

- **两个外部中断源：/INT0**和**/INT1**，外部中断请求信号分别从引脚/INT0 (**P3.2**) 和/INT1 (**P3.3**) 引入。外设如按键、掉电检测电路信号等可以请求外部中断。
- **二种中断触发方式：低电平触发**和**下降沿触发**，可通过编程进行选择。

(2) 定时中断

- **T0、T1溢出中断：**当T0、T1定时**时间到**或发生溢出时，向CPU请求中断。

(3) 串行口中断

当串行口**发送完**一个字节数据或**接收到**一个字节数据时，产生中断请求。

3. 中断入口

5个中断源的**中断入口地址是固定的**。当CPU响应某中断源的中断请求后，**硬件自动**将断点地址压入堆栈保护，并将此中断源的**中断入口地址赋给PC**，使CPU执行该中断的中断服务程序。

中 断 源	入口地址
外部中断0（/INT0）	0003H
定时器/计数器0（T0）	000BH
外部中断1（/INT1）	0013H
定时器/计数器1（T1）	001BH
串行口中断（TX或RX）	0023H

5个中断源的6个中断请求信号，分别为：/INT0、T0、 /INT1、T1、TX、RX，其中TX和RX是串行口中断的发送中断请求和接收中断请求，这两个请求共用一个中断源。

9.2.2 中断的控制

8051微控制器中与中断系统有关的SFR有TCON、SCON、IE、IP。

1. 定时器/计数器控制寄存器TCON（Timer Control）

TCON的字节地址为88H，是可以位寻址的SFR。

	7	6	5	4	3	2	1	0
位符号	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
英文注释	Timer1 Overflow	Timer1 Run	Timer0 Overflow	Timer0 Run	Interrupt External 1 flag	Interrupt 1 Type control bit	Interrupt External 0 flag	Interrupt 0 Type control bit

- TF0：T0溢出标志，溢出时由硬件置1，并请求中断，CPU响应后，由硬件自动将TF0清0；查询方式时，要用软件清0。
- TF1：T1溢出标志，溢出时由硬件置1，并请求中断，CPU响应后，由硬件自动将TF1清0；查询方式时，要用软件清0。

9.2.2 中断的控制

1. 定时器/计数器控制寄存器TCON（Timer Control）

	7	6	5	4	3	2	1	0
位符号	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
英文注释	Timer1 Overflow	Timer1 Run	Timer0 Overflow	Timer0 Run	Interrupt External 1 flag	Interrupt 1 Type control bit	Interrupt External 0 flag	Interrupt 0 Type control bit

- **IE0：$\overline{\text{INT0}}$中断标志**，发生$\overline{\text{INT0}}$中断时，硬件置IE0为1，并向CPU请求中断。
- **IE1：$\overline{\text{INT1}}$中断标志**，发生$\overline{\text{INT1}}$中断时，硬件置IE1为1，并向CPU请求中断。
- **IT0：外部中断0的触发方式选择位**，设置为“0”时，表示选择**低电平触发**；设置为“1”时，表示选择**下降沿触发（边沿触发）**。
- **IT1：外部中断1的触发方式选择位**，设置为“0”时，表示选择**低电平触发**；设置为“1”时，表示选择**下降沿触发（边沿触发）**。

实际使用时，常采用下降沿触发方式，低电平触发很少使用。

1. 定时器/计数器控制寄存器TCON（Timer Control）

外部中断的检测过程：

CPU每个机器周期检测一次$\overline{\text{INT0}}$、$\overline{\text{INT1}}$引脚，对中断请求信号的要求为：

- **下降沿触发方式：**$\overline{\text{INT0}}$、$\overline{\text{INT1}}$引脚上中断请求信号的**高、低电平至少应各保持一个机器周期**；
- **低电平触发方式：**$\overline{\text{INT0}}$、$\overline{\text{INT1}}$引脚上中断请求信号的**低电平应保持到CPU响应中断为止**。

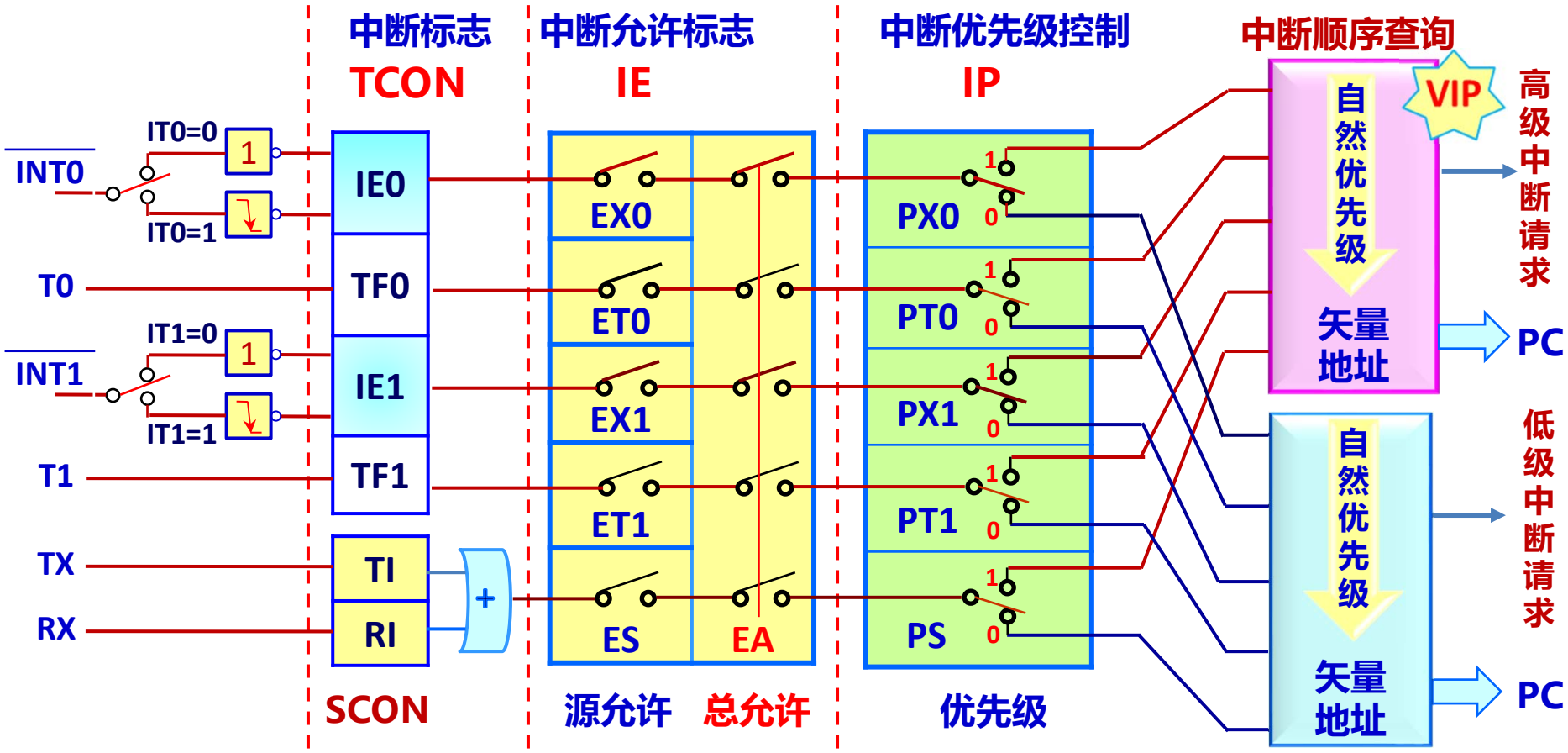
2. 串行口控制寄存器SCON（Serial Control）

SCON的字节地址为**98H**，可**位寻址**，用于串行口的操作管理，其中两位为串行口的中断标志**RI**和**TI**。

	7	6	5	4	3	2	1	0
位符号	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
英文注释	Serial Mode bit0	Serial Mode bit1	Serial Mode bit2	Receive Enable	Transmit bit 8	Receive bit 8	Transmit Interrupt flag	Receive Interrupt flag

- **TI**：发送中断标志。当串行口**发送完**一帧数据时，**硬件自动将TI置位。**
- **RI**：接收中断标志。当串行口**接收到**一帧数据时，**硬件自动将RI置位。**

9.2.2 中断的控制



当中断源有请求时，会在相关的SFR建立相应的中断标志位，表示该中断源请求了中断（伴随中断请求在SFR中设置标记）。 6个中断请求对应的中断标志分别为IE0、TF0、IE1、TF1、TI、RI，其中TI和RI这两个标志经一个“或门”后，输出串口的中断请求。

9.2.2 中断的控制

中断标志如何清除的

关于中断标志的清除

- **T0、T1中断标志：**在中断工作方式下，TF0、TF1一直保持到CPU响应中断，并由**硬件自动清0**；如果采用查询方式，则此标志需要软件清0。
- **外部中断标志：**对于下降沿触发方式，IE0、IE1一直保持到CPU响应中断，并由**硬件自动清除**。如果是低电平触发方式，只有当中断引脚变为高电平时，才会消除。
- **串行口中断标志：**不论在中断方式还是查询方式，均必须通过**软件清除**。

9.2.2 中断的控制

总结：中断标志的产生和清除

中断名称	中断入口	中断源	中断标志	中断标志建立条件	标志清除
外部中断0	0003H	$\overline{\text{INT0}}$	IE0	符合触发条件时	硬件自动清除（下降沿） 引脚变为高电平（低电平）
外部中断1	0013H	$\overline{\text{INT1}}$	IE1		
外部中断触发方式：下降沿、低电平；触发方式选择：IT0/IT1					
T0中断	000BH	T0	TF0	T0溢出	中断方式：硬件自动清除 查询方式：软件清除
T1中断	001BH	T1	TF1	T1溢出	
串行口中断	0023H	TX	TI	发送完一帧数据	软件清除
		RX	RI	接收到一帧数据	

3. 中断允许控制寄存器 IE（Interrupt Enable）

中断允许控制寄存器 IE的字节地址为A8H，是可位寻址的SFR。IE用于管理各中断源中断的允许与禁止。

	7	6	5	4	3	2	1	0
位符号	EA	—	—	ES	ET1	EX1	ET0	EX0
英文注释	Enable All interrupts			Enable Serial interrupt	Enable Timer1 interrupt	Enable External 1 interrupt	Enable Timer0 interrupt	Enable External 0 interrupt

微控制器复位后，IE中各位均被清0，即禁止所有中断。

3. 中断允许控制寄存器 IE（Interrupt Enable）

- **EA： CPU中断允许位。**

  EA = 1，CPU开中断，结合各中断源的中断允许位，确定各中断源的允许和禁止。

  EA = 0，CPU关中断，禁止响应任何中断请求。

- **ES：串行口中断允许位。**

  ES = 1，允许串行口的接收和发送中断；ES = 0，禁止串行口中断。

- **ET1/ET0：T1/T0中断允许位。**

  ET1/ET0 = 1，允许T1/T0中断；ET1/ET0 = 0，禁止T1/T0中断。

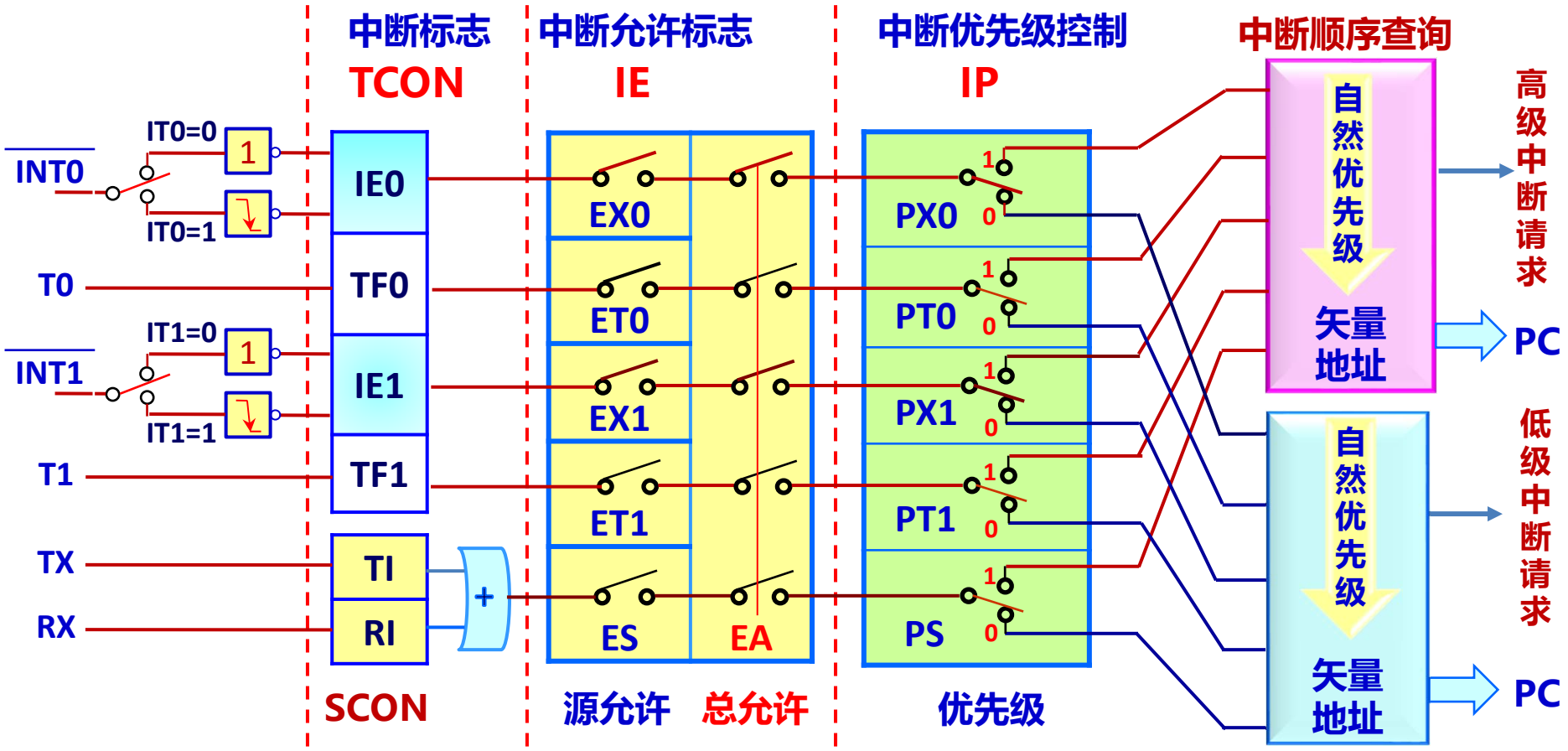
- **EX1/EX0：INT1/INT0中断允许位。**

  $\overline{\text{INT1}}/\overline{\text{INT0}}$=1，允许$\overline{\text{INT1}}/\overline{\text{INT0}}$中断； $\overline{\text{INT1}}/\overline{\text{INT0}}$=0，禁止$\overline{\text{INT1}}/\overline{\text{INT0}}$中断。

通过对IE的设置，实现对各中断源中断允许和禁止的控制。

8051微控制器能实行**二级控制**，**EA**是总控制位，各中断源有分控制位。只有当总控制位EA=1，即**CPU中断开放时，对各分控制位的设置(开放或禁止相应中断源)才有效**。

9.2.2 中断的控制



当**EA=0**时（“总允许开关”断开），所有中断源的中断标志无法传递到CPU而被禁止；
当**EA=1**时（“总允许开关”闭合），这时CPU中断开放；
但各中断源的请求标志是否能传递到CPU，还取决于各中断源分控制位的设置，当某个中断源的分控制位=1，则对应的“源允许开关”闭合，该中断才被允许；反之，则被禁止。

9.2.2 中断的控制

4. 中断优先级寄存器IP（Interrupt Priority）

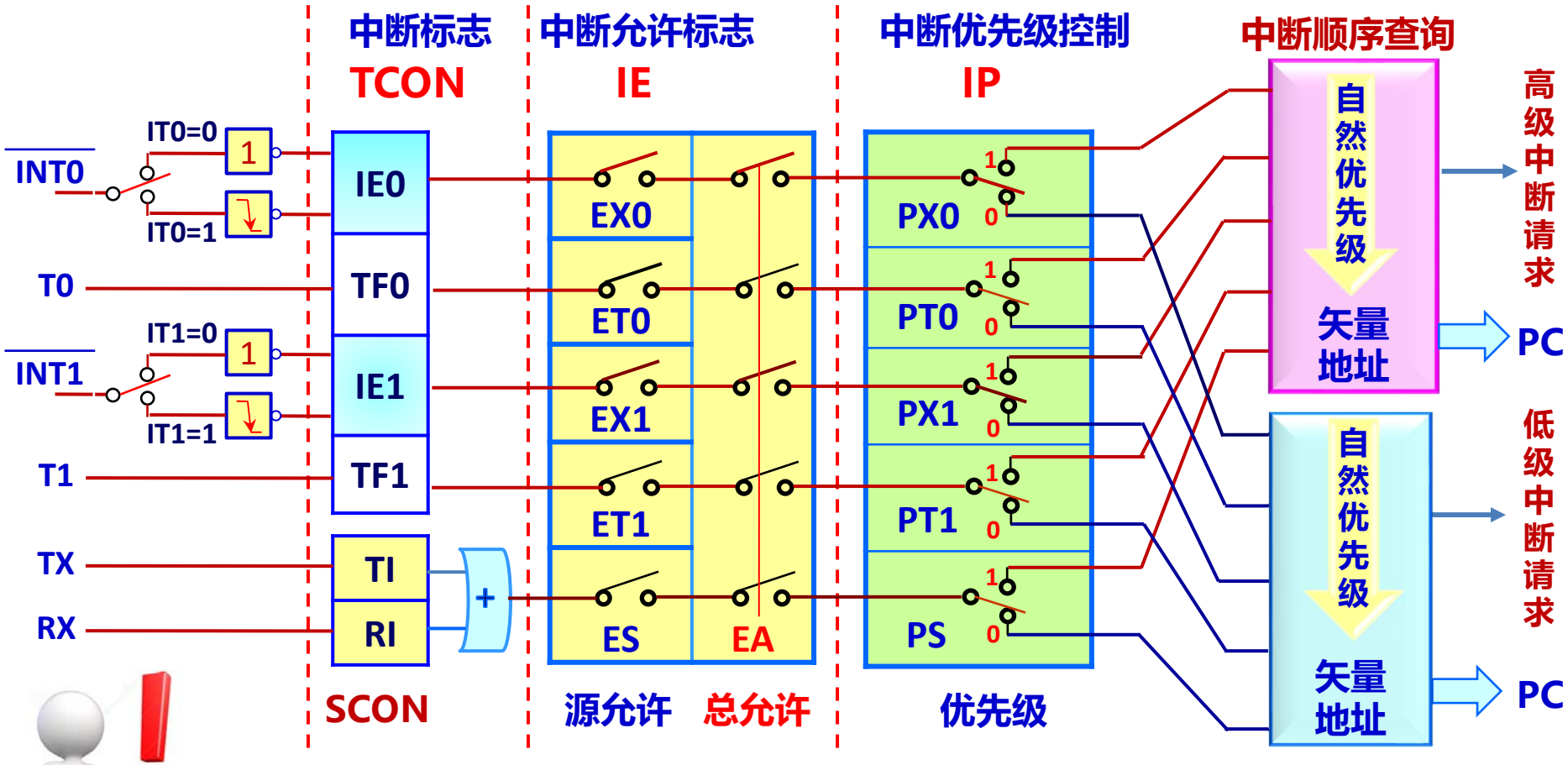
字节地址为**B8H**，可位寻址。用于管理各中断源的优先级别。

	7	6	5	4	3	2	1	0
位符号	—	—	—	PS	PT1	PX1	PT0	PX0
英文注释				Serial Interrupt Priority	Timer1 Interrupt Priority	External 1 Interrupt Priority	Timer0 Interrupt Priority	External 0 Interrupt Priority

微控制器复位后，IP内容为0，所有中断源均被设置为低优先级中断。

位符号	中断优先级控制位功能说明
PS	串行口中断优先级控制位。PS=1，选择高优先级； PS=0，选择低优先级。
PT1	T1中断优先级控制位。 PT1=1，选择高优先级； PT1=0，选择低优先级。
PX1	INT1中断优先级控制位。 PX1=1，选择高优先级； PX1=0，选择低优先级。
PT0	T0中断优先级控制位。 PT0=1，选择高优先级； PT0=0，选择低优先级。
PX0	INT0中断优先级控制位。 PX0=1，选择高优先级； PX0=0，选择低优先级。

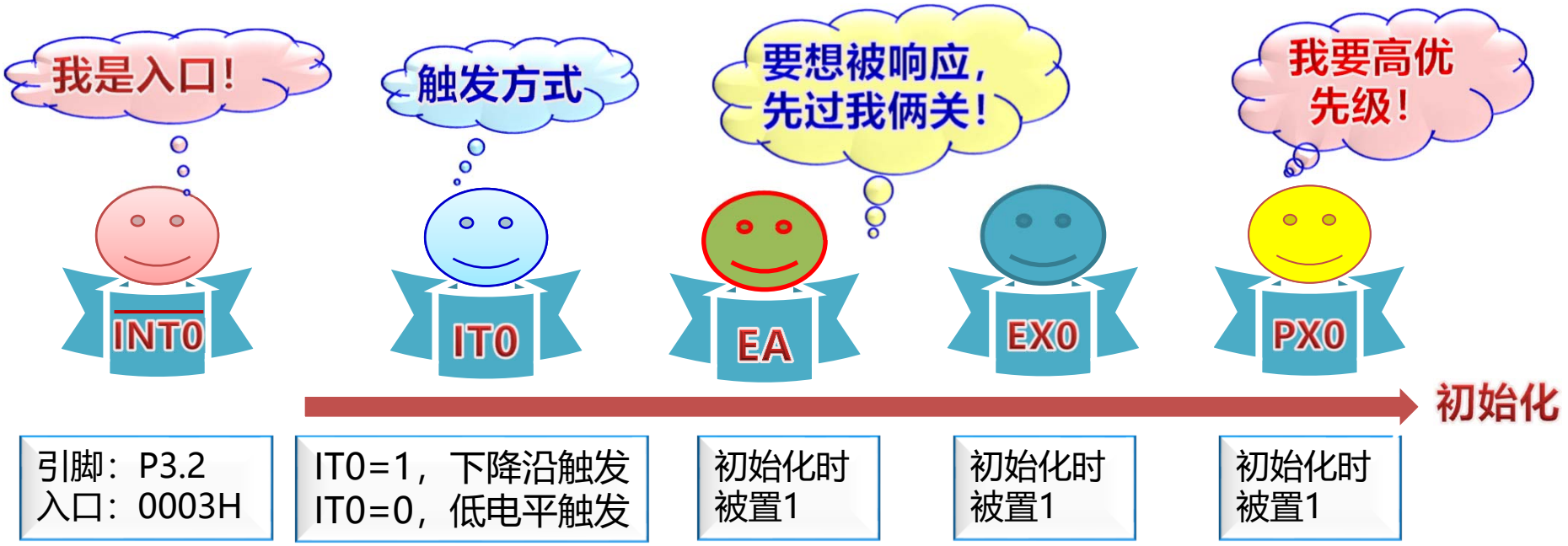
9.2.2 中断的控制



与两个优先级对应，8051中断系统内部有两个不可寻址的“**优先级标志**”，CPU响应了哪个级别的中断请求，相应级别的“**优先级标志**”被置位，以此来指示CPU在执行高级或低级中断服务程序，从而实现**中断嵌套**的控制。

9.2.2 中断的控制

以$\overline{\text{INT0}}$的中断服务被CPU响应为例



4. 中断优先级寄存器IP（Interrupt Priority）

例：若一个微控制器系统有2个中断源，一个是T0的10ms定时中断，一个是外部$\overline{\text{INT0}}$的按键中断，希望T0中断在任何时刻能及时响应，则可以设置T0中断为高优先级中断，$\overline{\text{INT0}}$中断为低优先级中断。

即 SETB PT0

CLR PX0

这样，T0中断能打断INT0的中断服务，反之则不能。

对IP编程可把5个中断设置为高低两个优先级，它们遵循:

- **低级中断能被高级中断打断，但不能被同级中断打断；**
- **高级中断不能被任何中断打断，一定要返回主程序并再执行一条指令后，才能响应新的中断请求。**

9.2.2 中断的控制

同级优先级中断请求，哪个优先响应呢?

当CPU同时接收到**几个同优先级**的中断请求时，哪一个先得到响应，取决于CPU中断系统内部的查询顺序。这相当于在每个优先级内，还存在一个辅助优先级结构，其**优先顺序**如下：

中断源	中断优先级
1 外部中断0	最高
2 定时器T0中断	↓
3 外部中断1	
4 定时器T1中断	
5 串行口中断	最低

填空题 2分

设置

8051微控制器的外部中断有___[填空1]____和____[填空2]____两种触发方式。

作答

正常使用填空题需3.0以上版本雨课堂

单选题 1分

设置

微控制器响应中断时，保护现场的工作_____。

A 由CPU自动完成

B 在中断响应时完成

C 由中断服务程序完成

D 在主程序中完成

提交

单选题 1分

设置

处于同一级别的5个中断源同时请求中断时，CPU响应中断的次序为_____。

A 串行口、T1、/INT1、T0、/INT0

B /INT0、 T0、 /INT1、 T1、串行口

C 串行口、 /INT1 、T1、 /INT0、T0

D T0、 /INT0 、T1、 /INT1、串行口

提交

内容提要

1. 中断系统概述
2. 8051微控制器的中断系统
3. 中断处理过程
4. 中断程序设计
5. IO端口扩展外部中断源

主要介绍

- 中断响应的自主操作
- 中断响应条件
- 中断响应过程
- 中断响应时间
- 响应中断与调用子程序的异同

9.3.1 中断响应的自主操作过程

中断响应的自主操作：

在中断检测和中断响应过程中，由中断系统硬件**自动完成**的操作。

1. 中断请求的自动查询

8051微控制器中，中断系统在**每个机器周期的S6状态**查询各中断源是否有请求（即相应的中断标志是否为1），并按优先级管理规则处理同时请求的中断源，且在**下一个机器周期的S1状态响应最高级中断请求**。

有以下情况则除外：

- CPU正在处理**相同或更高优先级中断**；
- 正在执行多机器周期指令，并还未执行到最后一个机器周期；
- 正在执行**RETI** 指令或**读/写IE**、**IP**的指令，则要**延后一条指令**予以响应。

9.3.1 中断响应的自主操作过程

2. 中断响应时的自主操作

- 置位相应的“优先级标志”，以标明所响应中断的优先级别；
- 中断源标志清0（TI、RI除外）；
- 中断断点地址装入堆栈保护（不保护寄存器等）；
- 中断入口地址装入PC，以便使程序转到中断入口地址处。

3. 中断返回时的自主操作

- “优先级标志”清0；
- 断点地址送入PC，以便使程序返回到断点处。

特别注意和RET的区别

9.3.2 中断响应条件

中断响应：中断源发出中断请求、在满足CPU中断响应条件之后，CPU处理中断请求的过程。

中断响应的6个基本条件：

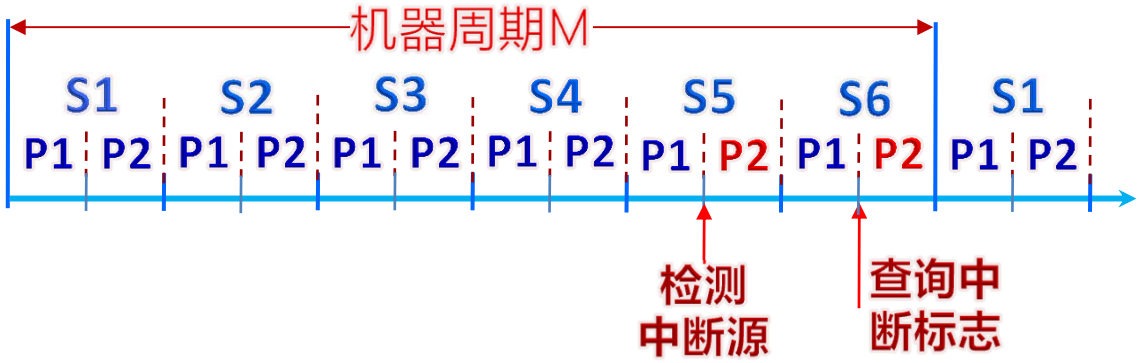
1）有中断源发出**中断请求**；
2）CPU中断允许位（总允许）置位，即**EA = 1**；
3）申请中断的中断源的**中断允许位**（源允许）置位，即允许该中断源中断。

CPU在每个机器周期（S6状态），按优先次序查询各中断标志，找到所有有效的中断请求，并对其优先级排队，如果满足：

4）无同级或高级中断正在服务；
5）现行指令已执行完毕；
6）若执行指令为RETI或是读/写IE、IP指令时，则该指令的下一条指令也执行完毕。

CPU便在紧接着的下一个机器周期响应中断，否则将丢弃中断查询结果。

9.3.3 中断响应过程



中断响应过程：

1）CPU在**每个机器周期**(**S5P2**)**检测中断源**，并按优先级别和自然顺序**查询**(**S6P2**)各中断标志。若查询到有效的中断标志（中断标志为1），按优先级别进行处理，即响应中断；

2）自动设置“**优先级标志**”为1，即指出CPU当前正在处理的中断优先级，以阻断同级或低级中断请求；

3）自动**清除中断标志**（TI和RI除外）；

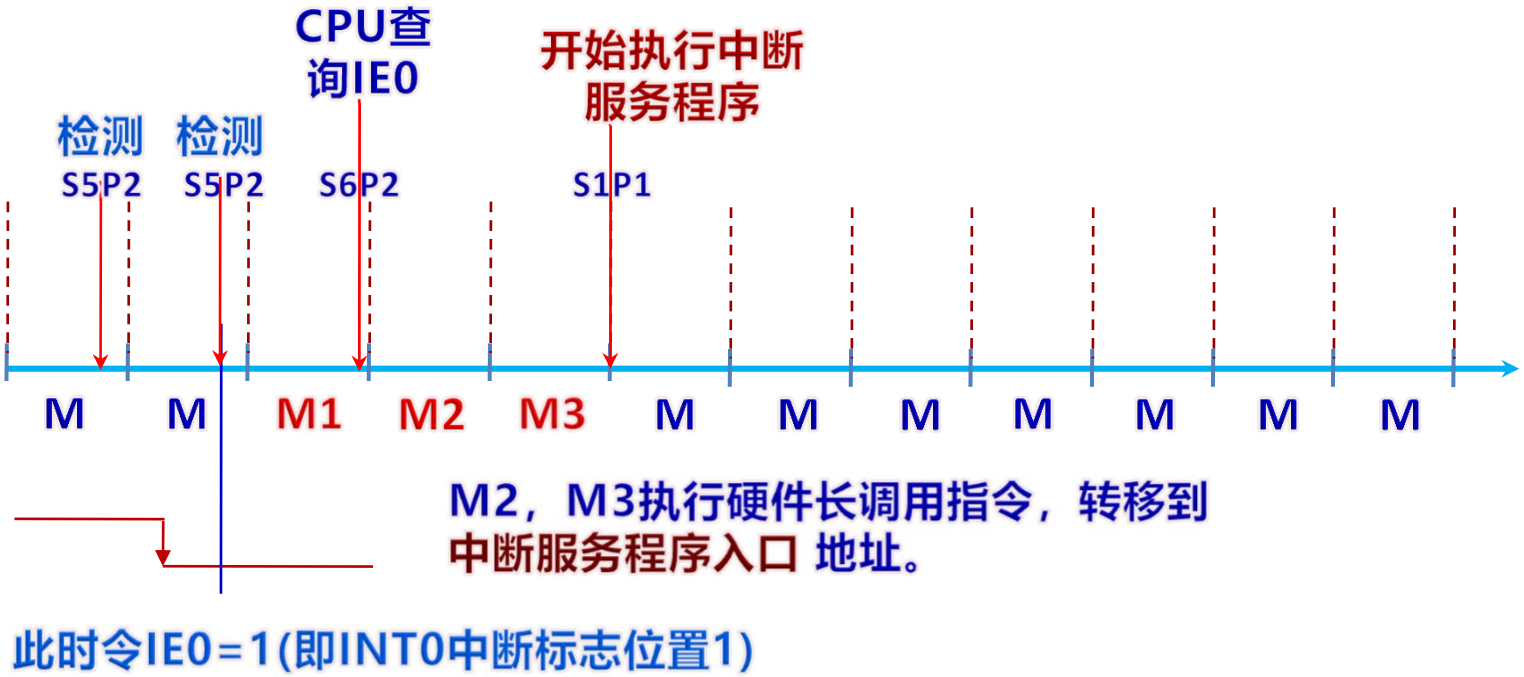
4）自动**保护断点**，即将现行PC值（即断点地址）压入堆栈，并根据中断源把相应的中断程序入口地址装入PC；

5）**执行中断服务程序**，直至遇到RETI指令为止；

6）RETI指令**清除**“**优先级标志**”；从堆栈中弹出断点地址给PC，使CPU**返回到中断处**，继续执行主程序。

9.3.4 中断响应时间

中断的响应时间：最短为3个机器周期，最长为8个机器周期。



3个机器周期情况：查询中断标志位占1个机器周期，硬件自动保护断点地址（相当于自动插入一条长调用LCALL指令）需要2个机器周期。

9.3.4 中断响应时间

8个机器周期情况：如果检测到中断标志位时，CPU正在执行**RETI指令或访问IE、IP的指令**（2个机器周期），则执行该指令后，还必须再执行一条指令才能响应中断。若再执行的一条指令恰好为**乘法或除法指令**（4个机器周期），再加上自动保护断点地址的2个机器周期，总共需要8个机器周期。

如果存在多个中断源，而且CPU正在处理高级或同级中断，那么中断响应的时间还取决于正在执行的中断服务程序的长短。

9.3.5 中断响应与调用子程序的异同点

1. 响应中断与调用子程序的相同点

- **中断当前正在执行的程序**，转去执行子程序或中断服务程序。
- 由**硬件自动把断点地址压入堆栈**。
- 子程序和中断服务程序的**现场保护和恢复**，**都需要编写程序实现**。执行中断程序和子程序的返回指令时，都会**自动返回到断点**处，继续原程序的执行。
- 都可以**实现嵌套**。中断程序可以实现**2级嵌套**，子程序可以更**多级的嵌套**。

2. 响应中断与调用子程序的差别

- 中断请求是**随机的**，在程序执行的任何时刻都有可能发生；而子程序的调用是由程序设计**安排**的。
- 响应中断后，转去执行存放在相应**中断入口地址处**的中断服务程序，而子程序的地址由程序设计时安排的。
- 中断响应是**受控的**，其响应时间会受一些因素影响；子程序响应时间固定。
- 中断服务程序的返回指令是**RETI**，子程序的返回指令是**RET**，两者不能互换。

单选题 1分

设置

8051微控制器CPU开中断的指令是_____。

A SETB EA

B SETB ES

C CLR EA

D SETB EXO

提交

单选题 1分

设置

执行中断返回指令时，从堆栈顶部弹出的地址送给_____。

A A

B Cy

C PC

D DPTR

提交

填空题 2分

设置

1. 8051MCU响应中断时，断点地址是_____[填空1]_____保护的，现场(寄存器等内容)则需要_____[填空2]_____保护。

作答

内容提要

1. 中断系统概述
2. 8051微控制器的中断系统
3. 中断处理过程
4. 中断程序设计
5. IO端口扩展外部中断源

9.4.1 中断初始化

主要介绍中断的初始化、汇编中断服务程序设计，以及中断程序设计举例。

中断初始化：对中断相关**SFR**（TCON、SCON、IE和IP）的设置，使得MCU的中断系统能够按照设置要求对中断源进行管理和控制。

中断初始化主要包括以下内容：

- **设置CPU中断控制位**，选择CPU中断的允许或禁止；
- **设置各中断源的中断控制位**，选择各中断源中断的允许或禁止；
- **设置各中断源的中断优先级别**，选择高优先级或低优先级；
- **设置外部中断请求的触发方式**；
- **对相关中断源的初始化**（如定时器/计数器或串行口的初始化）。

9.4.2 汇编中断服务程序设计

1. 保护现场和恢复现场

堆栈保护、切换工作寄存器组、存储器保护

- 中断服务程序的**保护现场**和**恢复现场**的方法，与子程序类似（**3种方法**）。
- 若要在执行当前中断程序时禁止高优先级的中断（即**不允许中断嵌套**），则在进入中断程序后先关闭CPU中断（令EA=0），在中断返回前再予以开放。

2. 中断程序的安排

- 各中断入口地址之间只**间隔8个字节**，一般的中断服务程序容纳不下。
- **最常用的方法：**在中断入口地址处安排一条**无条件转移指令**，使程序跳转到用户安排在其它ROM区域的中断服务程序。

中断服务程序的设计原则：程序尽量简短、执行时间尽量短

```
        ORG   0000H
        LJMP  MAIN

        ORG   0003H
        LJMP  INT0SUB

        ORG   000BH
        LJMP  T0SUB

        ORG   0013H
        LJMP  INT1SUB

        ORG   001BH
        LJMP  T1SUB

        ORG   0023H
        LJMP  SPSUB

        ORG   0030H
MAIN:   MOV   SP, #5FH
        ……
```

9.4.2 汇编中断服务程序设计

例5-1：假设程序中有2个中断源：INT0和T1，要求INT0为高优先级。

主程序：

```
        ORG   0000H
        LJMP  MAIN      ;跳转到主程序

        ORG   0003H     ;INT0中断入口地址
        LJMP  INT0SUB   ;跳转到实际INT0中断服务程序存放空间
          ……
        ORG   001BH     ;T1中断入口地址
        LJMP  T1SUB     ;跳转到实际T1中断服务程序存放空间
          ……
        ORG   0030H     ;  实际主程序存放区
MAIN:   MOV   SP,#5FH   ;  设置堆栈区
          ……
        SETB  IT0       ;选择INT0为下降沿触发方式
        SETB  EA        ;CPU开中断
        SETB  EX0       ;INT0开中断
        SETB  ET1       ;T1开中断
        SETB  PX0       ;设置INT0为高优先级
          ……
        SJMP  $         ;模拟主程序
```

9.4.2 汇编中断服务程序设计

中断程序：

```
            ORG     0800H     ；INT0中断服务程序存放区
INT0SUB:    PUSH    ACC
            PUSH    PSW
            ……
            ……

            POP     PSW
            POP     ACC
            RETI              ；中断返回

T1SUB:      PUSH    ACC       ；定时器T1中断服务程序
            PUSH    PSW
            ……
            ……
            POP     PSW
            POP     ACC
            RETI              ；中断返回

            END
```

9.4.3 中断程序设计举例

例5-2：试编写程序，将外部RAM 3000H开始的20H个单元的数据，传送到内部RAM 40H开始的20H个单元中。允许外部中断/INT0，下降沿触发。

主程序（汇编）：

```
        ORG   0000H
        LJMP  MAIN

        ORG   0003H
        LJMP  INT0SUB

        ORG   0040H
MAIN:   MOV   SP, #6FH          ;更改堆栈区域
        SETB  EA                ;CPU中断允许
        SETB  EX0               ;INT0中断允许
        SETB  IT0               ;下降沿触发
        MOV   DPTR, #3000H      ;外部数据存储器地址指针
        MOV   R0, #40H          ;内部数据存储器地址指针
        MOV   R2, #20H          ;传送的数据个数
LOOP1:  MOVX  A, @DPTR
        MOV   @R0, A
        INC   R0
        INC   DPTR
        DJNZ  R2, LOOP1
        SJMP  $
```

中断程序（汇编）：

```
; 假设中断程序要用到 A、DPTR、R0-R7，会影响标志位
        ORG   1000H
INT0SUB: PUSH  ACC
        PUSH  DPH
        PUSH  DPL
        PUSH  PSW
        SETB  RS0       ;修改工作寄存器组，
                        ;中断程序中用第1组的R7-R0
        ……

        CLR   RS0       ;恢复主程序使用的第0组工作寄存器
        POP   PSW
        POP   DPL
        POP   DPH
        POP   ACC
        RETI

        END
```

9.4.3 中断程序设计举例

主程序

```
          地址  机器码          指令
                              ORG    0000H
(PC)→     0000  020040        LJMP   MAIN
                              ORG    0003H
(PC)→     0003  021000        LJMP   INTOSUB
                              ORG    0040H
(PC)→     0040  75816F  MAIN: MOV    SP,#6FH
(PC)→     0043  D2AF          SETB   EA
(PC)→     0045  D2A8          SETB   EX0
(PC)→     0047  D288          SETB   IT0
(PC)→     0049  903000        MOV    DPTR,#3000H
(PC)→     004C  7840          MOV    R0,#40H
(PC)→     004E  7A20          MOV    R2,#20H
(PC)→     0050  E0    LOOP1:  MOVX   A,@DPTR
(PC)→     0051  F6            MOV    @R0,A
(PC)→     0052  08            INC    R0
          0053  A3            INC    DPTR
          0054  DAFA          DJNZ   R2,LOOP1
          0056  80FE          SJMP   $
```

中断服务子程序

```
          地址  机器码          指令
                              ORG    1000H
(PC)→     1000  C0E0  INTOSUB: PUSH   ACC
(PC)→     1002  C083          PUSH   DPH
(PC)→     1004  C082          PUSH   DPL
(PC)→     1006  C0D0          PUSH   PSW
(PC)→     1008  D2D3          SETB   RS0
          ……    ……            ……
(PC)→     1020  C2D3          CLR    RS0
(PC)→     1022  D0D0          POP    PSW
(PC)→     1024  D082          POP    DPL
(PC)→     1026  D083          POP    DPH
(PC)→     1028  D0E0          POP    ACC
(PC)→     102A  32            RETI
                              END
```

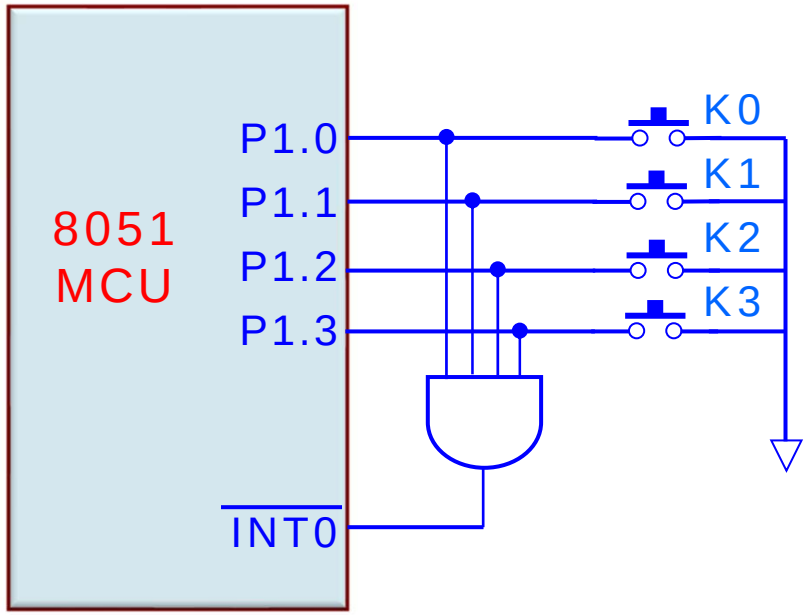
	76H	……
(SP)→	75H	PSW
(SP)→	74H	DPL
(SP)→	73H	DPH
(SP)→	72H	ACC
(SP)→	71H	00H ←PCH
(SP)→	70H	52H ←PCL
(SP)→	6FH	

内容提要

1. 中断系统概述
2. 8051微控制器的中断系统
3. 中断处理过程
4. 中断程序设计
5. IO端口扩展外部中断源

9.5 利用I/O端口扩展外部中断源

主要介绍利用I/O端口扩展外部中断源的方法和程序设计



将4个按键（外部中断源）连接到一个4“**与门**”的输入端，当其中一个或几个按键按下时，“与门”输出从高电平变为低电平，该下降沿触发MCU的INT0。在中断服务程序中，按程序设置的顺序查询4条I/O口线的状态，确定本次是哪个按键按下引起的中断，然后进行相应的按键处理。

9.5 利用I/O端口扩展外部中断源

主程序:

```
        ORG     0000H
        LJMP    MAIN
        ORG     0003H
        LJMP    INT0SUB
                ... ...
        ORG     0100H
MAIN:   SETB    EA
        SETB    EX0
        SETB    IT0
        CLR     KEYFLAG     ; 按键标志清0
LOOP:   JNB     KEYFLAG, LOOP
        CLR     KEYFLAG
        JNB     P1.0, K0    ; K0键按下，则转移
        JNB     P1.1, K1    ; K1键按下，则转移
        JNB     P1.2, K2    ; K2键按下，则转移
        JNB     P1.3, K3    ; K3键按下，则转移
        SJMP    LOOP        ; 没键按下，继续查询
```

```
K0:     ...         ; K0键处理程序
        SJMP    LOOP
K1:     ...         ; K1键处理程序
        SJMP    LOOP
K2:     ...         ; K2键处理程序
        SJMP    LOOP
K3:     ...         ; K3键处理程序
        SJMP    LOOP
```

中断程序:

```
        ORG     1000H
INT0SUB: SETB   KEYFLAG
        RETI

        END
```

单选题 1分

设置

由于各中断入口地址的间隔只有8个单元，因此通常在中断入口地址后放_____。

A MOV指令

B JMP @A+DPTR指令

C LCALL指令

D LJMP或SJMP指令

提交

单选题 1分

设置

8051微控制器执行RETI指令后，_____。

A 程序返回到响应中断时的下一条指令

B 程序返回到 LCALL的下一条指令

C 程序返回到主程序开始处

D 程序返回到响应中断时执行的一条指令

提交

THE END

THANK YOU